Synchronization of Chaotic Systems

TJ Bollerman

May 4, 2015

Contents

---

1 Review of Chaotic Systems and the Lorenz System

Chaotic systems are deterministic, but are extremely sensitive to initial conditions. This sensitivity of the initial conditions makes long-term prediction of chaotic systems almost impossible unless the initial conditions are known exactly. One of the most famous chaotic systems is the Lorenz system. The Lorenz system is defined by

$$\dot{x} = s(y - x) \tag{1}$$

$$\dot{y} = rx - y - xz \tag{2}$$

$$\dot{z} = xy - bz \tag{3}$$

Lorenz originally formulated these equations to describe atmospheric weather dynamics. The system quickly grew to fame, as emergent chaotic properties of the system became known. The system dynamics of the Lorenz equation depend of the parameters s, r, b. For the right choice of the system parameters the solution traces out two the familiar strange attractors.

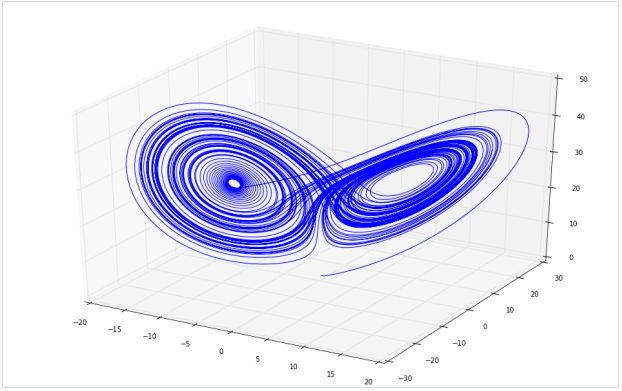

Figure 1: The phase space of the Lorenz system is plotted here for the initial conditions $[0, 1, 0]^T$ and $s = 10$, $r = 28$ and $b = \frac{8}{3}$

Many initial conditions will trace out the same strange attractors in phase space for the same parameter values, however the actual values of different solutions will differ dramatically. Figure 2 shows how two solutions with very slightly different initial conditions differ as time increases.

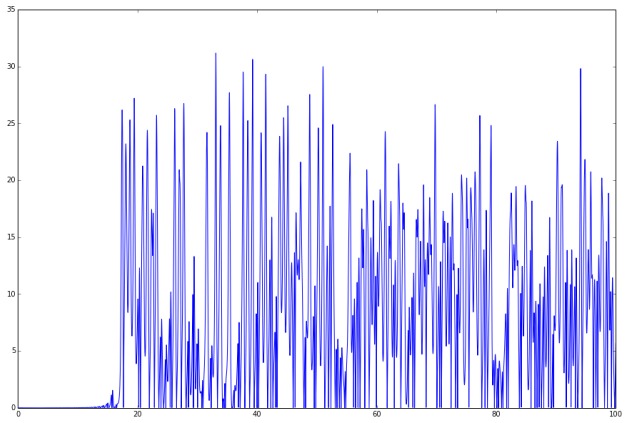

Figure 2: The difference of two solutions of the Lorenz equation with parameters $s = 10$, $r = 28$ and $b = \frac{8}{3}$ as a function of time is plotted here. One of the solutions has initial conditions $[0, 1, 0]^T$ and the other $[0, 1, 0.01]^T$

2 Lyapunov Exponents

As we have seen, solutions of the same chaotic system will diverge quickly even if there is only a very small difference in the initial conditions. The Lyapunov exponents provide a way to quantify this behavior. The Lyapunov exponents are defined as

$$|\delta \mathbf{Z}| \approx e^{\lambda t} |\delta \mathbf{Z_o}| \tag{4}$$

where $|\delta \mathbf{Z}|$ is the distance between the two solutions at time t, $|\delta \mathbf{Z_o}|$ is the distance of the initial conditions of the solutions and λ is the Lyapunov exponent. It is important to note that there is actually a spectrum of Lyapunov exponent as the rate of divergence may differ in different direction. The largest Lyapunov exponent is usually discussed as the Lyapunov exponent since it controls the predictability of the system. The Lyapunov exponent is very hard to find analytically for all but the simplest systems, instead numerical techniques are often used.

The Lyapunov exponent is intuitively very clear. Negative exponents imply asymptotically stable equilibrium points and stable trajectories, as different solutions converge to each other. A zero exponent implies that the solutions will stay the same distance apart for all time. An example of a system with zero Lyapunov exponents would be a mass on a spring with no damping. A positive exponent usually implies chaos (but other requirements must be met as well). Zooming in on the beginning of figure 2 we clearly see a positive exponent for the Lorenz system.

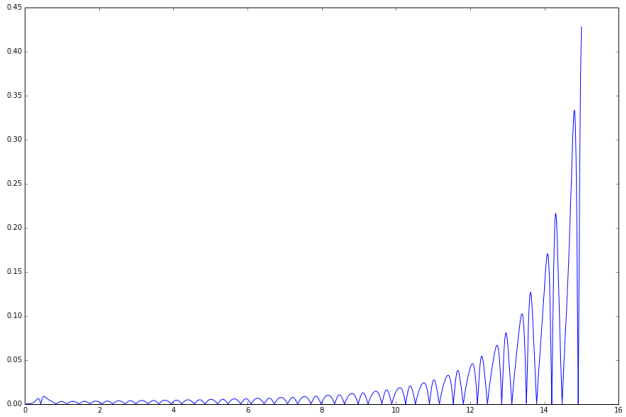

Figure 3: The exponential growth of the distance between the two solutions as described in figure 2 is clearly seen here

While figure 3 clearly shows the exponential growth of the distance between two solutions these mysterious 'lumps' appear. The distance in solutions continuously expands and contracts. While the theory of this phenomenon is left out of this summary for the sake of brevity, it is also the reason why systems with positive exponents do not necessarily have to blow up. [3]

3 Subsystems and Synchronous Subsystems

Louis Pecora and Thomas Carroll showed in 1989 that is possible to synchronize two chaotic systems. Synchronizing systems means that even with different initial conditions, the systems will converge together. Synchronization is thus obviously strongly linked to Lyapunov exponents. Synchronization in chaotic systems would seem impossible at first glance, but Carroll and Pecora showed that the subsystems w and w' will synchronize only if the sub-Lyapunov exponents are all negative. This theorem is only necessary, but not sufficient as it says nothing about the set of initial conditions which will allow for synchronization of w and w'. In order to understand this theorem is it necessary to first understand subsystems. [1]

Let an n dimensional system be defined as the following

$$\dot{u} = f(u) \tag{5}$$

We can divide the system in two parts arbitrarily. Let

$$v = [u_1, ..., u_m]^T \tag{6}$$

$$w = [u_{m+1}, ..., u_n]^T \tag{7}$$

and

$$g = [f_1(u), ..., f_m(u)]^T \tag{8}$$

$$h = [f_{m+1}(u), ..., f_n(u)]^T \tag{9}$$

It's clear that

$$\dot{v} = g(v, w) \tag{10}$$

$$\dot{w} = h(v, w) \tag{11}$$

Now, we can create a new subsystem w' identical to w but use the solution v instead of the variable v'. Intuitively, as the solution v is used as an input for w' it is often useful to think of v driving the system w'.

$$\dot{v} = g(v, w) \tag{12}$$

$$\dot{w} = h(v, w) \tag{13}$$

$$\dot{w'} = h(v, w') \tag{14}$$

Now lets look at the error between w' and w, obviously synchronization will occur only if

$$w' - w \to 0 \quad as \quad t \to \infty \tag{15}$$

After defining the subsystems w' and w we can understand Carroll and Pecora's theorem. Carroll and Pecora showed in their paper that the Lorenz system has two synchronous sub-systems. Recalling the form the Lorenz equations

$$\dot{x} = s(y - x) \tag{16}$$

$$\dot{y} = rx - y - xz \tag{17}$$

$$\dot{z} = xy - bz \tag{18}$$

Consider these sub-systems

$$\dot{x}_1 = s(y - x_1) \tag{19}$$

$$\dot{z}_1 = x_1 y - b z_1 \tag{20}$$

and

$$\dot{y}_2 = rx - y_2 - x z_2 \tag{21}$$

$$\dot{z}_2 = x y_2 - b z_2 \tag{22}$$

We can show that sub-system 1 has negative Lyapunov exponents by examining the Jacobian.

$$J = \begin{bmatrix} -s & 0 \\ y & -b \end{bmatrix} \tag{23}$$

The characteristic polynomail is

$$(s + \lambda)(b + \lambda) = 0 \tag{24}$$

The roots are

$$\lambda_1 = -b \tag{25}$$

$$\lambda_2 = -s \tag{26}$$

Since b and s are positive, the origin is asymptotically stable and thus has negative Lyapunov exponents. While it is more complicated to show subsystem 2 has negative Lyapunov exponents, Carroll and Pecora used numerical techniques to find that for the system parameters $s = 10$, $r = 28$ and $b = \frac{8}{3}$ the Lyapunov exponents for sub-system 1 and 2 are $(-1.81, -1.86)$ and $(-2.67, -9.99)$ respectively.

It would be desirable to have two full-sized system synchronize. In 1993 Kevin Cuomo and Alan Oppenhiem had the following idea: let $x(t)$ drive subsystem 2, then since $y_2 - y \to 0$ as $t \to \infty$ let y_2 drive subsystem 1. Using this idea we get the following full dimension system that should synchronize with the Lorenz with the same system parameters. [2]

$$\dot{x}_s = s(y_s - x_s) \tag{27}$$

$$\dot{y}_s = rx - y_s - xz_s \tag{28}$$

$$\dot{z}_s = xy_s - bz_s \tag{29}$$

Notice the solution x from the Lorenz system is used as an input in the system where we would expect to see x_s in the differential equations for $\dot{y}_s$ and $\dot{z}_s$. While this plan makes intuitive sense, we check the error dynamics of the Lorenz system and the synchronous system as defined above.

$$e = [x \quad y \quad z]^T - [x_s \quad y_s \quad z_s]^T \tag{30}$$

It is easy to show the error dynamics are as follows.

$$\dot{e}_1 = s(e_2 - e_1) \tag{31}$$

$$\dot{e}_2 = -e_2 - xe_3 \tag{32}$$

$$\dot{e}_3 = xe_2 - be_3 \tag{33}$$

Using the following positive definite Lyapunov function we can check the stability of the origin.

$$V = \frac{1}{2}(\frac{1}{s}e_1^2 + e_2^2 + 4e_3^2) \tag{34}$$

Taking the derivative of the Lyapunov function

$$\dot{V} = \frac{1}{s}e_1\dot{e}_1 + e_2\dot{e}_2 + 4e_3\dot{e}_3 \tag{35}$$

$$\dot{V} = -(e_1 - \frac{1}{2}e_2)^2 - \frac{3}{4}e_2^2 - 4be_3^2 \tag{36}$$

we can conclude the origin is globally asymptotically stable, and thus the system $[x_s \quad y_s \quad z_s]^T$ will synchronize with the system $[x \quad y \quad z]^T$. Using $s = 10$, $r = 28$ and $b = \frac{8}{3}$ the distance between the two systems is plotted in figure 4. The Lorenz system starts with initial conditions of [010] while the synchronous system as defined in equation 27-29 have an initial conditions of [123]

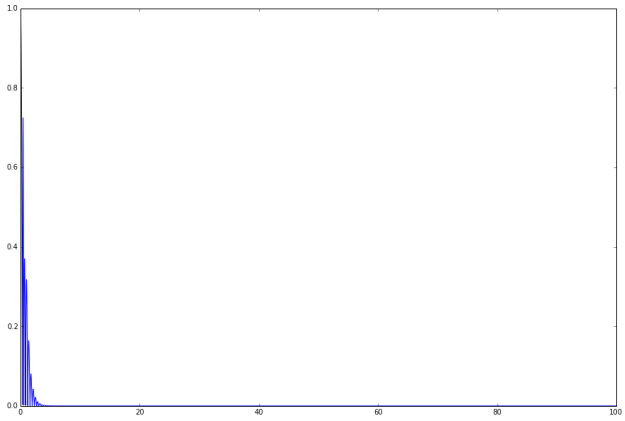

Figure 4: The distance between the two systems are plotted as a function of time

Figure 4 clearly shows how startling fast the systems synchronize even when there initial conditions are very far away.

4 Synchronous Choas Cryptogropher

Kevin Cuomo and Alan Oppenhiem showed that it is possible to get two full dimensional chaotic systems to synchronize with each other by using the output from one of the systems to drive the other. Cuomo and Oppenhiem simulated this effect using circuits and showed the capability of the method to encrypt messages. [2] Instead of using circuits, I wrote a simulation written in python to demonstrate the method.

Say you want to send your friend a song but you want to make sure that no one else can listen to the song. Tell your friend before hand the system parameters s, r and b making sure the values of the parameters create a chaotic system. The system parameters act as the key for the crypt.

Read your song into your program. Note that songs are merely just a list of numbers that describe the sound pressure as a function of time. For this demonstration, the classic "Call Me Maybe" by Carly Rae Jepsen is used. The waveform of the song is plotted below.

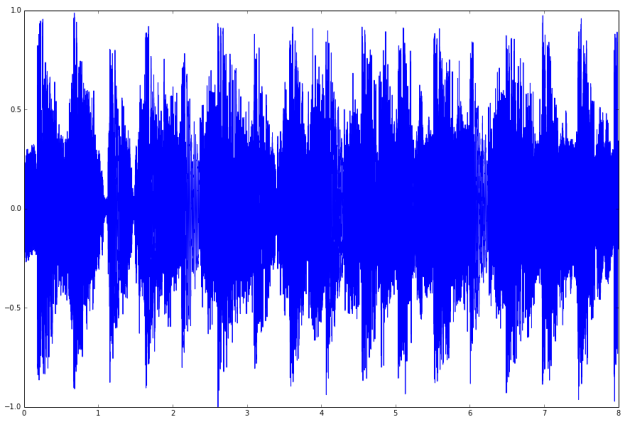

Figure 5: Call Me Maybe" by Carly Rae Jepsen

Then, solve a Lorenz system with the parameters agreed upon earlier. Build the message $m(t)$ as

$$m(t) = s(t) + x(t) \tag{37}$$

where $x(t)$ is the x solution of the Lorenz system and $s(t)$ is the song in which we want to encrypt. It is very important that $x(t)$ is much more powerful than $s(t)$. Since $x(t)$ is chaotic, the song is unrecognizable noise and thus safe for transit.

Your friend then receives the message. To decrypt the message, your friend should solve the synchronous system

$$\dot{x}_s = s(y_s - x_s) \tag{38}$$

$$\dot{y}_s = rm - y_s - mz_s \tag{39}$$

$$\dot{z}_s = my_s - bz_s \tag{40}$$

Since the systems are synchronous, $x_s \to x$. Thus to decrypt the message

$$s(t) \approx m(t) - x_s(t) \tag{41}$$

Figure 6 shows the decrypted song compared to the actual song. While some information is lost due to the encryption and decryption, it is clear that the method works. Playing the decrypted song, there is no noticeable difference between the two versions.

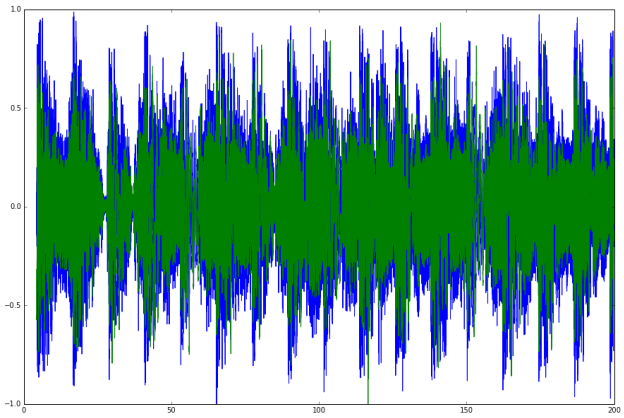

Figure 6: The full song (blue) and decyrpted song (green) are shown.

A natural question to ask is how sensitive is synchronization to the system parameters. In figure 7, the system parameters are changed by 30%. The figure shows that the systems did not synchronize, and playing the songs confirms the song is unrecognizable. Changing the system parameters to only 10% does not change the figure significantly, but the song becomes barely recognizable.

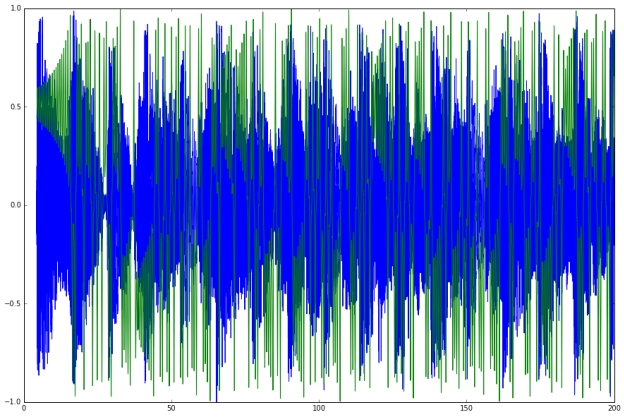

Figure 7: The full song (blue) and decyrpted song (green) are shown with parameters differing 30%

The theory and method of using chaos synchronization as described by Kevin Cuomo and Alan Oppenhiem to encrypt data is thus confirmed.

5 Code

```
#### Read in the Song and Plot and Play it ####

(sample_rate,song) = read("FavSong.wav")

tf = len(song)/sample_rate
numSteps = len(song)
dt = tf/numSteps

t = np.linspace(0,tf,numSteps)
```

```
fig, ax = plt.subplots();
fig.set_size_inches(15,10);
ax.plot(t,song)

!afplay FavSong.wav


#### Solve a Lorenz system and add x(t) to the message ####

x = np.zeros(numSteps)
y = np.zeros(numSteps)
z = np.zeros(numSteps)

x[0] = 0
y[0] = 1
z[0] = 0

tf = 200.
dt = tf/numSteps
t = np.linspace(0,tf,numSteps)

for i in range(numSteps-1):
x[i+1] = x[i] + (s * (y[i]-x[i]))*dt
y[i+1] = y[i] + (r*x[i] - y[i] - x[i]*z[i])*dt
z[i+1] = z[i] + (x[i]*y[i] - b*z[i])*dt
message = x + .001*song

write("Message.wav",sample_rate,message)
!afplay Message.wav

#### Using the message as an input, solve the synchronous system #####

x_s = np.zeros(numSteps)
y_s = np.zeros(numSteps)
z_s = np.zeros(numSteps)

x_s[0] = 1.
y_s[0] = 2.
z_s[0] = 3.

for i in range(numSteps-1):
x_s[i+1] = x_s[i] + (s * (y_s[i]-x_s[i]))*dt
y_s[i+1] = y_s[i] + (r*message[i] - y_s[i] - message[i]*z_s[i])*dt
z_s[i+1] = z_s[i] + (message[i]*y_s[i] - b*z_s[i])*dt


#### Plot, play and compare the decrypted song ####

ic_error_length = 8000

decrypt = message - x_s
decrypt = decrypt[ic_error_length:]
decrypt = decrypt/abs(decrypt).max()

fig, ax = plt.subplots();


fig.set_size_inches(15,10);
ax.plot(t[ic_error_length:],song[ic_error_length:],t[ic_error_length:],decrypt)
```

```
write("Decrypt.wav",sample_rate,decrypt)
!afplay Decrypt.wav
```

References

[1] Carroll, Thomas L., and Louis M. Pecora. "Synchronizing chaotic circuits." Circuits and Systems, IEEE Transactions on 38.4 (1991): 453-456.

[2] Cuomo, Kevin M., and Alan V. Oppenheim. "Circuit implementation of synchronized chaos with applications to communications." Physical review letters 71.1 (1993): 65-68.

[3] Cvitanovic, Predrag, et al. "Chaos: Classical and Quantum (Niels Bohr Institute, Copenhagen, 2008)." ChaosBook. org.