Project 2: Largest Lyapunov Exponents

Eric LaForest

March 9, 2010

Contents

List of Figures

References

[1] ROSENSTEIN, M. T., COLLINS, J. J., AND DE LUCA, C. J. A practical method for calculating largest lyapunov exponents from small data sets. *Phys. D 65*, 1-2 (1993), 117–134.

[2] SPROTT, J. C. *Chaos and Time-Series Analysis*. Oxford University Press, 2003.

[3] WOLF, A., SWIFT, J., SWINNEY, H., AND VASTANO, J. Determining lyapunov exponents from a time series. *Physica D: Nonlinear Phenomena 16*, 3 (July 1985), 285–317.

1 Model

The system for this simulation is the FitzHugh-Nagumo (FHN) model:

$$\dot{V} = V - \frac{V^3}{3} - W + X$$

$$\dot{W} = 0.08(V + 0.7 - 0.8W)$$

driven by the output $X(t)$ of a Forced Negative Resistance Oscillator (FNRO):

$$\dot{X} = Y$$

$$\dot{Y} = 0.2(1 - X^2)Y - X^3 + F(t)$$

itself driven by a base oscillator stimulus $F(t)$:

$$F(t) = A\cos(\omega t)$$

The parameters of this experiment are the amplitude A and frequency ω of the base oscillator, varied over a range that was empirically determined to give some interesting variations in the final results:

$$A \in \{A_i | 16 \leq A_i \leq 18\}$$

$$\omega \in \{\omega_j | 2 \leq \omega_j \leq 5\}$$

This simulation investigates the chaotic behaviour of this system over this range by calculating the Largest Lyapunov Exponent (LLE) for both $V(t)$ and $X(t)$. As the system formulas are available, Wolf's algorithm [3] can be used to determine the LLEs. As a cross-check, a time-series is generated and analyzed for LLEs using Rosenstein's [1] algorithm. These algorithms, as well as additional clarifications, are also described in a more accessible manner in Sprott [2, Ch. 5.6 and 10.4].

2 Analysis Methods

Since I was writing the analysis code from scratch, in order to provide a cross-check against programming errors and parameter maladjustment, I implemented two different algorithms to find the LLE: Wolf's and Rosenstein's.

2.1 Wolf's Algorithm

Wolf's algorithm is straightforward and uses the formulas defining the system. It calculates two trajectories in the system, each initially separated by a very small interval R_0. The first trajectory is taken as a reference, or 'fiducial' trajectory, while the second is considered 'perturbed'. Both are iterated together until their separation $abs(R_1 - R_0)$ is large enough, at which point an estimate of the LLE can be calculated as $\lambda_1 = \frac{1}{\Delta t} \log_2 abs(\frac{R_1}{R_0})$. The perturbed trajectory is then moved back to a separation of $sign(R_1)R_0$ towards the fiducial, and the process repeated. Over time, a running average of λ_1 will converge towards the actual LLE.

In this analysis, the separation was deemed sufficient at $3R_0$ since $\log_2(3) > 1$, meaning at least one bit of information is gained about λ_1. Given double-precision numbers, Sprott recommends $R_0 = 10^{-10}$ as sufficiently small yet much larger than the minimum precision. The algorithm is iterated until the convergence error is less than 0.01. Finer precision was possible, but took an impractical amount of time to compute.

2.2 Rosenstein's Algorithm

Rosenstein's algorithm works on recorded time-series, where the system formulas may not be available. It begins by reconstructing an approximation of the system dynamics by embedding the time-series in a phase space where each point is a vector of the previous m points in time (its 'embedding dimension'), each separated by a lag of j time units. Although Taken's theorem states that an embedding dimension of $2D + 1$ is required to guarantee to capture all the dynamics of a system of order D, it is often sufficient in practise to use $m = D$. Similarly, although an effective time lag must be determined experimentally, in most cases $j = 1$ will suffice.

Given this embedding of the time-series, for each point I find its nearest neighbour (in the Euclidean sense) whose temporal distance is greater than the mean period of the system, corresponding to the next approximate cycle in the system's attractor. This constraint positions the neighbours as a pair of slightly separated initial conditions for different trajectories. The mean period was calculated as the reciprocal of the mean frequency of the power spectrum of the time-series, calculated in the usual manner using the FFT.

I can now perform a process similar to Wolf's algorithm to approximate the LLE: for each point and its nearest neighbour I calculate the logarithm of their separation, and then average the estimates together. This process is then repeated one step forward in time for each pair of neighbours, giving another average estimate. The fact that these estimates are repeatedly averaged over multiple trajectories spread over the entire time-series allows for fast and accurate results, even in the presence of noise (which is absent in this generated time-series) and a paucity of data points.

These estimates over time can be then fit to a line using least-squares, whose slope is the calculated LLE. Only the first few points are useful since as the distance in time increases, it is likelier that neighbours will begin re-converging, and thus the slope falls towards zero. In this analysis, the first 5 points were found to give a meaningful fit. If a least-squares fit could not be found, then the slope was assumed to be zero.

3 Results

In order to check the output of each algorithm, I first display them alongside each other, along with the variations in the system parameters.

3.1 Raw Output: Wolf

Figures 1 and 2 show the LLEs, calculated by Wolf's algorithm, of the FitzHugh-Nagumo (FHN) model ($V(t)$) and the Forced Negative Resistance Oscillator (FNRO) ($X(t)$) over the combined ranges of amplitude (A) and frequency (ω) of the base oscillator, both varied in intervals of 0.1.

Finer resolution was not practical as the calculation of each LLE took approximately 28 seconds, for a total of almost 5 hours for 600 data points. The cause was due to the poor convergence behaviour of the algorithm, and that my implementation attempted to calculate the LLE of all four values in the system ($V(t), W(t), X(t), Y(t)$), which ended up being unnecessary effort.

Overall, the change in amplitude has no effect on the LLEs of the system as it is always high enough to cause the FNRO to dominate the FHN system. A more interesting choice of amplitude might have been closer to 1, where the effect of stimulus to the FHN system begins to cause action potentials, and thus possibly some additional chaotic behaviour.

Conversely, the effect of varying the frequency was significant and repeatable. The effect is made more visible in Figure 2.

3.2 Raw Output: Rosenstein

Figured 3, 4, and 5 show the same juxtaposed parameter variations and LLEs as before, but calculated using Rosenstein's algorithm applied to a pre-generated sequence of 1000 data points over a time span of 0 to 100 in steps of 0.1.

Given the much greater efficiency of this algorithm, each LLE took only about 2 seconds to compute, and thus a finer resolution of 0.01 was possible in the variation of the parameters. This precision also allowed for a more detailed look in Figure 5.

As hoped, the output of Rosenstein's algorithm agrees with that of Wolf's algorithm, although with much more precision and speed. Both show the same general rise and fall in the LLE as the frequency varies, and neither are affected by the change in amplitude.

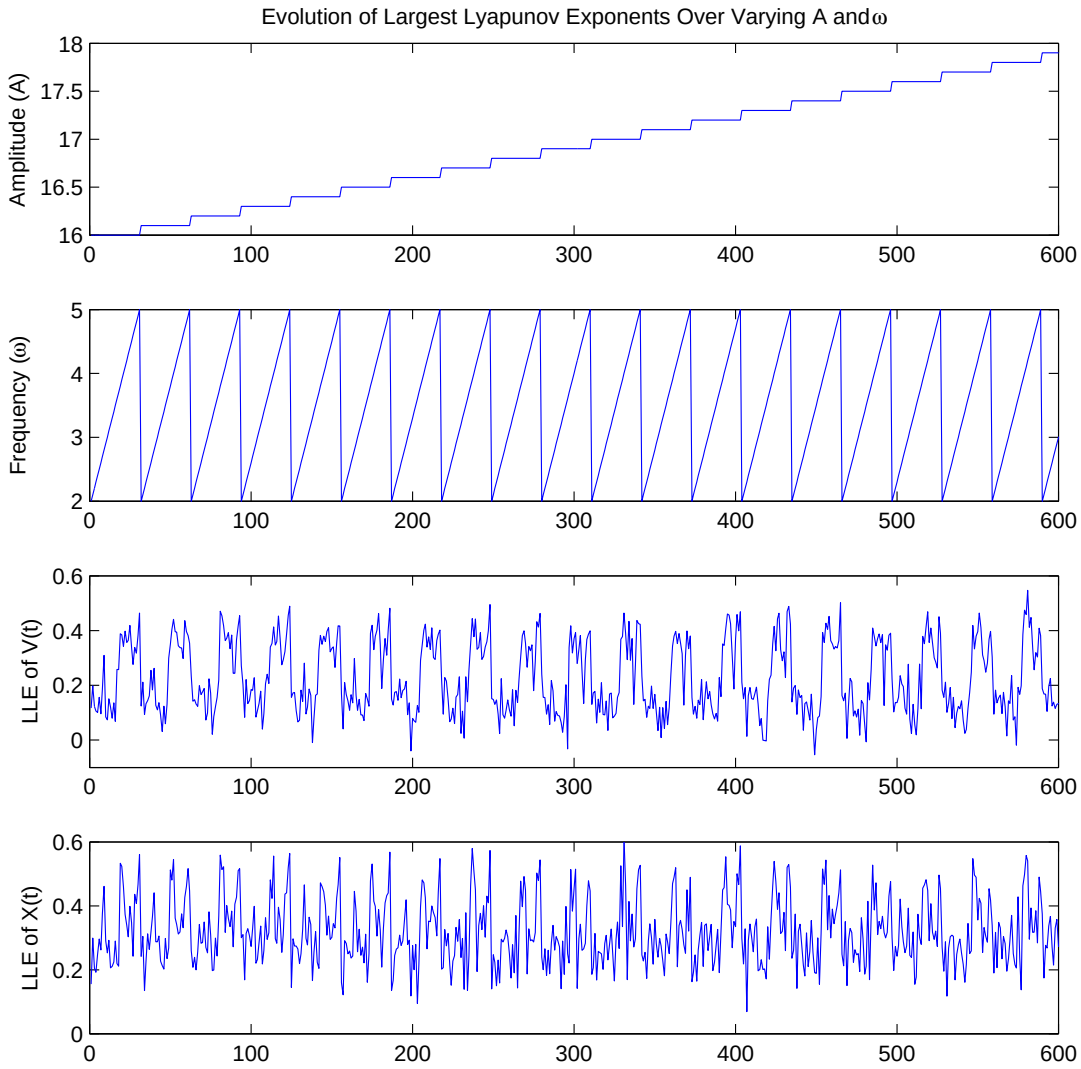

Figure 1: Wolf's Algorithm: Raw Output

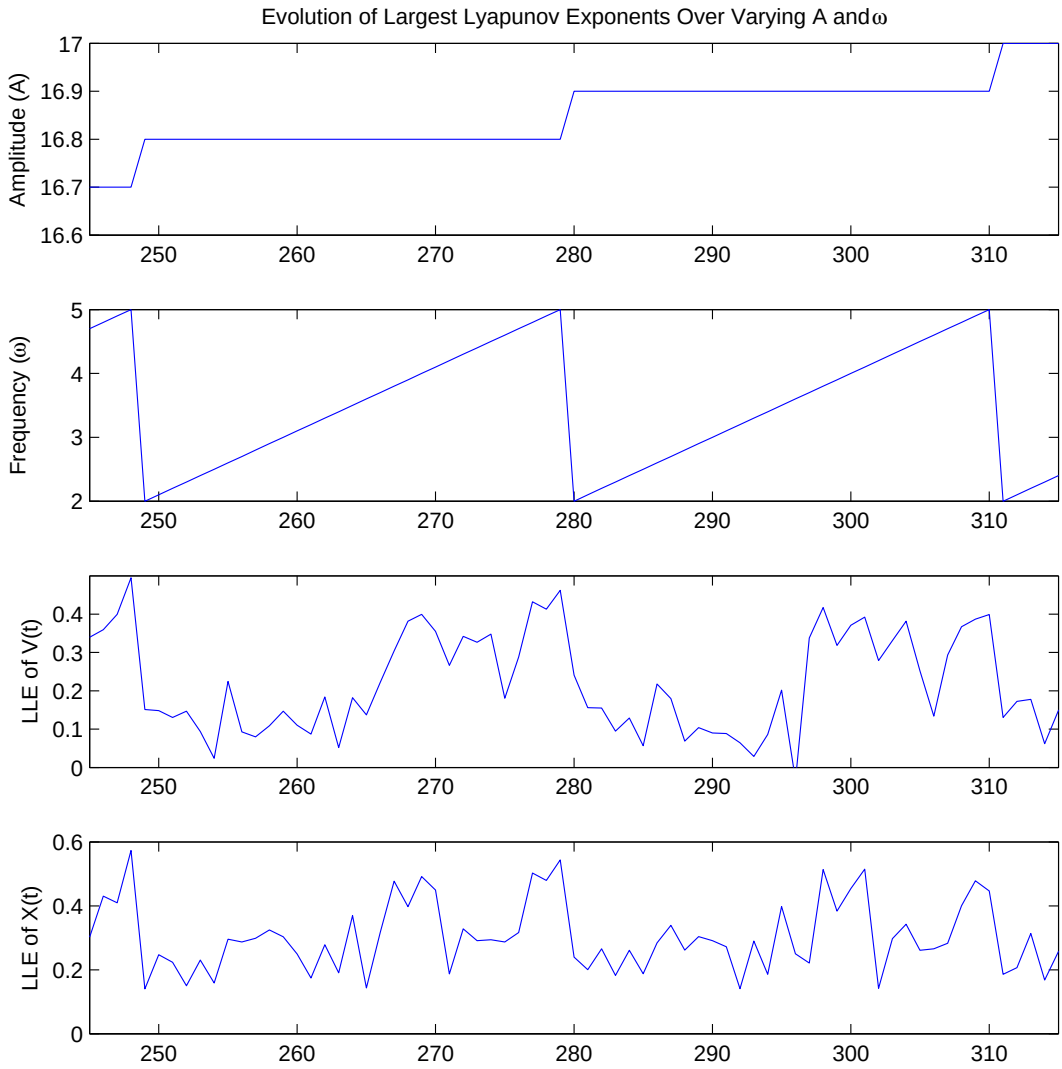

Figure 2: Wolf's Algorithm: Raw Output (Zoomed)

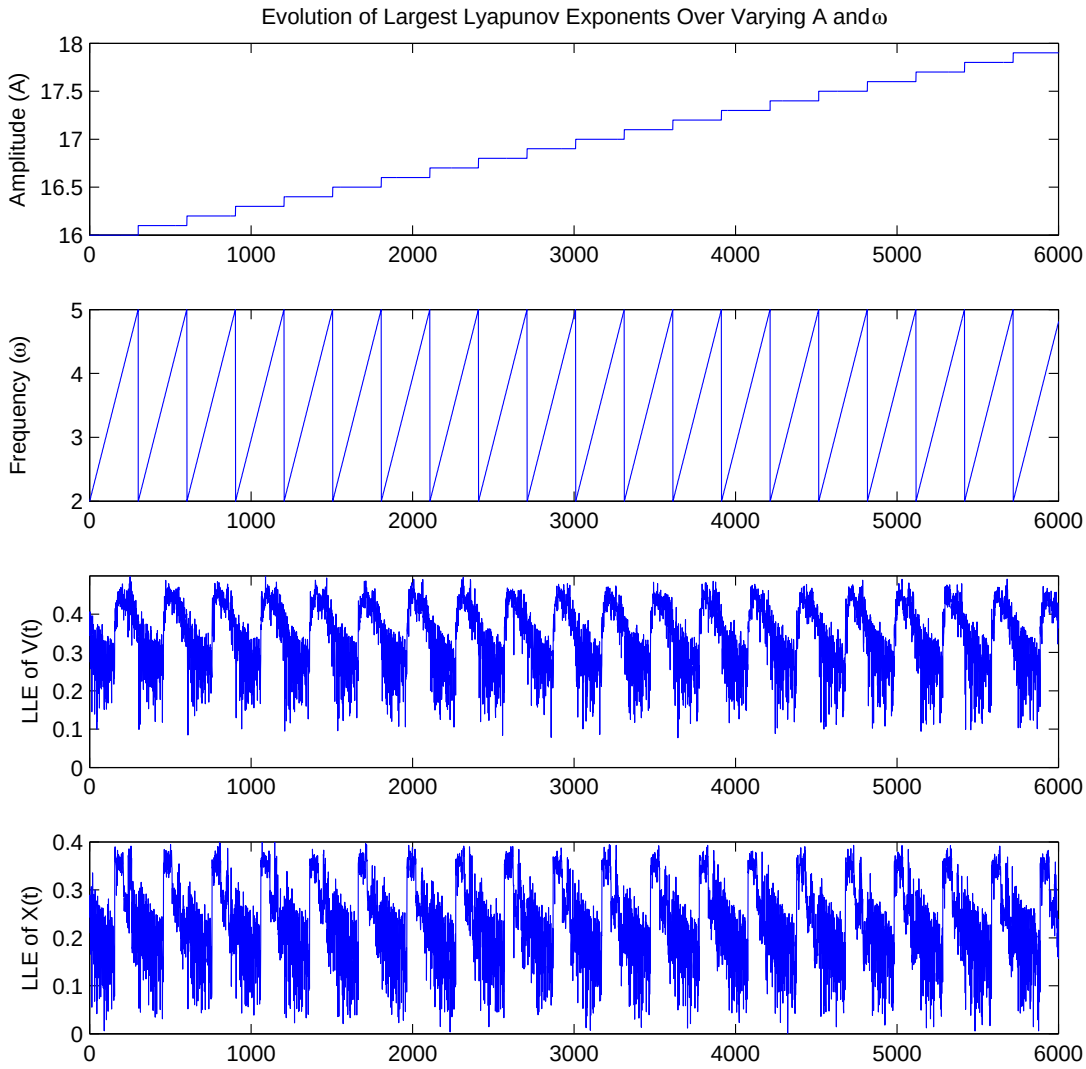

Figure 3: Rosenstein's Algorithm: Raw Output

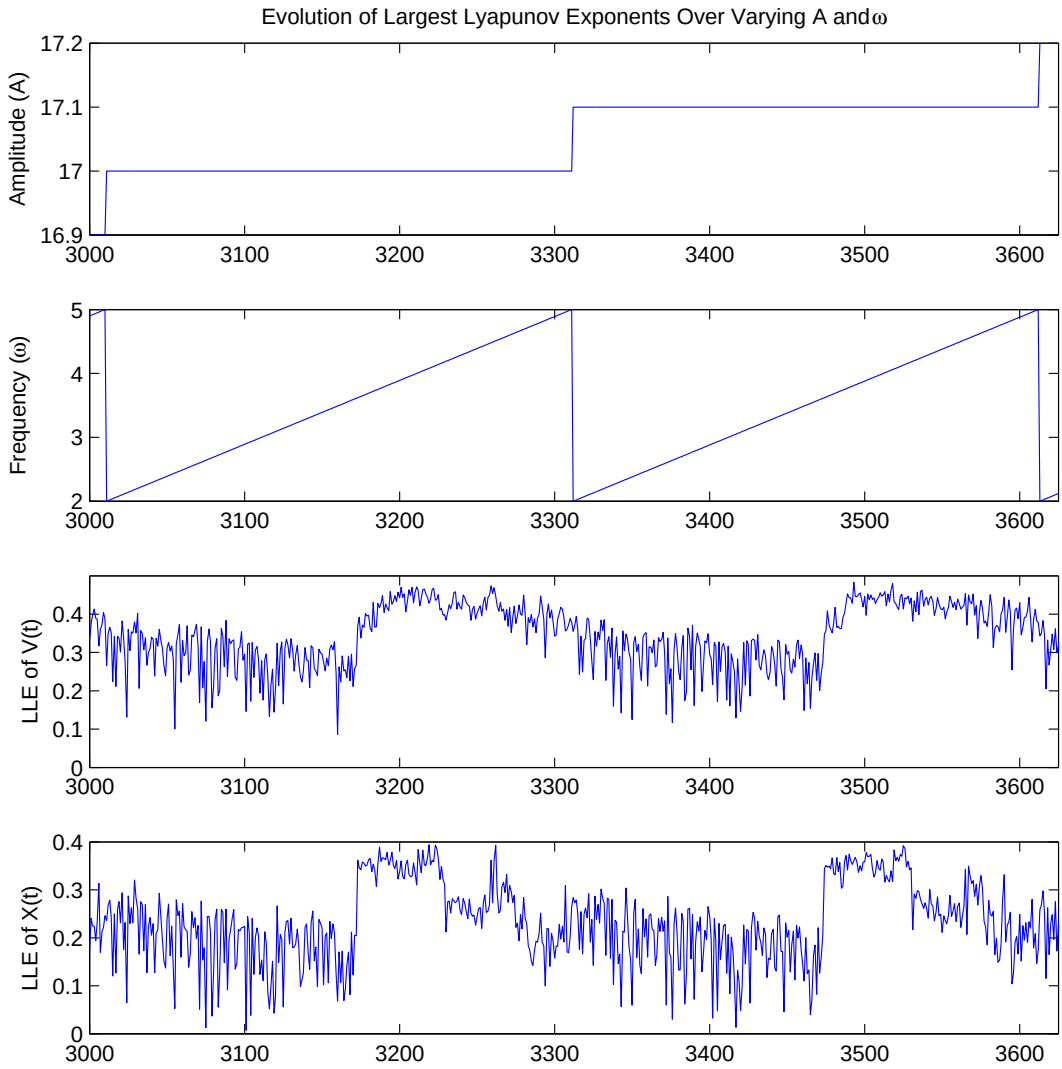

Figure 4: Rosenstein's Algorithm: Raw Output (Zoomed)

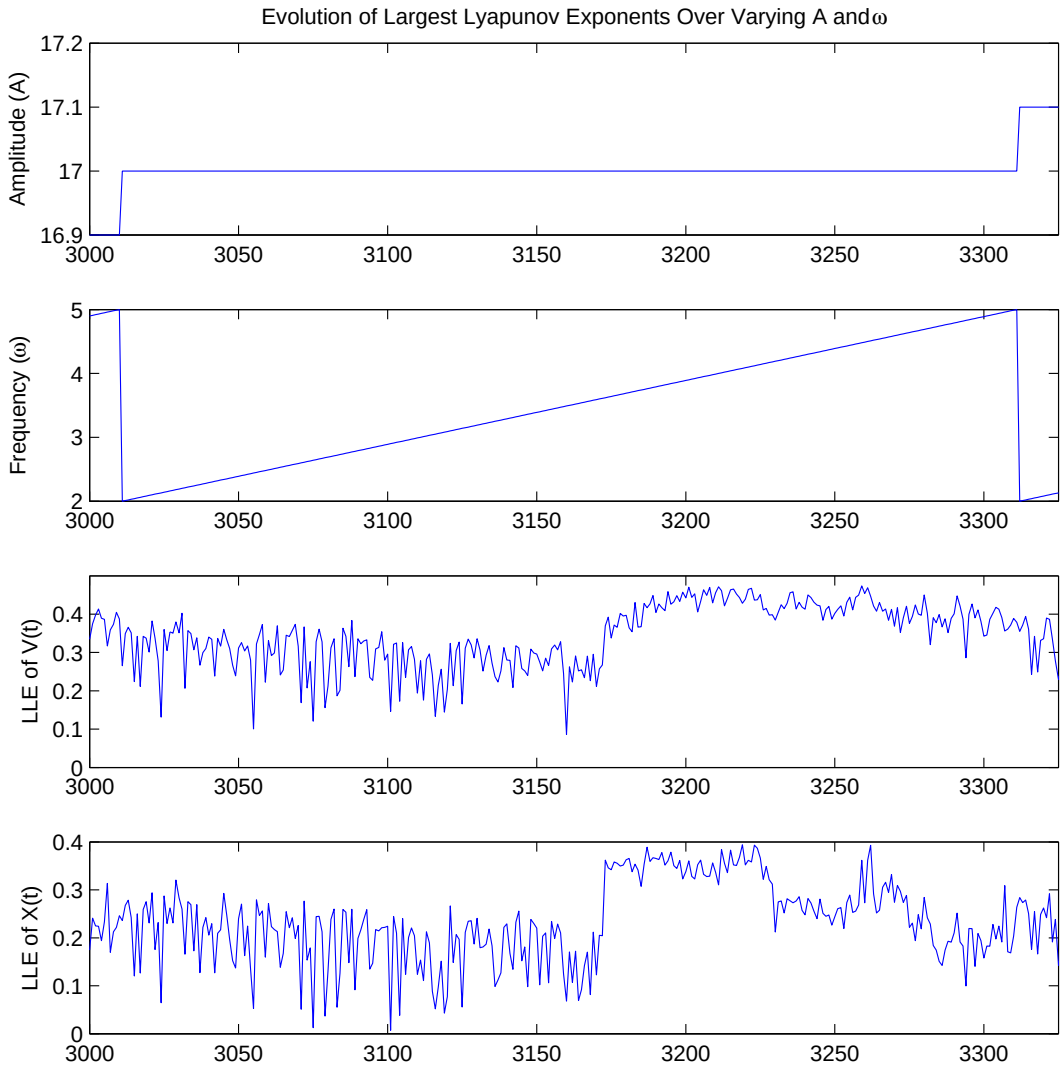

Figure 5: Rosenstein's Algorithm: Raw Output (Zoomed Further)

3.3 Relationship of LLEs

To further verify the relationship suggested by the previous figures between the LLEs of the FNRO and FHN models as the parameters change, Figures 6 and 7 show a scatter plot of the LLEs of the FHN model ($V(t)$) as a function of the LLEs of the driving FNRO ($X(t)$), with the base oscillator frequency (ω) as the colour of each point. The change in amplitude had no effect and is ignored.

Even with the coarse resolution afforded by Wolf's algorithm, is it clear that the relationship between the LLEs is proportional, if not exact. One can see that the LLEs are lower with a lower frequency (blue and cyan), jump higher somewhere between a frequency of 3 and 4 (yellow and orange), and then decrease along a shallower slope as the frequency increases from 4 to 5 (red).

These relationships are much more clear when using Rosenstein's algorithm, in Figure 7. The much greater precision and quantity of results clearly show the progression of the LLEs as the frequency increases: first in the middle (blue) and decreasing together until some point above 3 (cyan), then increasing again with a very sudden jump between 3 and 4 (green). Afterwards, the LLEs of the FNRO remain relatively steady while those of the FHN jump up (yellow). They then both decrease together along a shallower line (orange and red). These relationships match those first hinted at in Figures 3, 4, and 5.

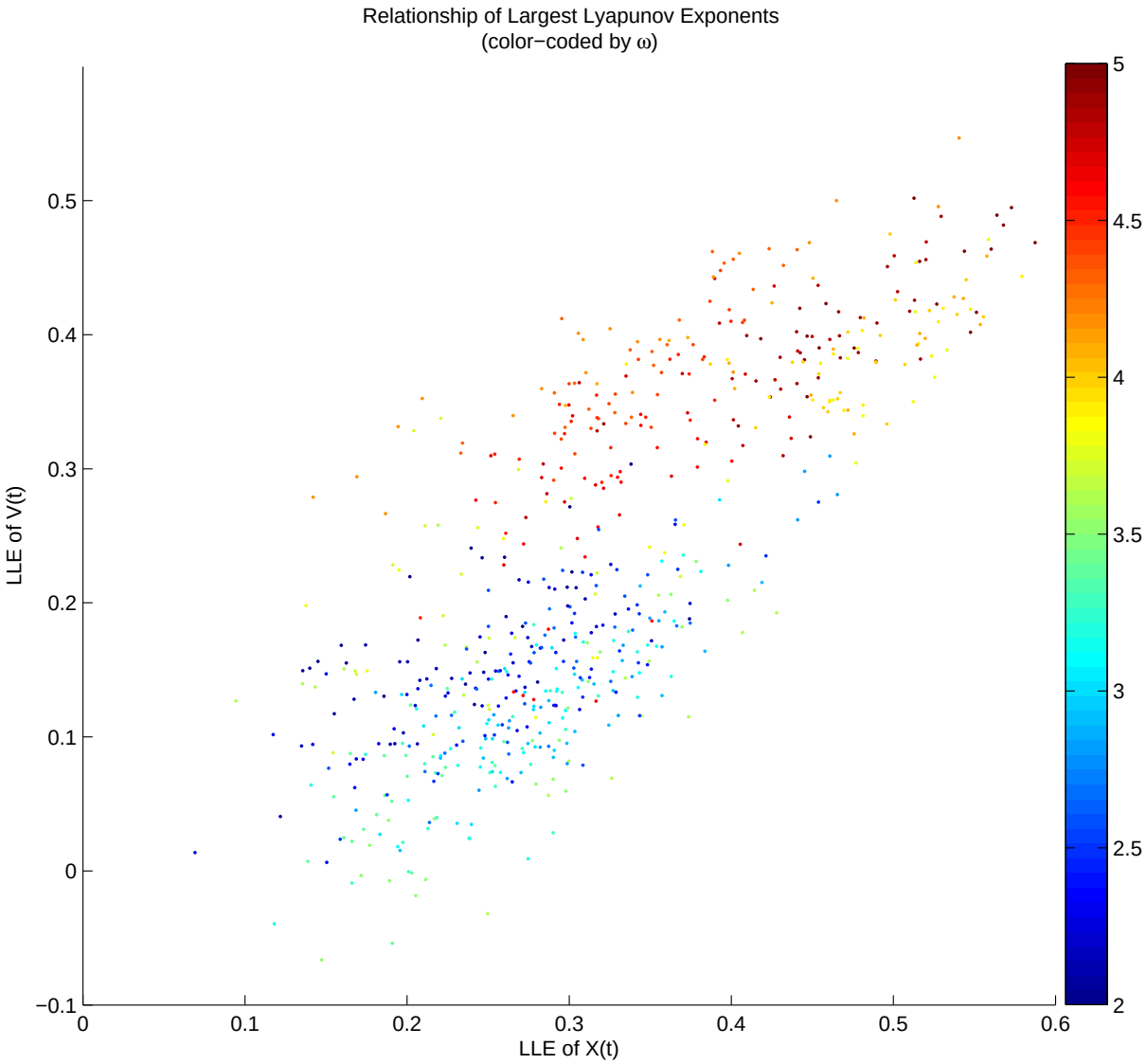

Figure 6: Wolf's Algorithm: Relationship of LLEs

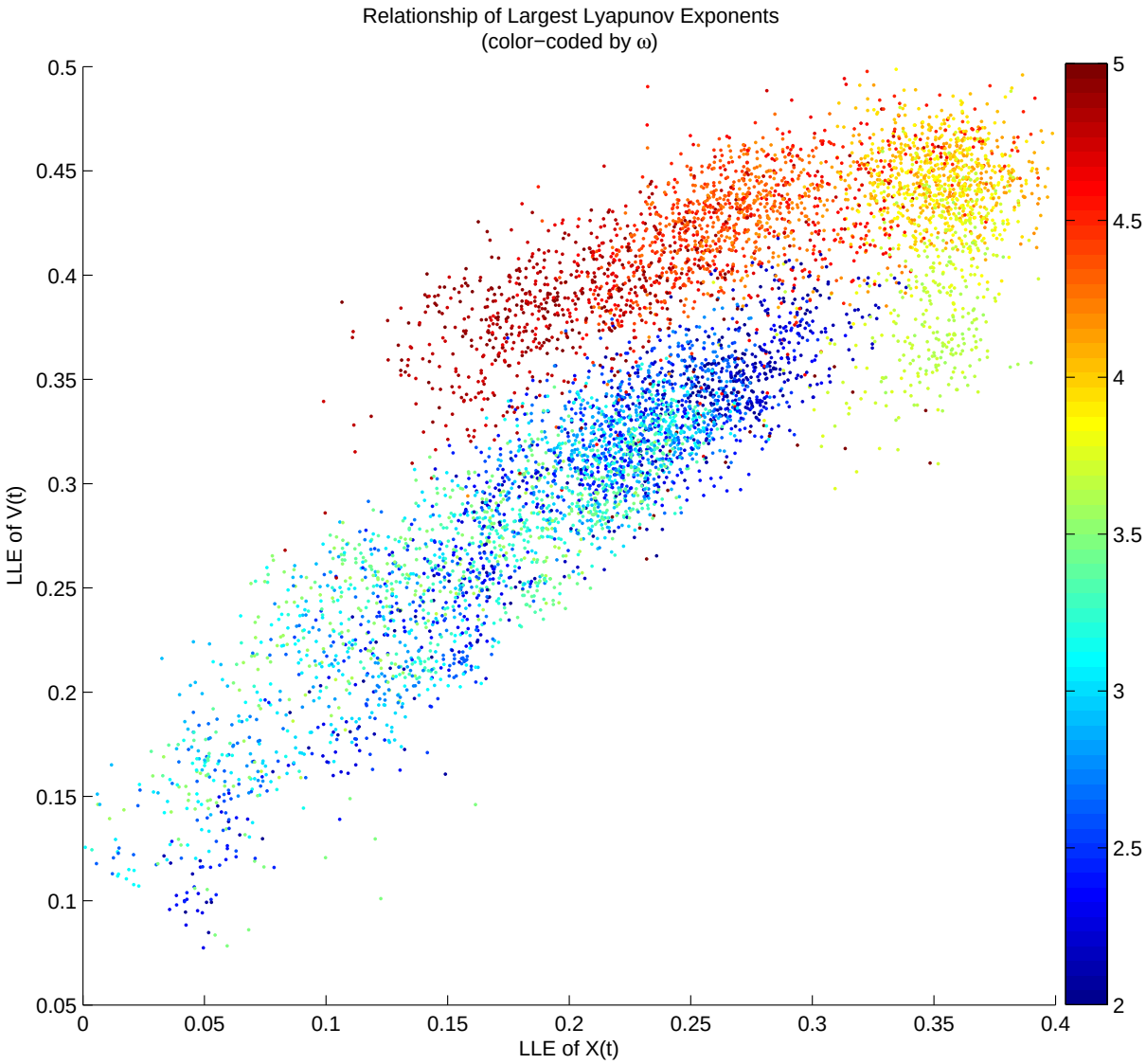

Figure 7: Wolf's Algorithm: Relationship of LLEs