Reservoir computing with dipole coupled nanomagnets array

Hikaru Nomura, Kazuki Tsujimoto, Yuki Kuwabiraki, and Ryoichi Nakatani

Graduate School of Engineering, Osaka University, Suita, Osaka 565-0871, Japan

Ferdinand Peper

Center for Information and Neural Networks,
National Institute of Information and Communications Technology,
Osaka University, Suita, Osaka 565-0871, Japan

Eiiti Tamura and Taishi Furuta

Graduate School of Engineering Science,
Osaka University, Toyonaka, Osaka 560-8531, Japan

Shinji Miwa

Graduate School of Engineering Science,
Osaka University, Toyonaka, Osaka 560-8531, Japan

Center for Spintronics Research Network,
Osaka University, Toyonaka, Osaka 560-8531, Japan

The Institute for Solid State Physics, The University of Tokyo, Kashiwa Chiba 277-8581, Japan

Minori Goto and Yoshishige Suzuki

Graduate School of Engineering Science,
Osaka University, Toyonaka, Osaka 560-8531, Japan

Center for Spintronics Research Network,
Osaka University, Toyonaka, Osaka 560-8531, Japan
Abstract

We demonstrate the feasibility of reservoir computing based on dipole coupled nanomagnets array through macrospin simulations. The reservoir is composed of $2 \times 10$ nanomagnets. We use the static magnetization directions of the nanomagnets as reservoir states. To update the reservoir states, we change the uniaxial anisotropy along the symmetrical axis of the disk-shaped nanomagnets using a voltage induced magnetic anisotropy change. Binary tasks with AND, OR and XOR functions were performed to evaluate the performance of the nanomagnets array reservoir. As a result, the output matrix of the reservoir computing can be trained to perform AND, OR and XOR functions with an input delay of up to three steps.
Recurrent neural networks (RNN) are recognized as a promising scheme to realize high performance artificial intelligence in small sized package with small energy consumption. An RNN consists of multiple nodes with closed feedback loops, and information represented as states of its nodes, which are updated through interactions between the nodes. Many physical phenomena can be used for the realization of RNNs.

In 2000, Cowburn and Welland introduced a magnetic quantum cellular automaton (MQCA) composed of nanomagnets that interact with each other via magneto-static interactions. In MQCA, information is stored as the directions of magnetizations of nanomagnets, whereas logic operations are executed via the magneto-static interactions, resulting in a transmission wire, majority logic gate, shift register and other functionalities. The MQCA has information non-volatility and binary computing capability. Moreover, based on the magneto-static interactions between the nanomagnets, the MQCA has the potential to calculate non-linear functions. Though the MQCA is a good candidate to realize RNN devices, it is very difficult to implement learning mechanism on the MQCA, since the intensity of an inter-magnet interaction (synapse weight) is determined by the geometrical alignment of magnetic cells.

Reservoir computing has appeared as a technique that may solve such learning problems when realized in a given physical system. Reservoir computing uses a reservoir of nodes that interact with each other, like in RNN’s, but it differs from the latter in that it contains a layer of output nodes of which the connections to the nodes in the reservoir can be trained. Learning in reservoir computing is thus limited to only an output matrix. Because of this simple structure, most physical systems can be utilized as a reservoir, and it is this point combined with its surprisingly high performance that has garnered attention over the last decade. Recently, a magnetic nano oscillator was introduced as a node in the reservoir computing. This method is suitable for high speed operation, with switching speeds typically in the nano-second range. However, when we consider a slow task such as a voice recognition, operation with static magnetization of nanomagnets with an external clock could be preferred. When we use the static magnetization direction as a reservoir state, we can control an operation’s timing by changing the external clock frequency.

Here, we demonstrate reservoir computing with a nanomagnets array using macrospin simulations. We use static magnetization directions to represent the reservoir states. To check the performances of the nanomagnets array reservoir, we performed computational tasks on binary numbers.
To simulate the magnetization behavior of the nanomagnets, we perform macro-spin simulations. We assume that the nanomagnets take a single domain state, and have uniaxial anisotropies along the z axis. For the magneto-static interactions between the nanomagnets, we consider dipole interactions. Therefore, the Landau-Lifshitz-Gilbert (LLG) equation becomes,

$$\frac{dM_i}{dt} = -\gamma_{\text{LL}} M_i \times H_{\text{eff},i} - \frac{\alpha \gamma_{\text{LL}}}{M_s} M_i \times (M_i \times H_{\text{eff},i}),$$  \hspace{1cm} (1)

$$H_{\text{eff},i} = H_{k,i} + H_{\text{dipole},i}.$$  \hspace{1cm} (2)

$$H_{k,i} = \frac{2}{\mu_0 M_s^2} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & K_{u,i}(t) \end{pmatrix} M_i,$$  \hspace{1cm} (3)

$$H_{\text{dipole},i} = \sum_{j=1}^{N_{\text{mag}}} w_{\text{dipole},ji} M_j$$  \hspace{1cm} (4)

$$w_{\text{dipole},ji} = \frac{V_j}{4\pi r_{ji}^3} \begin{pmatrix} 3r_{s,ji}^2 - r_{i,ji}^2 & 3r_{s,ji} r_{y,ji} & 3r_{s,ji} r_{z,ji} \\ 3r_{s,ji} r_{y,ji} & 3r_{y,ji}^2 - r_{j,ji}^2 & 3r_{y,ji} r_{z,ji} \\ 3r_{s,ji} r_{z,ji} & 3r_{y,ji} r_{z,ji} & 3r_{z,ji}^2 - r_{j,ji}^2 \end{pmatrix},$$  \hspace{1cm} (5)

where $N_{\text{mag}}$ is the number of the nanomagnets, $i$ and $j$ are the indices of the nanomagnets ($i, j \in \{1, 2, \ldots, N_{\text{mag}}\}$), $M_i$ is the magnetization, $H_{\text{eff},i}$ is the effective field, $\alpha$ is the damping constant, $\gamma_{\text{LL}} = \gamma/(1 + \alpha^2)$ is the gyromagnetic ratio in the Landau-Lifshitz form, $\gamma$ is the gyromagnetic ratio, $H_{k,i}$ is the uniaxial anisotropy field, $H_{\text{dipole},i}$ is the dipole field, $K_{u,i}(t)$ is the anisotropy constant ($0 \leq K_{u,i}(t) \leq K_{u,\text{max}}$), $K_{u,\text{max}}$ is the maximum value of the anisotropy constant, $\mu_0$ is the permeability of vacuum, $M_s$ is the saturation magnetization, $V_j$ is the volume of the nanomagnets and $r_{ji} = \{r_{x,ji}, r_{y,ji}, r_{z,ji}\}$ is the directional vector from nanomagnet $j$ to nanomagnet $i$. The parameters used here are as follows: $N_{\text{mag}} = 20$, $M_s = 1.3 \times 10^6$ A/m, $\gamma = 2.211 \times 10^5$ m/As, $K_{u,\text{max}} = 0.1 \times 10^3 M_s^2 J/m^3$ and $\alpha = 0.5$. We solve the LLG equation using a Runge-Kutta fourth-order method.

Figure 1 shows a schematic illustration of the nanomagnets array reservoir. The nanomagnets array is composed of $2 \times 10$ nanomagnets. The radius and thickness of the nanomagnets are all 20 nm and 1 nm, respectively. The gaps between the nanomagnets are 20 nm. The magnetization direction of the nanomagnet with index of 1 in Fig. 1 is used for input.

For a reservoir state $x$, we use the $x$-component of the normalized magnetizations $\mu_{x,i}$ and bias.
FIG. 1. Schematic top view of recurrent neural network based on a nanomagnets array. The numbers shown at the lower right of the dots are the indices of the dots.

FIG. 2. Update scheme of the reservoir state. The uniaxial anisotropy constants of the nanomagnets are modified at each stage.

With a fixed value of 1. The reservoir state at step \( k \) then becomes

\[
\mu_{x,j}(k) = \frac{1}{2} \frac{M_{x,j}(k)}{M_s} + \frac{1}{2}
\]

\[
x_k = \{\mu_{x,1}(k), \mu_{x,2}(k), \ldots, \mu_{x,N_{\text{mag}}}(k), 1\}.
\]

In an actual device, noise exists in the magnetization measurements, so, the number of significant figures of the magnetization measurements is limited. Therefore, we rounded the \( x \)-component of the magnetizations to no more than three significant figures.

To update the reservoir states, we change the uniaxial anisotropy of the nanomagnets. Fig. 2 shows uniaxial anisotropy constants \( K_u \)'s of the nanomagnets at each stage. At the beginning of
each stage, the $K_u$’s of the nanomagnets are changed. After changing the $K_u$’s, magnetization direction of nanomagnets with $K_u = 0$ relaxes to the local magnetic field direction that is a weighted sum of the field produced by the other nanomagnets (See eq. (4)). This system is a kind of infinite-range model. The magnetic field is determined self consistently. Therefore, the network is recurrent and it conducts a nonlinear operation on the sum of the field. We classify nanomagnets into Group I to III according to the way of $K_u$’s change (See Fig. 1). In the first stage ($p = 1$), the values $K_u$ of the nanomagnets in Group II and III are changed to 0. Under this configuration, the magnetizations of the nanomagnets in Group II are easily rotated by the stray fields from the other nanomagnets, and the information stored in Group I is mainly transferred to Group II by partly corporating the information from the other nanomagnets. In the second stage ($p = 2$), we fix the magnetizations of the nanomagnets in Group II by changing the $K_u$’s of the nanomagnets in Group II to $K_{u,\text{max}}$. This also yields a nonlinear operation on the sum of the field. By repeating this process, the information stored in the nanomagnets is partly transferred in the y-axis direction in corporating information from the other nanomagnets in a nonlinear way. In actual devices, a bias voltage could be used to control the uniaxial anisotropy constant of the individual nanomagnets ([17, 18]) and the frequency of the bias voltage can be used as an external clock. We used the magnetization direction at the end of the third stage ($p = 3$) as a reservoir state $x_k$.

The output $o_k$ of the nanomagnets array reservoir computing then becomes,

$$ o_k = x_k \cdot w_f $$

where $w_f$ is an output vector of the reservoir computing and $f$ is a teacher function for the output vector. An output vector $w_f$ is trained to minimize the square error between the teacher function and output of the reservoir computing $\left( \sum^N_{\text{test}} (f - o_k)^2 \right)$.

Here we define an error rate between the teacher function $f$ and the trained reservoir computing output $o_k$ as

$$ \frac{1}{N_{\text{test}}} \sum^N_{\text{test}} |f - o_k|, $$

where $N_{\text{test}}$ is the number of binary input data, $m$ and $n$ are the input delays of the teacher functions for the test and the training, respectively.

For input $u_k$, we use uniformly distributed random binary data (Fig. 3(a)). The input values are written as a magnetization direction of the input nanomagnet $M_{i=1}$ at the beginning of the first stage ($p = 1$). When the input value $u_k$ at step $k$ is 1, we set the magnetization of the input
FIG. 3. Typical values of (a-b) input, (c-e) teacher functions, and (f-h) trained reservoir computing outputs.

FIG. 4. Output of the AND, OR and XOR functions. The output of each functions can be divided into different classes with red lines. The white circles and red circles denote that the output values of the functions are 0 and 1, respectively.

As teacher functions $f$, we use AND, OR, and XOR functions. Fig. 4 shows output values of these functions. The white circles and red circles denote that the output values of the functions are 0 and 1, respectively. Here we consider the way to classify the output values from the input values $A$ and $B$. The output of the AND/OR functions are linearly separable (e.g. $A + B = 1.25$ and $A + B = 0.25$). On the other hand, the XOR outputs can only be separated by a non-linear function (e.g. $4A^2 - 5A + 2AB - 4B + 3B^2 = -1$). Non-linear classification tasks like the XOR function task are considered more difficult than linear classification. For the supervised learning of the output matrix we use the input values at steps $k$ and $k-n$ as the teacher functions, whereby $n$ is a delay of the input values (See Fig. 3(b) for $n = 3$). The teacher functions become, AND ($u_k$, $u_{k-n}$), OR ($u_k$, $u_{k-n}$) and XOR ($u_k$, $u_{k-n}$) (See. Fig. 3(c-d)). Note that only the input value of $u_k$ is written...
FIG. 5. Error rates between the teacher functions $f(u_k, u_{k-m})$ and the nanomagnets array reservoir output $o_k = x \cdot w_{f(u_k, u_{k-n})}$, where $m$ and $n$ are the input delay in the teacher function for the test and training, respectively.

The AND/OR function is used to measure the performance of a linear operation, and the XOR function is used to measure the performance of a non-linear operation with delayed input.

In this study, 100 binary input data samples ($N_{\text{train}} = 100$) were used for training. Fig. 3(f-h) show typical output of the nanomagnets array reservoir. By comparing the data in Fig. 3(c-e) and Fig. 3(f-h), the output of the reservoir computing with trained output vectors shows good agreement with the teacher functions.

Fig. 5 shows the error rates with various target functions and teacher functions. The error rates were measured with four sets of 1000 binary input data ($N_{\text{test}} = 1000$). Up to an input delay of three steps ($n \leq 3$), when the delay in the teacher function for test $m$ and training $n$ are the same ($m = n$), the error rate is almost 0. On the other hand, the error rates of the other data for AND/OR and XOR are almost 0.25 and 0.5 respectively. Ratios of 0 and 1 in the AND, OR and XOR functions outputs are 0.75 to 0.25, 0.25 to 0.75 and 0.5 to 0.5, respectively. Therefore, the error
rate between the AND/OR teacher function with random binary data is 0.25, while in the case of the XOR teacher function it becomes 0.5. These results show that the nanomagnets array reservoir shown in Fig. 1 can perform AND, OR and XOR functions up to the input delay of three steps.

In other words the nanomagnets array in Fig. 1 can consider as a four-bit shift register. With this scheme, three rows, i.e. one data row and two buffer rows, are required to perform one bit-shift operation. Therefore, an array with 10 (= $3 \times 3 + 1$) rows can perform four bit shift operation and data older than three step will be lost. This is the reason that the input delay $n$ is limited to up to three steps. By increasing the number of elements in the row, we can increase the upper limit of the input delay. However, this requires more accuracy for the magnetization measurements in proportion to the increment of the distances between the input nanomagnets and nanomagnets that store the oldest data. Here we only demonstrated the nanomagnets array in a square lattice. However, in reservoir computing, the connection strength between the nodes is generally chosen to be random in order to perform general tasks. A randomly arranged nanomagnets array has a good potential to act as the reservoir for reservoir computing and it will likely show good performance for general tasks.

In this letter, we introduced and demonstrated reservoir computing based on a nanomagnets array with static magnetizations. The nanomagnets array reservoir can be trained to perform AND, OR and XOR functions with up to the input delay of three steps. We expect that devices derived from this method can be widely used as neural network hardware based on spintronics in the near future.

This research and development work was supported by the Ministry of Internal Affairs and Communications, JAPAN.

---

*nomura@mat.eng.osaka-u.ac.jp*

[1] H. T. Siegelmann and E. D. Sontag, *Turing computability with neural nets*, Applied Mathematics Letters 4 (1991) (6), pp. 77–80

[2] R. P. Cowburn and M. E. Welland, *Room Temperature Magnetic Quantum Cellular Automata*, Science 287 (2000) (5457), pp. 1466–1468

[3] A. Imre, *Majority Logic Gate for Magnetic Quantum-Dot Cellular Automata*, Science 311 (2006) (5758), pp. 205–208
[4] H. Nomura and R. Nakatani, *NAND/NOR Logical Operation of a Magnetic Logic Gate with Canted Clock-Field*, Applied Physics Express 4 (2010) (1), p. 013004

[5] H. Nomura, N. Yoshioka, S. Miura and R. Nakatani, *Controlling operation timing and data flow direction between nanomagnet logic elements with spatially uniform clock fields*, Applied Physics Express 10 (2017) (12), p. 123004

[6] A. Orlov, A. Imre, G. Csaba, L. Ji, W. Porod and G. H. Bernstein, *Magnetic Quantum-Dot Cellular Automata: Recent Developments and Prospects*, Journal of Nanoelectronics and Optoelectronics 3 (2008) (1), pp. 55–68

[7] H. Jaeger, *The “echo state” approach to analysing and training recurrent neural networks*, techreport 148, GMD – German National Research Institute for Computer Science (2001)

[8] W. Maass, T. Natschlger and H. Markram, *Real-Time Computing Without Stable States: A New Framework for Neural Computation Based on Perturbations*, Neural Computation 14 (2002) (11), pp. 2531–2560

[9] H. Jaeger, *Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication*, Science 304 (2004) (5667), pp. 78–80

[10] D. Verstraeten, B. Schrauwen, M. D’Haene and D. Stroobandt, *An experimental unification of reservoir computing methods*, Neural Networks 20 (2007) (3), pp. 391–403

[11] L. Appeltant, M. Soriano, G. V. der Sande, J. Danckaert, S. Massar, J. Dambre, B. Schrauwen, C. Miraasso and I. Fischer, *Information processing using a single dynamical node as complex system*, Nature Communications 2 (2011), p. 468

[12] K. Vandoorne, P. Mechet, T. V. Vaerenbergh, M. Fiers, G. Morthier, D. Verstraeten, B. Schrauwen, J. Dambre and P. Bienstman, *Experimental demonstration of reservoir computing on a silicon photonics chip*, Nature Communications 5 (2014)

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

[14] K. Fujii and K. Nakajima, *Harnessing Disordered-Ensemble Quantum Dynamics for Machine Learning*, Physical Review Applied 8 (2017) (2)

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

[16] T. Furuta, K. Fujii, K. Nakajima, S. Tsunegi, H. Kubota, Y. Suzuki and S. Miwa, *Macromagnetic
Simulation for Reservoir Computing Utilizing Spin Dynamics in Magnetic Tunnel Junctions, Phys. Rev. Applied 10 (2018), p. 034063,
URL: https://link.aps.org/doi/10.1103/PhysRevApplied.10.034063

[17] M. Weisheit, S. Fahler, A. Marty, Y. Souche, C. Poinsignon and D. Givord, Electric Field-Induced Modification of Magnetism in Thin-Film Ferromagnets, Science 315 (2007) (5810), pp. 349–351

[18] T. Maruyama, Y. Shiota, T. Nozaki, K. Ohta, N. Toda, M. Mizuguchi, A. A. Tulapurkar, T. Shinjo, M. Shiraishi, S. Mizukami, Y. Ando and Y. Suzuki, Large voltage-induced magnetic anisotropy change in a few atomic layers of iron, Nat. Nanotech. 4 (2009) (3), pp. 158–161