[jh250021]

High quality modeling of fluid dynamics using reservoir computing

Yoshitaka Saiki (Hitotsubashi University)
Kengo Nakai (Okayama University)

1 Introduction.

Reservoir computing, a brain-inspired machine-learning technique that employs a data-driven dynamical system, is effective in predicting time series and frequency spectra in chaotic behaviors, including fluid flow and global atmospheric dynamics [1, 2, 4, 5, 6].

In our previous work, we inferred macroscopic behaviors of a three-dimensional fluid flow with chaotic behaviors by applying the same method [2, 3]. We show that the reservoir dynamics constructed from only past data of energy functions can infer the future behavior of energy functions and reproduce the energy spectrum. In our procedure of the inference, we assume no prior knowledge of a physical process of a fluid flow except that its behavior is complex but deterministic. It is also shown that we can infer a time-series data from only one measurement by using the delay coordinates. These imply that the obtained two reservoir systems constructed without the knowledge of microscopic data are equivalent to the dynamical systems describing macroscopic behavior of energy functions. It should be remarked that such dynamical systems describing macroscopic behaviors cannot be derived from the Navier–Stokes equation.

2 Reservoir computation.

What's Reservoir computation?

- a relatively high-dimentional fixed neural-network composed of simple nonlinear dynamical systems
- determination of output layer
- For Lorenz system and Kuramoto–Sivashinsky system, inference [1, 5, 6]

2.1 Procedure of training.

○ **1st step (generating a reservoir vector)**

$\mathbf{r}(t+\Delta t) = (1-\alpha)\mathbf{r}(t) + \alpha\tanh(\mathbf{A}\mathbf{r}(t) + \mathbf{W}_{\rm in}\mathbf{u}(t))$.

$\mathbf{A}, \mathbf{W}_{\rm in}$: sparse random matrix, whose maximal eigenvalue is controlled.

$\mathbf{r} \in \mathbb{R}^N$: reservoir vector

N: dimension of reservoir vector

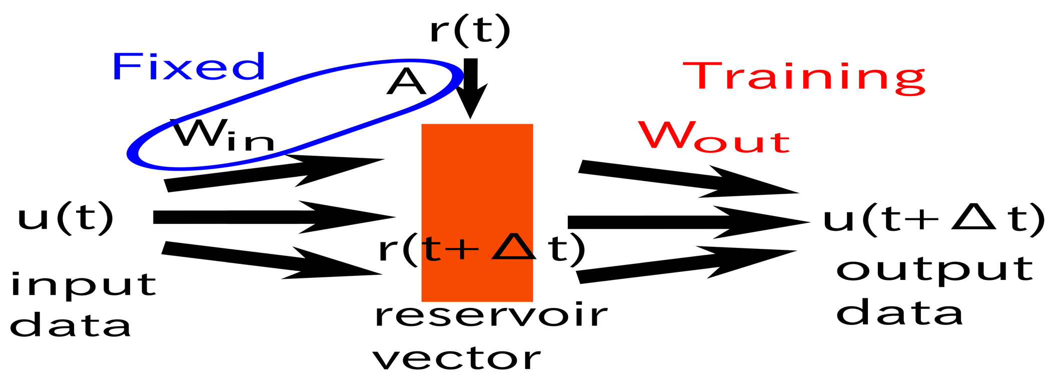


Figure 1: Schematic picture of a reservoir computing (training phase)

○ **2nd step (determination of output layer)**

We determine $\mathbf{W}_{\rm out}$ s.t.

$$^{\forall}t < T \qquad \mathbf{W}_{\rm out}\mathbf{r}(t+\Delta t) \approx \mathbf{s}(t+\Delta t).$$

$\Rightarrow$ Determine it s.t. the form is minimized:

$$\sum_{l=1}^{L} \|(\mathbf{W}_{\rm out}\mathbf{r}(l\Delta t)) - \mathbf{s}(l\Delta t)\|^2 + \beta[Tr(\mathbf{W}_{\rm out}\mathbf{W}_{\rm out}^T)].$$

2.2 Procedure of inference.

Using the $\mathbf{W}^*_{\rm out}$, we infer the time-series $\mathbf{s}$.

$\hat{\mathbf{s}}(t) = \mathbf{W}^*_{\rm out}\mathbf{r}(t)$

$\mathbf{r}(t+\Delta t) = (1-\alpha)\mathbf{r}(t) + \alpha\tanh(\mathbf{A}\mathbf{r}(t) + \mathbf{W}_{\rm in}\hat{\mathbf{s}}(t))$.

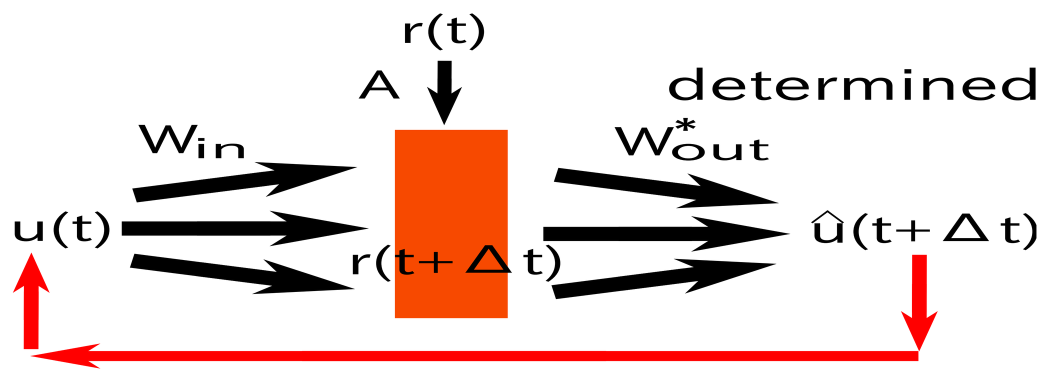


Figure 2: Schematic picture of a reservoir computing (prediction phase)

This reservoir system corresponds to the data-driven model of $\mathbf{u}$.

3 Our previous work

Energy functions:

$$E_0(k,t) := \frac{1}{2}\int_{D_k} |\mathcal{F}_{[v]}(K,t)|^2 dK,$$

where $D_k := \{K \in \mathbb{R}^3 \mid k - 0.5 \le |K| < k + 0.5\}$. To get rid of the high-frequency fluctuation, we take short-time average $E(k,t) = \sum_{s=t-99\Delta t}^{t} E_0(k,s)/100$. Then, $\tilde{E}(k,t)$ is the normalized value of each variable $E(k,t)$.

In the training phase for $t \in (0,T]$, $\mathbf{W}^*_{\rm out}$ is determined by setting

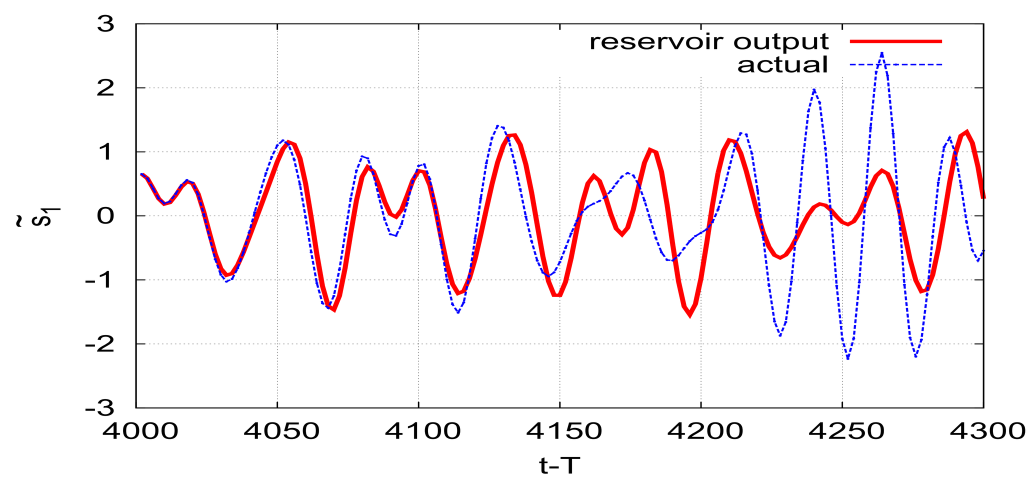
$\mathbf{u}(t) = (\tilde{E}(3,t), \tilde{E}(3,t-\Delta\tau), \cdots, \tilde{E}(3,t-23\Delta\tau))^t$,

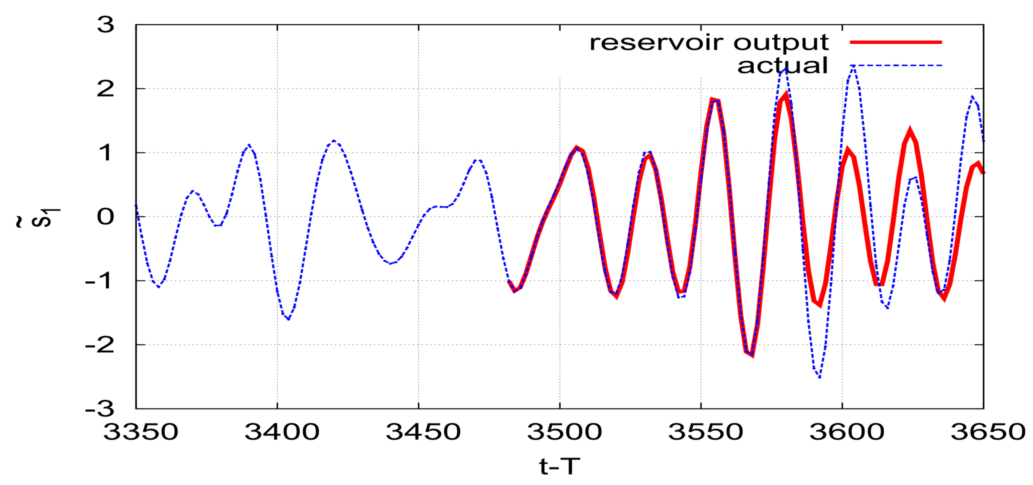
$\mathbf{s}(t) = (\tilde{E}(3,t), \tilde{E}(3,t-\Delta\tau), \cdots, \tilde{E}(3,t-23\Delta\tau))^t$,

where $\Delta\tau = 2.5$. In short, we construct a model inferring $E(3)$ using only $E(3)$ with delay coordinates at the previous step.

○ Inference of time-series of $E(k)$.

When $t-T < 100$, inferred time-series data obtained from our reservoir system (red line) almost coincides with that of reference data obtained from the DNS of NS (blue line).





4 Our project.

We consider the following equation:

$\mathbf{r}(t+\Delta t) = (1-\alpha)\mathbf{r}(t) + \alpha\tanh(\mathbf{A}\mathbf{r}(t) + \mathbf{W}_{\rm in}\mathbf{u}(t) + \xi\mathbf{1})$,

ξ: a scalar constant

$\mathbf{1} = (1,1,,,,,1)^T$.

$\xi\mathbf{1}$ can reduce the dimension N of reservoir computing. We will clarify the least number of N to describe the same dynamics and use it to construct a data-driven model of fluid flow.

We had previously considered the following:

$$\mathbf{W}_{\rm out}\mathbf{r}(t+\Delta t) \approx \mathbf{s}(t+\Delta t).$$

By adding a quadratic term as the second term of the left hand side, the accuracy of the prediction is increased [7].

$$\mathbf{W}_{\rm out}\mathbf{r}(t+\Delta t) + \mathbf{r}(t)^{\rm T}\mathbf{W}^*_{\rm Qout}\mathbf{r}(t) \approx \mathbf{s}(t+\Delta t).$$

We use the term to construct a data-driven model of fluid flow.

Reference

[1] Z. Lu, B. R. Hunt, and E. Ott, Chaos 28, 061104 (2018).

[2] K. Nakai and Y. Saiki, Physical Review E 98, 023111 (2018).

[3] K. Nakai and Y. Saiki, Discrete Contin. Dyn. Syst. S 14, 3641079 (2021).

[4] M. U. Kobayashi, K. Nakai, Y. Saiki and N. Tsutsumi, Physical Review E 104 (4), 044215 (2021).

[5] J. Pathak, Z. Lu, B. Hunt, M. Girvan, and E. Ott, Chaos 27, 121102 (2017).

[6] J. Pathak, B. Hunt, M. Girvan, Z. Lu, and E. Ott, Phys. Rev. Lett. 120, 024102 (2018).

[7] A. Ohkubo and M. Inubushi, Scientific Reports 14, 30918 (2024).