Twin vortex computer in fluid flow

Ken Goto¹, Kohei Nakajima²,∗ and Hirofumi Notsu³

¹ Division of Mathematical and Physical Sciences, Kanazawa University, Kakuma, Kanazawa 920-1192, Japan
² Graduate School of Information Science and Technology, The University of Tokyo, Bunkyo-ku, 113-8656 Tokyo, Japan
³ Faculty of Mathematics and Physics, Kanazawa University, Kakuma, Kanazawa 920-1192, Japan
∗ Author to whom any correspondence should be addressed.
E-mail: k_nakajima@mech.t.u-tokyo.ac.jp

Keywords: vortex, reservoir computing, physical reservoir computing, fluid dynamics, twin vortex, Kármán vortex

Abstract

Fluids exist universally in nature and technology. Among the many types of fluid flows is the well-known vortex shedding, which takes place when a fluid flows past a bluff body. Diverse types of vortices can be found in this flow as the Reynolds number increases. In this study, we reveal that these vortices can be employed for conducting certain types of computation. The results from computational fluid dynamics simulations showed that optimal computational performance is achieved near the critical Reynolds number, where the flow exhibits a twin vortex before the onset of the Kármán vortex shedding associated with the Hopf bifurcation. It is revealed that as the Reynolds number increases toward the bifurcation point, the input sensitivity of the twin vortex motion also increases, suggesting the modality of information processing within the system. Our finding paves a novel path to understand the relationship between fluid dynamics and its computational capability.

1. Introduction

Fluids can be found everywhere in nature, exhibiting rich and diverse dynamics, as well as instabilities, which are a source of inspiration for many mathematicians, physicists, engineers, and biologists in the field of nonlinear science. For example, vortices in fluid dynamics play an active role in biological behavior control, such as the swimming of fish [1] or flapping of butterflies [2]. In this study, we focus on the diverse nature of fluid dynamics and make a novel attempt to elucidate its potential as a computational resource.

A vortex shedding is a renowned phenomenon in fluid dynamics that takes place when a flow collides with an obstacle in the direction of fluid travel (figure 1); a typical example is the flow past a circular cylinder. The mechanism generating vortices has attracted the attention of many researchers [3–6]. The flow in this system is steady and symmetrical at low Reynolds numbers, but at higher Reynolds numbers, a pair of vortices, known as a twin vortex, are generated behind the object. A twin vortex grows in proportion with the Reynolds number; however, when the Reynolds number exceeds a certain value, vibration occurs downstream, resulting in a Kármán vortex, which has two alternating rows of vortices (cf figure 2(A) and supplementary videos (https://stacks.iop.org/NJP/23/063051/mmedia)). In this study, the flow past a circular cylinder was used as a test bed for clarifying the computational performance of fluid dynamics and the types of vortices they form.

Several attempts have been made to implement computations that focus on fluids. The construction of logic circuits using the fluidity of a droplet and the potential applications for soft robots are typical examples [7–9]. In this paper, we propose another paradigm of information processing based on fluids, where we managed a machine learning framework called reservoir computing (RC) [10, 11]. RC is a framework of recurrent neural network learning and an information processing technique that exploits the input-driven transient behaviors of high-dimensional dynamical systems, called a reservoir. This method can be implemented by adjusting linear and static readout weights from a reservoir. Moreover, an arbitrary
Figure 1. Behavior of a typical vortex shedding when the flow is past a circular object, and an illustration of node preparation for the reservoir. The grid behind the cylinder represents the nodes used for the reservoir; let the set of nodes be \( \Omega_R \). The time series plots are normalized results (\( u \) is velocity, and \( p \) represent pressure) of the fluid simulation and the input. However, rather than using all such time series in the reservoir, the time data immediately preceding the next input are extracted and used. The total number of output/observation nodes is \((10 + 1) \times (6 + 1) \times 3 = 231\). Only several time series from a single spatial location are shown for clarity. All these diagrams are drawn under the conditions of \( Re = 50 \) and \( \tau \in [0.99, 1.01] \).

Figure 2. Fluid behavior and the analysis of bifurcation according to the Reynolds number. (A) The vortex types denote fluid behavior according to the Reynolds number. (B) Denoting the Hopf bifurcation using the second component of numerical velocity, \( u_{2,k} \), which is simply denoted by \( u_2 \) in this caption. The mean and deviation for \( \max_{P \in \Omega_R, k \in [0,T_{r} - 1]} u_2(P,(k + 1)\tau) \) and \( \min_{P \in \Omega_R, k \in [0,T_{r} - 1]} u_2(P,(k + 1)\tau) \) are plotted according to the Reynolds number. (C) The phase portrait \((u_2, \partial u_2/\partial \tau)\), which was constructed at \((x_1, x_2) = (5.5, 0.0)\) and \( Re = 10, 40, 50, \) and 100.

time series can be accurately generated when the reservoir exhibits sufficient nonlinearity and memory \([12]\). A recent development in RC is the use of the dynamics of a physical system as a reservoir, which can be referred to as physical reservoir computing (PRC) \([13]\). Examples of this implementation can be found in a range of physics fields, from mechanical \([14–16]\) to photonics \([17–19]\), spins \([20–25]\), and quantum \([26–28]\), where each platform exhibits a particular computational property intrinsic to its respective spatio-temporal scales. It is important to note that the first PRC was implemented using the dynamics of a water surface \([29]\), where the mechanical inputs are injected to the surface as perturbations and the resulting shock waves propagate and interact each other to process input information. We build on this seminal approach and investigate the computational account of vortex types in fluid flow, which has not been clarified to date. Similar studies have recently been proposed to implement a recurrent neural network in physical wave systems \([30]\) and to exploit nonlinear waves, such as rogue waves, dispersive shocks, and solitons, as a reservoir for neuromorphic devices \([31, 32]\). Because vortices can be found universally in nature, if we could exploit them for our computational purposes, then we can outsource the computational loads to physical dynamics, which results in energy-efficient information processing.

In this study, we exploited the dynamics of flow past a circular cylinder as a reservoir to solve temporal machine learning tasks, and thereby systematically analyzed the fluid computational capabilities. The reservoir dynamics was simulated through Navier–Stokes equations expressed by a set of partial differential
equations (PDE). First, we investigated the bifurcation structure of the reservoir driven by the input stream to understand its dynamic property according to the Reynolds number and show that it exhibits Hopf bifurcation. Second, we analyzed the echo state property (ESP) \cite{33}, which provides a prerequisite for dynamics to function as a successful reservoir. We will demonstrate the relationship between the ESP and the stability of the fluid dynamics associated with the bifurcation structure. It was found that the ESP breaks down when the Reynolds number reaches the critical bifurcation point at the onset of the Kármán vortex shedding associated with symmetrical breaking of the twin vortex. Third, we focus on the morphological behavior of the twin vortex before the critical point and reveal that the long-diameter oscillation of the twin vortex is sensitive to input stream, suggesting a mechanism for the processing of input information in the system. Next, we implemented two typical benchmark tasks in the machine learning context, using the reservoir to demonstrate the range of computational capabilities and the dependence of the computational performance on the Reynolds number. The optimal computational performance was achieved near the critical Reynolds number, where the flow exhibits a twin vortex before the onset of Kármán vortex shedding. Finally, we demonstrate a prediction task of missing fluid variables, using PRC as an illustration of the potential application scenario of our scheme. To the best of our knowledge, this study is the first to discuss the computational capability of vortices generated using fluid dynamics.

2. Results

We start by introducing our reservoir settings in the next subsection. Subsequently, we move to analyze the reservoir property from dynamical systems point of view, including bifurcation structures and input-driven phenomena. Then, we investigate the information processing capability of our reservoir through a number of simulation experiments and show how the dynamical property of the system affects the information processing capability.

2.1. Flow past a circular cylinder as a reservoir

As for our reservoir system, we considered flows past a circular cylinder governed by the two-dimensional Navier–Stokes equations with inputs and solved it numerically using the stabilized Lagrange–Galerkin (LG) method \cite{34, 35}. The details of this problem, this method, and the input settings are provided in appendix and in the supplementary materials (SM). Here, we provide the definition of our reservoir using the numerical solution (\(u_h, p_h\)) = ((u_{h1}, u_{h2})^T, p_h), which approximates the pair of velocity \(u = (u_1, u_2)^T\) and pressure \(p\) of fluid flow, where \(h\) is a mesh size in the computation and the superscript \(T\) means the transposition. We used two parameters, \(\Delta t\) and \(\tau\), for time in the flow simulation. \(\Delta t\) is a time-step size in the simulation and \(\tau\) is a transient time for the reservoir that determines its timescale. Although the sizes of \(h\) and \(\Delta t\) are related to the ‘error’ between the numerical solution (\(u_h, p_h\)) and the exact solution (\(u, p\)), we describe the setup of our reservoir here without \(\Delta t\) to avoid confusion and with subscript \(S\) to indicate only that (\(u_h, p_h\)) is numerical solution, and the set of values of (\(u_h, p_h\)) at point \(P = (P_1, P_2)^T \in \mathbb{R}^2\) and time \(k\tau\) \((k \in \mathbb{N})\) is denoted by (\(u_{h}(P, k\tau), p_{h}(P, k\tau)\)). The flow passes a unit circular cylinder located at the origin. Let \(\Omega\) be a bounded computational domain in \(\mathbb{R}^2\) considered for the reservoir node preparations and

\[
\Omega_R = \{ \mathbf{P} = (P_1, P_2)^T \in \mathbb{R}^2; \ P_1 = i + 0.5, \ i = 0, 1, \ldots, 10, \ P_2 = -3, -2, \ldots, 2, 3 \} \quad (\subset \Omega)
\]

be a set of points that, for the sake of simplicity, were rewritten as \(\Omega_R = \{ \mathbf{P}^i \}_{i=1}^{S} \) for \(S = \#\Omega_R = 77\) (cf figure 1). Let \(\tau = 5\) \((\in \mathbb{R})\) be a transient time in the flow simulation, \(T_r = 10500\) \((\in \mathbb{R})\) be the final time of the flow simulation, \(T_r = \lfloor T/\tau \rfloor = 2100\) \((\in \mathbb{N})\) be the total time in our reservoir, and \(N_r = 3S = 231\) \((\in \mathbb{N})\) be the total number of computational nodes in our reservoir. The values of numerical solution at time \((k + 1)\tau\),

\[
\{ (u_{h}(P^i, (k + 1)\tau), p_{h}(P^i, (k + 1)\tau)); \ P^i \in \Omega_R \},
\]

correspond to the \(i\)th input of reservoir, \(r_i \in \mathbb{R}\) \((k \in \mathbb{N} \cup \{0\})\), used most of the time interval \([k\tau, (k + 1)\tau]\). Throughout this study, a one-dimensional random number sequence, \(\{r_i\}_{i=0}^{T_r} \subset \mathbb{R}\), was used for the input stream \(z(t)\) and imposed as the (inflow) Dirichlet boundary condition, \(\mathbf{u}(\mathbf{x}, t) = \mathbf{z}(t) = (z(t), 0)^T\) on the inflow boundary, in the system, cf the SM for the construction of \(z(t)\) from \(\{r_i\}_{i=0}^{T_r}\). The range of \(r_i\) was fixed at \([0.99, 1.01]\) in the studies discussed in the main text, and multiple different conditions in the input range are investigated in the SM. Using the numerical solution, (\(u_h, p_h\)), we
defined our reservoir \( \{ x^\ell_k \in \mathbb{R}; \; i = 1, \ldots, N_\ell, \; k = 0, \ldots, T_\ell - 1 \} \) (\( N_\ell = 3S = 231, \; T_\ell = 2100 \)) by
\[
\begin{align*}
x^\ell_k & = \eta_1(q^\ell, (k + 1)\tau), \\
x^\ell_k & = \eta_2(q^\ell, (k + 1)\tau), \\
x^\ell_k & = \eta_3(q^\ell, (k + 1)\tau),
\end{align*}
\]
for \( \ell = 1, \ldots, S(= 77) \). In short, we employ the state of fluid flow at time \((k + 1)\tau\) as our reservoir corresponding to the \( k \)th input at time \( k\tau \).

Then, the output \( y_k \) was computed with
\[
\hat{y}_k = \sum_{i=0}^{N_r} w^r_{\text{out}} x^i_k,
\]
where \( x^i_k = 1 \) is a bias and \( w^r_{\text{out}} \) is the readout weight of the \( i \)th computational node \( x^i_k \). As usual, in the framework of RC, the target function, \( y_k = f(t_k, t_{k-1}, \ldots) \), was learned by adjusting the linear readout weights. The ridge regression, known as \( L^2 \) regularization [36], was employed to obtain the optimal weights. Details of the training procedures for the weights are given in appendix. The performance of the system output is evaluated by comparing \( \hat{y}_k \) with the target output \( y_k \) in the evaluation phase.

### 2.2. Dynamical property of the flow reservoir

In this subsection, we first analyze the bifurcation structure of our reservoir according to the change of Reynolds number and then investigate how the response property to external inputs is regulated according to this bifurcation. Furthermore, the vortex structures, emerging before the onset of the bifurcation point, and its behavior to the external input are investigated in detail.

#### 2.2.1. Hopf bifurcation of the cylinder wake with inputs

The bifurcation structure of the cylinder wake without inputs was first studied theoretically and numerically by Dušek and Le Gal [37], who revealed it to be a Hopf bifurcation. To date, however, the bifurcation structure of a system with inputs has not been clarified. Figure 2(A) shows the typical behavior of vortices according to each Reynolds number in our system. A twin vortex (Re = 10 and Re = 40) and Kármán vortex (Re = 50 and Re = 100) are observed. Figures 2(B) and (C) show the behavior of circular cylinder flow using a max/min-velocity diagram and a phase portrait, respectively. We then find that velocity is the periodic type solution at Re = 50 and 100, which means our system has non-chaotic behavior, and the Hopf bifurcation occurs near the critical Reynolds number (around Re = 45). This result is similar to the system without inputs.

#### 2.2.2. Echo state property and common-signal-induced synchronization

When a signal-driven dynamical system is used as a successful reservoir, the dynamical system must accomplish common-signal-induced synchronization [38], which is an outcome of ESP [33]. We analyzed two measures, \( E_m \) and \( \lambda \), in each Reynolds number, where \( E_m \) represents an indicator of synchronization and asymptotically converges to 0 if synchronization occurs, and \( \lambda \) is the amplification factor of perturbation and represents the stability/instability of the flow field (see appendix for the definitions of \( E_m \) and \( \lambda \)). Figure 3 shows the outcome of the averaged \( |1 - E_m| \) and \( \lambda \) with each Reynolds number. Both \( |1 - E_m| \) and \( \lambda \) tend to increase depending on the Reynolds number, suggesting a critical point \((\in [40, 50])\) where synchronization breaks down during the transition from the twin vortex to the Kármán vortex. The synchronization phenomenon is unlikely to occur under the condition of Re \( \geq 50 \) (i.e. the Kármán vortex is generated). However, in the region where synchronization is collapsed (Re \( \in (45, 100) \)), periodic behavior occurs (figures 2(B) and (C)). Furthermore, the onset of flow instability near the critical Reynolds number has been known to be a manifestation of Hopf bifurcation. Note that the critical point differs from that of chaos in the flow field (chaotic behavior appears at a larger Re (cf. e.g. [37, 39])). In our setup, \( z \) uses an input with a small swing, such as \([0.99, 1.01]\); thus, \( \lambda \) is almost the same for systems without input (figure 3). Interestingly, the behaviors of \( |1 - E_m| \) and \( \lambda \) correspond to the task results presented in figure 5, which will be explained in detail in a later section.

#### 2.2.3. Twin vortex behavior

The twin vortices are known to grow large according to the Reynolds number. We calculated the long diameter of the twin vortices using the stream function (figure 4(A)). The behavior of this long diameter becomes a brief indicator of the input effects because the long diameter is observed to be oscillating in response to the input stream (see the supplementary videos). To clarify the vortex internal structures of our process, we investigate the long diameter of the twin vortices using the stream function (figure 4(A)).
Figure 3. Analysis of ESP and common-signal-induced synchronization. The mean for $|1 - E_m|$ and $\lambda$ are plotted according to the Reynolds number. The error bars show standard deviation.

Figure 4. Analysis of the twin vortices’ long diameters and input sensitivity according to the Reynolds number. (A) Overview of the long diameters of twin vortices. (B) The long-diameter range according to the Reynolds number. The red line represents the averaged long diameter, and the gray fill indicates the max/min long diameter at $Re = 10, 20, 30, 40, \text{ and } 45$. (C) The mean of cross-correlations $Cor_d(i)$, $i = 1, 2$, which are between the stream function at the two fixed points $(x_1, x_2) \approx (0.52, -0.2), (3.3, -0.32)$, and the inputs are plotted according to the delay $d$ by the Reynolds number. The error bars show the standard deviation in each plot.

system, we focus on the twin vortices (i.e. $Re \in [10, 45]$) and, here, analyze the input-sensitivity of the twin vortices. Primarily, the twin vortices are described, but as a comparison, the Kármán vortex cases are also described. Let us introduce the long diameter of the twin vortices:

$$\gamma(t) = \max \{x_i; \psi(x, t) > 0, x = (x_1, x_2)^T \in \Omega, x_2 < 0\},$$

$$\tilde{\gamma}(t) = \max \{x_i; \psi(x, t) < 0, x = (x_1, x_2)^T \in \Omega, x_2 > 0\},$$

where $\psi$ is the stream function satisfying $u_1 = \partial \psi / \partial x_2$ and $u_2 = -\partial \psi / \partial x_1$. In figure 4(B), the averaged $(\gamma + \tilde{\gamma})/2$ and max/min $\gamma$ or $\tilde{\gamma}$ for time are plotted according to the Reynolds number. We observed the oscillation of the long diameter by the numerical simulation, and then the long diameter was calculated. Subsequently, we determine the expanded long-diameter range with the input effects (figure 4(B)). Next, we calculated the cross-correlation between inputs $r_n$ and the stream function $\psi_1^n = \psi(x_{min}, n\Delta t), \psi_2^n = \psi(x_{max}, n\Delta t)$, where $x_{min} \approx (0.52, -0.2)$ and $x_{max} \approx (3.3, -0.32)$. These coordinates represent the
Figure 5. Results of evaluations of memory capacity and NARMA tasks. (A) Results for linear and nonlinear memory capacity were plotted at each Reynolds number. (B) Results for the NARMA tasks, in terms of NMSE, were plotted at the Reynolds numbers of $Re = 10, 20, 30, 40, 45, 50, 60, 75,$ and $100.$ In both (A) and (B), the plots and error bar show the mean and standard deviation, respectively. We clarified the involvement of the vortex in computational capability by comparing LR, no-object, and ESN with those of the system. We determined that the peak of the computational performance was approximately $Re = 40,$ immediately before the bifurcation point. The background color shading indicates whether the flow was stable (yellow) or unstable (purple).

2.3. Analysis of computational performances

We systematically investigated the computational performance of our reservoir through a number of experiments. As for comparisons, we discuss the results obtained by the flow simulated without the cylinder (called no-object), which can clarify the contribution of vortices to the reservoir dynamics in some cases. Furthermore, we compare the performance of our system with a simple linear regression (LR) model, $y_k = w_{LR}^1 r_k + w_{LR}^0,$ which bypassed the reservoir part and directly connected the input to the readout, clarifying the contribution of the reservoir, and with an echo-state network (ESN) [10], which is the standard recurrent neural network (details for the setting of ESN are provided in the SM). For these models, training data, conditions, and procedures are kept the same with our system for fair comparisons.

2.3.1. Evaluation of the memory capacity

Here, we evaluated the memory capacity of the system by investigating whether the system can reproduce previous inputs and nonlinearly process them using its current states. In this study, we applied Legendre polynomials for each time step expressed as $y_k = P_n(r_k - d),$ and $d$ represents delay given a value, $d = 0, 1, \ldots, 50:$

$$P_n(l) = \frac{1}{2^n n!} \frac{d^n}{d^l} [(l - 1)^n].$$
Figure 6. A typical performance on the NARMA tasks in the evaluation phase at $Re = 40$. In each plot, the performance of the LR system, system, and no-object are compared with the target. We confirmed that the system can trace accurately through the involvement of a vortex.

The finite products of Legendre polynomials for each time step were used in [12]; however, for simplicity, the nonlinear degree of Legendre polynomials for each delay was changed and treated as a target in the present study, for example, if $n = 3$, $P_3 = \frac{1}{2}(5y_k^3 - \cdots)$. In this task, we used the memory function and memory capacity as follows:

$$\text{MF}(d) = \frac{\text{cov}^2(y_k, \hat{y}_k)}{\sigma^2(y_k)\sigma^2(\hat{y}_k)},$$

$$\text{MC} = \sum_{d=1}^{\infty} \text{MF}(d),$$

where $\sigma^2(y_k)$ represents the variance of $y_k$. The objective of this task was to quantify whether the system could regenerate previous inputs and emulate the nonlinear functions of previous inputs. The quantified values are referred to as linear and nonlinear memory capacity, respectively. The memory capacity showed a high value if the system outputs successfully emulated the target outputs for each target delay. Figure 5(A) shows the results of MC in the evaluation phase. The most striking feature is that MC increased as the system approached the critical Reynolds number and suddenly decreased if the Reynolds number exceeded the critical value. In other words, MC increased as the state of the dynamics neared the move from the twin vortex to the Kármán vortex and significantly decreased after the transition. At $Re = 40$, immediately before the bifurcation point, the system had the most memory, with $MC_{Re=40} \approx 9.579$ as the actual value of linear memory. Moreover, the extensive twin vortices had more input sensitivity (cf figure 4(C)), and we infer that the implementation of greater computational capability occurred via greater input sensitivity. We confirmed that the highest MC in the system showed a similar value for linear memory as the results for ESN10 (which has ten nodes; this expression applies hereafter), while system performance in the case of
nonlinear memory was comparable to ESN100. Furthermore, our system can represent both odd and even functions, whereas some reservoirs, such as ESN, can only demonstrate one of the functions because of the constraint of the setting of activation functions [12].

### 2.3.2. Emulating nonlinear dynamical systems

We here evaluate whether our system is capable of emulating nonlinear dynamical systems using nonlinear autoregressive moving average (NARMA) models (see appendix for details). The objective of this emulation task was to test the system capability to process input history with nonlinearity. The essential point of the NARMA model was to include the long-term dependencies of the system with \( n \)th time lag, where the corresponding NARMA model is called NARMA \( n \) model in this study.

Figure 5(B) shows the task performance of our system with respect to error in the evaluation phase. The averaged normalized mean squared error (NMSE) was calculated in each NARMA task separately based on each Reynolds number as \( \text{NMSE} = \frac{\sum (y_k - \hat{y}_k)^2}{\sum y_k^2} \). We compared the system with no-object and observed that vortices work effectively, depending on the task difficulty and the Reynolds number. We also conducted comparisons of the system based on ESN or LR and observed that the fluid dynamics worked effectively. The results showed that the Reynolds number at optimal performance coincided with the critical Reynolds number of the system (figure 5(B)). In NARMA2, NARMA3, and NARMA4, system performance at \( \text{Re} = 40 \) was much better than the result of ESN with 600 nodes. For example, system performance at \( \text{Re} = 10 \) and 100 was inferior to the performance level at \( \text{Re} = 40 \) (figure 5 and the supplementary videos). The plots in figure 6 are representative examples of the actual performance of NARMA2, NARMA3,
NARMA4, and NARMA5 at Re = 40 and demonstrate that the system can clearly trace the target model when compared with the LR system (see the supplementary videos). We confirmed that tracking accuracy was improved by the involvement of a vortex, in comparison with that observed in the no-object case. We can infer that nonlinear processing capability was low at Re = 10 because the input sensitivity of vortices was low even though the synchronization occurred and that the computational capability was low at Re = 100 because synchronization was broken.

2.3.3. Time series predictions of missing fluid variables

RC enables highly accurate predictions of time series data and can estimate the Lyapunov exponents of high dimensional spatio-temporal chaotic systems [40, 41]. The behaviors of fluid variables [42] or the two-dimensional turbulent Rayleigh–Bénard convection flow [43] has also been predicted Modeled using RC. This phase aimed to implement the prediction of fluid variables (velocity and pressure) using the own dynamics of the system based on the framework of PRC. Dynamics other than the variable to be predicted are used as the reservoir to predict the unknown variable. Thus, if the target output is $u_1$, such as, $y_k = u_1(P', k\tau)$, the reservoir \{\(x_k^{(1)} \in \mathbb{R}; \ i = 1, \ldots, N_{rr}, \ k = 1, \ldots, T_r\} is defined by

\[
\begin{align*}
    x_k^{(2(-1)+1} & = u_2(P', (k - 1) \tau + \Delta t), \\
    x_k^{(2(-1)+2} & = p(P', (k - 1) \tau + \Delta t),
\end{align*}
\]

for \(\ell = 1, \ldots, S\) and \(N_{rr} = 2S = 154\).

Figure 7 depicts a typical example of an error, with the target output and actual performance for each target. The color map in figure 7 shows that errors increased as the Reynolds number increased; however, the actual performance successfully overlapped the target (figures 7(A)–(C)). As shown in figure 7(D) for each target, although $u_2$ was moderate, errors of $u_1$ and $p$ increased remarkably according to the Reynolds number, especially when the Reynolds number exceeded a critical number. This observation indicates that making predictions under conditions where a Kármán vortex is generated is relatively more difficult than when a twin vortex is generated. Figure 7(D) shows that time series prediction was easy with twin vortices in all cases. Time series prediction using the Kármán vortex was difficult with LR and ESN but comparatively easier using the PRC system. The ESP of the system was collapsed in the region Re $\geq 50$. If the reservoir does not satisfy ESP, its dynamics cannot be uniquely determined by the input signal. Therefore, even if we used optimal weights trained in the training phase, errors increased in the evaluation phase. We can infer that prediction performance was degraded in this Reynolds number region because of flow instability, as the performance of the externally connected ESN model was significantly degraded. Moreover, in the aforementioned memory capacity and NARMA tasks, computational performance may decrease in this Reynolds number region, and we can consider that prediction performance decreases when using PRC. In the region Re $\geq 50$, the ESN did not work, and the prediction capability decreased when using other variables; however, the prediction capability of the system was better than using the ESN that constructs an input-only function because the target variable was both a function of inputs and a function of other variables that contain some of the target variable components, including its history.

3. Discussion

In this study, we first investigated the bifurcation of our system and demonstrated that Hopf bifurcation occurs, followed by the synchronization phenomenon to the input stream. We investigated ESP in terms of direct calculation using reservoir dynamics and stability analysis and determined that the synchronization phenomenon gradually collapsed as the phase shifted from a twin vortex to a Kármán vortex. However, we note that this correspondence between the stability analysis and the Reynolds number held true only when input intensity was low and did not necessarily hold true when we used a high input intensity (cf SM). The input sensitivity of vortices was investigated, and the details of its internal structure satisfying ESP were discussed, suggesting that the input sensitivity of dynamics is useful in estimating computational capability. Moreover, we demonstrated that flow dynamics including vortices can be used as a resource for temporal machine learning tasks and that computational performance is maximized in the phase just before synchronization failure. We also showed that in the Reynolds number band where ESP is collapsed, using the remainder of the variables in the same system is more effective for making predictions about the behavior of a missing variable than constructing external prediction models from scratch. This result suggests, in a sense, fluid problems are better solved by fluid computers.

There are still many free parameters that should be investigated in this system. One of the crucial free parameters that affect the information processing capability of the system would be the number of sampling points or the spatial configuration of the sampling points. By altering the sampling locations, according to
the vortex structure of the dynamics, the information processing capability would change even if the number of computational nodes is fixed. The choice of the reservoir states also would be important. In our current study, we used \((u_h, p_h)\) in each spatial point, but we have found that \(u_h\) is more important than \(p_h\), which is shown in SM. If the number of sampling points (that is related to the number of sensors to be utilized in a physical experiment) is limited, then we can use \(u_h\) as a much more effective parameter. Several parameters would alter the property of the dynamics drastically. For example, the scaling parameter of the input intensity drastically changes the bifurcation structures of the entire system, as discussed earlier (cf SM). Furthermore, the duration of time \(\tau\) assigned to each input, which is now fixed to \(\tau = 5\), would also affect the dynamics and thus alters the information processing capability of the system when varied.

Another important parameter in this context is the shape of the object. This parameter will alter the spatial structure of the vortex and would be crucial for the information processing capability of the system. Although we have focused on the vortex shedding that takes place when a fluid flows past a bluff body, we expect that our scheme presented in this study, investigating the ESP and the common-signal-induced synchronization, can be easily applied to various types of fluid systems and is useful.

Another important next step would be to prove the universality of the proposed system. The universality of the RC system has been studied since the original model of the liquid state machine was proposed [11]. Because RC is targeting temporal tasks rather than static tasks in general, universality in this context means to uniformly approximate any filter with a fading memory property. It was shown that if the system consists of a filter bank, containing a number of filters with fading memory, with a point-wise separation property and of a readout with a universal approximation property, then this system can approximate any filter with a fading memory property in any precision. This result was induced based on the work by Boyd and Chua [44], and recently, there have been interesting theoretical results suggesting that the ESN with a linear and static readout can also give rise to a universal approximator of filters having fading memory in some conditions [45–47]. Although it is out of scope of this paper, it should be a fruitful direction to explore in the future.

Considering that the vortices in fluid flow can be found universally in nature and technology, the potential of our findings cannot be overestimated. We expect our findings to be verified in real physical experiments as a future step. Experiments using particle image velocimetry (PIV) would be one of the realistic options for this purpose. Toward the real implementation of the PRC with the fluid flow, it is important to check the effect of noise. We have investigated it in the NARMA\(n\) task with \(n = 2, 3, 4, 6\) for a fixed Reynolds number, \(Re = 40\), where the input variance \(\sigma^2\) takes five values up to 20\% (cf SM). The results are natural, that is, the effect of noise becomes large when the noise increases. In addition to the numerical experiments for the noise effect, we also investigated the computational performance of the twin vortex system by using just velocity (or just pressure) in the NARMA\(n\) (\(n = 2, 3, 4, 6\)) and the memory capacity tasks (cf SM), as the PIV can measure velocity only and the measurement of pressure might not be easy. It is observed that the computational performance of the system with just velocity in the NARMA task is similar to the one with both velocity and pressure as presented and that the pressure contribution is not significant. These additional numerical experiments provide the possibility of physical implementation of the velocity-based PRC via the PIV.

Acknowledgments

This work is supported by JST CREST Grant No. JPMJCR2014, JST PRESTO Grant No. JPMJPR16EA and JSPS KAKENHI Grant No. JP18H01135. This work is partially based on results obtained from a project commissioned by the New Energy and Industrial Technology Development Organization (NEDO).

Data availability statement

The data generated and/or analysed during the current study are not publicly available for legal/ethical reasons but are available from the corresponding author on reasonable request.

Appendix A. Computational fluid dynamics

Let \(\Omega \subset \mathbb{R}^2\) be a bounded domain; \(\partial \Omega\) is the boundary of \(\Omega\), and \(T_f\) is a positive constant. We suppose that
∂Ω comprises four parts, Γ_i ⊂ ∂Ω, i = 1, ..., 4. Our problem is to find (u, p) : Ω × (0, T_f) → R^2 × R such that

\[
\frac{\partial u}{\partial t} + (u \cdot \nabla)u - \nabla \cdot \sigma(u, p) = 0 \quad \text{in} \quad \Omega \times (0, T_f), \tag{A1a}
\]

\[
\nabla \cdot u = 0 \quad \text{in} \quad \Omega \times (0, T_f), \tag{A1b}
\]

\[
u = z \quad \text{on} \quad \Gamma_1 \times (0, T_f), \tag{A1c}
\]

\[
[\sigma(u, p)n] \cdot n^\perp = 0, \quad u \cdot n = 0 \quad \text{on} \quad \Gamma_2 \times (0, T_f), \tag{A1d}
\]

\[
\sigma(u, p)n = 0 \quad \text{on} \quad \Gamma_3 \times (0, T_f), \tag{A1e}
\]

\[
u = 0 \quad \text{on} \quad \Gamma_4 \times (0, T_f), \tag{A1f}
\]

\[
u = u^0 \quad \text{in} \quad \Omega, \quad at \ t = 0. \tag{A1g}
\]

where \( u = (u_1, u_2)^T \) is the velocity, \( p \) is the pressure, \( \sigma(u, p) = (2/Re)D(u) - pI : \Omega \times (0, T_f) \rightarrow R^{2 \times 2} \) is the stress tensor, \( D(u) = (1/2)(\nabla u + (\nabla u)^T) : \Omega \times (0, T_f) \rightarrow R^{2 \times 2} \) is the identity tensor, \( n : \partial Ω \rightarrow R^2 \) is the outward unit normal vector, \( n^\perp : \partial Ω \rightarrow R^2 \) is the unit tangential vector, \( z : \Gamma_1 \times (0, T_f) \rightarrow R^2 \) is a given boundary velocity, and \( u^0 : \Omega \rightarrow R^2 \) is a given initial velocity.

In our system, we set for \( L = 7.5 \),

\[
\Omega = \{ x \in R^2 ; \ x_1 \in (-L, 3L), \ x_2 \in (-L, L), |x| > 0.5 \},
\]

\[
\Gamma_1 = \{ x \in \partial \Omega ; \ x_1 = -L, \ x_2 \in [-L, L] \},
\]

\[
\Gamma_2 = \{ x \in \partial \Omega ; \ x_1 = (-L, 3L), \ x_2 \in (-L, L) \},
\]

\[
\Gamma_3 = \{ x \in \partial \Omega ; \ x_1 = 3L, \ x_2 \in [-L, L] \},
\]

\[
\Gamma_4 = \{ x \in \partial \Omega ; \ x | = 0.5 \}.
\]

Thus, \( Ω \) is the computational domain, \( Γ_1 \) is an inflow boundary on the left side, \( Γ_2 \) is a slip boundary on the top and bottom sides, \( Γ_3 \) is a stress-free boundary on the right, and \( Γ_4 \) is the no-slip boundary on the circle. We also set \( z = (z, 0)^T \) for an input function \( z = z(t) \in R^2 \), where the input \( z \) takes a random value at each fixed time interval (cf SM). The range of \( z \) is set as \([0.99, 1.01]\), and experiments with the other ranges are reviewed in the SM.

We solved problem (A1) numerically using the stabilized LG method [34, 35], a stabilized finite element method combined with the method of characteristics. Moreover, the material derivative was discretized along the trajectory of a fluid particle as

\[
\left[ \frac{\partial u}{\partial t} + (u \cdot \nabla)u \right](x, t^n) \approx \frac{u^n - u^{n-1} - u^n - \Delta t}{\Delta t}.
\]

where \( \Delta t \) is a time increment for the flow computation, and \( N_T = [T_f/\Delta t] \), \( t^n = n\Delta t \), and \( u^n = u(\cdot, t^n) \), \( p^n = p(\cdot, t^n) \) for \( n \in \{0, \ldots, N_T\} \). Let \( h > 0 \) be a representative mesh size. The numerical solution to be obtained by the LG method is a series of piece-wise linear continuous functions defined in \( Ω \), \( \{ (u^n_i, p^n_i) : Ω \rightarrow R^2 \times R ; n = 1, \ldots, N_T \} \). Note that \( (u^n_i, p^n_i) \) approximates \( (u^n, p^n) \). For the fully discretized scheme, please refer to the scheme in the LG method section of the SM.

Appendix B. Training the readout weights using ridge regression

We will briefly review the framework of RC. Assume a reservoir map \( \tilde{F} : R^M \times R^m \rightarrow R^m \) \( (m, M \in N) \), and a readout map \( h : R^M \rightarrow R \), an infinite discrete input \( z = (z_0, z_1, z_2, \ldots) \in (R^m)^N \), and an output signal \( y \in R^2 \), then a reservoir state vector \( x_k \in R^M \) and an output signal are determined by \( x_k = \tilde{F}(x_{k-1}, z_k) \) and \( y_k = h(x_k) \), respectively. To exploit a reservoir map as a filter, it is preferable to have its state be a function of the previous input sequence, \( x_{k-1} = E(z_{k-1}, z_{k-2}, \ldots) \), which is related to the concept of ESP, explained in the main text. These expressions are a generalization of the definitions of a reservoir computer, and the readout map \( h \) is a linear map. This map, which is expressed as a linear and static readout weights, is adjusted to emulate the target outputs.

We used ridge regression [36] following \( L^2 \) regularization in the process of learning. The detailed steps are as follows. Ridge regression was used to minimize the residual sum of squares between system and target outputs by penalizing the size of the weights in the training phase. In our setup, the training phase is defined as time domain \( 200 \leq k \leq 1399 \) and the optimal weights are \( w_{out}^j \), the problem to be solved is
\[ \beta^* = \arg \min_{\beta} \left\{ \sum_{t=200}^{1399} (y_k - \hat{y}_k)^2 + \lambda_R \sum_{\ell=0}^{\nu} (w^\ell_{\text{out}})^2 \right\}, \]

where \( \lambda_R \in \mathbb{R}_{>0} \) is the ridge parameter. By collecting a training data set of 1200 timesteps, we obtained a \( 1200 \times N_r \) matrix \( X \), where \( N_r \) is the number of nodes in the reservoir. Furthermore, the target output corresponding to 1200 timesteps was determined by \( y = [y_{200}, \ldots, y_{1399}]^T \). Then we obtained the optimal weights, \( W_{\text{out}} = [w^0_{\text{out}}, \ldots, w^{N_r-1}_{\text{out}}]^T \), by

\[ W_{\text{out}} = (X^T X + \lambda_R I)^{-1} X^T y. \]

We must determine the appropriate \( \lambda_R \), for which the degrees of freedom (df) is crucial. The value of df is determined by

\[ df = \sum_{i=1}^{N_r} \frac{\lambda_i}{\lambda_i + \lambda_R}, \]

where \( \{\lambda_i\} \) are the eigenvalues of \( X^T X \). Because df is a one-to-one correspondence with \( \lambda_R \) and its maximum value is \( N_r \), we can obtain a unique \( \lambda_R \) by increasing the df from 1 to \( N_r \) by one, and the corresponding \( W_{\text{out}} \) can also be obtained. The residual sum of squares \( \text{RSS} = \sum_{k=200}^{1399} (y_k - \hat{y}_k)^2 \) can be calculated using \( W_{\text{out}} \), and we can obtain Akaike’s information criterion (AIC) as follows:

\[ \text{AIC} = M \log(\text{RSS}) + 2df, \]

where \( M \) is the training size of 1200 used in our research. By calculating AIC according to each df, the optimal \( \lambda_R \) and \( W_{\text{out}} \) that minimize AIC can be obtained.

**Appendix C. Descriptions of an indicator of synchronization and the amplification factor**

In common-signal-induced synchronization, the states of two different initial condition reservoirs driven by the same input sequence approach the same value over time. Intuitively, it indicates that the reservoir asymptotically washes away information related to the initial conditions. This condition indicates that if a certain input sequence is injected at any time, it will return the same response, which expresses the minimum characteristics necessary for the reservoir to implement a reproducible input–output relationship and calculation.

A concept similar to common-signal-induced synchronization is ESP, in which the current network state \( \bar{r}(t) \) is expressed as a function of only the previous input series \( \bar{z}(t) \), independent of the initial value \( \bar{r}(0) \). Thus, there exists \( E \) such that \( \bar{r}(t) = E(\ldots, \bar{z}(t-1), \bar{z}(t)) \) [33]. In this study, we directly investigated the degree of synchronization using two reservoir states corresponding to two identical input sequences with different initial values and using the amplification factor of perturbation [5], where, to get different initial values, different initial input sequences are used. Let \( \bar{x}_k \) and \( \bar{x}_k \in \mathbb{R}^{N_r} \) be two reservoir states corresponding to two identical input sequences with different initial values; furthermore, we calculated \( E_m \) as follows:

\[ E_m = \left\| \bar{x}_{k+1} - \bar{x}_k \right\|_{\ell^2(\Omega)} \left\| \bar{x}_{k+1} - \bar{x}_k \right\|_{\ell^2(\Omega)}^{-1} \]

for the norm \( \left\| \bar{x}_k \right\|_{\ell^2(\Omega)} = \left( \sum_{i=1}^{N_r} (x^i_k)^2 \right)^{1/2} \). \( E_m \) asymptotically converges to 1 if \( \bar{x}_k \) and \( \bar{x}_k \) systems are synchronized with each other.

Then, the stability analysis of the fluid solution using perturbation was used to investigate whether the perturbation given as an initial value increased or decreased with time. The amplification factor of the perturbation was obtained by numerically solving the perturbation equation as described in the SM, and we obtained the following \( \lambda \) (let \( \bar{u}_k = \{\bar{u}_k\}_{k=0}^{N_r} \) be the perturbation for the velocity obtained by solving the perturbation equation in the SM):

\[ \lambda = \frac{1}{n} \log \left( \frac{\left\| \vec{u}^0 \right\|_{\ell^2(\Omega)}}{\left\| \vec{u}^0 \right\|_{\ell^2(\Omega)}} \right), \]

which is the amplification factor of the perturbation \( \bar{u}_k \). If \( \lambda > 0 \), the perturbation increases exponentially, and the solution is unstable in \( \Omega \) in this sense. Because this stability analysis is usually performed for steady solutions, we analyzed our setup and the steady solution for comparison, conducting the analysis at \( z = 1 \) (called no-input). Notably, the amplification factor of the perturbation, obtained by stability analysis of the fluid solution and the Lyapunov exponent (an indicator used to quantitatively evaluate initial value sensitivity containing a feature of chaos), differ in derivation.
Appendix D. Descriptions of NARMA models

NARMA models used in this study were proposed by Atiya and Parlos [48]. The essential point of the NARMA model was to include the long-term dependencies of the system with nth time lag, which requires learning systems to possess certain amount of nonlinearity and memory to emulate it. We initially introduce the NARMA model of a second-order nonlinear dynamical system as follows:

\[ y_{k+1} = 0.4y_k + 0.4y_{k-1} + 0.6r_k^2 + 0.1, \]

where \( y_k \) and \( r_k \) are the state and input of the model at timestep \( k \), respectively. For the input \( r_k \), the same uniform random real value used to the reservoir system was linearly scaled to the range \([0, 0.2]\) and injected.

In this study, this system was called the NARMA2, and the NARMA system such that the nth order nonlinear dynamical system can be written as follows:

\[ y_{k+1} = 0.3y_k + 0.05y_k \left( \sum_{j=0}^{n-1} y_{k-j} \right) + 1.5r_{k-n+1}r_k + 0.1. \]

We have varied \( n \) from 3 to 6 and tested the performance. Accordingly, these tasks were called NARMA3, NARMA4, NARMA5, and NARMA6 in the main text.

ORCID iDs

Kohei Nakajima https://orcid.org/0000-0001-5589-4054

References

[1] Liao J C, Beal D N, Lauder G V and Trianfayllou M S 2003 Fish exploiting vortices decrease muscle activity Science 302 1566–9
[2] Srygley R B and Thomas A L R 2002 Unconventional lift-generating mechanisms in free-flying butterflies Nature 420 660–4
[3] Oerter H Jr 1990 Wakes behind blunt bodies Annu. Rev. Fluid Mech. 22 539–62
[4] Huere P and Monkekwitz P A 1990 Local and global instabilities in spatially developing flows Annu. Rev. Fluid Mech. 22 473–537
[5] Takimoto Y and Mizushima J 2010 Mechanism of sustained oscillations in a fluid flowing past a circular cylinder obstacle Phys. Rev. E 82 056316
[6] Chomaz J-M 2005 Global instabilities in spatially developing flows: non-normality and nonlinearity Annu. Rev. Fluid Mech. 37 357–92
[7] Prakash M and Gershenfeld N 2007 Microfluidic bubble logic Science 315 832–5
[8] Katsikis G, Cybulski J S and Prakash M 2015 Synchronous universal droplet logic and control Nat. Phys. 11 588–96
[9] Wehner M, Truby R L, Fitzgerald D J, Mosadeq B, Whitesides G M, Lewis J A and Wood R J 2016 An integrated design and fabrication strategy for entirely soft, autonomous robots Nature 536 451–5
[10] Jaeger H and Haas H 2004 Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication Science 304 78–80
[11] Maass W, Natschläger T and Markram H 2002 Real-time computing without stable states: a new framework for neural computation based on perturbations Neural Comput. 14 2531–60
[12] Dambre J, Verstraeten D, Schrauwen B and Massar S 2012 Information processing capacity of dynamical systems Sci. Rep. 2 514
[13] Nakajima K 2020 Physical reservoir computing-an introductory perspective Japan J. Appl. Phys. 59 060501
[14] Nakajima K, Li T, Hauser H and Pfeifer R 2014 Exploiting short-term memory in soft body dynamics as a computational resource J. R. Soc. Interface. 11 20140437
[15] Nakajima K, Hauser H, Li T and Pfeifer R 2015 Information processing via physical soft body Neurorobotics 82 056316
[16] Nakajima K, Hauser H, Li T and Pfeifer R 2018 Exploiting the dynamics of soft materials for machine learning Neurocomputing 339–47
[17] Vandoorne K, Dierckx W, Schrauwen B, Verstraeten D, Baets R, Bienstman P and Campenhout J V 2008 Toward optical signal processing using photonic reservoir computing Opt. Express 16 11182–92
[18] Larger L, Soriano M C, Brunner D, Appelant L, Gutierrez J M, Pesquera L, Mirasso C R and Fischer I 2012 Photonic information processing beyond Turing: an optoelectronic implementation of reservoir computing Opt. Express 20 5241–9
[19] Brunner D, Soriano M C, Mirasso C R and Fischer I 2013 Parallel photonic information processing at gigabyte per second data rates using transient states Nat. Commun. 4 1364
[20] Torrejon J et al 2017 Neuromorphic computing with nanoscale spintronic oscillators Nature 547 428–31
[21] Furuta T, Fujii K, Nakajima K, Tsunegi S, Kubota H, Suzuki Y and Miwa S 2018 Macromagnetic simulation for reservoir computing utilizing spin dynamics in magnetic tunnel junctions Phys. Rev. Appl. 10 034063
[22] Tsunegi S, Taniguchi T, Nakajima K, Miwa S, Yakushiji K, Fukushima A, Yuasa S and Kubota H 2019 Physical reservoir computing based on spin torque oscillator with forced synchronization Appl. Phys. Lett. 114 164101
[23] Yamaguchi T, Akashi N, Tsunegi S, Kubota H, Nakajima K and Taniguchi T 2020 Periodic structure of memory function in spintronics reservoir with feedback current Phys. Rev. Res. 2 022389
[24] Yamaguchi T, Akashi N, Nakajima K, Kubota H, Tsunegi S and Taniguchi T 2020 Step-like dependence of memory function on pulse width in spintronics reservoir computing Sci. Rep. 10 1–8
[25] Akashi N, Yamaguchi T, Tsunegi S, Taniguchi T, Nishida M, Sakurai R, Wakao Y and Nakajima K 2020 Input-driven bifurcations and information processing capacity in spintronics reservoirs Phys. Rev. Res. 2 043303
[26] Fujii K and Nakajima K 2017 Harnessing disordered-ensemble quantum dynamics for machine learning Phys. Rev. Appl. 8 024030
[27] Nakajima K, Fujii K, Negoro M, Mitarai K and Kitagawa M 2019 Boosting computational power through spatial multiplexing in quantum reservoir computing Phys. Rev. Appl. 11 034021
[28] Ghosh S, Opala A, Matuszewski M, Paterek T and Liew T C 2019 Quantum reservoir processing Npj Quantum Inf. 5 1–6
[29] Fernando C and Sojakka S 2003 Pattern recognition in a bucket Advances in Artificial Life (Berlin, Heidelberg, Germany: Springer) pp 588–97
[30] Hughes T W, Williamson I A D, Minkov M and Fan S 2019 Wave physics as an analog recurrent neural network Sci. Adv. 5 eaay6946
[31] Marcucci G, Pierangeli D and Conti C 2020 Theory of neuromorphic computing by waves: machine learning by rogue waves, dispersive shocks, and solitons Phys. Rev. Lett. 125 093901
[32] Silva N A, Ferreira T D and Guerreiro A 2021 Reservoir computing with solitons New J. Phys. 23 023013
[33] Jaeger H 2010 The ‘echo state’ approach to analysing and training recurrent neural networks GMD Report: German National Research Center for Information Technology 148
[34] Süli E 1988 Convergence and nonlinear stability of the Lagrange–Galerkin method for the Navier–Stokes equations Numer. Math. 53 459–83
[35] Notsu H and Tabata M 2016 Error estimates of a stabilized Lagrange–Galerkin scheme for the Navier–Stokes equations Numer. Math. 50 361–80
[36] Hastie T, Tibshirani R and Friedman J 2009 The Elements of Statistical Learning: Data Mining, Inference, and Prediction (Springer Series in Statistics) 2nd edn (New York: Springer)
[37] Dušek J, Le Gal P and Fraunie P 1994 A numerical and theoretical study of the first Hopf bifurcation in a cylinder wake J. Fluid Mech. 264 59–80
[38] Lu Z, Hunt B R and Ott E 2018 Attractor reconstruction by machine learning Chaos 28 061104
[39] Williamson C H K 1996 Vortex dynamics in the cylinder wake Annu. Rev. Fluid Mech. 28 477–539
[40] Pathak J, Lu Z, Hunt B R, Girvan M and Ott E 2017 Using machine learning to replicate chaotic attractors and calculate Lyapunov exponents from data Chaos 27 121102
[41] Lu Z, Pathak J, Hunt B, Girvan M, Brockett R and Ott E 2017 Reservoir observers: model-free inference of unmeasured variables in chaotic systems Chaos 27 041102
[42] Nakai K and Saiki Y 2018 Machine-learning inference of fluid variables from data using reservoir computing Phys. Rev. E 98 023311
[43] Pandey S and Schumacher J 2020 Reservoir computing model of two-dimensional turbulent convection Phys. Rev. Fluids 5 113506
[44] Boyd S and Chua L 1985 Fading memory and the problem of approximating nonlinear operators with Volterra series IEEE Trans. Circuits Syst. 32 1130–61
[45] Grigoryeva L and Ortega J-P 2018 Echo state networks are universal Neural Netw. 108 495–508
[46] Gomon L and Ortega J P 2019 Reservoir computing universality with stochastic inputs IEEE Trans. Neural Netw. Learn. Syst. 31 100–12
[47] Cachiero C, Gomon L, Grigoryeva L, Ortega J P and Teichmann J 2020 Discrete-time signatures and randomness in reservoir computing (arXiv:2010.14615)
[48] Atiya A F and Parlos A G 2000 New results on recurrent network training: unifying the algorithms and accelerating convergence IEEE Trans. Neural Netw. 11 697–709