A Power-Shift Method to

Control Nonlinear

Systems

By

Erikk Kenneth Tilus Burton

Bachelor of Science in Physics
Oklahoma State University
Stillwater, Oklahoma
2017

Bachelor of Science in Mathematics
Oklahoma State University
Stillwater, Oklahoma
2017

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
DOCTOR OF PHILOSOPHY
July, 2022

A Power-Shift Method to

Control Nonlinear

Systems

Dissertation Approved:

Dr. Donna K. Bandy

Dissertation Advisor

Dr. Albert T. Rosenberger

Dr. Joseph Haley

Dr. Frank D. Blum

ACKNOWLEDGMENTS

I would like to thank my advisor Dr. Donna Bandy for her support and wisdom she has imparted over the years. It is thanks to her tutelage that I even started doing research. I would also like to thank Joshua Hall and Dylan Chapman for their help and camaraderie. Their knowledge, hard work and personalities made research far more manageable. Without them, I would be far behind where I am now.

I would like to thank my mom and my dad for their love and support over the years. For all they have done I am eternally grateful.

Acknowledgments reflect the views of the author and are not endorsed by committee members or Oklahoma State University.

Name: Erikk Kenneth Tilus Burton

Date of Degree: July 2022

Title of Study: A POWER-SHIFT METHOD TO CONTROL NONLINEAR SYSTEMS

Major Field: PHYSICS

Abstract:

A theoretical technique is developed herein which allows for the experimental access of attractors in a coherent laser with injected signal. This technique is theoretically tested for use in other nonlinear systems such as an optomechanical system. The technique, known as the power-shift method, works by rapidly changing the control parameter of the nonlinear dynamical system in order to access specific dynamics in the system.

Noise and time lags in the shift are added to study the viability of the technique under experimental conditions. The time lags are found in some situations to enhance the effectiveness of the power-shift method. Another version of the power-shift method is developed which generalizes the technique to other system parameters, namely the gain parameter. Pulse versions of the technique are tested using Gaussian, sech^2, and square wave pulses. The pulses are initially tested with a constant peak and then with a constant area under the curve while varying the full width half maximum.

The power-shift method is also tested in an optomechanical system. In this system, the power-shift method succeeds and possesses a new option in the form of shifting from stable system points. By using stable points, the system need not be timed to access attractors therefore removing a potential point of failure.

The investigation finds that the power-shift method is able to access coexisting attractors with small levels of noise and with time lags in the power shift. Of the pulse variants, the square wave shows the most promise; this is followed by the Gaussian pulse. The investigation finds that fast intense pulses are effective at perturbing the system to an attractor.

Finally, examples of how a power-shift method might be executed experimentally are demonstrated along with examples shown in the laser with injected signal and the optomechanical system.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Figure Page

CHAPTER I

INTRODUCTION

1.1 Background

For the past 50 years, the field of nonlinear dynamics has grown to find use in various disciplines from physics to economics [1]. Nonlinear dynamics involves the study of dynamic systems modeled by nonlinear differential equations. These nonlinear differential equations are typically unsolvable through analytical means and therefore need to be solved numerically. This requires heavy use of computer simulations to demonstrate the system's evolution over time. A defining feature of nonlinear dynamics is multistability or multiple "coexisting attractors." Nonlinear systems are sensitive to initial conditions which means, even with all the parameters exactly the same, a different set of initial conditions may result in completely different behaviour. This causes these systems to be difficult to predict or control, which continues to challenge investigators to search for various experimental ways to investigate and utilize nonlinear systems.

1.2 History

Originally, scientists controlled nonlinear systems using two main techniques: (i) by introducing feedback loops to the system [2, 3] (ii) by changing the system parameters [4, 5]. Scientists such as Ott, Grebogi and Yorke set about controlling chaos via feedback methods [2]. Later Pyragas [3] stabilized unstable orbits using feedback loops which do not require, a priori, knowledge of the system to use. Scientists such as Pisarchik [4], and Li and Sprott controlled nonlinear systems by changing the system parameters [5]. Both of these methods changed the nature of the systems making them ultimately different systems. If one wishes to investigate coexisting attractors already present in the system, these techniques do not work. The problem then, is how does one control a nonlinear system without fundamentally changing the system in the process? This question led to the creation of the power-shift method, a method that keeps the system unchanged in its original parameters.

1.3 Power-shift method

The power-shift method is a technique which uses the system's own dynamics to control the system and is specifically designed for experimental use. The essence of the power-shift method is perturbing the control parameter while the system is at a certain state in order to access a coexisting attractor. In this process, only the control parameter is changed leaving the system intact.

The technique is similar to the one Chizhevsky in 2000 developed [6]. This technique used a modulated loss pulse to perturb a CO_2 laser in order to move the system to and from different attractors by timing the pulse at specific times on the limit cycle. The system remains unaltered as the parameters are the same after the pulse. As opposed to Chizhevsky's technique, the power-shift method uses the known values of the dynamical states to access attractors and is mainly a shift instead of a pulse.

1.3.1 Power-shift history

Before the power shift

Attractors were originally accessed by perturbing or directly setting the variable values to conditions within the desired attractor's basin. The problem with directly setting the variables is that it is not realistic in experiment. Variables such as the electric field and the polarization cannot be set directly without fundamentally changing the experimental condition, thus unintentionally affecting other variables. This fact led to investigations into other ways attractors are accessed.

One part of analyzing attractors is measuring the range of the control parameter for which the attractor is available. This is called the attractor domain. The attractor domain is measured by adiabatically scanning the control parameter as in Ref. [7]. This is performed in incremental steps of the control parameter then allowing the system to evolve in time to its stable state. When attempting to map out attractor domains in the laser with injected signal, the system sometimes falls off the attractor because the step size in the input signal is too large. This begs the question; what if, instead of accidentally falling off the attractor, one intentionally falls off to a different attractor using a large step in the control parameter? This question led to the creation of the power shift.

Literature on power shift

In January 2021, Burton et al. [8] published the initial results of the power-shift method. The study tested the basic theoretical uses of the technique such as its ability to discover and reliably access attractors, along with some preliminary results on its experimental viability. Experimentalists have new tools to control nonlinear systems that are effective and straight forward to apply.

1.3.2 Power-shift variants

To expand on the power-shift method, additional experimental-type conditions are developed which generalize the method. In particular, the gain parameter is tested as the parameter to shift. Additionally, instead of a sudden or linear shift, the use of an interim shift where the parameter is modulated only temporarily is investigated. This is done with different types of pulses: square, Gaussian and sech^2 pulse. The system is *temporarily* changed but ultimately it remains the same at the end of the process because it ends with the same parameters as before. These pulses are therefore merely perturbations to the system.

Some preliminary investigations by Hall et al [9] show viability for gain shifting. In this work, an interim gain shift successfully moves the system from one attractor to another which shows there is a probability that the variant technique is successful The interim pulse used is essentially a wide square wave pulse where the parameter is temporarily modified to a new value and then returns to its original value after an interval of time. Hall and others kept the time fixed to return to its original value at $t = 10,000\tau$

This dissertation broadens the study of pulses by first changing the pulse duration while keeping a constant peak and then by keeping a constant area under the curve.

1.4 Attractor basics

This study predominately examines attractors. According to Ref. [10], an attractor is a set of states where neighboring states within its basin of attraction asymptotically approaches with time. The attractor is the smallest unit which cannot be subdivided into smaller attractors.

Examples of attractors

To better understand attractors, let us look at some different oscillating systems starting with the simple harmonic oscillator governed by the equation:

$$m\frac{d^2x}{dt^2} + m\omega_0^2 x = 0. \tag{1.4.1}$$

In this case, the system oscillates with an amplitude based on its initial conditions. See Fig. 1 for the phase spaces of two different initial conditions for the simple harmonic motion.

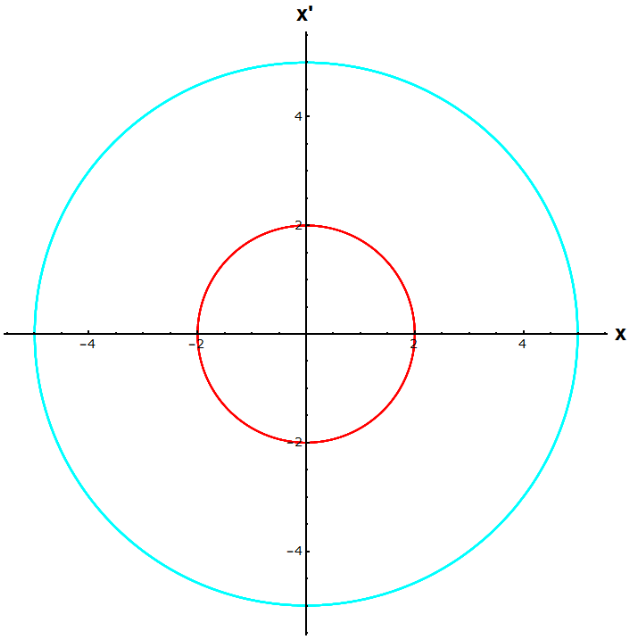


Figure 1: The phase spaces for a simple harmonic oscillator. The red and blue circles arise from two different initial conditions, $x' = 0, x = 2$ and $x' = 0, x = 5$ respectively.

This system has no attractors because the states the system cycles through do not attract any neighboring states. The beginning circle is the cycle the system stays on.

If one adds a damping term to equation 1.4.1, it becomes:

$$m\frac{d^2x}{dt^2} + \epsilon\frac{dx}{dt} + m\omega_0^2 x = 0. \tag{1.4.2}$$

Here the system loses energy until it eventually reaches its stopping point at $\frac{dx}{dt} = x = 0$. Because all the other states asymptotically approach this point, it is the systems only attractor. This single point is known as a steady state and is the simplest type of attractor.

If one switches the damping term to $-\epsilon(1-x^2)$, then the system becomes the nonlinear van der Pol equation; in particular, one without the forced oscillation term in Ref. [11], the equation becomes:

$$m\frac{d^2x}{dt^2} - \epsilon(1-x^2)\frac{dx}{dt} + m\omega_0^2 x = 0. \tag{1.4.3}$$

If $\epsilon > 0$, then the system is dissipative when $x^2 > 1$, and driven when $x^2 < 1$. Except when $x = \frac{dx}{dt} = 0$, the system *always* approaches the same amplitude regardless of initial conditions. The amplitude produced does not depend on the initial position of the system as it does in the linear simple harmonic oscillator. See Fig. 2 for an example of the attractor and two initial conditions. The attractor does not change with different conditions; if the trajectory changes then it means the system is now on a different attractor. For more information on attractor, refer to the Appendix, Attractor information.

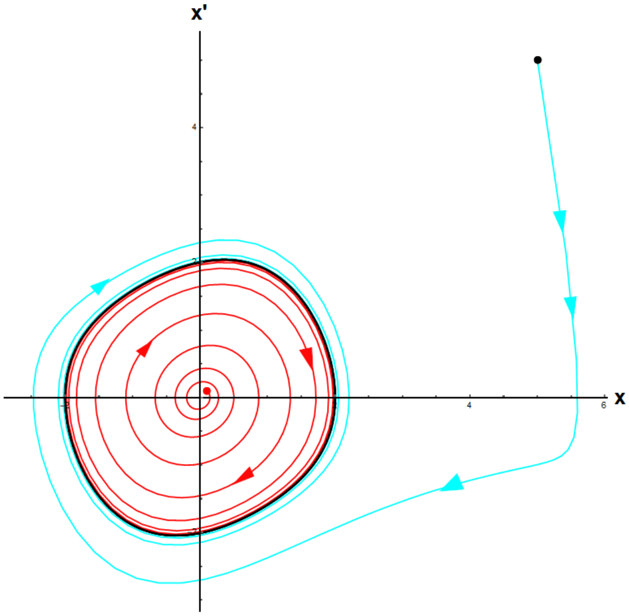


Figure 2: The van der Pol attractor is the limit cycle in black, the black and red dots are two different initial conditions, the spirals from the dots are the transient behavior as the system evolves to the limit cycle.

CHAPTER II

THE MODELS

Two different models are used to test the power-shift method's capacity to access attractors, the laser with injected signal and an optomechanical system. Both models are nonlinear and possess a control parameter in the form of an incoming laser signal.

2.1 The laser with injected signal (LIS)

The laser with injected signal is a model known for its universal application to nonlinear systems in general. In the model, an optical signal with frequency ω_0 is injected into a laser ring cavity with cavity frequency ω_c and atomic frequency ω_a. The three frequencies are close, but are typically detuned from the injected signal. The laser ring cavity has two perfectly reflective mirrors and two partially reflective mirrors. The different frequencies interact along with the nonlinearities and produce various nonlinear phenomena. At low injected signal amplitudes, beat notes are found. At high input amplitudes, injection locking occurs leaving a single point in phase space or a constant signal in time. In between these extremes the system can possess various nonlinear phenomena such as limit cycles, tori and chaos. It is in this region the bulk of research is done. Figure 3 shows a schematic of LIS.

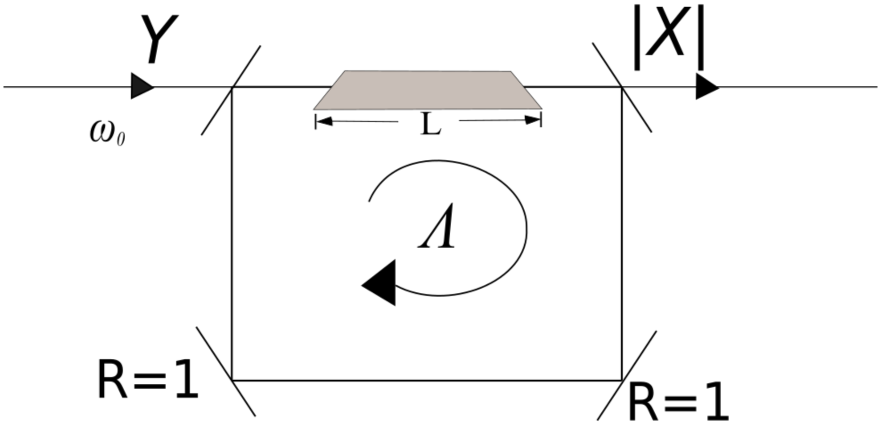


Figure 3: LIS schematic.

The system is modeled by the coupled nonlinear equations:

$$\frac{dX}{dt} = -\tilde{\kappa}[(1 - i\frac{\Phi}{\tilde{\kappa}})X - Y + 2CP], \tag{2.1.1}$$

$$\frac{dP}{dt} = -(1 + i\tilde{\Delta})P + XD, \tag{2.1.2}$$

$$\frac{dD}{dt} = -\tilde{\gamma}[\frac{1}{2}(XP^* + X^*P) + D + 1]. \tag{2.1.3}$$

where X is the complex electric field, P is the complex polarization, and the real population difference is D. This investigation uses the parameters from Ref. [12]. The parameters are all scaled by the polarization dephasing rate $\gamma_\perp$, $\tilde{\kappa}$ is the cavity relaxation rate. $\tilde{\Delta}$ is the detuning between the laser and atomic frequency. Φ is the mistuning between the laser and cavity frequency. Y is the injected signal amplitude, $\tilde{\gamma}$ is the scaled atomic linewidth, and C is the gain parameter. The parameter values used are: $\tilde{\Delta} = 0.5$, $\frac{\Phi}{\tilde{\kappa}} = -0.5$, $\tilde{\kappa} = 0.1$, $\tilde{\gamma} = 0.01$, $C = 3$. These parameters are chosen because they reveal interesting phenomena while still being within the bounds of reality.

Laser classes

There are three general types of laser classes [13, 14], Class A, B, and C. In Class A, the polarizing dephase rate and population relaxation rates are much greater than the cavity relaxation rates. In Class B, the polarizing dephase rate greatly exceeds the cavity and atomic relaxation rates. In Class C, all the rates are similar. The parameters in this study fall between Class B and C.

2.1.1 Stability analysis

One of the first steps of analyzing a new system is to check for instabilities. The equations are solved around steady state, then tested for stability by linearising the equations and using an analysis routine called the Huritz criterion. To check the stability of steady states the Hurwitz criterion is used [15]. The nonlinear equations are linearised by expanding around steady state and then partial derivatives are taken with respect to each state variable. These are placed into a matrix and z is subtracted along the diagonal. The determinant is taken and the z terms are combined to create the equation:

$$f(z) = c_n z^n + c_{n-1}z^{n-1} + c_{n-2}z^{n-2} + c_{n-3}z^{n-3} + ... + c_0 \tag{2.1.4}$$

The coefficients are placed on a matrix in this form:

$$H(f) = \begin{bmatrix} c_{n-1} & c_{n-3} & c_{n-5} & ... & 0 \\ c_n & c_{n-2} & c_{n-4} & ... & 0 \\ 0 & c_{n-1} & c_{n-3} & ... & 0 \\ 0 & c_n & c_{n-2} & ... & 0 \\ . & . & . & ... & . \end{bmatrix} \tag{2.1.5}$$

Then the matrix is used to create multiple determinants where each h_n is the determinate is an $n \times n$ matrix made by taking the top left portion of $H(f)$. If any of the c_i terms or the h_n determinants of $H(f)$ are negative, then the system is unstable at steady state.

The parameters in this study yield a steady-state curve shown in Fig. 4. At $Y = 2.16478$, the system reaches injection locking. Note, the entire lower branch and a portion of the upper branch is unstable, which means there is no possibility for bistable behaviour in this system.

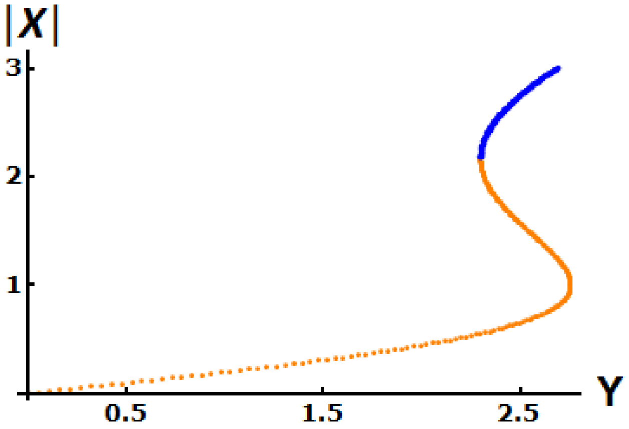


Figure 4: The steady-state curve for LIS.

2.1.2 LIS attractors

The LIS system used for this study contain eight known attractors (AI - AVIII) and two periodic windows within chaos. Figure 5 is a map of the known attractors and windows in this model. Here the attractors are sorted by their relative ‘characteristic frequency’.

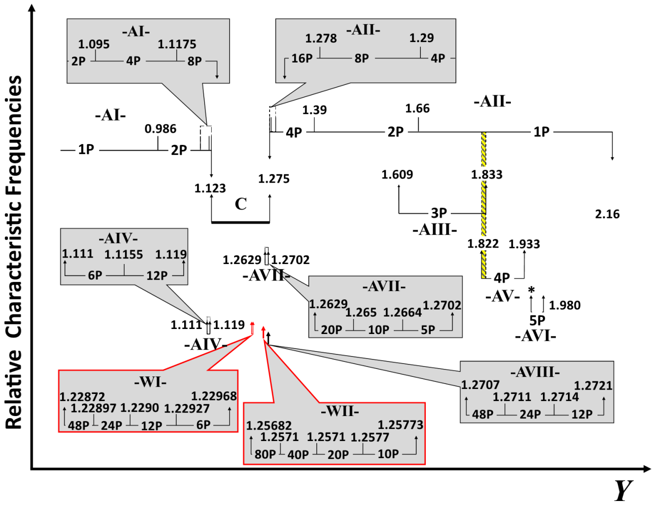


Figure 5: A map of the attractors in LIS. The abscissa is the injected signal amplitude Y. The ordinate is the relative characteristic attractor frequency. The yellow shaded region is three coexisting attractors. The * is the lower domain edge of AVI, $Y = 1.948$.

That is, by the lowest unique frequency which exists throughout the entirety of the attractor domain relative to other attractors. WI and WII are two periodic windows in quasichaos. The first seven attractors were discovered using traditional theoretical tools such as variable perturbations, adiabatic scans of Lyapunov exponents, and global sweeps [7].

2.2 Optomechanical system (OM)

The other model explored is an optomechanical system used in Ref. [16]. The model is based on an optical fiber tapered to a few microns which touches, or is closely placed next to, a microresonator along its equator; light is injected through one end of the fiber [17]. The light, while in the untapered region, is confined by total internal reflection between the core and cladding. Once in the tapered region, the light changes from the core-cladding guidance to cladding-air guidance which creates an external evanescent field that couples into the resonator. Eventually the resonator's evanescent field allows the light to couple back into the fiber. The photons optically tunnel from the tapered fiber into the resonator which excites a whispering gallery mode (WGM) and then tunnels back into the fiber. A schematic for this is shown in Fig. 6.

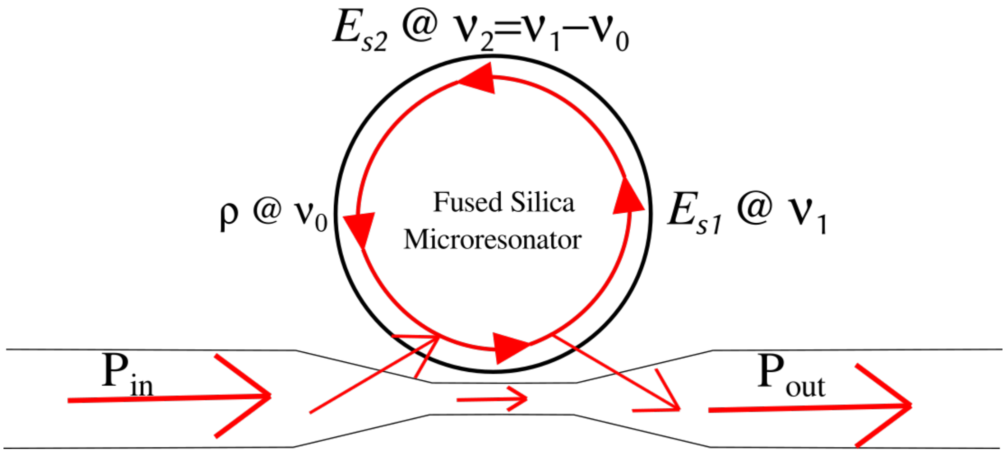


Figure 6: OM schematic. Light travels through an optical fiber with some optical tunnelling in and out of the microresonator sphere.

The photon that enters the glass bead-like resonator interacts with the medium creating a new photon and a phonon. The phonon and new photon's momentum and energy add up to that of the original photon. The modeled equations are [18]:

$$\dot{E}_{s1} = -[k_1 - i2\pi\tilde{\Delta}_1]E_{s1} - [\frac{t_1t_2}{2\tau} - ic_1\rho]E_{s2} + i\frac{t_1}{\tau}E_{in}, \tag{2.2.1}$$

$$\dot{E}_{s2} = -[k_2 - i2\pi\tilde{\Delta}_2]E_{s2} - [\frac{t_1t_2}{2\tau} - ic_1\rho^*]E_{s1} + i\frac{t_2}{\tau}E_{in}, \tag{2.2.2}$$

$$\dot{\rho} = -[\frac{\Gamma_B}{2} + i2\pi(\tilde{\Delta}_2 - \tilde{\Delta}_1) - i2\pi\tilde{\nu}_0]\rho + ic_2(\tilde{\Delta}_2 - \tilde{\Delta}_1)E_{s1}E^*_{s2}, \tag{2.2.3}$$

where E_{s1} is the electric field of the incoming photons, E_{s2} is the electric field of the generated photons and ρ is the density wave of the phonon. All three of these variables are complex and can be split into real and imaginary components, making the system six dimensional. E_{in} is the input electric field's amplitude. $k_j = \pi\frac{c}{\lambda Q_j}$ and Q_j $(j = 1, 2)$ are the cavity qualities of individual field modes. it_j are the input/output coupling coefficients of E_{sj}. $\tilde{\Delta}_j$ are the scaled detuning of the two field modes from the input resonance frequency. ν_0 is the scaled acoustic whispering gallery mode frequency. τ is the round trip time of modes 1 and 2. Γ_B is the Brillouin linewidth or acoustic relaxation rate based on silica with $\lambda = 1,550nm$ P_{in} is the input power, $c_1 = 2 \times 10^5$ and $c_2 = (2\pi)4.14 \times 10^{-19}$ are the scaled couplings, and the output power P_{out} is give by:

$$P_{out} = |c_1\sqrt{P_{in}} + it_1E_{s1} + it_2E_{s2}|^2c_1^{-2}. \tag{2.2.4}$$

Time in the OM system is measured in microseconds, frequency in Megahertz, and power is measured in milliwatts. In this study, the parameters are:

$$\Gamma_B = 13, \tilde{\Delta}_1 = 2\pi, \tilde{\Delta}_2 = 44 \cdot 2\pi, a = 175, k_j = \frac{6.08 \times 10^8}{Q_j}, \nu_0 = 50(2\pi),$$
$$Q_1 = Q_2 = 10^8, t_j = 2.29\sqrt{\frac{a}{Q_i}}, \tau = 3 \times 10^{-8}a.(j = 1, 2)$$

2.2.1 OM attractors

This system contains attractors and chaos much like LIS but also contains ghosting and stable states which can coexist with with either one. The system has more attractors than the LIS system used herein. For $P_{in} \geq 20$ only one attractor appears to exist and eventually it collapses to a single stable point in phase space in the tested region. The OM system possesses tori and chaos along with quasichaos, hyperchaos, and limit cycles. The stable points contained in the system prove useful for power shifting as they provide a base that does not need to be timed.

A map of some of the system attractors is shown in Fig. 7 along with the steady state that is color coded on the reverse S-curve of the model. The blue regions are stable and the orange are unstable. There is a region of bistability around $P_{in} = 1.76$ shown in Fig. 8. While it is close, there is not any bistability around $P_{in} = 3.5$.

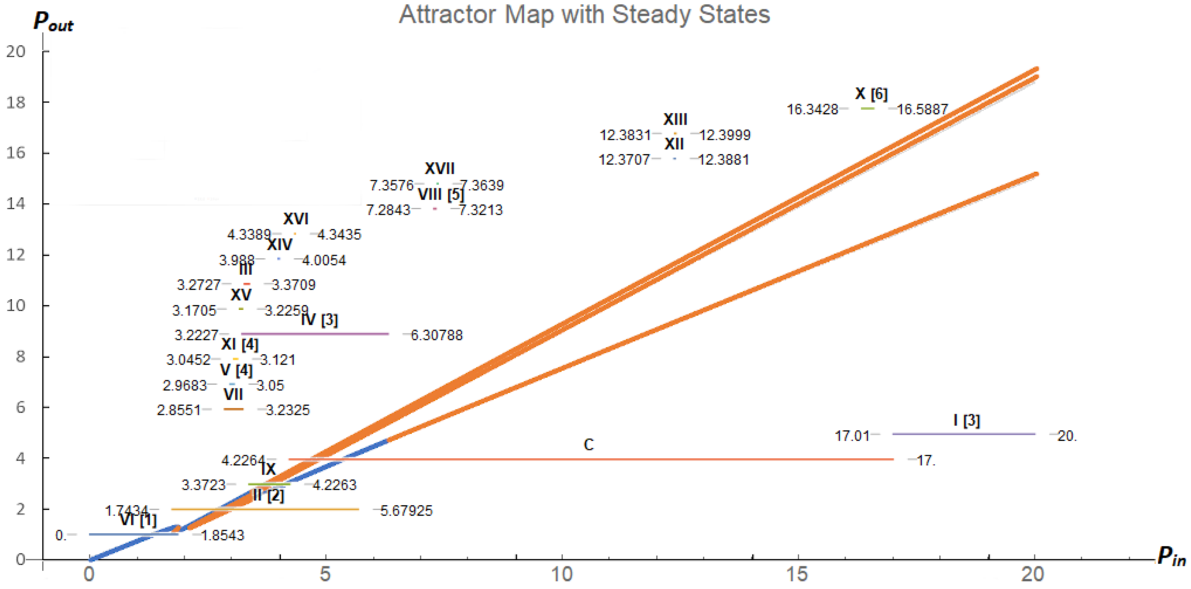


Figure 7: [1] steady state, [2] steady state → limit cycle → chaos [3] chaos → limit cycle → steady state, [4] limit cycle → chaos, [5] limit cycle → torus → limit cycle, [6] chaos → limit cycle. A map of the attractors in the OM model. The abscissa is the P_{in}. The ordinate is the output power.

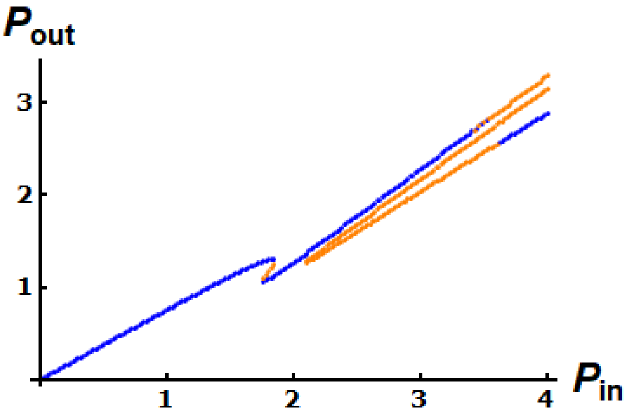


Figure 8: The steady-state curve for the low power region of OM. The abscissa is the P_{in}. The ordinate is the attractor's output power.

CHAPTER III

POWER-SHIFT METHOD TECHNIQUES

3.1 Power shift in LIS

The power-shift method is investigated using programs written in Wolfram Mathematica. The nonlinear equations are solved numerically with NDSolve with a time step $\Delta t = 0.2$ for LIS. The method for power shifting varies depending on whether the shift is a sudden change, a linear ramp, or a pulse. The Mathematica programs used for the power shift are shown in the Appendix section 5.6.

3.1.1 Method to discover attractors

Below is the generic procedure to discover attractors in a nonlinear system via power shift:

Step 1: Take the desired range of the control parameter and divide it into equally spaced segments. This creates a sequence of n numbers.

Step 2: Shuffle the segment values to create a new random sequence of n numbers.
The Mathematica program uses RandomSample[] to shuffle the list of
Y values.

Step 3: Initiate the dynamic evolution using the first control value of the sequence
and allow the system to evolve for an interval T of time. T should be long enough
for the system to settle on a limit cycle if one exists. If possible, record the state of the system at time T.

Step 4: Shift the control parameter to the next value in the random sequence and let the system evolve for an additional interval T of time.

Repeat steps 3 and 4 until all the values in the sequence are run. Create another random sequence and repeat the process. Then compare the dynamics found at each value of the control parameter. Two contradicting phase space images imply coexistence.

3.1.2 Results of the Power-shift discovery

The domain of interest from $0 < Y < 2$ is divided into 501 segments and shuffled to make two random sequences. The sequences are evolved for $T = 20,000\tau$ so the system can settle. At $t = T$, the final conditions are recorded and Y is changed to the next value in the sequence.

In the first two sequences a discrepancy in the dynamic trajectory appears at $Y = 1.272$. In one sequence the phase space is of the expected chaos but in the other, a previously unknown limit cycle appeared. The phase spaces of the two are shown together in Fig. 9. This discrepancy implies a new attractor. The new attractor is drawn in red lines and the chaos in black dots. Their trajectories are close together in phase space which could explain why AVIII was not discovered earlier.

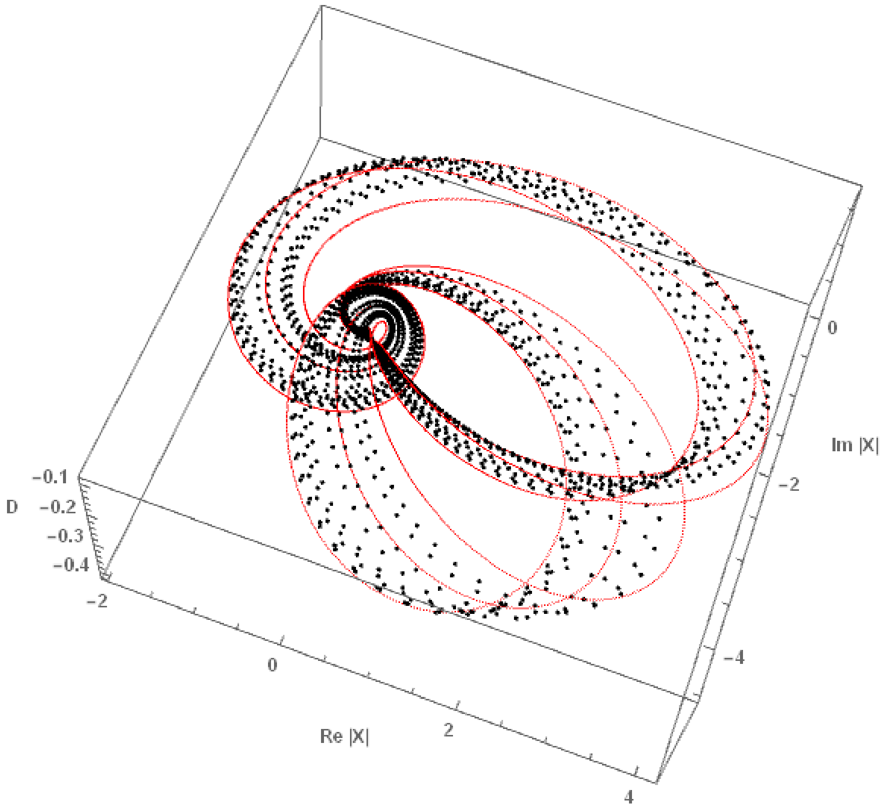


Figure 9: AVIII (red lines) coexisting with chaos (black dots)

In addition to the new attractor, 5 of the 7 other attractors appeared in the first two sequences with power shifting. In a 3rd sequence the other attractors appear.

3.1.3 Method to reliably access attractors

The procedure for *discovering* attractors is not reliable for *accessing* attractors, because of its inherent randomness. To account for this, another implementation of the power-shift method is needed. The method needs to power shift intentionally instead of randomly to access the desired attractor reliably. This means starting on a known attractor, called the base attractor, and power shifting at some location on it so the system evolves to a new attractor, called the resultant attractor. If the resultant attractor is the desired attractor, then the power shift is successful.

Due to the fractal nature of attractor basins, only some of a base attractor's points may reside in the basin of attraction of a desired attractor (at a different control parameter value) while the rest reside in another. This means a blind power shift is unreliable. As an example of an unreliable power shift, one can look at the discovery of AVIII.

Starting from the power-shift results in the first two sequences, it is known that the power shift from $Y = 1.04$ to $Y = 1.272$ can access AVIII. Figures 10 show the temporal and phase space graphs of base AI at $Y = 1.04$. The point's color describes the resulting attractor when power shifted from the point to $Y = 1.272$. Red signifies chaos and blue is AVIII. Only eight points successfully evolve to AVIII, the rest go to chaos. This base attractor is therefore not a reliable power shift candidate because of the small number of points on the limit cycle that successfully evolve to AVIII. Additionally, these points are dispersed throughout the limit cycle meaning that timing needs to be extremely accurate.

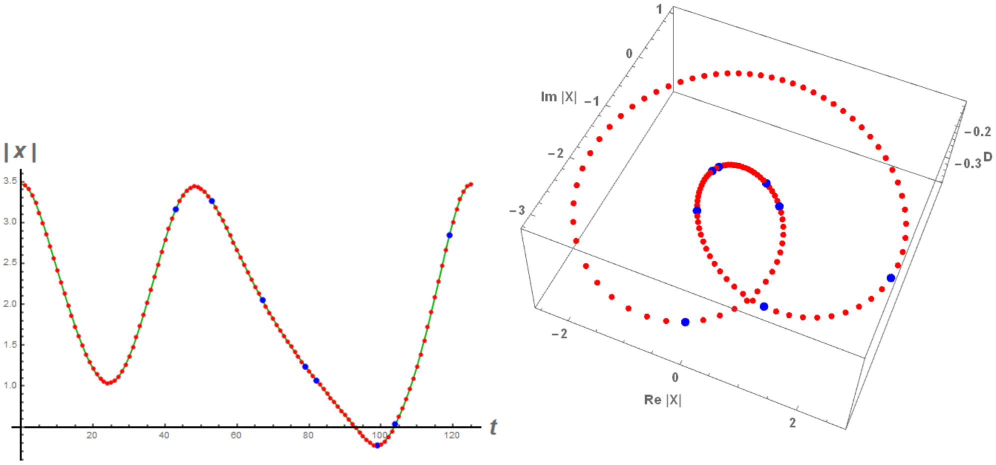


Figure 10: Power shift from $Y = 1.04$ to $Y = 1.272$

There are two ways to increase the reliability of the power shift: (i) time the shift to be within a certain time window and (ii) change the base attractor to one where said time window is as large as possible.

Time after peak

In order to time the shift one needs some landmark on an attractor's structure to begin the timing. For LIS, we use the largest peak of the output amplitude $|X|$ observed in the temporal dynamics. Because a limit cycle has a known fundamental frequency and amplitude along with a closed trajectory, any coordinate on the limit cycle can be described by the amount of time it takes to evolve the system from the designated peak to the coordinate. This time is called *the time after peak* (TAP). By doing so, a multidimensional limit cycle can be reduced to a single dimension. After power shifting at TAP and allowing the system to evolve, the resultant attractor and TAP are recorded in a table and then graphed to find any potential contiguous TAP points. The power shift in the NDSolve thus becomes WhenEvent$[t = TAP, Y \rightarrow Y_{pert}]$ see Fig. 11, for example. Here the maximum tranche of TAPS is graphed as a function of the control parameter Y (associated with AIII).

Finding a good base attractor

To find a better base attractor, one can test different starting control parameter values to find the base attractor with the largest tranche of points which go to the desired attractor. To find a good base attractor, an attractor is selected to be the base attractor and is broken down into sections evenly divided in time. This gives a number of points to sample around the base attractor. The system is power shifted to a new control parameter value and allowed to evolve from each point on the base attractor. The resulting attractor is analyzed to see if it is the desired attractor or not. Once all the points are tested, the resulting attractors and the power-shifted location on the base attractor are both recorded in a table. The largest continuous string of successful power shifts is recorded and evaluated as the base attractors viability.

Reliable results

AV is chosen as the desired attractor because its domain is relatively small but is shown to still survive some noise. This makes it a reasonable attractor to try to access experimentally. Figure 11 is created to look for a better base attractor to access AV. Here the time tranche to AV is analyzed starting from base AIII. The abscissa is the initial injected signal amplitude Y and the ordinate is the size of the largest time window (or tranche of points) on the limit cycle which evolves to AV. All the power shifts for the graph shift to $Y = 1.84$. In this case $Y = 1.63$ possesses the largest tranche.

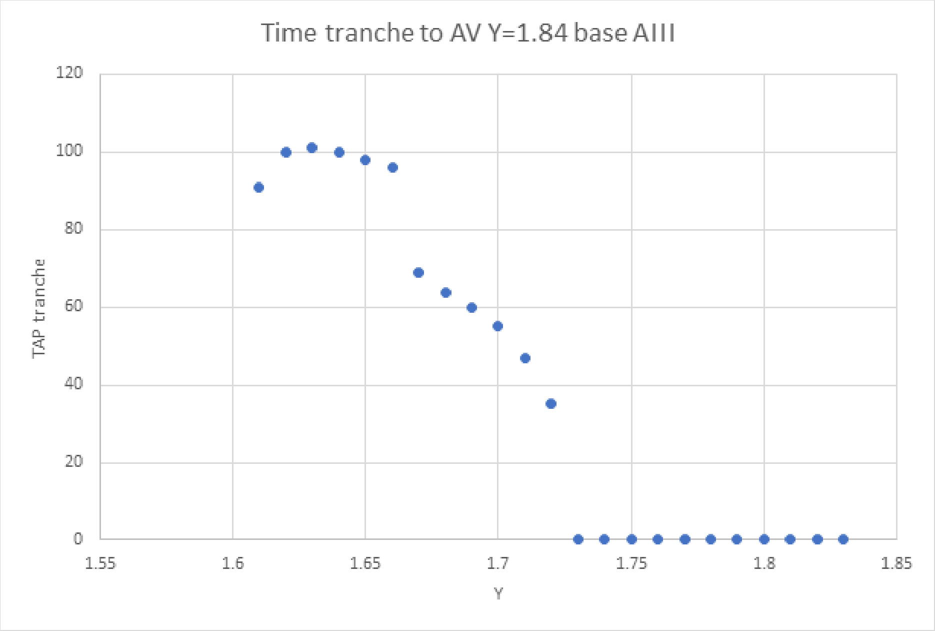


Figure 11: The results of finding a more optimal power shift to AV from base AIII

The power shift is displayed on the temporal graph of the limit cycle of AIII in Fig. 12. Here the blue dots are the places on the limit cycle where a power shift from $Y = 1.63$ to $Y = 1.84$ successfully accesses AV. The red are where the power shift instead accesses

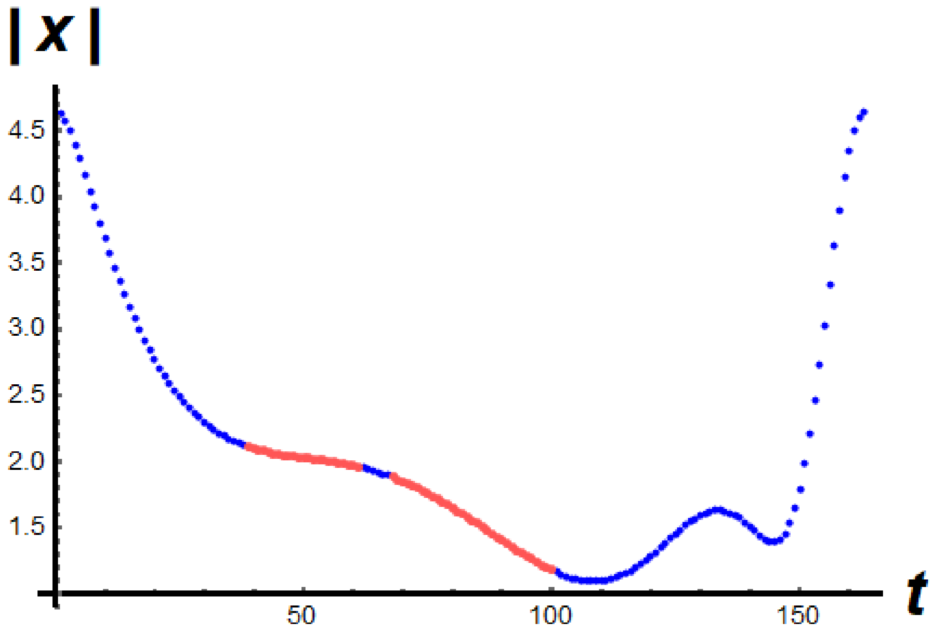


Figure 12: The optimal power shift from AIII to AV. Red is where the system evolves to AII when power shifted from $Y = 1.63$ to 1.84. Blue is where the system evolves to AV instead.

AII. Because limit cycles have closed trajectories, the end of the period lines up with the beginning of the graph making the two blue ends one large tranche.

3.1.4 Time window integrity

While a tranche may be discovered, the basin's fractal nature may mean that there are small discontinuities inside the time window. To check for these potential anomalies, the data set is increased to 2000 random points within the time window. These points are tested to see if any evolve to an attractor other than AV when power shifted. These points are generated by taking 2,000 pseudorandom numbers within the time window and then are used as the TAP for each power shift. This is first done without noise to see the statistical chance of success and then with noise to find a success rate under experimental conditions.

3.1.5 Noise

To simulate realistic conditions, a Gaussian noise is incorporated into the injected signal frequency by modulating $\tilde{\Delta}$ and ϕ. This noise has a full width at half maximum (FWHM) of 0.003 and is added at a rate of 0.2τ. The "WhenEvent" and "NormalDistribution" functions make the Gaussian noise: WhenEvent[Mod[t,0.2]==0, RAR→ NormalDistibution[0,nW] where nW $=$ FWHM/$\sqrt{2 * Ln(2)}$. The results of adding Gaussian noise is shown in Fig. 13. The power shift is the same base and shift as in Fig. 12

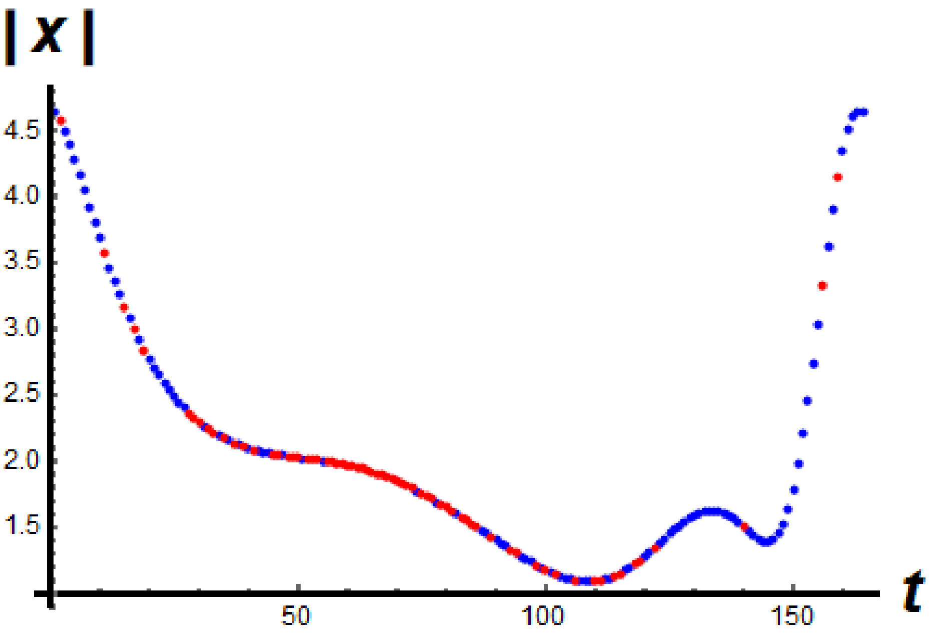


Figure 13: The results of the power shift while adding Gaussian noise with FWHM = 0.003

Integrity test results

Of the 2,000 points taken, all 2,000 went to the expected attractor AV. With Gaussian noise with $FWHM = 0.003$, $1,890$ points still evolve to AV. This shows the tranche holds and is not likely to be within a fractal part of the basin.

3.2 Experimental conditions

3.2.1 Time lag

In an experimental system, the input amplitude does not instantly change; the amplitude changes over some time. To account for this lag time, a scheme for the time lag is developed below:

$$\begin{cases} Y(t) = Y - 0, & t \leq tLag, \\ Y(t) = Y_0 + (t - TAP) * (Y_{Pert} - Y_0)/tLag, & TAP \leq t \leq TAP + tLag, \\ Y(t) = Y_{Pert}, & TAP + tLag \leq t. \end{cases} \tag{3.2.1}$$

The variable $tLag$ is the time it takes for the control parameter to reach its final value. This equation gives the power shift a linear ramp. Figure 14 reveals the results of a linear ramp of $tLag = 15\tau$ on the power shift. By using a lag of 15τ the number of points which go to the desired attractor actually slightly increased. Figure 15 shows, with a 15τ lag, the system falls off 7τ earlier than without the lag and gets back on AV 11τ earlier giving a net increase of 7τ to the tranche.

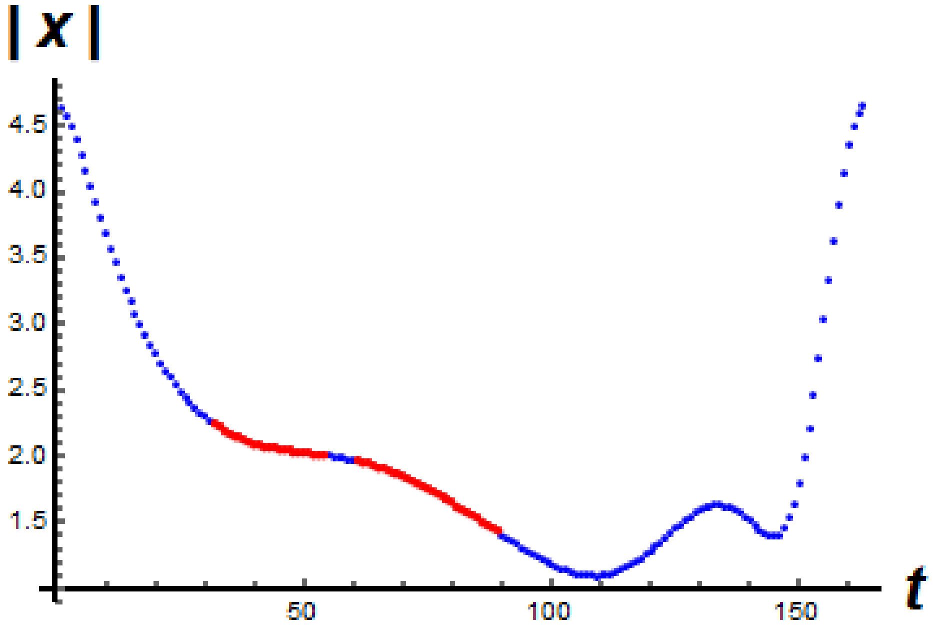


Figure 14: Results of the time lag power shift for a time lag of 15τ

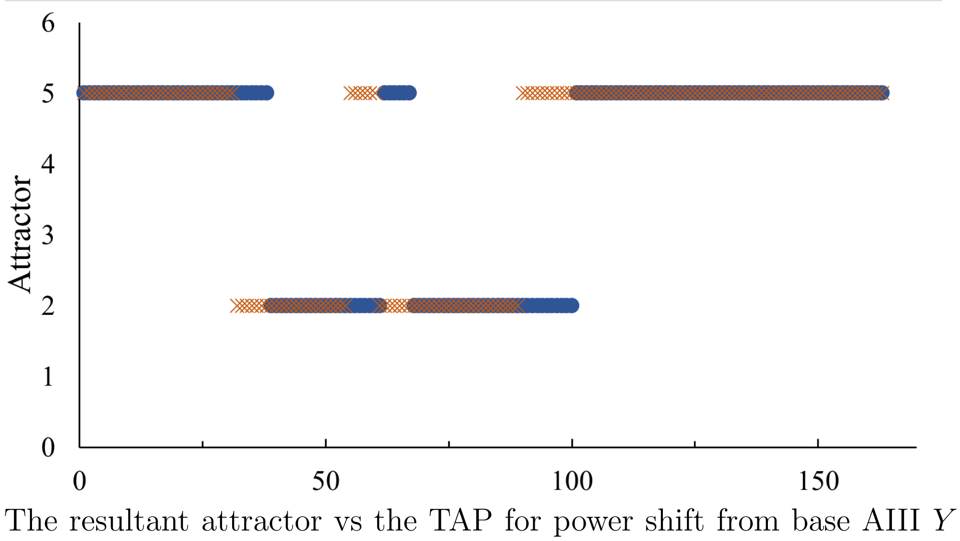


Figure 15: The resultant attractor vs the TAP for power shift from base AIII $Y = 1.63$ to $Y = 1.84$. The blue dots are the results for no lag in the shift and the orange xs are the results for the 15τ time lag

3.2.2 Noise and time lag

Figure 16 shows the results when both Gaussian noise and a time lag are added. There are some points inside the tranche which go to AII instead.

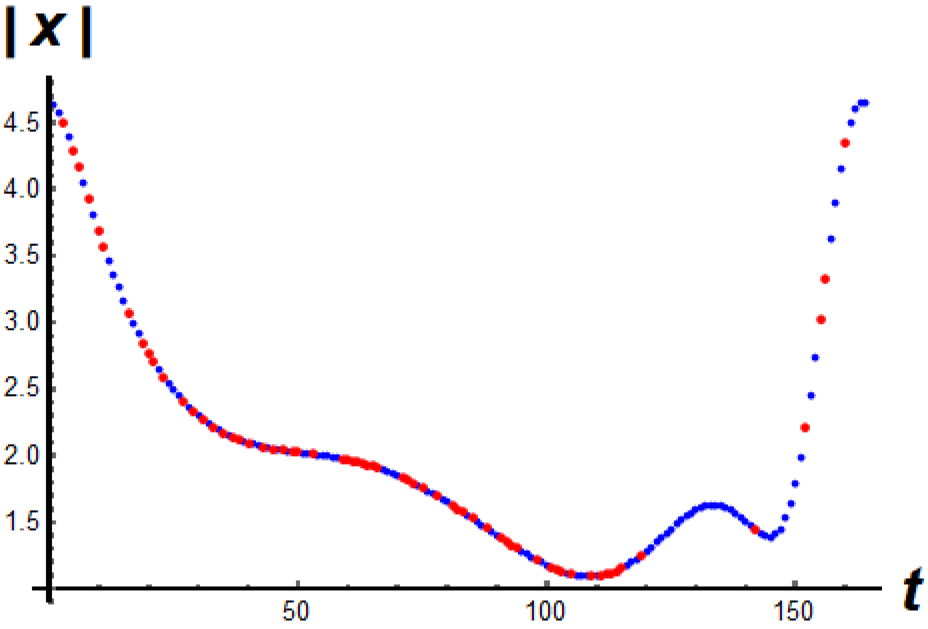


Figure 16: Results of adding Gaussian noise and a 15τ time lag to the power shift

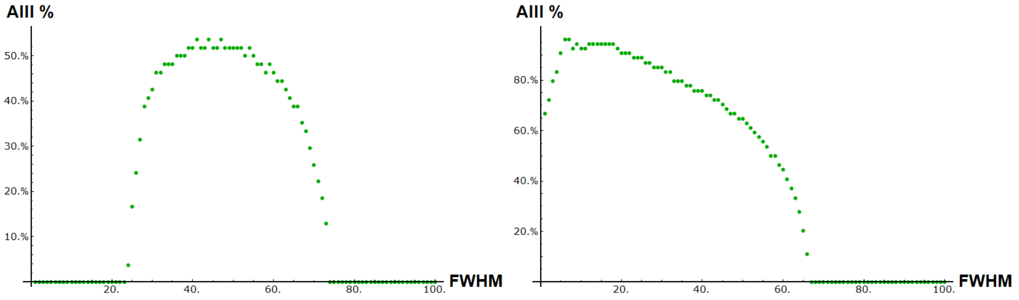
CHAPTER IV

POWER-SHIFT VARIANTS

4.1 Varying the pulse

There may be situations where one wishes to access a coexisting attractor without changing the current control parameter. In these cases, a pulse is more appropriate than a shift. Using a pulse instead of a shift is closer to how Chizhevsky performed his experiments with a pulse-loss perturbation [6]. Chizhevsky used a pulse of $15ns$ duration for his experiment. In this study, three different pulse shapes are tested. These pulse shapes have their FWHM adjusted to find the ideal width in two different tests. The first test keeps the pulse peak constant and the second keeps the area under the curve constant resulting in varying peak heights. The control is a pulse with FWHM$=50\tau$ and peak of 0.16. The percent of the base attractor which the pulse sends the system to AIII is plotted verses the FWHM of the pulse. See Fig. 17b This study uses a Gaussian, a sech^2, and a square pulse. A sech^2 pulse more closely models the shape of certain symmetric type of laser pulses such as passively Q-switched microchip lasers [19].

The pulse's FWHM for all the pulse types is varied to see which width gives the best success rate. The pulse peak stays constant for this while the gain changes by $\Delta C = 0.3$. That is, the gain parameter changes from $C = 3$ to $C = 3.3$. The test results for the Gaussian pulse varying the FWHM is shown in Fig 17a. The abscissa is the pulse's FWHM. The ordinate is the % chance the system goes from base AII to AIII (assuming the pulse is not timed and is therefore at a random spot on the limit cycle). For a constant area under the pulse, Fig. 17b results for the Gaussian pulse. The area is 18.7997; it is calculated using a pulse with peak $\Delta c = 0.3$ and FWHM $= 50\tau$.



(a) constant peak $C = 3.3$

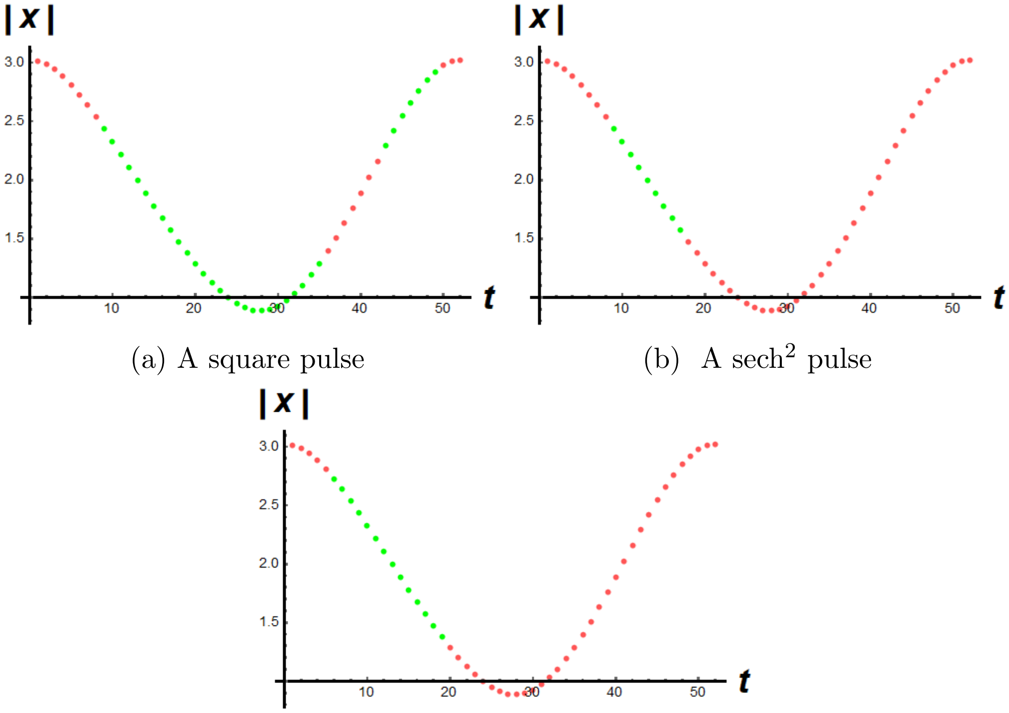
(b) constant area=18.7997

Figure 17: The constant (a) peak and (b) area percentage of times the pulse moved the system from AII to AIII

With a constant pulse, the peak value changes from around 50τ to around 7τ.

4.2 Power-shift pulse

With a square wave pulse the results change to that of Fig. 18a. Figure 18b shows the results of a sech^2 pulse instead. In Fig. 18c, a Gaussian pulse is used to perturb the system. The red dots are where on the limits cycle the gain pulse successfully perturbed the system to AIII.



(a) A square pulse

(b) A sech^2 pulse

(c) A Gaussian pulse

Figure 18: A (a) square pulse with a time lag $= 10\tau$ and peak time $= 50\tau$, and a (b) sech^2, and (c) Gaussian pulse with FWHM$= 50\tau$ and peak at $Y = 1.96$ from $Y = 1.8$.

4.3 Gain-pulse modulation

Another generalization for the power-shift method is to use a different parameter in the shifting process. Out of all the parameters in LIS, the gain parameter is particularly useful for its ease of change experimentally. It is therefore chosen as the next variable to shift. The gain is modulated using different pulse shapes to find which one offers the greatest chance to access attractors. The system ultimately is unchanged because the system parameters (in theory) return to their original state.

4.3.1 Square pulse

The following is a scheme to theoretically create a square pulse:

$$C(t) = \begin{cases} c_0 & t \leq TAP \\ c_0 + \ (c_{pert} - c)(t - TAP)/rmp & TAP \leq t \leq TAP + rmp \\ c_{pert} & TAP + rmp \leq t \leq TAP + len + rmp \\ c_{pert} - (c_{pert} - c)(t - TAP)/rmp & TAP + len + rmp \leq t \leq TAP + len + 2 * rmp \\ c_0, & TAP + len + 2 * rmp \leq t \end{cases} \tag{4.3.1}$$

The original gain value is c_0, c_{pert} is the new value of C, *rmp* is the time it takes the system to go from c_0 to c_{pert} and vice versa, and *len* is the time the system stays at the new gain. Figure 19 Illustrates the square pulse.

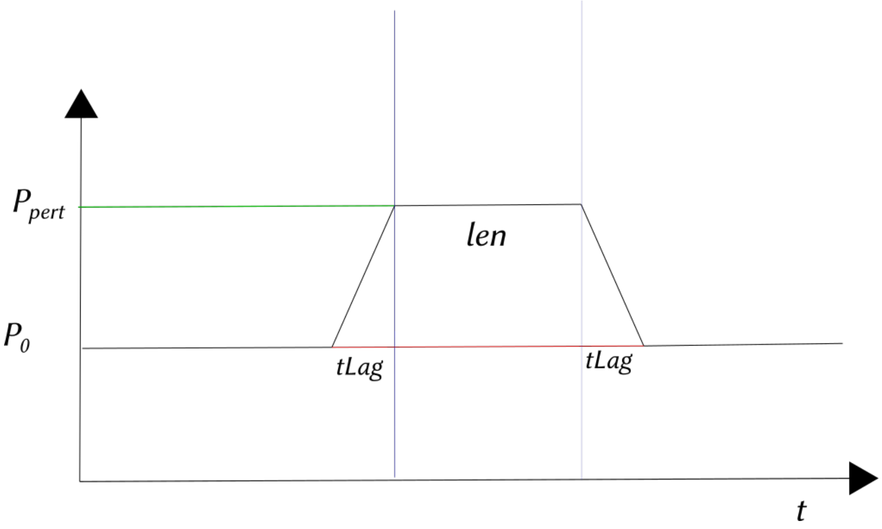


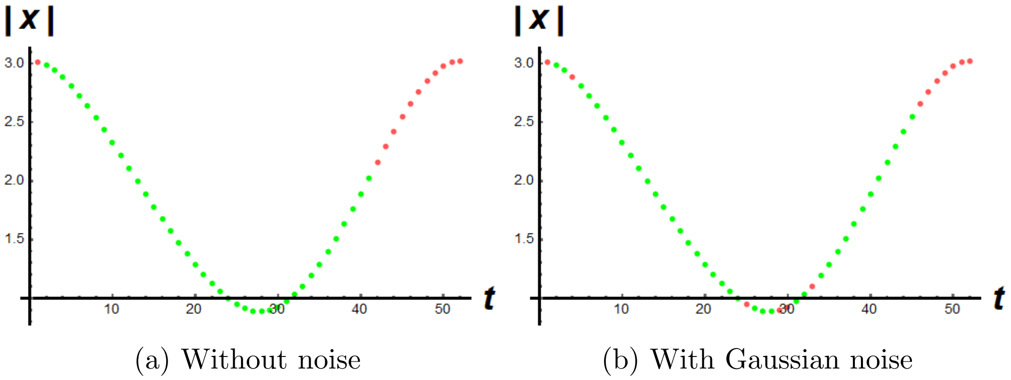
Figure 19: The schematic of the square wave pulse. The abscissa is time, the ordinate is the modulated parameter.

Figure 20a shows the results of a square pulse on AII at $Y = 1.8$. Figure 20b shows the results with a Gaussian noise added. The pulse took 10τ to go from $C = 3$ to $C = 3.3$ where it stayed for 50τ. In this case, the pulse actually perturbed the system to AIII more times than it failed. When Gaussian noise with FWHM $= 0.01$ is applied the system still goes to AIII fairly often. From this we see there is some merit to the square pulse.

4.3.2 Sech^2 pulse

To simulate the gain pulse, the c becomes a time dependent piecewise function. For a gain shift with time lag the function is written as:

$$\begin{cases} C(t) = c_0, & t \leq tLag \\ C(t) = c_0 + (t - TAP) * (c_{Pert} - c_0)/tLag, & TAP \leq t \leq TAP + tLag \\ C(t) = c_{Pert}, & TAP + tLag \leq t \end{cases}$$



(a) Without noise

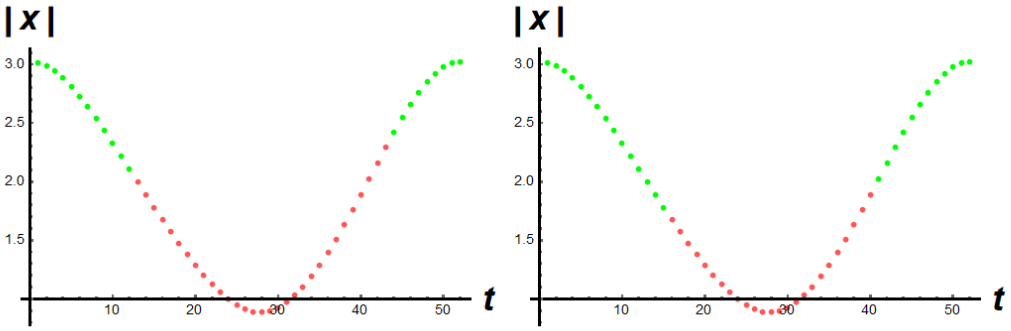
(b) With Gaussian noise

Figure 20: A square pulse from $C = 3$ to $C = 3.3$ (a) without and (b) with noise $= 0.01\tau$ time lag $= 10\tau$ peak duration $= 50\tau$

When a sech^2 pulse is used, the gain becomes the function:

$$C(t) = c_0 + (c_{pert} - c_0)sech(t - TAP - prd)^2. \tag{4.3.2}$$

Note, *prd* is a multiple of the attractor's period. This is done because the program starts the system at $t = 0$ and in order to have a proper Gaussian curve the system needs to start further back. Figure 21a shows the sech^2 gain pulse effect on the system. The results of Gaussian noise on the gain pulse are shown in Fig. 21b. The green dots are where the Gaussian pulse sends the system to AIII. The red are where the system remains on AII.



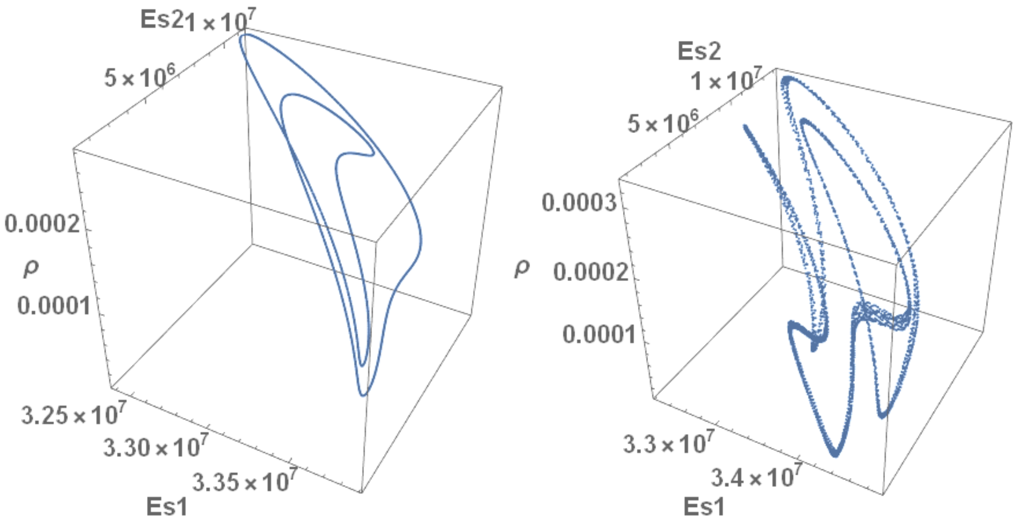
(a) Results for sech^2 pulse on the gain

(b) Results for Gaussian gain pulse

Figure 21: A (a) sech^2 and (b) Gaussian pulse FWHM$= 50\tau$

4.4 Optomechanical power shift

In the optomechanical system, a power shift works the same as in LIS except the variable is the input power, P_{in}, instead. Two different power shifts to the same final power are shown in Figs. 22. The shift in Fig. 22(a) goes from $P_{in} = 0.03$ to $P_{in} = 4$ the system ultimately lands on AIX. Figure 22(b) goes from $P_{in} = 0.035$ to $P_{in} = 4$ and lands on AXIV. The base attractor for these power shifts, AVI,is a stable point at the starting P_{in}. Because it is stable, time does not change the variables.



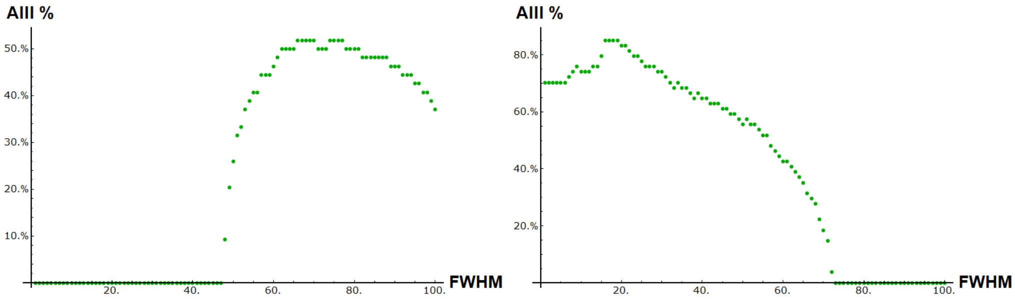
(a) AIX

(b) AXIV

Figure 22: Power shift to $P_{in} = 4$ from (a) $P_{in} = 0.03$ and (b) $P_{in} = 0.035$ in the OM model

4.5 Pulse variation results

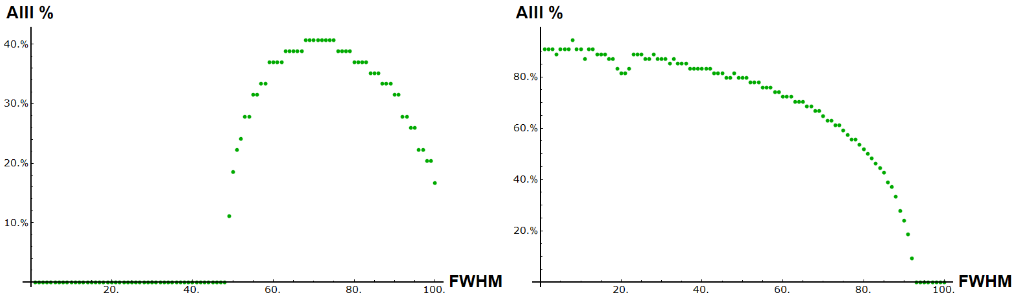
The results for varying the power-shift pulse with constant peaks are shown in Fig. 23a. When the area under the pulse is kept constant, a different pattern emerges as seen in Fig. 23b. The constant area shows higher percent success with short pulses compared to the constant peaks. A sech^2 pulse version is shown in Fig. 24a When the area under the pulse is kept constant, the results change to Fig. 24b. The results are similar to the Gaussian only with a lower % chance of success.



(a)

(b)

Figure 23: Results of the power-shift Gaussian pulse varying the FWHM of the pulse keeping (a) the peak the same and (b) keeping the peak a constant area of 10.0265 under the curve



(a) With constant peak (b) With constant area under the curve

Figure 24: A sech^2 pulse (a) with a constant peak and (b) with constant area

CHAPTER V

APPLICATIONS AND EXAMPLES

5.1 Conclusions

5.1.1 Power-shift pulse effectiveness

Of the three pulses, the square wave pulse shows the highest probability of moving the system to a new attractor. The Gaussian pulse was more effective than the sech^2 pulse. From here it seems that the more abrupt the change for the given pulse, the more effective it is for changing attractors.

5.1.2 Gaussian noise effects

Small amounts of Gaussian noise on the frequency does not destroy the larger attractors. The power shift and pulse variants are still relevant with noise but but there is some reduction in the reliability. Namely, the system prefers AII as more noise is added until the other attractors vanish. The smaller domain attractors vanish before the larger ones making them less likely to appear in experiment.

5.2 Experimental methods

To perform the power-shift method experimentally, an experimentalist needs to use a computer setup to control the laser system. An electro-optic modulator can shift the power at speeds faster than the time lag used in this study. The computer can record the temporal output via an oscilloscope. To discover attractors, some software can compare the output of two different sequences and find any discrepancies between the two outputs which implies coexisting attractors.

5.3 Roadmap to access attractors

One can create a "roadmap" to access system attractors using only a power shift. Table 1 is an example of one using the LIS system. The ± represents the maximum allowable time uncertainty in which the power shift is viable. Table 1 reads as the instructions on how to access any of the eight known attractors. It is read simply by combining the header with the attractor row of interest. If, for example, one wished to land on AVI using Table 1, one would read the row and follow the instructions. According to Table 1: To land on AVI start on AIII at $Y_0 = 1.7$ with period 159τ. Then at TAP $= 15 \pm 12\tau$, shift to $Y = 1.96$.

Table 1: Roadmap to access the LIS attractors

To land on:	Start on	At $Y_0 =$	with has a period	Then at TAP:	Shift Y to:
AI	Any	Any	$---$	Any	≤ 1
AII	Any	Any	$---$	Any	1.54
AIII	AII	1.54	108τ	$7 \pm 49\tau$	1.8
AIV	AI	0.854	63τ	$3 \pm 4\tau$	1.114
AV	AIII	1.7	159τ	$154 \pm 26\tau$	1.84
AVI	AIII	1.7	159τ	$15 \pm 12\tau$	1.96
AVII	AIII	1.8	157τ	$138 \pm 2\tau$	1.269
AVIII	AIII	1.7	159τ	$126.5 \pm 2\tau$	1.272

5.4 Power shift vs. pulse variants

A power shift relies on the dynamics of the attractors in the system to access other attractors. The gain pulse shows success comparable to that of the injected signal amplitude pulse. This implies parameters other than the standard control parameter are viable for shifting.

5.5 OM Power shift from steady states

In the optomechanical model, there are some steady-state domains. This means once on these points the system remains. For this case, timing becomes irrelevant in the OM system. Table 2 gives an example of how to use the power-shift method to go from a steady state to an attractor. As done is Table 1, instructions are given on how to acquire an attractor by starting somewhere in phase space and power shifting to a different value. If one wishes, for example, to access AVII, Table 2 states: To land on AVII, start on AVI at $P_{0in} = 0.555$ then shift P_{in} to $P_{in} = 3.1$.

Table 2: Attractor roadmap for OM system starting from a steady state, then power shifting to an attractor.

To land on Attractor:	Start on	At P_{0in}=	then shift P_{in} to:
AVI	Any	Any	1
AII	Any	Any	2
AIX	AVI	0.03	4
AXIV	AVI	0.035	4
AIV	AVI	0.08	3.616
AV	AVI	0.005	3
AVII	AVI	0.555	3.1
AX	AIV	5.33	16.46

The drawback to using steady states is that they are zero-volume attractors. This means finding a steady state within a specific attractor's basin is difficult. In contrast a limit cycle with a large volume is statistically more likely to have tranches in a specific basin.

5.6 Results of shifting off limit cycles

The power-shift method is shown to be capable of discovering and reliably accessing attractors in a laser with injected signal model. It also survives reasonable amounts of Gaussian noise and time lags. The pulse version, in general, is shown to be effective in changing to a different attractor provided the right amplitude and FWHM is used. A Gaussian pulse is found to generally be better at shifting attractors compared to a sech^2 pulse with similar peaks. A square wave pulse is more effective than the Gaussian. In essence, short intense pulses appear more effective than slow shallow pulses. The gain pulse variant shows promise with a square wave pulse.

One of the more interesting results is how the limit cycles possess tranches of points on the limit cycle (large time windows) where the system moves to only one attractor. This seems to contradict the fractal nature of the basin of attraction. A reason could be: while the basin is fractal, the basins still have large areas which are contiguous. Additionally, the attractor shape may actually conform to the basin's general shape in higher dimensions. This could be because the five dimensional shape of the LIS attractors have a similar shape to the 5 dimensional basin. The continuous sections are beneficial for experiment because one can reliably access attractors without worrying about fractal basins.

REFERENCES

[1] Wenjuan Wu, Zengqiang Chen, and W. H. Ip, *Complex nonlinear dynamics and controlling chaos in a Cournot duopoly economic model*, Nonlinear Analysis: Real World Applications **11** (2010), no. 5, 4363–4377.

[2] Edward Ott, Celso Grebogi, and James A. Yorke, *Controlling chaos*, Physical Review Letters **64** (1990), no. 11, 1196–1199.

[3] K. Pyragas, *Continuous control of chaos by self-controlling feedback*, Physics Letters A **170** (1992), no. 6, 421–428.

[4] A. N. Pisarchik, *Controlling the multistability of nonlinear systems with coexisting attractors*, Physical Review E **64** (2001), no. 4, 046203.

[5] Chunbiao Li and J. C. Sprott, *Finding coexisting attractors using amplitude control*, Nonlinear Dynamics **78** (2014), no. 3, 2059–2064.

[6] V. N. Chizhevsky, *Coexisting attractors in a CO2 laser with modulated losses*, Journal of Optics B: Quantum and Semiclassical Optics **2** (2000), no. 6, 711–717.

[7] D. K. Bandy, J. R. Hall, and M. E. Denker, *Predicting the evolutionary dynamic behavior of a laser with injected signal using Lyapunov exponents*, Physical Review A **92** (2015), no. 1, 013841.

[8] E. K. T. Burton, J. R. Hall, D. M. Chapman, and D. K. Bandy, *Shifts in control parameter dynamically access individual attractors in a multistable system*, Nonlinear Dynamics **105** (2021), no. 2, 1877–1883.

[9] J. R. Hall, E. K. T. Burton, D. M. Chapman, and D. K. Bandy, *Experimentally Viable Techniques for Accessing Coexisting Attractors Correlated with Lyapunov Exponents*, Applied Sciences **11** (2021), no. 21, 9905.

[10] Eric W. Weisstein, *Attractor*, https://mathworld.wolfram.com/Attractor.html.

[11] Herbert Goldstein, Charles Poole, and John Safko, *Classical Mechanics*, 3rd ed., Addison Wesley, 2002.

[12] D. K. Bandy, E. K. T. Burton, J. R. Hall, D. M. Chapman, and J. T. Elrod, *Predicting attractor characteristics using Lyapunov exponents in a laser with injected signal*, Chaos: An Interdisciplinary Journal of Nonlinear Science **31** (2021), no. 1, 013120.

[13] Neal B. Abraham and Lorenzo M. Narducci, *Laser Physics & Laser Instabilities*, World Scientific Publishing Company, Singapore, September 1988.

[14] J. R. Tredicce, F. T. Arecchi, G. L. Lippi, and G. P. Puccioni, *Instabilities in lasers with an injected signal*, Journal of the Optical Society of America B **2** (1985), no. 1, 173.

[15] Dimitar K. Dimitrov and Juan Manuel Peña, *Almost strict total positivity and a class of Hurwitz polynomials*, Journal of Approximation Theory **132** (2005), no. 2, 212–223.

[16] A. T. Rosenberger, *Slow light in an optomechanical microresonator system*, SPIE OPTO (San Francisco, California, United States) (Selim M. Shahriar and Jacob Scheuer, eds.), February 2017, p. 1011914.

[17] Limu Ke, *THE DYNAMICS OF CROSS-POLARIZATION COUPLING IN A SINGLE WHISPERING-GALLERY MICRORESONATOR*, Ph.D. thesis, Oklahoma State University, 2019.

[18] *Model developed by Professor Rosenberger.*

[19] T. Erneux, P. Peterson, and A. Gavrielides, *The pulse shape of a passively Q-switched microchip laser*, The European Physical Journal D **10** (2000), no. 3, 423.

[20] John C. Oxtoby, *The Poincaré Recurrence Theorem*, Measure and Category: A Survey of the Analogies between Topological and Measure Spaces (John C. Oxtoby, ed.), Graduate Texts in Mathematics, Springer US, New York, NY, 1971, pp. 65–69.

[21] George Contopoulos, *Attractors in conservative systems*, International Journal of Bifurcation and Chaos in Applied Sciences and Engineering **16** (2006).

[22] Mac E. Van Valkenburg and Wendy M. Middleton (eds.), *Reference Data for Engineers: Radio, Electronics, Computers and Communications*, 9th edition ed., Newnes, Boston Oxford, October 2001.

[23] S Zafar Ali, *First observation of Quasi-Chaos in Erbium doped fiber ring laser*, 12.

[24] Donna K. Bandy, Luigi A. Lugiato, and Lorenzo M. Narducci, *Coexisting attractors in a laser with an injected signal*, Journal of the Optical Society of America B **2** (1985), no. 1, 148.

APPENDICES

Attractor information

In order for a nonlinear system to contain attractors, energy must not be conserved. This is because according to Ref. [20] a conservative system always returns to or within some neighborhood of the starting point. Because of this, a conservative system does not typically asymptotically approach an attractor. The only exception is when a conservative system escapes and approaches infinity. In this case, infinity acts as the attractor [21]. If the system is only dissipative, then the attractor is a single point. In order for the attractor to have a non-zero volume, it must be driven and dissipative.

Limit cycles

A limit cycle is an attractor which has repeating amplitudes and a characteristic frequency regardless of the initial conditions used to reach it [22]. The van der Pol equation with $\epsilon > 0$ yields a limit cycle attractor. The limit cycle's trajectory is closed in phase space. A limit cycle in phase space is shown in Fig. 25. To get the full image of the attractor,

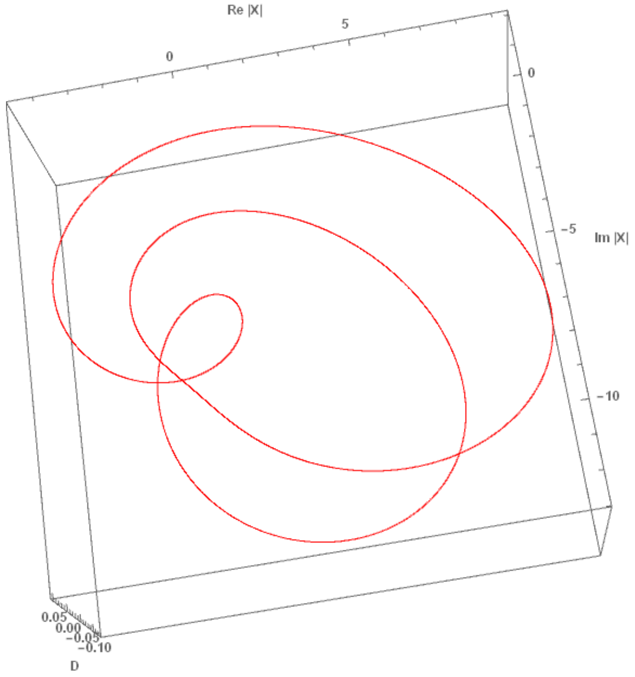
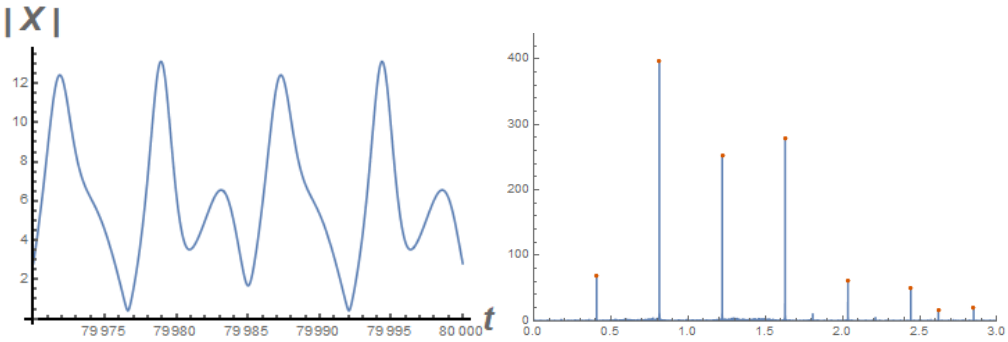


Figure 25: An example of a limit cycle in phase space. The points in red are the values that the system cycles through as it evolves.

other methods are used as well including a temporal and Fourier series shown in Figs. 26. A



(a) Temporal

(b) Fourier

Figure 26: The (a) temporal and (b) Fourier series for the limit cycle shown in Fig. 25

temporal solution is a graph showing the values of a variable with respect to time. A Fourier series breaks down the attractor's temporal dynamic into a series of sines and cosines.

Coexisting attractors

It is possible for a system to possess two or more attractors for the same set of parameters. The system will show different dynamics when the only difference is different initial conditions. When a system has multiple possible attractors, it is said to possess coexisting attractors or have coexistence. Figure 27 shows an example of two coexisting attractors.

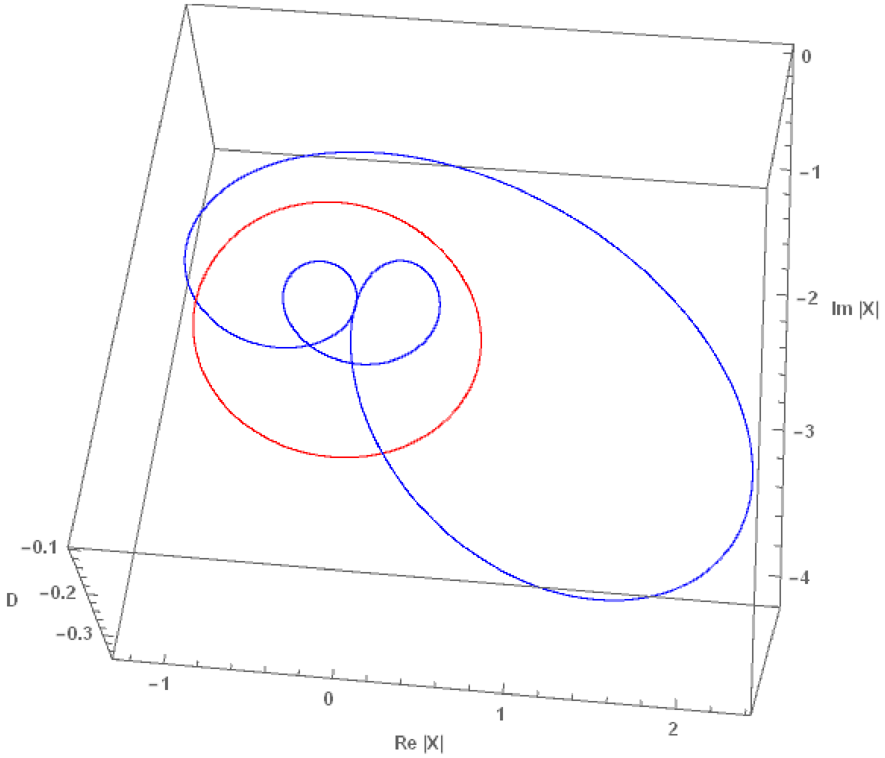


Figure 27: An example of coexisting attractor in LIS phase space, AII and AIII

Poincaré region

Some attractors have complicated structures which require additional methods to analyze. In those cases a Poincaré map is useful. A Poincaré map is created by tracing the points which pass through a selected plane in phase space while the system evolves [11].

Strange attractors

An interesting type of dynamic that shows up in nonlinear systems is called a strange attractor. If one looks at the phase space, temporal, and Fourier series for a strange attractor, it looks chaotic i.e a messy phase space and a broadband spectrum in the Fourier series. A Poincaré in the phase space however, reveals a structure. The Poincaré of a strange attractor is in green in Fig. 28. This structure constitutes a set of points where the system is attracted to. This means, by the prior definition, a strange attractor is in fact an attractor despite being chaotic. Note, not all chaos is an attractor. The chaos found in conservative systems is not an attractor as it does not asymptotically approach a set of points.

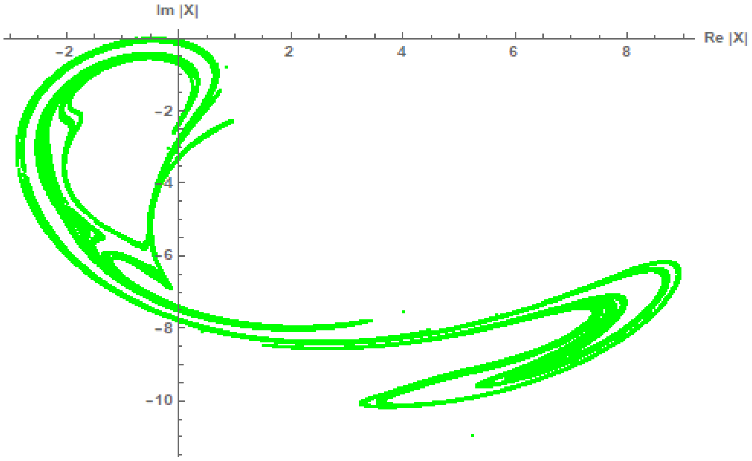


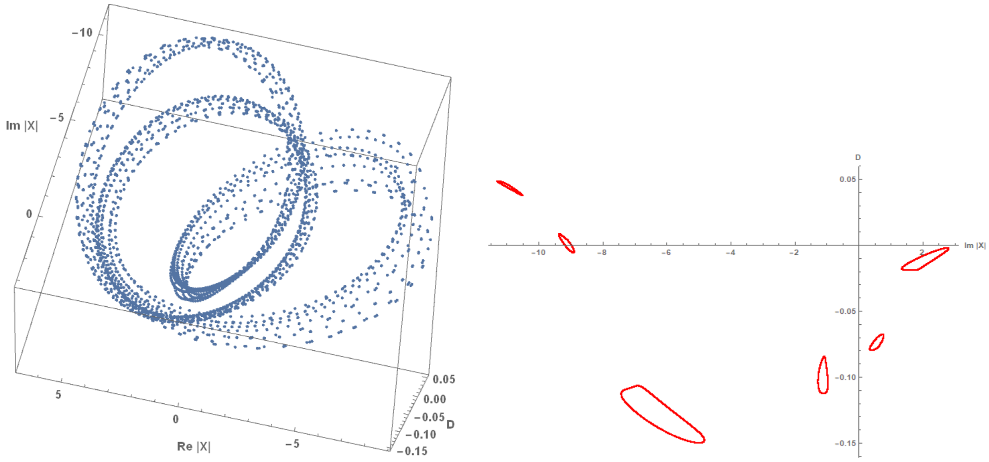
Figure 28: An example of a strange attractor in green. The axes are the Real and imaginary components of X and the $D = 0.05$.

Quasichaos

Unlike regular chaos, quasichaos may possess overriding frequencies and converges to a certain trajectory regardless of initial conditions [23]. Quasichaos still has positive Lyapunov exponents but is typically of smaller value than “full-blown” chaos.

Tori

Another attractor type is called a torus. A torus is an attractor of a higher dimension than a limit cycle. A limit cycle can be mapped to a single dimension. In this dissertation, time is the dimension used. In a torus, there is an extra frequency which is incommensurate with the fundamental frequency. This results in a non closed trajectory. A torus needs at least two coordinates to map. Figure 29 shows the phase space and Poincaré map of a torus



(a) Phase Space

(b) Poincaré map

Figure 29: An example of the (a) phase space and (b) Poincaré map of a torus attractor

attractor found in LIS. This attractor exists for a different set of parameters than the one used in this investigation. It is found using the parameters in Ref. [24]. Note, instead of dots there are ellipses where the system crosses the plane.

In regards to the Power-shift method, a torus does not make for a good base attractor because it lacks the timing reliability of a limit cycle due its lack of a closed trajectory.

Periodic windows

Within the chaotic domain there are small regions of the control parameter where the chaos becomes periodic. These are called windows of periodicity or periodic windows. The periodic windows are not labeled as attractors in Fig. 5 but instead as windows WI and WII. By the definition used in this study periodic windows are a type of attractor. However, for the sake of clarity they are referred to as windows to distinguish them from the regular limit cycles and tori found.

Programs

Fourier analysis

```
Case = 2;

Attractor = 3;

YLoad = 1.65;

YInitial = 1.65;
```

YMax = 1.68;

YMin = 1.64;

YStep = 0.01;

thres = 0.5;

Y = YInitial;

Upload = True;

wthywtni = {{Y, “Ω Zero”}};

δv = 0;

$A_0 = \delta$v;

$B_0 = \delta$v;

$R_0 = \delta$v;

$S_0 = \delta$v;

$d_0 = \delta$v;

Δt = 0.02;

$t_{\text{limit}} = 30000$;

If[Upload,

file =

ToExpression[

Import[“Lis System\\Case ” <> ToString[Case] <> “\\Attractor ”

<> ToString[Attractor] <> “\\SteadyState Y=” <> ToString[YLoad] <>

“.txt”]];

$A_0 = \mathrm{file}[[1]];$

$B_0 = \mathrm{file}[[2]];$

$R_0 = \mathrm{file}[[3]];$

$S_0 = \mathrm{file}[[4]];$

$d_0 = \mathrm{file}[[5]];$

,

$\{A_0, B_0, R_0, S_0, d_0\}]$;

Print[file];

If[Case $== 1$,

$c = 20.0;$

$K = 0.5;$

$\phi = -1.0;$

$\gamma = 0.05;$

$\Delta = 1.0];$

If[Case $== 1.1$,

$c = 21.0;$

$K = 0.5;$

$\phi = -1.0;$

$\gamma = 0.05;$

$\Delta = 1.0];$

If[Case $== 2$,

$c = 3.0;$

$K = 0.1;$

$\phi = -0.5 * K;$

$\gamma = 0.01;$

$\Delta = 0.5];$

If[Case $== 2.1$,

```
c = 3.1;
K = 0.1;
φ = −0.5 ∗ K;
γ = 0.01;
Δ = 0.5];
If[Case == 2.2,
c = 3.2;
K = 0.1;
φ = −0.5 ∗ K;
γ = 0.01;
Δ = 0.5];
If[Case == 3,
c = 3.0;
K = 0.1;
φ = −0.5 ∗ K;
γ = 4.0 ∗ 10^ − 4;
Δ = 0.5];
If[Case == 4,
c = 3.0;
K = 5.24;
φ = −1.0;
γ = 1.0 ∗ 10^ − 4;
Δ = 1.0];

While[YMin<=Y<=YMax,
solution = NDSolve[{
$A'[t] == -K * \left(A[t] + \left(B[t] * \frac{\phi}{K}\right) - Y + (2 * c * R[t])\right)$ ,
$B'[t] == -K * \left(B[t] - \left(A[t] * \frac{\phi}{K}\right) + (2 * c * S[t])\right)$ ,
$R'[t] == -(R[t] - (\Delta * S[t])) + (A[t] * d[t])$,
$S'[t] == -((\Delta * R[t]) + S[t]) + (B[t] * d[t])$,
$d'[t] == -\gamma * ((A[t] * R[t]) + (B[t] * S[t]) + d[t] + 1)$,
$A[0] == A_0$,
$B[0] == B_0$,
$R[0] == R_0$,
$S[0] == S_0$,
$d[0] == d_0$
}, $\{A, B, S, R, d\}, \{t, 0, t_{\text{limit}}\}$ , PrecisionGoal → ∞, MaxSteps → ∞,
MaxStepSize → Δt,
Method → "StiffnessSwitching"];

AA[t_]:=$A[t]$/.solution;
BB[t_]:=$B[t]$/.solution;
RR[t_]:=$R[t]$/.solution;
SS[t_]:=$S[t]$/.solution;
dd[t_]:=$d[t]$/.solution;

$X$[t_]:=$\sqrt{\text{AA}[t]^2 + \text{BB}[t]^2}$/.solution;
$P$[t_]:=$\sqrt{\text{RR}[t]^2 + \text{SS}[t]^2}$/.solution;
Print [Plot [$X[t], \{t, t_{\text{limit}} - 400, t_{\text{limit}}\}$ , PlotRange → All,
AxesLabel → {Style[$t$, Large, Bold], Style[$|X|$ ,
Large, Bold]}, AxesStyle → Thick]];
```

```
datas = Table[{AA[t][[1]], BB[t][[1]], dd[t][[1]]},
{t, t_limit − 4000, t_limit(*,Δt*)}] ;
Print[ListPointPlot3D[datas, AxesLabel → {“Re |X|”, “Im |X|”, “D ”},
BaseStyle →
{FontWeight → “Bold”, FontSize → 18}, PlotRange → Full]];


t2 = t_limit;
t1 = t2/2;
xtabtemp = Table[X[t][[1, 1]], {t, t1, t2 − 1}];
FX = Flatten[Abs[Fourier[xtabtemp]]];
FX = Drop[Table[{N[(n − 1)/(t2 − t1)] ∗ 2Pi, Extract[FX, {n}]},
{n, 0, Length[FX] − 1}], 2];


fouriertab = {};
top = Max[Table[FX[[it, 2]], {it, 1, Length[FX]}]] ∗ 1.1;
temp1 = 0;
temp2 = 0;
temp3 = 0;
thres = 0.01;
ofx = {Y};
n = 2; While[n ≤ Length[FX] − 1,
temp1 = Extract[FX, {n, 2}];
temp2 = Extract[FX, {n + 1, 2}];
temp3 = Extract[FX, {n − 1, 2}];
If[temp1 > temp2&&temp3 < temp1,
ptn = ListPlot[{Tooltip[Extract[FX, n]]},
PlotStyle → {PointSize[0.009], RGBColor[.847, 0.347, 0]}];
If[temp1 > thres, fouriertab = Append[fouriertab, ptn];
If[temp1 > temp2&&temp3 < temp1, AppendTo[ofx, Extract[FX, {n, 1}]],
False], False], False];

n++];
(*AppendTo[wthywtni, ofx]; *)
Print[Show[{ListLinePlot[FX, PlotRange → {{0, 0.5}, {0, top}}],
fouriertab}]];

Print[“Y=” <> ToString[Y] <>
“ Case=” <> ToString[Case]];


Print[“c = ” <> ToString[c] <>
“; Y = ” <> ToString[Y] <>
“; K = ” <> ToString[K] <>
“; ϕ = ” <> ToString[ϕ] <>
“; γ = ” <> ToString[γ] <>
“; Δ = ” <> ToString[Δ] <>
“; δv = ” <> ToString[δv] <>
“; Time Range(t1)=” <> ToString[t1]];


t_stop = t_limit;


Export[“Lis System\\Case ” <> ToString[Case] <> “\\Attractor ” <>
```

```
ToString[Attractor] <> "\\
SteadyState Y=" <> ToString[Y] <> ".txt",
Table [{A[t], B[t], R[t], S[t], d[t]}, {t, t_stop, t_stop, 1}] /.
solution[[1]]];



A_0 = First [A [t_stop] /.solution] ;
B_0 = First [B [t_stop] /.solution] ;
R_0 = First [R [t_stop] /.solution] ;
S_0 = First [S [t_stop] /.solution] ;
d_0 = First [d [t_stop] /.solution] ;
Y = Y + YStep;
];
(*wthywtni = Append[wthywtni, {"..."}];
Export["Lis System\\Case " <> ToString[Case] <> "\\Attractor " <>
ToString[Attractor] <> "\\Excel Data\\Table of Y Omega sub 0.xlsx",
wthywtni]; *)
```

Attractor finder code

```
Case = 2;
chain = {{y, a, b, r, s, d}};
δv = 0;
A_0 = δv;
B_0 = δv;
R_0 = δv;
```

```
$S_0 = \delta\mathrm{v}$;
$d_0 = \delta\mathrm{v}$;

If[Case == 1,
$c = 20.0$;
$K = 0.5$;
$\phi = -1.0$;
$\gamma = 0.05$;
$\Delta = 1.0$];
If[Case == 1.1,
$c = 21.0$;
$K = 0.5$;
$\phi = -1.0$;
$\gamma = 0.05$;
$\Delta = 1.0$];
If[Case == 2,
$c = 3.0$;
$K = 0.1$;
$\phi = -0.5 * K$;
$\gamma = 0.01$;
$\Delta = 0.5$];
If[Case == 2.1,
$c = 3.1$;
$K = 0.1$;
$\phi = -0.5 * K$;
$\gamma = 0.01$;
$\Delta = 0.5$];
```

```
If[Case == 3,
c = 3.0;
K = 0.1;
ϕ = −0.5 ∗ K;
γ = 4.0 ∗ 10^ − 4;
Δ = 0.5];
If[Case == 4,
c = 3.0;
K = 5.24;
ϕ = −1.0;
γ = 1.0 ∗ 10^ − 4;
Δ = 1.0];


YMin = 0;
YMax = 2;
steps = 2000;
Ybase = Table[YMin + i ∗ (YMax − YMin)/steps, {i, 0, steps}];
YChain = RandomSample[Ybase];
Print[YChain[[5]]];
tLimit = 20000;
Δt = 0.02;
n = 0;
runNum = 1;
While[runNum ≤ 100,
If[FileExistsQ[“Lis System\\Case ” <> ToString[Case] <> “\\MC\\MC Run ” <>
ToString[runNum] <> “ Y From ” <> ToString[YMin] <> “ to ” <> ToString[YMax]
<> “ Steps=” <> ToString[steps] <> “ tlim=” <> ToString[tLimit] <> “.xlsx”],
```

runNum++, Print[“Run num=” <> ToString[runNum]

<> “.”]; Break[]]];

While[n < Length[YChain],

n++;

Y = YChain[[n]];

Print[“Y=” <> ToString[N[Y]] <> “.”];

(*Run NDSolve here and collect new ics at *)

solution = NDSolve[{

$A'[t] == -K * \left(A[t] + \left(B[t] * \frac{\phi}{K}\right) - Y + (2 * c * R[t])\right),$

$B'[t] == -K * \left(B[t] - \left(A[t] * \frac{\phi}{K}\right) + (2 * c * S[t])\right),$

$R'[t] == -(R[t] - (\Delta * S[t])) + (A[t] * d[t]),$

$S'[t] == -((\Delta * R[t]) + S[t]) + (B[t] * d[t]),$

$d'[t] == -\gamma * ((A[t] * R[t]) + (B[t] * S[t]) + d[t] + 1),$

$A[0] == A_0,$

$B[0] == B_0,$

$R[0] == R_0,$

$S[0] == S_0,$

$d[0] == d_0$

}, {A, B, S, R, d}, {t, 0, tLimit}, PrecisionGoal → ∞, MaxSteps → ∞, MaxStepSize → Δt];

AA[t_]:=$A[t]$/.solution;

BB[t_]:=$B[t]$/.solution;

RR[t_]:=$R[t]$/.solution;

```
SS[t_]:=S[t]/.solution;
dd[t_]:=d[t]/.solution;

X[t_]:=$\sqrt{\mathrm{AA}[t]^2+\mathrm{BB}[t]^2}$/.solution;
P[t_]:=$\sqrt{\mathrm{RR}[t]^2+\mathrm{SS}[t]^2}$/.solution;
Print[Plot[X[t], {t, tLimit − 1500, tLimit}, PlotRange → All, AxesLabel →
{Style[t, Large, Bold],
Style[|X| , Large, Bold]}, AxesStyle → Thick]];
datas = Table[{AA[t][[1]], BB[t][[1]], dd[t][[1]]}, {t, tLimit − 4000, tLimit(*,Δt*)}];
Print[ListPointPlot3D[datas, AxesLabel → {“Re |X|”, “Im |X|”, “D ”}, BaseStyle →
{FontWeight →
“Bold”, FontSize → 18}, PlotRange → Full]];
$A_0$ = First[A[tLimit]/.solution];
$B_0$ = First[B[tLimit]/.solution];
$R_0$ = First[R[tLimit]/.solution];
$S_0$ = First[S[tLimit]/.solution];
$d_0$ = First[d[tLimit]/.solution];

link = {Y, $A_0$, $B_0$, $R_0$, $S_0$, $d_0$} ;
AppendTo[chain, link];
(*loop end*)
];
Print[chain];
CreateDirectory[“Lis System\\Case ” <> ToString[Case] <> “\\MC”];
Export[“Lis System\\Case ” <> ToString[Case] <> “\\MC\\MC Run ” <>
ToString[runNum] <> “ Y From ” <> ToString[YMin] <> “ to ” <>
ToString[YMax] <> “ Steps=” <> ToString[steps] <> "
```

```
 tlim=" <> ToString[tLimit] <> “.xlsx”, chain];
(*Exportthechainwiththenumberintheexcelfile′stitle*)
```

Power shift codes

LIS Power shift

```
Clear[“Global`*”]
Case = 2;
Attractor = 3;
YLoad = 1.63;
YInitial = 1.63;
AttractorR = 5;
AttractorL = 2;
thres = 0.5;
Y = YInitial;
YPert = 1.84;

runN = 2000;
tLag = 15;
tR = 49;

outPut = True; (*set if you want to see output to check phase space*)
SetSharedVariable[attrat]
(*switched from the test Y vary*)
attrat = {{“TAP”, “Att ”}};
Upload = True;
```

```
wthywtni = {{Y, “Ω Zero”}};
δv = 0;
A_0 = δv;
B_0 = δv;
R_0 = δv;
S_0 = δv;
d_0 = δv;
thr = 0.15;
Δt = 0.02;
tLimit = 30000;
tPert = 1490;
nW0 = 0.003
nW = nW0/Sqrt[2 ∗ Log[2]];
nWeight = 1;
nRate = 0.2;
noiseT = 20000;
num = 1;
(*make sure not to overwrite previous runs*)
While[num ≤ 100,
If[FileExistsQ[“Lis System\\Case ” <> ToString[Case] <> "\\Power shift data\\run
" <> ToString[num] <> “ FN GNoise Power shift from Attractor ” <> ToString[Attractor]
<> “ for Y = ” <> ToString[Y] <> “ to ” <> ToString[YPert] <> “ tLag=” <>
ToString@tLag <> “ gnWeight=” <> ToString[nW0] <> “.xlsx”], num++; ,
Print[“run number ” <> ToString[num] <> “.”]; Break[]]];
If[num ≥ 101, Print[“too many runs”]; Exit[], False];
If[Upload,
file = ToExpression[Import[“Lis System\\Case ” <> ToString[Case] <> “\\Attractor ”
<> ToString[Attractor] <> “\\SteadyState Y=” <> ToString[YLoad] <> “.txt”]];



A_0 = file[[1]];
B_0 = file[[2]];
R_0 = file[[3]];
S_0 = file[[4]];
d_0 = file[[5]];

file = ToExpression[Import[“Lis System\\Case ” <> ToString[Case] <> "\\Attractor
 " <> ToString[AttractorR] <> “\\SteadyState Y=” <> ToString[YPert] <> “.txt”]];
A_1 = file[[1]];
B_1 = file[[2]];
R_1 = file[[3]];
S_1 = file[[4]];
d_1 = file[[5]];
prd = ToExpression[Import[“Lis System\\Case ” <> ToString[Case] <> "\\Attractor
 " <> ToString[AttractorR] <> “\\Periods\\Y=” <> ToString[YPert] <> “.txt”]];
Print[prd];
prd0 = ToExpression[Import[“Lis System\\Case ” <> ToString[Case] <> "\\Attractor
 " <> ToString[Attractor] <> “\\Periods\\Y=” <> ToString[YLoad] <> “.txt”]];
Print[prd0];
,
{A_0, B_0, R_0, S_0, d_0}] ;
TAP0 = 7;
If[tR ≥ TAP0,
TAP0+=prd0; , False];


If[Case == 1,
c = 20.0;
k = 0.5;
ϕ = −1.0;
γ = 0.05;
Δ = 1.0];
If[Case == 2,
c = 3.0;
k = 0.1;
ϕ = −0.5 ∗ k;
γ = 0.01;
Δ = 0.5];
If[Case == 3,
c = 11.6883;
k = 1.0 ∗ 10^ − 3;
ϕ = −5.0 ∗ 10^ − 4;
γ = 1.0 ∗ 10^ − 5;
Δ = 5.0 ∗ 10^ − 4];
If[Case == 4,
c = 3.0;
k = 5.24;
ϕ = −1.0;
γ = 1.0 ∗ 10^ − 4;
Δ = 1.0];
```

```
A5peaks[yv_]:= − 263.55yv^2 + 982.43yv − 912.61;
A3peaks[yv_]:= − 5414.9yv^4 + 37358yv^3 − 96674yv^2 + 111207yv − 47975;


If[Upload,


solution0 = NDSolve[{
A'[t] == −k ∗ (A[t] + (B[t] ∗ ϕ/k) − YPert + (2 ∗ c ∗ R[t])) ,
B'[t] == −k ∗ (B[t] − (A[t] ∗ ϕ/k) + (2 ∗ c ∗ S[t])) ,
R'[t] == −(R[t] − (Δ ∗ S[t])) + (A[t] ∗ d[t]),
S'[t] == −((Δ ∗ R[t]) + S[t]) + (B[t] ∗ d[t]),
d'[t] == −γ ∗ ((A[t] ∗ R[t]) + (B[t] ∗ S[t]) + d[t] + 1),
A[0] == A_1,
B[0] == B_1,
R[0] == R_1,
S[0] == S_1,
d[0] == d_1
}, {A, B, S, R, d}, {t, 0, tLimit}, PrecisionGoal → ∞, MaxSteps → ∞,
MaxStepSize → Δt];
AA1[t_]:=A[t]/.solution0[[1]];
BB1[t_]:=B[t]/.solution0[[1]];
RR1[t_]:=R[t]/.solution0[[1]];
SS1[t_]:=S[t]/.solution0[[1]];
dd1[t_]:=d[t]/.solution0[[1]];


atrad[t_]:={AA1[t], BB1[t], RR1[t], SS1[t], dd1[t]}/.solution0[[1]];
Print[ListPointPlot3D[Table[{AA1[t], BB1[t], dd1[t]}, {t, tLimit − prd − 0.5,
tLimit, 0.25}]]];
```

```
lineR = Table[{AA1[t], BB1[t], RR1[t], SS1[t], dd1[t]}, {t, tLimit − prd − 0.5,
tLimit, Δt}]
, False];


solution = NDSolve[{
A′[t] == −k ∗ (A[t] + (B[t] ∗ $\frac{\phi}{k}$) − Y + (2 ∗ c ∗ R[t])) ,
B′[t] == −k ∗ (B[t] − (A[t] ∗ $\frac{\phi}{k}$) + (2 ∗ c ∗ S[t])) ,
R′[t] == −(R[t] − (Δ ∗ S[t])) + (A[t] ∗ d[t]),
S′[t] == −((Δ ∗ R[t]) + S[t]) + (B[t] ∗ d[t]),
d′[t] == −γ ∗ ((A[t] ∗ R[t]) + (B[t] ∗ S[t]) + d[t] + 1),
A[0] == $A_0$,
B[0] == $B_0$,
R[0] == $R_0$,
S[0] == $S_0$,
d[0] == $d_0$
}, {A, B, S, R, d}, {t, 0, tLimit}, PrecisionGoal → ∞, MaxSteps → ∞,
MaxStepSize → Δt];


AA[t_]:=A[t]/.solution;
BB[t_]:=B[t]/.solution;
RR[t_]:=R[t]/.solution;
SS[t_]:=S[t]/.solution;
dd[t_]:=d[t]/.solution;


X[t_]:=$\sqrt{\mathrm{AA}[t]^2 + \mathrm{BB}[t]^2}$/.solution;


Print[Plot[X[t], {t, tLimit − 1000, tLimit}, PlotRange → All, AxesLabel → {Style[t,
Large, Bold], Style[|x| , Large, Bold]}, AxesStyle → Thick]];
datas = Table[{AA[t][[1]], BB[t][[1]], dd[t][[1]]}, {t, tLimit − 1000, tLimit}];
spiral0Base = ListPointPlot3D[datas, AxesLabel → {"Re |X|", "Im |X|", "D "},
BaseStyle → {FontWeight → "Bold", FontSize → 18}, PlotRange → Full, PlotStyle
→ Green];
Print[spiral0Base];
(*need to find CTTC for base att*)

firstRun = Plot[X[t], {t, tLimit − 1000, tLimit}, PlotRange → All, AxesLabel →
{Style[t, Large, Bold], Style["|X|", Large, Bold]}, AxesStyle → Thick];
Print[firstRun];
maxP = Max[Last/@Level[Cases[firstRun, _Line, Infinity], {−2}]];
Print[maxP];
sns = Position[firstRun, Max[Last/@Level[Cases[firstRun, _Line, Infinity],
{−2}]]][[1]];

Print[sns = ReplacePart[sns, Length[sns] → 1]];

Print[Extract[firstRun, sns]];
maxXPoint = {AA[Extract[firstRun, sns]][[1]], BB[Extract[firstRun, sns]][[1]],
RR[Extract[firstRun, sns]][[1]], SS[Extract[firstRun, sns]][[1]],
dd[Extract[firstRun, sns]][[1]]};
Print[maxXPoint];

A1 = maxXPoint[[1]];
B1 = maxXPoint[[2]];
R1 = maxXPoint[[3]];
```

```
S1 = maxXPoint[[4]];
d1 = maxXPoint[[5]];




SetSharedVariable[prog];
prog = 0;
Dynamic[Row@{ProgressIndicator[prog, {0, Ceiling[prd0]}], prog, “/”, Ceiling[prd0]},
TrackedSymbols → {}, UpdateInterval → 2]
ParallelDo[


If[tLag == 0, YV[t_] = Piecewise[{{Y, t < TAP}, {YPert, t ≥ TAP}}], YV[t_] =
Piecewise[{{Y, t < TAP}, {Y + (t − TAP) ∗ (YPert − Y)/tLag,
TAP ≤ t < TAP + tLag}, {YPert, t ≥ tLag + TAP}}]];
nWeight = 1;


solution = NDSolve[{
A′[t] == −k ∗ (A[t] + (B[t] ∗ (ϕ−rar[t])/k) − YV[t] + (2 ∗ c ∗ R[t])) ,
B′[t] == −k ∗ (B[t] − (A[t] ∗ (ϕ−rar[t])/k) + (2 ∗ c ∗ S[t])) ,
R′[t] == −(R[t] − ((Δ + rar[t]) ∗ S[t])) + (A[t] ∗ d[t]),
S′[t] == −(((Δ + rar[t]) ∗ R[t]) + S[t]) + (B[t] ∗ d[t]),
d′[t] == −γ ∗ ((A[t] ∗ R[t]) + (B[t] ∗ S[t]) + d[t] + 1),
A[0] == A1,
B[0] == B1,
R[0] == R1,
S[0] == S1,
d[0] == d1,
rar′[t] == 0,
rar[0] == 0,
WhenEvent[0 == Mod[t, nRate], rar[t] → nWeight∗
RandomVariate[NormalDistribution[0, nW]]],
WhenEvent[t > noiseT, nWeight = 0]}, {A, B, S, R, d, rar}, {t, 0, tLimit},
PrecisionGoal → ∞,
MaxSteps → ∞, MaxStepSize → Δt, Method → “StiffnessSwitching”];

AA[t_]:=A[t]/.solution;
BB[t_]:=B[t]/.solution;
RR[t_]:=R[t]/.solution;
SS[t_]:=S[t]/.solution;
dd[t_]:=d[t]/.solution;

pointShift = {{First[A[TAP]/.solution], First[B[TAP]/.solution],
First[d[TAP]/.solution]}};
(*where the first shift happens*)
datSB = ListPointPlot3D[pointShift, AxesLabel → {“Re |X|”, “Im |X|”, “D ”},
BaseStyle → {FontWeight → “Bold”, FontSize → 12, PointSize → Large}, PlotStyle →
Magenta, PlotRange → Full];

X[t_]:=√(AA[t]^2 + BB[t]^2)/.solution;



datas = Table[{AA[t][[1]], BB[t][[1]], dd[t][[1]]}, {t, noiseT − 4000, noiseT}];
spiral = ListPointPlot3D[datas, AxesLabel → {“Re |X|”, “Im |X|”, “D ”},
BaseStyle →
```

```
{FontWeight → “Bold”, FontSize → 18}, PlotRange → Full, PlotStyle → Blue];


datas = Table[{AA[t][[1]], BB[t][[1]], dd[t][[1]]}, {t, tLimit − 4000, tLimit
(*,Δt*)}];


(*Attractor checker here*)
AEpeaks = {};
temporalX = Table[{t, If[X[t][[1, 1]] > X[t − Δt][[1, 1]]&&X[t][[1, 1]] >
X[t + Δt][[1, 1]]
, X[t][[1, 1]]; AppendTo[AEpeaks, Round[X[t][[1, 1]], 0.0001]], X[t]
[[1, 1]]]},
{t, tLimit − prd − 0.5, tLimit, Δt}];



AEpeaks = DeleteDuplicates[AEpeaks];
AtPeakDiff = Sort[Max[AEpeaks] − AEpeaks];
(*Print[cc]; *)
If[outPut,
If[Max[AtPeakDiff] ≥ A5peaks[YPert] ∗ (1 − thr)&&Max[AtPeakDiff] ≤ A5peaks[YPert]∗
(1 + thr), AppendTo[attrat, {TAP, AttractorR}];
Print[Grid[{{Show[datSB, spiral, spiral0Base, PlotRange → All, BaseStyle →
{FontWeight → “Bold”, FontSize → 12, PointSize → Small}]}, {Graphics@
Text[“run ” <> ToString[prog] <> “ Went to Att ” <> ToString[AttractorR]
<> “”]}}]]; , AppendTo[attrat, {TAP, AttractorL}];
Print[Grid[{{Show[datSB, spiral, spiral0Base, PlotRange → All, BaseStyle →
{FontWeight → “Bold”, FontSize → 12, PointSize → Small}]}, {Graphics@Text
[“run ” <> ToString[prog] <> “ Went to Att ” <> ToString[AttractorL] <> “”]
}}]]]; , If[Max[AtPeakDiff] ≥ A5peaks[YPert] ∗ (1 − thr)&&Max[AtPeakDiff] ≤
A5peaks[YPert] ∗ (1 + thr), AppendTo[attrat, {TAP, AttractorR}]; , AppendTo[attrat,
{TAP, AttractorL}]; ]
];
prog++;
, {TAP, 1, Ceiling[prd0]}];

Export[“Lis System\\Case ” <> ToString[Case] <> “\\Power shift data\\run ” <>
ToString[num] <> “ FN GNoise Power shift from Attractor ” <>
ToString[Attractor] <> “ for Y = ” <> ToString[Y] <> “ to ” <>
ToString[YPert] <> “ tLag=” <> ToString@tLag <> “ gnWeight=”
<> ToString[nW0] <> “.xlsx”, attrat];
```

OM Power shift

```
Case = 20;
Attractor = 0;
PLoad = 3.75;
P1 = 0.05;
PMax = 3;
PMin = 0;
PStep = 0.001;
thres = 0.5;
P = P1;
PPert = 16.46;
Upload =!True;
wthywtni = {{P, “Ω Zero”}};
δv = 0;
```

$A_0 = \delta\mathrm{v};$

$B_0 = \delta\mathrm{v};$

$F_0 = \delta\mathrm{v};$

$G_0 = \delta\mathrm{v};$

$R_0 = \delta\mathrm{v};$

$S_0 = \delta\mathrm{v};$

$\Delta\mathrm{t} = 0.001;$

$t_{\mathrm{limit}} = 500;$

TAP = 200;

If[Upload,

file = ToExpression[

Import[“ROM\\Case ” <> ToString[Case] <>

“\\Attractor ” <> ToString[Attractor] <>

“\\SteadyState P=” <> ToString[PLoad] <>

“.txt”]];

$A_0 = \mathrm{file}[[1]];$

$B_0 = \mathrm{file}[[2]];$

$F_0 = \mathrm{file}[[3]];$

$G_0 = \mathrm{file}[[4]];$

$R_0 = \mathrm{file}[[5]];$

$S_0 = \mathrm{file}[[6]];$

,

$\{A_0, B_0, F_0, G_0, R_0, S_0\}]$;

Print[file];

$\mathrm{v0} = 50 * 2. * \pi;$

$\mathrm{c1} = 2 * 10^5;$

$\mathrm{c2} = 4.15 * 10^{-19};$

$\mathrm{c3} = \frac{1}{\mathrm{c1}^2};$

$\mathrm{Q1} = 10^8;$

$\mathrm{Q2} = 10^8;$

$\mathrm{k1} = \frac{6.08*10^8}{\mathrm{Q1}};$

$\mathrm{k2} = \frac{6.08*10^8}{\mathrm{Q2}};$

$a = 175;$

$\mathrm{t1} = 2.29\sqrt{\frac{a}{\mathrm{Q1}}};$

$\mathrm{t2} = 2.29\sqrt{\frac{a}{\mathrm{Q2}}};$

$\tau = 3 * 10^{-8}a;$

$\xi = \frac{\mathrm{t1t2}}{2\tau};$

If[Case == 1,

$\Gamma\mathrm{B} = 10^2;$

$\Delta 1 = 0 * 2. * \pi; (*(0\mathrm{to}5) * 2 * \pi*)$

$\Delta 2 = 50 * 2. * \pi(*(40\mathrm{to}60) * 2 * \pi*)$

];

If[Case == 2,

$\Gamma\mathrm{B} = 3 * 10^{-2};$

$\Delta 1 = 0 * 2. * \pi; (*(0\mathrm{to}5) * 2 * \pi*)$

$\Delta 2 = 50 * 2. * \pi; (*(40\mathrm{to}60) * 2 * \pi*)$

];

```
If[Case == 7,
ΓB = 0.011;
Δ1 = 2 * 2. * π; (*(0to5) * 2 * π*)
Δ2 = 50 * 2. * π, False(*(40to60) * 2 * π*)
];
If[Case == 20,
ΓB = 13;
Δ1 = 1 * 2. * π; (*(0to5) * 2 * π*)
Δ2 = 44 * 2. * π, False(*(40to60) * 2 * π*)
];
If[Case == 9.1,
ΓB = 13;
Δ1 = 1 * 2. * π; (*(0to5) * 2 * π*)
Δ2 = 44.013 * 2 * π, False(*(40to60) * 2 * π*)
];
Δ21 = Δ2 − Δ1;
While[PMin ≤ P ≤ PMax,
Print[TAP];
PV[t_] = Piecewise[{{P, t < TAP}, {PPert, t ≥ TAP}}];

solution = NDSolve[{
A′[t] == −(k1)A[t] − Δ1B[t] − ξF[t] − c1R[t]G[t]−
c1S[t]F[t],
B′[t] == −(k1)B[t] + Δ1A[t] − ξG[t]+
c1R[t]F[t] − c1S[t]G[t] + t1/τ c1 * √PV[t],
F′[t] == −(k2)F[t] − Δ2G[t] − ξA[t]−
c1R[t]B[t] + c1S[t]A[t],
```

$G'[t] == -(\mathrm{k2})G[t] + \Delta 2F[t] - \xi B[t]+$

$\mathrm{c1}R[t]A[t] + \mathrm{c1}S[t]B[t] + \frac{\mathbf{t2}}{\tau}\mathrm{c1} * \sqrt{\mathrm{PV}}[t],$

$R'[t] == -\frac{\mathbf{\Gamma B}}{\mathbf{2}}R[t] + \Delta 21S[t] - \mathrm{v0}S[t]-$

$\mathrm{c2}B[t]F[t]\Delta 21 + \mathrm{c2}A[t]G[t]\Delta 21,$

$S'[t] == -\frac{\mathbf{\Gamma B}}{\mathbf{2}}S[t] - \Delta 21R[t] + \mathrm{v0}R[t]+$

$\mathrm{c2}A[t]F[t]\Delta 21 + \mathrm{c2}B[t]G[t]\Delta 21,$

$A[0] == A_{\mathbf{0}},$

$B[0] == B_{\mathbf{0}},$

$F[0] == F_{\mathbf{0}},$

$G[0] == G_{\mathbf{0}},$

$R[0] == R_{\mathbf{0}},$

$S[0] == S_{\mathbf{0}}$

$\}, \{A, B, F, G, R, S\}, \{t, 0, t_{\mathbf{limit}}\}$, PrecisionGoal → ∞,

MaxSteps → ∞, MaxStepSize → Δt,

Method → “StiffnessSwitching”];

Print[“test1”];

AA[t_]:=$A[t]$/.solution;

BB[t_]:=$B[t]$/.solution;

FF[t_]:=$F[t]$/.solution;

GG[t_]:=$G[t]$/.solution;

RR[t_]:=$R[t]$/.solution;

SS[t_]:=$S[t]$/.solution;

ES1[t_]:=$\sqrt{A[t]^2 + B[t]^2}$/.solution;

ES2[t_]:=$\sqrt{F[t]^2 + G[t]^2}$/.solution;

ρ[t_]:=$\sqrt{R[t]^2 + S[t]^2}$/.solution;

pOut[t_]:=

```
Abs[c1 * √PV[t] + i * t1 * ES1[t] + i * t2 * ES2[t]]^2*
c1^(−1)/.solution;


Print [Plot [pOut[t], {t, t_limit − 2, t_limit} ,
PlotRange → All, AxesLabel → {Style[t, Large, Bold],
Style["Pout", Large, Bold]}, AxesStyle → Thick]];
preShift =
Table[{ES1[t][[1]], ES2[t][[1]], ρ[t][[1]]},
{t, TAP − 10, TAP, 0.02}];
datas = Table[{ES1[t][[1]], ES2[t][[1]], ρ[t][[1]]},
{t, t_limit − 30, t_limit, 0.01(*,Δt*)}] ;
Print[
Show[ListPointPlot3D[datas,
AxesLabel → { "Es1", "Es2", "ρ " }, BaseStyle →
{FontWeight → "Bold", FontSize → 18},
PlotRange → Full, BoxRatios → {1, 1, 1}],
ListPointPlot3D[preShift,
AxesLabel → { "Es1", "Es2", "ρ " },
PlotStyle → Red], PlotRange → All]];


Print["P=" <> ToString[P] <>
" Case=" <> ToString[Case]];
(*Export["ROM\\Case " <> ToString[Case] <>
"\\Attractor " <> ToString[Attractor] <>
"\\SteadyState Y=" <> ToString[P] <> ".txt", ppos]; *)


(*Print["c = " <> ToString[c] <>
“; Y = ” <> ToString[$Y$] <>
“; K = ” <> ToString[$K$] <>
“; $\phi$ = ” <> ToString[$\phi$] <>
“; $\gamma$ = ” <> ToString[$\gamma$] <>
“; $\Delta$ = ” <> ToString[$\Delta$] <>
“; $\delta$v = ” <> ToString[$\delta$v] <>
“; Time Range(tf1)=” <> ToString[tf1]];
*)

$t_{\mathbf{stop}} = t_{\mathbf{limit}}$;

$A_0$ = First[$A$[TAP]/.solution];
$B_0$ = First[$B$[TAP]/.solution];
$F_0$ = First[$F$[TAP]/.solution];
$G_0$ = First[$G$[TAP]/.solution];
$R_0$ = First[$R$[TAP]/.solution];
$S_0$ = First[$S$[TAP]/.solution];
$P$+=PStep;
Print [$A_0$] ;
];
```

Poincaré map

```
Clear[“Global`*”]

Case = 1;

Attractor = 4;

YLoad = 1.1192;

YInitial = 13;

YMax = 14;

YMin = 4.0;

YStep = 0.1;

Y = YInitial

Upload = True;



If[Case == 1,

c = 20;

K = 0.5;

φ = −2.0;

γ = 0.05;

Δ = 1.0];

If[Case == 2,

c = 3;

K = 0.1;

φ = −0.5;

γ = 0.01;

Δ = 0.5];

If[Case == 3,
```

$c = 3;$

$K = 1.0/3.0;$

$\phi = -3.0;$

$\gamma = 1.0 * 10\wedge - 4;$

$\Delta = 1.0];$

If[Case == 4,

$c = 4;$

$K = 0.125;$

$\phi = -3.0;$

$\gamma = 1.0 * 10\wedge - 4;$

$\Delta = 0.375];$

If[Case == 5,

$c = 3;$

$K = 1.0/3.0;$

$\phi = -3.0/2.0;$

$\gamma = 0.0006;$

$\Delta = 0.5];$

$\delta\mathrm{v} = N[0.001, 64];$

$A_0 = \delta\mathrm{v};$

$B_0 = \delta\mathrm{v};$

$R_0 = \delta\mathrm{v};$

$S_0 = \delta\mathrm{v};$

$d_0 = \delta\mathrm{v};$

tab2 = {{Y, “Ω Zero”}};

$\Delta\mathrm{t} = 0.05;$

$t_{\mathrm{limit}} = 20000;$

(*ics*)

If[Upload,

file = {6.466399618413315, −7.751209696152986, −0.24649692245477814,

0.3556284953127857, −0.03859780519097036};

$A_0 = \text{file}[[1]];$

$B_0 = \text{file}[[2]];$

$R_0 = \text{file}[[3]];$

$S_0 = \text{file}[[4]];$

$d_0 = \text{file}[[5]];$

,

$\{A_0, B_0, R_0, S_0, d_0\}]\,;$

$m = 0$

$m{+}{+}$

$\text{eqns} = \{A'[t] == -K * (A[t] + (B[t] * \phi) - Y + (2 * c * R[t])),$

$B'[t] == -K * (B[t] - (A[t] * \phi) + (2 * c * S[t])),$

$R'[t] == -(R[t] - (\Delta * S[t])) + (A[t] * d[t]),$

$S'[t] == -((\Delta * R[t]) + S[t]) + (B[t] * d[t]),$

$d'[t] == -\gamma * ((A[t] * R[t]) + (B[t] * S[t]) + d[t] + 1)\};$

psect[{A0_, B0_, R0_, S0_, d0_}]:=

Reap[NDSolve[{eqns, $A[0] ==$ A0, $B[0] ==$ B0, $R[0] ==$ R0, $S[0] ==$ S0,

$d[0] ==$ d0, WhenEvent[$d[t] == -0.05$, Sow[{$A[t], B[t]$}]]},

{}, {t, 0, 1000000}, MaxSteps → ∞]][[−1, 1]]

```
caredata = Map[psect, {file}]; (*insert steady state here*)


giraffe = ListPlot[caredata, ImageSize → Medium,
AxesLabel → {“Re |X|”, “Im |X|”},
BaseStyle → {FontWeight → “Bold”, FontSize → 10}, PlotStyle → Green]
```

VITA

Erikk Kenneth Tilus Burton

Candidate for the Degree of

Doctor of Philosophy

Thesis: A POWER-SHIFT METHOD TO CONTROL NONLINEAR SYSTEMS

Major Field: Physics

Biographical:

Education:

Completed the requirements for the Doctor of Philosophy in Physics at Oklahoma State University, Stillwater, Oklahoma in May, 2022.

Completed the requirements for the Bachelor of Science in Physic at Oklahoma State University, Stillwater, Oklahoma in 2013.

Completed the requirements for the Bachelor of Science in Mathematics at Oklahoma State University, Stillwater, Oklahoma in 2013.

Worked as teaching assistant from August 2017-2022

Performed research under Dr. Bandy from June 2014-2022

Professional Membership:

OPTICA