Investigating Small Device Implementation of FRET-based Optical Reservoir Computing

Masafumi Tanaka[1] Jaehoon Yu[2] Masaki Nakagawa[3] Naoya Tate[4] Masanori Hashimoto[5]
[1]*Osaka University* [2]*Tokyo Institute of Technology* [3]*Fukuoka Institute of Technology*
[4]*Kyushu University* [5]*Kyoto University* hashimoto@i.kyoto-u.ac.jp

2022 IEEE 65th International Midwest Symposium on Circuits and Systems (MWSCAS) | 978-1-6654-0279-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/MWSCAS54063.2022.9859432

***Abstract*—This work aims to develop a reservoir computing (RC) device based on the phenomenon of FRET between quantum dots (QDs), where FRET is a phenomenon in which excited states are transferred between QDs, and its properties change depending on the distance between QDs. We propose a compact structure in which optical input/output and quantum dots are adjacently placed without lenses or delay lines. The proposed structure exploits the QD-based optical reservoir as an intermediate layer and adopts memory to enable recurrent inputs. We evaluate the feasibility of the proposed structure by applying tasks that require nonlinearity. Simulation-based experimental results show that the proposed device can perform logistic map, time-series XOR, and NARMA10.**

***Index Terms*—Reservoir computing, optical reservoir, quantum dots, FRET, fluorescence**

I. Introduction

Reservoir computing is expected to facilitate the model training since only the output connection is trained while the randomly-generated middle layer is unchanged. Physical reservoir, which replaces the middle recurrent layer with physical phenomena, is drawing attention [1] since it may process the data more efficiently than conventional digital circuits. To enable reservoir computing, the physical reservoir should have the properties of non-linearity, short-term memory, and high dimensionality. Various physical reservoirs have been studied in a variety of things. Examples includes light-based reservoir [2], oscillator-based reservoir [3], and mechanical reservoir using physical properties [4].

In the IoT era, edge computing is demanded in various applications instead of traditional centralized data processing. Edge computing requires processing large amounts of data, which imposes efficient computation since computation resources and power are often limited. Deep learning requires a lot of power to perform the computation, making us find alternatives. Also, the need for computing closer to the user requires devices that are smaller in size.

Fluorescence resonance energy transfer (FRET) is a phenomenon in which excited states are transferred, for example, between neighboring quantum dots (QDs). The state transfer probability depends on the QD types and the distance between the QDs. Therefore, a randomly-generated QD network, which includes QDs densely, inherently exposes various nonlinear relationships between input excitation light and output fluorescence. Also, the excitation state represents the memory, and therefore, FRET in the QD network is a promising behavior for a physical reservoir. On the other hand, when capturing the fluorescence with a photo-diode (PD) array or an image sensor, the nano-meter scale of QD and micro-meter scale of PD make it challenging to know which QD emits each photon. Also, the state holding duration and the fluorescence lifetime are tens of nano-seconds, which may necessitate repeated evaluations. For overcoming such issues, optical devices often use lenses and delay lines. However, they prevent the computing device from being small in size.

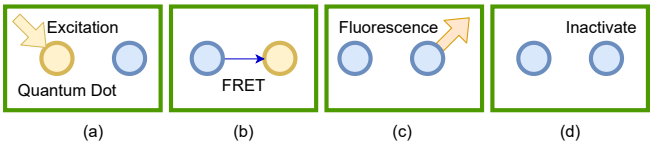

Fig. 1. QD behaviours.

This study proposes a simple structure that can be easily miniaturized without using lenses or delay lines while utilizing FRET for computing. The proposed structure includes excitation light sources, such as tiny LEDs, a sheet including a QD network, a filter that passes fluorescence but eliminates excitation light, and a photo-diode array. We map several tasks requiring nonlinearity on the device and confirm the feasibility of the proposed device using a FRET simulator.

II. Quantum Dots and FRET

Fig. 1 illustrates the QD phenomena focused in this work. When excitation light activates a quantum dot (a), the excitation energy may move to neighboring QDs (FRET) (b), emit a photon as fluorescence (c), or become inactivated without fluorescence (d). The probability of excitation energy transfer is modeled in [5] as $\frac{1}{1+(R/R_0)^6}$, where R is the distance between QDs, and R_0 is the Förster distance, which depends on the QD type. When $R = R_0$, the energy transfer occurs at the probability of 50 %.

The wavelength of the fluorescence uniquely depends on the QD type. The excitation light should have a shorter wavelength than that of the fluorescence to enable the excitation. On the other hand, the wavelength difference between excitation and fluorescence is used to filter out the excitation light and solely observe fluorescence. Besides, the FRET network can be configured with multiple types of QDs, but this work supposes that a single type of QD is used for the QD network as the first step of FRET-based reservoir computing.

Fig. 2 shows the input and output light intensities for a two-QD system, which is obtained by the FRET simulator [6] with

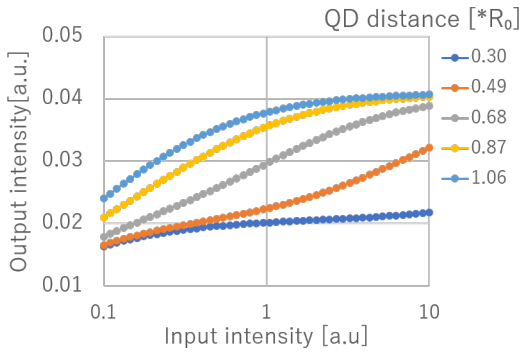

Fig. 2. Relationship between input and output light intensities.

a similar setup with Section IV. The distance values between the QDs are expressed as a ratio to the Förster distance. You can see a nonlinear relationship between input and output intensities depending on the distance. Note that, without FRET, such distance-dependency would not arise. Even when many QDs are randomly placed, the density difference makes the input-output relationship diverse.

III. Proposed Device Structure

A. Device structure

We propose the device structure whose cross-section is shown in Fig. 3 for FRET-based reservoir computing. The red box in the figure represents the physical reservoir, i.e., the intermediate layer in reservoir computing. The light source 2D-array (input), a sheet including many quantum dots randomly, and PD 2D-array (output) are placed in a straight line. The light source provides the optical input to excite the QDs, and PD acquires the intensity of the fluoresce. For pursuing form-factor minimization, lenses are not adopted in this structure. The light source, QDs, and PDs would be stacked and housed in a single package in the expected final implementation.

As mentioned in Section I, the memory of the excited state in quantum dots is very short, and QD-wise fluorescence detection is infeasible due to the mismatch in size between QD and PD. Therefore, the proposed structure introduces a digital memory to compose a recurrent network. Also, single-photon detection is difficult since the sensitivity of ordinary PD is not high enough, and moreover, the photon emission is isotropic in all solid angles. Hence, repeated input and accumulation are necessary to obtain the digitized reservoir output stably. Taking into account such accumulation time, the reservoir in the proposed structure operates at the discretized time like a sequential circuit. The recurrent behavior proceeds with this discretized time using the feedback through the memory. Finally, the digitized reservoir output is given to a lightweight machine learning model, such as linear support vector machine and ridge regression, to obtain the final output.

Fig. 4 shows a closer view of the positional relationship between the light sources, QDs, and PDs. QDs are much smaller than PDs, as mentioned earlier, and the photon emission is isotopic. Therefore, a PD receives the photons coming from multiple, rather many, QDs in this configuration. However, even in this case, we can obtain the nonlinear relationship between the input and output, as shown in Fig. 2 is obtained and used for reservoir computing. There is a filter between the QDs and PDs to filter out the excitation light. We can find thin yet high rejection-ratio bandpass filters, e.g., in [7].

B. Network mapping

In the proposed structure, we can give a switching matrix function to the feedback memory. Namely, we can choose some of the PD outputs for feedback and determine the locations of the feedback inputs. In addition to the conventional training of the output part, we can, in other words, need to determine the feedback matrix.

This section presents a method that maps an echo state network (ESN) into the proposed device. Here, ESN is a basic type of non-physical reservoir implementation [8]. The middle layer corresponding to the reservoir consists of artificial neurons whose recurrent connections and weights are randomly determined. A nonlinear transformation is performed by using a nonlinear activation function.

This work proposes to selectively map a compact ESN achieving high performance into the proposed device structure by adjusting the memory switching matrix. Fig. 5 shows the correspondence between the ESN and the network on the proposed device. Several PDs and light sources compose a single unit (node). The following explanation supposes that a set of 3×3 light source array and 3×3 PD array is considered equivalent to a node in an ESN. In Fig. 5, the sizes of the light source and PD are equal for simplifying the explanation.

The spatial and temporal overlap of the light from the light sources can provide interaction between multiple light sources. The arrows (called edges) between the nodes from PD to the light sources mean that the PD output is delayed by one step, and the light sources give recurrent inputs. By choosing the relative position of the light source and PD, the so-called weights can be set since the spatial overlap varies. In the device, the nodes are separated in time using external memory. As for space, the nodes are separated by a sufficient distance from each other. Therefore, FRETs do not occur between nodes, nor do they affect each other unexpectedly.

IV. Experiments

A. Setup

We randomly generate QD networks for experiments according to the conditions listed in Table I. In this study, QD585 [9] is assumed. Other device parameters are also listed in Table I. We use two types of light sources; DC and pulsed sources. When using DC light source, its intensity represents the input value. In the case of the pulsed source, the pulse count in a unit time represents the input value. More concretely, the input light is pulsed with a period of 10/(input value) [ns], where the pulse width is constantly 1 ns.

In this work, we simulate the QD behavior using a simulator reported in [6]. We use a Monte Carlo option of the simulator to stochastically simulate the state transitions between QD

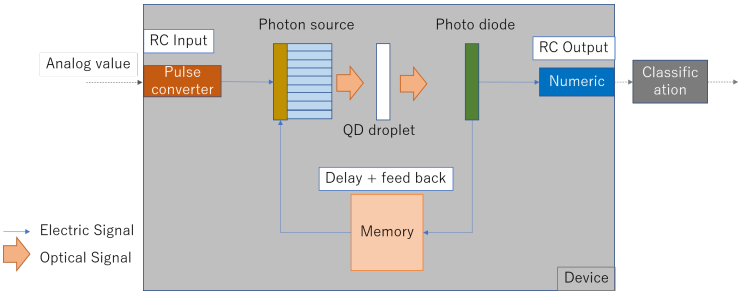

Fig. 3. Proposed device structure.

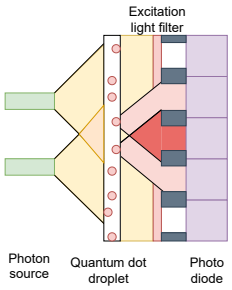

Fig. 4. Closer view near QDs.

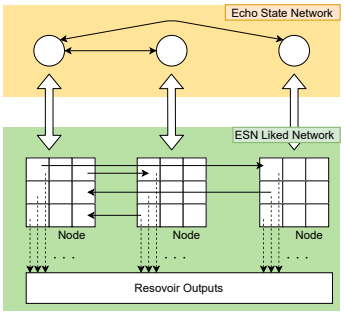

Fig. 5. Network construction.

TABLE I
SIMULATION SETUP.

QD density	1000 dots/μm^2
# of MC trials	200
PD size	600 nm
PD pitch	1000 nm
QD-PD distance	1000 nm
LS-QD distance	1000 nm
Sim. step	100 ns

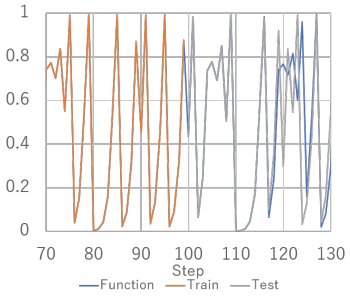

Fig. 6. Prediction result of logistic map in time domain.

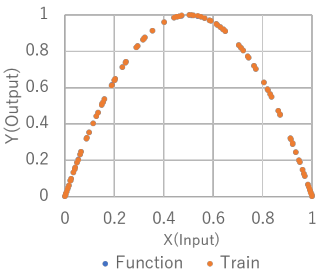

Fig. 7. Prediction result of logistic map in input-output domain.

excitation, FRET, fluorescence, and inactivation on the basis of tRSSA [10]. On the other hand, the original FRET simulator simulates the state of the QDs only. Therefore, we constructed a simulation framework that reproduces the entire proposed device structure and simulates it as a device using the above-mentioned FRET simulator as a core engine. The input light and the fluorescence, for example, decay according to the squared distance in the framework.

B. Memory-unnecessary tasks

We first test two tasks that approximate nonlinear functions that do not require memory.

1) Logistic map: A logistic map is a chaotic system where even a tiny error can significantly disturb the calculation. The logistic map is represented by

$$x_{t+1} = \alpha x_t(1 - x_t),$$

where $\alpha = 4.0$ and the initial value $x_0 = 0.2$.

In this logistic mapping, the output value becomes the next input value. Therefore, the next input shall be based on the predictions one after another in the prediction process. Ridge regression is used to learn the function. The mean squared error (MSE) in training was 9.61×10^{-9}. Here, 5×5 PD array and DC light source are assumed.

Fig. 6 shows the original function Y, the trained data $Train$, and the prediction $Predict$. The training is performed in the first 100 steps, and the prediction is performed after 100 steps are passed. Because of the chaotic nature of the system, the predictions eventually become different. However, the first 17 steps, i.e., 100 to 117 steps, are well approximated, which means the approximation is feasible. Fig. 7 shows an X-Y diagram plotted with the input on the horizontal axis and the output on the vertical axis. We can see that the function and the prediction are almost identical.

2) XOR: Two-input XOR $y = \mathrm{XOR}(x_1, x_2)$ is used to check whether the nonlinearity can be derived in the proposed structure. We use linear support vector machine (SVM) for output classification. Here, it should be noted that linear SVM cannot approximate XOR solely. Two hundred cases are used for training, and additional 50 cases are used for evaluation. The inputs x_1 and x_2 are given from two locations with pulsed light sources. In this experiment, the pulse frequencies for 0 and 1 are 50 MHz and 100 MHz, respectively.

We executed this experiment in two configurations; 2×2 and 3×3 PD arrays. The input lights are given to the most distant diagonal locations. In the case of the 3×3 PD array, both the training and evaluation attained 100% accuracy. However, in the 2×2 case, the training and evaluation accuracies are both 75%, which means the XOR function is not approximated well. In the 2×2 case, there are only three variations of the distance between the light source and the PD, whereas there were six variations in the 3×3 case. We think this difference in the distance variation count contributed to the accuracy.

C. Memory-necessary temporal tasks

We next test tasks requiring memory in the reservoir.

1) Time-series XOR: Time-series XOR takes random 0 and 1 inputs and predicts the result of XORing the previous and current inputs.

$$d(n) = \mathrm{XOR}(u(n), u(n-1)),$$

where $u(n)$ is the randomly generated input at time n. The same pulsed light source as Section IV-B2 is used for inputting $u(n)$. As for the feedback input, the pulse period is inversely

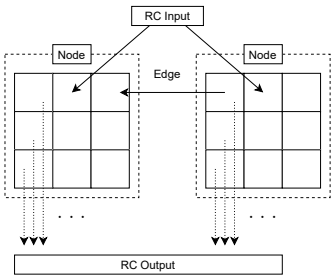

Fig. 8. Network structure for time-series XOR.

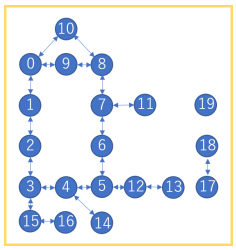

Fig. 9. NARMA10 network structure

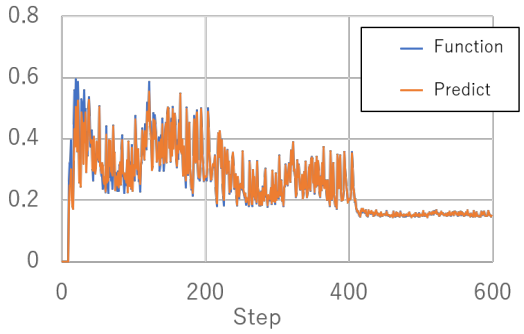

Fig. 10. NARMA10 prediction result.

proportional to the amount of photons received by the associated PD. The absolute value of the period is adjusted within the period range of the input light source for $u(n)$.

Fig. 8 exemplifies one network configuration we tested. There is one edge from the right node to the left node. In this configuration, the right node works as one-step memory and provides one-step previous input to the left node. The left node can process the current input and the previous input coming from the right node. Each node consists of 3×3 PDs per node, and there are 18 outputs in total as a reservoir. Linear support vector machine is applied to the reservoir outputs. We experimentally confirmed that both training and evaluation attained 100% accuracy.

D. NARMA10

Finally, we test NARMA [11], which is often used for evaluating reservoir computing capability. NARMA is expressed by the following equation.

$$d(n+1) = a_1 d(n) + a_2 d(n) \sum_{i=0}^{m-1} d(n-i) + a_3 u(n-m+1) u(n) + a_4,$$

where a_i is constants, and $u(n)$ is the input at time n. The output d depends on the inputs of previous m steps, which means m-step memory is necessary. In the following, we evaluate a popular setting of $m = 10$, which is called NARMA10.

Due to the large number of steps in NARMA10 and the consequent long simulation time, the number of FRET simulation trials was set to 100. This experiment supposes online training, meaning that the weights are updated sequentially every step by the sequential least-squares method. The number of training steps was 200 steps for three different sets of NARMA10 parameters, resulting in 600 steps in total.

First, appropriate network structures were explored. We generated many ESNs changing the number of nodes and edges. We then trained and evaluated them using a metric of RMSE. Among them, we chose a 20-node network having cyclic structures shown in Fig. 9 since it achieves good accuracy. Then, we mapped this network on the proposed device structure following the procedure discussed in Section III. Each node is assumed to have 3×3 PD array and 3×3 light source array.

Fig 10 shows the NARMA10 function $d(n)$ and the predicted results. The blue line represents $d(n)$, which is the target for prediction. The orange line shows the predicted value. The blue and orange lines are slightly distant at the beginning of the training, but they overlap as the steps progress. The RMSE is 0.020.

V. Conclusion

This work proposed a compact device structure for FRET-based optical reservoir computing. The proposed structure, which includes tiny light sources, a randomly-generated dense QD network in a thin sheet, optical filter and a PD array, is suitable for a single package implementation and resultantly edge computing. Simulation-based experiments mapping logistics map, XOR, and NARMA10 tasks showed that the proposed structure could execute those tasks and provide nonlinearlity, memory and high dimensionality demanded for reservior computing.

Acknowledgement

This work was supported by JST CREST Grant Number JPMJCR18K2, Japan.

References

[1] G. Tanaka, et al.,"Recent advances in physical reservoir computing: A review", *Neural Networks*, Vol. 115, pp. 100-123, 2019.
[2] S. S. Vatsavai and I. Thakkar, "Silicon Photonic Microring Based Chip-Scale Accelerator for Delayed Feedback Reservoir Computing," arXiv:2101.00557, 2021.
[3] J. Torrejon, "Neuromorphic computing with nanoscale spintronic oscillators," *Nature*, Vol. 547, pp. 428-431, 2017.
[4] K. Nakajima, et al.,"Information processing via physical soft body," *Scientific Reports*, Vol. 5, 10487, 2015.
[5] M. Hoefling, H. Grubmülle,"In silico FRET from simulated dye dynamics," *Computer Physics Communications*, Vol. 184, No. 3, pp. 841-852, 2013.
[6] M. Nakagawa, et al., "Spatiotemporal model for FRET networks with multiple donors and acceptors: Multicomponent exponential decay derived from the master equation," *Journal of the Optical Society of America*, Vol. 38, No. 2, pp. 294-299, 2021.
[7] K. Sasagawa, Y. Ohta, M. Kawahara, M. Haruta, T. Tokuda, and J. Ohta, "Wide field-of-view lensless fluorescence imaging device with hybrid bandpass emission filter," *AIP Advances*, Vol. 9, No. 3, 035108, 2019.
[8] H. Jaeger, "The "echo state" approach to analysing and training recurrent neural networks," *German National Research Center for Information Technology GMD Technical Report*, Vol. 148, 2001.
[9] S. H. Ko, K. Du, J. A. Liddle,"Spectra Database hosted at the University of Arizona," *Angew. Chem. Int. Ed.*, 52.4, 1193-1197, 2013.
[10] V. H . Thanh, C. Priami, "Simulation of Biochemical Reactions with Time-Dependent Rates by the Rejection-based Algorithm," *The Journal of Chemical Physics*, 143, 054104, 2015.
[11] H. Jaeger, "Adaptive Nonlinear System Identification with Echo State Networks," in *Proc. NIPS*, pp. 609–616, 2002.