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

Author:

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

Abstract:

Simulation of accurate models for experimental systems is vital to determining future research and validating existing research. I will implement a model for a system of coupled nonlinear time-delayed feedback loops. The model for each independent loop will be a 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 system [3,4]. This will be then used to predict synchronization behavior for previously unexplored system parameters τ and ϕ as it relates to the coupling strength. Time permitting, the time required to achieve synchronization and its dependence on system parameters will be explored.

Background

For highly productive experimental research to be conducted it is important to explore reasonable paths of investigation. With the breadth of available topics and directions determining the most fruitful paths can be difficult. One solution to this is effective modeling and prediction of experimental behavior through computer simulations. One current field of research is the synchronization of nonlinear systems.

Of current interest are nonlinear systems that involve a time-delayed feedback. One such system explored in detail by Kouomuo [1] is comprised of a laser, Mach-Zehnder inferometer, filtering, delay and amplification.

(Diagram)

Through basic mathematical relationships 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 $\tau$and 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.

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 so has 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 current experimental setup related to this project has 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 actually 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:

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

Where we have 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:

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

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 modification and feedback of the output of the filter, just as our function does. So, the final equation we will model is:

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

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 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 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 on the surface. There may be some issue from the combination of both digital and analog system components, which will be addressed if there is time since it is largely hidden in the established and tested hardware design of the digital signal processing board.

The second concern is developing an effective method for coupling two of these systems. A bi-directional coupling of the Lorenz system might look like:

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

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:

$$\mathbf{u}_1[n+1] = \mathbf{A}\mathbf{u}_1[n] + \mathbf{B}f(y_1[n-k]) + \mathbf{B}\gamma(f(y_2[n-k]) - f(y_1[n-k]))$$
$$y_1[n] = \mathbf{C}\mathbf{u}_1[n] + Df(y_1[n-k])$$
$$\mathbf{u}_2[n+1] = \mathbf{A}\mathbf{u}_2[n] + \mathbf{B}f(y_2[n-k]) + \mathbf{B}\gamma(f(y_1[n-k]) - f(y_2[n-k]))$$
$$y_2[n] = \mathbf{C}\mathbf{u}_2[n] + Df(y_2[n-k])$$

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:

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

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 I have chosen to implement a method 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 my 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 I will be able to ignore this. Since we seek to have highly accurate comparison to experimental results however, should there prove time later in the project I will introduce quantization into the model to reflect this expected behavior.

Validation

The simulation develop will take part in three stages, each independently verifiable. The first will involve implementing a single loop model as developed above. This will be verified against published work by Kouomuo et al. [1] on such systems. Specifically I will look for characteristic behavior of the system at unique parameter settings. Four such characteristic curves are displayed below, with their corresponding system parameters.

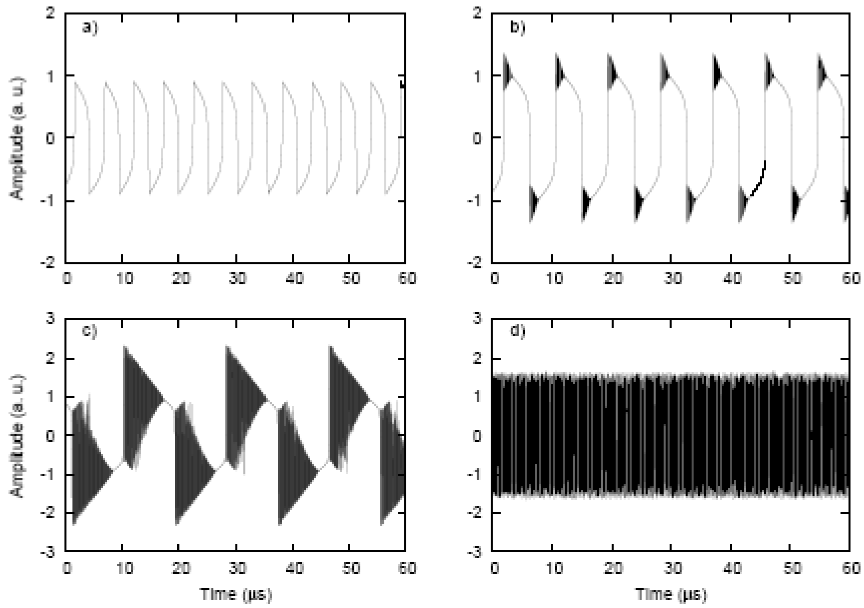


Figure 2.7. Birth, evolution and destruction of the breathers as the nonlinear feedback strength parameter β is increased, when $\phi = -\pi/4$ (symmetric case). $a)$ $\beta = 1.5$ $b)$ $\beta = 2.0$ $c)$ $\beta = 3.0$ $d)$ $\beta = 3.5$.

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

The final stage of implementation will be a combination of the previously mentioned models. To verify this I will compare 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]. I will demonstrate synchronization 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:

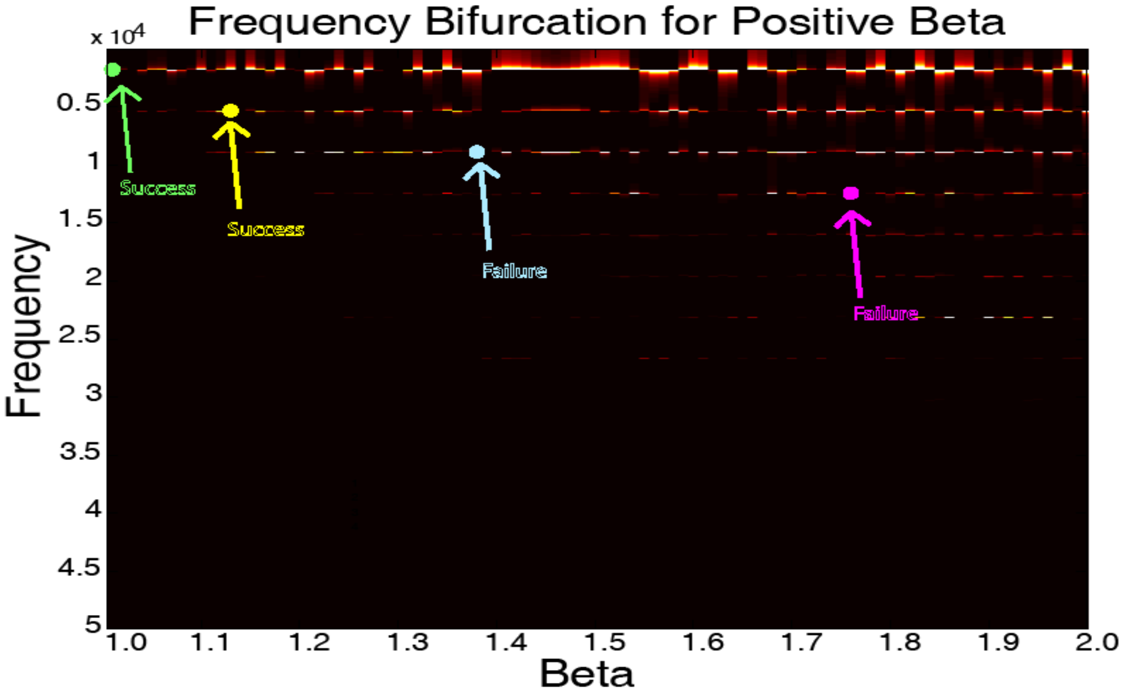
$$\gamma_k = (-1)^{k+1}[1 + \frac{(\varepsilon R^2 - k^2\pi^2)^2}{2k^2\pi^2 R^2}] \text{ and } \omega_k = k\frac{\pi}{R}$$

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:

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

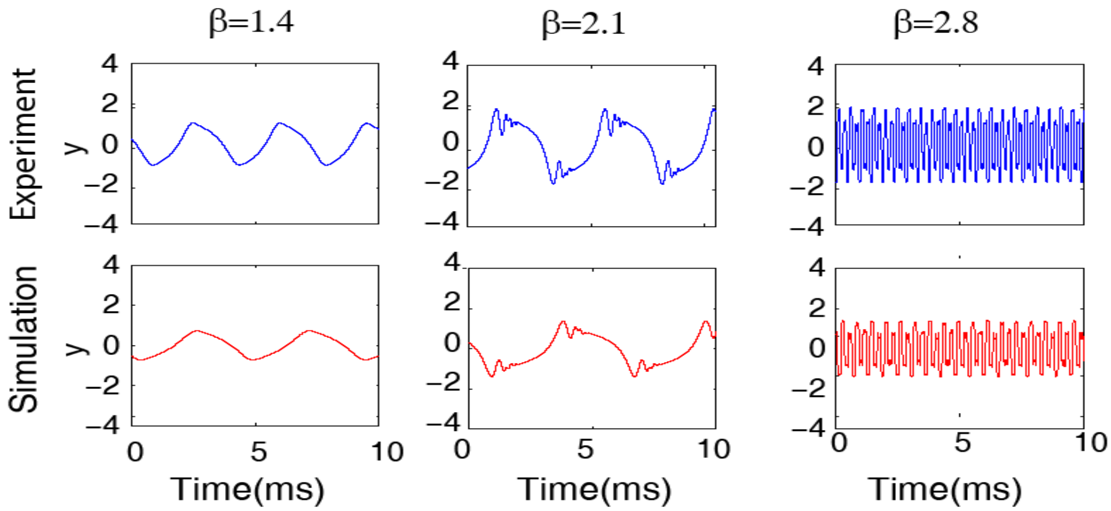
It is worth noting that these are only approximate solutions, so exact agreement to experimental or simulated results may not occur.

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.

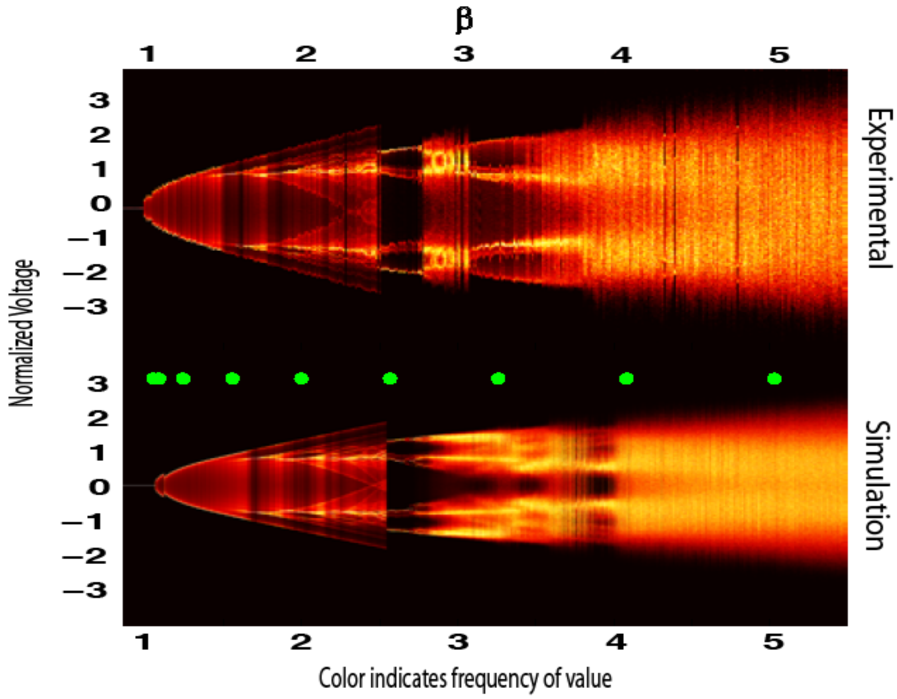


Marked on the graph are two successes and two failures of the simulations to match the analytical results. For higher values of β we see similar results to the 3rd and 4th points where the correct frequency is predicted but generally to the right (a higher β value) than the analytical results predict.

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.



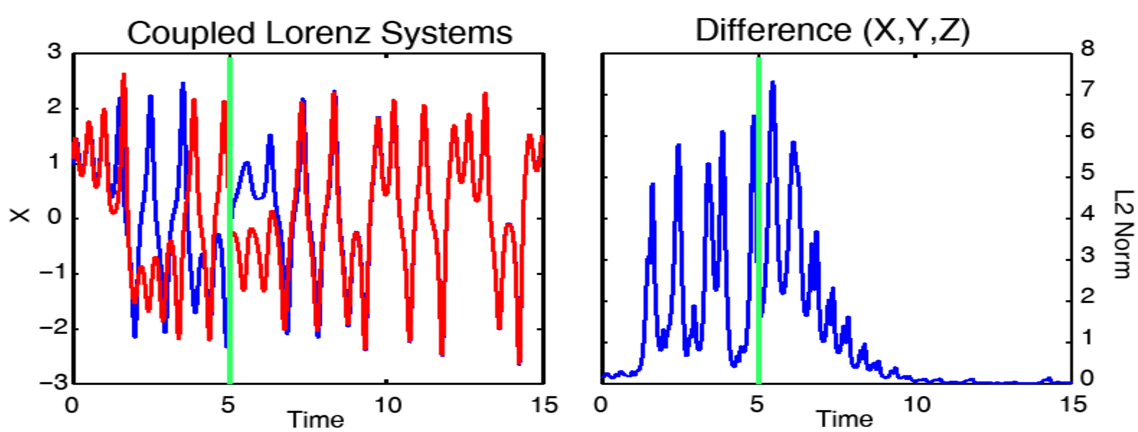
Here we can see that the simulated time series exhibit nearly identical behavior as the experimental data. The differences are a slight amplitude difference (simulation is ~80% of experimental) and a slight frequency mismatch. The amplitude difference is likely caused by an incorrect scaling factor in converting the experimental data to displayable data; however, the frequency mismatch is of a minor 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). With these slight mismatches accounted for though the model does give very good agreement 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. Though, given its ability to accurately reproduce a significant portion of the experimental data, I feel that the simulation can still be qualified as an over-all success. Also indicated on this diagram are the predicted β values for bifurcations.

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.

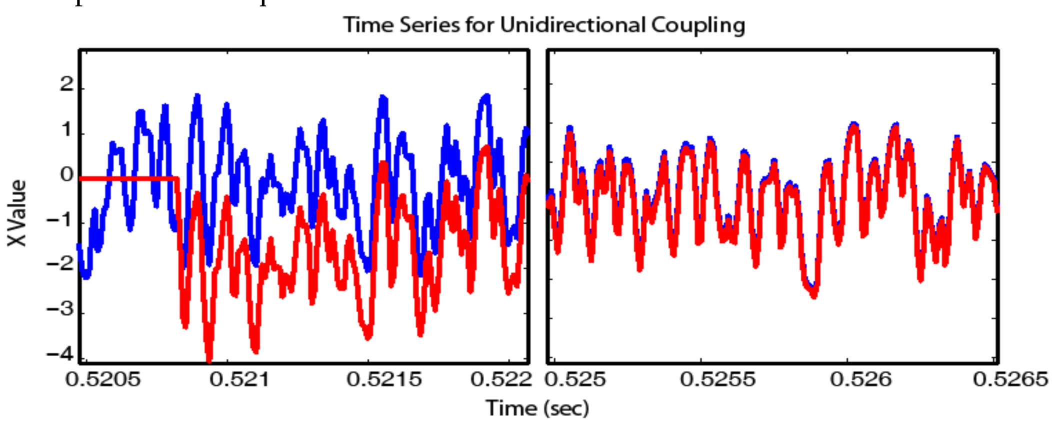
Synchronization of Couple Nonlinear Time-Delayed Feedback loops

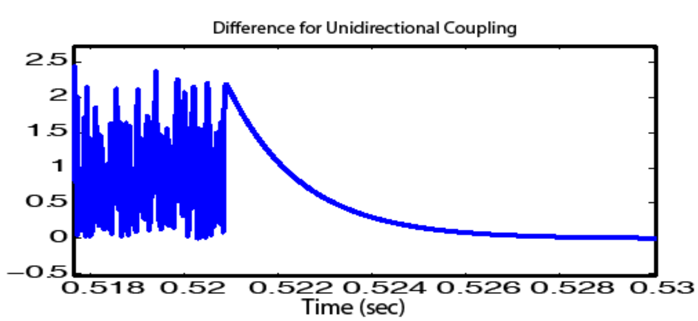
We previously developed equations for coupled Mach-Zehnder loops, and now we seek to simulate them, and attempt to find regimes under which synchronization can occur. Previously published in literature are results for synchronization of a master loop to an open loop as well as at the specific γ value 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:

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

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

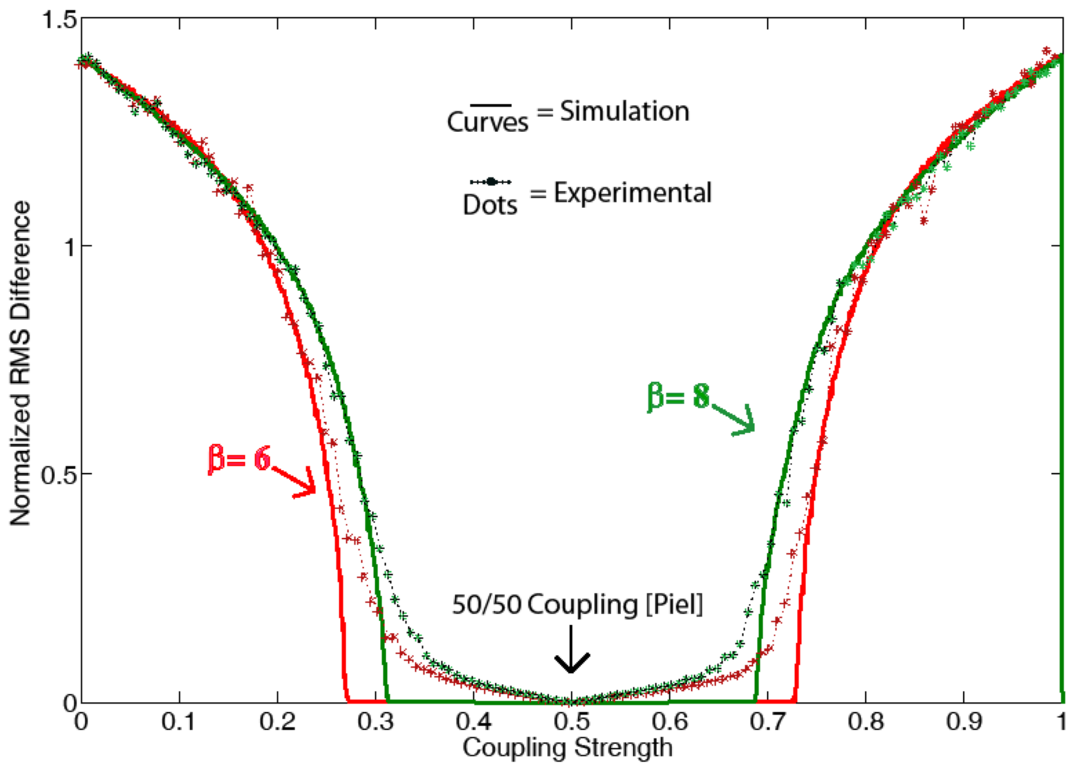
One can see then that the initially defined equations are just a special sub-set of these were: $\gamma_{11} = \gamma_{22} = 1-\gamma_{12}$. Argysis has demonstrated (and used) synchronization under a very specific regime of these were $\gamma_{11}=\gamma_{21}=1$ and $\gamma_{12}=\gamma_{22}=0$. That is, open-loop, unidirectional coupling. He explores this under a variety of coupling delays and conditions which I have checked against, 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 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% identical synchronization does occur, but 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 further expansion of the code and exploration into noise, quantization, and parameter mismatching will be vital to bring the simulation into greater agreement with the experiments.

Use of Code

Once validation has occurred, this code can be utilized to predict new and interesting behavior. I will perform simulations where previously unexplored system parameters are examined. Specifically the work will generate empirical conditions for synchronization based on variations in time delay (k) and optical biasing (ϕ) compared to the strength of system coupling (γ).

Expansion of the Code

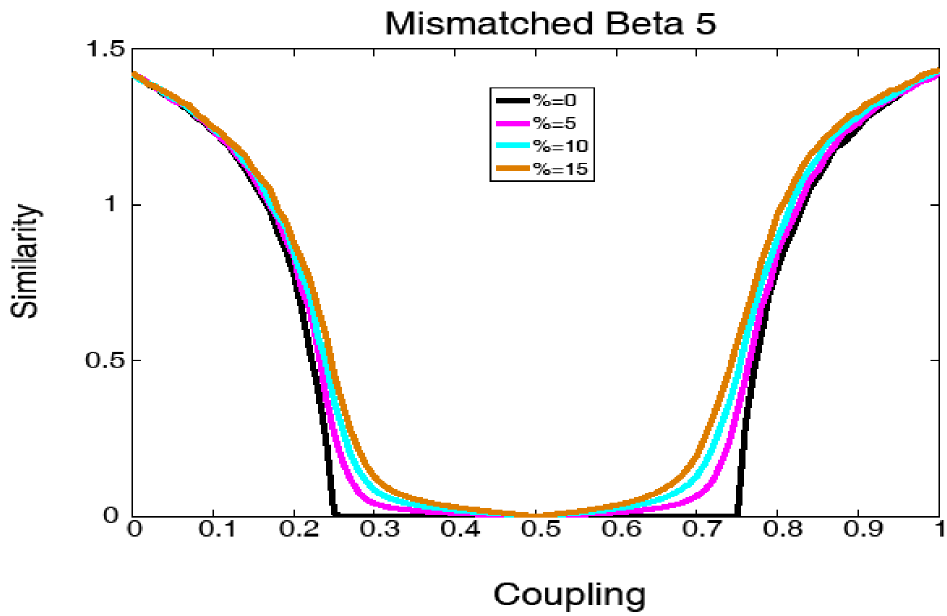
In light of the experimental data it has become important to expand the code in a number of ways. The first is to allow individual specification of parameters for both systems. In terms of the equations, we've essentially introduced subscripts onto many of the important system parameters and included 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 DSP 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]. This had provided promising results which will be discussed below.

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. We see below the preliminary results of these investigations.



Further Work

There are a few directions in which work will continue with this project. The first is to fully explore how mismatched parameters will affect synchronization. This is vital to understanding how the experiment can actually be improved. However to achieve this in a reasonable time period a second direction must be explored first. While an individual simulation with the code takes a very minimal amount of time (<1sec) when we begin sweeping large amounts of parameter space each individual simulation quickly builds up (for example 100-1000 γ values for 2-4 β values for multiple *k* values etc.), thus making a thorough investigation take hours or even days to explore one phenomena. In order to alleviate this problem I will be porting my code from MATLAB to C/C++ in order to more easily parallize the execution. There are two issues that require changing for this. The first will be generating and then inputting filter parameters into a C code. For this I will generate the values in MATLAB and just directly input them for the code. The second is that our current algorithm involves matrix multiplication, which while possible in C requires including other libraries or extensive for loops. Luckily however, for this specific case of a system we are dealing with very small order filters, which therefore generate very small order matrices. This will allow direct unrolling of the matrix multiplications into just 4 lines of multiplies, which should be significantly more efficient than making calls to another C library.

Milestones:

Implementation & Verification of individual simulations

Implementation & Verification of final, combined simulation
Generation of new results
Expansion & further development of code

Project Schedule

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

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