Prediction of flow dynamics using point processes

Yoshito Hirata,[1] Thomas Stemler,[2,3] Deniz Eroglu,[3,4] and Norbert Marwan[3]
[1]*Institute of Industrial Science, The University of Tokyo, 4-6-1 Komaba, Meguro-ku, Tokyo 153-8505, Japan*
[2]*School of Mathematics and Statistics, The University of Western Australia, Crawley, Washington 6009, Australia*
[3]*Potsdam Institute for Climate Impact Research, P.O. Box 60 12 03, 14412 Potsdam, Germany*
[4]*Department of Physics, Humboldt University of Berlin, Newtonstrasse 15, 12489 Berlin, Germany*

(Received 17 November 2017; accepted 18 December 2017; published online 4 January 2018)

Describing a time series parsimoniously is the first step to study the underlying dynamics. For a time-discrete system, a generating partition provides a compact description such that a time series and a symbolic sequence are one-to-one. But, for a time-continuous system, such a compact description does not have a solid basis. Here, we propose to describe a time-continuous time series using a local cross section and the times when the orbit crosses the local cross section. We show that if such a series of crossing times and some past observations are given, we can predict the system's dynamics with fine accuracy. This reconstructability neither depends strongly on the size nor the placement of the local cross section if we have a sufficiently long database. We demonstrate the proposed method using the Lorenz model as well as the actual measurement of wind speed. *Published by AIP Publishing.* https://doi.org/10.1063/1.5016219

Current developments of measurement techniques and hardware enable us to record time-continuous data with very high sampling rates for long times. To understand such data intuitively, we need to describe the data parsimoniously so that such description can reproduce the original time series. Thus, here we propose to represent such time-continuous data only by the series of times when the orbit passes a local cross section. We show that a time series together with some measurements of the past dynamics of the system is sufficient information to predict the future dynamics of the system. There are two applications: an immediate application is to save, or send, time-continuous data using small memory or low channel capacities and therefore make big data more manageable; a more abstract application is to approximate details in a time series when only few observations are possible, based on detailed measurements taken during different times.

Recent advances in measurement technology enable us to record data of time-continuous systems with very high sampling rates. While these advances are welcome, there are several problems in accessing, analysing, and even storing such datasets. These problems are nowadays summarised under the umbrella of *big data* and arise in diverse areas ranging from bioinformatics to systems' health engineering.[1,2] Here, we are presenting a method based on the recurrence of dynamical systems to categorise and rank such datasets. Ranking the recurrence times of our current state against the recurrence times of the big dataset allows us to predict the future dynamics of the system.

Our research is motivated by recent achievements in the time series analysis of non-uniformly sampled data.[3–9] We assume, we have a time-continuous dynamical system, that we are able to measure for a certain amount of time. This observational series of the high dimensional state space is our database. After recording the full dynamics in the phase space for our database, we no longer measure the full dynamics but record the crossing times. These crossing times are the times when the trajectory of the system passes through a local cross section on the attractor of the dynamical system. This dataset is our non-uniform sampled dataset, which we use to estimate the current state of the system. This estimation is done by finding similarities between the current crossing time sequence and the crossing times calculated from the database of the full dynamics.

To detect the similarity, we facilitate the Victor and Purpura distance.[10] The information contained in such crossing times has been studied since Poincaré[11] and is the foundation of recurrence-based time series analysis.[12] The Victor and Purpura distance[10] offers a natural metric to rank and detect similarities between consecutive time windows of the crossing times.[5–10,13] Using cross prediction,[14] we exploit these similarities to estimate the current state and predict the future system's dynamics.

Our paper is organised as follows: After giving a short summary of our method, we first give details on the information contained in the crossing time series and introduce the Victor and Purpora distance. Then, we introduce the cross prediction method and illustrate our method using the low-dimensional Lorenz system.[15] In addition, we apply our method to predict the actual wind speed data.

Predicting the dynamics of the system will be done by facilitating the information in the database. This record of the past dynamics is used in two ways: first, to compare the crossing time series of the database with our current crossing time series; second, we predict the future dynamics by approximating the future state of the system from the past dynamics in the database. The length of the database is critical and determines the quality of our prediction. The recording time has to be chosen long enough to allow first a good

state estimation and then a realistic approximation of the future dynamics. Since our state estimation as well as the cross prediction rely on similarity between past and future dynamics, the longer the record the more reliable the prediction becomes. One of the main results of this paper is that even for realistic short recording times, our method results in quite reliable predictions.

Formally, we assume a dynamical system f_t: $M \to M$, where M is an m-dimensional manifold and f_t is the diffeomorphism representing how a point y in M moves to a point $f_t(y)$ in M after some time t. Our database contains the full dynamics of $y(t)$ for $t = [0, \Delta t, 2\Delta t, \ldots, L]$, where L denotes the recording time and Δt is the sampling time. If the full dynamics cannot be directly measured, one has to reconstruct the dynamics using delay embedding.[16,17] We will discuss this further, when we introduce our environmental example of predicting wind speeds.

From $t > L$, we do not have access to the full dynamics anymore, but measure the times $\{t_i\}$ the trajectory $y(t)$ crosses a local section $\Sigma \subset M$, e.g., $y(t_i) \in \Sigma$. We choose Σ to be perpendicular to the flow at a point of the trajectory and the size of Σ is given by its radius r [cf. Fig. 1(a)]. (In practice, we choose a point in the time series and calculate the direction the point is moving. This direction gives the normal vector for our local cross section Σ.) A lot is known about such crossing time series. For example, it can be shown that a d-dimensional vector $I_i = (t_{i+1} - t_i, t_{i+2} - t_{i+1}, \ldots, t_{i+d} - t_{i+d-1})$ provides a one-to-one embedding of the dynamics under mild conditions relating to periodic points in the phase space[18–20] as long as $d > 2m$. We exploit this result but instead of measuring $d+1$ consecutive crossings, our measuring time w is chosen so that our crossing time series always satisfies the inequality $d > 2m$

$$\max_i \{t_{i+2m+1} - t_i\} < w. \tag{1}$$

We therefore record the crossing time series of our system for w time units. To determine w, we use the time series $y(t)$ in the database and measure the crossing time series $\{t_i\}_{DB}$. This set of times $\{t_i\}_{DB}$ is going to be used to determine w. Therefore, w is the maximum time span needed in the recorded history (the database) to satisfy the condition $d > 2m$. Note that, especially if the recording time L is short, choices of w, as the upper bound of Eq. (1), should be taken conservatively. As we will see, in practice, one can even use a small value for w, e.g., an upper bound for only a section i of the time series, and still get a good prediction. We discuss the techniques used to improve the prediction of such choices of w later.

Given the current crossing time series $\{t_i\}$, we use the Victor and Purpura distance metric[10] to find similar sequences of length w in $\{t_i\}_{DB}$. The closest sequences in $\{t_i\}_{DB}$ provide the base for our state estimation and the prediction of the dynamics for $t > L$. To apply the Victor and Pupura distance, we represent the crossing time series $\{t_i\}$ and $\{t_i\}_{DB}$ as their corresponding spike trains. The value of such spike trains is 0 everywhere and is equal to 1 only when the trajectory $y(t) \in \Sigma$ [cf. Fig. 1(b)]. The Victor and Purpura metric determines the minimum cost to convert one spike train into another. This metric has been used extensively in the context of neuroscience, e.g., Ref. 5, where the spike trains naturally arise as the output of idealised neurons.

Let the current spike train be $U = \{u_\xi | \xi = 1, 2, \ldots, \Xi\}$ and let $V = \{v_\theta | \theta = 1, 2, \ldots, \Theta\}$ represent one of the possible spike trains of the database. u_ξ denotes the time for the ξth event within U and v_θ is the time for the θth event within V. Thus, given two initial conditions $x_u, x_v \in X \subseteq M$ at the beginnings of the two windows, we have $f_{u_\xi}(x_u), f_{v_\theta}(x_v) \in \Sigma$ for each ξ and θ. When we apply the shift of events, we pair up one event u_ξ from U with another v_θ from V. Therefore, each u_ξ and v_θ may not belong to more than one pair. In addition, we define C as the set of such pairs taken from U and V. With this, we can define the Victor Purpura distance as

$$\delta(U, V) = \min_C \left\{ \sum_{(u_\xi, v_\theta) \in C} \lambda |u_\xi - v_\theta| + \Xi + \Theta - 2|C| \right\}. \tag{2}$$

This edit distance δ satisfies the necessary three conditions for a metric: (i) non-negative or, zero if two time windows are identical, (ii) symmetric, and (iii) the triangle inequality.

Intuitively, the metric can be understood by taking into account the possible edits that can transform U into V: either we can align event u_ξ and v_θ or we have to delete/create events in the spike trains. Both these edits appear in the RHS of (2). The cost of aligning the two events is proportional to the time difference between the two events, e.g., $\lambda |u_\xi - v_\theta|$ controls the cost. Since the cost of creating and deleting the events is chosen to be equal to 1, δ depends on the difference between the total number of events—given by $\Xi + \Theta$—and twice the number of possible pairs $-2|C|$. This Victor

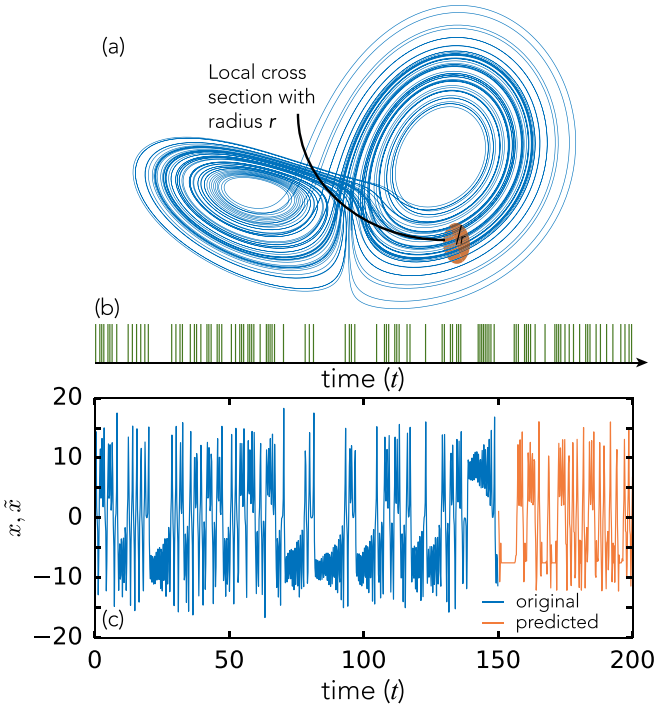

FIG. 1. A schematic overview of our prediction method. (a) Local section Σ with radius r, oriented perpendicular to the flow; (b) spike train having a non-zero amplitude every time the trajectory passes through Σ; (c) x-component of the Lorenz system (blue) and $\tilde{x}$ our prediction (green). The blue part of the trajectory represents the database used for the cross prediction.

Purpura distance has been used to evaluate synchronization among neurons.[21,22] Further details and a thorough mathematical description of the Victor Purpura distance can be found in the literature.[8,23,24]

Comparing the current crossing time sequence in this way with all the crossing time sequences in the database, we identify the 10 sequences that are closest in terms of their distance δ. The average of their end points is going to be the state estimate at the current time. For extremely long database recording times and very high data precision one might use just the end point of the sequence with minimal δ as the current state estimate. Under realistic conditions, however, using the average and standard deviation of about 10 sequences with similar recurrence properties—in the Victor Purpura distance sense—improves the stability of the prediction and helps us to evaluate its reliability. One can understand the Victor-Purpura distance for a spike train as being similar to the Euclidean distance for usual vector spaces.

Using cross prediction has the advantage that we do not require access to the equations governing the dynamics of $y(t)$. Instead, the prediction of the future dynamics is provided as the average of the recorded evolution of the 10 states defining our state estimate. We define t_0 as the current point in time and use $h = 1, 2, \ldots, 10$ as the index of the 10 states. Consequently, $t_{0,h}$ denotes the end points of the 10 most similar sequences in the database. Then, the estimation of the current state is

$$\tilde{y}(t_0|t_0) = \frac{1}{10}\sum_{h=1}^{10} y(t_{0,h}). \tag{3}$$

Similarly, the prediction of $\tilde{y}$ for $n\Delta t$ time steps in the future is

$$\tilde{y}(t_0 + n\Delta t|t_0) = \frac{1}{10}\sum_{h=1}^{10} y(t_{0,h} + n\Delta t), \tag{4}$$

and its corresponding prediction uncertainty can be measured by the variations of the 10 sequences

$$\sigma_{\tilde{y}}(t_0 + n\Delta t|t_0) = \sqrt{\frac{1}{10}\sum_{h=1}^{10}(y(t_{0,h} + n\Delta t) - \tilde{y}(t_0 + n\Delta t|t_0))^2}.$$

We use the uncertainty of our prediction to accept or reject the prediction of $t_0 + n\Delta t$. The main issue we will have is that the ensemble of the 10 sequences with the lowest δ might lose coherence, because they might be close in the Victor-Purpura sense, but not all of them have to be close to the system's state on Σ. When this happens the uncertainty of the prediction grows: $\sigma_{\tilde{y}}(t_0 + n\Delta t|t_0) > \sigma_{\tilde{y}}(t_0 + n\Delta t|t_0 + n\Delta t)$. In such a case, we would reject the prediction and instead let $\tilde{y}(t_0 + n\Delta t|t_0 + n\Delta t)$ be our prediction and replace t_0 by $t_0 + n\Delta t$ to reinitialise for further prediction.

Especially when L is short, it can be necessary to work with a small value of w. For such w the inequality (1) will not be satisfied for all times, but only for some part of the time series in the database. To still achieve a good state estimation, we consider that just before or after the current time window, there is or is not a spike. For our current spike train, we therefore have the option that a spike does (not) appear just before (after) the beginning (end) of the window. We combine these 4 options with the corresponding 4 options of our database $\{t_i\}_{DB}$ and determine the δ of the 16 possibilities. As we see later, including these 16 possibilities in our δ minimisation greatly improves the accuracy of the prediction.

Such a prediction can be seen as the orange line in Fig. 1(c). Clearly visible are the time periods during which we rejected the prediction (see the instances of constant amplitude). These coincide with periods of low or no recurrence that are large gaps in the train sequence in panel (b). For practical applications, we would use a larger value of w to eliminate these areas of prediction rejection. To qualitatively assess the prediction, we evaluate the cross-correlation coefficient $R(y, \tilde{y})$ between the prediction and the true dynamics of the system. For each t_0, we are going to predict the dynamics using Eq. (4) and evaluate $R(y, \tilde{y})$.

As a first application of this method, we apply our algorithm to the Lorenz 1963 system,[15] using the standard parameters $\sigma = 28$, $\rho = 10$, and $b = 8/3$. We are going to systematically vary the radius r of the local section Σ and the database recording time L. These parameter changes directly impact the upper bound w of Eq. (1). We calculated statistics Cv and Lv from these spike trains based on Ref. 25, which evaluate the global and local variabilities for spike trains for judging whether the spike trains follow a Poisson process or not. If a spike train follows a Poisson process, its Cv and Lv fluctuate around 1.[25] We found that Cv and Lv were 0.8603 ± 0.1904 and 0.6276 ± 0.1790, respectively, over 10 samples. While the estimate for Cv was not significantly different from 1, that for Lv was significantly different from 1 ($P \approx 0.019$), meaning that the spike trains are likely to be different from a Poisson process, or a standard point process.[25]

In Fig. 2, we see two outputs of our algorithm in comparison with the true x-component of the Lorenz system. Since it is hard to visualise the state estimates together with their predictions, we instead only show the state estimates together with the prediction up to the next estimate. As a reinitialisation time we used $\Delta t = 0.2$ in both experiments. Since this value is very small, the prediction is seldom rejected, and the data shown highlights the accuracy of our state estimate more than the quality of our prediction. The main difference between the two datasets shown in panel (a) and (b) is the recording time L and the radius r of the local section Σ. We use $r = 10$ in panel (a) and $r = 3$ in (b). Consequently, the recurrence rate with which the trajectory returns to Σ is much higher in (a) than in (b) as we can see from the spike trains below the trajectory. Given the high recurrence rate we need a short recording time $L = 150$ for $r = 10$, while for $r = 3$ we need longer recording times ($L = 4950$). Similarly, the recurrence rate helps us to choose the upper bound w of Eq. (1): (a) $w = 2$ and (b) $w = 10$. The comparison of the true and predicted dynamics clearly shows that our algorithm can be optimised to work for both choices of r and approximates the dynamics of the system well.

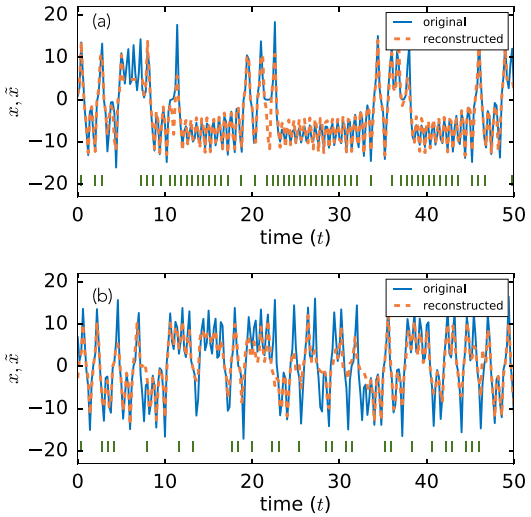

FIG. 2. Comparison between the predicted $\tilde{x}$ (dashed line) and the original x (solid line) time series of the Lorenz system: (a) recording time of the database $L = 150$, radius of local section $r = 10$, upper bound of Eq. (1) $w = 2$ and $\Delta t = 0.2$; (b) $L = 4950$, $r = 3$, $w = 10$ and $\Delta t = 0.2$; at the bottom of both panels the corresponding spike trains are shown and $\tilde{x}(t_0)$ is given by Eqs. (3) and (4).

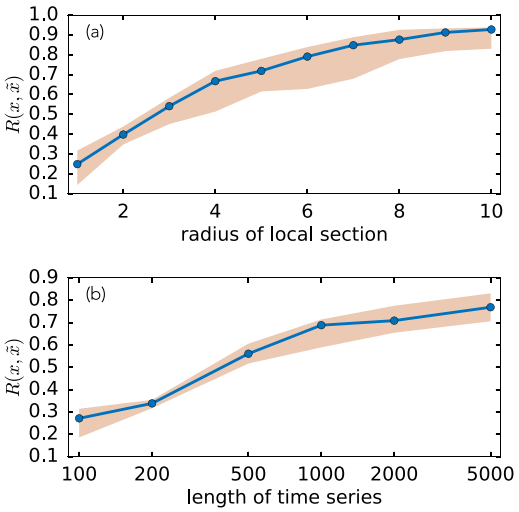

FIG. 3. The cross-correlation coefficient $R(x, \tilde{x})$ between the prediction and the true dynamics in dependence of the size of the partition and recording length. Cross-correlation was calculated for a prediction of 50 time units. Shown is the median correlation for 10 randomly placed Σs as well as the 50% confidence interval. (a) fixed upper bound $w = 2$ and $L = 150$ while varying the size of the portion and (b) $w = 10$ and a fixed radius $r = 3$ of the partition while varying the recording length L. N.B. that we tend to have positive correlation coefficients because we predicted the values generated from the true dynamics based on spike trains.

The large deviations of $\tilde{x}$ in Fig. 2(a) around $t = 12$ and 22 are caused by our choice of $w = 2$ and a very short recording time $L = 150$. Given $w = 2$, it is not always possible to satisfy the condition (1). For that reason, our prediction diverges from the true dynamics. In addition, the short recording of the past dynamics in combination with the cross-prediction technique does not allow us to get the amplitude of some of the oscillations right, e.g., $25 < t < 30$. As we can see for $L = 4950$ and $w = 10$, these problems disappear, and we get better estimates and predictions [Fig. 2(b)]. While the recurrence rate is lower, we have more knowledge about the past dynamics and the higher value of w makes it easier to satisfy the condition (1).

We want to understand the influence of L and r on our prediction algorithm in more detail. For this we determine the correlation between the original $x(t)$ time series of Lorenz with predictions of the next 50 time units, while varying L or r. Moreover, we use these experiments to demonstrate that the location of the local section Σ is of minor importance for the predictions. For each of our experiments, we therefore use 10 random positions for Σ and report in Fig. 3 the median value as well as the 50% confidence interval. In Fig. 3(a), we fixed $w = 2$, $L = 150$ and vary the size of the partition, while in (b) we chose $w = 10$, $r = 3$ and vary L.

A sufficiently large size of Σ enables us to cross-predict the original time series almost perfectly [Fig. 3(a)], because for large r we have a high recurrence rate and the condition of Eq. (1) is likely to be met. When we reduce the size of the local cross section, the correlation coefficient between the original time series and the cross-predicted time series decreases. The lower recurrence rate makes it impossible to satisfy the condition Eq. (1), namely $d > 2m$, for all times, leading to the worse prediction. But, even in such a case, a sufficiently large recording time L as well as a larger w leads to a high correlation coefficient [Fig. 3(b)]. Thus, we expect for a database of a sufficiently long recording time, which we can still reconstruct the original time series faithfully. Moreover, the medians and 50% confidence intervals in Figs. 3(a) and 3(b) show that the positions of the local cross sections are of less importance for our prediction algorithm. For example, when we prepared the local cross section at $x = 0$ with $\dot{x} > 0$, we still can reconstruct some meaningful signal as shown in Fig. 4.

To further understand these results in Figs. 2 and 3 and gain some insight about the importance of the different measurements (recorded history in the database and crossing times), we analysed the predictability of flows from crossing

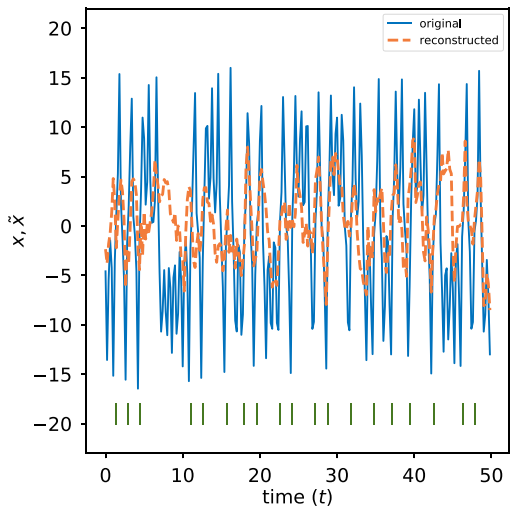

FIG. 4. A spike train generated by the local cross section at $x = 0$ and $dx/dt > 0$, and the reconstruction of the original signal. The original signal is shown in the blue solid line and its reconstruction, the red dashed line. Here we used $L = 19950$, $\Delta t = 0.2$ and $w = 30$. At the bottom we show the spike train for the local cross section. The correlation coefficient between the original and reconstructed signals is 0.3926. (Because we have a fewer events, we needed the larger L and w to have the better results.)

times as an information-theoretic problem. Our analysis shows that when a local cross section becomes small, and therefore generated less frequent events, the time resolution of our data becomes the dominating factor. For our analysis, we partition our spike train time series into α bins of equal size. Let β be the number of events within the spike train. Moreover, each bin can only contain one spike. Then, the number of signals N we can send by a spike train is

$$N = \binom{\alpha}{\beta} = \frac{\alpha!}{\beta!(\alpha-\beta)!}. \tag{5}$$

Assuming that all signals are equally likely, we can calculate the amount of information in the series as[26]

$$\log_2 N = \log_2 \frac{\alpha!}{\beta!(\alpha-\beta)!}. \tag{6}$$

Applying Stirling's approximation, we get

$$\begin{aligned}\log_2 N &\approx \alpha\left\{-\frac{\beta}{\alpha}\log_2\frac{\beta}{\alpha} - \frac{\alpha-\beta}{\alpha}\log_2\frac{\alpha-\beta}{\alpha}\right\}\\ &= \beta\log_2\frac{\alpha}{\beta} + (\alpha-\beta)\log_2\frac{\alpha}{\alpha-\beta}\\ &\geq \beta\log_2\frac{\alpha}{\beta}.\end{aligned} \tag{7}$$

Given that $\alpha \gg \beta$, we can assume that α and β are related by $\alpha = \beta 2^{\gamma}$. Therefore, the RHS of the equation can be written as

$$\beta\log_2\frac{\alpha}{\beta} = \beta\log_2 2^{\gamma} = \beta\gamma. \tag{8}$$

Since β gives the total number of spikes in the sequence, and we only allow one spike per partition, the maximum β is directly related to the time resolution of the sequence. Therefore, we conclude that the time resolution given by β is directly proportional to the total amount of information contained in the time series. This has two consequences. For the real world data, the time resolution of the data is of upmost importance. Using our algorithm in the real world applications would require instruments with high precision, but at least it is only the time precision and not also the amplitude precision on top. The second consequence of Eq. (8) relates to the efficiency of our method. Once the database is recorded, the data required to predict the future dynamics has a very small footprint. For example, the original dataset shown in Fig. 2(b) has 40 000 bytes and its crossing time series $\{t_i\}_{DB}$ has 152 ± 40 bytes without compression in disk space (We used here the MATLAB command "whos" to evaluate the disk space we need for storing these variables.). The spike train recorded to estimate the state and consequently predict the dynamics was $0.38 \pm 0.10\%$ of its original size. Having such a small memory demand makes this method attractive for the real world applications.

As one example for a real world application, we are use our algorithm to predict wind speed.[27] The original measurements were observed for 24 h from around 2 pm on 1 September 2005 at about 1 m above the ground level at the Institute of Industrial Science, the University of Tokyo, Tokyo, Japan using an ultrasonic anemometer. The original measurements were made with 50 Hz, but we first sub-sample the time series using every tenth point. This sub-sampling allows us to use a longer L, and therefore the database contains more of the long term changes in the dynamics. Since we do not have access to the full state space of this environmental system, we have to reconstruct the attractor of the dynamics by using delay embedding.[16,17] Given that the system seems to have the features of high-dimensional dynamics,[28] we have to use a high dimensional embedding. But the total recording time of the wind speeds is 10 000 s, and consequently we cannot use a very high embedding dimension, without making the recurrence rate too low for our prediction algorithm. In addition, we only use the first 9000 s as our database and use the remainder for comparison with our prediction. As a compromise and after several tests, we decided to use a 10 dimensional embedding space. We find that our algorithm succeeded in capturing the large scale features of the dynamics, i.e., the prolonged changes of the wind velocity, is similar in the experimental data and the prediction (Fig. 5). On the other hand, our predictions fail to reproduce the finer details of the system's dynamics. We conjecture that a higher embedding dimension together with a longer time series would be able to overcome these limitations. Another way to improve the performance of the method could be to use marked spike trains, by assigning some additional value to each event. Several distances have been proposed for such marked point processes,[23,29–31] which might increase the prediction performance for such high dimensional dynamics.

Applying the method of Ref. 32 to the aforementioned wind data shows the value greater than 99% point of 2.36, meaning that the wind data should be regarded as non-stationary. The applicability of our method for non-stationary data should be evaluated more closely in future research, although our results in Fig. 5 seem to show some promise in this line of research.

Our numerical results and to some extent the results on wind speeds show that a local cross section Σ has the generic property to reconstruct the underlying dynamics of a flow almost perfectly. This is similar to the generating partitions, which have been extensively studied in maps.[33–38] It is a non-trivial task to find a generating partition in maps. Our results show that in flows this is much

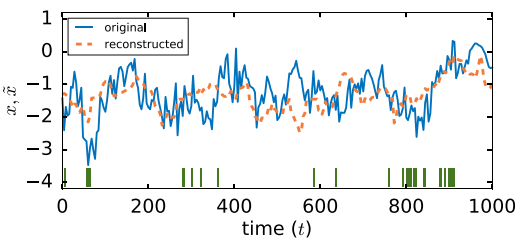

FIG. 5. Example of the wind speed time series (solid line) together with our prediction (dash-dotted line). The parameters of our algorithms are the size of the time window $w = 200$ s, recording time $L = 9000$ s, $\Delta t = 4$ s and the size of the local section $\Sigma = 5$ m/s. Our algorithm predicts the following 1000 s.

easier and the position of Σ does not matter much for the prediction quality (cf. Fig. 3).

While there are many studies that try to predict the next crossing time based on previous crossings (for example, Ref. 18), the method presented here is different. Instead of just focussing on the next crossing, we are able to estimate the complete future dynamics using cross prediction. If we regard the observations at a local cross section as a coincidence detector,[39,40] our results could explain how we can share the same experience even if we perceive a phenomenon in different ways, or in our case by different local cross sections. Hence, our results might also have some implications in the field of theoretical neuroscience.

We presented a method for the state estimation and prediction of flows, given a database of the past dynamics and a crossing time series. Identifying the current state is done by ranking the crossing time sequences of the database according to their distance from the current crossing time series. We showed that our method requires little memory to store and send time series information via a spike train if the database is shared between the sender and the receiver. Our method does not require us to store and send an entire time series under this assumption. Instead, we simply record or send the times when a trajectory passes the local cross section. This advantage can be important for sensor networks,[41] where each device has to store and communicate lots of environmental information under severe energy constraints. Similarly, our method may be used to reconstruct the missing data within observations, such as gaps in, e.g., satellite images partly covered by clouds,[42] historical series of sunspot numbers,[43] or historical phenological data such as the start of the cherry blossom in Japan.[44]

Y.H. would like to appreciate Professor Kazuyuki Aihara for the discussions. The research of Y.H. was supported by Kozo Keikaku Engineering, Inc. The research of D.E. has been supported by the German-Israeli Foundation for Scientific Research and Development (GIF), GIF Grant No. I-1298-415.13/2015.

[1]C. S. Greene, J. Tan, M. Ung, J. H. Moore, and C. Cheng, J. Cell. Physiol. **229**, 1896–1900 (2014).
[2]W. Q. Meeker and Y. Hong, Qual. Eng. **26**, 102–116 (2014).
[3]M. McCullough, K. Sakellariou, T. Stemler, and M. Small, Chaos **26**, 123103 (2016).
[4]K. Sakellariou, M. McCullough, T. Stemler, and M. Small, Chaos **26**, 123104 (2016).
[5]Y. Hirata and K. Aihara, J. Neurosci. Methods **183**, 277–286 (2009).
[6]Y. Hirata and K. Aihara, Physica A **391**, 760–766 (2012).
[7]Y. Hirata, E. J. Lang, and K. Aihara, "Recurrence plots and the analysis of multiple spike trains," in *Springer Handbook of Bio-/Neuroinformatics*, edited by N. Kasabov (Springer, Berlin, Heidelberg, 2014), pp. 735–744.
[8]I. Ozken, D. Eroglu, T. Stemler, N. Marwan, G. B. Bagci, and J. Kurths, Phys. Rev. E **91**, 062911 (2015).
[9]D. Eroglu, F. H. McRobie, I. Ozken, T. Stemler, K.-H. Wyrwoll, S. F. M. Breitenbach, N. Marwan, and J. Kurths, Nat. Commun. **7**, 12929 (2016).
[10]J. D. Victor and K. P. Purpura, Network: Comput. Neural Syst. **8**, 127–164 (1997).
[11]H. Poincaré, "Sur le problème des trois corps et les équations de la dynamique," Acta Math. **13**, 1–270 (1890).
[12]N. Marwan, M. C. Romano, M. Thiel, and J. Kurths, "Recurrence plots for the analysis of complex systems," Phys. Rep. **438**(5–6), 237–329 (2007).
[13]Y. Hirata, K. Iwayama, and K. Aihara, Phys. Rev. E **94**, 042217 (2016).
[14]M. L. Van Quyen, J. Martinerie, C. Adam, and F. J. Varela, Physica D **127**, 250–266 (1999).
[15]E. N. Lorenz, J. Atmos. Sci. **20**, 130–141 (1963).
[16]F. Takens, Lect. Notes Math. **898**, 366 (1981).
[17]T. Sauer, J. A. Yorke, and M. Casdagli, J. Stat. Phys. **65**, 579 (1991).
[18]T. Sauer, Phys. Rev. Lett. **72**, 3811–3814 (1994).
[19]J. P. Huke and D. S. Broomhead, Nonlinearity **20**, 2205–2244 (2007).
[20]S. Smale, "Stable manifolds for differential equations and diffeomorphisms," in *Topologia Differenziale* (Springer, 2011), pp. 93–126.
[21]K. MacLeod, A. Bäcker, and G. Laurent, Nature **395**, 693–698 (1998).
[22]R. Nomura, Y. Liang, and T. Okada, PLoS One **10**, e0140774 (2015).
[23]S. Suzuki, Y. Hirata, and K. Aihara, Int. J. Bifurcation Chaos **20**, 3699–3708 (2010).
[24]Y. Hirata and K. Aihara, Chaos **25**, 123117 (2015).
[25]S. Shinomoto, K. Shima, and J. Tanji, Neural Comput. **15**, 2823–2842 (2003).
[26]T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. (John Wiley & Sons, Hoboken, 2006).
[27]Y. Hirata, D. P. Mandic, H. Suzuki, and K. Aihara, Philos. Trans. R. Soc. A **366**, 591–607 (2008).
[28]Y. Hirata, T. Takeuchi, S. Horai, H. Suzuki, and K. Aihara, Sci. Rep. **5**, 15736 (2015).
[29]F. P. Schoenberg and K. E. Tranbarger, Environnmentrics **19**, 271–286 (2008).
[30]H. Hino, K. Takano, and N. Murata, R J. **7**, 237–248 (2015).
[31]K. Iwamaya, Y. Hirata, and K. Aihara, Phys. Lett. A **381**, 257–262 (2017).
[32]M. B. Kennel, Phys. Rev. E **56**, 316–321 (1997).
[33]P. Grassberger and H. Kantz, Phys. Lett. A **113**, 235 (1985).
[34]R. L. Davidchack, Y.-C. Lai, E. M. Bollt, and M. Dhamala, Phys. Rev. E **61**, 1353 (2000).
[35]J. Plumecoq and M. Lefranc, Physica D **144**, 231 (2000).
[36]J. Plumecoq and M. Lefranc, Physica D **144**, 259 (2000).
[37]M. B. Kennel and M. Buhl, Phys. Rev. Lett. **91**, 084102 (2003).
[38]Y. Hirata, K. Judd, and D. Kilminster, Phys. Rev. E **70**, 016215 (2004).
[39]P. König, A. K. Engel, and W. Singer, Trends Neurosci. **19**, 130–137 (1996).
[40]R. Azouz and C. M. Gray, Proc. Natl. Acad. Sci. U.S.A. **97**, 8110–8115 (2000).
[41]I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, IEEE Commun. Mag. **40**, 102 (2002).
[42]R. Liu, R. Shang, Y. Liu, and X. Lu, Remote Sens. Environ. **189**, 164–179 (2017).
[43]F. Clette, L. Svalgaard, J. M. Vaquero, and E. W. Cliver, Space Sci. Rev. **186**, 35–103 (2014).
[44]Y. Aono and K. Kazui, Int. J. Climatol. **28**, 905–914 (2008).