Controlled transitions between cupolets of chaotic systems

Matthew A. Morena,^{a)} Kevin M. Short, and Erica E. Cooke

Integrated Applied Mathematics Program, University of New Hampshire, Durham, New Hampshire 03824, USA

(Received 5 November 2013; accepted 7 January 2014; published online 23 January 2014)

We present an efficient control scheme that stabilizes the unstable periodic orbits of a chaotic system. The resulting orbits are known as cupolets and collectively provide an important skeleton for the dynamical system. Cupolets exhibit the interesting property that a given sequence of controls will uniquely identify a cupolet, regardless of the system's initial state. This makes it possible to transition between cupolets, and thus unstable periodic orbits, simply by switching control sequences. We demonstrate that although these transitions require minimal controls, they may also involve significant chaotic transients unless carefully controlled. As a result, we present an effective technique that relies on Dijkstra's shortest path algorithm from algebraic graph theory to minimize the transients and also to induce certainty into the control of nonlinear systems, effectively providing an efficient algorithm for the steering and targeting of chaotic systems. © 2014 AIP Publishing LLC. [<http://dx.doi.org/10.1063/1.4862668>]

Efficiently directing a dynamical system to a desired state is a goal of many engineering applications. This process is known as “targeting” and is often achievable using small nonlinear controls. For a chaotic system, the goal of many control methods is either to stabilize the system onto one of its many unstable periodic orbits (UPOs) or to induce the system to transition between its UPOs until a targeted orbit is attained. The set of UPOs plays a significant role in determining the dynamics of a chaotic system and is said to form the skeleton of an associated attractor. In this paper, we combine a particularly effective chaos control method with algebraic graph theory and present a new technique allowing for the efficient transitioning between periodic orbits of chaotic systems. This allows one to navigate efficiently around a chaotic attractor and in physical applications conserve energy and power.

LETS).^{16–18} This technique allows for the creation of thousands of cupolets, which are essentially highly accurate approximations of the UPOs of chaotic systems. Cupolets represent a subset of the UPOs but are distinguished because they are stabilized by the control method independently of their initial condition. This leads to a unique pairing between a cupolet and a given sequence of controls. We show that the most straightforward method for transitioning between pairs of cupolets may be performed inexpensively, but this may involve considerable chaotic transients during the transition periods.

To improve this result, we present a graph-theoretic approach based on Dijkstra's shortest path algorithm that minimizes the transients.¹⁹ Effectively, this shows that the problem of targeting UPOs can be replaced by that of finding optimal paths within a graph. One advantage of our method is that path-searching is already a mature topic in computer science and can be performed very efficiently.²⁰ Our approach also induces certainty into the control of chaotic systems as it directs the transients along segments of intermediary cupolets whose orbits may be predicted in advance.

Cupolets are controlled and stabilized periodic orbits of a chaotic system that would normally be unstable without the presence of the control mechanism. These orbits originated from a secure communication application but have been found to be useful in a variety of other applications as well.^{8,9,18} Very little information is required to generate cupolets (typically 16-bit control sequence are used) and yet they are characterized by the richness of their frequency spectra and how accessible their dynamical behaviors are via small controls. Only small perturbations are required to generate cupolets, which in turn suggests that they are shadowing true periodic orbits and theorems have been developed that establish conditions under which this holds.^{5–7,17}

I. INTRODUCTION

In recent years, much attention has been focused on controlling chaotic systems via small perturbations and in particular on two types of chaos control. One type seeks to stabilize the UPOs that are typically embedded densely within an attractor.^{1–4} This abundance of UPOs is utilized in a variety of theoretical and practical applications and provides a useful way of investigating a chaotic system since many of its properties can be expressed in terms of these orbits.^{5–9} Another approach is to rapidly direct trajectories to desired locations in phase space, a process known as *targeting*.^{10–12} Several algorithms have since been developed that combine both types of chaos control techniques in order to switch a chaotic system from one stabilized periodic orbit to another.^{13–15}

In this paper, we describe an application of one control method that efficiently stabilizes the UPOs of chaotic systems onto *cupolets* (*C*haotic, *U*nstable, *P*eriodic, *O*rbit-

II. GENERATING CUPOLETS

Our method for producing cupolets is derived from a chaotic communication scheme developed by Hayes,

^{a)}Electronic mail: matthew.morena@wildcats.unh.edu

FIG. 1. Double scroll oscillator showing the control surfaces.

Grebogi, and Ott (HGO).^{2,3} The HGO method applies small perturbations along a control surface in order to steer trajectories of a chaotic system around an attractor. This method was further adapted by Parker and Short^{16,21} who demonstrated that repeatedly applying control information would drive a chaotic system onto a unique periodic orbit. These controls can be made arbitrarily small and thus do not grossly alter the topological structure of the orbits on the chaotic attractor.

HGO controlled the double scroll system, also known as Chua's system,²² to follow a prescribed symbolic sequence. The differential equations describing the double scroll system are given by

$$\begin{aligned} C_1 \dot{v}_{C1} &= G(v_{C2} - v_{C1}) - g(v_{C1}), \\ C_2 \dot{v}_{C2} &= G(v_{C1} - v_{C2}) + i_L, \\ L \dot{i}_L &= -v_{C2}, \end{aligned}$$

where

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

with parameter values $C_1 = \frac{1}{9}$, $C_2 = 1$, $L = \frac{1}{7}$, $G = 0.7$, $m_0 = -0.5$, $m_1 = -0.8$, and $B_p = 1$. The attractor of this system contains two lobes that each surround an unstable fixed point. Figure 1 shows a typical trajectory that has outlined the attractor of this system as well as the positions of two control planes. The control planes are assigned the values 0 and 1 so that a binary symbolic sequence may be recorded whenever a trajectory passes through a control plane.

To generate a desired symbolic sequence, a Poincaré surface of section is first defined on each lobe by

$$i_L = \pm GF \quad \text{and} \quad |v_{C1}| \leq F,$$

where $F = \frac{B_p(m_0 - m_1)}{(G + m_0)}$. Both half-planes are two-dimensional, but because the attractor is nearly two-dimensional close to this surface, the intersection between the Poincaré surface and the

FIG. 2. Top view of the intersection of the Poincaré surface with the 1-lobe of the double scroll attractor.

attractor is approximately one-dimensional. The asterisks in Figure 2 mark where the trajectory in Figure 1 pierces a Poincaré half-plane. This set of points may be well-approximated by a line extending from each fixed point, fitted with the least squares method. Figure 2 shows a top view of this line.

Constructing a suitable Poincaré surface is a strategy commonly used when searching for UPOs of chaotic systems, whereby trajectories can be encoded and controlled via symbolic dynamics. For instance, if a system evolves in n dimensions, but is locally $(n-1)$ dimensional, then one may obtain an $(n-2)$ dimensional control plane by fixing a Poincaré hyperplane within the lower-dimensional region. This technique can be adapted for general unimodal maps and is particularly effective when $n=3$, such as in the Lorenz and Rössler systems.^{3,4} The effect is to simplify the dynamics from the ambient phase space to a more manageable Poincaré map. More complicated control surfaces and symbolic dynamics have also been used to control chaotic systems.^{23–25}

We thus partition the intersection between the double scroll attractor and our Poincaré surface into M equally spaced bins and record the distance from the center of each bin to the center of the corresponding lobe. The bin centers are then used as initial conditions from which the chaotic system is simulated without control. After N loops around the attractor, each resulting trajectory produces a symbolic sequence, which is the binary sequence of lobes the trajectory visits. These sequences are in turn represented by a coding function, $r_N(x)$, which maps an initial condition, x , to a binary representation of its future symbolic sequence. That is, if x results in the binary sequence $b_1 b_2 b_3 \dots$ (where b_i is either 0 or 1, representing a visit to lobe 0 or lobe 1, respectively), its sequence is mapped to the binary decimal $0.b_1 b_2 b_3 \dots$ given by

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

In this manner, both $r_N(x)$ and the future symbolic sequence of every bin center on each Poincaré section are known for

FIG. 3. Coding function $r(x)$.

N -many loops around the attractor. Figure 3 shows a plot of this function.

The next step of the HGO method is to run the simulation until a trajectory crosses the Poincaré surface, say at x_0 . The goal is to steer the trajectory through x_0 so that in the future, the trajectory traces out the desired symbolic sequence. Thus, if the last digit of the symbolic sequence of x_0 differs from the first digit of the desired sequence, we search $r_N(x)$ for the bin center nearest to x_0 having the same symbolic sequence as x_0 except for the last digit. More precisely, we locate an x_1 on the control plane as x_0 such that $|r_N(x_0) - r_N(x_1)| = 2^{-N}$, and then perturb the trajectory to start at the new point, x_1 . These perturbations are known as *macrocontrols* and for sufficiently large N are guaranteed to be small.

If the last digit of the symbolic sequence of x_0 is the same as the first digit of the desired sequence, we let the trajectory pass through the control plane (essentially) unperturbed and evolve until its next intersection with the Poincaré surface. In a slight modification of the original HGO method, in each unperturbed situation, *microcontrols* reset the trajectory to the center of the bin where it has intersected the control plane. This minimizes the accumulation of round-off error and simplifies the identification of the bin boundaries. This also suggests that some cupolets may intersect, which would occur exactly in the center of a bin. As we show in Sec. III, this allows for an instantaneous transition between intersecting cupolets.

More generally, the process of controlling a chaotic system occurs whenever a trajectory crosses one of the control planes. The option to perturb or not perturb the trajectory is described by a unique, binary control sequence with 1 meaning apply a macrocontrol and 0 meaning apply only a microcontrol. It was shown by Parker and Short^{16,21} that for nearly all finite control sequences, if the controls are repeatedly applied to the system then after an initial transient, the trajectory will close up on itself and form a unique periodic orbit, *independent of its initial condition*. Any transient is the result of the trajectory evolving while the chaotic system sifts through all possible states until it reaches one where the

FIG. 4. The period five cupolet C00001.

behavior of an UPO falls into synchrony with the control sequence, thus stabilizing the orbit. These approximate chaotic unstable periodic orbits are called *cupolets*.^{16–18} Figure 4 shows a period five cupolet. Cupolets have also been generated from other chaotic systems, such as the Lorenz and Rössler systems, and from general unimodal maps.

Note that due to the symmetry of the double scroll system about the origin, non self-symmetric cupolets exist in pairs. That is, if a cupolet's orbit is not distributed evenly about the origin, a “symmetric cousin” of this cupolet exists and the co-ordinates of these cupolets differ only by a reflection about the origin. Since pairs of these cupolets have both the same control sequence and flow time and since the labeling of the control planes is arbitrary, for each pair, we report only the cupolet whose orbit does not visit the 0-lobe more frequently than the 1-lobe. The cupolet shown in Figure 4 is an example of a non self-symmetric cupolet.

We implement this control technique in order to generate a finite but sufficiently large set of cupolets. For instance, over 8800 cupolets may be stabilized very efficiently by sending in 16-bit or shorter control sequences, and we typically partition each control plane into 2000 bins. For organizational purposes, we adopt a naming convention for the cupolets that incorporates the control sequences that generate them. For instance, the cupolet shown in Figure 4 is stabilized by repeatedly applying the controls “00001” and is hence named C00001.

The key points of this control method are that all of these different dynamical behaviors are easily accessible via small controls and that the cupolets may be stabilized regardless of the initial state of the system and yet are in a one-to-one correspondence with the control sequences. These properties make it possible to transition from one cupolet to another, as we discuss in the remainder of this paper.

III. CUPOLET TRANSITIONS

Until recently, the targeting of a chaotic system's UPOs relied on the ergodicity of the system. This guarantees that a free-running trajectory will eventually wander arbitrarily

close to any location in its phase space, whereupon the system may be controlled to keep its trajectory within some acceptable tolerance of a desired periodic orbit. It is not generally known *a priori* the wait time for a capture, nor the path a chaotic trajectory will take as it explores its phase space. Although one may calculate statistically an expected wait period, this can be very long for high-dimensional systems. To accelerate an acquisition, a variety of targeting techniques have been designed to efficiently steer a chaotic system along an optimal path to a desired location.^{10–15}

In contrast, all that is required for a cupolet to remain stabilized is the repeated application of its control sequence to the system. These control sequences are unique (up to a cyclic permutation) since they direct a chaotic system onto one specific cupolet. Applying different controls would therefore cause a cupolet to destabilize and revert to chaotic behavior. If another control sequence were to be periodically applied, then, after some intermediary transient, the chaotic trajectory would re-stabilize onto the periodic orbit of a second cupolet. We thus define the transient part of a cupolet transition to be the section of the trajectory from where it leaves an initial cupolet to the point where it stabilizes onto the target cupolet.

This is the idea behind cupolet transitions and we now address ways to make the transition more efficient.

A. Blind transitions

For instance, one may switch or *transition* from any initial cupolet A to some other target cupolet B as follows. When cupolet A has completed at least one period, the control sequence governing cupolet B may be sent into the system, regardless of where cupolet A is found along its orbit. The new control sequence instructs the control scheme to perturb the initial cupolet away from its periodic orbit and acquire the orbit of cupolet B. As cupolet A destabilizes, the trajectory leaves the periodic orbit as seen in Figure 5(a) and begins evolving along an intermediate transient. The controls for cupolet B are then periodically repeated until the trajectory stabilizes onto cupolet B and completes one period. This is illustrated in Figures 5(b)–5(d), which captures the transition from cupolet C00 to cupolet C01.

This type of transition is known as a *blind transition* because once the trajectory is destabilized from an initial cupolet, it evolves unpredictably along a chaotic transient

FIG. 5. Blind transition: in (a), destabilizing from the orbit of cupolet C00 (solid line, blue squares) to the dashed green path of the transient; in (b), the cupolets C00 (period 1) and C01 (solid line, red triangles, period 4); and in (c) and (d), the blind transition from C00 to C01, including the green chaotic transient.

until its stabilization onto a targeted cupolet. While this technique is guaranteed to destabilize the initial cupolet and subsequently stabilize the second cupolet, it suffers from two limitations. First, when the trajectory is set free from an initial cupolet, a chaotic system's inherent nonlinearity makes predicting the length of the ensuing transient nearly impossible. Second, in some blind transitions, we have observed the transient stabilizing not onto the desired target cupolet but onto the target cupolet's symmetric cousin. Therefore, to ensure consistency when comparing a set of transitions, we consider only those transitions that successfully acquire the intended target cupolet and not its symmetric cousin. In Sec. IV, we demonstrate that blind transitions involve a wide range of transient lengths, many of which are significant.

B. Switchable cupolets

As stated in Sec. II, it is possible for two cupolets to intersect if they pass through the same bin on a control plane. If a transition is initiated in the bin where two cupolets intersect and an appropriate cyclic permutation of the second cupolet's control sequence is implemented, then the length of the resulting transient is zero: as one cupolet destabilizes, its trajectory immediately stabilizes onto the second cupolet. Consequently, two cupolets are said to be *switchable* if they pass through the same bin on a control plane, meaning that one may transition between switchable cupolets and avoid any intermediate transient. Transitioning between switchable cupolets is therefore a special instance of a blind transition, which would result if the correct order of control bits of the second cupolet is applied exactly at the bin where the cupolets intersect.

For switchable cupolets A and B, the strategy is to merge instantly off of cupolet A and onto cupolet B. Only one such permutation of controls will allow for such a zero-length transition because the bits of a cupolet's control sequence are order-specific to the particular bins through which that cupolet passes. If a wrong control is implemented, thereby interrupting the periodic application of a cupolet's control sequence, then cupolet A will be perturbed and will undergo transient chaos until it is redirected onto another periodic orbit. However, if the correct bit of the control sequence governing cupolet B is implemented beginning from a bin where the cupolets intersect, then the trajectory will simultaneously destabilize from cupolet A and stabilize onto cupolet B.

As an example, consider the bin, lobe, and control information for the cupolets A and B given in Table I. These cupolets are defined by the control sequences "001" and "001011," respectively, and are switchable since they both pass through bin 133 on lobe 1. Notice that for cupolet A, the control bit applied to bin 133 on lobe 1 is a "1" and for cupolet B, the control is a "0". For cupolet B, this corresponds to the underlined bit in its control sequence: "001011" and the desired permutation of the control bits is therefore "011001." This is the correct permutation because this sequence begins with the underlined control in the sequence and then cycles through the rest of the control sequence. When this particular control sequence is implemented in the bin shared by these cupolets, the transition from cupolet A onto cupolet B happens instantly.

TABLE I. Bin, lobe, and control information corresponding to four cupolets with control sequences (a) "001," (b) "001011," (c) "011111," and (d) "01."

(a) Cupolet A		
Lobe	Bin	Control
1	1522	0
0	1943	0
1	133	1
1	832	0
0	245	0
0	45	1
(b) Cupolet B		
0	1684	0
1	34	0
1	1023	1
1	133	0
1	1987	1
1	579	1
(c) Cupolet C		
1	1734	0
0	3	1
0	1798	1
1	34	1
1	749	1
1	1462	1
(d) Cupolet D		
0	3	0
1	1552	1
0	513	0
0	1005	1
1	142	0
0	177	1

Switching between intersecting cupolets thus requires minimal control information in order to avoid the intermediate transients, which are unavoidable when transitioning between non-intersecting cupolets. In the next sections, we present an algorithm that minimizes these transients and allows for the efficient switching between cupolets, which do not intersect.

C. Controlled transitions

The transients that ensue when transitioning between switchable cupolets are always of zero-length since these transitions require only the appropriate substitution of one control sequence for another. Using this idea, it is possible to transition between initial and target cupolets by making known intermediate jumps between the orbits of intersecting cupolets, thus avoiding the unpredictability that characterizes blind transitions. A transition is instead directed along known segments of other cupolets, thereby making predictable both the path of the transition and its length. This is accomplished by starting at an initial cupolet and making a finite series of transitions between a sequence of switchable cupolets until a target cupolet is acquired. This type of

transition is known as a *controlled transition* because, unlike a blind transition, we can choose the trajectory that will be taken when switching between the intermediate cupolets. There is thus no risk of a controlled transition stabilizing onto the symmetric cousin of a target cupolet. We propose a way of measuring the lengths of both blind and controlled transitions in Sec. III D.

Controlled transitions are represented by a transition sequence $\{C_1, C_2, \dots, C_N\}$, where C_1 and C_N are the desired initial and target cupolets that may be connected via controlled transitions by the intermediary cupolets, $C_2, \dots, C_{N-1}$, which are pairwise switchable. That is, cupolet C_j is switchable with both C_{j-1} and C_{j+1} , where $j \in \{2, \dots, N-1\}$. In a blind transition, no intermediary cupolets are explicitly utilized and a transition sequence consists of only the initial and target cupolets, $\{C_1, C_2\}$.

A quick scan of the bin and lobe information for cupolets A and C in Table I reveals that they are not switchable. However, since cupolet B intersects both cupolet A, at bin 133 on lobe 1, and cupolet C, at bin 34 on lobe 1, cupolet B is switchable with both cupolets and may thus be used as an intermediary cupolet to switch between cupolets A and C via the transition sequence, $\{A, B, C\}$. This transition is realized by first following cupolet A until it reaches bin 133 on its 1-lobe, whereupon applying the correct permutation of the control sequence for cupolet B, “011001,” enables the instantaneous transition onto cupolet B. The trajectory then rides cupolet B through bins 1987 and 579 on lobe 1, through bin 1684 on lobe 0, until it reaches bin 34 on lobe 1, exactly where cupolets B and C intersect. To stabilize immediately onto cupolet C, the controls, “011,” must then be implemented. This completes a controlled transition from cupolet A to cupolet C by way of cupolet B.

Notice that the entire controlled transition is not zero-length since the transient follows cupolet B through four of its bins (i.e., through four loops around the attractor). There may in fact be several different controlled transitions between pairs of non-intersecting cupolets and these may result in varying transition lengths. One may also transition between cupolets A and C via cupolet D. In this case, the transition sequence is $\{A, D, C\}$ with the transient riding cupolet D through five intersections with the control planes. In the following sections, we describe one method of determining all of the possible controlled transitions between any two cupolets so that the most efficient transition may be found.

D. Measuring cupolet transitions

It is first necessary to determine a method of measuring the transitions, so that they can be compared. The length of a transition between any two cupolets is defined as a map

$$\mathcal{L} : \mathcal{C} \rightarrow \mathbb{N} \cup \{\infty\},$$

where $\mathbb{N} = \{0, 1, 2, \dots\}$ and $\mathcal{C} = \{C_1, C_2, \dots, C_N\}$ is a transition sequence consisting of two or more cupolets with C_1 the initial cupolet and C_N the target cupolet. For the double scroll system, one way to determine $\mathcal{L}$ is by counting the

total number of loops around the attractor that the ensuing transient must make when switching from the initial to the target cupolet, including the final loop where the transient stabilizes onto the target cupolet. Since trajectories pass through a control plane once per loop, we can simply count the number of intermediate bins (equivalently, the number of intersections with control planes) necessary to complete the transition. In the event that the transient of a cupolet transition does not stabilize onto a target cupolet, as which might occur in a blind transition (where the symmetric cousin is instead acquired), then the transition is assigned infinite length.

For a controlled transition, each intermediate cupolet, $C_j \in \{C_2, \dots, C_{N-1}\}$, contributes to the transition's length by counting the loops the trajectory makes while riding C_j beginning with its transition from C_{j-1} and ending with its transition to C_{j+1} . Note that if the initial and target cupolets, C_1 and C_2 , are themselves switchable, then $\mathcal{L}(\{C_1, C_2\}) = 0$, since the transition is instantaneous and hence does not require any intermediate cupolets to be carried out.

As an example, the length of the controlled transition $\{A, B, C\}$ from Sec. III C is determined by counting the number of bins that are visited while riding a segment of the intermediary cupolet B. This gives $\mathcal{L}(\{A, B, C\}) = 4$. Similarly, $\mathcal{L}(\{A, D, C\}) = 5$. It is also possible to transition from cupolet C to cupolet A via cupolet B in which case $\mathcal{L}(\{C, B, A\}) = 2$. In general, the length of the transition between an initial and a target cupolet is not the same as the length of the reverse transition from the target to the initial cupolet. This is because the flow direction is in forward (i.e., simulated) time and is generally not reversible.

E. Modeling cupolets using graph theory

The controlled transition having the shortest length between two cupolets can be found using graph theory. To do this, a weighted digraph is constructed by representing the control bins as vertices and the cupolets as edges. Two bins are thus considered connected if there exists a cupolet whose forward evolution passes from one bin to the other. Each directed edge between two vertices (i.e., bins) also carries a weight that represents the number of intermediary bins the associated cupolet visits when connecting the two corresponding bins. An example of this is illustrated in Figure 6, where a bin space of five bins and six arbitrary cupolets A, B, C, D, E, and F are represented by a weighted multi-digraph of five vertices. However, in a weighted digraph, it is only possible to have one directed edge joining two vertices. As such, if two cupolets both connect the same bin pair, we take as the connecting edge the cupolet that passes through the least number of intermediate bins. The weighted digraph corresponding to the multi-digraph of Figure 6 is seen in Figure 7.

We do not include singleton loops in these graphs since they each represent a cupolet that connects the same bin to itself and thus do not provide any additional information about the connectedness of a graph. Furthermore, if there is a cupolet that connects some bin A to another bin B, then that same cupolet also connects bin B to bin A. Every directed

FIG. 6. An example weighted multi-digraph of arbitrary cupolets A, B, C, D, E, and F. The five vertices represent control bins and the directed edges represent cupolets. The number associated to a cupolet indicates how many bins that cupolet passes through when connecting one bin to another. Not all of the edges have been drawn due to space considerations.

edge thus has a corresponding edge in the opposite direction, although the weights of these edges are not necessarily the same because we traverse along a cupolet only in one direction. Also, there may not exist an edge between a set of vertices. This occurs if no cupolet has been found that passes through both of the associated bins.

Dijkstra's shortest path algorithm is a graph-theoretic method for determining the path with the smallest cumulative weight between any two vertices that belong to the same connected component of a weighted digraph.¹⁹ This could involve traversing through several intermediate vertices in order to join two given vertices. In terms of cupolet transitions, this corresponds to a sequence of switchable cupolets that connects a pair of control bins while collectively passing through the least

number of intermediary control bins. In our formulation, this represents the shortest path between a bin pair. To determine the shortest controlled transition between any initial and target cupolets, one would compare all possible bin pairs between the two cupolets and select a bin pair that may be connected by the least weighted sequence of intermediary cupolets. The shortest controlled transition between these cupolets would then begin and end from the bins of this bin pair.

In general, the number of pairs to be checked is $m*n$, where m is the number of bins through which the initial cupolet passes and n is the number of bins through which the target cupolet passes. We typically generate double scroll cupolets using a bin space of 4000 total bins and in a standard library of over 8800 cupolets many cupolets pass through

FIG. 7. Weighted digraph of the multi-digraph from Figure 6 for which we will determine the shortest path from bin 1 to every other vertex using Dijkstra's shortest path algorithm.

as many as 100 bins or more. As such, we would need to compare all bins of the initial cupolet with those of the target cupolet. Therefore, the number of bin pairs from which a controlled transition must be calculated can be as high as 10 000 for some cupolet pairs. To speed up calculations, we first determine the transition of shortest length between every possible bin pair and record these lengths along with the corresponding cupolets that connect the bins in a separate cupolet look-up table. Then, when searching for the shortest controlled transition between any two cupolets, one can compare the transition lengths of all of the cupolets' bin pairs in order to obtain the shortest transition.

To demonstrate Dijkstra's algorithm, we will use the weighted digraph graph shown in Figure 7. A preliminary step is first to represent the graph by a weighted adjacency matrix, M , where each element, M_{ij} , is the weight of the directed edge from vertex i to vertex j . The elements of M along the diagonal are zero since singleton loops are excluded while other elements in M are set to ∞ if there is no directed edge between the two vertices. The adjacency matrix, M , and associated cupolet look-up table, C , corresponding to the weighted digraph of Figure 7 are

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

and

$$C = \begin{bmatrix} 0 & \text{Cup B} & \text{Cup D} & \text{Cup E} & \text{Cup C} \\ \text{Cup F} & 0 & \infty & \text{Cup D} & \text{Cup A} \\ \text{Cup A} & \infty & 0 & \infty & \text{Cup E} \\ \text{Cup B} & \text{Cup C} & \infty & 0 & \infty \\ \text{Cup A} & \text{Cup B} & \text{Cup F} & \infty & 0 \end{bmatrix}.$$

The first step in the algorithm is to select a starting or current vertex, say vertex 1. Our goal will be to determine the shortest path from vertex 1 to every other vertex. The next step is to construct a vector for the path weights. Since our starting vertex is $v_c = \{1\}$, the first row of the weight matrix, $w = [0, 7, 1, 1, 16]$, is taken to be the initial weight vector. This vector gives the path weight from vertex 1 to all other vertices; e.g., the path from vertex 1 to vertex 2 has weight $w_2 = 7$. We then determine if this is indeed the shortest path from vertex 1 to vertex 2, or if a more efficient path can be found. To do this, we split the remaining vertices into two sets, V and U , those which have been visited and those which are unvisited. Since we have only visited one vertex, we set $V = \{1\}$ and $U = \{2, 3, 4, 5\}$. Dijkstra's algorithm then proceeds as follows:

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

The weight vector does not change during the first iteration of the algorithm. For the second iteration, the first step

moves the current vertex to the closest unvisited vertex, $v_c = \{3\}$. This gives $V = \{1, 3\}$ and $U = \{2, 4, 5\}$. We now check to see if the weight of the path between vertex 1 (the starting vertex) and every other unvisited vertex is shorter than a path which instead detours through vertex 3.

The first weight that is checked is $w_2 = 7$. We compare this weight with that of first passing through vertex 3, which is $w_3 + M_{3,2} = 1 + \infty = \infty$ because no edge exists between vertices 2 and 3. Therefore, element two in the weight vector remains a seven. Next, we examine w_4 , which is currently set to 1. Again, there is no edge between vertices 3 and 4 and the weight remains the same. Finally, we check w_5 which is currently 16. We compare $w_5 = 16$ with $w_3 + M_{3,5} = 1 + 13 = 14$ and observe that it is shorter to pass through vertex 3 in getting from vertex 1 to vertex 5. Having completed step four of the algorithm, we update the weight vector, giving $w = [0, 7, 1, 1, 14]$.

Dijkstra's shortest path algorithm is continued in a similar fashion and the final weight vector is $w = [0, 3, 1, 1, 8]$. This reveals the weights of the shortest paths from vertex 1 to all other vertices and may be confirmed by inspecting Figure 7. If we compare the original weight vector to the new weight vector, we see that entries w_2 and w_5 have changed. This is because it is shorter to get from vertex 1 to vertex 2 through vertex 4 and also to connect vertices 1 and 5 by passing first through vertex 4 and then through vertex 2. For the graph in Figure 7, where there are only five bins, Dijkstra's algorithm would need to be executed four more times, each time using a different initial vertex, so as to determine the least weighted path between every possible bin pair.

Dijkstra's algorithm may thus be used to determine the shortest controlled transition between any two cupolets, even if they are non-intersecting, provided that at least two of their bins are connectable by cupolets. The targeting problem for the UPOs of chaotic systems therefore reduces to searching for optimal paths through a graph. In its formulation presented here, Dijkstra's algorithm runs in $O(|V|^2)$, where $|V|$ is the number of vertices in the graph, but the algorithm may be optimized to run in $O(|V|\log|V| + |E|)$, where $|E|$ is the number of edges.²⁰

IV. RESULTS

We now provide four examples that demonstrate both blind and controlled transitions using the techniques presented in Sec. III. These examples are representative of the numerous cupolet transitions that we have simulated. For comparison purposes, the initial and target cupolets that we transition between are not switchable so as to guarantee that the lengths of their transitions are greater than zero.

As a preliminary step, we first obtain a subset of 100 randomly chosen cupolets from a larger library of over 8800 cupolets. Each pair of initial and target cupolets is selected from this subset. To execute the controlled transitions, we use the graph-theoretic approach described in Sec. III E to determine from among the cupolet subset the shortest controlled transition that connects a pair of initial and target cupolets. This way, the controlled transitions are obtained

only from intermediate cupolets belonging to this subset, which could be enlarged in order to obtain controlled transitions of potentially shorter lengths. This same cupolet subset is used in each of these four examples. We then implement a large number of blind transitions between the initial and target cupolets. This is accomplished by applying every permutation of the control bits that govern the target cupolet to each bin through which the initial cupolet passes. The result is a total of $nb*nc$ many blind transitions, where nb is the number of bins in the orbit of the initial cupolet and nc is the number of control bits required to generate the target cupolet.

In each example, we determine the lengths of the resulting transitions according to the method described in Sec. III D. We then plot the distribution and frequency of these lengths using a histogram in order to juxtapose the two

transition techniques. Only finite-length transitions are used in creating the histograms, which are all found in Figure 8.

Our first example is the transition from the initial cupolet C0101011 to the target cupolet C001111. The bin, lobe, and control information for these cupolets is listed in Table II. Since C0101011 has period 14, to perform the blind transitions each of the 6 cyclic permutations of “001111” is repeatedly applied to each of the 14 bins of C0101011. The result is a set of 84 blind transitions, thirty of which stabilize onto the symmetric cousin of the target cupolet and are therefore not of finite length. The lengths of the finite transitions range from 1 to 20, with an average length of 10.1111. In seeking the shortest controlled transition (using Dijkstra’s algorithm), we observe that the intermediate cupolet C0001011011 allows for a controlled transition of length one via the transition sequence, {C0101011, C0001011011, C001111}. According

FIG. 8. Histograms comparing the lengths of different transitioning techniques between various cupolets. The hollow bars represent the blind transitions while the solid bars are that shortest controlled transitions.

TABLE II. Bin, lobe, and control information for cupolets C0101011, C0001011011, and C0011111.

(a) Initial cupolet C0101011		
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
(b) Intermediate cupolet C0001011011		
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
(c) Target cupolet C0011111		
1	1559	0
1	1686	0
1	1143	1
1	1790	1
1	520	1
1	959	1

to Table II, this transition is accomplished by switching from C0101011 to C0001011011, at bin 1145 on lobe 1, and then by riding C0001011011 until its next intersection with a control plane, at bin 1790 on lobe 1. This is where the intermediate cupolet switches to the target cupolet. Thus, within our subset of 100 cupolets, we are able to find a controlled transition with $\mathcal{L}(\{C0101011, C0001011011, C0011111\}) = 1$. The lengths of these transitions are compared using the histogram shown in Figure 8(a).

Another example is the transition from the period-10 cupolet C0010110101 to the period-4 cupolet C01. The lengths of the twenty blind transitions range from 3 to 21 in the finite case, an average of 9.9167, while there are 8 blind transitions of infinite length. In contrast, the shortest controlled transition has length one. The distribution of these transition lengths is seen in Figure 8(b). As with the first example, these results suggest that controlled transitions can be more efficient than blind transitions, even when using only a small subset of the available cupolets.

The third example is the transition between cupolets C0000010011 and C00001, whose orbit has been seen in Figure 4. This transition is illustrated in Figure 9, which is explained more fully in Sec. V. C0000010011 has period 10 and of the resulting 50 blind transitions, approximately two fifths have infinite length, while the lengths of the remaining transitions range from 6 to 42, an average of 20.3103. The shortest controlled transition has length 8 and is represented by the transition sequence

$$\{C0000010011, C00111111, C0010101111, C0010100101, C00001\}.$$

The length of this controlled transition is compared to those of the blind transitions in Figure 8(c). This particular controlled transition has a length greater than any other shortest controlled transition that we found when using Dijkstra's algorithm to find the shortest controlled transition between all possible cupolet pairs in this cupolet subset. This suggests that within our formulation cupolets, C0000010011 and C00001 are well-separated.

Although the shortest controlled transition in this third example is longer than several of its corresponding blind transitions, it is important to be aware of three key points. First, while the controlled transitions are restricted to the subset of 100 cupolets, the blind transitions are allowed to use the full dynamics of the system and consequently may be traversing cupolets that do not belong to this subset. Indeed, the shortest blind transition of this third example rides along cupolet C0000100001001, which is not contained in our cupolet subset. When we extend the subset to the full cupolet library and include the cupolets' symmetric cousins, we find that the length of the shortest controlled transition of this example does in fact drop to 6, the length of this shortest blind transition. Enlarging the cupolet subset does not affect the lengths of the blind transitions.

Second, there is no way of predicting which combination of bin location and control bits result in shorter blind transitions without performing the equivalent of the shortest path analysis. If instead we employ a controlled transition, we know the exact trajectory of a controlled transition and can predict its length without running any prior simulations.

Finally, note that there are other ways of measuring cupolet transitions and hence of defining what constitutes an optimal path around a chaotic attractor. Our algorithm does so by finding an intermediate sequence of switchable cupolets that collectively traverses the double scroll attractor in the fewest loops. One may alternatively seek a cupolet transition sequence that minimizes the time-integrated path length of the transients or that minimizes the number of cupolets that must be traversed.^{11,26} Our approach may be quite easily adapted for these and other analogous approaches, provided that the corresponding weighted adjacency matrix and cupolet look-up table are amended to reflect the different path weights.

Our final example is the transition between cupolets C0010110011 and C0000000011, the results of which are displayed in Figure 8(d). The initial cupolet has period 10 and of the resulting 90 blind transitions, 50 do not stabilize onto

FIG. 9. The controlled transition of the third example of Sec. IV illustrating a navigation of the double scroll attractor. Beginning from the initial point 0 on lobe 0, neighborhoods of the attractor's 1-lobe may be sequentially visited by the path of this controlled transition.

the desired target cupolet. The lengths of the finite transitions range from 64 to 98, an average of 80.675. In contrast, the shortest controlled transition requires only one loop around the attractor. This example shows how effective controlled transitions can be in situations where the lengths of blind transitions can be very long.

V. DISCUSSION AND CONCLUDING REMARKS

We have presented two techniques for transitioning between the cupolets of chaotic systems. In our initial investigations, controlled transitions are usually more efficient and manageable than blind transitions, which typically involve longer chaotic transients that cannot be predicted in advance. This demonstrates that cupolets can provide a framework for various applications where it is desirable to have short transitions between orbits and where the dynamics are to be controlled.

Figure 9 illustrates this idea while also demonstrating the controlled transition sequence of our third example from Sec. IV. The black triangles mark the transition points where our algorithm guides the double scroll system from one cupolet to another while the black arrows indicate the flow direction. The black circles represent various neighborhoods that are visited by this controlled transition. This sequence of controlled transitions would navigate a trajectory sequentially starting from the indicated initial point to the other neighborhoods. This would allow for the calculations of global properties—such as Lyapunov exponents—to be localized over isolated regions of the attractor or within neighborhoods of UPOs.

Steering and targeting trajectories of nonlinear systems has been a goal of several experimental and theoretical

research studies, particularly in applications of low-energy transfer, unstable ballistic capture, and spacecraft guidance.^{27–33} These investigations exploit the same properties of chaotic systems that allow for cupolet generation, namely their sensitive dependence on initial conditions and abundance of UPOs. Controlled transitions between cupolets thus offer a novel approach to these applications. In particular, the deterministic nature of controlled transitions allows for assurances of length and, in physical situations, power and fuel consumption. For instance, when manipulating a spacecraft is required, it could be instructed to apply a sequence of controls to the spacecraft in order to conserve fuel and steer the spacecraft onto orbits of more desirable cupolets.

We also wish to point out that our algorithm for obtaining efficient controlled transitions between cupolets induces certainty into the control and steering of nonlinear systems onto periodic orbits. Effectively, this addresses the issue that although chaotic systems support short-term prediction capabilities, the same cannot be achieved in the long-term. In contrast, the algorithm presented here makes it possible to direct a chaotic trajectory onto a large set of the embedded UPOs, and hence to multiple locations around an attractor, simply by riding along segments of known periodic orbits, rather than depending on the trajectory's wandering evolution. This not only allows for the desired extraction or concatenation of controlled and periodic trajectories from a sea of chaotic ones but it also means that a chaotic trajectory's future behavior can be controlled for a long time.

Thus, the wide variety of dynamical behaviors provided by a set of cupolets, combined with their accessibility by small controls, makes cupolets particularly useful pathways when investigating a chaotic attractor. In summary, cupolets

have proven to be a valuable tool in exploring chaotic systems and appear to be promising for future applications.

ACKNOWLEDGMENTS

The authors thank the anonymous reviewers for their careful reading and helpful suggestions.

- ¹E. Ott, C. Grebogi, and J. A. Yorke, "Controlling chaos," *Phys. Rev. Lett.* **64**(11), 1196–1199 (1990).
- ²S. Hayes, C. Grebogi, and E. Ott, "Communicating with chaos," *Phys. Rev. Lett.* **70**(20), 3031 (1993).
- ³S. Hayes, C. Grebogi, E. Ott, and A. Mark, "Experimental control of chaos for communication," *Phys. Rev. Lett.* **73**(13), 1781–1784 (1994).
- ⁴D. Viswanath, "Symbolic dynamics and periodic orbits of the Lorenz attractor," *Nonlinearity* **16**(3), 1035 (2003).
- ⁵C. Grebogi, S. M. Hammel, J. A. Yorke, and T. Sauer, "Shadowing of physical trajectories in chaotic dynamics: Containment and refinement," *Phys. Rev. Lett.* **65**(13), 1527–1530 (1990).
- ⁶B. A. Coomes, H. Koçak, and K. J. Palmer, "Shadowing orbits of ordinary differential equations," *J. Comput. Appl. Math.* **52**(1), 35–43 (1994).
- ⁷B. A. Coomes, H. Koçak, and K. J. Palmer, "Long periodic shadowing," *Numer. Algorithms* **14**(1), 55–78 (1997).
- ⁸K. M. Short, R. A. Garcia, M. Daniels, J. Curley, and M. Glover, "An introduction to the KOZ scalable audio compression technology," in *AES 118th Convention Paper, Barcelona* (2005).
- ⁹K. M. Short, R. A. Garcia, and M. Daniels, "Scalability in KOZ audio compression technology," in *AES 119th Convention Paper, New York, NY* (2005).
- ¹⁰T. Shinbrot, E. Ott, C. Grebogi, and J. A. Yorke, "Using chaos to direct trajectories to targets," *Phys. Rev. Lett.* **65**(26), 3215 (1990).
- ¹¹E. M. Bollt *et al.*, "Optimal targeting of chaos," *Phys. Lett. A* **245**(5), 399–406 (1998).
- ¹²C. G. Schroer and E. Ott, "Targeting in Hamiltonian systems that have mixed regular/chaotic phase spaces," *Chaos* **7**(4), 512–519 (1997).
- ¹³W. L. Ditto, S. N. Rauseo, and M. L. Spano, "Experimental control of chaos," *Phys. Rev. Lett.* **65**(26), 3211 (1990).
- ¹⁴E. Barreto, E. J. Kostelich, C. Grebogi, E. Ott, and J. A. Yorke, "Efficient switching between controlled unstable periodic orbits in higher dimensional chaotic systems," *Phys. Rev. E* **51**, 4169–4172 (1995).
- ¹⁵J. Starrett and R. Tagg, "Control of a chaotic parametrically driven pendulum," *Phys. Rev. Lett.* **74**(11), 1974 (1995).
- ¹⁶A. T. Parker, "Topics in chaotic secure communication," Ph.D. thesis (University of New Hampshire, 1999).
- ¹⁷K. Zarringhalam, "CUPOLETS: Chaotic unstable periodic orbits theory and applications," Ph.D. thesis (University of New Hampshire, 2007).
- ¹⁸K. Zarringhalam and K. M. Short, "Generating an adaptive multiresolution image analysis with compact cupolets," *Nonlinear Dyn.* **52**, 51–70 (2008).
- ¹⁹F. M. Carrano and J. J. Prichard, *Data Abstraction and Problem Solving with C++: Walls and Mirrors*, 3 ed. (Addison-Wesley, 2002).
- ²⁰M. L. Fredman and R. E. Tarjan, "Fibonacci heaps and their uses in improved network optimization algorithms," *J. ACM (JACM)* **34**(3), 596–615 (1987).
- ²¹K. M. Short and A. T. Parker, "Security issues in chaotic communications," *Paper presented at the SIAM Conference on Dynamical Systems*, Snowbird, UT, May 23–27 (1999).
- ²²T. Matsumoto, L. Chua, and M. Komuro, "The double scroll," *IEEE Trans. Circuits Syst.* **32**(8), 797–818 (1985).
- ²³K. T. Hansen, "Alternative method to find orbits in chaotic systems," *Phys. Rev. E* **52**(3), 2388 (1995).
- ²⁴B.-L. Hao and W.-M. Zheng, *Applied Symbolic Dynamics and Chaos* (World Scientific, Singapore, 1998).
- ²⁵Z.-B. Wu and J.-Y. Zeng, "A method to find unstable periodic orbits for the diamagnetic Kepler problem," *Phys. Scr.* **61**(4), 406 (2000).
- ²⁶E. G. Johnson, "Controlled transitions between orbits in nonlinear systems," Master's thesis (University of New Hampshire, NH, USA, 2009).
- ²⁷T. von Rosenvinge, J. C. Brandt, and R. W. Farquhar, "The international cometary explorer mission to comet Giacobini-Zinner," *Science* **232**(4748), 353–356 (1986).
- ²⁸E. M. Bollt and J. D. Meiss, "Targeting chaotic orbits to the moon through recurrence," *Phys. Lett. A* **204**(5), 373–378 (1995).
- ²⁹W. S. Koon, M. W. Lo, J. E. Marsden, and S. D. Ross, "Low energy transfer to the moon," *Celestial Mech. Dyn. Astron.* **81**(1–2), 63–73 (2001).
- ³⁰E. Belbruno, *Capture Dynamics and Chaotic Motions in Celestial Mechanics: With Applications to the Construction of Low Energy Transfers* (Princeton University Press, 2004).
- ³¹E. E. N. Macau and C. Grebogi, "Control of chaos and its relevancy to spacecraft steering," *Philos. Trans. R. Soc. A* **364**(1846), 2463–2481 (2006).
- ³²K. E. Davis, R. L. Anderson, D. J. Scheeres, and G. H. Born, "Optimal transfers between unstable periodic orbits using invariant manifolds," *Celestial Mech. Dyn. Astron.* **109**(3), 241–264 (2011).
- ³³P. R. Griesemer, C. Ocampo, and D. S. Cooley, "Targeting ballistic lunar capture trajectories using periodic orbits," *J. Guidance, Control, Dyn.* **34**(3), 893–902 (2011).