UNSTABLE PERIODIC ORBIT AND SYMBOLIC DYNAMICS

Seungchan Lee

Center of Advanced Vehicular Systems
Mississippi State University
slee @cavs.msstate.edu

ABSTRACT

This paper presents two novel methods to interpret nonlinear time series, interval analysis and symbolic dynamics. The former one is related to find unstable periodic orbits (UPOs), and the letter one is related for modeling nonlinear time-series. From these two concepts, we can find a possibility for modeling chaotic system in terms of symbolic dynamics. The UPOs have not been much used for modeling, but their use still restricted to control systems. Although the second one is still opening problem for modeling deterministic chaotic system, its powerful efficiency enables us to transform complex time-series into symbolic sequence. This symbolic sequence constitutes symbol space like embedding space. Although these two concepts, UPOs and symbolic dynamics, are not much related to each other, it can be suitable tools for modeling real world signals. We present the method to find UPOs from phase space, and briefly review the procedure for modeling chaotic time-series using symbolic dynamics. And then, we combine those two concepts and make a new technique for modeling nonlinear time-series. Finally, we discuss the drawbacks and possibilities of these techniques.

1. INTRODUCTION

Unstable periodic orbits (UPOs) are a well known characteristic of chaotic time series data. It constitutes fundamental building block of reconstructed phase space, and gives a symbolic meaning on it [1]. The detection of periodic orbits in chaotic systems has been continuously issued in many fields. A particular important application is in the control chaotic systems where the first essential step is often the determination of periodic orbits. One interesting application of UPOs can be found [2], where one can use symbolic meaning of UPOs to generating partition in symbol space. In practice, finding the unstable periodic orbits in the chaotic data is a difficult task. It usually requires numerical studies but there is no guarantee that there exists a true periodic trajectory that stays near a computer generated one. Here, we present one method to detect and locate UPOs [1], and further investigate their applications [2] [3].

Mostly known method for reconstructing complex time-series is embedding space, which uses time-delay of measured time series and embedding dimension. Another different way to be issued recently is symbolic space, which views the dynamics as the transition between a finite set of symbols [2][3][4].

Symbolic description of deterministic chaotic system originated from mathematical discipline of symbolic dynamics. The first attempt was found in Hadamard's work [5], where he introduced a symbolic description of sequences in geodesic flows on surfaces of negative curvature. This work further has been extended by many researchers. Another point of view was started by Poincare, who proposed that the complex time evolution of chaotic system could be depicted using a kind of strobocoscopic sampling of the multi-dimensional phase space trajectory [4]. Poincare surface captures temporal evolution of time-series in phase-space and successive intersection between these surfaces reduces the dimensionality of the problem.

Topological considerations in the continuous space turn into grammatical considerations in the symbolic space, concerning the presence or absence of legal transitions and various "words" occurring in the symbolic sequence. So, it is natural to relate Unstable periodic orbits (UPOs) embedded in chaotic attractor to the symbolic dynamics. Recently, the link between UPOs and symbolic dynamics is proposed by Davidchack [2], where he tries to estimate a generating partition based on UPOs of the system. M.B. Kennel [3] further investigates and proposes an improved method for that purpose. An interesting attempt is shown in [9], where topological voiceprints are used for speaker verification.

The rest of this paper is organized as follows. Section 2 provides a background of interval method and finding UPOs from that. Section 3 briefly introduces symbolic dynamics and presents two novel methods for symbol sequence partitioning. Section 4 discusses limitations and possibilities of those methods. Finally, section 5 gives a conclusion.

2. UNSTABLE PERIODIC ORBITS

In this section, we present the theory of interval methods for finding unstable periodic orbits (UPOs).

2.1 Principles of Interval analysis

Interval analysis is a new and growing branch of applied mathematics [6][7]. It is an approach to treat an interval as a new kind of number. Computations in properly rounded interval arithmetic produce results which contain both ordinary machine arithmetic results and also infinite precision arithmetic results. Here we present a very short introduction to the interval analysis. In this paper, we denote intervals by boldface letters. Furthermore, if $\mathbf{x}$ is an interval, we denote its endpoint by $\underline{\mathbf{x}}$ and $\overline{\mathbf{x}}$. Thus, $\mathbf{x} = [\underline{\mathbf{x}}, \overline{\mathbf{x}}]$. By an n-dimensional interval vector, we mean an ordered n-tuple of intervals $\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$. Thus, if $\mathbf{x}$ is a two-dimensional vector, then $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2)$ for some intervals $\mathbf{x}_1 = [\underline{\mathbf{x}}_1, \overline{\mathbf{x}}_1]$ and $\mathbf{x}_2 = [\mathbf{x}_2, \mathbf{x}_2]$. A two-dimensional interval vector also represent a two-dimensional rectangle of points (x_1, x_2) such that $\underline{\mathbf{x}}_1 \le x_1 \le \overline{\mathbf{x}}_1$ and $\underline{\mathbf{x}}_2 \le x_2 \le \overline{\mathbf{x}}_2$

Next, we define basic arithmetic operations on the set of intervals. Given two intervals $\mathbf{x} = [\underline{\mathbf{x}}, \overline{\mathbf{x}}], \mathbf{z} = [\underline{\mathbf{z}}, \overline{\mathbf{z}}]$, the four operations, addition, subtraction, multiplication, and division, are defined as the endpoint operation

$$\begin{aligned}
&\mathbf{x} + \mathbf{z} = [\underline{\mathbf{x}} + \underline{\mathbf{z}}, \overline{\mathbf{x}} + \overline{\mathbf{z}}] \\
&\mathbf{x} - \mathbf{z} = [\underline{\mathbf{x}} - \underline{\mathbf{z}}, \overline{\mathbf{x}} - \overline{\mathbf{z}}] \\
&\mathbf{x} \times \mathbf{z} = [\min(\underline{\mathbf{x}}\underline{\mathbf{z}}, \underline{\mathbf{x}}\overline{\mathbf{z}}, \overline{\mathbf{x}}\underline{\mathbf{z}}, \overline{\mathbf{x}}\overline{\mathbf{z}}) \max(\underline{\mathbf{x}}\underline{\mathbf{z}}, \underline{\mathbf{x}}\overline{\mathbf{z}}, \overline{\mathbf{x}}\underline{\mathbf{z}}, \overline{\mathbf{x}}\overline{\mathbf{z}})] \\
&1/\mathbf{x} = [1/\underline{\mathbf{x}}, 1/\overline{\mathbf{x}}]
\end{aligned} \tag{1}$$

where the condition of the division is the same in real arithmetic. It does not divide by "0".

Nearly all numerical computation is carried out with "fixed-precision", approximate arithmetic. The arithmetic hardware of computers is designed to carry out approximate arithmetic in "fixed-precision". It is possible to program a computer to carry out the operations of interval arithmetic with appropriate rounding, when necessary, of left and right computed endpoints, so that the machine computed interval result always contains the exact interval result. In some cases, this requires adding (or subtracting) a "low order bit" to (or from) the right (or left) hand endpoint of a machine computed interval result. This can be done in such a way that the machine computed interval result not only contains the exact interval result, but the machine computed right endpoint is the smallest machine number not less than the correct right endpoint and the machine computed left endpoint is the largest machine number not greater than the correct left endpoint. This is called "best possible" rounded interval arithmetic.

2.2 Interval Newton operator, Krawczyk operator

Let us consider a function $\mathbf{x} \to f(\mathbf{x})$, $\mathbf{x}, f(\mathbf{x}) \in R^m$. In order to investigate the existence of zeros of f in an m-dimensional interval vector $\mathbf{x}$ one evaluates the interval Newton operator

$$N(\mathbf{x}) = x_0 - (f'(\mathbf{x}))^{-1} f(x_0), \tag{2}$$

where $f'(\mathbf{x})$ is the interval matrix containing all Jacobian matrices of the form $f'(\mathbf{x})$ for $x \in \mathbf{x}$ and x_0 is an arbitrary initial point belonging to the interval vector $\mathbf{x}$.

The above equation simply modified the following.

$$f'(\mathbf{x}^{(k)})(N(\mathbf{x}^{(k)})) - x^{(k)}) = -f(x^{(k)}), \tag{3}$$

where, $\mathbf{x}^{(k)}$ is the current interval, and $x^{(k)}$ is a point in the interior of $\mathbf{x}^{(k)}$, usually taken to be the midpoint. Once the initial value is set, Newton method iteratively calculate the solution. And we need not to compute the inverse of $f'(\mathbf{x})$ in order to evaluate $N(\mathbf{x})$. The Interval Newton operator has two important properties [8] which can be used to prove the existence and uniqueness of zeros of f.

Theorem 1 :

1. If $N(\mathbf{x}) \subset \mathrm{int}(\mathbf{x})$ then $f(\mathbf{x}) = 0$ has a unique solution in $\mathbf{x}$.
2. If $N(\mathbf{x}) \cap \mathbf{x} = \phi$ then there are no zeros of f in $\mathbf{x}$.

The interval Newton operator can be used only when the interval matrix $f'(\mathbf{x})$ is regular, i.e. composed of nonsingular matrices. The following operator can be used for a wider class of systems.

The Krawczyk operator is defined as

$$K(\mathbf{x}) = x_0 - Cf(x_0) - (Cf'(\mathbf{x}) - I)(\mathbf{x} - x_0), \tag{4}$$

where x_0 is an arbitrary point belonging to $\mathbf{x}$ (usually one uses the midpoint of $\mathbf{x}$) and C is a preconditioning matrix. It is usually chosen as the inverse of $f'(x_0)$.

2.3 Finding Unstable periodic orbits (UPO's)

The Krawczyk operator can be used for proving the existence of period-n cycles of f by applying it to the map $F = (R^m)^n \mapsto (R^m)^n$ defined by

$$[F(z)]_k = x_{(k+1) \bmod n} - f(x_k) \tag{5}$$

for $k = 0, ..., n-1$ where $z = (x_0, ..., x_{n-1})$ [1]. The zeros of F correspond to fixed points of f^n. Using a higher dimensional map F allows us to deal with longer periodic orbits.

In order to find all period-n cycles of f in the region Ω, we use the combination of the generalized bisection [15] and Krawczyk method described above. At the beginning the set Ω is covered by boxes (m-dimensional interval vectors). For each interval vector $\mathbf{x}$, we produce the sequence $(\mathbf{x}_i)_{i=0}^{n-1}$,

where $\mathbf{x}_i = f^i(\mathbf{x})$, set $\mathbf{z} = (\mathbf{x}_0,...,\mathbf{x}_{n-1})$, and then the interval operator $K(\mathbf{z})$ is evaluated. Finally, we use theorem 1 to prove that there is exactly one fixed point of f in $\mathbf{x}$ (if the assumption of the first part holds) or that there are no fixed points of f^n in $\mathbf{x}$ (if the assumption of the second part holds). If neither of these two assumptions is fulfilled, the interval vector $\mathbf{x}$ is divided into smaller parts and the computations are repeated. Below is this algorithm using simple model language [1].

```
procedure  FindPeriodicOrbitsInBox(x)
   x_0 ← x;
   for i ∈ {1,....., N -1}  do begin
         x ← f(x_{i-1});
   end
   z ← (x_0, x_1,..., x_{n-1}),
   compute  K(z);
   if K(z) ⊂ z then begin
      Q ← Q + 1;
      record x;
      return;
   end
   if  K(z) ∩ z = φ  then begin
       return;
   end
   divide  x into y_1,....., y_{2m};
   for i ∈ {1,..., 2}  do begin
        FindPeriodicOrbitsInBox(y_i);
   end
end of FindPeriodicOrbitsInBox
```

Figure 1 : Procedure for finding periodic orbits

Q is a global variable, which at the beginning of the computations is initialized to be zero, and at the end is equal to the number of fixed points of f^n in the region considered. In a typical implementation of generalized bisection for finding all zeros of the map F defined by equation (5), the division is performed on the box $\mathbf{z}$. This means that in order to find all periodic orbits of an m-dimensional map, we are searching the mn-dimensional space.

Once the box enclosing the periodic point $\bar{\mathbf{x}}$ is found, we can find a very narrow enclosure of its position by iterating the Krawczyk operator $(\bar{x} \in K(\mathbf{x}) \subset \mathbf{x})$. We can also find an enclosure of the Jacobian matrix of f^n at the periodic point and decide the stability of the orbit.

3. SYMBOLIC DYNAMICS

In this section, we present a guideline using symbolic dynamics for modeling chaotic time series. We use symbolization as the process to convert time-series data into symbolic space. Also, there are some similarities between symbolic analysis and time-delay embedding. Whenever the points come out, we mention it. In the subsequent section, we will link UPOs and symbolic dynamics for modeling chaotic time series.

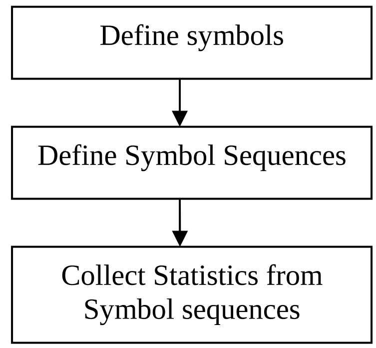

Figure 2: Illustrating the flow of symbolization.

3.1 Procedure for mapping to symbolic space

The procedure for symbolization starts with the following questions. What information in measurement data is significant for our purpose, and how can we efficiently convert them into appropriate symbols? The answer to these questions has been suggested by many researchers. Usually, the first step of symbolization is to define symbols, i.e., assign the range of original data into a specific symbol value. Then, how can we decide the range of time-series? This is the key ingredient of symbolization.

One can define symbols in scalar time series, and the other does that in phase space [2][3]. Due to the irregular behavior of chaotic data, the letter method will be more desirable in some applications. Our goal in this step is to minimize the loss of information of original data. In this purpose, one can use the entropy to measure the information loss. How can we acquire the number of symbols and the choice of partition? When the entropy is maximized, optimal choice of two factors is possible [10]. Also, we must be careful not to lose important information.

After deciding the number of symbols and partition between them, we need to collect groups of symbols such that it should give a well defined structure. Then, how can we capture all information – finite symbol sequence length? This is similar to choosing time delay and embedding dimension in phase space. In order to find optimal time delay and embedding dimension in phase space, we use autocorrelation function, mutual information function. The same methods can be applied to symbolic space.

There are two possible ways to define symbol sequence, code series [11] and symbol tree [12]. The former approach divides the measured time series into finite set of discrete values (Figure 3) and then evaluate relative frequency of all possible sequences in the data defined by a symbol-

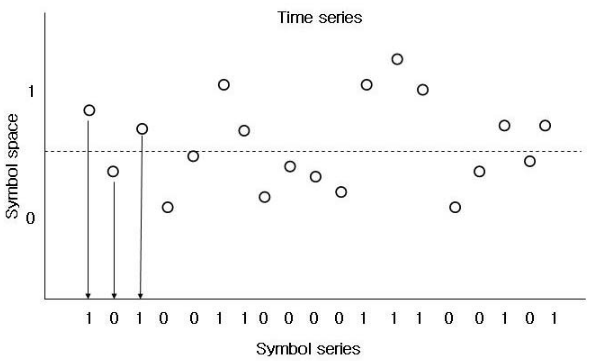

Figure 3: Process of code series method

sequence vector of L cycles length. For example, if we let L=5, we determine the relative frequency of occurrence for each possible sequential combination of five symbols.

The letter one is explained as the following. We have a stream of symbols, which are quantized from continuous-valued observations. And then the symbol tree is composed of parallel branches, each of which represents a possible sequence of the available symbols (Figure 4).

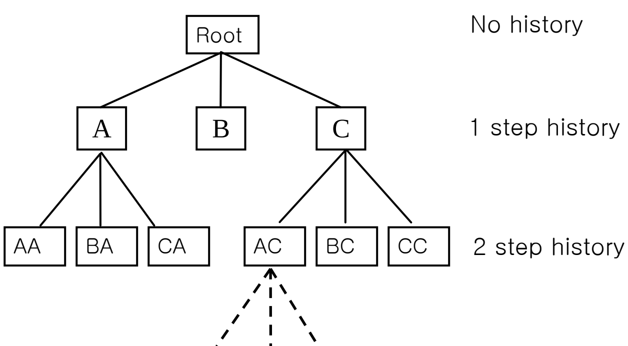

Figure 4. Example of small context tree

The length of the sequences determines the depth of the tree and thus the number of braches. For a fixed sequence length of L successive symbols, the total number of braches is n^L, and thus the number of possible sequence increases exponentially with tree depth. However, this has some drawbacks for any given dynamical system. All sequences are not realizable sequence such that we have nonoccurring sequences. This was improved by context tree [12][13], which allows some of the possible sequences to be shortened to reflect reduced predictability over long times (Figure 4).

The ultimate focus on symbolic dynamics in time-series analysis is to get the statistics from the symbol-sequence series. According to no intersection theorem in chaotic system, all time-series with different initial condition have unique trajectory in phase space. Following orbit, the relative frequencies for the intersections in various regions can be statistically quantified, and the resulting temporal sequence of symbols could be studied as a replacement for original variables [4].

Various types of statistics can be determined from the estimated symbol-sequence probability distribution. The most frequently used method is histogram, which is easy and convenient to calculate. From the statistics of symbol sequence, it often results in redundancy configuration between the frequencies of different sequences that reduces actual degree of freedom. In [14], weighted context tree method is proposed by Kennel and Mees to minimize such redundancy.

Examples of information measures for symbol-sequence frequencies include the Shannon and order-q Renyi entropies defined, respectively, as

$$H = -\sum_i p_i \log_2 p_i \tag{6}$$

and

$$H^q = \frac{1}{1-q} \log_2 \sum_i p_i^{\ q} \tag{7}$$

where p is the histogram of symbol-sequence frequencies. The base-2 logarithm places the entropies in units of bits.

3.2 Two generating partition algorithm

Sequence partitioning is the central issue in symbolic dynamics, when time-series data map into symbolic space. There have been many methods for this purpose, here I introduce two recent methods. One is using UPOs [2], the other one is using the concept of global energy [3]. From these examples, we can see the possibility of UPOs in nonlinear time series analysis. Here, we reduce the detailed explanation and equation for that.

3.2.1 UPOs algorithm

As we said in the introduction, UPOs give symbolic meaning on the chaotic attractor. Until now UPOs have been mainly used for control systems. So, this attempt is really valuable for the future research. The notion of generating partition is based on splitting of the phase space in terms of measurable set. The coarse features of chaotic attractors are typically revealed by a relatively small number of short UPOs, while increasingly longer orbits refine the feature without altering the general structure. Therefore, orbit points of longer UPOs are the most likely to be assigned the same symbols as the nearby points belonging to shorter UPOs.

Consider an N-dimensional dynamical system $\mathbf{x}_{n+1} = f(\mathbf{x}_n)$ and assume that we know the location of the UPOs of up to a relatively large period. The number K of symbols necessary for the symbolic representation must be large enough to allow for unique encoding of all UPOs in the system: $K \geq \max_p \sqrt[p]{N_p}$,

where N_p is the total number of orbit points of period p, including orbit points with periods that are factors of p. Our goal is to assign each orbit point $\mathbf{x}_i$ a symbol $s_i \in A = \{\alpha_1, \ldots\ldots, \alpha_K\}$, such that all the UPOs are represented by distinct symbol sequences.

In order to have a quantitative measure, we define the following proximity function of order p for an arbitrary point x in the phase space:

$$Z_p^{(k)}(x) = \sum_{i=1}^{N_{\leq p}} \frac{\delta_{\alpha_k, s_i}}{|\mathbf{x} - \mathbf{x}_i|^2}, \ \mathrm{k} = 1, \ldots\ldots, \mathrm{K}, \qquad (8)$$

where N is the total number of orbit points whose periods are less than or equal to p and δ_{α_k, s_i} is the Kronecker delta, which select for the sum only those points encoded by the symbol α_k. The choice of the function $|\mathbf{x} - \mathbf{x}_i|^{-2}$ is not unique, as long as it satisfies the following requirement: it must be a positive monotone decreasing function which tends to $+\infty$ in the limit $\mathbf{x} \rightarrow \mathbf{x}_i$. We can now divide the phase space into K domains $\{B_k\}_{k=1}^K$ such that $Z_p^{(k)}(\mathbf{x}) \geq Z_p^{(j)}(\mathbf{x}), j \neq k$, and, therefore, define a partition which distinguishes all UPOs up to at least period p

3.2.2 Relaxation algorithm

Here, we briefly presents relaxation algorithm to generating partitions for chaotic time series. Detailed procedure can be found in [3]. We have a time series of points $x_i \in R^d$, and we wish to assign symbols s_i to each observed point. Our final objective is to choose a partition where neighbors in the symbolic representation $\sum$ are neighbors in continuous state space R^d. Similar to other optimization problem, first define initial neighbors in $\sum$, and then update the partition until the above criterion satisfied. We can view this as the similar problem to finding a ground state of a spin glass in statistical mechanics. Each point is assigned a configuration-dependent energy: points whose symbolic neighbors are close physical neighbors are given lower energy. The optimal partition is defined as the configuration of s_i which yields the lowest global energy.

4. LIMITATIONS AND POSSIBILITY

We reviewed Newton family methods for finding unstable periodic orbits in section 2. This method is a well known classical way to approximate solutions to an equation. While it is very feasible method to perform, it assumes the followings. We know the equation of motion and the equation is differentiable and continuous equation. Since we often do not know the equations of motion in many cases, it is difficult to estimate UPOs from observed time series using that method. One possible way is to use close returns. Whenever the trajectory approaches any UPO, we can capture the point for periods. It is still problematic that it requires large amount of data and it gives the unclear definition of closeness.

When generating partition in chaotic data, each point in the attractor defines a unique bi-infinite symbolic sequence. Thus each UPO should have a unique periodic symbolic sequence. As we said in the first paragraph, it is often difficult to find the UPOs with sufficient accuracy and quantity to apply the UPOs algorithm. Most UPO locating methods use the points in the time series to create a model of the dynamics and then find UPOs from the model. However, an accurate model requires a large amount of rather noise free data. In particular, with noisy, finite length time series, it is especially hard to find the high period orbits which are necessary to localize the partition boundary. In spite of these defects and uncertainty, UPOs and symbolic dynamics embedded in chaotic time series is still attractive research area for modeling and will eventually progress.

5. CONCLUSION

This paper starts with the possibility of Unstable periodic orbits (UPOs) for modeling nonlinear time series. Although it's not easy to find the link between UPOs and modeling, the fact that UPOs give the hierarchical meaning in the phase space has always stimulated us. We first looked into interval analysis for finding unstable periodic orbits (UPOs) on the chaotic attractor. This method can successfully find all cycles with lower period as well as higher period.

The idea of symbolization is to map specific time-series ranges into symbol space. Although this has some uncertainty in nonlinear system, it will be a very powerful tool for modeling.

Until now, active researches have performed in these fields, but the existence of actual link between two is not guaranteed. However, recent experimental result shows that the hierarchical structure and symbolic dynamics of nonlinear time-series link together such that generates realizable modeling tools in the near future. Symbolic dynamics also combined other topological methods such that it results in more reliable modeling technique. One more we want to mention is the following. After we convert time series into symbolic space, we can treat it as a usual character. So, we also apply various grammars on it and try to modeling it.

6. REFERENCES

[1] Zbignew Galias, “Rigorous investigation of the Ikeda map by means of interval arithmetic,” Nonlinearity, 15, 1759-1779, 2002

[2] R.L. Davidchack, Y.-C. Lai, E.M. Bollt, and M. Dhamala, Phys. Rev. E 61, 1353, 2000

[3] M. Buhl and M. B. Kennel, Phys. Rev. E 71, 046213, 2005

[4] C.S. Daw, C.E.A. Finney, E.R. Tracy, “A review of symbolic analysis of experimental data,” Review of Scientific Instruments, 74(2), 915-930, 2003

[5] J. Hadamard, J. Math. Pures Appl. 4, 27, 1898

[6] R.E. Moore, “Methods and Applications of Interval Analysis,” SIAM Studies in Applied Mathematics, 1979

[7] G. Alefeld, “Inclusion methods for systems of nonlinear equations – the interval Newton method and modifications,” in Topics in Validated Computations, ed. J. Herzberger, IMACS-GAMM, 7-26, 1994

[8] A. Neumaier, “Interval Methods for Systems of Equations,” Cambridge University Press, 1990

[9] M. A. Trevisan, M. C. Eguia, G. B. Mindlin, Physica D. Vol. 200, 75-80, 2005

[10] M. Lehrman, A. B. Rechester, and R. B. White, Phys. Rev. Lett. **78**, 54, 1997

[11] C. S. Daw, M. B. Kennel, C. E. A. Finney, F. T. Connolly, Phys. Rev. E 57, 2811, 1998

[12] M. B. Kennel, A. I. Mees, Phys. Rev. E 61, 2563, 2000

[13] M. B. Kennel, A. I. Mees, Phys. Rev. E 66, 056209, 2002

[14] M. B. Kennel, A. I. Mees, in Nonlinear Dynamics and Statistics, edited by A. I. Mees (Birkhauser Verlag, Boston, 2001), pp. 387-412

[15] M. Dellnitz, O. Schütze, S. Sertl1, “Finding zeros by multilevel subdivision techniques,” IMA J. Numer. Anal. 22, 167-185, 2002