Acoustic Synchrony

Justin Blythe*
Georgia Tech Department of Physics
(Dated: December 14, 2012)

Many different biological systems experience mutual synchrony [2]. We have studied the particular case of synchrony amongst two crickets using an adapted integrate and fire model studied by Mirollo and Strogatz. By using a microphone and speaker coupling to model a cricket, we tested to see if the crickets would synchronize. I found conclusive data for mutual synchrony when running numerical simulations, but no global synchronization for live simulations.

Introduction

The reason that neighboring crickets tend to synchronize is because of their mating habits. A sexually responsive female cricket is attracted to the chirp rhythm and the male cricket who tends to chirp in the lead [1]. Thus the male crickets will slow down there chirp rate to synchronize with a preceding chirper or speed up it's chirp rate to synchronize with a quickly imitating chirp. A quantitative analysis of this phenomina was carried out by Thomas J. Walker in 1969 [5]. He found that a cricket chirping at a constant rate, which consisted of him holding a tape player, resulted in the neighboring crickets to synchronize with his recorded chirp.

We used this model to test both the biological accuracy of Stogatz's integrate-and-fire equation and to create an experimental procedure of this system. The problem is that no one (at least what I could find) has tried to electronically simulate crickets, outside of a numerical simulation. So we attempted to analize Mirollo and Strogatzs' model for oscillators. The equation that Strogatz and Mirollo gave was a spin-off of Peskin's model for self-synchronization of the cardiac pace-maker.

$$\frac{dx_i}{dt} = S_0 - \gamma x_i \tag{1}$$

Where x_i is the state of an oscillator with $x_i \in [0, 1]$, S_0 is a property of the oscillator in units of Hz, and γ is a dissipative term which gives the phase curve a concave down shape. Strogatz and Mirollo declared that this is a necessary part for the integrate-and-fire model to synchronize. The criterion for synchrony is as follows,

FIG. 1: A snowy tree cricket.

$$x_i = 1 \Rightarrow\ x_j(t^+) = \eta + x_j(t) \quad \forall j \neq i \tag{2}$$

where η is the associated coupling strength of the oscillator in question. This says that once one of the oscillators has fired the current state of *all* the other oscillators are shifted by η, bringing the states closer together. Note: Peskin's model put the "outside stimulus" parameter inside the integrand [3].

To follow Walker's model for synchrony we had to modify how this equation was implemented. Instead of global synchrony, we use a lead and follow model where the leader does not change it's chirp rate when the another cricket has chirped, but the follower will. To account for outside noise we added a time dependent noise term to the integrand. This is a common approach when dealing with oscillators [4]. Thus the resultant equation for our model is

$$\frac{dx_i}{dt} = S_0 - \gamma x_i + \zeta(t) \tag{3}$$

$$x_L = 1 \Rightarrow\ x_F(t^+) = \eta + x_F(t) \iff i = 2 \tag{4}$$

where $\zeta(t)$ is the external noise, x_L is the state of the lead chirper, and x_F is the state of the follower chirper.

Experimental Setup

To simulate live crickets we procured three sets of speakers and three microphones. Each pair of speakers and microphone was to be one of our crickets. The speakers that we used were Logitech S120 2.0 Multimedia Speakers and the microphone was a inexpensive $2 microphone whose brand name I cannot locate. In order to access multiple crickets simultaneously each microphone and speaker coupling was controlled by a StarTech.com 7.1 USB Audio Adapter External Sound Card. The latency for such a device is existent, but for our purposes and budget, this was the best option.

*Electronic address: jblythe29@gmail.com

FIG. 2: This is the speaker/microphone couplings that modeled our crickets.

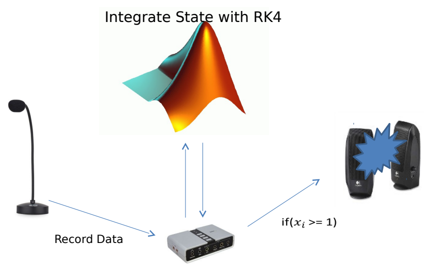

FIG. 3: This is a schematic of how our the communication between the sound card and MATLAB worked.

Once each sound card was plugged in and installed correctly we needed a way to control the input and output to each respective sound card *and* have a way to integrate the equation discussed above. This is where MATLAB 2011b came to the rescue. MATLAB has built-in functions that allow the user to designate the input and output to an object while also distinguishing between multiple sound cards. Using the object method of sound I/O in MATLAB, the function will not stop and wait until the sound card has finished executing the command (a new feature). This is optimal for our purposes because the state variable of each the respective cricket needs to be integrated constantly *and* it should be recording almost all of the time, which turned out to be the most difficult part of the setup.

In the function a while loop ran through the lines of code. See FIG. 3. At the beginning it would access the recorder object and tell it to start recording. In order to tell if the another cricket had chirped it would need to access the recorded sound (an array); if the max value was large enough it would then add η to the current state.

The problem with this is that the loop reads the code so fast that it would skip integration if it was still recording; adding a long pause is out of the question as well because we desired real-time integration. Accounting for this took many hours of fine tuning, which in the end resulted with the function only recording in short time intervals and a short pause after the recorder object was called. The function must also integrate the state if the recorder object was still recording, as required by real-time analysis. A similar issue arose when the player object was called, this resulted in yet another pause. I believe this function could have been optimized, but time was short and we wanted some data.

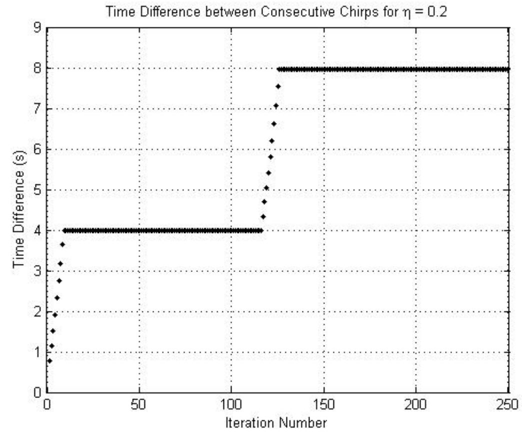

FIG. 4: Plot of the time difference between consecutive chirps with a numerical simulation.

Results

Before data collection began we needed a good set of parameter values that modeled a snowy tree cricket. This was done using different values of γ. S_0 is a characteristic parameter of an oscillator which, after solving Eq. 1 analytically, one can show is related to the period of an uncoupled oscillator.

$$\tau = \gamma^{-1} ln \left(\frac{S_0}{S_0 - \gamma} \right) \tag{5}$$

So by plugging in the value of γ and the known average chirp interval of a snowy tree cricket, which is 2 seconds, one will find that $S_0 \approx 0.5\, Hz$; for small dissipation values S_0 is approximately the natural frequency of the oscillator.

When we began to collect the data we wanted the chirp rate to be comparable to the natural frequency of a snowy tree cricket. When using the above S_0 of a system initially would not have worked because of the logic in the program that we had used. In order for the chirp rate to be 2 seconds apart the S_0 parameter was set to $\approx 12\, Hz$; each cricket having a slightly different S_0.

However, with $S_0 \approx 12Hz$ I the found behavior to be inconsistent with a numerical simulation. The behavior of two crickets in a leader and follower system was found to shift in and out of phase by 180° with a

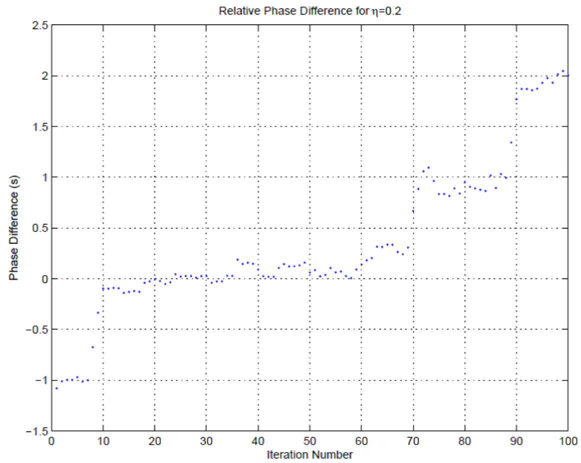

FIG. 5: Plot of the time difference between consecutive chirps with the live experiment. With $\gamma = 0.05$, $S_{0F} = 11.98Hz$, $S_{0L} = 12.01$, and $\eta = 0.3$. Here the crickets stay move in and out of phase by 180°.

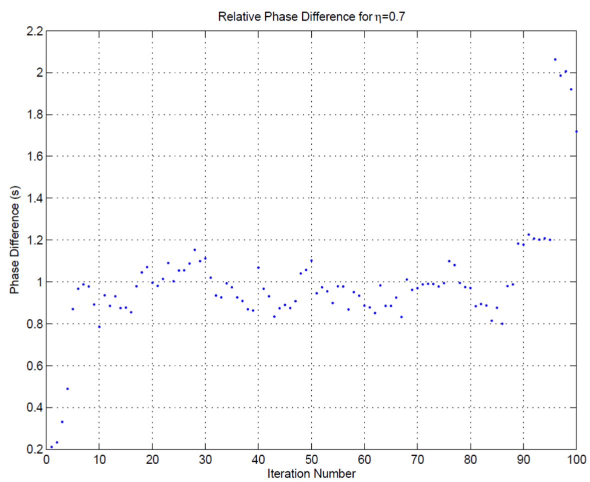

FIG. 6: Plot of the time difference between consecutive chirps with the live experiment. With $\gamma = 0.05$, $S_{0F} = 11.98$, $S_{0L} = 12.01$, and $\eta = 0.7$. Here the crickets are in a relatively constant phase lock 180°out of phase.

coupling strength of 0.2. See FIG. 5. This plot shows the iteration number versus the time difference between consecutive chirps. So the crickets started 180° out of phase, then synchronized for more than half of duration of data collection, proceeded out of phase by 180° again (presumably perturbed out by the noise parameter), and finished in sync with each other. Note: The natural chirp interval was 2 seconds, thus the phase difference in intervals of two corresponds with synchrony. In a separate run with a $\eta = 0.7$ we that the two crickets stayed in a relatively constant phase lock 180° out of phase almost the entire time. See FIG. 6.

In the numerical simulation (with $\zeta(t) \sim 10^{-3}$) I found that they would go in and out of synchrony. However,

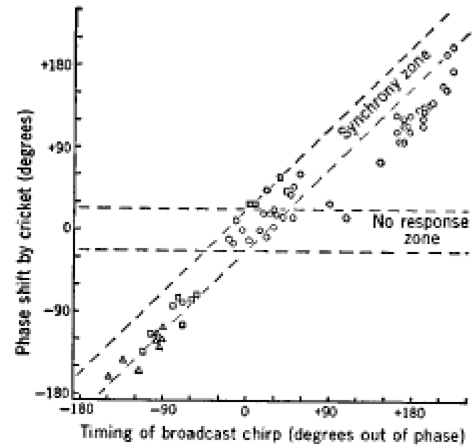

FIG. 7: Plot from Walker's 1969 journal article in Science [5]. This shows that a broadcast chirp (the leader) would only synchronize for certain phase shifts from follower and initial out of phase chirps from the leader.

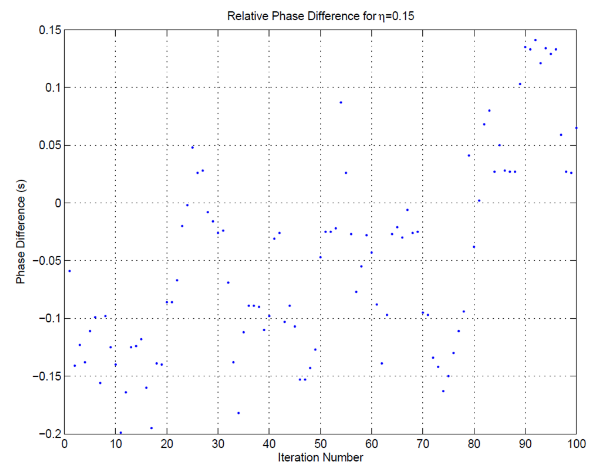

FIG. 8: This plot shows a run of our live simulation that led to synchrony. Here $\eta = 0.15$

instead of stopping for a short lived phase lock when 180° out of phase it would keep adding to their phase until they reached synchrony once again, and they would stay there until perturbed out of this state by $\zeta(t)$. See FIG. 4.

Discussion

Our ultimate goal was to reproduce Walker's model. He found that the leader/follower method would synchronize for certain values of phase difference and the responding crickets phase shift. See FIG. 7. Due to complications with the experiment and limited time we were not able to reproduce these results. What we could have done is have the chirp not only increment by η, but also decrement. This would have been possible by running all of the I/O devices through a single MATLAB function simultaneously, instead of using separate MATLAB windows to operate each cricket individually.

The most interesting outcome with the flawed model we used is that the crickets would still synchronize . The duration of synchrony did not last for long most of the time, but for certain values it stayed relatively in sync the entire duration of data collection. See FIG. 8. If you look carefully the plot has a very narrow axis with maximum values between -0.2 and 0.15 seconds. The human ear would not be able to distinguish between the beginning of one and the end of the other. This tendency was quite rare but still existent for certain values of η and S_0. The implications of which show that our model is good but the application of it in the live simulation needs tweaked.

Our model also implies that the additive time dependent noise parameter affects the synchrony of the pulse-coupled oscillators but they will always come back to a state of synchrony with the follower cricket ahead of the lead crickets by a full chirp interval. Which is synchrony with phase differences of 2π. If time was not a factor, it would have been easy to show that decrements in η could possibly keep them in a constant state of synchrony.

[1] Michael D. Greenfield and Igor Roizen. Katydid synchronous chorusing is an evolutionarily stable outcome of female choice. *Nature*, 364:618–620, 1993.

[2] Renato E. Mirollo and Steven H. Strogatz. Synchronization of pulse-coupled biological oscillators. *SIAM Journal of Applied Mathematics*, 50:1645–1662, 1990.

[3] C. E. Peskin. *Mathematical Aspects of Heart Physiology.* New York University, 1975.

[4] P. Tass, M. G. Rosenblum, J. Weule, J. Kurths, A. Pikovsky, J. Volkmann, A. Schnitzler, and H.-J. Freund. Detection of n: m phase locking from noisy data: Application to magnetoencephalography. *Physics Review Letters*, 81:3291–3294, 1998.

[5] Thomas J. Walker. Acoustic synchrony: Two mechanisms in the snowy tree cricket. *Science*, 166:891–894, 1969.