Physical reservoir computing in analog-digital hybrid circuit systems comprising discrete semiconductor devices

Yuki Abe[1], Kose Yoshida[1], Megumi Akai-Kasaya[2] and Tetsuya Asai[2]

[1]Graduate School of Information Science & Technology,
Hokkaido University
Kita 14 Nishi 9, Kita-ku, Sapporo 060-0814, Japan
Phone: +81 11-706-6080
E-mail: {abe.yuki.cx, yoshida.kose.r7}@ist.hokudai.ac

[2]Faculty of Information Science & Technology,
Hokkaido University
Kita 14 Nishi 9, Kita-ku, Sapporo 060-0814, Japan
Phone: +81 11-706-6080
E-mail: {akai, asai}@ist.hokudai.ac

Abstract

One of the recent trends in the field of artificial intelligence is reservoir computing (RC). RC uses a recurrent neural network, and the main feature of the model is its fixed internal structure. Owing to this fixed structure, the network of an RC can be replaced with a dynamical system, called a "physical reservoir." This report describes a design and benchmarks using a physical reservoir computing device. We physically implemented a physical reservoir computing device using an electrical circuit, ran a time-series prediction benchmark on the device, and found a significant deviation from the simulation results. Through investigations, it was revealed that 8-bit quantization caused the deviation.

1. Introduction

The demands of machine learning technology have recently been growing. The development of the Internet has made our lives more convenient, enabling users to collect huge numbers of data, which has been labeled "big data." By analyzing big data, we can predict and suggest human behaviors. Therefore, machine learning technology is trending these days. However, software-based machine learning, which depends on both a CPU and a GPU, requires a large number of resources (computational resources, energy resources, and capital)[1]. To solve this problem, hardware-based machine learning is proposed. In addition, processing through dynamics, i.e., physical reservoir computing, is one of the keys to this field[2]. Hence, there are many studies regarding physical reservoir computing, such as an MRAM device and robotics arm[3, 4]. In this study, we physically implemented a physical reservoir computing device using electronic circuits proposed in previous studies[5]. In addition, we combined the device with a learning device to design a low-power AI device[6]. Then, we evaluated and compared its simulation performance with that of an actual device.

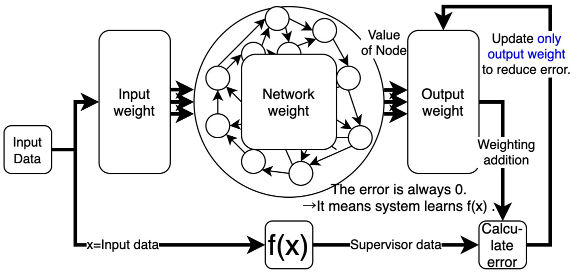


Figure 1: Concept diagram of RC

2. Reservoir computing

2.1 Outline

First, we describe how reservoir computing differs from existing RNN-based models. An RNN is a family of artificial neural network models, which include models composed of network structures with recurrent connections. In addition, error back propagation is widely used as an RNN learning method. However, applying error back propagation to a whole network requires many computational resources. To solve this problem, the echo state network (ESN) and liquid state machine (LSM) were proposed, without changing the features of the existing RNN[7, 8, 9]. These models differ from the existing models in that only the output weights are updated, as shown in Figure 1[10]. In addition, its fixed network, which works by storing previous states of the network, is called a reservoir, and models using it are called reservoir computing models. Because of their fixed structure, networks can be replaced with physical systems, which are called a physical reservoir[2], and this feature is paving the way for "processing by dynamics."

2.2 Evaluation method

Next, we define the benchmark used for a prediction accuracy. In this report, we use the normalized root mean square

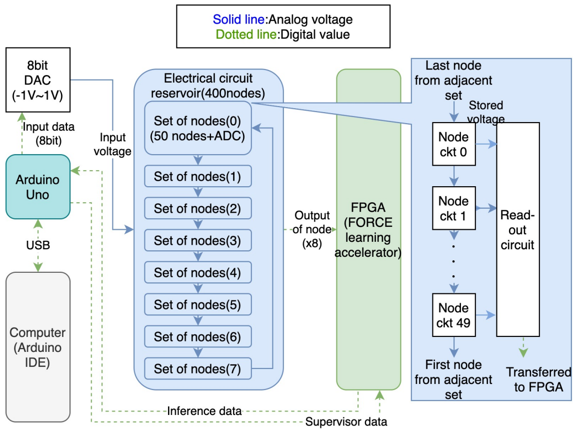


Figure 2: Construction of actual device

difference (NRMSD) of NARMA10-task (NARMA10) and memory capacity (MC) as benchmarks. MC is a measure of the memory capacity of past inputs, and NARMA10 is a time-series dataset based on the recent states of a network[11, 7]. For the definitions of NARMA10 and MC, please see the references herein. NRMSD is a measurement of the error between two series. The smaller the NRMSD of NARMA10 is, the higher the prediction accuracy.

$$\mathrm{NRMSD} = \frac{\sqrt{\sum_{t=1}^{T} (y_t - z_t)^2}}{\mathrm{Ave}(y)\sqrt{T}}. \tag{1}$$

The definition is provided in (1). For each function and variable, Ave(a) is the average of data a, T is the number of data, y is the supervisor data, and z is inference data.

3. Construction of Actual device

3.1 Design details of actual device

The actual device comprises an electrical circuit reservoir that functions as a physical reservoir, an FPGA that functions as the output weights as well as a learning system, and an Arduino Uno that serves as a controller, as shown in Figure 2. Arduino Uno controls the timing, generates 8-bit input data, and calculates 8-bit supervisor data. Input data are 8-bit unsigned integers, and are converted into the input voltage, which has a dynamic range of [-1,1]V, using DAC. Supervisor data are 8-bit signed integers, and have a dynamic range of [-128,127]. These conversions are slightly complex, and are summarized in Figure 6. The input and supervisor are updated at 250 Hz. The FPGA is used to implement the FORCE learning accelerator, which updates the output weights and computes the inference data[6]. FORCE learning is an online learning method, and is ideal for edge computing[12]. Inference data are transferred to Arduino to evaluate the prediction accuracy. For further details on the architecture of

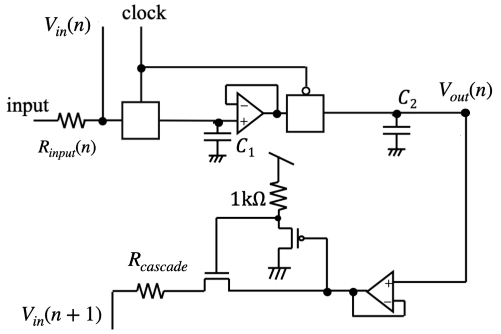


Figure 3: Schematics of node circuit

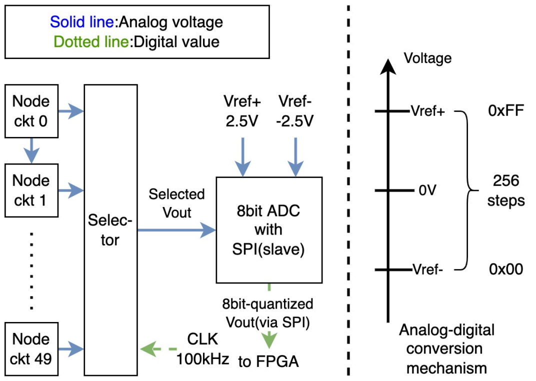


Figure 4: Readout and transferring mechanism

the accelerator, please see the references. Finally, the Electrical Circuit Reservoir is a physical reservoir composed of discrete semiconductor devices. It has a 400-node ring network structure, where each node is designed following the schematics in Figure 3, as proposed in previous studies [5]. Regarding each variable in Figure 3, $R_{input(n)}$ is the input weight resistance of the n-th node, $input$ is the input voltage, $v_{in}(n)$ and $v_{out}(n)$ are the stored voltages of the n-th node, and $R_{cascade}$ is the network weight resistance. Stored voltages, which are the network outputs, are converted into digital 8-bit values, and these values are transferred to FPGA for learning. This readout mechanism is shown in Figure 4, and runs at 100 kHz(node/second). To connect these devices following the diagram in Figure 2, the actual device was completed, as shown in the Figure 5.

3.2 Difference in SPICE model with actual device

Next, we describe the structure of the Simulation Program with Integrated Circuit Emphasis (SPICE) model. The SPICE model is almost the same as the model used in the previous study, with 400 node circuits connected in series[5]. The input voltage with a dynamic range of [-1,1]V, the node readout voltage, and the supervisor data are all represented as continuous values, not quantized values. The values are read out

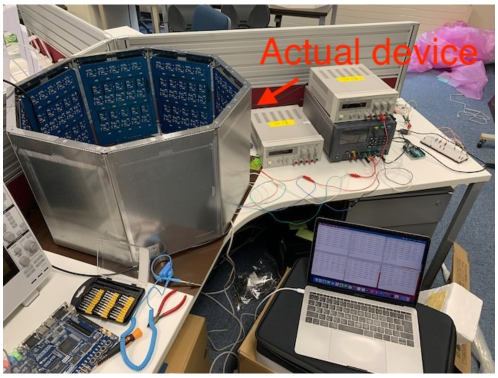


Figure 5: Photograph of actual device

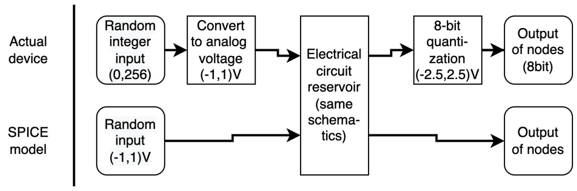
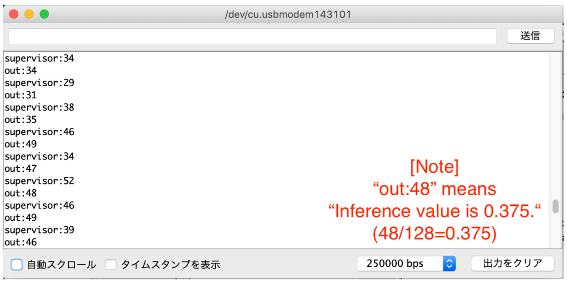

Actual device
SPICE model
Random integer input (0,256)
Convert to analog voltage (-1,1)V
Electrical circuit reservoir (same schematics)
8-bit quantization (-2.5,2.5)V
Output of nodes (8bit)
Random input (-1,1)V
Output of nodes

Figure 6: Difference in actual device with SPICE model

and used for software-based learning. These differences are summarized in Figure 6.

3.3 Evaluation method and parameters

We next describe the performance measurement method. For the SPICE model, we ran a circuit-simulation on NGSPICE, as in the measurement method of the previous study, and obtained voltage values stored in the nodes[5]. These values and input data are used for BATCH learning using Python program. For the actual device, supervisor and inference data are received through the Serial Monitor of Arduino IDE, as shown in Figure 7. We compared the supervisor data with inference data to evaluate the prediction accuracy. In both cases, we use 5000 random input data, i.e., from 0 to 1000 for initialization, from 1000 to 4000 for training, and from 4000 to 5000 for evaluation.



```
supervisor:34
out:34
supervisor:29
out:31
supervisor:38
out:35
supervisor:46
out:49
supervisor:34
out:47
supervisor:52
out:48
supervisor:46
out:49
supervisor:39
out:46
```

[Note]
“out:48” means
“Inference value is 0.375.“
(48/128=0.375)

Figure 7: Serial Monitor from Arduino IDE

Table 1: Evaluation of each model

Task	SPICE model	Actual device
NARMA10	0.086	0.111
MC	72.73	8.02

Table 2: Hypothesis of degrading MC

Case	Factor	Hypothesis
1	Thermal noise	Thermal noise prevents voltage storage in a node.
2	Quantization noise	Voltage stored in a node is converted into an 8-bit value by an ADC. The quantization noise degraded the accuracy.

4. Evaluation and consideration

4.1 Evaluation

Results of NARMA10-task and MC-task on the SPICE model and actual device are shown in Table 1. In NARMA10-task, the prediction accuracy of the SPICE model is better than that of the actual device, although there was no significant difference. By contrast, for the MC-task, we can see that there was a large difference between the performance of the actual device and that of the SPICE model. This suggests that some factors may have reduced the MC of the actual device.

4.2 Consideration of performance difference

Based on these results, we considered the factors that may degrade the MC. The difference in performance may attributed to the difference between the SPICE model and the actual device, and thus we summarized the possible factors of the accuracy degradation in Table 2. We conducted an investigation into each of these factors.

4.3 Effect of thermal noise

First, we investigated the effect of thermal noise. To determine the effect of thermal noise, we imposed a noise current on the SPICE model and evaluated its performance. As shown in Figure 8, we added a noise current source to the node circuit of the SPICE model. We adjusted the RMS intensity of the noise from 1 nA to 1μA, and investigated the effect on the MC. As a supplement, because this electrical circuit simulation requires significant computational resources and a several days of running time, we shortened the learning condition. Under this investigation, we used 500 random input data from 0 to 100 for initialization, from 100 to 400 for training, and from 400 to 500 for evaluation. The results are shown in Table 3, which indicates that the MC degrades when the intensity of thermal noise exceeds 100 nA. However, the

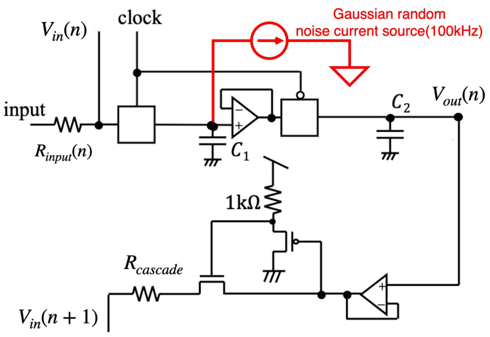


Figure 8: Schematics of node circuit with noise current source

Table 3: Effects of thermal noise on MC

RMS intensity(nA)	1	10	100	1000
MC	62.24	61.28	52.48	38.21

intensity of 100 nA is too high, considering that realistic noise is generally no more than a few nA. Therefore, through this investigation, it was shown that the actual device has a slight tolerance to thermal noise.

4.4 Effect of 8-bit quantization

Finally, we investigated the effect of quantization noise. In the SPICE model, input voltage and node voltage are represented as continuous values. By contrast, in the actual device, the input voltage is generated from 8-bit values and the node voltage is 8-bit-quantized using ADC. Therefore, to determine the effect of quantization noise, we applied pseudo-quantization, which is expressed through Functions (2) and (3), to the SPICE model and checked the effect on the MC.

$$\mathrm{DAC}(x) = \frac{\lfloor x \times 128 \rfloor}{128}, \tag{2}$$

$$\mathrm{ADC}(x) = \frac{\lfloor x \times 51.2 \rfloor}{51.2}. \tag{3}$$

The flow of the operation is shown in Figure 9. The prediction accuracy of the SPICE model with pseudo-quantization applied is shown in Table 4. The NARMA10-task showed a slight decrease, whereas the MC-task showed a decrease in accuracy of 50%. This result suggests that the quantization noise may have a significant impact on prediction accuracy of the device.

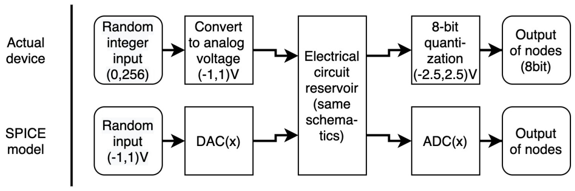

Actual device
SPICE model
Random integer input (0,256)
Convert to analog voltage (-1,1)V
Electrical circuit reservoir (same schematics)
8-bit quantization (-2.5,2.5)V
Output of nodes (8bit)
Random input (-1,1)V
DAC(x)
ADC(x)
Output of nodes

Figure 9: Flow of pseudo-quantization

Table 4: Effect of pseudo-quantization on SPICE model

Task	SPICE model	8-bit SPICE model
NARMA10	0.086	0.119
MC	72.73	35.54

5. Conclusions

In this study, we designed a physical reservoir computing device composed of electronic circuits and an FPGA, reported the results of an evaluation of the spice model, and discussed their performances. The actual device showed a similar performance in NARMA10-task, but a significant degradation in the prediction accuracy in MC-task. The results of an additional investigation suggested that the cause of the degradation was quantization noise. In the future, we would like to continue to verify the performance degradation caused by quantization noise as well as other factors. We would also like to redesign the device based on these investigations.

Acknowledgment

The authors would like to sincerely thank TDK Co. for their cooperation in this research.

References

[1] H.Momose, et al., Jpn. J. Appl. Phys., 59 050502, 2020.

[2] M.Inubushi et al., Reservoir Computing, pp 97-116, 06 August, 2021

[3] H.Nomura, et al., Jpn. J. Appl. Phys., 58 070901, 2019.

[4] K.Nakajima, Brain Evolution by Design, pp 403-414, 08 February, 2017

[5] S.Suzuki, et al., NCSP'20, Feb. 28-Mar. 2, 2020.

[6] K.Minamikawa, et al., NCSP'20, Feb. 28-Mar. 2, 2020.

[7] H.Jaeger, Advances in neural information processing systems, Vol.15, pp.609–616, 2003.

[8] H.Jaeger, Bonn, Germany: German National Research Center for Information Technology GMD Technical Report, vol.148, no.34, p.13, 2001.

[9] W. Maass, Computability in context: computation and logic in the real world, pp.275 – 296, 2011.

[10] G. Tanaka, et al., Neural Networks, Vol.115, pp.100-123,2019.

[11] H.Jaeger, GMD Report, Vol.152, 2001.

[12] D.Sussillo, L.F.Abbott, Neuron,, Vol.63, no.4, pp 544-557, August 27,2009.