Synchronizing variables and states in HH networks

Uriel I. Morone and Michael Eldridge
Department of Physics
University of California San Diego

Abstract

In this paper we introduce a technique called time-delay synchronization. This technique uses data from a single state-variable measurement at different times to determine the values of unmeasured state variables and parameters in a model of an experiment. We test the technique using simulated data and show that we are able to accurately determine the values of all state variables in a network of two or three Hodgkin-Huxley neurons, using only data from a single voltage measurement. In the case of the two neuron network we are also able to determine the values of GABA and Glutamate conductances in all synapses. In the three neuron network, we are able to to determine all state variables, however we found that determining the parameters of the synapse connections is much harder and were only able to do so for particular cases.

1 Introduction

One of the principle purposes of any physical theory is to make quantitative predictions about the universe. Often this takes the form of a model, or system of differential equations, which together describe the evolution of a physical system over time. Then, with appropriate initial conditions these models can be numerically integrated and the physical behavior studied in detail and future behavior can be predicted. Unfortunately many interesting physical systems cannot be measured completely. That is, we have not developed the mechanical techniques to accurately measure variables which we theoretically believe are necessary to accurately predict behavior. In the specific case of neurons, it is possible to measure voltage fluctuations. However, there are no good techniques for directly measuring the gating variables, which are necessary if one is to predict the neuron's behavior. Furthermore, recording simultaneous voltage measurements from multiple neurons is often very difficult if not impossible, so these are also undetermined in a network model.

An inability to measure all the variables makes it impossible to predict the behavior in any part of the system. Besides the inconvenience of being unable to match a numerical simulation to experiment there is a more fundamental problem that arises from an inability to predict. There are many possible models to describe most complicated systems. Especially in biology all the

mechanisms operating in a given system, whether its the dynamics of protein production or the spiking of an action potential, are not known. This means that many different models can be proposed for one particular process. These models can all have have qualitatively similar behavior. Classically, the primary way to distinguish between competing physical models is to see which ones do a better job of predicting experiments. Models that fail to predict observed behavior, no matter how elegant or well-reasoned, must be tossed out. Afterwards appropriate models can be compared on criteria such as accuracy and complexity, but at least approximate experimental verification is a must. If we are unable to measure all the degrees of freedom in an experiment, we cannot predict how our models expect the system to behave, and as a result we cannot verify or invalidate different models.

Luckily, the degrees of freedom in a model are not truly independent because their evolution depends on one another. In practical terms this means that any one measurement carries information about all of the other variables in the model. What we will use in this paper is a technique to extract this information and use it help us determine the values of the unmeasured variables, and in some cases parameters, for a small network of Hodgkin-Huxley neurons.

A well known technique used in nonlinear dynamics is known as "synchronization" [2] [1] [3] . This technique uses the phenomenon that identical systems, when coupled through one or more state variables, will often eventually display the same behavior in all of the state variables. To address the measurement problem described previously, one can take a dynamical system:

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x})$$

Then, we modify the variable for which we have measurements, and couple those measurements to our model. If we are able to measure x_i, we would modify the dynamics to be:

$$\frac{dx_i}{dt} = f_i(\mathbf{x}) + g(y_i(t) - x_i(t))$$

Where, $y_i(t)$ are the measurements we have obtained and g is simply a coupling constant we can tune to optimize our results. In the case of a neuron, for example, it is possible to measure voltage so x_i would be the voltage variable in our model and y_i is the time series of the measurements that we have recorded.

As might be expected, this technique has certain restrictions. In particular, a sufficient number of variables must be measured for the remainder to synchronize. The more degrees of freedom in a system, the more measurements are required. However, the question might be asked, what can we do if more measurements are not available?

The technique we use in the rest of this paper is referred to as "time-delay synchronization". It is an extension of the procedure above which seeks to pull even more information about a system from a single measurement. This should not be entirely unexpected, because while the value of x_i tells us something about the state of the system, the value of $\frac{dx_i}{dt}$ does as well, and so

on with additional derivatives which are not utilized at all in the previous synchronization procedure.

It is well known from the study of chaotic dynamics that plotting a variable versus itself after some delay can reproduce the properties of a traditional phase space trajectory with independent degrees of freedom. We can construct a pseudo-phase space or *embedding* space by plotting, for example, $x(t)$, $x(t+\tau)$, and $x(t+2\tau)$ in a three-state model, where τ is some constant time delay [5] [6] [7]. If this system is chaotic, a trajectory in the embedding space will have the same fractal dimensions as a plot of $\ddot{x}(t)$, $\dot{x}(t)$, and $x(t)$ or any other choice of three independent state variables. This is possible because $x(t+\tau)$ contains information about $\dot{x}(t)$. Furthermore $x(t+\tau)$, $x(t+2\tau)$ together contain information about $\ddot{x}$ and $\dot{x}$, and so on. This idea can be extended to higher dimensions by simply taking additional embedding dimensions: $x(t), x(t+\tau), ..., x(t+(DE-1)*\tau)$, where DE is the number of embedded dimensions.

Using a similar concept, we can look at the not only how our measured state variable $x_i(t)$ evolves over time, but also how the time-delayed coordinate $x_i(t+\tau)$ evolves, to learn about the state of other variables. The procedure for doing so is described in section 2 below. Section 3 shows the results of applying this technique to a 2 neuron network. Section 4 shows the results for a 3 neuron model.

2 Time-Embedding Procedure

We can use embedded dimensions to extract additional information about the unmeasured states, and use this information to synchronize our model. We can consider an embedding vector

$$\mathbf{s}(t) = [s_1(t), s_2(t), ..., s_{DE}(t)] = [x_1(t), x_1(t+\tau), ..., x_1(t+(DE-1)\tau)]$$

where $x_1(t)$ is the measured state. In what follows the Voltage of neuron 1 was always used, though this need not be the case. In general, measurements from different states could be used, with time delayed variables from each one making up the entries of s_n. The outline of the time-delay synchronization is the following:

1. Choose arbitrary initial conditions
2. Calculate the Jacobian of the dynamical equations, $DF_{ij}(t) = \frac{df_i}{dx_j}(x_1, ...x_D)$
3. Starting with the Identity Matrix, $A_{ij}(t_0) = \delta_{ij}$ evolve A according to the Variational Equation

$$\frac{dA_{ij}}{dt} = DF_{ik} * A_{kj}$$

The entries of A represents the response of each state variable at a later time due to a perturbation in another variable at the current time. I.e.:

$$A_{ij}(t) = \frac{dx_i(t)}{dx_j(t_0)}$$

4. Make a new matrix $\frac{ds_i}{dx_j}$ by stacking the rows of A for which you have measured data.

   For example, if $x_1(t)$ is the measured state variable, after evolving A forward in time, one would form the matrix $\frac{d\mathbf{s}}{d\mathbf{x}}$ out of

$$\frac{ds_i}{dx_j} = A_{1,j}(t_0 + i * \tau)$$

5. Inverting $\frac{ds}{dx}$, using Singular Value Decomposition, we obtain $\frac{dx_i}{ds_j}$
6. Finally, using the deviation of our model from the data trajectory $\mathbf{\Delta s} = (\mathbf{s^{data}} - \mathbf{s^{model}})$ we can obtain a estimate of the deviation in the state variables $\Delta x_i = \frac{dx_i}{ds_j}\Delta s_j$, which we introduce as a forcing term in the dynamical equations:

$$\frac{dx_i(t)}{dt} = f_i(t) + g\frac{dx_i}{ds_j}\Delta s_j$$

This procedure allows us to synchronize all the state variables simultaneously, rather than only the measured components. The added coupling makes it possible to synchronize models that require more than the available number of measurements.

3 Two Neuron Network

A two neuron network provides the simplest case in which to test the time-delay embedding technique. To begin with a scenario designed to provide the best chance for success, we look to so called "twin experiments". These experiments are performed by integrating dynamical equations forward in time to produce idealized data. Once this data has been produced, the synchronization algorithm can be passed a limited subset of this data. This procedure permits the algorithm to access only the limited data set on which it is designed to operate, while allowing the experimenters to maintain complete knowledge of all variables and thus better assess the performance of the algorithm. Using twin experiments, we can investigate the successes, limits, and idiosyncrasies of an algorithm while tailoring the techniques for use in an experimental setting.

A two neuron network with possible inhibitory and excitatory connections allows for sixteen possible network topologies. It is important to note that some possible topologies do not permit state and parameter estimation under any conceivable estimation scheme. To see this, consider the case in which there are no synaptic connections between our two neurons, called neuron 1 and neuron 2. In this case, because neuron 1 receives no information from neuron 2, it cannot produce any estimation of the state of neuron 2 which is better than a random guess. More generally, since the voltage of neuron 1 depends on the voltage of neuron 2 only through the synaptic gating dynamics, if the synaptic conductivities associated with neuron 1 vanish, neuron 1 will have no information about the behavior of neuron 2.

For each of the possible topologies in which neuron 1 receives some information about the voltage of neuron 2, we were able to successfully estimate the state of neuron 2 and synaptic conductivities using only the voltage measured at neuron 1. Since we allowed for two possible synapses per neuron, six equations were required to describe each neuron, resulting in 12 state variables needed to specify the state of the network along with 4 additional parameters needed to determine the strengths of the synapses. Representative results of this procedure are shown in the table below.

gGABA1	Error	gGABA2	Error	gGlu1	Error	gGlu2	Error
1.2	0.63%	0.0	ϵ	0.0	ϵ	0.0	ϵ
0.0	$1.3x10^{-11}$	0.0	$3.1x10^{-3}$	0.2	0.04%	0.4	0.19%
1.2	0.43%	0.0	$1.3x10^{-3}$	0.2	0.82%	0.4	0.15%
1.2	0.4%	0.6	10.1%	0.2	3.6%	0.4	15.7%

Table 1: Results of the estimation procedure for the two neuron network. The error column contains either the percent or absolute error. The symbol ϵ indicates that the algorithm converged to within machine precision.

4 Three Neuron Network

When we increase the number of connected neurons to three, there are a total of 18 state variables. There is a Voltage, m, n, and h gating variable for each Neuron and two synapse gating variables for each neuron that couple it to the other two. Due to the added complexity of introducing a third neuron, we only had time to use inhibitory connections in this network. Even so, the three neuron network proved particularly tricky to synchronize, especially for arbitrary connection strengths.

	Neuron 1	Neuron 2	Neuron 3
Neuron 1	0	1.5	0.5
Neuron 2	1.0	0	0.0
Neuron 3	0.5	2.0	0

Table 2: Conductance Parameters for 3-Neuron Network that successfully synchronized. Along the top are the Neurons the connection is from, on the left are the Neurons the connection is to which are being inhibited. So N1 is inhibited strongly by N2 and weakly by N3, N2 is only inhibited by N1, N3 is strongly inhibited by N2 and weakly by N1

When the neurons were connected with connection strengths shown in Table 2 we were able to successfully synchronize all 18 of the state variables. Initially, the connection strengths, $gGABA_{m,n}$ were "known" to the model. What this means is that the model used the same fixed values of GABA conductances as were used in generating the data. However, of the variables only the Voltage data of Neuron 1 was available to the model. Nevertheless, with this single measurement we were eventually able to accurately synchronize the model, as shown in Fig 1. The synchronization error (essentially the squared percentage error per state variable) dropped to about 10^{-4} by the end of the synchronization, Fig 2.

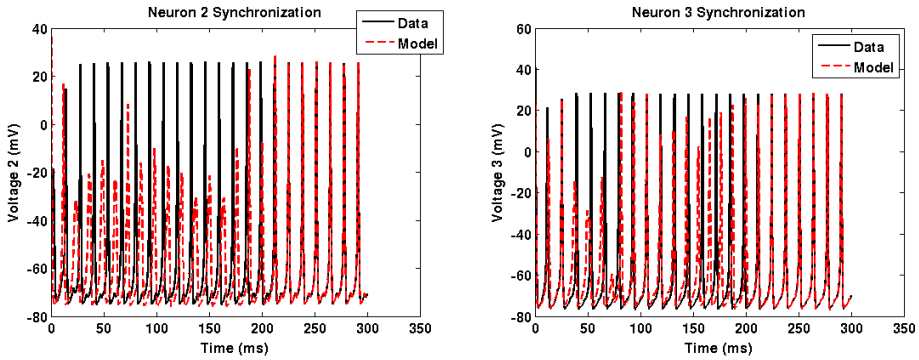

Figure 1: Voltages of Neuron 2 and Neuron 3. Black is the (unknown) data value. Red shows the value of the variable in the model which we were able to bring within 0.1% of the data value, using only information from Voltage of Neuron 1.

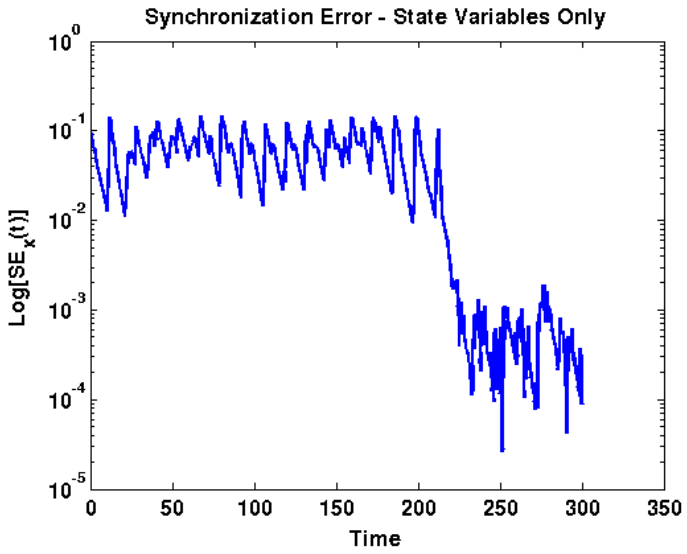

Figure 2: Synchronization Error (percent error per state) over time.

We can confirm that the synchronization is succifienctly accurate by seeing how long we can predict for. Figure 3 shows the data trajectory 5000 ms after the synchronization period (which lasted 300 ms as shown in Figure 1). As well as the model which we obtained only from the measurement of Voltage 1. The trajectories are practically indistinguishable because of how well the model predicts the data trajectory.

On the other hand, certain arrangements of connections were unable to synchronize. This should not be too surprising, because if certain connections are very small two of the Neurons might only weakly related to each other. The less significant the measured variable is on another, the more difficult it is going to be to synchronize them and in some cases it won't be possible at all. Fig 4 shows an arrangement that fails to synchronize. We see that although Neuron 2 synchronized very effectively Neuron 3 was unable to approach its true value at all. This is most likely due to how weakly Neuron 3 was coupled

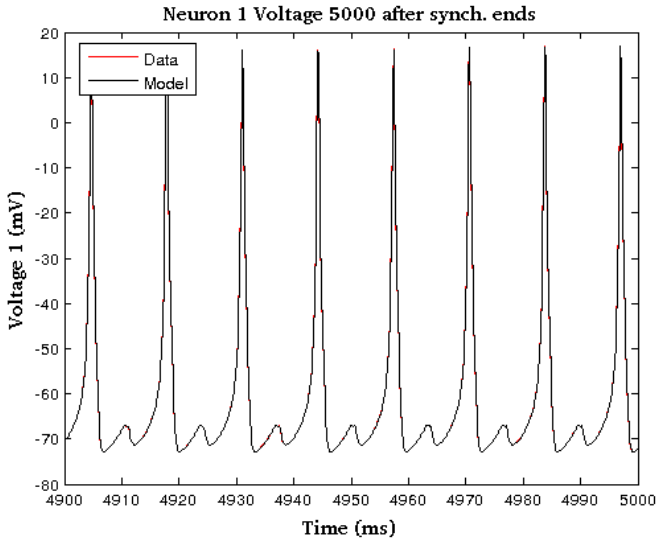

Figure 3: Neuron 1 Voltage variable. Data and Model trajectories were synchronized for 300 ms, then were decoupled and allowed to run forward independently. We see that even 5000 ms afterwards, the two trajectories are practically indistinguishable.

especially to Neuron 1, but also to Neuron 2. This meant that the measured component was able to achieve good accuracy even without driving Neuron 3 to the correct value.

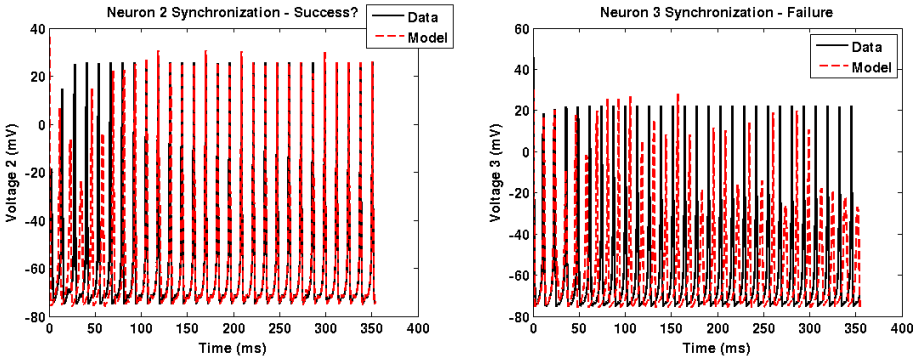

Figure 4: Voltages when neurons fail to synchronize. Although Neuron 2 is able to match the Data value, the Neuron 3 variables fail completely. The difference in this model is a different arrangement of gGABA connections at each synapse.

Given this difficulty in accurately determining variables with certain network topologies, it should not be surprising that trying to actually determine the network topology is extremely difficult. The more connections are unknown, the more unlikely it is that the model will synchronize to the data, and the worse the divergence seems to be. However, for the relatively simple case of only determining the two inhibitory inputs to Neuron 1, the model is able to synchronize with data. Figures 5 shows the Voltage of Neuron 2, the m-gating variable of Neuron 1, and the r31 gating variable, which inhibits Neuron 3 due to Neuron 1. These are shown only as a sample of the 18 state variables,

to show the model successfully synchronized in more than just the Voltage variables. Meanwhile, Figure 4 shows $gGABA_{1,2}$ and $gGABA_{1,3}$, which are the conductances for the inhibitory currents into Neuron 1. As we can see the time-delay synchronization is able to successfully pull these parameters from initial values of 0 towards their true data values. The best synchronization that we were able to achieve was within 5% in the state variables.

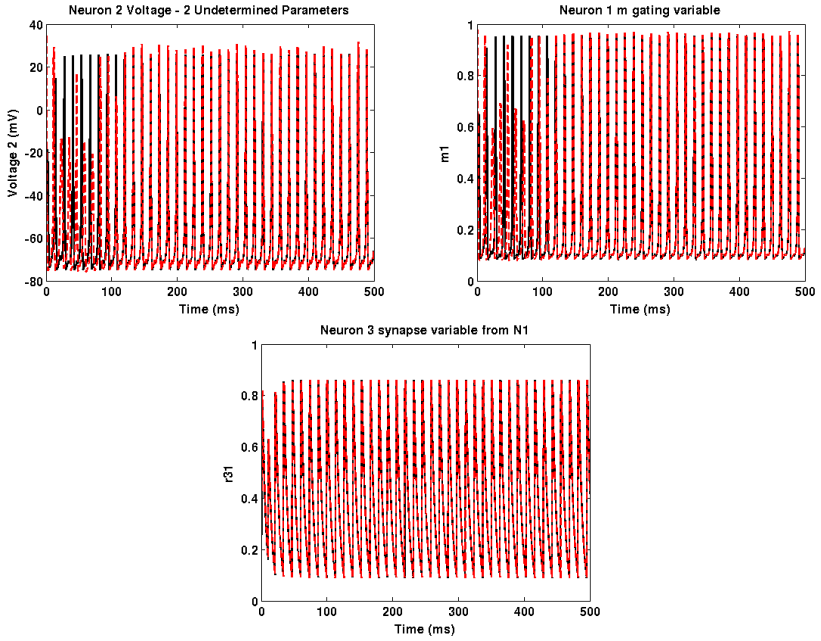

Figure 5: Left: Voltage Neuron 2. Middle: m-variable Neuron 1. Right: $r31$ synapse variable inhibiting Neuron 3 due to 1. Synch. with two parameters ($gGABA_{12}$ & $gGABA_{13}$ undetermined

The final values of the parameters that we were able to obtain were $gGABA_{12} = 1.2692mS$ and $gGABA_{13} = 0.4556mS$. Longer synchronization did not improve these estimates, as they tend to asymptote. Unfortunately, Figure 7 shows that these estimates are insufficient, because they do not really allow us to predict into the future a substantial amount of time. This demonstrates the difficulty of what we are trying to do. Even when the technique

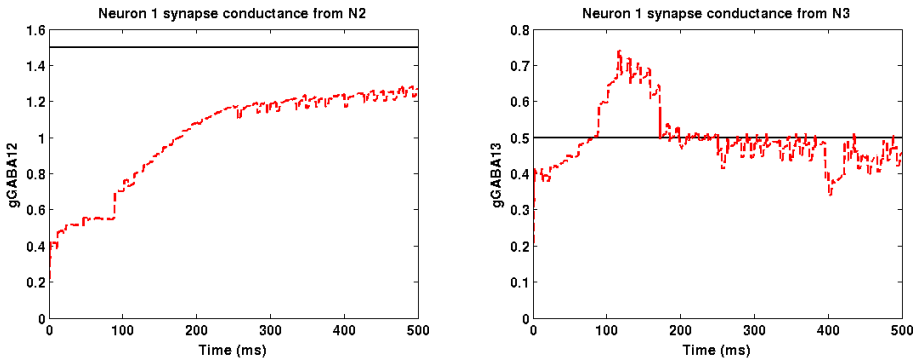

Figure 6: $gGABA_{12}$ and $gGABA_{13}$ after 500 seconds of synchronization

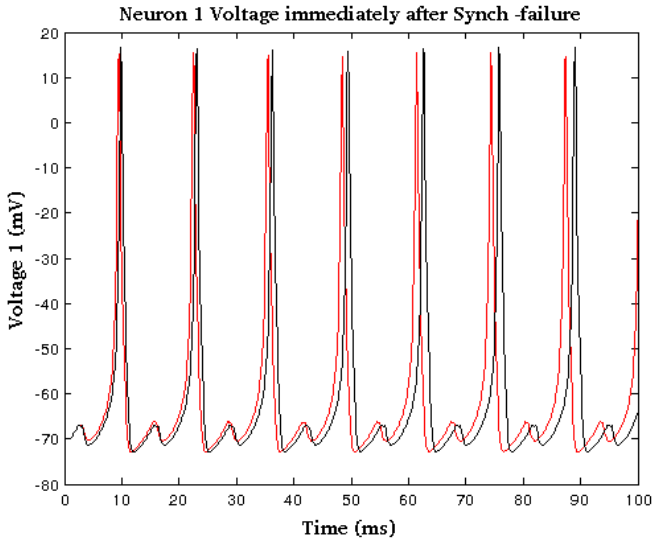

Figure 7: Attempt to predict when parameters are not accurately estimated. Even with a different of gGABA: 1.2692 vs 1.5 and 0.4556 vs 0.5 the trajectories immediately diverge.

apparently works approximately and might give us a better idea of the parameters, it is still insufficient to do what we need, which is accurate prediction.

However, it is possible that because the state variables managed to approach their correct values even without the exact parameters, there might be a way to combine this synchronization with some other optimization techniques to accurately determine the parameters.

5 Conclusion

Using time-delay embedding synchronization, we can successfully estimate the state and topology of two neuron networks from a single voltage measurement. In limited circumstances, we can extend this result to three neuron networks but the problem appears to become drastically more difficult as the size of the network increases. While not ideal, this result is unsurprising as the number of conceivable topologies grows with the number of neurons in the network as $2^{2N(N-1)}$. Using synchronization to estimate network parameters requires the use of time-delay embedding and is a novel result. To show that this result has applicability to experimental data, two further results would be required. First, we would need to illustrate that these results were robust under noise. While showing this result is beyond the scope of this project, the use of embedding dimensions implicitly averages over data points and seems likely to reduce the impact of noisy data. Second, barring the existence of an experimentally viable two neuron network, we would need to show that this technique can be used to make predictions in larger networks. As it seems unlikely that we can use time-delay synchronization to estimate the state of large networks, we would need to show that we could isolate a network of a few neurons from the effects of the larger network. Designing injected currents

that assist with this feat is an area for future study. Overall, while our results are of limited experimental utility in their current form, they provide novel results and illustrate an important proof of concept.

References

[1] U. Parlitz, L. Junge and L. Kocarev, Phys. Rev. E 54 6253 (1996) 6253.

[2] U. Parlitz, Phys. Rev. Lett. 76 (1996) 1232.

[3] H.D.I Abarbanel, D. R. Creveling, R. Farsian, and M. Kostuk, SIAM J. Appl. Dyn. Syst. 8 (2009) 1341.

[4] D. Aeyels, SIAM J. Control Optim. 19 (1981) 595.

[5] D. Aeyels, Syst. Control Lett. 1 (1981) 92.

[6] T. Sauer, J. A. Yorke, and M. Casdagli, J. Stat. Phys. 65 (1991) 579.

[7] Henry D. I. Abarbanel, Analysis of Observed ChaoticData, Springer-Verlag, New York, 1996.