

University of New Hampshire

University of New Hampshire Scholars' Repository

Master's Theses and Capstones

Student Scholarship

Fall 2008

Controlled transitions between orbits in nonlinear systems

Erica G. Johnson

University of New Hampshire, Durham

Follow this and additional works at: <https://scholars.unh.edu/thesis>

Recommended Citation

Johnson, Erica G., "Controlled transitions between orbits in nonlinear systems" (2008). *Master's Theses and Capstones*. 386.

<https://scholars.unh.edu/thesis/386>

This Thesis is brought to you for free and open access by the Student Scholarship at University of New Hampshire Scholars' Repository. It has been accepted for inclusion in Master's Theses and Capstones by an authorized administrator of University of New Hampshire Scholars' Repository. For more information, please contact Scholarly.Communication@unh.edu.

**Controlled Transitions Between Orbits in
Nonlinear Systems**

BY

ERICA G. JOHNSON

B.S., University of New Hampshire, 2006

Submitted to the University of New Hampshire
in partial fulfillment of
the requirements for the degree of

Master of Science

in

Mathematics: Applied Mathematics

September 2008

UMI Number: 1459500

INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

UMI[®]

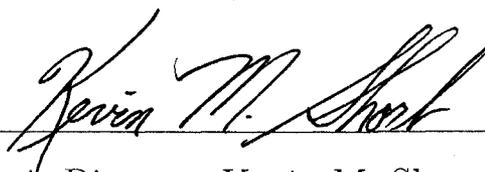
UMI Microform 1459500

Copyright 2008 by ProQuest LLC.

All rights reserved. This microform edition is protected against unauthorized copying under Title 17, United States Code.

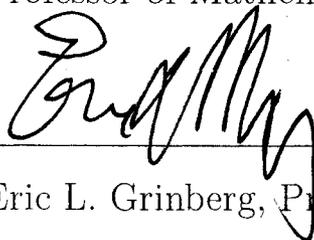
ProQuest LLC
789 E. Eisenhower Parkway
PO Box 1346
Ann Arbor, MI 48106-1346

This dissertation has been examined and approved.



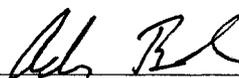
Thesis Director, Kevin M. Short

Professor of Mathematics



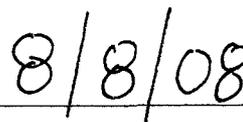
Eric L. Grinberg, Professor of

Mathematics



Adam Boucher, Lecturer of

Mathematics



Date

DEDICATION

To my family, with love.

ACKNOWLEDGMENTS

Thank you Kouros Zarringhalam and Kevin Short for all your guidance.

TABLE OF CONTENTS

	Page
Dedication	iii
Acknowledgments	iv
LIST OF TABLES	vii
LIST OF FIGURES	viii
Abstract	x
CHAPTER	
1. BACKGROUND	1
1.1 Introduction	1
1.2 Methods of controlling chaos	3
2. CUPOLETS	9
2.1 Introduction	9
2.2 Generating Stabilized Periodic Orbits	9
2.3 Application and Properties of Cupolets	14
3. SWITCHING BETWEEN INTERSECTING CUPOLETS	16
3.1 Introduction	16
3.2 Blind Transition	16
3.3 Switchable Cupolets	20
4. SWITCHING BETWEEN NON-INTERSECTING CUPOLETS	23
4.1 Introduction	23
4.2 Controlled Transitions	23
4.3 Measuring Transitions	26

4.4	Basis for Cupolets	29
4.5	Modeling Cupolets using Graph Theory	31
4.6	Dijkstra's Shortest Path Algorithm	37
4.7	Decomposing Adjacency Matrices	42
5.	RESULTS AND CONCLUSION	54
5.1	Results	54
5.2	Conclusion	62
	BIBLIOGRAPHY	63

LIST OF TABLES

Table	Page
3.1 The information about bins, lobes, and controls (a) for cupolet A control sequence '100' and (b) for cupolet B with control sequence '101100'	21
4.1 The set of bin pairs for which the shortest path must be determined or order to find the controlled transition with the smallest length from cupolet B and cupolet C	33
5.1 Bin, lobe, and control information corresponding to cupolet '0101011.'	55
5.2 Bin, lobe, and control information corresponding to cupolet '001111.'	55
5.3 Bin, lobe, and control information corresponding to cupolet 0001011011.'	57

LIST OF FIGURES

Figure	Page
1-1 (a) The butterfly effect of the Lorenz system and its projections on to (b) $x - y$ plane, (c) $x - z$ plane and (d) $y - z$ plane.	4
2-1 Double scroll oscillator. (a) Electrical schematic and (b) Nonlinear resistance g	10
2-2 A typical trajectory of the double scroll oscillator projected on the x - y plane	11
2-3 Poincaré Surface. (a) Passing through the center of each attractor and (b) Least square approximation of points through control surface	12
2-4 Coding function $r(x)$	13
3-1 Continuous switch from the solid line, cupolet 'C00', to the dotted line of the transient	17
3-2 Blind Transition (a) the blue orbit is cupolet '00' and the red orbit is cupolet '01' (b) the green line is the transition	18
4-1 The information about bins, lobes, and controls (a) for cupolet A control sequence '100', (b) for cupolet B with control sequence '101100', (c) for cupolet C with control sequence '101', and (d) for cupolet D with control sequence '01'	25
4-2 The bin information for five cupolets in a bin space with three bins	30
4-3 Undirected graph of the cupolets A,B,C, D and E in the three bin space. The vertices represent bins and the edges represent the cupolets which connect those bins.	32

4-4	Weighted multi-digraph of the cupolets A,B,C, D, and E in the three bin space. The vertices represent bins and the directed edges represent cupolets where the subscript is the name of the cupolet and the number represents number of bins visited if going from one bin to the next on that cupolet.	35
4-5	Weighted digraph of the cupolets A, B, C, D, and E in the three bin space. The vertices represent bins and the directed edges represent cupolets and how many bins must be visited to get from one bin to the next.	36
4-6	An example weighted digraph for which we will determine the shortest path from vertex 1 to every other vertex using Dijkstra's algorithm.	39
4-7	The bin information for four cupolets A,B,C and D in a bin space with three bins	42
4-8	Cupolets A,B,C, and D from the three bin space. The vertices represent cupolets and the edges mean that both cupolets pass through the same bin.	43
4-9	Three different three step paths between Cup B and vertex Cup C. The directed edge with a 1 is the first step in the path, the directed edge with a 2 is the second step in the path, the directed edge with a 3 is the third step in the path.	45
4-10	A simple graph with eight vertices	51
4-11	Calculations for determining all paths between vertex one and vertex six as shown in Figure 4-10.	52
5-1	Comparison of different transitioning techniques between cupolet 0101011 and cupolet 001111.	58
5-2	Comparison of different transitioning techniques between cupolet '0010110101' and cupolet '01'.	59
5-3	Comparison of different transitioning techniques between cupolet '0000001111' and cupolet '00001'.	61
5-4	Comparison of different transitioning techniques between cupolet '0010110011' and cupolet '000000011'.	61

ABSTRACT

Controlled Transitions Between Orbits in Nonlinear Systems

by

ERICA G. JOHNSON

University of New Hampshire, September, 2008

In recent years, several methods of controlling chaotic systems have been developed and implemented. The main idea in each method is to stabilize on an orbit around a chaotic attractor, which generally has a dense set of unstable periodic orbits. One such control scheme repeatedly applies a sequence of controls to a double scroll oscillator. Most control sequences result in the stabilization of an approximate unstable periodic orbit regardless of initial condition. These stabilized periodic orbits are called chaotic unstable periodic orbit-lets (cupolets). Due to the nature of cupolets, it is possible to switch between cupolets, and thus periodic orbits, by changing from one control sequence to another. Switching between orbits is a continuous and smooth transition, but may involve significant chaotic transients.

We will present three methods of transitioning between cupolets and suggest some applications of this procedure. The first method involves applying the second control sequence at a location on the first orbit. The second method is a zero-length transition which can be used if two cupolets intersect. The third method is applicable when transitioning between non-intersecting cupolets. This method switches between intermediate cupolets in an efficient controlled manner in getting from one cupolet to the next.

CHAPTER 1

BACKGROUND

1.1 Introduction

In tracking controlled transitions between orbits derived from nonlinear systems, it is first imperative to understand how these orbits are created. In this chapter we will provide the necessary background information about chaotic nonlinear systems. We will also discuss various methods of controlling chaotic systems which ultimately lead to the creation of chaotic unstable periodic orbit-lets, which are described in full in Chapter 2.

We are concerned with systems of nonlinear differential equations of the form

$$\dot{\vec{x}} = F(\vec{x})$$

where \vec{x} is a finite vector and $F(\vec{x})$ is a vector of functions which depend on \vec{x} . A system is considered nonlinear if there exists any non-linear terms. The system of differential equations

$$\dot{x} = F_1(x, y) = x^2 + y$$

$$\dot{y} = F_2(x, y) = x + xy$$

is considered nonlinear due to the term x^2 in $F_1(x, y)$ and the term xy in $F_2(x, y)$. In working with systems of differential equations which are nonlinear, it is often possible

to determine characteristics about the system such as the location of fixed points, where

$$\dot{\vec{x}}_0 = 0,$$

stable orbits, and unstable orbits. Unfortunately, it is often not possible to determine the exact solution \vec{x} .

Although the exact solution of a nonlinear system of equations cannot always be determined, an approximation of the solution can be found using numerical integration. Numerical integration uses known values of the solution to estimate subsequent values. One method of numerical integration is fourth-order Runge-Kutta defined as

$$\vec{x}_{n+1} = \vec{x}_n + \frac{1}{6}(\vec{k}_1 + 2\vec{k}_2 + 2\vec{k}_3 + \vec{k}_4),$$

$$\text{where } \vec{k}_1 = F(\vec{x}_n)\Delta t$$

$$\vec{k}_2 = F(\vec{x}_n + \frac{1}{2}\vec{k}_1)\Delta t$$

$$\vec{k}_3 = F(\vec{x}_n + \frac{1}{2}\vec{k}_2)\Delta t$$

$$\vec{k}_4 = F(\vec{x}_n + \vec{k}_3)\Delta t.$$

Runge-Kutta results in an accurate approximation to the solution which does not require an extremely small time step Δt .

The approximation of the solution to the nonlinear system of equations allows one to examine long term behavior of the system. With the advancement of technology in the early 1990's, it was finally possible to plot the trajectories of the nonlinear systems from a variety of initial conditions in a reasonable time frame. Researchers were able to analyze nonlinear systems of differential equations whose long-term effects were not known otherwise.

In examining these nonlinear systems of differential equations, it became clear that for some systems, it is not possible to predict long term behavior independent of initial condition since trajectories from initial conditions which are very close can vary immensely as time goes on. These systems are said to be chaotic where the chaos is defined as "aperiodic long-term behavior in a deterministic system that exhibits sensitive dependence on initial conditions" [10]. This means the trajectories of the nonlinear system are governed heavily by the initial condition and nearby trajectories separate exponentially fast. These trajectories also do not settle down to fixed points or stable periodic orbits as $t \rightarrow \infty$.

There are some chaotic systems which which have a strange attractor. This is an attractor for which any trajectory starting in the closed set around the attractor, stays there for all time but never stabilizes onto a periodic orbit. In examining the Lorenz system

$$\begin{aligned}\frac{dx}{dt} &= \sigma(y - x) \\ \frac{dy}{dt} &= x(\rho - z) - y \\ \frac{dz}{dt} &= xy - \beta z\end{aligned}$$

it is clear there is a strange attractor. Around each attractor in the Lorenz system there is a dense set of unstable period orbits. The general shape of the Lorenz system is a butterfly and can be seen in Figure 1-1.

1.2 Methods of controlling chaos

In working with systems which are naturally chaotic, it is often desirable to stabilize the system. Some systems which have been stabilized using chaotic controls include turbulent fluids, chemical reactions, and cardiac tissues. The first method of

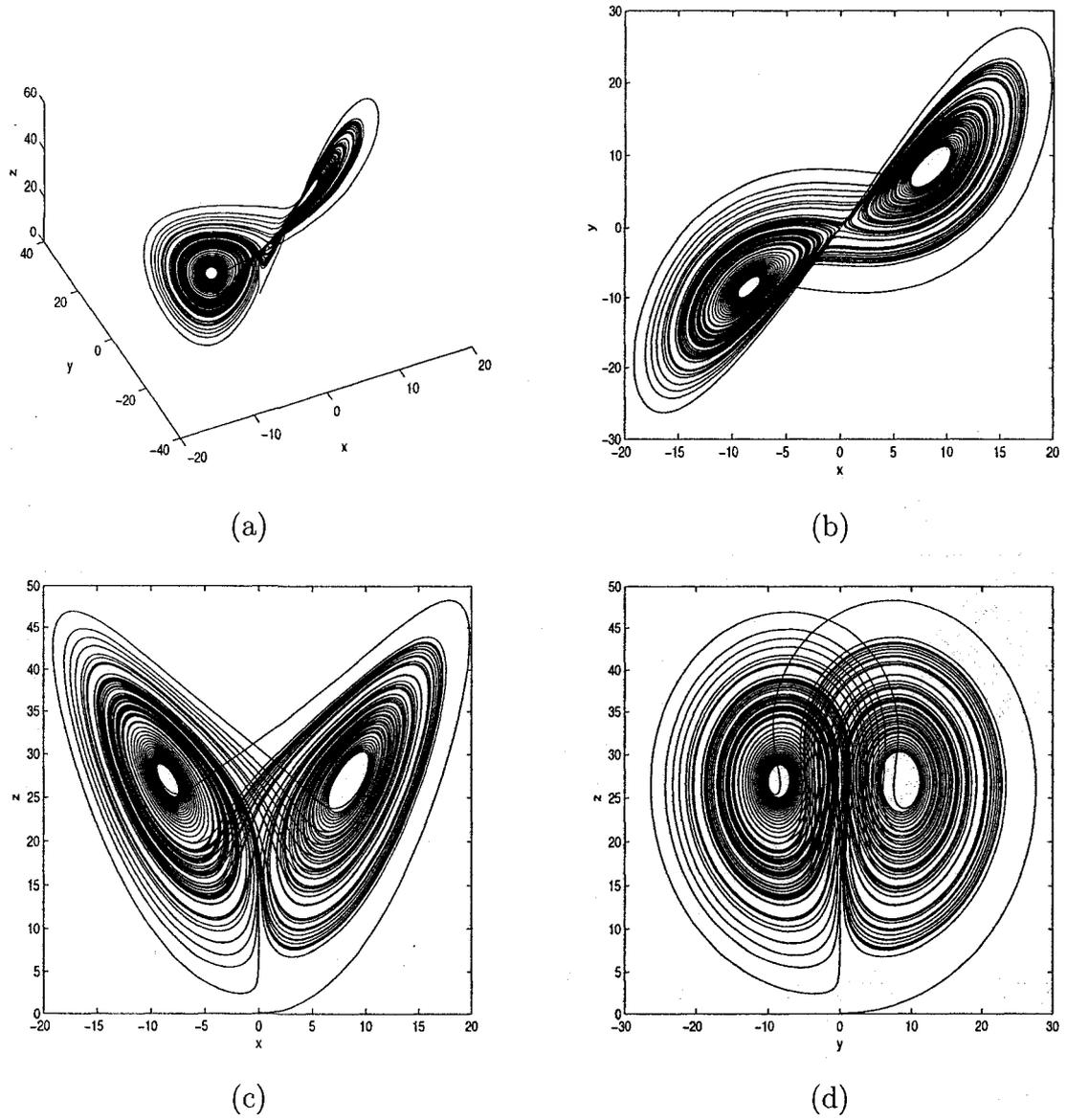


Figure 1-1. (a) The butterfly effect of the Lorenz system and its projections on to (b) $x - y$ plane, (c) $x - z$ plane and (d) $y - z$ plane.

controlling chaos we would like to discuss is the OGY method [7]. To control chaotic behavior in a nonlinear system, the OGY method applied small perturbations to an available parameter in a nonlinear system with a chaotic attractor. Since there is a dense set of unstable periodic orbits around the attractor, OGY did not believe that making small perturbations to the system would create new orbits whose properties would be very different from existing orbits.

In implementing the OGY method, it is first necessary to determine the low-period unstable periodic orbits and choose one which results in a desired performance of the system. In controlling chaos using this method, small perturbations are made at each time step to direct the trajectory so that it remains on the desired periodic orbit. If the perturbations are continually applied it is possible to remain on the periodic orbit, and thus the unstable periodic orbit becomes a stable periodic orbit.

The next method of controlling chaos, developed by Troy Shinbrot, uses small perturbations on an available parameter to direct the trajectory from an arbitrary initial state to a desired periodic orbit[8]. Shinbrot demonstrated this method by successfully reaching a targeted orbit in a one-dimensional map. The section of the trajectory from the initial condition to where the system stabilizes onto the desired orbit is called a transient.

The trajectory of a chaotic system is complicated and unpredictable. It is possible to exploit the unpredictable behavior of chaotic system to practical applications such as secure communication. In 1992, Kevin Cuomo and Alan Oppenheim used the idea of synchronized chaos to transmit secret messages [2]. To do so, they masked the message with chaos on the sending end and informed the person receiving the

message how to subtract the chaos and listen to the secret message. In this way, the code can only be decoded only if the exact initial condition which governs the chaos in the message is known. A person would not be able to guess an initial condition which would match the chaos and thus would not be able to subtract off the noise and determine the underlying message.

Another method of communicating using chaos was developed by Hayes, Grebogi, and Ott called the HGO method. This method utilizes chaotic controls which steer the trajectory of the system to follow a desired path and encode the message in the path by using the associated symbolic dynamics. Each character in the symbolic dynamics represents an attractor for which the trajectory loops around.

The HGO method first examines a free-running oscillator of the nonlinear system and records the value of the coordinates through a set of Poincaré surfaces. From each point on the Poincaré surface, the system is allowed to run free and the first N bits of the associated symbolic dynamics are recorded in a coding function $r(x)$.

To use the HGO communication scheme it is necessary to determine the grammar of the system. For each nonlinear system, only certain symbolic combinations are possible and thus to encode a message, the grammar must be known. The desired code is then constructed using the grammar of the system.

To encode a message, HGO developed an algorithm in which they would start at a point $x = x_a$ on a Poincaré surface and store the corresponding $r(x_a)$ in a code register to some chosen length N . The system is then set free and continues until it intersects with the next Poincaré surface at $x = x_b$. At this point the code register is shifted left one bit, leaving the right most bit in the register empty. The right

most bit of the code register is then filled with the first bit in the desired message. This code registry is then compared with the symbolic dynamics associated with the current location of the trajectory, $x = x_b$. If the code register and $r(x_b)$ are the same, the trajectory continues through x_b with no change. If the last bit in the code register is different than that of $r(x_b)$, the trajectory is shifted to the closest point on the Poincaré surface, $x = x'_b$ such that $r(x'_b)$ is equal to the code register. For each successive pass through a Poincaré surface, the code register is shifted, the desired bit from the message is inserted, and the trajectory is adjusted accordingly.

The adjustments made to the system, which correspond to shifting the desired code through the code register, are the controls made to the nonlinear system. By implementing the controls as described above, HGO ensured that "after a small transient, the dynamics of the oscillator are locked to the information source and the symbolic dynamics of the oscillator is always N bits behind the information source."

UNH Professor Kevin Short, along with Dr. Andrew Parker, adapted a method in which they use a control scheme similar to that of HGO. They periodically applied short control sequences to a double scroll oscillator with two Poincaré surfaces. They discovered that the trajectories in the chaotic system eventually closed in on themselves, creating periodic orbits. These periodic orbits are called chaotic unstable periodic orbit-lets (cupolets) and are described in full in Chapter 2.

In this thesis we will describe another way of controlling chaotic systems which will utilize our knowledge of cupolets and will result in a way to transition between cupolets in a controlled and efficient manner. We will develop a way in which we can steer the trajectory of a system from one chaotic unstable periodic orbit-let to

another by applying minimal controls to the system. In Chapter 2, we describe the method of creating cupolets and describe their properties. In Chapter 3, we propose several methods of control schemes which result in the switch from one cupolet to another. In Chapter 4, we describe a control scheme which results in controlled transitions between any two chaotic unstable periodic orbit-lets. These results may eventually have applications in the control of nonlinear systems, such as those which occur in chemical reactions, fluid flow, weather events, and maneuvering objects in outer space.

CHAPTER 2

CUPOLETS

2.1 Introduction

In this chapter, we will describe a control scheme which results in chaotic stabilization of (approximate) controlled unstable periodic orbit-lets (cupolets). We will then suggest some applications of cupolets and discuss some important properties. The method of stabilizing the cupolets is adapted from the communication scheme developed by Hayes, Gerbogi, and Ott, (HGO) [4], described in Chapter 1. The HGO method was generated from a circuit, see Figure 2-1 (a), whose governing differential equations result in a double scroll oscillator. The governing equations are linear except for the internal function $g(v)$ which is explicitly shown in Figure 2-1 (b). The result of the non-linear term in the double scroll oscillator is a system which is highly sensitive to initial conditions and exhibits aperiodic long-term behavior. A system of this nature is called a chaotic system.

2.2 Generating Stabilized Periodic Orbits

The differential equations governing the double-scroll oscillator are

$$\begin{aligned}C_1 \dot{v}_{C_1} &= G(v_{C_2} - v_{C_1}) - g(v_{C_1}), \\C_2 \dot{v}_{C_2} &= G(v_{C_1} - v_{C_2}) + i_L, \\Li_L \dot{i}_L &= -v_{C_2},\end{aligned}\tag{2.1}$$

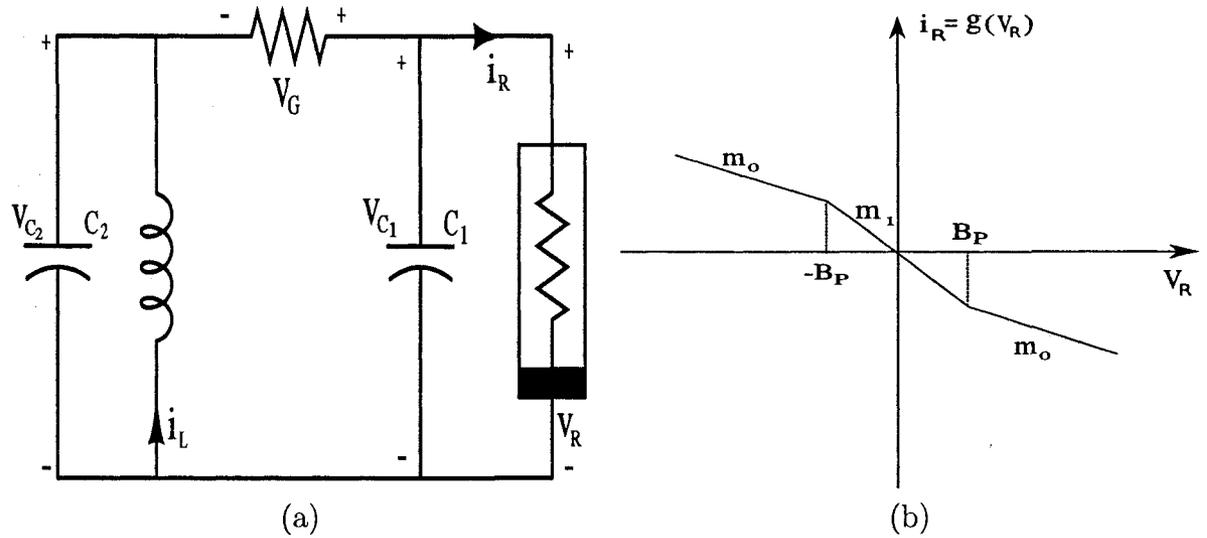


Figure 2-1. Double scroll oscillator. (a) Electrical schematic and (b) Nonlinear resistance g .

where

$$g(v) = \begin{cases} m_1 v, & \text{if } -B_p \leq v \leq B_p, \\ m_0 (v + B_p) - m_1 B_p, & \text{if } v \leq -B_p, \\ m_0 (v - B_p) + m_1 B_p, & \text{if } v \geq B_p, \end{cases} \quad (2.2)$$

where $C_1 = \frac{1}{9}$, $C_2 = 1$, $L = \frac{1}{7}$, $G = 0.7$, $m_0 = -0.5$, $m_1 = -0.8$, and $B_p = 1$.

The attractor associated with the double scroll equations is composed of two loops connected by a transition region. A typical trajectory of the system can be seen in Figure 2-2.

The first step in creating cupolets is to define a Poincaré surface through each loop of the attractor. The Poincaré surface is a control half-plane starting at the center of the loop and defined by

$$i_L = GF, |v_{C_1}| \leq F,$$

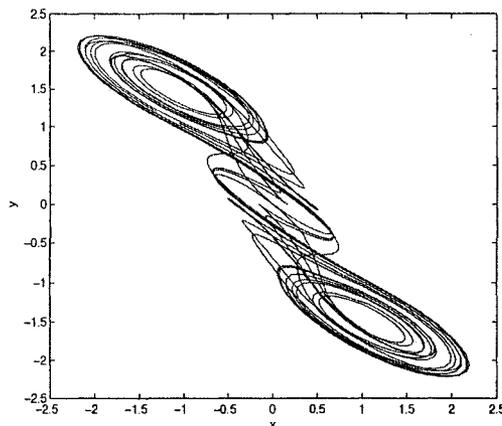


Figure 2-2. A typical trajectory of the double scroll oscillator projected on the x-y plane

where $F = \frac{B_p(m_0 - m_1)}{G + m_0}$. For record keeping purposes, the two planes are labeled 0 and 1, and can be seen in Figure 2-3 (a).

The chaotic system is allowed to run freely and the locations at which the trajectories of the system intersect the half-planes are recorded. The result is a series of points which can be approximated, using a least squares fit, to a line, see Figure 2-3 (b).

Each line is then partitioned into 2000 sections and the area between each partition is called a bin. The chaotic system is then run from the center of each bin without any controls. The sequence of lobes through which each trajectory passes is recorded using 0's and 1's. This sequence is stored in a coding function, $r(x)$, where x is the initial condition from which the sequence began. For each x , the associated binary sequence a_1, a_2, a_3, \dots is mapped to the binary decimal $0.a_1a_2a_3\dots$, where each a represents either a 1 or a 0, and is defined by

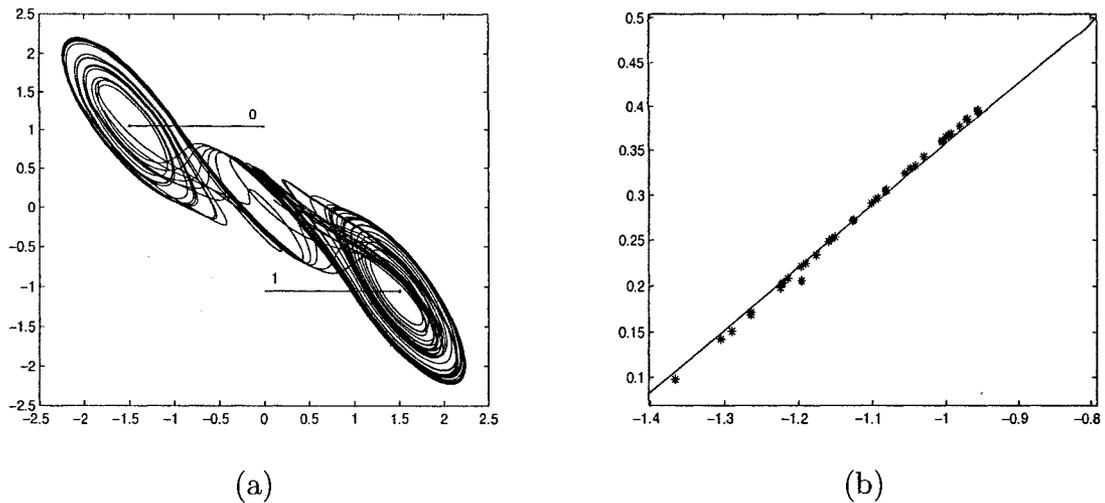


Figure 2-3. Poincaré Surface. (a) Passing through the center of each attractor and (b) Least square approximation of points through control surface

$$r(x) = \sum_{n=1}^{\infty} \frac{a_n}{2^n}.$$

The coding function can be seen in Figure 2-4 .

In generating the cupolets, the coding function is truncated to a length N and defined by

$$r_N(x) = \sum_{n=1}^N \frac{a_n}{2^n}.$$

Given any point x_0 on the line passing through the Poincaré surface, there exists a neighborhood $N_\delta(x_0)$, $\delta > 0$ around x_0 such that for every point $x \in N_\delta(x_0)$, x and x_0 have the same symbolic sequence for N times around the attractor. Therefore, since all $x \in N_\delta(x_0)$ share the same symbolic sequence, they also have the same value of the coding function $r(x)$ for some length N .

Since there is a neighborhood around every x_0 on the Poincaré surface with the same $r_N(x)$, there is also a set outside the neighborhood which differs from $r_N(x)$.

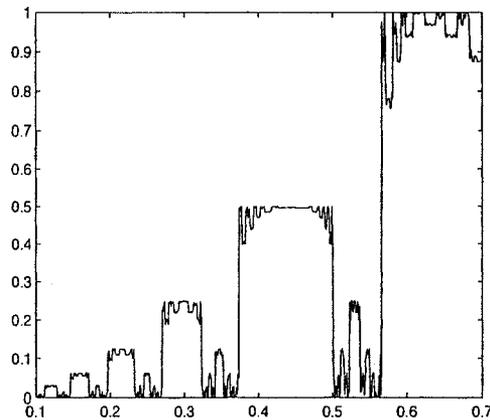


Figure 2-4. Coding function $r(x)$

Therefore, there exists a point x_1 , such that x_0 and x_1 pass through the same lobes $N - 1$ times around the attractor and differ only at the N^{th} lobe. This means that $r_N(x_0)$ and $r_N(x_1)$ are the same except for the last digit. The point x_1 can be explicitly found for any x_0 along the line passing through the Poincaré surface by $r_N(x_0) - r_N(x_1) = 2^{-N}$. This procedure is done for every bin on each Poincaré surface. Each time, the center of the bin is denoted by x_0 and the point x_1 is found such that it is the center of the closest bin on the same lobe whose $r_N(x_0) - r_N(x_1) = 2^{-N}$. This information is stored in a matrix called M_{bins} .

The scheme adapted by Parker and Short requires a control sequence, composed of 0's and 1's, to be sent into the system. A simulation is run and is able to progress freely until it passes through a Poincaré surface. At this point, a microcontrol is applied and the trajectory of the system is shifted to the center of the bin through which it passed. Once the trajectory is in the center of the bin, the first control in the control sequence is applied. If the control is a 0, the trajectory is free to continue

without any additional controls. If the control is a 1, the trajectory of the system is shifted to the nearest bin, which results in a trajectory which will be on a different lobe after N loops around the attractor. The procedure of applying a microcontrol, which shifts the trajectory to the center of the bin, and then implementing the next control in the control sequence is applied repeatedly to the system. A 16 bit control sequence is continually repeated and in almost all cases, the trajectory closed up on itself. In some cases, the 16 bit control sequence, if applied from a different initial condition, did not produce the same orbit. These sequences were discarded. The remaining 16 bit control sequences stabilize onto unique chaotic unstable periodic orbit-lets regardless of initial condition.

2.3 Application and Properties of Cupolets

The technique of creating cupolets, described in Section 2.2, is applied using 16 bit control sequences, the result of which is approximately 8,800 cupolets. It has been shown that cupolets are useful for several applications including secure communications, data and music compression, and image processing. These are the applications which have been investigated thus far, but someday the understanding of cupolets could be used to develop control methods for other nonlinear systems such as fluid flow, weather events, and maneuvering objects in outer space.

There are several properties about cupolets which need to be described to ensure it is indeed possible to transition between cupolets. The first property is that every initializing cupolet is generated independently of initial condition. This means it is possible to transition from one cupolet to the next by simply changing the control

sequence. Since cupolets are generated independent of initial condition, it is possible to begin sending in the controls for the second cupolet at any location on the first orbit and the dynamical evolution will stabilize onto the second orbit.

The second important property is that every cupolet has microcontrols applied to it. The microcontrols shift the trajectory to the center of every bin through which it passes. The microcontrols not only minimize the accumulation of round off error, but also suggest that some cupolets will intersect. This intersection will occur exactly in the center of a bin. If we are transitioning between two cupolets which pass through the same bin, it is possible to switch from one cupolet to the next at their intersection by sending in the second control sequence.

CHAPTER 3

SWITCHING BETWEEN INTERSECTING CUPOLETS

3.1 Introduction

In this Chapter, we will discuss different methods of transitioning between cupolets. The first transitioning technique, described in Section 3.2, is a method which will allow for a transition between two cupolets which do not intersect. We then define a metric which allows us to measure the length of the blind transitions. In Section 3.3 we describe a method for transitioning between cupolets which intersect.

3.2 Blind Transition

In this section we will describe a method which can be implemented to transition between any two cupolets. The blind transition utilizes the property, described in Section 2.3, that cupolet creation is independent of initial condition. In developing the blind transition between two cupolets, a library of cupolets is used. This library contains all 8,800 cupolets in separate files, each containing the x , y , and z coordinates.

The algorithm for switching from any cupolet A to a cupolet B is as follows. First, the control sequence for cupolet A is sent into the dynamical system which creates the cupolet as described in Section 2.3. When cupolet A has completed at least one period, the control sequence governing cupolet B, is sent into the system. We begin with our original blind transition which sends the control sequence for cupolet B into

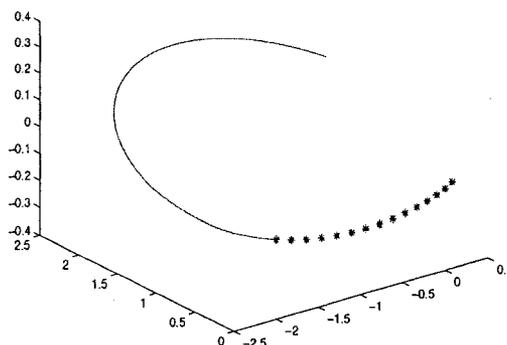


Figure 3-1. Continuous switch from the solid line, cupolet 'C00', to the dotted line of the transient

the system at the point on cupolet A which is closest to the attractor on the first lobe. This location is chosen for our convenience. The original blind transition leaves the first cupolet in a continuous manner as seen in Figure 3-1. The control sequence for cupolet B is then periodically repeated until the transition stabilizes onto cupolet B (see Figure 3-2) and completes one entire period. We define the original blind transition to be the section of the transition between the point it leaves the cupolet A to the point it stabilizes onto cupolet B.

The original blind transition works in transitioning between every pair of cupolets. Unfortunately, since the original blind transition is set free in the non-linear system, there is no way of predicting how long the transition will be. In an effort to predict the length of a blind transition, we began experiments in which we left the first cupolet from different locations around the orbit. To do this, it was first necessary to find the exact bins through which each cupolet passes. Due to the way the cupolets are created, where each cupolet passes through the center of each bin, this is easily done

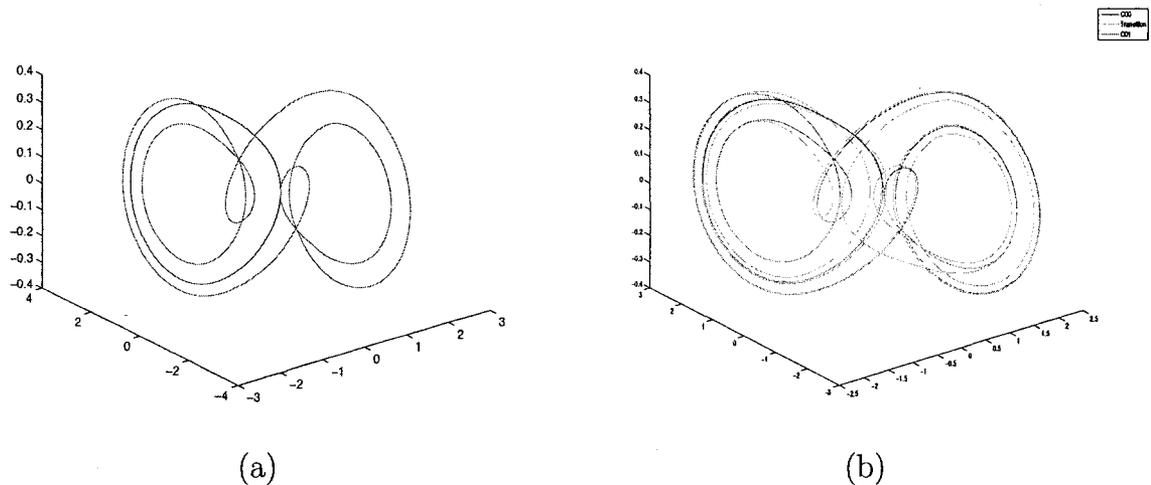


Figure 3-2. Blind Transition (a) the blue orbit is cupolet '00' and the red orbit is cupolet '01' (b) the green line is the transition

by tracing the route of each cupolet. When a cupolet passes through a control plane, the data recorded includes control implemented, the bin number, and corresponding lobe. Once this is done for every cupolet, we initiated our blind transition from each bin (which has a corresponding lobe) on the first orbit. Our initial guess was that transitions initiated from bins which were close to the second orbit would be shorter, but we could not detect a correlation between the two variables.

It is important to note the cyclic nature of the control sequence. To remain stabilized onto a cupolet, the control sequence is periodically repeated. For our second experiment we sent in every cyclic permutation of the second control sequence into the system beginning from every bin on the first orbit. Again, we were unsuccessful in trying to predict a blind transition which would be shorter than the rest. What we did discover however was that in some cases, the length of the transition between two cupolets was zero. This occurred when the correct cyclic permutation of the

second cupolet was sent in from a bin through which the first and second cupolet both passed. We describe these transitions in more detail in Section 3.3.

It is necessary to determine a method of measuring the transitions so they can be compared. We have chosen to measure the length of the transition by the number of loops around the attractor the transition must make in getting from one cupolet to the next. We begin counting the loops after leaving the first cupolet and include the final loop where the transition stabilizes onto the second cupolet. It is important to note that there is more than one length for each pair of cupolets since it is possible to initiate the transition from anywhere on the first orbit with any cyclic permutation of the second control code.

The length, \mathcal{L}_{Blind} , of the blind transition between any two cupolets, is a map

$$\mathcal{L}_{Blind} : L \times B \times C \rightarrow \mathbb{R}$$

where L is the lobe from the initial point on the first orbit, B is the bin from the initial point on the first orbit, C is the cyclic permutation of the second control sequence sent into the system, and \mathbb{R} is the set of all real numbers. The length, \mathcal{L}_{Blind} , will not necessarily be the same for different bins on the first cupolet and for different cyclic permutations of the second control sequence. We did several experiments where we compared the length of original blind transition, the blind transitions from different bins with different cyclic permutations of the second control sequences, and the controlled transition described in Section 4.2. The comparison of these transitions can be seen in Chapter 5.

3.3 Switchable Cupolets

The next transitioning scheme is implemented if two cupolets intersect. As stated in Section 2.3, it is possible for two cupolets to intersect if they pass through the same bin. As mentioned in the Section 3.2, if a blind transition is initiated in the bin through which two cupolets intersect and the correct cyclic permutation of the second control sequence is implemented, the length of the blind transition is zero. We think of this special blind transition as a simple switch between the two cupolets and say the cupolets are "switchable."

To determine if two cupolets are switchable, it is first necessary to examine the exact bins through which each cupolet passes. A quick comparison between the two sets of bins is done to see if any are the same. If the two cupolets pass through at least one bin which is the same, the next step in switching between two cupolets, let's say cupolet A and cupolet B, is to determine which cyclic permutation of the control sequence for cupolet B should be sent into the system. To do so, the bin, lobe, and control information about cupolet B is analyzed. It is important to remember the cyclic nature of the control sequence. To remain stabilized onto a cupolet, the control sequence is periodically repeated. At every point where the cupolet passes a control plane, the same control is implemented. Thus, to switch from cupolet A to cupolet B, in a specific bin, the correct cyclic permutation of the control sequence must be implemented to ensure the trajectory stabilizes onto cupolet B.

Let us assume that the bin, lobe, and control information for some cupolet A and cupolet B is shown in Table 3.1. In this case, the control sequence for cupolet A is '100100' and the control sequence for cupolet B is '101100'.

A quick scan of the bin and lobe information between cupolet A and cupolet B reveals that the two cupolets are in fact switchable, since they both pass through bin 133 on lobe 1. Thus, the location which will result in a zero-length transition between cupolet A and cupolet B is bin 133 on lobe 1. To switch between the two cupolets, the correct cyclic permutation of the second control sequence should be sent while in bin 133 on lobe 1. Referring to Table 3.1, we can see that cupolet B requires a “0” control in bin 133, and it corresponds to the underlined control in the control sequence for B: '101100'. There is only one cyclic permutation of the control sequence which will result in instantaneous stabilization onto cupolet B. That cyclic permutation begins with the underlined control in the sequence and continues the control sequence. The correct permutation of the control sequence is '001011' and the length of this transition $\mathcal{L}_{Blind}(1, 133, 001011) = 0$.

The reason only one permutation of the control sequence will work is due to the fact that cupolets are derived from a chaotic system and if a wrong control is implemented in a bin, the trajectory will be perturbed and will undergo transient chaos until it re-stabilizes onto the cupolet. When the correct control sequence is implemented in the bin shared by two cupolets, the blind transition from the first cupolet instantly stabilizes onto the second cupolet and thus, is no longer a chaotic and unpredictable transition. The length of the blind transition between switchable cupolets is always zero.

CHAPTER 4

SWITCHING BETWEEN NON-INTERSECTING CUPOLETS

4.1 Introduction

In this chapter will describe several ways to transition between cupolets which do not intersect using simple switches between intersecting cupolets, called controlled transitions. We begin by describing controlled transitions in Section 4.2. We then describe several approaches of determining the controlled transition which has the smallest length, where the length of the transition is defined in Section 4.3.

4.2 Controlled Transitions

In Section 3.3, we described a way in which it is possible to switch between cupolets which pass through the same bin on the same lobe. The blind transition when switching between intersecting cupolets is of zero-length and only requires the change from one control sequence to another. Using this idea, it is possible to transition from any cupolet to any other cupolet, regardless of whether they intersect. This is done by starting at the initial cupolet and making a finite number of switches between intersecting cupolets until the terminal cupolet is reached. We will refer to this type of transition as a controlled transition, since we know the exact trajectory the transition will take in getting from one cupolet to the next.

The controlled transition will be defined by a transition function, T . The transition function, T , takes as inputs a cupolet C_1 , the initial cupolet in the transition, and a cupolet C_N , the terminating cupolet in the transition. The transition function then returns a sequence of cupolets beginning with C_1 and ending with C_N . An example of this is

$$T(C_1, C_N) = \{C_1, C_2, \dots, C_N\}.$$

Since the transition function defines a controlled transition, adjacent elements in the sequence are switchable. For example, C_1 is switchable with C_2 and C_{N-1} is switchable with C_N . If C_1 and C_N are switchable, there are no intermediate cupolets and thus $T(C_1, C_N) = \{C_1, C_N\}$.

Let us say we want to get from some cupolet A to a different cupolet C. As these cupolets evolve around the attractor, they visit a sequence of bins and are subject to controls from the control sequence. We have constructed a hypothetical example where the bin, lobe, and control information for four cupolets A, B, C, and D is shown in Figure 4-1.

A quick scan of the bin and lobe information for cupolet A and cupolet C reveals that they do not intersect, thus are not switchable. Fortunately, there exists another cupolet, B, which intersects both cupolet A, at bin 133 on lobe 1, and cupolet C, at bin 1987 on lobe 1. Thus, to transition from cupolet A to cupolet C, one possible controlled transition is $T(A, B) = \{A, B, C\}$.

The first step in the transition from cupolet A to cupolet C is to follow cupolet A until it reaches bin 133 on lobe 1. At this point, a permutation of the control sequence for cupolet B is applied. The permutation of the control sequence resulting

Cupolet A			Cupolet B		
Bin	Lobe	Control	Bin	Lobe	Control
133	1	1	1023	1	1
832	1	0	1684	0	0
245	0	0	1987	1	1
45	0	1	579	1	1
1522	1	0	133	1	0
1943	0	0	34	1	0
133	1	1	1023	1	1
⋮	⋮	⋮	⋮	⋮	⋮

(a)
(b)

Cupolet C			Cupolet D		
Bin	Lobe	Control	Bin	Lobe	Control
1462	1	1	1552	1	0
1734	1	0	177	0	1
3	0	1	513	0	0
1798	0	1	1005	0	1
1987	1	0	142	1	0
749	1	1	3	0	1
1462	1	1	1552	1	0
⋮	⋮	⋮	⋮	⋮	⋮

(c)
(d)

Figure 4-1. The information about bins, lobes, and controls (a) for cupolet A control sequence '100', (b) for cupolet B with control sequence '101100', (c) for cupolet C with control sequence '101', and (d) for cupolet D with control sequence '01'

in immediate stabilization with cupolet B is '001011'. The controlled transition will follow the trajectory of cupolet B through bin 34 on lobe 1, through bin 1023 on lobe 1, and then will reach bin 1987 on lobe 1. It is in this bin that cupolet B and cupolet C intersect. To stabilize immediately onto cupolet C, the permutation of the control sequence for cupolet C, namely '011,' is sent into the system. It is clear that the controlled transition from cupolet A to cupolet C is not zero length since the transition follows the trajectory of cupolet B in getting from bin 133 on lobe 1 to bin 1987 on lobe 1. It is necessary to develop a way of measuring the length of a controlled transition.

4.3 Measuring Transitions

It is necessary to construct a way of measuring the length of a controlled transition so different transitions can be compared, since we are ultimately searching for the shortest transition. The length, $\mathcal{L}_{Controlled}$, will be similar to the length, $\mathcal{L}_{Chaotic}$, as described in 3.2, since it will measure the length of the controlled transition by accounting for the number of loops around the attractor the transition must make in getting from one cupolet to the next. The length of a controlled transition, $\mathcal{L}_{Controlled}$, between any two cupolets, is a map

$$L_{Controlled} : C \rightarrow \mathbb{R}$$

where C is a sequence of two or more cupolets and \mathbb{R} is the set of all real numbers.

The number of elements in the sequence of cupolets sent into $\mathcal{L}_{Controlled}$ can vary. A sequence containing two elements occurs when $T(C_1, C_2) = \{C_1, C_2\}$, meaning the

two cupolets are switchable. In this case

$$\mathcal{L}_{Controlled}(\{C_1, C_2\}) = \mathcal{L}_{Controlled}(\{C_2, C_1\}) = 0,$$

since the transition is instantaneous and does not have to make any loops around the attractor.

If a sequence containing three elements $\{C_1, C_2, C_3\}$ is being measured, it is important to note that adjacent cupolets are switchable (i.e. C_1 and C_2 are switchable, C_2 and C_3 are switchable). The only time the controlled transition will loop around the attractor will be while riding C_2 in getting from the switching point between C_1 and the switching point between C_3 . We begin counting the loops around the attractor after the switch with C_1 and include the final loop where C_2 switches with C_3 . Due to the construction of the bins, it is possible to count the loops around the attractor by instead counting the bins which are visited while on C_2 .

From the example described Section 4.2, the length the controlled transition $T(A, C) = \{A, B, C\}$ will be $\mathcal{L}_{Controlled}(\{A, B, C\}) = 4$. This is because we simply count the bins which must be visited while riding cupolet B. In this case, the initial switch from cupolet A occurs in bin 133 on lobe 1 so we begin our count after this point. We follow the trajectory of cupolet B through bin 34 on lobe 1 (one), bin 1023 on lobe 1 (two), bin 1684 on lobe 0 (three), and finally we reach bin 1987 on lobe 1 (four). At this point we are able switch to cupolet C and the count is over.

It is also possible to transition from C to A through B but the length of the controlled transition will not be the same. In this case, $\mathcal{L}_{Controlled}(\{C, B, A\}) = 2$ since the controlled transition switches to cupolet B in bin 1987 on lobe 1 then passes

through two bins and reaches the switching point with cupolet A. From this simple example it is clear that

$$\mathcal{L}_{Controlled}(\{A, B, C\}) \neq \mathcal{L}_{Controlled}(\{C, B, A\}).$$

In the case where there are more than three elements in the sequence of the controlled transition being measured, it is possible to split the sequences into smaller three element sequences. Let us say we want to know the length of the controlled transition

$$T(C_1, C_N) = \{C_1, C_2, C_3, \dots, C_{N-1}, C_N\}.$$

To determine the length of this transition, we must determine the number of loops around the attractor which must be visited by the intermediate C_2, C_3, \dots, C_{N-1} . Since the number of bins which must be visited by any intermediate cupolet relies only on the location of the switch between the two adjacent cupolets, we can split the length of large sequence into the sum of the lengths of the three adjacent element sequences. In this case, we can split the length of the sequence as follows

$$\begin{aligned} \mathcal{L}_{Controlled}(\{C_1, C_2, C_3, \dots, C_{N-1}, C_N\}) &= \mathcal{L}_{Controlled}(\{C_1, C_2, C_3\}) \\ &+ \mathcal{L}_{Controlled}(\{C_2, C_3, C_4\}) \\ &+ \dots \\ &+ \mathcal{L}_{Controlled}(\{C_{N-2}, C_{N-1}, C_N\}). \end{aligned}$$

Now, to measure the length of a sequence with four or more elements, we need only to determine the length of the three element sequences as described previously.

There are often several different controlled transitions between two non-intersecting cupolets. For example, let us say we want to transition between cupolet A and cupolet C, but this time we wish to transition through cupolet D, as shown in Table 4-1. The length of this transition will be $\mathcal{L}_{Controlled}(\{A, D, C\}) = 5$. If we compare this transition to the one which instead uses cupolet B as its intermediate cupolet, with $\mathcal{L}_{Controlled}(\{A, B, C\}) = 4$, we see that the transition with the smaller length is $T(A, C) = \{A, B, C\}$. In the following sections we will describe methods of determining all the different controlled transition between two cupolets so the transition with the shortest length can be found.

4.4 Basis for Cupolets

In this section, we will describe a method to determine the controlled transition with the shortest length between any two cupolets. This method involves determining a set of connected cupolets which span, or visits, every bin. The motivation for this method came from the idea that if there existed one super cupolet which passed through every bin, this cupolet could be used as the intermediate cupolet in the controlled transition between any two cupolets. Unfortunately, no super cupolet exists.

It is first necessary to determine a set of cupolets which are connected and span every bin. We will refer to this set of cupolets as a quasi-basis for the cupolets. We use the term quasi-basis because the construction of this set is analogous to a basis in n-dimensional space where the set of vectors in the basis can reach every point in the n-dimensional space. In our case, the quasi-basis will be a set of cupolets which

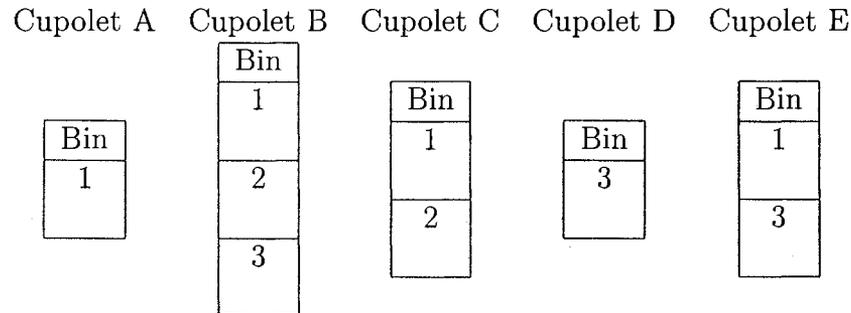


Figure 4-2. The bin information for five cupolets in a bin space with three bins

can reach every one of the 4,000 bins. We will refer to this set of bins as a bin space, which is simply a set containing all the bins which must be reached by the quasi-basis.

The set of cupolets which will be used as a quasi-basis for the bin space must be connected, meaning every cupolet is switchable with at least one other cupolet. This is due to the fact that if one of the cupolets in the quasi-basis is not connected it does not intersect with any of the other cupolets in the quasi-basis and it is impossible to switch between the non-connected cupolet and the rest of the quasi-basis. Thus, this non-connected cupolet does not add anything to the quasi-basis.

To demonstrate the idea of a basis, let us use an example where there are three bins in the bin space and five cupolets, as shown in Figure 4-2. Using this scenario, one may want to choose the set $\{C,D\}$ as the quasi-basis for the bin space since the two cupolets pass through bin 1, 2, and 3. It is clear that bin 1 and bin 2 are connected through cupolet C. Unfortunately, cupolet C and D are not connected since they do not share any of the same bins and thus, it is impossible to switch between them. Therefore, this choice does not work as a quasi-basis.

Another guess for the quasi-basis of the bin space using the example from Figure 4-2 is the set containing $\{B\}$. This cupolet connects the three bins and reaches all bins in the bin space. Using $\{B\}$ it is possible to make controlled transitions between any two cupolets. One such transition is $T(A, D) = \{A, B, D\}$. First, the switch is made from cupolet A and B in bin 1. The transition then travels along cupolet B through bin 2 then reaches bin 3 where the switch is made from cupolet B to the terminating cupolet, D. In this case, $\mathcal{L}_{Controlled}(\{A, B, D\}) = 2$. On the other hand, it is also possible to make a controlled transition between cupolet A and D by switching through cupolet E, where $\mathcal{L}_{Controlled}(\{A, E, D\}) = 1$.

Clearly, finding a quasi-basis for the bin space is possible in some cases. Unfortunately, the distance between two cupolets is not necessarily minimized by restricting all switches to be through the elements in the quasi-basis. In the example described above, the length of the controlled transition between cupolet A and D using the quasi-basis $\{B\}$ is two, and the length of the transition using cupolet E, not in the quasi-basis, is one. As a result, determining a quasi-basis for the cupolets may not be the most useful tool in determining the controlled transition with the shortest length between any two cupolets.

4.5 Modeling Cupolets using Graph Theory

The next method we used in trying to determine the controlled transition with the shortest length between any two cupolets utilizes graph theory. We first chose a model which designated the bins as vertices and the cupolets as edges. There are several types of graphs which can be constructed using this model. The two types of

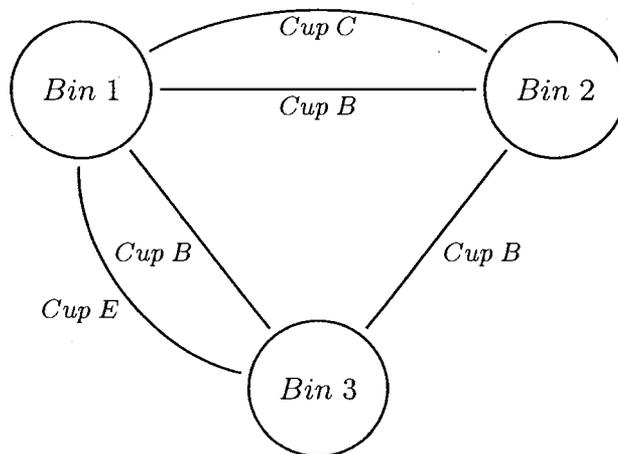


Figure 4-3. Undirected graph of the cupolets A,B,C, D and E in the three bin space. The vertices represent bins and the edges represent the cupolets which connect those bins.

graphs we chose are an undirected graph and a weighted directed graph (digraph). The undirected graph of the cupolets described in Figure 4-2 is shown in Figure 4-3.

In modeling the cupolets using an undirected graph, we are trying to determine the controlled transition with the smallest length between any two cupolets. To do so, we can determine the controlled transition leaving from any bin from the first cupolet and arriving at any bin on the second cupolet. In this way, we construct a number of bin pairs. The first bin in the pair corresponds to a bin on the first cupolet and the second bin in the pair corresponds to a bin on the second cupolet. If, for example, we wanted to get from cupolet B to cupolet C we would have the bin pairs listed in Table 4.1 (this is a trivial example since we know cupolet B and cupolet C are switchable). We then determine if there is a path in the graph which allows us to get from the first bin in the pair to the second.

In this undirected graph, the edges represent a cupolet which pass through two bins, or vertices. As you can see from the undirected graph, the bins are all connected,

Cupolet B	Cupolet C
1	1
1	2
2	1
2	2
3	1
3	2

Table 4.1. The set of bin pairs for which the shortest path must be determined or order to find the controlled transition with the smallest length from cupolet B and cupolet C

thus, it is possible to switch between any two bins. Since it is possible to switch between any two bins, it is also possible to get from any cupolet to another. It is important to note that singleton loops are not allowed in this graph. The reason they are excluded is due to the fact that they do not provide any additional information about the connectedness of the bins. If singleton loops were allowed, they would represent a cupolet which passed through the bin, and thus, connected the bin to itself.

One important characteristic in the undirected graph, as shown in Figure 4-3, is that in this case, all of the vertices are adjacent. Two vertices are said to be adjacent if there exists an edge between them. This information can be demonstrated in an adjacency matrix, A , where the element, A_{ij} , is denoted by 1 one if there exists an edge between vertex i and j and a 0 if there does not exist an edge between vertex i and j . The adjacency matrix is always symmetric since the graph is undirected. For our three bin space, the adjacency matrix is

$$A = \begin{bmatrix} \text{Bin 1} \rightarrow \text{Bin 1} & \text{Bin 1} \rightarrow \text{Bin 2} & \text{Bin 1} \rightarrow \text{Bin 3} \\ \text{Bin 2} \rightarrow \text{Bin 1} & \text{Bin 2} \rightarrow \text{Bin 2} & \text{Bin 2} \rightarrow \text{Bin 3} \\ \text{Bin 3} \rightarrow \text{Bin 1} & \text{Bin 3} \rightarrow \text{Bin 2} & \text{Bin 3} \rightarrow \text{Bin 3} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

where the diagonal elements of A are zero because singleton loops are not allowed in our graph. If singleton loops were allowed, the result would be ones along the diagonal because for every bin which can be reached by a cupolet is therefore connected to itself.

Although the adjacency matrix is informative about the connectedness of the graph, it does not reveal any information about which cupolet gets us from one bin to the next. If instead we filled the adjacency matrix with name of the edge between the two vertices, we would end up with the following

$$A = \begin{bmatrix} \text{Bin 1} \rightarrow \text{Bin 1} & \text{Bin 1} \rightarrow \text{Bin 2} & \text{Bin 1} \rightarrow \text{Bin 3} \\ \text{Bin 2} \rightarrow \text{Bin 1} & \text{Bin 2} \rightarrow \text{Bin 2} & \text{Bin 2} \rightarrow \text{Bin 3} \\ \text{Bin 3} \rightarrow \text{Bin 1} & \text{Bin 3} \rightarrow \text{Bin 2} & \text{Bin 3} \rightarrow \text{Bin 3} \end{bmatrix} = \begin{bmatrix} 0 & \text{CupB} & \text{CupB} \\ \text{CupB} & 0 & \text{CupB} \\ \text{CupB} & \text{CupB} & 0 \end{bmatrix}$$

The problem with this adjacency matrix is that there exists a different adjacency matrix

$$A = \begin{bmatrix} 0 & \text{CupC} & \text{CupE} \\ \text{CupC} & 0 & \text{CupB} \\ \text{CupE} & \text{CupB} & 0 \end{bmatrix},$$

and thus, the adjacency matrix is not unique. Another problem with this adjacency matrix is that it does not contain any information about the length of the transition between the two bins. For example, it is possible to use either cupolet B or cupolet E in getting from bin 1 to bin 3. In this case, however, the number of bins which must be visited in going from bin 1 to bin 3 is two if traveling on cupolet B, while

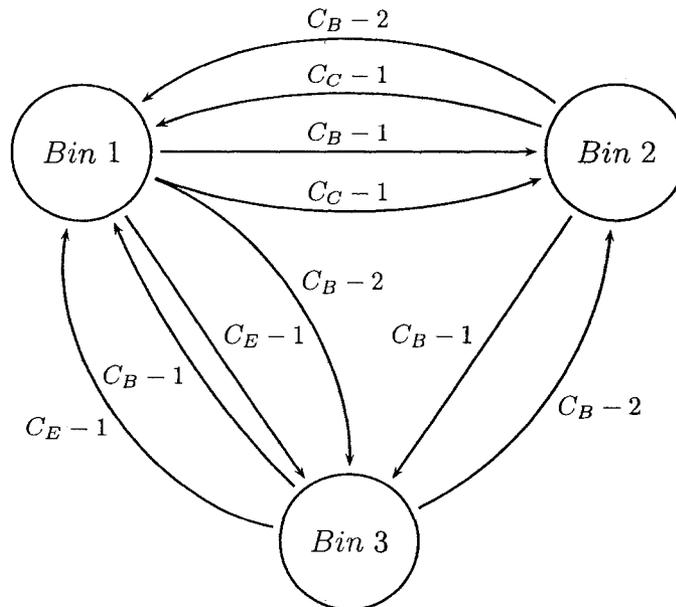


Figure 4-4. Weighted multi-digraph of the cupolets A,B,C, D, and E in the three bin space. The vertices represent bins and the directed edges represent cupolets where the subscript is the name of the cupolet and the number represents number of bins visited if going from one bin to the next on that cupolet.

the number of bins is one if traveling on cupolet E. This information is imperative and as such we will next use a weighted digraph to represent the bins and cupolets.

A weighted digraph is a graph where each edge is replaced with a directed edge that carries a weight. In a weighted digraph, it is only possible to have one directed edge from one vertex to another. As such, if there are two cupolets which connect two bins, as shown in the multi-digraph in Figure 4-4, we choose the cupolet which has the smallest weight. In Figure 4-5, a weighted digraph is used to model the cupolets in the space containing three bins as described in Figure 4-2.

Again we do not allow singleton loops in the weighted digraph because they do not provide any additional information. It is now possible to construct an adjacency matrix from the weighted digraph. This adjacency matrix again has zeros along the diagonal, but will not be symmetric. The entries of the adjacency matrix, A_{ij} , for

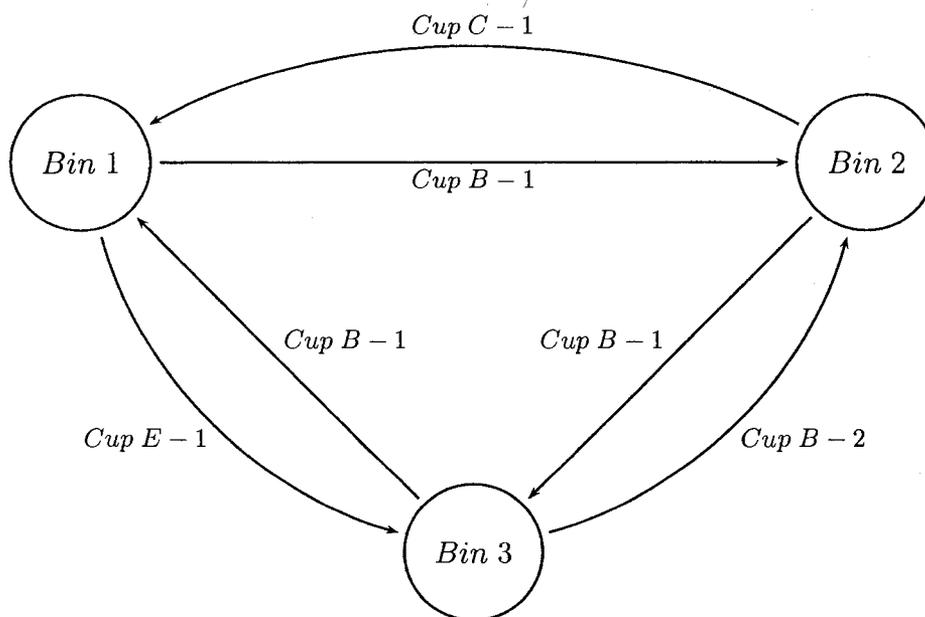


Figure 4-5. Weighted digraph of the cupolets A, B, C, D, and E in the three bin space. The vertices represent bins and the directed edges represent cupolets and how many bins must be visited to get from one bin to the next.

the weighted digraph will contain the weight between vertex i to vertex j .

$$A = \begin{bmatrix} \text{Bin 1} \rightarrow \text{Bin 1} & \text{Bin 1} \rightarrow \text{Bin 2} & \text{Bin 1} \rightarrow \text{Bin 3} \\ \text{Bin 2} \rightarrow \text{Bin 1} & \text{Bin 2} \rightarrow \text{Bin 2} & \text{Bin 2} \rightarrow \text{Bin 3} \\ \text{Bin 3} \rightarrow \text{Bin 1} & \text{Bin 3} \rightarrow \text{Bin 2} & \text{Bin 3} \rightarrow \text{Bin 3} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 2 \\ 1 & 1 & 0 \end{bmatrix}.$$

Since it is possible to model the connected bins as a weighted digraph, we could use Dijkstra's algorithm to determine the path with the smallest weight between any two bins. Once this path is found for every bin pair between the two cupolets, the path with the smallest weight is chosen and that becomes the length of the controlled transition.

With the simple example of five cupolets in a three bin space, the maximum number of bin pairs which need to be checked is six, which occurs in the transition between cupolet B and cupolet C. In the set of over 8,800 cupolets with 4,000 bins,

the number of bin pairs can be very large. Some cupolets pass through as many as 50 bins and as such, would have to compare all 50 bins with those bins of the target cupolet. The number of pairs to be check is $m * n$ where m is the number of bins which the first cupolet pass through and n is the number of bins the second cupolet passes through. Therefore, the number of bin pairs for which the transition must be calculated can be as high as 2500 for some cupolets.

To speed up calculations, it is possible to first determine the transition of shortest length between every possible bin pair. The information about the length of the transition and the cupolets which must travelled between each bin pair can be stored in a table. Then, when searching for the shortest controlled transition between any two cupolets, one can look in the table at the information for each bin pair between the two cupolets, compare the lengths, and find the shortest transition.

4.6 Dijkstra's Shortest Path Algorithm

To determine the path with the smallest weight between any two vertices we could use Dijkstra's shortest path algorithm [1]. The path with the smallest weight between any two vertices corresponds to the controlled transition with the smallest length between two bins. To determine this path we will use a directed graph, as described in Section 4.5, where the vertices represent bins and an edge represents the cupolet which connects the two bins (vertices) while passing through the least number of intermediate bins. We will use the number of bins the cupolet must pass through in getting from one bin to another as the weight of the edge, and store the name of the cupolet in a separate table.

It is important to note that if there is a cupolet which connects some bin A to another bin B, then that same cupolet also connects bin B to bin A. Therefore, every directed edge has a corresponding edge in the opposite direction. The weight of these corresponding edges are not necessarily the same since we only traverse along a cupolet in one direction. Also, there may not exist an edge between a set of vertices. This occurs if no one cupolet passes through both of the bins.

To implement Dijkstra's shortest path algorithm, it is necessary to have a starting vertex. In this case, the vertices represent bins and, as such, the starting vertex will be the first bin in the bin pair which corresponds to a bin through which the first cupolet passes. Using Dijkstra's algorithm, we will be able to determine the weight of the path corresponding to every bin pair with the starting vertex as the first element in the bin pair.

To demonstrate Dijkstra's algorithm we will use the graph shown in Figure 4-6 and set our first starting vertex to be vertex one. Using vertex one as our starting vertex, will be able to determine the weight of the path for the bin pairs (1,1), (1,2), (1,3), (1,4), and (1,5). Dijkstra's algorithm begins at the starting vertex and visits all the other vertices in the graph. It repeatedly checks the closest, but still unvisited vertices, and determines if the new vertex results in a smaller weighted path than the already known path.

The first step in Dijkstra's algorithm is to construct a weight matrix. The weight matrix corresponding to the weighted digraph in Figure 4-6 is

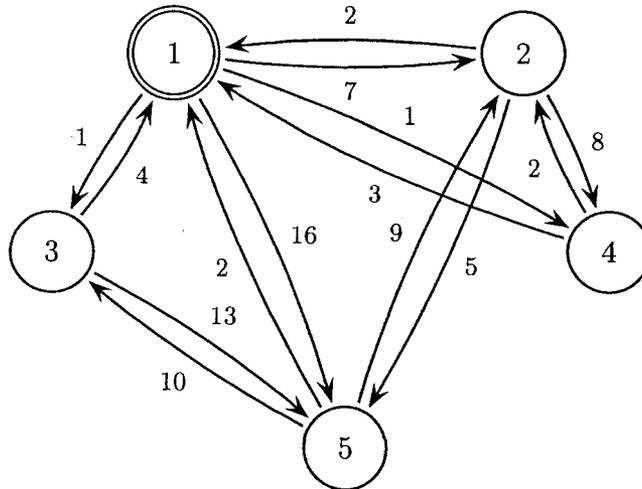


Figure 4-6. An example weighted digraph for which we will determine the shortest path from vertex 1 to every other vertex using Dijkstra's algorithm.

$$M = \begin{bmatrix} 0 & 7 & 1 & 1 & 16 \\ 2 & 0 & \infty & 8 & 5 \\ 4 & \infty & 0 & \infty & 13 \\ 3 & 2 & \infty & 0 & \infty \\ 2 & 9 & 10 & \infty & 0 \end{bmatrix}$$

where each element M_{ij} is the weight of the directed line from vertex i to vertex j . The elements of M along the diagonal are zero since the weight is zero to get from a vertex to itself. Other elements in M are set to ∞ because there is not a directed line between the two vertices.

The next step is to construct a vector for the weights. Since our starting vertex is vertex one, the first row of the adjacency matrix M , $w = [0 \ 7 \ 1 \ 1 \ 16]$, will be our weight vector. At this point we imagine we are sitting on vertex one and will set this to be our current vertex, so $CV = \{1\}$. We then split the remaining vertices into two sets, those which have been visited and those which are unvisited. Since we are still

sitting on the first vertex, we have only visited one vertex and thus, $V = \{1\}$, and the rest are unvisited $U = \{2, 3, 4, 5\}$. The goal in the remainder of the algorithm is to determine the shortest path from the starting vertex to every other vertex. This path will either be the single edge with the initial vertex, or it will be a multi-step path through one or more vertices. The algorithm is as follows:

1. Set $CV = \left\{ j \mid w_j = \min_{k \in U} \{w_k\} \right\}$
2. Set $V = V \cup CV$ and $U = U \setminus CV$
3. If $U = \emptyset$, done.
4. $\forall k \in U, w_k = \min\{w_k, w_{CV} + M_{CV,k}\}$
5. Go to step 1.

The weight vector does not change the first pass through the algorithm. For the second pass through the algorithm, the first step is to set $CV = \{3\}$. This means we have now moved from the starting vertex to the closest unvisited vertex. According to the algorithm, we then set $V = \{1, 3\}$ and $U = \{2, 4, 5\}$. We are now checking to see if the weight of the path between the starting vertex and every other unvisited vertex is smaller if the path is directly from vertex one, or if a smaller path exists by passing instead through vertex three.

The first weight we will check is w_2 which is currently seven. We compare this weight with the weight of first passing through vertex three, $w_2 + M_{2,3} = 7 + \infty = \infty$. We find that this weight is ∞ since no edge exists between vertices two and three. Therefore, element two in the weight vector remains a seven. Next, we examine w_4 which is currently set to one. Again, there is no edge between vertices three and four

and the weight remains the same. Finally, we check w_5 which is currently sixteen. We compare $w_5 = 16$ with $w_3 + M_{3,5} = 1 + 13 = 14$ and determine that it is shorter to pass through vertex three in getting from vertex one to vertex five. The table containing the cupolets which must be traveled in getting from vertex one to vertex five is updated to include the information necessary in traveling instead through vertex three.

Dijkstra's shortest path algorithm is continued in a similar fashion and the result is $w = [0 \ 5 \ 1 \ 1 \ 12]$. A careful inspection of the graph in Figure 4-6 confirms that Dijkstra's shortest path algorithm is correct. If we compare the original weight vector and the new weight vector we see that the second and last entries have changed. This is due to the fact that it is shorter to get from vertex one to vertex two through vertex four and also from vertex one to vertex five through vertex two. For the graph in Figure 4-6, where there are only five bins, Dijkstra's algorithm would need to be executed four more times, each time using a different initial vertex, to determine the smallest weighted path for every possible bin pair.

Dijkstra's algorithm could be used to determine the controlled transition with the smallest length between every bin pair for the cupolets where there are 4,000 bins. Then, to determine the shortest transition between any two cupolets, one could compare the possible bin pairs between the two cupolets and determine which pair corresponded to the shortest transition. We instead chose to use a different method which is sub-optimal to Dijkstra's algorithm but is more simple to construct. This method is described in the following section.

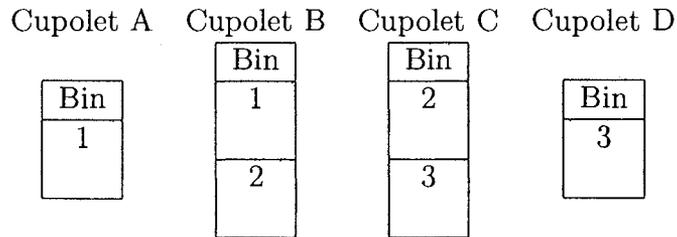


Figure 4-7. The bin information for four cupolets A,B,C and D in a bin space with three bins

4.7 Decomposing Adjacency Matrices

In this section we will describe a way to use adjacency matrices and their properties to determine the shortest length controlled transition with the smallest number of intermediate cupolets between any two cupolets. These adjacency matrices are different from previous matrices in that they do not contain the weight associated with an edge between two vertices, but instead contain a 1 if there exists an edge between the vertices and a zero otherwise. To demonstrate adjacency matrices, we will use an example where there are only four cupolets and three bins, as shown in Figure 4-7.

In creating the adjacency matrix for the system, it is first necessary to construct a graph describing the cupolet and bin information. A simple graph will be used, which is different from the previously described directed graphs, since the vertices will represent cupolets instead of bins. There will be an edge between two cupolets if they pass through at least one of the same bins. The graph for set of four cupolets, described in Figure 4-7, can be seen in Figure 4-8.

It is important to note that while it is possible to have a singleton loop around every vertex, we have chosen not to allow loops since they do not provide any addi-

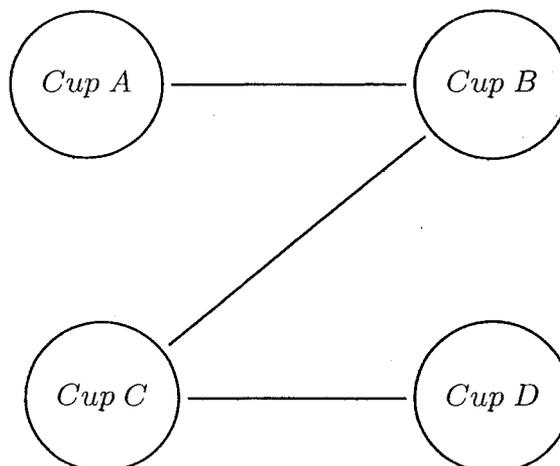


Figure 4-8. Cupolets A,B,C, and D from the three bin space. The vertices represent cupolets and the edges mean that both cupolets pass through the same bin.

tional information since we know every cupolet shares at least one bin with itself. We have also chosen not to specify the bin(s) corresponding to each edge. This was done because a simple comparison of the bin information between both cupolets is enough to determine all the bins through which they intersect.

The adjacency matrix can now be constructed using the entries 0 and 1. If an entry in the adjacency matrix A_{ij} is a zero, there is no edge between vertex i and j , which means cupolet i does not have a common bin with cupolet j . If the entry in the adjacency matrix A_{ij} is a one, there is an edge between vertex i and j meaning the cupolets have at least one bin in common. The adjacency matrix for the graph in Figure 4-8 is as follows where Cup A is represented by the first row and column, Cup B is represented by the second row and column, and so on.

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

One special property about this type of adjacency matrix, as described in [5], is that if we multiply A by itself r times, the element in $(A^r)_{ij}$ stands for the number of r -step paths between vertex i and vertex j , where " r -step" means the number of edges in the path. In other words, $(A^r)_{ij}$ is the number of ways one can get from cupolet i to cupolet j visiting r intermediate cupolets. Using the graph from Figure 4-8, we can compute A^2 and A^3 which are as follows

$$A^2 = AA = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 1 \\ 1 & 0 & 2 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

$$A^3 = AAA = \begin{bmatrix} 0 & 2 & 0 & 1 \\ 2 & 0 & 3 & 0 \\ 0 & 3 & 0 & 2 \\ 1 & 0 & 2 & 0 \end{bmatrix}$$

Let us assume we are trying to transition from cupolet B to cupolet C. Since $A_{2,3} = 1$ we see that we can switch between the two cupolet since the entry is a one. If we were trying to find another path between cupolet B and cupolet C we could examine $(A^3)_{23} = 3$ where $r = 3$. This entry means there are three, 3-step paths between cupolet B and cupolet C and the three paths are shown in Figure 4-9.

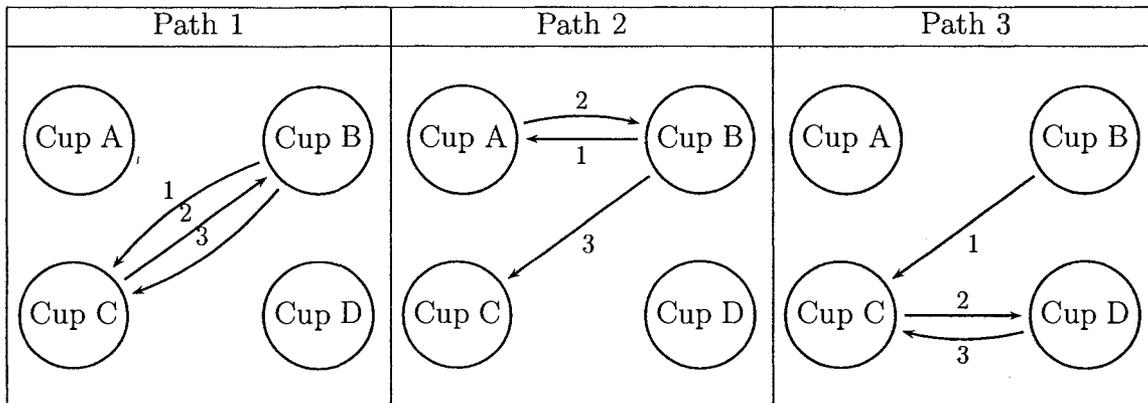


Figure 4-9. Three different three step paths between Cup B and vertex Cup C. The directed edge with a 1 is the first step in the path, the directed edge with a 2 is the second step in the path, the directed edge with a 3 is the third step in the path.

Determining the number of paths which require the smallest number of intermediate cupolets between any two vertices is the first step in determining the transition between any two cupolets. This is done for every cupolet pair (i, j) by finding the smallest r such that $(A^r)_{ij} \neq 0$. Once this is done, it is possible to determine the exact paths between the two cupolets. The method we chose involves decomposing the matrix A^r to expose the reason why $(A^r)_{ij} \neq 0$.

To determine the exact path between any two cupolets we must first examine the structure inherent in powers of symmetric matrices. First, we will denote the row of a matrix by the symbol representing the matrix, subscripted with an index. For example, the third row in matrix C will be C_3 , and the fourth row in the matrix $DD = D^2$ will be $(D^2)_4$. As such, we can define any matrix using only the rows. Since the adjacency matrix is symmetric, $A = A^T$, we can also define the matrix A using only the columns which equal the transpose of the rows. The two ways of denoting any adjacency matrix A are as follows where we assume $A \in M_{N \times N}$

$$A = \left[\begin{array}{c} A_1 \\ \hline A_2 \\ \vdots \\ \hline A_N \end{array} \right] = \left[(A_1)^T \mid (A_2)^T \mid \dots \mid (A_N)^T \right].$$

One characteristic of symmetric matrices is that any power of a symmetric matrix is still symmetric. Thus, A^2 will be symmetric and is denoted as follows

$$\begin{aligned} A^2 &= \left[\begin{array}{c} A_1 \\ \hline A_2 \\ \vdots \\ \hline A_N \end{array} \right] \left[(A_1)^T \mid (A_2)^T \mid \dots \mid (A_N)^T \right] \\ &= \left[\begin{array}{cccc} A_1(A_1)^T & A_1(A_2)^T & \dots & A_1(A_N)^T \\ A_2(A_1)^T & A_2(A_2)^T & \dots & A_2(A_N)^T \\ \vdots & \vdots & \ddots & \vdots \\ A_N(A_1)^T & A_N(A_2)^T & \dots & A_N(A_N)^T \end{array} \right] \end{aligned}$$

where every element $(A^2)_{ij} = A_i(A_j)^T$ which is equal to the dot product $A_i \cdot A_j$. Therefore, $(A^2)_{ij} = A_i \cdot A_j = A_j \cdot A_i = (A^2)_{ji}$. Since A^2 is symmetric and we can denote A^2 using only its rows or columns as

$$A^2 = \left[\begin{array}{c} (A^2)_1 \\ \hline (A^2)_2 \\ \vdots \\ \hline (A^2)_N \end{array} \right] = \left[((A^2)_1)^T \mid ((A^2)_2)^T \mid \dots \mid ((A^2)_N)^T \right]$$

where each $(A^2)_j = A_j A$.

We will now show that for any A^K , each element $(A^K)_{ij} = A_i(A_j^{K-1})^T$ and each row of $(A^K)_j = A_j A^{K-1}$. We now prove these properties hold $\forall K \geq 2$ using induction. These properties hold for $K = 2$ since

$$(A^2)_{ij} = A_i(A_j)^T \text{ and } (A^2)_j = A_j A.$$

Next we assume that these properties hold for some $K = M$ where

$$(A^M)_{ij} = A_i(A_j^{M-1})^T \text{ and } (A^M)_j = A_j A^{M-1}.$$

As such,

$$\begin{aligned} A^{M+1} = AA^M &= \begin{bmatrix} A_1 \\ \hline A_2 \\ \vdots \\ \hline A_N \end{bmatrix} \left[((A^M)_1)^T \mid ((A^M)_2)^T \mid \dots \mid ((A^M)_N)^T \right] \\ &= \begin{bmatrix} A_1((A^M)_1)^T & A_1((A^M)_2)^T & \dots & A_1((A^M)_N)^T \\ A_2((A^M)_1)^T & A_2((A^M)_2)^T & \dots & A_2((A^M)_N)^T \\ \vdots & \vdots & \ddots & \vdots \\ A_N((A^M)_1)^T & A_N((A^M)_2)^T & \dots & A_N((A^M)_N)^T \end{bmatrix} \\ &= \begin{bmatrix} A_1 A^M \\ \hline A_2 A^M \\ \vdots \\ \hline A_N A^M \end{bmatrix}. \end{aligned}$$

Thus the two properties hold true $\forall K \geq 2$. A third property which can be derived from the second property is $(A^{M+1})_j = ((A^{M+1})_j)^T = (A_j A^M)^T = (A^M)^T (A_j)^T =$

$A^M(A_j)^T$ due to the fact that A^M and A^{M+1} are symmetric. The three properties are

1. $(A^{M+1})_j = A_j A^M$
1. $(A^{M+1})_j = A^M(A_j)^T$
3. $(A^{M+1})_{ij} = A_i(A_j^M)^T$.

Using these three properties we can determine the controlled transition between any two cupolets, or vertices, which require $r - 1$ intermediate cupolets in the transition, or an r -step path in the graph. Recall that for the two vertices i and j , we searched for the smallest r such that $(A^r)_{ij} \neq 0$. Now, we can rewrite $(A^r)_{ij} = A_i((A^{r-1})_j)^T$ where $(A^{r-1})_j = A_j A^{r-2}$ resulting in

$$(A^r)_{ij} = A_i(A_j A^{r-2})^T = A_i((A^{r-2})^T(A_j)^T) = A_i(A^{r-2})(A_j)^T.$$

From here it is possible to dissect the operations $A_i(A^{r-2})(A_j)^T$, and determine the exact route of each path of length r from vertex i to vertex j .

We have devised a way to determine all r -step paths between any vertex i and vertex j . In determining the set of possible vertices for each step in the path, it is first necessary to dissect $(A^r)_{ij}$ as follows using the properties of adjacency matrices

$$\begin{aligned} (A^r)_{ij} &= A_i(A^{r-2})(A_j)^T \\ &= A_i(A^{r-1})_j. \end{aligned}$$

In determining all paths between vertex i and j we will construct a series of sets which will be at a certain distance, or level, from vertex i . The first set will be $S(0) = \{i\}$

which represents the initial vertex in the path. This set is said to be at level zero, since it is zero steps from the first vertex.

Next, we are trying to determine sets which contain vertices one level from the initial vertex. The vertices one level from the initial vertex must also be $r - 1$ levels from the terminal vertex to ensure they will reach the terminal vertex in r steps as expected. We determine the set of level one vertices by comparing the vectors A_i , the i^{th} row in the matrix A , and $(A^{r-1})_j$, the j^{th} row in the matrix A^{r-1} . The elements which are non-zero in both vectors are put into a new set $S_{v_i}(1)$. The elements in this new set $S_{v_i}(1)$ are guaranteed to be one level from vertex i since they are non-zero in the vector A_i and $r - 1$ steps from vertex j since they are non-zero in the vector $(A^{r-1})_j$.

The sets for each level are constructed as follows

$$S_{v_1, v_2, \dots, v_L}(L),$$

where L represents the level from the initial vertex and $\{v_1, v_2, \dots, v_L\}$ is the sequence of vertices required to reach the set. The last element in the sequence, v_L , is an element from the set for a previous level. The set $S_{v_1, v_2, \dots, v_L}(L)$ is determined by comparing the vectors A_{v_L} and $(A^{r-L})_j$ and choosing the components which are non-zero in both vectors.

The algorithm for this procedure is as follows where i is the initial vertex and j is the terminating vertex:

1. Determine r such that $(A^r)_{i,j} \neq 0$
2. Set $S(0) = \{i\}$
3. For $1 \leq L \leq r - 1$, set $S_{v_0, v_2, \dots, v_{L-1}}(L)$ to the elements which are non-zero in both $A_{v_{L-1}}$ and $(A^{r-L})_j \quad \forall v \in S_{v_0, v_2, \dots, v_{L-2}}(L - 1)$

We will now demonstrate this method using the graph shown in Figure 4-10 where we want to determine all paths between vertex one and vertex eight. The adjacency matrix, A , and its first four powers are as follows

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

$$A^2 = \begin{bmatrix} 2 & 0 & 0 & 2 & 1 & 0 & 0 & 0 \\ 0 & 2 & 2 & 0 & 0 & 1 & 1 & 0 \\ 0 & 2 & 3 & 0 & 0 & 1 & 2 & 0 \\ 2 & 0 & 0 & 4 & 2 & 0 & 10 & 2 \\ 1 & 0 & 0 & 2 & 2 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 2 & 2 & 0 \\ 0 & 1 & 2 & 0 & 0 & 2 & 3 & 0 \\ 0 & 0 & 0 & 2 & 1 & 0 & 0 & 2 \end{bmatrix}$$

$$A^3 = \begin{bmatrix} 0 & 4 & 5 & 0 & 0 & 2 & 3 & 0 \\ 4 & 0 & 0 & 6 & 3 & 0 & 0 & 2 \\ 5 & 0 & 0 & 8 & 5 & 0 & 0 & 3 \\ 0 & 6 & 8 & 0 & 0 & 6 & 8 & 0 \\ 0 & 3 & 5 & 0 & 0 & 3 & 5 & 0 \\ 2 & 0 & 0 & 6 & 3 & 0 & 0 & 4 \\ 3 & 0 & 0 & 8 & 5 & 0 & 0 & 5 \\ 0 & 2 & 3 & 0 & 0 & 4 & 5 & 0 \end{bmatrix}$$

$$A^4 = \begin{bmatrix} 9 & 0 & 0 & 14 & 8 & 0 & 0 & 5 \\ 0 & 10 & 13 & 0 & 0 & 8 & 11 & 0 \\ 0 & 13 & 18 & 0 & 0 & 11 & 16 & 0 \\ 14 & 0 & 0 & 28 & 16 & 0 & 0 & 14 \\ 8 & 0 & 0 & 16 & 10 & 0 & 0 & 8 \\ 0 & 8 & 11 & 0 & 0 & 10 & 13 & 0 \\ 0 & 11 & 16 & 0 & 0 & 13 & 18 & 0 \\ 5 & 0 & 0 & 14 & 8 & 0 & 0 & 9 \end{bmatrix}$$

The first r for which the element $(A^r)_{ij} \neq 0$ occurs when $r = 4$, and the element

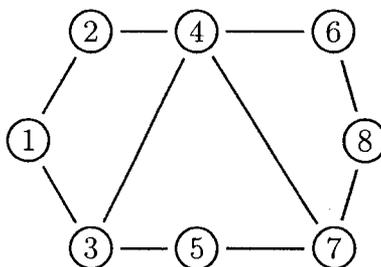


Figure 4-10. A simple graph with eight vertices

$(A^r)_{18} = 5$ literally means there are five, four-step paths between vertex one and eight. To determine all paths between vertex one and vertex eight, we will use the algorithm described previously.

We already have $r = 4$, so our next step is to set $S(0) = \{1\}$. We then set $L = 1$, meaning we are going to determine all the sets of vertices which are one level from the initial vertex. Since we only have one element in the set where $L = 0$, we will only have one set of level one vertices. We determine $S_1(1)$ by comparing $A_1 = [0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0]$ and $(A^3)_8 = [0 \ 2 \ 3 \ 0 \ 0 \ 4 \ 5 \ 0]$ and choose the elements which are non-zero in both vectors. The elements which are non-zero in both vectors are the second and third entries. Remember, the second entry corresponds to the second vertex and the third entry corresponds to the third vertex. Thus, we set $S_1(1) = \{2, 3\}$. These two vertices are said to be in the first level of the path since they have $L = 1$.

We now increment L and are looking for the sets $S_{1,2}(2)$ and $S_{1,3}(2)$ since $\{2, 3\} \in S_1(1)$. Both of these sets will contain vertices which are two levels from the initial vertex. The difference between the sets is that the first set has vertex two as its level one vertex, and the second set will have vertex 3 as its level two vertex. We will first determine the set $S_{1,2}(2)$ by comparing A_2 and $(A^2)_8$. These vectors can be seen

Level	Set	First Vector	Second Vector	Non – Zero Elements
0	$S(0)$			{1}
1	$S_1(1)$	$A_1 = [0\ 1\ 1\ 0\ 0\ 0\ 0\ 0]$	$(A^3)_8 = [0\ 2\ 3\ 0\ 0\ 4\ 5\ 0]$	{2, 3}
2	$S_{1,2}(2)$	$A_2 = [1\ 0\ 0\ 1\ 0\ 0\ 0\ 0]$	$(A^2)_8 = [0\ 0\ 0\ 2\ 1\ 0\ 0\ 2]$	{4}
2	$S_{1,3}(2)$	$A_3 = [1\ 0\ 0\ 1\ 1\ 0\ 0\ 0]$	$(A^2)_8 = [0\ 0\ 0\ 2\ 1\ 0\ 0\ 2]$	{4, 5}
3	$S_{1,2,4}(3)$	$A_4 = [0\ 1\ 1\ 0\ 0\ 1\ 1\ 0]$	$A_8 = [0\ 0\ 0\ 0\ 0\ 1\ 1\ 0]$	{6, 7}
3	$S_{1,3,4}(3)$	$A_4 = [0\ 0\ 1\ 0\ 0\ 0\ 1\ 0]$	$A_8 = [0\ 0\ 0\ 0\ 0\ 1\ 1\ 0]$	{6, 7}
3	$S_{1,3,5}(3)$	$A_5 = [0\ 0\ 1\ 0\ 0\ 0\ 1\ 0]$	$A_8 = [0\ 0\ 0\ 0\ 0\ 1\ 1\ 0]$	{7}

Figure 4-11. Calculations for determining all paths between vertex one and vertex six as shown in Figure 4-10.

in Figure 4-11, and the non-zero element in both is the fourth entry. Thus, we set $S_{1,2}(2) = \{4\}$. We then determine the set $S_{1,3}(2) = \{4, 5\}$ and continue this process until $L = r - 1$. The result of this procedure can be seen in Figure 4-11.

The final step is to determine the exact paths. We have found the sets up to the third level and we know the fourth level is the terminal vertex. As such, we can use the information about the third level sets and obtain our paths. From the set $S_{1,2,4}(3) = \{6, 7\}$, we now have two paths from vertex one to vertex eight which are

$$\{1, 2, 4, 6, 8\} \text{ and } \{1, 2, 4, 7, 8\}.$$

From the next set $S_{1,3,4}(3) = \{6, 7\}$, we obtain two more paths which are

$$\{1, 3, 4, 6, 8\} \text{ and } \{1, 3, 4, 7, 8\}.$$

We obtain the fifth and final path from the last set $S_{1,3,5}(3) = \{7\}$, which is

$$\{1, 3, 5, 7, 8\}.$$

At this point it is important to remember that the goal in representing the cupolets in a graph in such a way that the vertices represent cupolets and the edges represent a shared bin between two vertices, is to determine the controlled transition with the smallest length from any cupolet to another. We have described above a method which allows one to determine the paths between any two vertices. These paths corresponds to the controlled transitions between any two cupolets and can be written as

$$T(1, 8) = \{1, 2, 4, 6, 8\}$$

$$T(1, 8) = \{1, 2, 4, 7, 8\}$$

$$T(1, 8) = \{1, 3, 4, 6, 8\}$$

$$T(1, 8) = \{1, 3, 4, 7, 8\}$$

$$T(1, 8) = \{1, 3, 5, 7, 8\}.$$

The last step in this procedure is to determine the length of each controlled transition as described in 4.3 where the length corresponds to the number of times around the attractor necessary to transition between the two cupolets. At this point the controlled transition which is the shortest length is chosen and the entire process is complete.

CHAPTER 5

RESULTS AND CONCLUSION

5.1 Results

We have developed a control scheme which results in a controlled transition between any two chaotic unstable periodic orbit-lets. Several experiments have been conducted to test this method and the results confirm that controlled transitions between cupolets exist and can be much shorter in length than blind transitions. These experiments were conducted using a set of 100 cupolets from the entire set of over 8,800. We found that it was possible to transition from one unstable periodic orbit to any other in the set by applying a small number of controls at smart locations throughout the system.

The first result we would like to present is the transition between the cupolet with the control sequence '0101011' and cupolet with the control sequence '001111.' The bin, lobe, and control data for cupolet '0101011' can be seen in Table 5.1 and the same information corresponding to cupolet '001111' can be seen in Table 5.2. As you can see, the two cupolets do not pass through any of the same lobes in the same bins. As such, the cupolets are not switchable and the length of the transition between the two cupolets will be greater than zero.

We began our experiment using the original blind transition method. Using this method we departed cupolet '0101011' on lobe 1 in bin 1176 and began with the orig-

Lobe	Bin	Control
1	1176	0
1	1824	1
0	1145	0
0	1794	1
0	610	0
0	1049	1
0	1680	1
0	1176	0
0	1824	1
1	1145	0
1	1794	1
1	610	0
1	1049	1
1	1680	1

Table 5.1. Bin, lobe, and control information corresponding to cupolet '0101011.'

Lobe	Bin	Control
1	1559	0
1	1686	0
1	1143	1
1	1790	1
1	520	1
1	959	1
1	1559	0

Table 5.2. Bin, lobe, and control information corresponding to cupolet '001111.'

inal permutation of the second control sequence '00111.' The length of this transition is $\mathcal{L}_{Chaotic}(1, 1176, 001111) = 17$. The next transition scheme we imposed was the blind transition. For this method we begin from each of the 14 different bins around the first cupolet and implemented each of the 6 permutations of the second control sequence in each bin. The result was a set of $14 \times 6 = 84$ blind transitions whose lengths varied from one to twenty, with an average length of 10.238. It is important to note that the original blind transition is included in the set of blind transitions.

The next method we used was the controlled transition. To use this method we took a subset of one hundred cupolets from the original set of over 8,800 cupolets. Again we were transitioning from the initial cupolet '0101011' to the terminal cupolet '001111.' We found that there existed a cupolet, '0001011011' which allowed for a controlled transition of length one. To do this, we departed the initial cupolet on lobe 1 in bin 1145, as can be seen in Table 5.1, and switched to the intermediate cupolet '0001011011,' which also passes through this bin as can be seen in Table 5.3. At this point we rode the intermediate cupolet until it reached the next bin, 1790 on lobe 1, which intersects with the terminal cupolet in the transition, as seen in Table 5.2. Thus, with our small subset containing only 100 cupolets we were able to find a controlled transition with $\mathcal{L}_{Controlled}(0101011, 001111) = 1$.

Since the different methods of transitioning between cupolets measure the length of the transition by counting the number of loops around an attractor, we were able to compare the methods using a histogram as shown in Figure 5-1. The lengths of the transitions range from one to twenty. The original controlled transition fell in the

Lobe	Bin	Control
1	1790	0
1	499	0
1	887	0
1	1455	1
1	1868	0
0	1043	1
0	1676	1
0	1212	0
0	1854	1
1	1145	1
1	1790	0

Table 5.3. Bin, lobe, and control information corresponding to cupolet 0001011011.'

82.10 percentile of the lengths of all the transitions and the controlled transition fell in the 0 percentile.

In this example where we are transitioning between cupolet '0101011' and '001111,' we found that using the blind transition it is possible to get a transition whose length is the same as the controlled transition. Unfortunately, there is no way of predicting which combination of location and control sequence will result in this transition. The only way to determine if there is a blind transition of comparable length to the controlled transition is to test each of the 84 blind transitions to determine which is the shortest. If instead we use on the controlled transition, we know the exact trajectory of the transition and how long it will be.

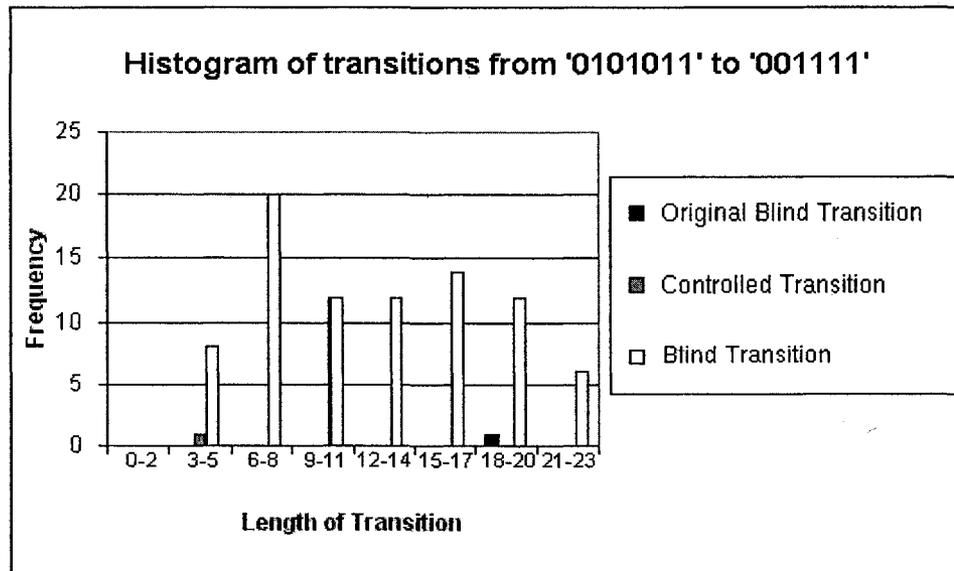


Figure 5-1. Comparison of different transitioning techniques between cupolet 0101011 and cupolet 001111.

We conducted several other examples, the first of which is the transition between cupolet '0010110101' and '01.' Using the different transitioning techniques between these cupolets we found that the controlled transition was in fact the shortest with a length of one. The distribution of the transitions can be seen in Figure 5-2. For this example, there were twenty blind transitions whose lengths range from 3 to 21 with an average length of 12.1. The original blind transition fell in the 10.0 percentile of the lengths of all the transitions and the controlled transition was in the 0 percentile. This example again shows that it is shorter to use the controlled transition.

Another example is the transition between cupolet '0000001111' and '00001.' This controlled transition had a length in the highest percentile of all the simulations we ran. There were 100 blind transitions ranging from a length of 3 to 42 with an average length of 23.07. In this case the controlled transition had a length of 11 which can be

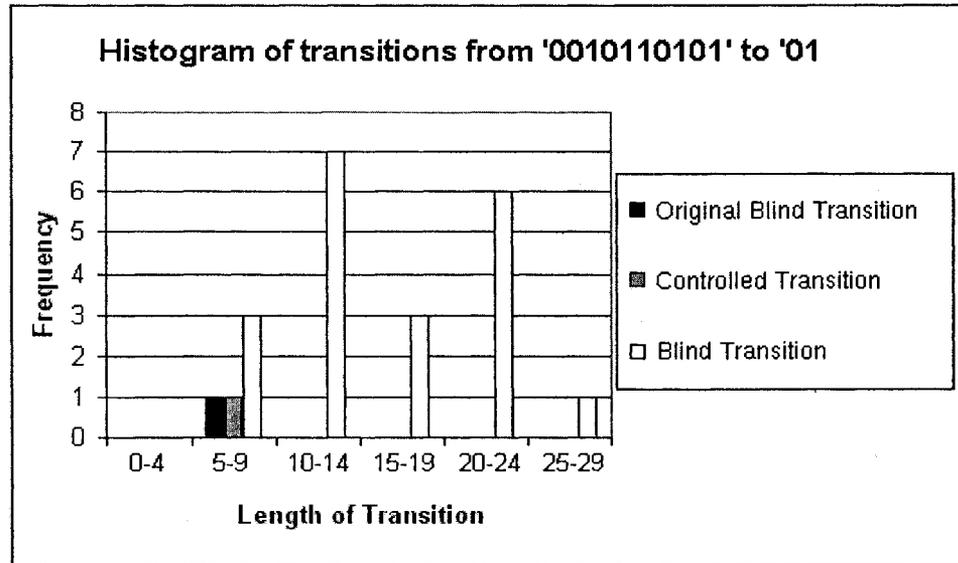


Figure 5-2. Comparison of different transitioning techniques between cupolet '0010110101' and cupolet '01'.

seen in Figure 5-3. Although this controlled transition was not the shortest transition we found, it is important to remember a few key points.

First, although there are a hand-full of blind transitions which are shorter than the controlled transition, there is no way to predict ahead of time which transitions will be the shortest. With the controlled transition, we know the exact trajectory of the transition and how long it will be without running any simulations. The only way to find a short blind transition is by testing every possible initial location and permutation of the second control sequence and then choosing the transition with the smallest length.

Second, the blind transitions are allowed to use the full dynamics of the controlled system. The controlled transitions are only restricted to a set of 100 cupolets. If we

were to add more cupolets to our set, we could potentially find a shorter controlled transition.

Third, we are only considering the controlled transitions with the smallest number of intermediate cupolets. This algorithm does not pick the optimal path in the graph, it instead chooses the path with the smallest number of intermediate cupolets. This is due to the fact that we only looked at r -step paths, where r is the smallest number such that $(A^r)_{ij} \neq 0$. Thus, we only found transitions with $r-1$ intermediate cupolets. Since it may be possible for a transition with more intermediate cupolets to have a shorter length, we need to check the higher values of r .

To do this, we could set an upper bound, Q , to be the length of the shortest controlled transition found for the smallest r such that $(A^r)_{ij} \neq 0$. We assume that each intermediate cupolet contributes at least a length of one to the total length of the controlled transition. Therefore, we only need to search higher values of r , each time setting Q to be the length of the shortest transition, while $r \leq Q$. If we allow r to be greater than Q , the length of the new transitions will be greater than or equal to the length of the shortest transition where the length is Q .

The final example we would like to present is between cupolet '0010110011' and '000000011.' There are 108 blind transitions whose lengths range from 64 to 108 with an average length of 79.52. As you can see in Figure 5-4, the controlled transition required only one intermediate cupolet and the length of the transition is one. This shows how powerful controlled transitions can be in systems where blind transitions can be very long.

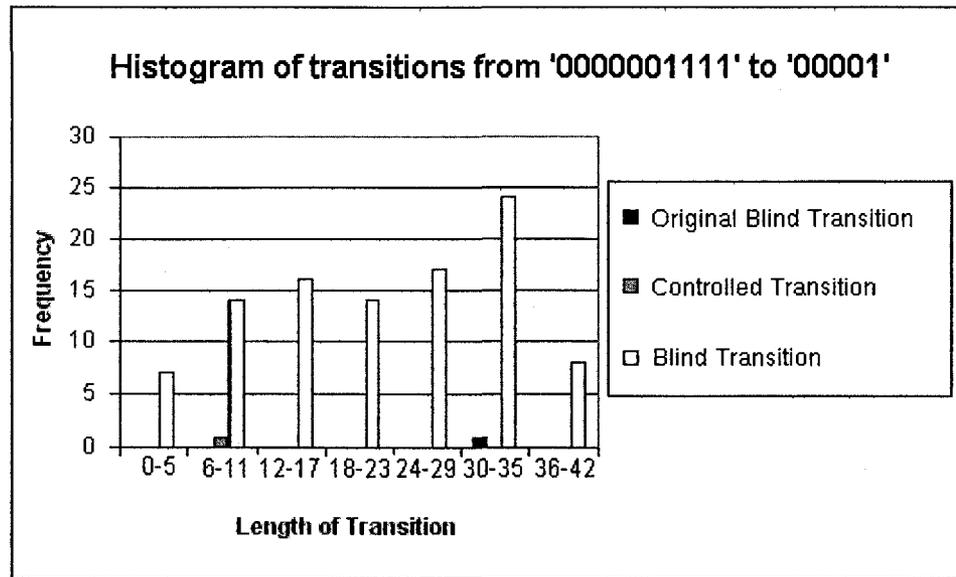


Figure 5-3. Comparison of different transitioning techniques between cupolet '0000001111' and cupolet '00001'.

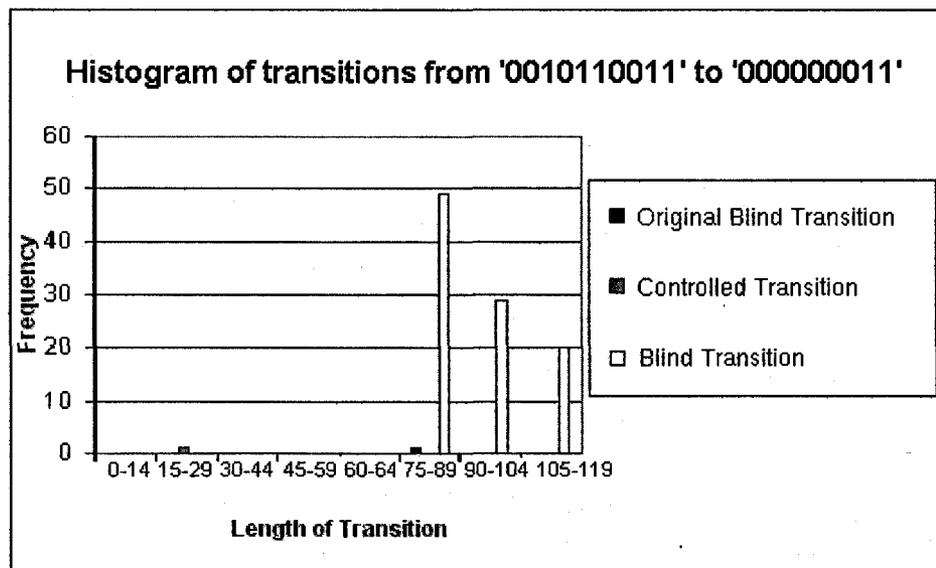


Figure 5-4. Comparison of different transitioning techniques between cupolet '0010110011' and cupolet '000000011'.

5.2 Conclusion

We believe controlled transitions can be applied to several areas where it is desirable to have short transitions between orbits. One such area of interest is guiding space shuttles, satellites, meteors, and other objects through space. Objects in space are attracted to larger bodies, such as planets and stars, and orbit around them. If we could match the orbit of a space shuttle in space to a cupolet, it is then possible to apply a low-energy control sequence to the shuttle, switch between intermediate cupolets, and shift to an orbit modeled by another cupolet. This would be useful if the initial orbit of the space shuttle was disastrous, for example an orbit which traveled too close to the sun. If the shuttle did not have enough fuel to move to a more safe orbit, we could instruct the passengers of the shuttle to apply a sequence of controls to the shuttle at precisely the right moment. In this way, the passengers could conserve their fuel until needed and continue their mission safely on the desired periodic orbit.

It is unclear at this point if the set of dense unstable periodic orbits around the attractors of the double scroll oscillator would actually be a good model for orbits which occur naturally. An interesting study would be to compare orbits which actually exist in outer space and determine if cupolets would be a good match for those orbits. If the cupolets were not a good match for those orbits, it is possible to repeat the process of creating cupolets cupolets using different chaotic system.

In summary, cupolets have once again proven to be a valuable tool in exploring chaotic systems and appear to be promising for future applications.

BIBLIOGRAPHY

- [1] Frank M. Carrano and Janet J. Prichard. *Data Abstraction and Problem Solving*. Addison Wesley, Boston, MA, USA, 2002.
- [2] Kevin M. Cuomo and Alan V. Oppenheim. Circuit implementation of synchronized chaos with applications to communications. *The American Physical Society*, July 1993.
- [3] Lawrence Fritts. *University of Iowa Electronic Music Studios: Music instrument Samples*. <http://theremin.music.uiowa.edu/>, 1997.
- [4] Scott Hayes, Celso Grebogi, and Edward Ott. Communicating with chaos. *The American Physical Society*, December 1992.
- [5] Bernard Kolman and David R. Hill. *Introductory Linear Algebra With Application*. Prentice Hall, Upper Saddle River, NJ, USA, 2001.
- [6] Alan V. Oppenheim and Alan S. Willsky with Ian T. Young. *Signals and Systems*. Prentice-Hall, Englewood Cliffs, NJ, USA, 1983.
- [7] Edward Ott, Celso Grebogi, and James A. Yorke. Controlling chaos. *The American Physical Society*, December 1989.
- [8] Troy Shinbrot, William Ditto, and et. al. Using the sensitive dependence of chaos (the butterfly effect) to direct trajectories in an experimental chaotic system. *The American Physical Society*, November 1991.
- [9] K. M. Short and R. A. Garcia. Signal analysis using the complex spectral phase evolution (cspe) method. *AES 20th Convention*, May 2006.
- [10] Steven H. Strogatz. *Nonlinear Dynamics and Chaos*. Addison-Wesley, Reading, MA, USA, 1994.