Simulation of coupled nonlinear time-delayed feedback loops using state-space representation

Author:

Karl Schmitt[1,2]
krbschmitt_at_math.umd.edu

Advisors:

Jim Yorke[1,2,3]
yorke_at_umd.edu

Rajarshi Roy [4,2,3]
lasynch_at_gmail.com

Tom Murphy[5,3]
tem_at_umd.edu

[1] Department of Mathematics, University of Maryland
[2] Institute for Research in Energy and Applied Physics (IREAP)
[3] Institute for Physics and Science Technology (IPST)
[4] Department of Physics, University of Maryland
[5] Department of Electrical and Computer Engineering, University of Maryland

Abstract:

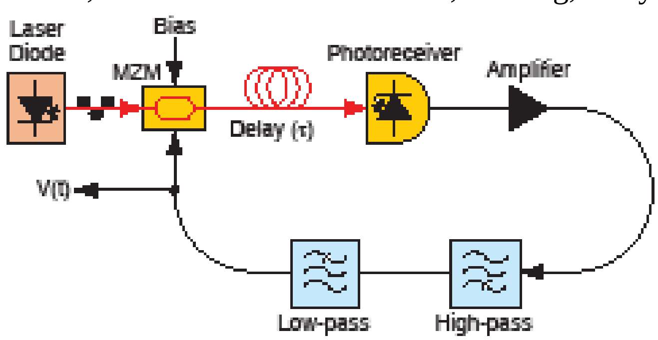
A discrete time model for single and coupled nonlinear time-delayed Mach-Zehnder feedback loops is developed and validated. The model is used to investigate synchronization regimes related to feedback strength, time delay and optical biasing for mutually coupled loops. A comparison between simulation and experimental results is presented for synchronization regimes, and the model is used to identify parameter mismatches in the experimental system. Identification of mismatches and appropriate adjustment of synchronization regimes is discussed in context of applications to secure communications and sensor networks.

Introduction

The simulation of accurate models for experimental systems is vital to determining future research and validating existing research. In particular, modeling a system of coupled nonlinear time-delayed feedback loops provides a number of unique avenues for numerical investigation. Such systems are under heavy study in efforts to develop new methods of secure communication and sensing. Of particular interest to this field is the nonlinear system involving Mach-Zehnder interferometers as all of its components are readily available commercially. Published models for this system have been developed using continuous time, and model reasonably the behavior of the system. In our research however, development of the system has progressed in a manner that suggests implementing a physically discrete system. This progression motivates a change in the models for the system as well. The model for each independent loop will be a discrete time, state-space representation of the loop in Kouomuo [1] and will be tested against published results for identical systems. Coupling schemes will be initially tested on well known and previously explored systems such as the Lorenz model [2]. The final implementation will be tested against published experimental data for such a coupled system [3,4]. Since research into implementation as either a communication or sensor device has shown a significant dependence on the accuracy of synchronization which is further dependent on the quality of parameter matching, this will explored in detail. We will identify via simulation existing parameter mismatching in an experimental system.

Background

The nonlinear time-delayed feedback system explored in detail by Kouomuo [1] is comprised of a laser, Mach-Zehnder inferometer, filtering, delay and amplification.



Through basic mathematical representations for each of these components one can form a model for the evolution of the system in terms of a time-delayed integro-differential equation as defined in Kouomuo:

$$x(t) + \tau \frac{d}{dt} x(t) + \frac{1}{\theta} \int_{t_o}^{t} x(s) ds = \beta \cos^2 [x(t-T) + \phi]$$

Here x(t) is a dimensionless variable with parameters of the normalized feedback gain β, the normalized bias offset ϕ, the high cut-off filter time constant τand the low cut-off filter time constant θ.

The generally established method for solving these equations would be traditional numerical methods such as RK4. However, one can examine the initial situation and formulate these equations using a completely different approach (presented below). Having established a basic nonlinear system, we can now examine more complicated behavior.

It has been observed both in natural systems and mathematical models that two nonlinear systems can achieve a synchronous state when coupled in an appropriate manner. Understanding such systems may lead to better communication techniques, advanced medical procedures and a significant improvement in understanding certain biological systems [6, 9, 10].

With either formulation it is fairly easy to cast this in the form of many published pieces of work about coupling systems of nonlinear equations. What becomes interesting is examining the behavior of such coupled systems. In published work on the Lorenz system it has been demonstrated that two such coupled systems can be made to synchronize. This seems counter-intuitive to the concept of nonlinear (chaotic) systems and has therefore sparked a variety of research. Of specific relevance to this project is published experimental work which has demonstrated that given the correct setup it is possible to achieve synchronization between two Mach-Zehnder loops.

Derivation of Alternative Model

The approach taken by Kouomuo was to model the filters using single-pole low-pass and high-pass filters. An alternative approach is to formulate them in state-space. Then the filtering would look like:

$$\dot{\mathbf{u}}(t) = \mathbf{A}\mathbf{u}(t) + \mathbf{B}x(t)$$

$$y(t) = \mathbf{C}\mathbf{u}(t) + Dx(t)$$

Here x(t) represents the input to a filter, y(t) is the output from the filter and A, B, C and D are constant matrices related to the filter used. Furthermore this can be easily converted to a discrete map equation. This is highly appropriate if one is considering a discrete-time filter such as might be implemented on a digital signal processing board. Since the experimentalists related to this project have chosen to implement the system in this manner we will use the discrete versions as follows[5]:

$$\mathbf{u}[n+1] = \mathbf{A}\mathbf{u}[n] + \mathbf{B}x[n]$$

$$y[n] = \mathbf{C}\mathbf{u}[n] + Dx[n]$$

Now we must include the concept of feedback. The simplest approach would be just a direct feedback where x[n]=y[n]. This, however, does not allow any dynamics besides the filter response to occur. Therefore we also include some function applied to the output of the filter, thus you could imagine something like:

$$x[n] = f(y[n-k])$$

Here is included the fact that we are using time-delayed feedback as represented by the argument [n-k]. This gives rise to a state-space representation that looks like:

$$\mathbf{u}[n+1] = \mathbf{A}\mathbf{u}[n] + \mathbf{B}f(y[n-k])$$

$$y[n] = \mathbf{C}\mathbf{u}[n] + Df(y[n-k])$$

By carefully choosing our state-space to be the canonical form derived from the z-transform of the discrete time filters we are interested in modeling, we can rewrite the top equation in terms of only the state-vector u, and generate our output at a later iteration via the simplified second equation. This gives us an iterative map in the following form:

$$\mathbf{u}[n+1] = \mathbf{Au}[n] + \mathbf{B}f(\mathbf{Cu}[n-k])$$

The final step in realizing what will be implemented is to actually introduce the function f(y[n]) from the system. In our case it is the exact same nonlinearity introduced in the Kouomuo paper, since it represented the gain amplification to and the feedback of the Mach-Zehnder modulator, just as our function does. So, the final equation we will model is:

$$\mathbf{u}[n+1] = \mathbf{Au}[n] + \mathbf{B}\beta\cos^2(\mathbf{Cu}[n-k] + \phi)$$

The main drawback to this approach for modeling the system is it requires knowledge of the matrices A, B, C and D related to the filter. There exists code to generate these matrices for some standard filter types and orders in Matlab, but any given high or low pass filter will not necessarily well conform to these standards. While it's possible to buy high-caliber filters designed to specific functions, these are very expensive, an unattractive solution with a goal of diverse application implementations. An alternative approach is to implement digital filters. As mentioned before, this is the approach taken in our current experiments. This allows the actual implementation of filters that precisely match the matrices generated (or to design a filter and then generate the matrices that match it exactly). There exists some concern for the numerical stability of the matrices, but the code in Matlab asserts that these matrices are the most stable of available methods for generating filtering characteristics. Therefore we will largely ignore any concern for stability from this issue. A different issue could arise in the discretization of the continuous time system to a discrete time system but, since the discretization is already inherent in the system we seek to model, it can be ignored initially. There is some issue from the combination of both digital and analog system components, which will be addressed by introducing linear interpolation as needed.

The second concern is developing an effective method for coupling two of these systems. A bi-directional coupled Lorenz system as described by Anishchenko [2] is:

$$\begin{aligned} \dot{x}_1 &= \sigma(y_1 - x_1) + \gamma(x_2 - x_1) & \dot{x}_2 &= \sigma(y_2 - x_2) + \gamma(x_1 - x_2) \\ \dot{y}_1 &= r_1 x_1 - x_1 z_1 - y_1 & \dot{y}_2 &= r_2 x_2 - x_2 z_2 - y_2 \\ \dot{z}_1 &= x_1 y_1 - z_1 b & \dot{z}_2 &= x_2 y_2 - z_2 b \end{aligned}$$

This is considered diffusive coupling in the literature. This same technique can be applied to our state-space representation. If we take a step back and consider where we have both an input and output term (x[n] and y[n]), it would make logical sense to couple in the input terms. That is we will introduce coupling in the x[n] term. However, recall that we've replaced the x[n] term with our f(y[n-k]) term, so, our coupling would then look like:

$$\begin{aligned} \mathbf{u}_1[n+1] &= \mathbf{Au}_1[n] + \mathbf{B}f(y_1[n-k]) + \mathbf{B}\gamma(f(y_2[n-k]) - f(y_1[n-k])) \\ y_1[n] &= \mathbf{Cu}_1[n] + Df(y_1[n-k]) \\ \mathbf{u}_2[n+1] &= \mathbf{Au}_2[n] + \mathbf{B}f(y_2[n-k]) + \mathbf{B}\gamma(f(y_1[n-k]) - f(y_2[n-k])) \\ y_2[n] &= \mathbf{Cu}_2[n] + Df(y_2[n-k]) \end{aligned}$$

Now we perform the same simplifications that we did earlier, as well as multiplying out the coupling term and recombining them giving us a simplified pair of equations:

$$\mathbf{u}_1[n+1] = \mathbf{Au}_1[n] + \mathbf{B}\beta\{(1-\gamma)\cos^2(\mathbf{Cu}_1[n-k] + \phi) + \gamma\cos^2(\mathbf{Cu}_2[n-k] + \phi)\}$$

$$\mathbf{u}_2[n+1] = \mathbf{A}\mathbf{u}_2[n] + \mathbf{B}\beta\{(1-\gamma)\cos^2(\mathbf{C}\mathbf{u}_2[n-k]+\phi) + \gamma\cos^2(\mathbf{C}\mathbf{u}_1[n-k]+\phi)\}$$

This is the final set of equations we will implement to actually model a coupled set of Mach-Zehnder loops.

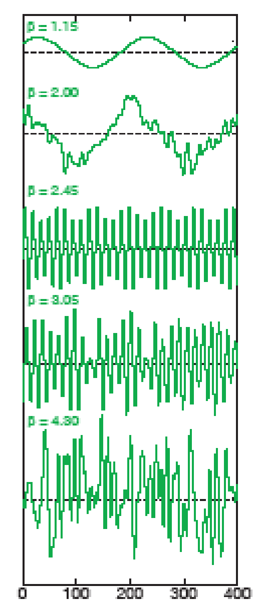
Implementation

Because the majority of published materials for this class of problems contain graphical representations and a primary experimental observation method is the display of time traces, it will be important to facilitate comparisons between simulation runs and this visual data. This suggests using a language or environment that incorporates an easily utilized graphical presentation component. Furthermore since the chosen method of implementation is dependant on matrix filtering constants, a language which has such code readily available or integrated for calculation of these coefficients would be preferred. To fulfill these requirements the primary implementation will be performed in Matlab with integrated C routines as needed for efficient calculations.

The largest predicted concern will be in comparison between published experimental results due to the quantization inherent in measurements. This quantization is not existent in the mathematical model without being explicitly included. Since there does exist characteristics that are dominant on scales significantly above the quantization error, for validation of the code this can be ignored.

Validation

The simulation development will took part in three stages, each independently verifiable. The first will involved implementing a single loop model as developed above. This was verified against published work by Kouomuo et al. [1] on such systems. Specifically characteristic behavior of the system at unique parameter settings is identified and compared. Four such characteristic curves are displayed to the right, with their corresponding system parameters.



The second stage was a separate implementation of a system of couple Lorenz models [2]. Again, characteristic behavior was looked for. Using commonly studied parameters of the system (σ=10, r1=28.8, r2=28, b=8/3) identical synchronization is demonstrated.

The final stage of implementation was a combination of the previously mentioned models. To verify this, a comparison is made against two sets of literature. Argyris et. al. [3] has published work where a set of oscillators coupled in an open loop configuration (γ=0 for system 1 and γ= 1 for system 2) synchronize and exhibit unique behaviors. Further, in a slightly more complicated case Piel et al. have demonstrated synchronization under very specific circumstances which involve bi-directional communication [4]. Synchronization is demonstrated under these specific conditions of γ=0.5, and conversely the lack of synchronization when these conditions are not met.

Results of Validation

Single Mach-Zehnder Non-Linear Time-Delayed Optical Feedback Loop

The first step is to verify against the analytical results published by Kouomou. He identifies the control parameter $\gamma_k = \beta \sin(2\varphi)$, and through analysis of the continuous time equation derives solutions for bifurcation points, and the frequencies that should appear at these bifurcations. We can calculate bifurcations according to:

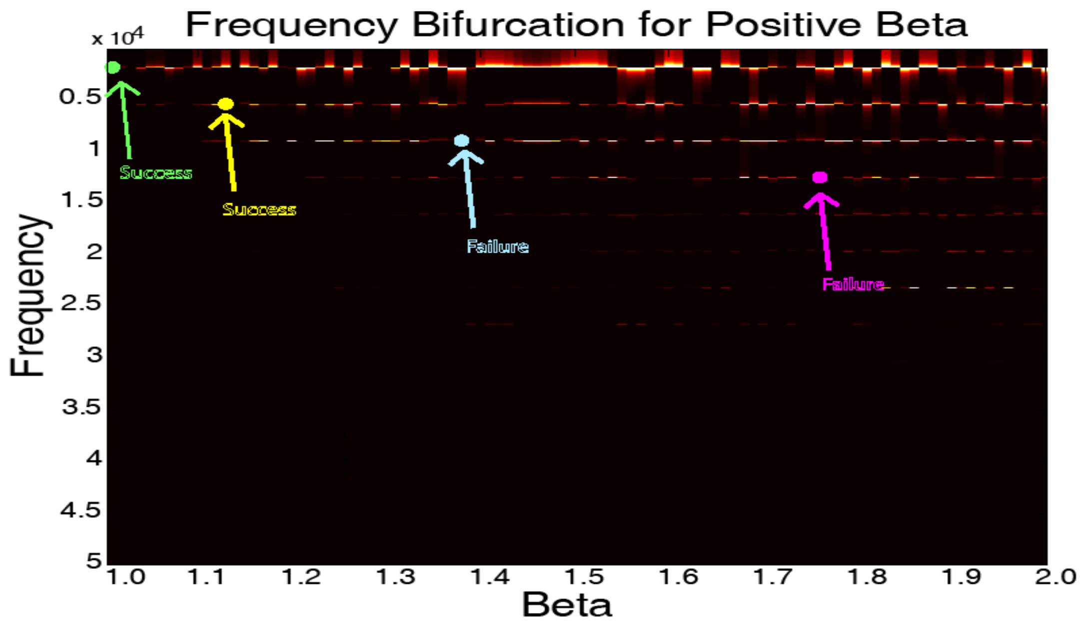
$$\gamma_k = (-1)^{k+1}[1 + \frac{(\varepsilon R^2 - k^2\pi^2)^2}{2k^2\pi^2 R^2}] \text{ and } \omega_k = k\frac{\pi}{R}$$

Where $R = \frac{T}{\tau}$ and $\varepsilon = \frac{\tau}{\theta}$ from the variables defined above. Additionally, the calculations for k=0 are slightly different giving the solutions:

$$\gamma_0 = -1 - \varepsilon R / 2 \text{ and } \omega_0 = \sqrt{\varepsilon / R}$$

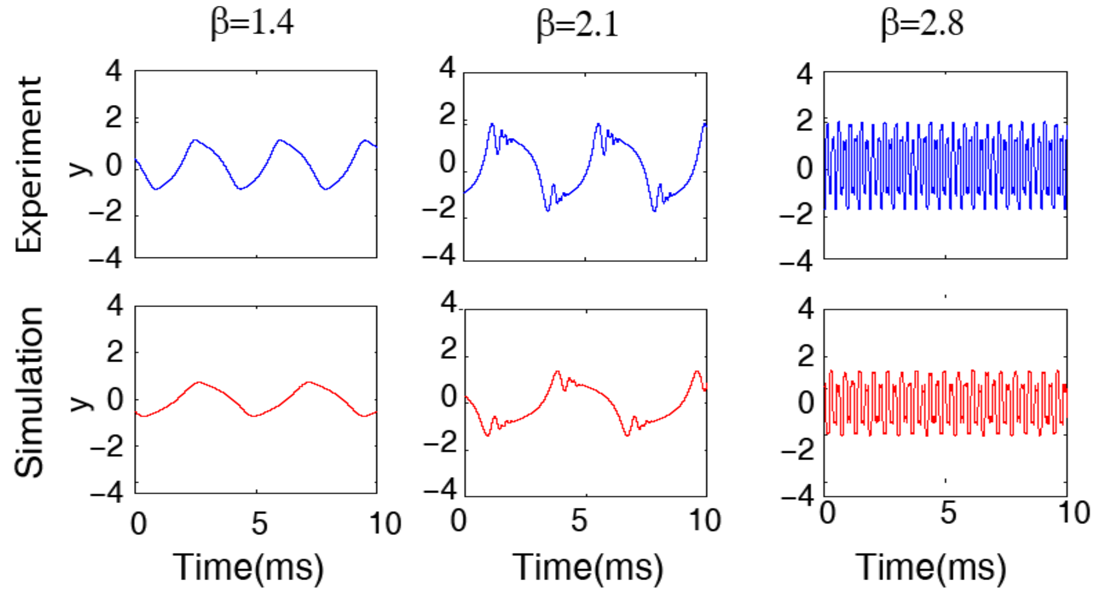
It is worth noting that these are only approximate solutions, so exact agreement to experimental or simulated results may not occur.

We can proceed to calculate the first few bifurcation points, as well as simulate the single loop system to compare. Because the bifurcations using a positive control parameter do not exhibit extremely unusual behavior it is easier to observe them, so we do not present the results for negative β though those results also show some correspondence. Below is a graph that includes the first four positive bifurcation points, plotted according to both their expected frequency and β value.

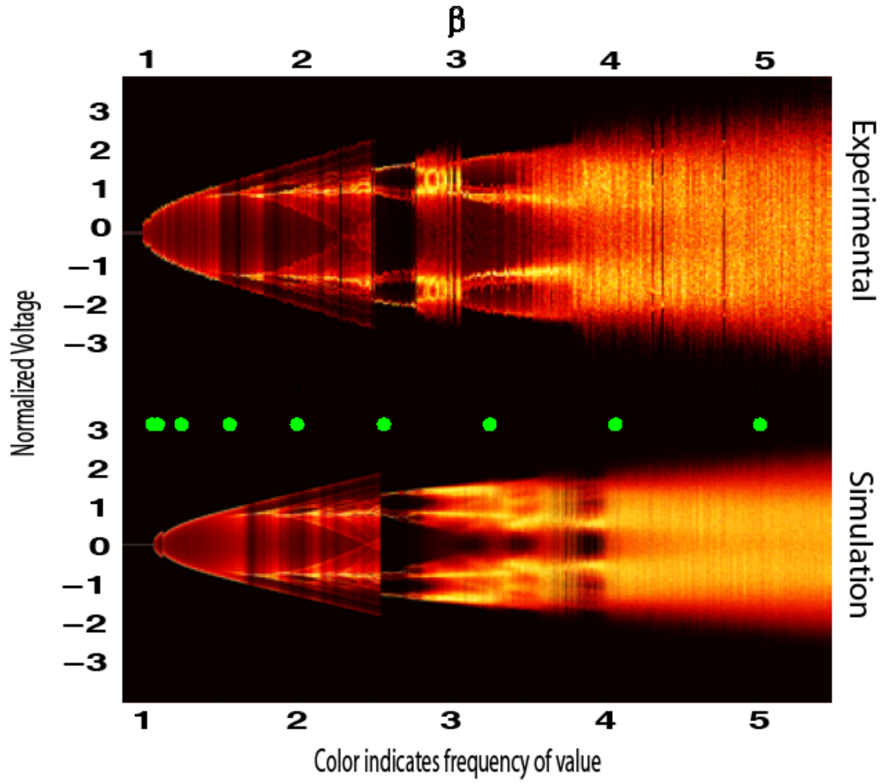


Marked on the graph are two successes and two failures of the simulations to match the analytical results. For higher values of β we see similar results to the 3rd and 4th points where the correct frequency is predicted but generally to the right (a higher β value) than the analytical results predict.

However, examining approximate analytical results do not always provide the information we seek. Since this code aims to simulate a physical system it is equally important to compare the simulation to experimental results. The simplest, most easily understood comparison is between time series that are generated in both the simulation and experiment.



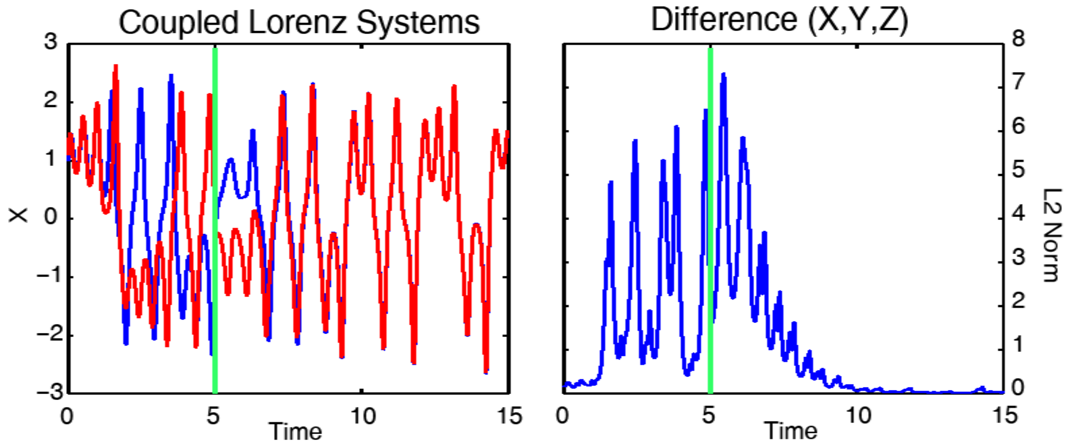
Here we can see that the simulated time series exhibit nearly identical behavior as the experimental data. The differences are a slight amplitude difference (simulation is ~80% of experimental) and a slight frequency mismatch. The amplitude difference is likely caused by an incorrect scaling factor in converting the experimental data to displayable data. However, the frequency mismatch is more of a concern, and potentially deserves further examination. The likely cause of it though is additional filtering occurring in the physical system (from various electronic parts such as the digital signal processing board and photo-detector) that is unaccounted for in the model (we are only modeling the directly implemented filtering). Ignoring these slight mismatches, the model does give very good qualitative agreement (and nearly quantitative) for most simulations. We can see this by looking at a very large spectrum of β values and taking the histogram of the time series. This provides what might be considered a 'value' bifurcation diagram:



Here there is a clear discrepancy at β~3, which is still under investigation between simulation and experiment. Possibilities are stray behavior in the experiment, histrionic behavior, or a significant failure of the model. However, given its ability to accurately reproduce a significant portion of the experimental data, the simulation can still be qualified as an over-all success. Also indicated on this diagram are the predicted β values for bifurcations. Worth noting is that many of the calculated bifurcation points to not align with any visible system alterations on either the experimental or simulated plots.

Synchronization in Coupled Lorenz Equations

The second stage of validation consisted of implementing the coupled Lorenz equations used as a model-basis for the coupling of the Mach-Zehnder equations (see above). The equations can be integrated using basic MATLAB ODE solvers. When we integrate these couple equations we find time series similar to the following:



The second plot shows the L2 norm of X, Y, and Z for the time trace shown. We can see that with the simple coupling outlined in the preceding sections it is possible to achieve isochronal synchronization. This of course has been demonstrated many times, but does encourage us that a similar implementation of the equations for Mach-Zehnder loops could work as well.

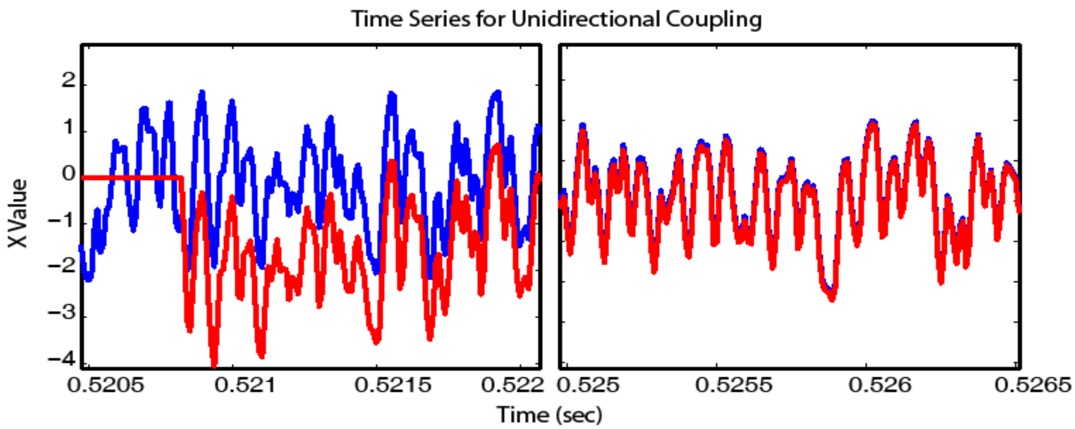
Synchronization of Couple Nonlinear Time-Delayed Feedback loops

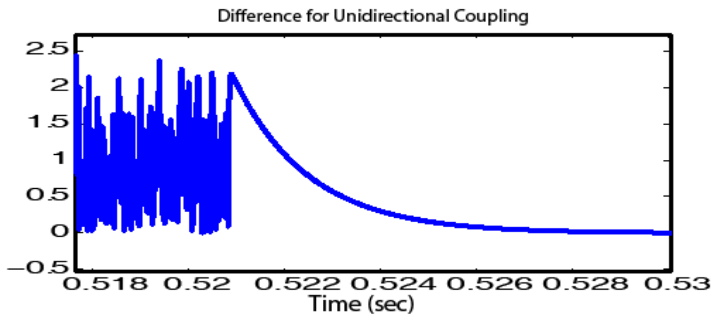
We previously developed equations for coupled Mach-Zehnder loops, and now we seek to simulate them, attempting to find regimes under which synchronization can occur. Previously published in literature are results for synchronization of a master loop to an open loop as well as at the specific γ value of .5 (50%) [3, 4]. To replicate these experiments the code has been implemented to allow individual specification of γ for each system, and for each interaction between systems. Specifically for reference we re-define the equations in the following way:

$$\mathbf{u}_1[n+1] = \mathbf{A}\mathbf{u}_1[n] + \mathbf{B}\beta\{\gamma_{11}\cos^2(\mathbf{C}\mathbf{u}_1[n-k]+\phi) + \gamma_{12}\cos^2(\mathbf{C}\mathbf{u}_2[n-k]+\phi)\}$$

$$\mathbf{u}_2[n+1] = \mathbf{A}\mathbf{u}_2[n] + \mathbf{B}\beta\{\gamma_{22}\cos^2(\mathbf{C}\mathbf{u}_2[n-k]+\phi) + \gamma_{21}\cos^2(\mathbf{C}\mathbf{u}_1[n-k]+\phi)\}$$

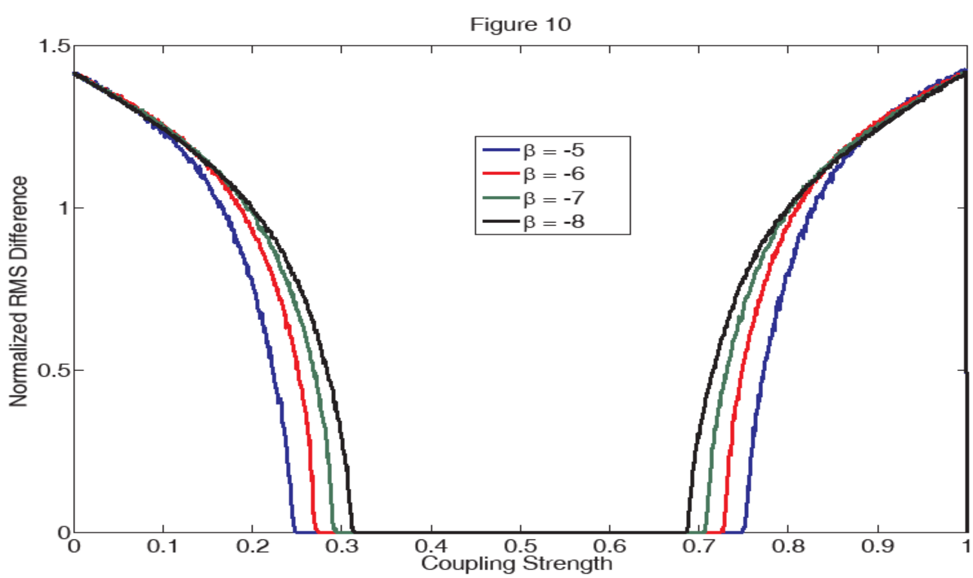
One can see then that the initially defined equations are just a special sub-set of these where: $\gamma_{11} = \gamma_{22} = 1 - \gamma_{12}$. Argysis has demonstrated (and used) synchronization under a very specific regime of these were $\gamma_{11} = \gamma_{21} = 1$ and $\gamma_{12} = \gamma_{22} = 0$. That is, open-loop, unidirectional coupling. He explores this under a variety of coupling delays and conditions which have been confirmed, but for the simplest case of the delay between systems being zero we do not need to dramatically change the above equations (merely impose those conditions) and we generate time series that look very similar in behavior to the coupled Lorenz equations.



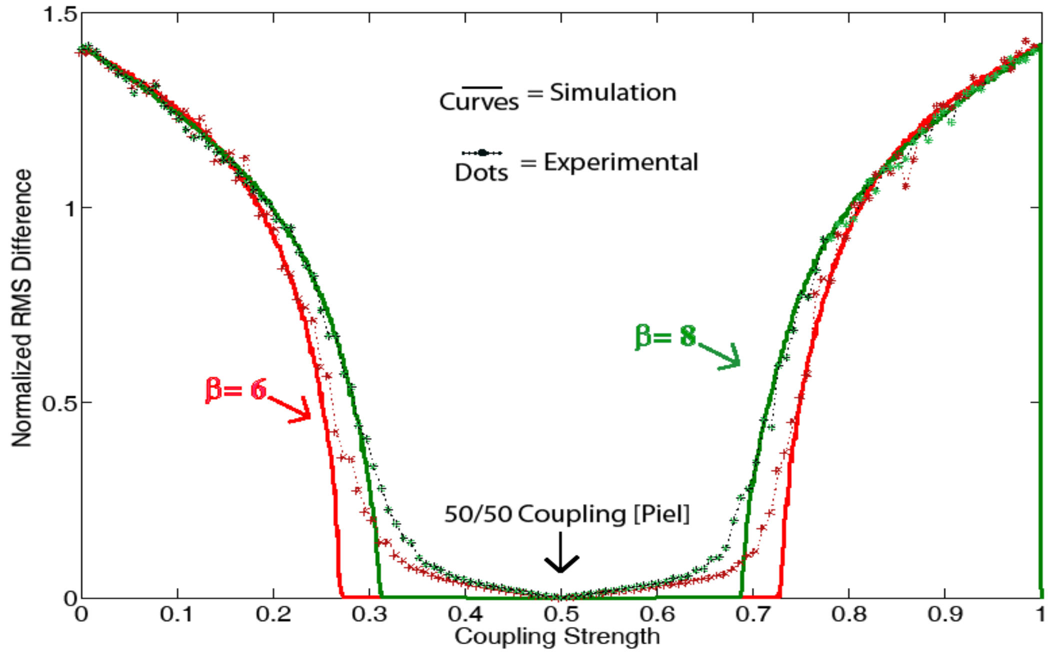


Here we can see that the two systems synchronize identically soon after the coupling has been turned on. It is also worth noting that rather than finding a mutual synchronization state, the second system actually synchronizes to the first one since the first system is actually a master/driver system and the second a slave.

We can also replicate Piel's experiments of mutually coupled systems synchronizing with a coupling of 50%, however, rather than show a number of plots related to this we can actually go ahead and take the simulations a step further and explore whether synchronization occurs for a variety of coupling strengths:



Here we see plotted on the y-axis the normalized RMS difference (averaged over 20 runs) between the last 10^4 entries of a time series for two mutually coupled systems. The x-axis indicates for what coupling strength this RMS difference occurred. We can see on this plot that in-fact identical synchronization occurs not only at 50% coupling, but at a wider variety of coupling strengths. To be certain that this is a real phenomena and not just artifacts of simulation we finally compare our synchronization results to experimental traces of the same thing which were included in the previous graph.

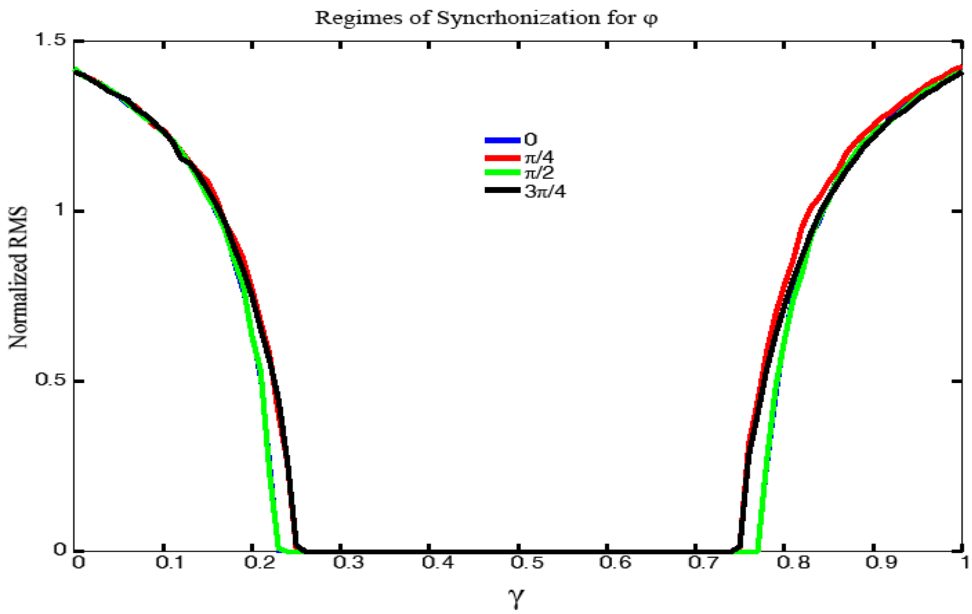


There we see that for two experimental systems at 50% coupling identical synchronization does occur while only nearly identical synchronization occurs at other locations. However the locations of drastic change from semi-synchronized behavior and completely unsynchronized behavior do match up. This suggests that expanding the code for parameter mismatching should (and does) bring the simulation into greater agreement with the experiments.

Alternative Synchronization Regimes

So far we have presented results for the synchronization regime of the model based solely on variations in β. However, β is not the only parameter that affects the dynamics of the system. Both *k* and φ are integral in determining the final behavior of the model. Therefore it is worth spending time to investigate the synchronization regimes of these variables as well. This investigation provides fairly straight forward results. We can see a similar trend to β, that is decreasing γ range, for an increase in *k*. However, the system is significantly less sensitive to changes of *k* suggesting that perhaps the discrete time system can be considered close enough to the continuous time to be 'infinite' dimensional. When we examine the synchronization regime for variations in φ we find a very different behavior. Here we see the periodic nature of $\cos^2$ reflected. The smallest regime corresponds to a biasing about the most sensitive points of $\cos^2$, $\frac{\pi}{4}+m\frac{\pi}{2}$, while the largest are related to the least sensitive sections of $\cos^2$, $m\frac{\pi}{2}$, where *m* is an integer. This periodic shifting of the regimes is shown below.

Figure 9



Final Improvements of Code

In light of the experimental data it became important to expand the code in a number of ways. The first was to allow individual specification of parameters for both systems. In terms of the equations, we've essentially introduced subscripts onto many of the important system parameters and included independent variables in the code to accommodate these. Specifically in addition to the expansion introduced in the validation section we now define our system in the following way:

$$\mathbf{u}_1[n+1]=\mathbf{A}\mathbf{u}_1[n]+\mathbf{B}\beta_1\{\gamma_{11}\cos^2(\mathbf{C}\mathbf{u}_1[n-k_{11}]+\phi_1)+\gamma_{12}\cos^2(\mathbf{C}\mathbf{u}_2[n-k_{12}]+\phi_2)\}$$
$$\mathbf{u}_2[n+1]=\mathbf{A}\mathbf{u}_2[n]+\mathbf{B}\beta_2\{\gamma_{22}\cos^2(\mathbf{C}\mathbf{u}_2[n-k_{22}]+\phi_2)+\gamma_{21}\cos^2(\mathbf{C}\mathbf{u}_1[n-k_{21}]+\phi_1)\}$$

A second improvement that was required in the code was the ability to do interpolation between points in the history. This is important in investigating the error between simulation and experiment because it turns out that while the digital signal processing board implements discrete time filters there is still an analog component of propagation both through the board and through the system which can introduce a non-integer delay. To account for this we introduce an intermediate step on each iteration of calculating the ‘real’ values of **u**[n-*k*]. For our purposes it is sufficient to simply introduce a linear interpolation scheme. More complicated schemes were tested but did not provide significant improvement.

Investigation of Simulation vs. Experiment

Investigating the difference between the experimental and simulation results is important for understanding whether there is an error in the model, mistakes in the experiment, or just something truly interesting going on. To this extent beyond analysis of basic synchronization regimes is has become important to investigate potential mismatches in the system parameters. Most of the analysis that finds synchronization is based on the concept that two non-linear systems can be matched exactly. This however is not the case for real systems. Once the code was expanded as above it became possible to investigate what a small mismatch in parameters might introduce in terms of synchronization error. Just as we introduced subscripts on (β, φ, k) we can investigate mismatches in each of these parameters.

In our numerical experiments with variation we introduced both negative and positive mismatches. The means that we chose a base value for the parameter and decreased (negative) or increased (positive) the parameter from that point. In the shown cases $\beta_1=\beta_2=6$, $k_{11}=k_{22}=k_{12}=k_{21}=22$, and $\varphi_1=\varphi_2=\frac{\pi}{4}$.

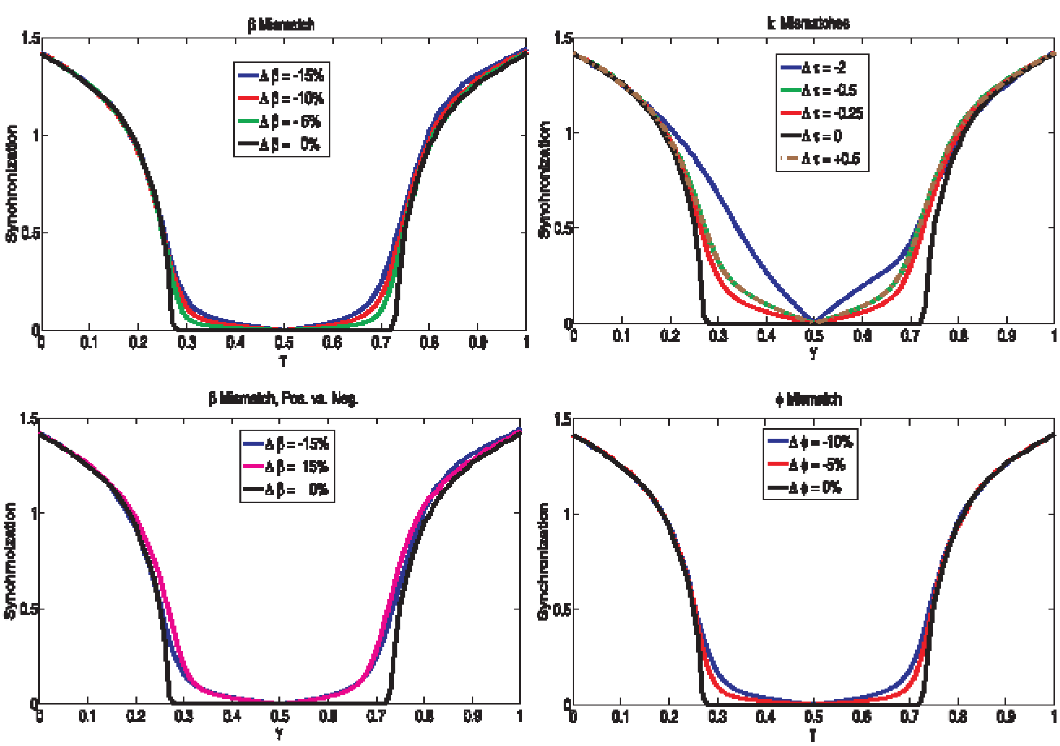
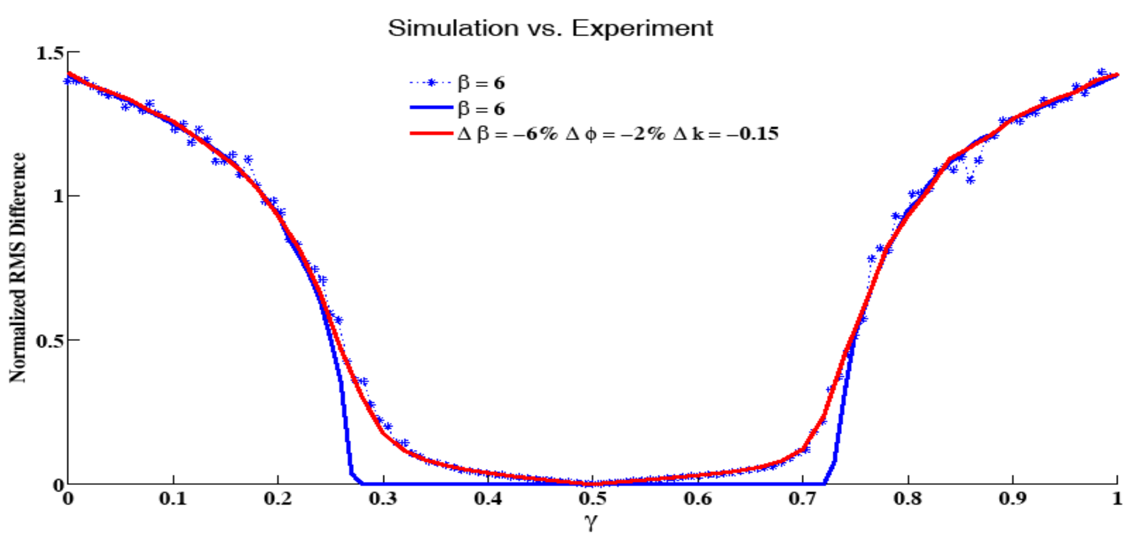


Figure 10a & 10b show various mismatches in β. Here we notice that RMS difference between systems grows steadily with increasing mismatch. There is a slightly greater difference for positive vs. negative β. This trait certainly fits with our finding above. Recall that as β increased the synchronization regime decreased therefore when we introduce a positive mismatch one system has a larger β and therefore an inherently smaller regime.

For variations in *k*, as shown in Figure 10c, we see that for a small (<10%) mismatch the region of synchronization disappears except for the 50% coupling location. The desynchronization is also symmetric as regards either a positive or negative variance (note the dashed, brown line). This likely stems from the fact that the general synchronization regime changes very little (if noticeable at all) over a delay change of 1. Further examining the smaller mismatches we note that while a very large change in *k* did not significantly affect the general synchronization regime here we see that even a .1 difference between systems causes a large shift towards desynchronization. With a base delay of 22, this .1 difference represents a mismatch of only 0.45%. This raises a fairly large concern since in the discrete digital system only integer time steps can effectively implemented. Fractional delay differences would require either significantly more complex filtering or unrealistic lengths of physical cabling.

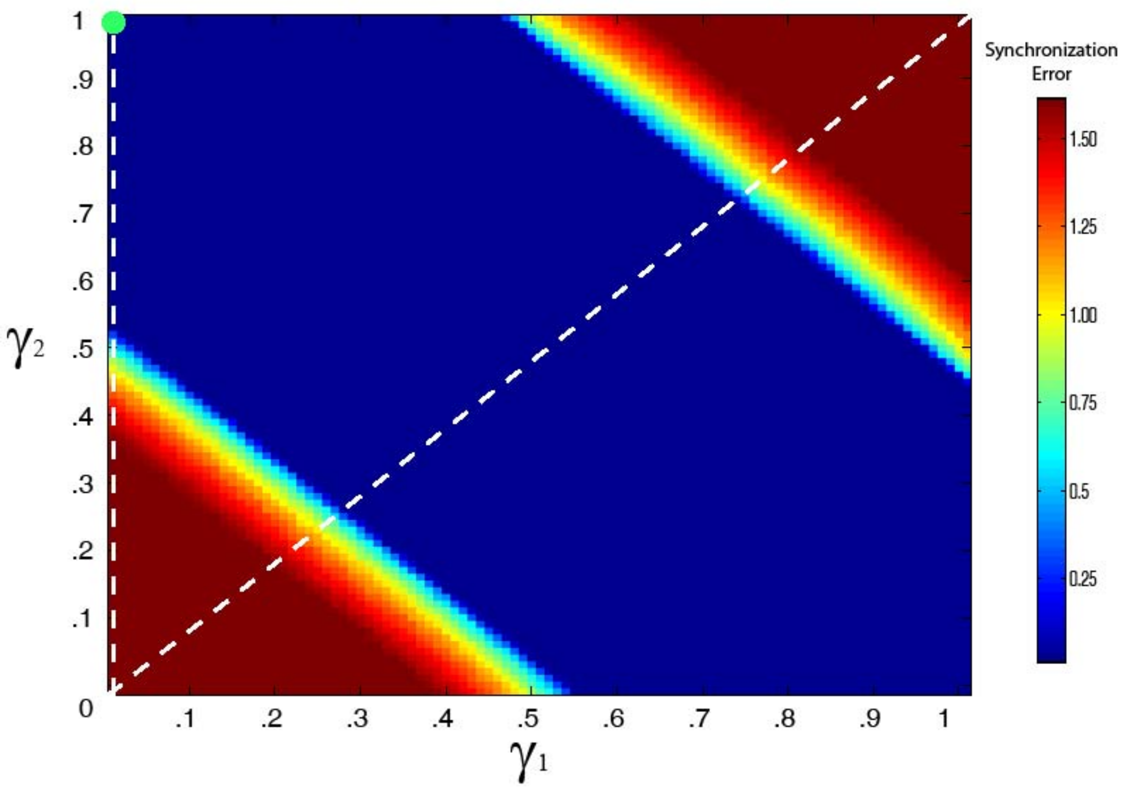
Lastly, shown in Fig 10d, for variations in φ we find that RMS difference is significantly less sensitive to mismatches. Here even a 10% difference only raises the floor slightly. Just as with *k* it shows symmetric increases with respect to positive and negative mismatches.

These investigations allow us to present a very promising improvement on comparison between simulation and experiment presented earlier. We can now generate a synchronization regime that includes a range of experimental error in the simulation. The end result is a synchronization regime that is nearly identical to the experimental data.



Comprehensive Coupling

There remains one variable we introduced subscripts for which we have not investigated (beyond a few simple values), γ. While we introduced fully independent labeling of γ above we will restrict our results to something that is a bit more experimentally reasonable, power-conserving coupling, that is, $\gamma_{12}+\gamma_{11}=\gamma_{22}+\gamma_{21}=1$. Alternatively we can think of it as specifying γ_1 and γ_2 then $\gamma_{12}=1-\gamma_2$ and $\gamma_{21}=1-\gamma_1$ .This still provides us with ample room for producing coupling demonstrated in both literature and our previous results. The entire realm of synchronization is presented in Fig 12.



The dotted lines represent the two “regions” we have swept in the above results. The vertical is for uni-directional coupling while the diagonal is for fully symmetric coupling. Lastly the dot represents the location of uni-directional, open loop synchronization. This is referenced fairly frequently in the literature and is often used for communication schemes.

Project Considerations

Below are presented the original timeline, mile-stones proposed at the beginning of the year and a brief description of the implementation results. With the exception of introducing quantization and noise, all of the project time-line elements have been achieved. This milestone was bypassed in deference to a fellow lab-mate who has begun investigations into that matter. We have however done significantly more investigations into novel results.

Milestones:

Implementation & Verification of individual simulations (complete)
Implementation & Verification of final, combined simulation (complete)
Generation of new results (in progress)
Expansion & further development of code (in progress)

Project Schedule

Goal/Stage		Completion Date
Implement and Validate Single Loop code	(complete)	Nov. 1^{st}
Implement and Validate Coupled Lorenz code	(complete)	1^{st} week Nov.
Implement and validate coupled MZ code	(complete)	Dec 1^{st}
Mid-Year Progress Report	(complete)	1^{st} Week Dec.
Generate Conditions for Time delay	(complete)	Jan. 1^{st}
Generate Conditions for Optical Biasing	(complete)	Jan. 1^{st}
Introduce Quantization and Noise in model		(Skipped)
Draft Final Report and Presentation		2^{nd} week April
Further Expansion of Code	(complete)	??

Implementation Comments

As stated earlier our choice was to implement the code in MATLab with extensions to C as needed for speed. With the current implementation a single loop can be simulated in <0.5 secs, and multiple loops synchronizing in <1.5 secs. This compares to approximately 2-3min for a single loop using the old code, and close to 4 or 5 minutes for synchronizing loops. Additionally, with the integration scheme memory storage was occasionally a bottle-neck depending on the length of time we wished to simulate (and the type of speed of the system we were simulating). With the discrete map equations however we are able to simulate 100's of micro-seconds of data without running into any memory cap. This is very useful for gathering large statistical data for averaging and comparing to experimental data.

References:

[1] Y.C. Kouomou. “Nonlinear Dynamics of Semiconductor Laser Systems with Feedback”, *Doctoral Thesis*, Palma de Mallorca, Oct 2006

[2] V.S. Anishchenko, A. N. Sil’chenko, and I.A. Khovanov. “Mutual synchronization and desynchronization of Lorenz systems” *Tech. Phys. Lett.* **24** 4, Apr 1998

[3] Argyris. “Chaos-based communications at high bit rates using commercial fibre-optic links” *Nature* **438**|17 Nov. 2005

[4] M. Piel, L. Larger, I. Fischer. “Versatile and robust chaos synchronization phenomena imposed by delayed shared feedback coupling” *Phys. Rev. E* **76** (2007) –045201(R)

[5] T. Murphy. “Discrete State-Space” *Personal Communications* Aug. 2008

[6] Wolf, *Progress in Optics*, Chap. 5 “Synchronization and communication with chaotic laser systems” Elsevier B.V. 2005

[7] L. Illing, D. Gauthier, and R. Roy “Controlling Optical Chaos, Spatio-temporal dynamics and patterns” *Advances in Atmoic, Molecular, and Optical Physics* **54** 2007

[8] L.M. Pecora et. al. “Fundamentals of synchronization in chaotic systems, concepts, and applications” *Chaos* **7** (4) 1997

[9] K-P Yip and N-H Holstein-Rathlou. “Chaos and Non-linear phenomena in renal vascular control” *Cardiovascular Research* **31(3):359-370**; 1996 doi:10.1016/S0008-6363(95)00083-6

[10] W. J. Freeman. “Synchronization of brain activity” *International Journal of Bifurcation and Chaos.* **10:2307-2322** 2000