

# Mutual Stabilization in Chaotic Hindmarsh–Rose Neurons

John E. Parker <sup>\*,†</sup>  and Kevin M. Short 

Integrated Applied Mathematics Program, Department of Mathematics and Statistics,  
University of New Hampshire, Durham, NH 03824, USA; kevin.short@unh.edu

\* Correspondence: john.parker@unh.edu

† Current address: Department of Mathematics, University of Pittsburgh, Pittsburgh, PA 15260, USA.

**Abstract:** Recent work has highlighted the vast array of dynamics possible within both neuronal networks and individual neural models. In this work, we demonstrate the capability of interacting chaotic Hindmarsh–Rose neurons to communicate and transition into periodic dynamics through specific interactions which we call mutual stabilization, despite individual units existing in chaotic parameter regimes. Mutual stabilization has been seen before in other chaotic systems but has yet to be reported in interacting neural models. The process of chaotic stabilization is similar to related previous work, where a control scheme which provides small perturbations on carefully chosen Poincaré surfaces that act as control planes stabilized a chaotic trajectory onto a cupolet. For mutual stabilization to occur, the symbolic dynamics of a cupolet are passed through an interaction function such that the output acts as a control on a second chaotic system. If chosen correctly, the second system stabilizes onto another cupolet. This process can send feedback to the first system, replacing the original control, so that in some cases the two systems are locked into persistent periodic behavior as long as the interaction continues. Here, we demonstrate how this process works in a two-cell network and then extend the results to four cells with potential generalizations to larger networks. We conclude that stabilization of different states may be linked to a type of information storage or memory.

**Keywords:** Hindmarsh–Rose neuronal model; chaotic dynamics; dynamical systems; cupolets; mutual stabilization



**Citation:** Parker, J.E.; Short, K.M. Mutual Stabilization in Chaotic Hindmarsh–Rose Neurons. *Dynamics* **2023**, *3*, 282–298. <https://doi.org/10.3390/dynamics3020017>

Academic Editor: Christos Volos

Received: 21 April 2023

Revised: 15 May 2023

Accepted: 17 May 2023

Published: 19 May 2023



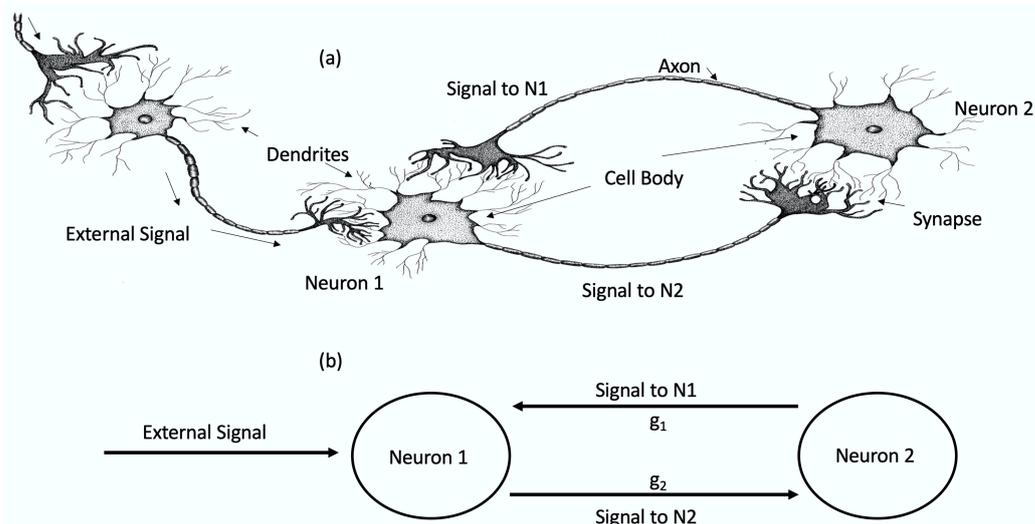
**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

This paper continues a series of developments investigating the possibility that neuron models in chaotic regimes might enter into a state of induced or mutual stabilization that is persistent and, through interaction, becomes periodic and locked in. The first approach to this work involved the Fitzhugh–Nagumo (FHN) model [1–3], where two neurons are bidirectionally and electrically coupled [4]. Figure 1 provides an artist’s rendering (A) and the schematic of the bidirectional network where  $g$  represents the interaction between the two neurons (B), reproduced from Parker and Short [4] with permission. In the FHN model, each neuron is represented by two dynamical variables, representing the membrane potential and the recovery dynamics. Since the system is two-dimensional, no chaos can exist, but when coupled together through an interaction function driven by the potential difference between the neurons, the system becomes four-dimensional and is shown to exhibit chaos through a large part of its parameter regime. It was shown in Parker and Short [4], Parker [5] that when a transient external impulse was injected into the coupled system, and the synapse “learning” dynamics was modeled by a sigmoidal synaptic learning function that was an extension of Hebb’s [6] and Oja’s rules [7], then the system could transition into a mutually stabilized periodic state.

In order to extend this work, an investigation was conducted into the possibility of stabilizing a fully chaotic neuron model using a control method that has been successful on other chaotic systems. The neuron model chosen was the Hindmarsh–Rose (HR) model [8],

since it is a three-dimensional system and has parameter regimes where the behavior is fully chaotic. The first paper on the HR study [9] looked at whether a particular control scheme could be used to stabilize the HR neuron onto a set of its unstable periodic orbits, which are generally dense within a chaotic system, but are all unstable so are not readily accessible. The control approach closely followed the work in Morena and Short [10,11], which was an extension of the initial work in Parker [12], Hayes et al. [13,14].



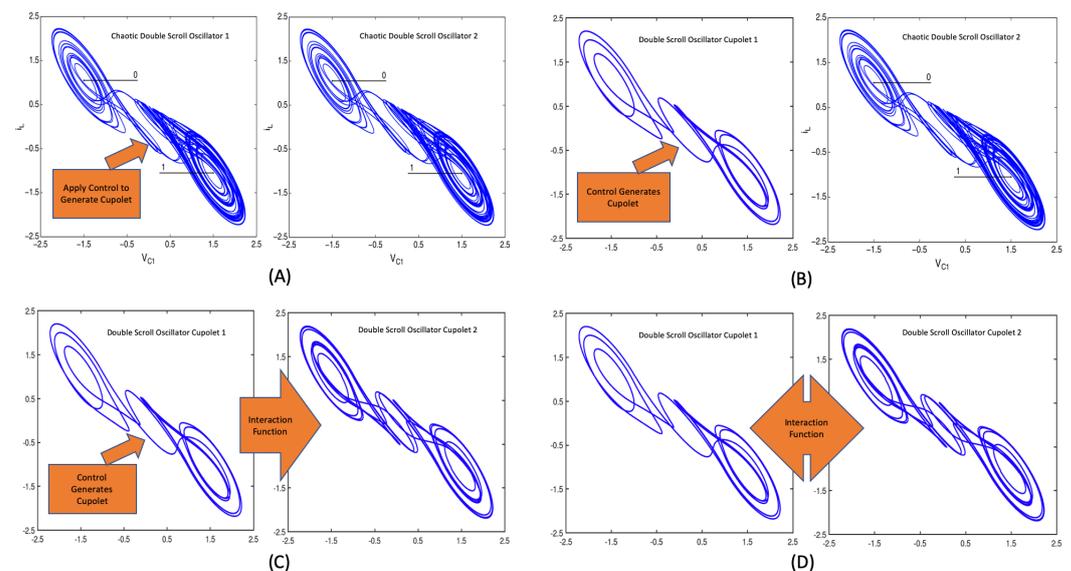
**Figure 1.** (a) Artist interpretation of two bidirectionally interacting neurons (N1 and N2) that receive an external signal from a nerve. (b) Schematic of (a) for two FHN neurons where  $g_i$  represents the synaptic connection between the neurons, as seen in Parker and Short [4]. Reproduced from John E. Parker and Kevin M. Short, “Sigmoidal synaptic learning produces mutual stabilization in chaotic FitzHugh–Nagumo model”, *Chaos* 30, 063108 (2020) <https://doi.org/10.1063/5.0002328> (accessed on 21 April 2023), with the permission of AIP Publishing.

In these works, the attractor of a dynamical system called the double scroll or Chua’s oscillator [15] was intersected by two control planes that were perpendicular to the local flow direction, but the methods can be extended to the Lorenz [16] and Rossler [17] systems, as well as one-dimensional maps [12]. The intersection with the control plane was effectively a line that was then divided into small bins. The controls were applied when the system passed through a bin; it was either allowed to pass through the bin relatively unperturbed, or it was kicked to a nearby bin. It was found that when a repeating series of kicks was applied to the chaotic system (encoded as a binary bit stream), it would pass through a chaotic transient, but would then stabilize onto a periodic orbit that was close to one of the system’s unstable periodic orbits [18]. The resulting periodic orbits have been called *cupolets* (chaotic unstable periodic orbit-lets). Most surprising was the fact that for almost all control codes up to 16 bits long, there was a one-to-one correspondence between the control code and the resulting cupolet, and the stabilization onto the cupolet was independent of the initial state of the chaotic system. Numerous extensions of this work have been found, including development of steering mechanisms to jump from one cupolet to another [19], determination of a fundamental set of cupolets [20], and a form of chaotic entanglement [10,11,21].

As a first step toward determining if stabilization of unstable periodic orbits was possible in a neuron model, the work in Parker and Short [9] examined the HR neuron and set up a similar control structure as used for the other chaotic systems listed in the previous paragraph. In the HR case, the control planes were positioned so that one control plane intersected the region where the HR model spikes and the other where the HR model moves into a refractory period after spiking. The kicks along the control plane serve as proxies for the impulsive kicks received by an incoming signal from a nearby neuron. It

was shown in Parker and Short [9] that stabilized cupolets were possible in the HR system when it was in a chaotic parameter regime. This demonstrated that the chaos in the HR system could be transformed into periodic behavior through the imposition of a series of impulsive signals, where the impulsive signals are idealizations of the action potential signals received from nearby neurons. The present paper will extend this result to show that when two HR neurons are coupled, it is possible that the necessary impulsive signals to induce stabilization onto periodic behavior can be provided by the coupled neuron and, further, that it is possible for two neurons to enter into a state of mutual stabilization that is persistent as long as the interaction continues.

In Parker and Short [4], we discussed an overview of published work on mutual stabilization in communicating double scroll oscillators and provide Figure 2 from Parker and Short [4], reproduced with permission, for illustration. In this figure, and as reported in Morena and Short [11], mutual stabilization was shown to exist between interacting cupolets of the double scroll oscillator. The authors demonstrated that there exist certain interaction functions that transform the visitation sequence (the symbolic dynamics) of one cupolet into a control sequence on the second cupolet. In mutual stabilization, this process repeats from the second cupolet back to the first cupolet, where the interaction function transforms the visitation sequence of the second cupolet into a control on the first cupolet. For certain interaction functions and certain cupolets, the cupolets would lock into a state of persistent, periodic behavior, a type of mutual stabilization called chaotic entanglement [11]. It also detailed various interaction functions that could be used to induce mutual stabilization between two double scroll oscillators.



**Figure 2.** Mutual stabilization order of events for the double scroll oscillator. (A) External control is applied to only System 1. (B) System 1 stabilizes onto a cupolet due to the external control. (C) Information from System 1, stabilized onto a cupolet, is transferred as a control onto System 2, leading to System 2 stabilization onto a cupolet. (D) Information is sent back from System 2 to System 1, replacing the original external signal. Both systems are locked into persistent, periodic orbits while the interaction function remains. Reproduced from John E. Parker and Kevin M. Short, “Sigmoidal synaptic learning produces mutual stabilization in chaotic FitzHugh–Nagumo model”, *Chaos* 30, 063108 (2020) <https://doi.org/10.1063/5.0002328> (accessed on 21 April 2023), with the permission of AIP Publishing.

In the present work, we apply a similar process to communicating Hindmarsh–Rose neurons with various network topologies to see if stabilization and, ultimately, mutual stabilization can occur. For mutual stabilization, the analogy to biological neurons is that a signal traveling along a nerve may reach a terminus that includes a penultimate neuron

that is connected over a synapse to a final neuron, but the final neuron is also connected back to the penultimate neuron, e.g., Figure 1a. In biological neurons, the communication would be through the electrical impulse of the action potential, and is a function of the state of the synapse. In our model, the action potentials will be simplified to delta-function-like impulses, and the synaptic communication between the neurons will be modeled by an interaction function called *Integrate-And-Fire* that was used in Morena and Short [11] and inspired by Integrate-And-Fire dynamics first published by Lapicque [22].

The following sections first provide a brief overview of Hindmarsh–Rose neural model and the *Integrate-And-Fire* interaction function. Then we show the results of applying this interaction function to several different networks, initially with a bidirectional two-neuron system, then a chain of neurons leading to the bidirectional two-neuron system, and finally a loop of neurons. We conclude by discussing the possibility that a wide variety of networks of neurons can be constructed and stabilized with proper selection of interaction function and network topology. Further, the results presented in this paper represent a mechanism where information becomes trapped and stored in the mutual stabilization of the neurons, so it can be considered to be a form of memory.

## 2. Materials and Methods

### 2.1. Hindmarsh–Rose Model

In this section, we briefly describe the Hindmarsh–Rose system, the control scheme for generating cupolets, and the interaction function for communication between multiple systems.

The dynamical system for the Hindmarsh–Rose neural model, as originally stated in Hindmarsh and Rose [8], is given in Equation (1). This nonlinear system provides a simple three-dimensional model for neural firing, specifically bursting dynamics where the neuron fires multiple spikes before a period of recovery, called the refractory period.

$$\begin{aligned}\dot{x} &= y - ax^3 + bx^2 + I - z \\ \dot{y} &= c - dx^2 - y \\ \dot{z} &= r(s(x - x_r) - z)\end{aligned}\tag{1}$$

In this model, the  $x$  dynamical variable represents the membrane voltage of the neuron, meaning that the typical spikes that are seen in neurons will occur in this variable. Both the  $y$  and  $z$  dynamical variables play a role in the recovery time to spike again, but at different time scales. The  $y$  dynamical variable is on a faster time scale and contributes to creating several spikes in a row to form a burst, while the  $z$  dynamical variable causes the system to recover between bursts. In the examples shown here, we used the parameters in a chaotic regime, as seen in Parker and Short [9], specifically,  $a = 1$ ,  $b = 3$ ,  $c = 1$ ,  $d = 5$ ,  $s = 4$ ,  $r = 0.006$ ,  $x_r = -8/5$ , and  $I = 3.25$ .

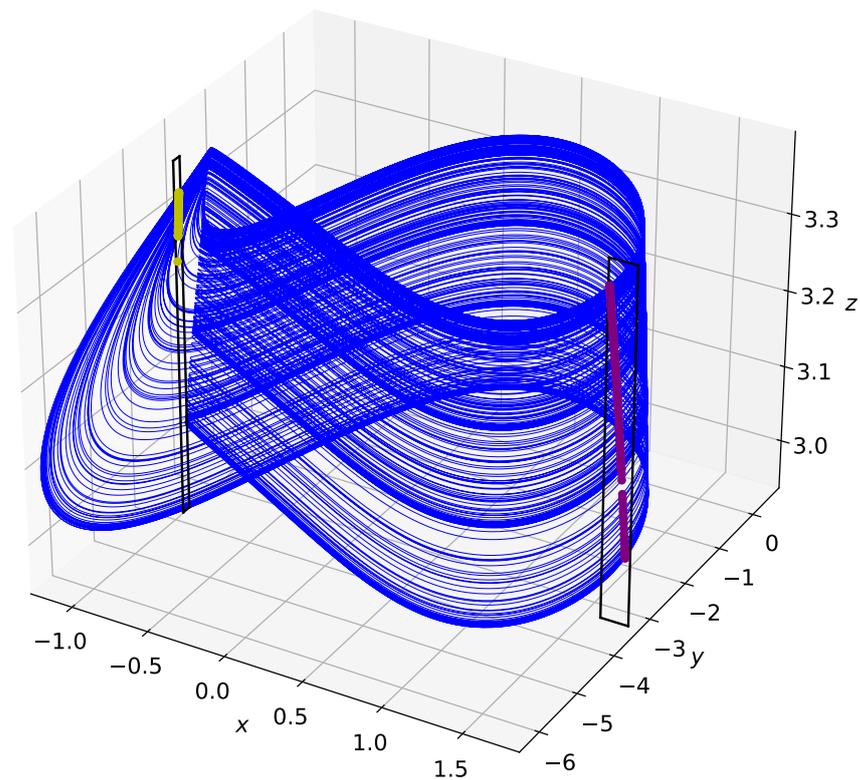
The HR model has been studied numerous times and we provide a few examples and note that the examples provided are not exhaustive. Innocenti et al. [23] and Innocenti and Genesio [24] investigated the chaotic dynamics of the HR model, while Doungmo Goufo and Tabi [25] explored chaotic dynamics with an external input. The reason why bursting oscillations occur in the HR model was explored in Wang [26]. A thorough bifurcation analysis was presented in Storace et al. [27]. The chaotic dynamics of the HR model were controlled in a similar procedure to this work in Jun et al. [28]. In other related work, Castanedo-Guerra et al. [29] showed synchronization of coupled HR neurons and Wang et al. [30] provided a bifurcation analysis on coupled HR neural dynamics. Recently, Branciforte et al. [31] introduced the concept of a hyperneuron, and presented an example of such based on the HR model, for in silico networks of neurons for improved parallelization. The last example is the current author's previous work, where the HR system in a chaotic parameter regime was controlled to produce cupolets in Parker and Short [9]. For the present paper, we use the same chaotic parameter regime that was used to generate cupolets in Parker and Short [9]. In the next section, we briefly describe the control scheme that is used to generate cupolets in the HR system.

## 2.2. Control Scheme

This section provides only a concise description of the method used to generate cupolets in the HR system. For a more thorough description of cupolet generation in the HR model, see Parker and Short [9].

The overall idea behind the control scheme is to repeatedly provide small perturbations to a trajectory when it intersects with particular Poincaré sections, which we will call control planes. These small perturbations are idealizations of the impulses that may be received from nearby neurons in biological systems. The perturbations are applied along two control planes, one that is designated as Poincaré Surface 0 (PS0) and the other as Poincaré Surface 1 (PS1), and one can create symbolic dynamics for a given trajectory simply by keeping track of the visits to the control planes. Figure 3, reproduced with permission from Parker and Short [9], shows the control planes and where the chaotic trajectory intersects with each control plane. The intersection with the control planes are approximately one-dimensional and are subdivided into  $M$  bins, with  $M$  large enough such that the bin width is small relative to the length of the control plane. In the work presented here, we use  $M = 1600$  bins on PS0 (bin size approximately  $9.5 \times 10^{-5}$  in the  $y$  direction) and  $M = 1600$  bins on PS1 (bin size approximately  $1.1 \times 10^{-4}$  in the  $x$  direction). The type of perturbation is determined by a binary control sequence that is cyclically iterated through with each intersection of a control plane. If the bit in the control sequence is a 0, a microcontrol is applied that moves the trajectory to the center of the bin that it intersected. If the bit in the control sequence is a 1, a macrocontrol is applied. A macrocontrol is defined as a kick to the nearest bin on the control plane that has the smallest non-zero difference in symbolic dynamics after a fixed interval. In other words, for the future  $N$  loops around the attractor, each bin has representative symbolic dynamics. The symbolic dynamics are the binary string representing the control planes visited by the trajectory, and a macrocontrol would shift the trajectory to the nearest bin with the smallest nonzero difference in symbolic dynamics. Consider  $N = 4$  loops around the attractor and two neighboring, but not necessarily adjacent, bins  $\alpha$  and  $\beta$  on PS1. The symbolic dynamics for bin  $\alpha$  can be represented by  $\alpha_1\alpha_2\alpha_3\alpha_4$  and is a binary sequence where each  $\alpha_i$  is a 0 or 1 to represent the control plane passed. The symbolic dynamics for bins  $\beta$  can be represented similarly. Then, a macrocontrol would perturb a trajectory from  $\alpha$  to  $\beta$  if  $\alpha_i = \beta_i$  for  $i = 1, 2, 3$  and  $\alpha_4 \neq \beta_4$ , and  $\beta$  is the nearest bin that has this property. This can be seen more clearly by defining a coding function that takes the symbolic dynamics of each bin as a binary decimal,  $r_N(x) = \sum_{n=1}^N \alpha_n / 2^n$  where  $x$  is the bin along the control plane. Then, a macrocontrol from bin  $j$  is simply defined as  $\min |r_N(x_k) - r_N(x_j)| > 0$  and  $\min |k - j|$ , where  $k \neq j$ .

As discussed in Parker and Short [9], this control scheme, when driven by a periodic sequence of binary controls, will generate periodic orbits in the chaotic system. The resulting periodic orbit is generally independent of the initial condition—essentially, the periodic orbit forms once the system dynamics stumble upon a part of the chaotic attractor that is “in sync” with the controls. The resulting periodic orbit is called a *cupolet* and would be unstable without the controls. For the simulations presented here and in [9], we consider a cupolet to have formed when the control sequence applies the same perturbations in the same sequence of bins, meaning that the trajectory is now stable under the controls. When using control sequences up to 12 bits, almost every control sequence generates a cupolet. The naming convention for a cupolet is simply a ‘C’ followed by the control sequence applied (e.g., C110101). In previous research, the control sequences were in essentially one-to-one correspondence with the resulting cupolets; however, for the HR system the control sequences do not all have a one-to-one correspondence, as some control sequences will generate homologous cupolets. Homologous cupolets occur when the cupolet structures differ yet the generating control sequences are the same and are designated by appending an alphabetical character to the end (e.g., C11A, C11B, etc.). Hence, the resulting homologous cupolet that arises is dependent on the initial bin where the system starts. For all control sequences up to 12 bits, we found that approximately 84% produced homologous cupolets.



**Figure 3.** Poincaré surfaces (black rectangles) acting as control planes on the Hindmarsh–Rose chaotic attractor. The left control plane represents Poincaré Surface 0 (PS0) and the right control plane represents Poincaré Surface 1 (PS1). Intersections of the trajectory with each control plane are shown. Reproduced from John E. Parker and Kevin M. Short, “Cupolets in a chaotic neuron model”, *Chaos* 32, 113104 (2022) <https://doi.org/10.1063/5.0101667> (accessed on 21 April 2023), with the permission of AIP Publishing.

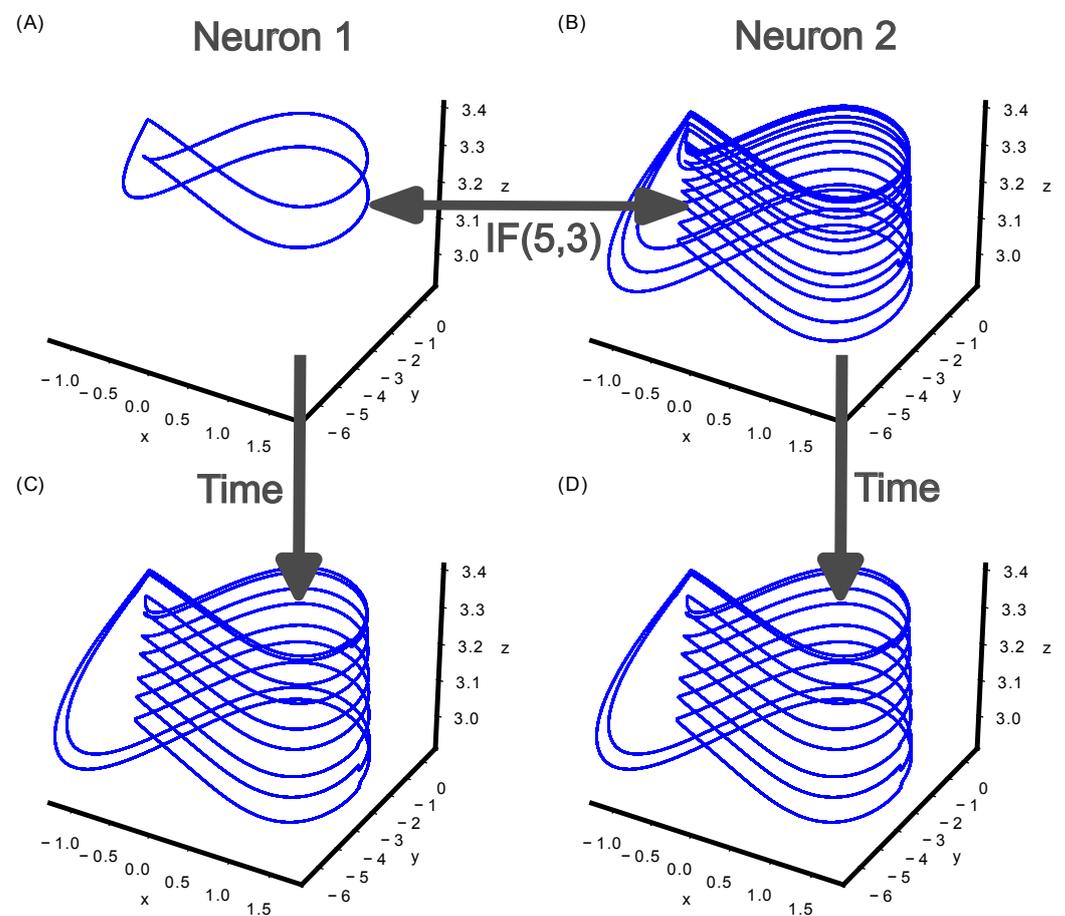
In the following section, we discuss how multiple HR neurons can communicate through specific interaction functions that transmit information from one HR neuron to another, and the resulting controls induce cupolet formation that leads to various forms of persistent states even though the neurons are in the chaotic regime.

### 2.3. Integrate-And-Fire Interaction Function

The Integrate-And-Fire neural model is one of the oldest models of neural behavior. Originally proposed in 1907 by Lapique [22], and translated in Brunel and van Rossum [32], the Integrate-And-Fire function reduces the action potential of a neuron to a simple mechanism. The neuron continually sums up all of its inputs until a threshold is reached, at which point an action potential is released as a delta-like spike. Morena and Short [11] implemented an analogous *Integrate-And-Fire* interaction function that is adapted from the Integrate-And-Fire neural and laser dynamics seen in Racicot and Longtin [33] and Hall et al. [34]. The version of the Integrate-And-Fire interaction function used in Morena and Short [11] allowed for each respective system to evolve toward a desired state before an interaction occurred. In all the interactions shown in this work, we do not allow for the systems to evolve toward a desired state before interaction, but, rather, allow the communication to happen instantaneously. The *Integrate-And-Fire* interaction function is used here to mediate communication between coupled HR neurons. Future work will explore relaxing the feedback time to allow for broader stabilization possibilities.

Each cupolet generates a visitation sequence, or symbolic dynamics, when the trajectory visits a sequence of the control planes, Figure 3. For example, consider Figure 4A, which shows cupolet C001. As cupolet C001 traverses one period, the trajectory visits PS1 twice (consecutively) and visits PS0 once, resulting in visitation sequence of V110. If the

first bit of the earliest intersection with cupolet C001 is with PS1 and 9 bits are recorded, we would expect V110110110. This process can be extended for any number of bits. For a receiving system that is connected via the *Integrate-And-Fire* (IF) interaction function, the most recent  $Q$  bits of the visitation sequence are read in and the control applied is either a 1 or 0, depending on if the bits sum to a threshold,  $\kappa$ . The return value is a 1 if the sum meets or exceeds  $\kappa$ , and a 0 if not. This is abbreviated as  $IF(Q, \kappa)$ . As an example, take the ten bit visitation sequence of C001, V1101101101. If using  $IF(4, 3)$  then any sequence of length  $Q = 4$  bits that contains two 0s will produce an output of 0, which occurs when the bits 0110 pass through  $IF(4, 3)$ , otherwise  $IF(4, 3)$  returns a 1. For the entire ten-bit visitation sequence,  $IF(4, 3)$  would then produce 1101101101... However, if we use  $IF(4, 2)$  then the output would be all 1s. Alternatively, using  $IF(Q = 3, \kappa = 3)$ , the returned value is a string of only 0s. A 0 occurs when  $Q = 3$  bits because any group of three bits does not sum to  $\kappa = 3$ , since only two consecutive 1s exists in the visitation sequence.



**Figure 4.** Mutual stabilization of two Hindmarsh–Rose neurons. (A) Neuron 1 stabilizes onto cupolet C001. (B) Neuron 2 stabilizes onto cupolet C01. (C) Neuron 1 receives the controls that are determined from the visitation sequence of Neuron 2 via  $IF(5, 4)$  and replaces the original control sequence 001. This results in the repeated control 11 leading Neuron 1 to stabilize onto cupolet C11A. (D) Neuron 2 receives the controls that are determined from the visitation sequence of Neuron 1 via  $IF(5, 3)$ , replacing the original control sequence 01. This results in the repeated control 11 leading Neuron 2 to stabilize onto cupolet C11A.

For the simulations presented in this work, after a control is applied to the receiving system, the trajectory continues and the process repeats at the next intersection with a control plane. While a cupolet evolves, the most recent  $Q$  length visitation sequence is constantly provided as input to the interaction function  $IF(Q, \kappa)$  to generate a control bit. This means that the most recent  $Q$  bits dictate the control output. An example is given in

Table 1, where  $Q = 4$  and  $\kappa = 4$ , and another example is given in Table 2, where  $Q = 4$  and  $\kappa = 3$ . In both tables, the  $IF(Q,\kappa)$  is applied to the visitation sequence from cupolet C11A, which is V0111101111.

**Table 1.** Generation of control sequence using  $IF(4,4)$  applied to a visitation sequence from C11A.

Visitation Sequence	Most Recent $Q = 4$ Bits	$IF(4,4)$	Control Sequence
...0111	0111	0	0
...01111	1111	1	01
...011110	1110	0	010
...0111101	1101	0	0100
...01111011	1011	0	01000
...011110111	0111	0	010000
...0111101111	1111	1	0100001
...01111011111	1111	1	01000011

**Table 2.** Generation of control sequence using  $IF(4,3)$  applied to a visitation sequence from C11A.

Visitation Sequence	Most Recent $Q = 4$ Bits	$IF(4,3)$	Control Sequence
...0111	0111	1	1
...01111	1111	1	11
...011110	1110	1	111
...0111101	1101	1	1111
...01111011	1011	1	11111
...011110111	0111	1	111111
...0111101111	1111	1	1111111
...01111011111	1111	1	11111111

The control scheme detailed in Section 2.2 applies a repeated binary control sequence where 1 corresponds to a macrocontrol and a 0 corresponds to a microcontrol. In Table 2, the repeated visitation sequence of C11A generates the control sequence 111111 as a result of  $IF(4,3)$ . This interaction function produces only a control bit of 1 that is applied when the trajectory visits a control plane. Additionally, this control sequence is the same control sequence that was used to generate the original cupolet C11A. Therefore, if C11A were to interact with a chaotic neuron unidirectionally via  $IF(4,3)$  then this second neuron will generate one of the homologous C11 cupolets.

### 2.4. Numerics

Numerical integration was performed using Runge–Kutta fourth-order method with a timestep of  $dt = 1/128$ . All numerical code, initial figures (some additional figure labeling was added using Inkscape), and further analysis were performed in Python. For more details on the numerical schemes used, integration, and code implementation, please refer to the GitHub repository in the Data Availability section.

### 3. Results

We present the results of several different topological networks of neurons that are communicating through the *Integrate-And-Fire* interaction function. First, we discuss two neurons interacting with two-way communication (bidirectional). We show how controls propagate to each neuron, leading the chaotic dynamics to stabilize onto a periodic orbit as long as the correct controls are applied. If the incorrect controls are applied, then the dynamics will remain chaotic.

### 3.1. Bidirectional Model

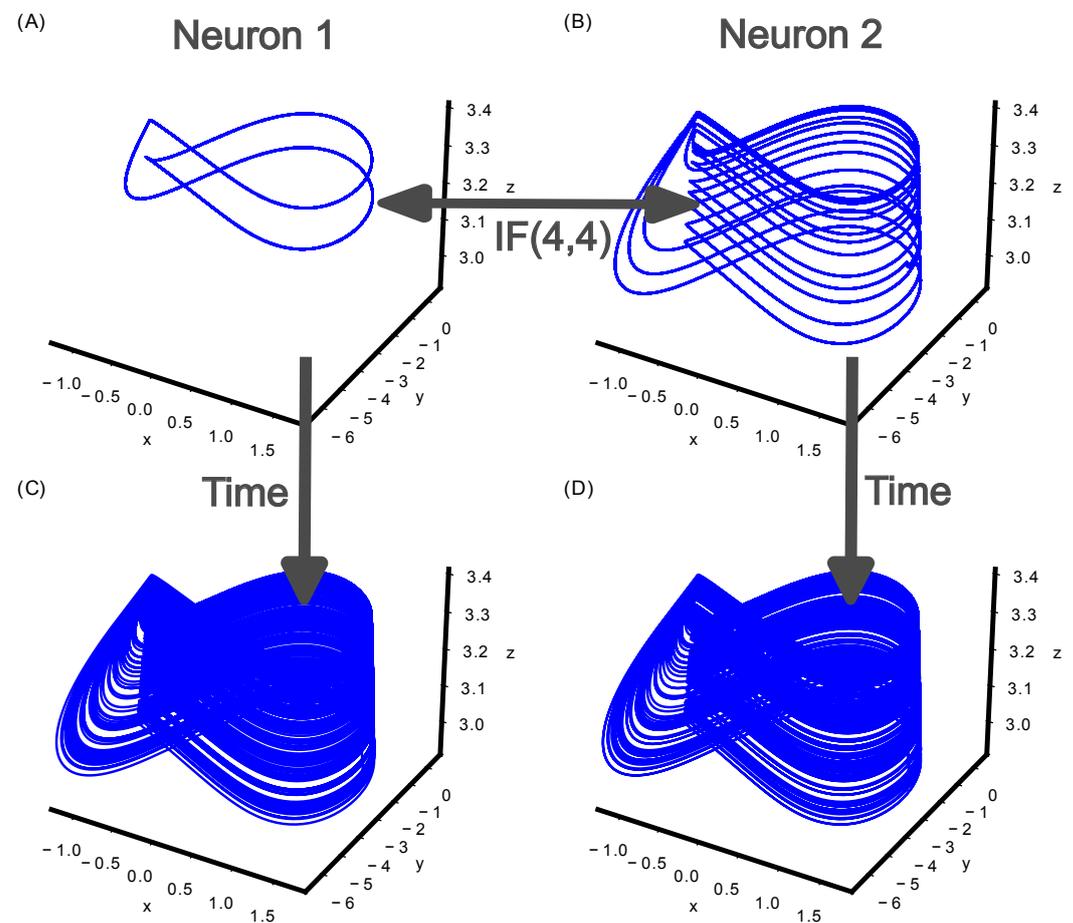
This section expands on the chaotic stabilization from Parker and Short [9] by demonstrating how mutual stabilization can explicitly occur between two interacting HR neurons. The mutual stabilization process is analogous to the mutual stabilization reported in Morena and Short [11], Morena [35] and illustrated in Figure 2. First, two neurons independently evolve in the chaotic regime. Then, a control sequence is applied to each neuron that is known to generate a cupolet. These two control sequences, which do not have to be identical, are indicative of separate neurons receiving different external signals. Next, we continue to apply the control sequence cyclically until each neuron has evolved for a certain number of iterations and stabilized onto a cupolet. We record the visitation sequence of each neuron. This process of generating a single cupolet is identical to the process described in Parker and Short [9]. Once the cupolets are established, the cupolets begin to communicate bidirectionally. That is, Neuron 1 can send information to Neuron 2 and Neuron 2 feeds back information to Neuron 1, where the method of communication is mediated by the interaction function  $IF(Q,\kappa)$ . At this point, the external controls can be replaced by the bidirectional communication between the neurons. If the choice of  $Q$  and  $\kappa$  are such that  $IF(Q,\kappa)$  generate known control sequences that lead to cupolets, then the neurons will transition to cupolets, leading to mutual stabilization. However, if this is not the case, where the output of  $IF(Q,\kappa)$  does not provide control sequences that lead to cupolets, then the neurons will not stabilize but, rather, will continue exhibiting chaotic dynamics.

Figure 4 shows an example of mutual stabilization for two Hindmarsh–Rose neurons that are communicating via  $IF(5,4)$ . After an initial chaotic transient, the control sequence 001 is repeatedly applied to Neuron 1, leading to Neuron 1 stabilizing onto cupolet C001. Figure 4A shows the resultant cupolet C001 during iterations 1,280,000 to 2,560,000. During the same simulation time, control 01 is repeatedly applied to Neuron 2, leading Neuron 2 to stabilize onto cupolet C01. Figure 4B shows the resultant cupolet C01 during iterations 1,280,000 to 2,560,000. Between iterations 2,560,000 to 5,120,000, the two neurons begin bidirectional communication mediated by  $IF(5,3)$ . Neuron 1 sends the most recent visitation sequence of length  $Q = 5$  to  $IF(5,3)$  and the output of  $IF(5,3)$  is the next control on Neuron 2. Neuron 2 sends the most recent visitation sequence of length  $Q = 5$  to  $IF(5,3)$  and the output of  $IF(5,3)$  is the next control bit on Neuron 1. For both Neuron 1 and Neuron 2, each visitation sequence of length  $Q = 5$  results in an output of 1 from  $IF(5,3)$ . This results in Neuron 1 stabilizing onto cupolet C11A (Figure 4C) and Neuron 2 stabilizing onto cupolet C11A (Figure 4D). This is one of the simplest cases of mutual stabilization where heterogeneous cupolets (Neuron 1 and Neuron 2 stabilized onto different cupolets, C001 and C01, respectively) transitioned to homogeneous cupolets (Neuron 1 and Neuron 2 stabilized onto the same cupolet, C11A) once the cupolets started interacting.

The mutual stabilization produced in Figure 4 is dependent on the specific interaction function between the two neurons. If the interaction function is  $IF(4,4)$ , then the last 4 bits of the respective visitation sequence are used instead of 5 bits. An  $IF(4,4)$  output value of 1 corresponds to four input bit values that are all 1, which means that four consecutive spikes occurred. Therefore, the only time the  $IF(4,4)$  function results in 1 is when the symbolic dynamics of a cupolet has a burst of at least four spikes. Figure 5 shows the same initial cupolets as Figure 4, yet no mutual stabilization occurs because of the difference in interaction function.

Figure 5A shows the cupolet C001 during iterations 1,280,000 to 2,560,000. During the same simulation time, the external control 01 is repeatedly applied to Neuron 2, leading Neuron 2 to stabilize onto cupolet C01. Figure 5B shows the cupolet C01 during iterations 1,280,000 to 2,560,000. Between iterations 2,560,000 to 5,120,000, the bidirectional communication between the neurons is mediated via  $IF(4,4)$ . The visitation sequences produced by Neuron 1 and Neuron 2 result in an output of mainly microcontrols, 0, from  $IF(4,4)$ . This does not result in Neuron 1 (Figure 5C) or Neuron 2 stabilizing onto a cupolet (Figure 4D) during the simulation time, even though the initial cupolet formed is identical to the

scenario in Figure 4. Both neurons return to chaotic neural firing since the controls did not drive the system to stabilize the chaotic trajectories onto cupolets.



**Figure 5.** Failed mutual stabilization of two communicating Hindmarsh–Rose neurons. (A) Neuron 1 stabilizes onto cupolet C001. (B) Neuron 2 stabilizes onto cupolet C01. (C) Neuron 1 receives the controls that are determined from the visitation sequence of Neuron 2 via  $IF(4,4)$  and replaces the original control sequence 001. The resultant controls are not sufficient to establish a cupolet. (D) Neuron 2 receives the controls that are determined from the visitation sequence of Neuron 1 via  $IF(4,4)$ , and replaces the original control sequence 01. The resultant control is not sufficient to establish a cupolet.

### 3.2. Networks

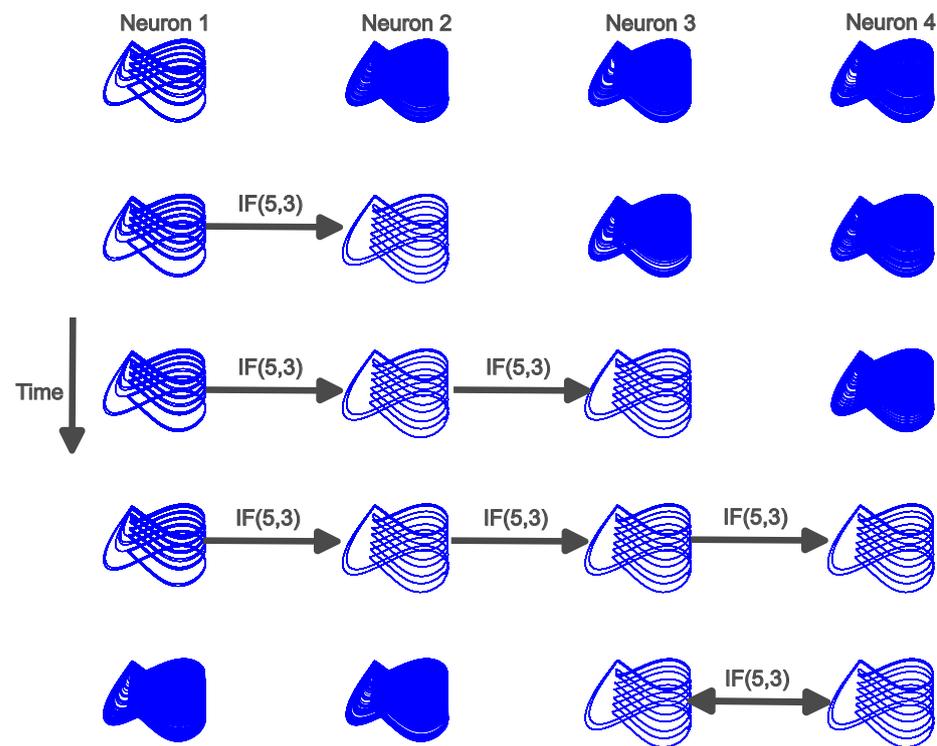
This section considers networks of neurons that are arranged in various topologies with the goal of achieving a version of mutual stabilization, either in the entire network or in a subset of the network. In each example, there is a feedback component that induces persistent mutual stabilization.

#### 3.2.1. Chain Mutual Stabilization

The first network demonstrates another case of mutual stabilization. This case occurs at the end of a chain of neurons, representing a nerve, and it involves the bidirectional interaction that was discussed in the previous section. The chain of neurons will comprise four neurons, where each neuron is initially exhibiting chaotic neural firing. The basic structure is as follows. At the beginning of the chain, Neuron 1 receives a binary control signal and this leads Neuron 1 to stabilize onto a cupolet. Information about this cupolet is passed through  $IF(Q,\kappa)$  to the next neuron, Neuron 2, which stabilizes onto a cupolet as a result of the interaction between the neurons. This process repeats down the chain until the last neuron, Neuron 4, stabilizes onto a cupolet. Neuron 1 communicates to Neuron

2, Neuron 2 communicates to Neuron 3, and then Neuron 3 communicates to Neuron 4. The last neuron, Neuron 4, communicates back to Neuron 3 through  $IF(Q,\kappa)$ , and the output of  $IF(Q,\kappa)$  replaces the communication that Neuron 3 receives from Neuron 2 (this represents the passing of the impulse that traveled up the nerve). The communication from Neuron 3 to Neuron 4 is unchanged. At this point, Neuron 3 talks to Neuron 4 and Neuron 4 talks to Neuron 3, but all other communication ceases between other neurons in the chain. The end result is that Neuron 3 and Neuron 4 can transition into a state of mutual stabilization while Neuron 1 and Neuron 2 no longer receive any input and thus return to chaotic neural firing. This process is analogous to the mutual stabilization seen in Figure 4. In that example, external controls were applied to both neurons before bidirectional communication occurred that led to mutual stabilization. The key difference in this example is that the controlling signal propagates down a chain of neurons (Neuron 1 to Neuron 4) before terminating at the bidirectional neurons (Neuron 4 back to Neuron 3), and the external controls end.

The goal of the network is to propagate the signal to the neural structure at the end of the chain that is represented by Neuron 3 and Neuron 4. The neurons only have unidirectional communication that is mediated by  $IF(Q,\kappa)$  until each neuron has stabilized onto a cupolet. Then, the last two neurons have bidirectional communication that results in mutual stabilization and the first two neurons no longer receive inputs and return to chaotic neural firing. Figure 6 shows the resulting phase portrait of each neuron for a specific example of chain mutual stabilization. The rows of Figure 6 are snapshots in time and the columns of Figure 6 represent each neuron. While moving down the figure, each row contains new interactions within the network. In row 1, Neuron 1 receives an external binary control input that represents an external stimulus. Neuron 1 receives the control sequence 11 (repeated cyclically) that results in Neuron 1 stabilizing onto cupolet C11B (top row, first column). Neuron 1 then begins communication with Neuron 2 via  $IF(Q,\kappa)$  where  $Q = 4$  and  $\kappa = 3$ . This also results in controls of 11 (repeated cyclically) for Neuron 2. Neuron 3 and Neuron 4 continue exhibiting chaotic dynamics and have not yet received any external input. Neuron 2 stabilizes onto a cupolet, C11A (second row, second column), and begins communication with Neuron 3 via  $IF(5,3)$  in row 3. The process continues and Neuron 3 stabilizes onto a cupolet, C11A (third row, third column), and starts to communicate with Neuron 4 via  $IF(5,3)$ . In row 4, the signal continues to propagate and Neuron 4 stabilizes onto a cupolet, C11A (fourth row, fourth column). This results in Neuron 4 stabilizing onto a cupolet, and all other neurons have continued the cupolet dynamics. In the last row, row 5, the only interactions that occur are Neuron 3 communicating with Neuron 4 and Neuron 4 communicating back to Neuron 3. The input to Neuron 1 and the interneuron communication ends between Neuron 1 to Neuron 2 and Neuron 2 to Neuron 3. Due to the bidirectional communication between Neuron 3 and Neuron 4, these two final neurons lock into persistent, periodic neural firing and therefore these neurons have mutually stabilized. This can be seen in the last row where Neuron 3 continues to evolve as cupolet C11A and Neuron 4 continues to evolve as cupolet C11A, showing homologous cupolets in mutual stabilization. Both Neuron 1 and Neuron 2 revert back to chaotic neural firing because there is no input and the chaotic behavior is the default state of each neuron. The simulation lasts for 100,000 time units, or 12,800,000 iterations. The first row is the snapshot of the simulation from time units 10,000 to 20,000. Each subsequent row displays the corresponding snapshot 20,000 time units later from the previous row. That is, row 2 displays time units 30,000 to 40,000, row 3 displays time units 50,000 to 60,000, and so on. This length of time is chosen so that each cupolet can be clearly seen and allows the same simulation time for the formation of the cupolets as was discussed previously.



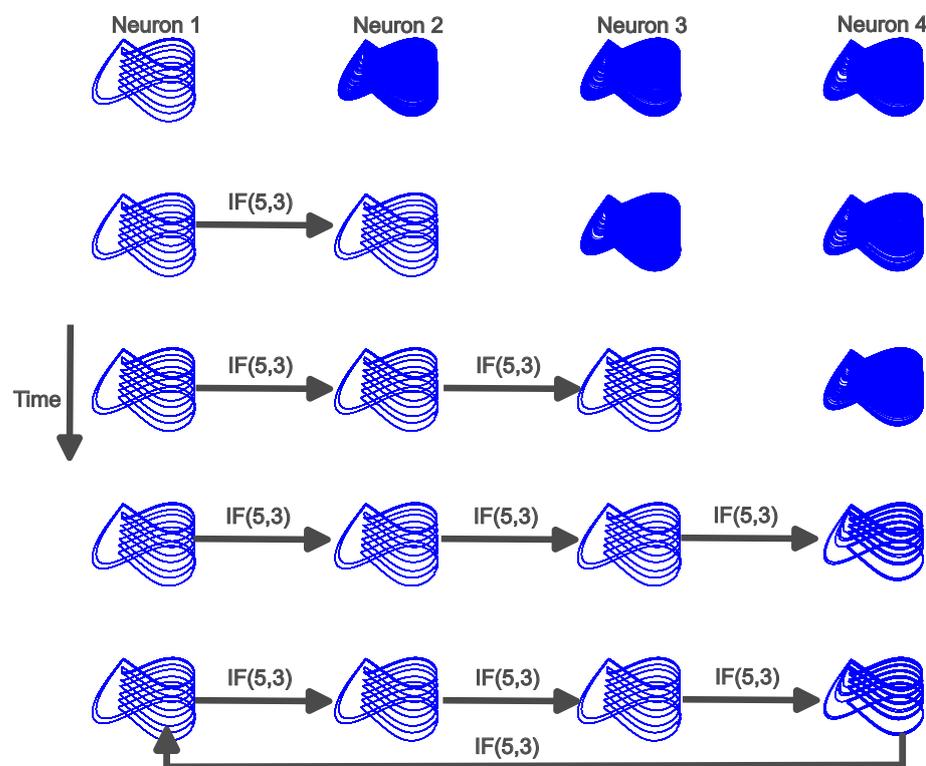
**Figure 6.** Phase plots of four neurons in a chain that results in mutual stabilization between the last two neurons with the interaction governed by  $IF(Q = 5, \kappa = 3)$ . Each column represents a neuron, while the rows are snapshots in time. In each row, the impact of the interactions moves to the right. In the first row, Neuron 1 has stabilized onto cupolet C11B and the other three neurons exhibit chaotic neural firing. The second row has Neuron 1 continuing as cupolet C11B, but communication has started with Neuron 2. Neuron 2 stabilizes onto cupolet C11A. In the third row, Neuron 1 and Neuron 2 continue the same interactions that resulted in the respective cupolets C11B and C11A. Neuron 2 communicates to Neuron 3, which results in Neuron 3 stabilizing onto cupolet C11A. In the fourth row, Neuron 3 communicates to Neuron 4, which results in Neuron 4 stabilizing onto cupolet C11A. In the final row, Neuron 3 and Neuron 4 mutually stabilize because of the bidirectional communication. Neuron 1 and Neuron 2 no longer receive any input and return to chaotic neural firing since that is the default state of each neuron.

### 3.2.2. Feedback Stabilization

The second network discussed here is a loop of four neurons. Similarly to the first case, a signal propagates down a chain of four neurons and the neural communication is mediated by  $IF(Q, \kappa)$  that subsequently stabilizes each respective neuron onto a cupolet in a manner similar to that described in Figure 6. After Neuron 4 stabilizes onto a cupolet, Neuron 4 communicates back to Neuron 1, where the communication is mediated by  $IF(Q, \kappa)$ . The output of  $IF(Q, \kappa)$  replaces the binary external control sequence that originally stabilized Neuron 1 onto a cupolet. All other communication between the neurons remains the same. This can result in each neuron in the loop stabilizing onto a cupolet, and we call this feedback stabilization. This is similar to the mutual stabilization case from the previous section except the entire network locks into persistent, periodic neural firing instead of just two neurons that are bidirectionally communicating. However, if any one of the neurons failed to receive the controlling stimulus, then that neuron would revert back to chaotic neural firing since that is the default state of each neuron.

As in Figure 6, the rows of Figure 7 are snapshots in time and the columns of Figure 7 represent each neuron. In each row, a new interaction has occurred. In row 1, Neuron 1 receives an external binary control input which results in Neuron 1 stabilizing onto a cupolet, C11A, while the other neurons continue to exhibit chaotic firing. In row 2, Neuron 1

communicates with Neuron 2 via  $IF(Q = 5, \kappa = 3)$  and Neuron 2 stabilizes onto cupolet C11A. In row 3, Neuron 1 continues communication with Neuron 2 via  $IF(5,3)$  and Neuron 2 begins communication with Neuron 3 via  $IF(5,3)$ , which results in Neuron 3 stabilizing onto cupolet C11A. Then, in row 4, Neuron 4 receives input from Neuron 3 through  $IF(5,3)$  and Neuron 4 stabilizes onto C11B. In the last row, row 5, Neuron 4 feeds back to Neuron 1 via  $IF(5,3)$  by replacing the external binary sequence with the output of  $IF(5,3)$ . Neuron 1 remains stabilized onto cupolet C11A with the feedback from Neuron 4. All other neural communication remains the same. The network has now achieved feedback stabilization, where every neuron is locked into persistent, periodic dynamics that will continue as long as the interaction persists. The simulation lasts for 100,000 time units, or 12,800,000 iterations. The first row is the snapshot of the simulation from time units 10,000 to 20,000. Each subsequent row displays the corresponding snapshot 20,000 time units later from the previous row, so row 2 displays time units 30,000 to 40,000, row 3 displays time units 50,000 to 60,000, and so on.



**Figure 7.** Phase plots of four neurons that result in feedback stabilization with the interaction governed by  $IF(Q = 5, \kappa = 3)$ . Each column represents a neuron while the rows are phase space plots corresponding to long snapshots in time. In the first row, Neuron 1 has stabilized onto cupolet C11A and the other three neurons exhibit uncontrolled, chaotic neural firing. The second row shows Neuron 1 stabilized onto C11A but communication begins to Neuron 2, which results in Neuron 2 stabilizing onto cupolet C11A. In the third row, Neuron 1 and Neuron 2 continue the same interactions and Neuron 2 begins communication with Neuron 3 which results in Neuron 3 stabilizing onto cupolet C11A. In the fourth row, Neuron 3 communicates to Neuron 4 which results in Neuron 4 stabilizing onto cupolet C11B. In the final row, Neuron 4 communicates back to Neuron 1 and Neuron 1 remains stabilized onto cupolet C11A. The network locks into feedback stabilization.

#### 4. Discussion

In previous work, Parker and Short [4] focused on demonstrating how mutual stabilization can occur between two FitzHugh–Nagumo neurons. That result was motivated by mutual stabilization of communicating cupolets of the double scroll oscillator, seen in Morena and Short [11], but the stabilization was achieved through neural learning that

changed the synaptic strength and led the system out of a chaotic regime. The mutual stabilization that resulted from the interacting FHN neurons allowed for a proof-of-concept for mutual stabilization in neural systems. In Parker and Short [9] we demonstrated how cupolets can be generated by following an approach equivalent to that used in previous cupolet research generated from the double scroll oscillator, where the *Integrate-And-Fire* interaction function was employed.

The work presented here enabled communication between neurons to be mediated by the *Integrate-And-Fire* interaction function from Morena and Short [11], adapted to be instantaneous in this work. This interaction function, abbreviated as  $IF(Q,\kappa)$ , receives the symbolic dynamics of a signal (length  $Q$ ) and returns a control bit of 1 if the symbolic dynamics sum to  $\kappa$  or 0 if not. Biologically, neurons receive input, commonly called a stimulus, and if the stimulus changes the membrane potential such that the membrane potential reaches a certain threshold, an action potential is released. The action potential corresponds to the spikes that occur in mathematical models of neurons. Therefore, in the approach presented here, we can consider that when the interaction function produces a 1, corresponding to a macrocontrol, this is analogous to the neuron receiving an action potential. Similarly, the interaction function  $IF(Q,\kappa)$  can be considered as an abstraction of the synaptic behavior that produces a stimulus received by a neuron. If the receiving neuron is stimulated in a manner that can produce a cupolet, then the neuron's dynamics shift towards a specific periodic orbit, forming a cupolet that is persistent as long as the communication continues. This is evident from Figure 4. However, if the control, acting as the stimulus, is not appropriate, then the receiving neuron will continue to evolve in a chaotic manner, as can be seen in Figure 5.

In Section 3.2.1, a control sequence stabilizes a chaotic neuron onto a cupolet and information of the cupolet propagates down the chain via the interaction function  $IF(Q,\kappa)$  until terminating at the final two neurons. The final two neurons could potentially mutually stabilize if the right communication signal is produced, as mediated via an interaction function. The chain mutual stabilization network, seen in Section 3.2.1, extended the mutual stabilization seen in Section 3.1 to demonstrate how mutual stabilization can occur from a signal coming down a chain of neurons which represents a nerve. The signal propagates along the neural pathway until reaching a neural structure that is represented here by the two final neurons in this network (Neurons 3 and 4). This structure then retains information about the signal by locking into a mutually stabilized state that persists as long as the interaction between the neurons continues. The neurons further down the chain no longer need to pass information since it is assumed that the stimulus has subsided after passing through Neurons 1 and 2. These neurons stop all interaction within the system and return to chaotic neural firing, ready to receive another signal and transmit information to the neural structure at the end of the nerve.

A second interesting case we address is feedback stabilization (Section 3.2.2), where a larger chain of unidirectional neurons is considered but the last neuron in the chain communicates back with the first neuron. The feedback between the last neuron and the first neuron can lock the entire loop of neurons into a stabilized state. The feedback stabilization network represents a larger structure that locks into a stabilized state. Once the last neuron in the chain feeds back to the first neuron, it replaces the original stimulus on the first neuron that triggered the chain of stabilization. The entire network locks into a stabilized, persistent, periodic state that would not exist without the feedback interaction between the neurons. There is only unidirectional communication between any pair of neurons, yet each neuron becomes stabilized because of the feedback from Neuron 4 to Neuron 1. Neuron 1, Neuron 2, and Neuron 3 stabilized onto cupolet C11A while Neuron 4 stabilized onto cupolet C11B. It should be noted that both homologs of cupolet C11 are involved, yet the specific homolog that stabilizes (C11A or C11B) is dependent on the state of the neuron at the instant that the controls are applied. Equivalent stabilizations would occur with different combinations of C11A and C11B. The network locks into feedback

stabilization, yet if the time series of the cupolets is examined, there is a phase difference in the firing between any pair of neurons.

The interaction between the neurons in each of the two networks discussed in this paper was mediated via IF(5,3). The initial control sequence on Neuron 1 was the same in each network, with control 11 leading to homologous C11 cupolets. By keeping the same initial cupolet and the same interaction function between neurons, we are able to show that the resultant stable dynamics of the network was dependent on the neural connections within the network. In the absence of these connections, each neuron would have no input and would exhibit chaotic dynamics since that is the default state of each neuron. The networks discussed in this paper are among the simplest types of networks that can exhibit a form of chaotic stabilization. This provides a foundation for building larger neural networks to explore further more complicated network dynamics.

A limitation of this work is that our implementation of the interaction function provided instantaneous communication between neurons, rather than allowing for some delay, as was incorporated in Morena and Short [11]. Further, the results presented here involved dynamical models simulated through Runge–Kutta methods, whereas the work in [11] was more of an information-theoretic existence proof. Even with this restriction, we did find mutual stabilization in the networks discussed, yet all of the stabilized neurons were homologs of C11. This is primarily a result of the early state of this research, since synchronization of two systems is simplest when a stream of macrocontrols is applied. All other cupolets that have been found have at least one 0 in the control sequence and that 0 control will need to be applied in the feedback to the first system “in sync” with the required application of a microcontrol. However, in the preliminary work presented here, the communication between units occurred instantaneously and more work will be needed to allow for variability in the feedback timing to aid in mutual stabilization among other cupolets. However, if the available cupolets are examined in a manner similar to Morena and Short [11], it is possible that many more potential candidates for mutual stabilization will be found. As a simple example, cupolet C001, seen in Figure 4A, has a visitation sequence of 110 repeatedly. An interaction function of IF(2,2) would produce 100100100... and so on, to be used as a control on a second system. Since this control is simply 001 cycled and then repeated, appropriate timing of the controls would lead the second system to stabilize onto C001 as well. Then, the communication can be fed back through to the original system that is also on C001, such that the two systems are locked into a mutual stabilization at the information-theoretic level. Future work will address implementing this example in the simulation model using Runge–Kutta, where timing of the feedback will play a crucial role. Future work will also perform an exhaustive search to find other cupolets and interaction functions that can lead to mutual stabilization.

## 5. Conclusions

This paper focused on demonstrating that chaotic stabilization can occur in a variety of network topologies, which all contain neurons existing in a chaotic parameter regime. The units communicate instantaneously through the control scheme that drives transitions from chaotic firing to periodic firing. The existence of this transition implies that this communication scheme may be pertinent for eliciting certain periodic, regular, firing patterns in structures comprising two or more neurons. Therefore, individual biological neurons may naturally exist in a chaotic state, but this paper shows that the interactions can regularize the dynamics. These results show that communication between the neurons determines whether mutual stabilization may occur or the neurons may continue to evolve in a chaotic regime. This suggests another potential role for the existence of chaos in neurons.

To place this in a broader context, Rabinovich and Abarbanel [36] cite central pattern generators, specifically lobster central pattern generators, as a prime example of the role of chaos in neurons. As the authors discussed, central pattern generators are crucial for producing rhythmic dynamics, yet the individual units that comprise the central pattern generators are often chaotic. In this regard, one interpretation of the mutual stabilization

presented here is similar to a central pattern generator where the collective network produces periodic or rhythmic dynamics when connected, yet the individual neurons produce chaotic dynamics when there are no interactions. Another example of a role for chaos in neurons was suggested in Parker and Short [4], where chaotic stabilization through interaction between two FitzHugh–Nagumo models occurred. That example suggested that the stabilization or locking of neurons into a specific state may be related to a form of “stored memory”. That is, when the system receives a specific stimulus, which leads to mutual stabilization, it retains information about the original signal that is reflected by the specific persistent, periodic orbit. The work presented in the present paper extends these ideas further, and demonstrates that it is possible for control information to become locked into neurons as a form of memory, even when the systems remain in a chaotic parameter regime. Future work will focus on generalizing and strengthening these results.

**Author Contributions:** Conceptualization, J.E.P. and K.M.S.; data curation, J.E.P.; formal analysis, J.E.P. and K.M.S.; investigation, J.E.P. and K.M.S.; methodology, J.E.P. and K.M.S.; software, J.E.P.; supervision, K.M.S.; validation, J.E.P. and K.M.S.; visualization, J.E.P.; writing—original draft, J.E.P.; writing—review and editing, J.E.P. and K.M.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** The data presented in this study will be openly available in the GitHub repository [https://github.com/jparker25/Parker\\_Short\\_2023\\_mutual\\_stabilization](https://github.com/jparker25/Parker_Short_2023_mutual_stabilization) (accessed on 21 April 2023) upon publication.

**Acknowledgments:** We would like to thank AIP Publishing for allowing us to reuse several figures. We would also like to thank Madeline Edwards, currently at the University of Pittsburgh, for providing the drawing in Figure 1. Lastly, we would like to thank the reviewers for providing constructive feedback to improve the presentation of this work.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

HR	Hindmarsh–Rose
IF	Integrate-And-Fire
PS0	Poincaré Surface 0
PS1	Poincaré Surface 1

## References

1. Fitzhugh, R. Thresholds and Plateaus in the Hodgkin-Huxley Nerve Equations. *J. Gen. Physiol.* **1960**, *43*, 867–896. [[CrossRef](#)] [[PubMed](#)]
2. FitzHugh, R. Impulses and Physiological States in Theoretical Models of Nerve Membrane. *Biophys. J.* **1961**, *1*, 445–466. [[CrossRef](#)] [[PubMed](#)]
3. Nagumo, J.; Arimoto, S.; Yoshizawa, S. An Active Pulse Transmission Line Simulating Nerve Axon. *Proc. IRE* **1962**, *50*, 2061–2070. [[CrossRef](#)]
4. Parker, J.E.; Short, K.M. Sigmoidal synaptic learning produces mutual stabilization in chaotic FitzHugh–Nagumo model. *Chaos Interdiscip. J. Nonlinear Sci.* **2020**, *30*, 063108. [[CrossRef](#)]
5. Parker, J.E. Existence of Mutual Stabilization in Chaotic Neural Models. Ph.D. Thesis, University of New Hampshire, Durham, NH, USA, 2021.
6. Hebb, D.O. *The Organization of Behavior: A Neuropsychological Theory*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 1950; Volume 63, p. 633. [[CrossRef](#)]
7. Oja, E. Simplified neuron model as a principal component analyzer. *J. Math. Biol.* **1982**, *15*, 267–273. [[CrossRef](#)]
8. Hindmarsh, J.L.; Rose, R.M. A Model of Neuronal Bursting Using Three Coupled First Order Differential Equations. *Proc. R. Soc. Lond. Ser. B Biol. Sci.* **1984**, *221*, 87–102. [[CrossRef](#)]
9. Parker, J.E.; Short, K.M. Cupolets in a chaotic neuron model. *Chaos Interdiscip. J. Nonlinear Sci.* **2022**, *32*, 113104. [[CrossRef](#)]
10. Morena, M.A.; Short, K.M. Cupolets and a Chaotic Analog of Entanglement. *arXiv* **2013**, arXiv:1302.2283.

11. Morena, M.A.; Short, K.M. On the Potential for Entangled States Between Chaotic Systems. *Int. J. Bifurc. Chaos* **2014**, *24*, 1450077. [[CrossRef](#)]
12. Parker, A.T. Topics in Chaotic Secure Communication. Ph.D. Thesis, University of New Hampshire, Durham, NH, USA, 1999.
13. Hayes, S.; Grebogi, C.; Ott, E. Communicating with chaos. *Phys. Rev. Lett.* **1993**, *70*, 3031–3034. [[CrossRef](#)]
14. Hayes, S.; Grebogi, C.; Ott, E.; Mark, A. Experimental Control of Chaos for Communication. *Phys. Rev. Lett.* **1994**, *73*, 1781–1784. [[CrossRef](#)] [[PubMed](#)]
15. Matsumoto, T.; Chua, L.O.; Komuro, M. The double scroll. *IEEE Trans. Circuits Syst.* **1985**, *32*, 797–818. [[CrossRef](#)]
16. Lorenz, E.N. Deterministic Nonperiodic Flow. *J. Atmos. Sci.* **1963**, *20*, 130–141. [[CrossRef](#)]
17. Rössler, O. An equation for continuous chaos. *Phys. Lett. A* **1976**, *57*, 397–398. [[CrossRef](#)]
18. Zarringhalam, K. CUPOLETS: Chaotic Unstable Periodic Orbits Theory and Applications. Ph.D. Thesis, University of New Hampshire, Durham, NH, USA, 2006.
19. Morena, M.A.; Short, K.M.; Cooke, E.E. Controlled transitions between cupolets of chaotic systems. *Chaos Interdiscip. J. Nonlinear Sci.* **2014**, *24*, 013111. [[CrossRef](#)]
20. Morena, M.A.; Short, K.M. Fundamental cupolets of chaotic systems. *Chaos Interdiscip. J. Nonlinear Sci.* **2020**, *30*, 093114. [[CrossRef](#)]
21. Short, K.M.; Morena, M.A. Signatures of Quantum Mechanics in Chaotic Systems. *Entropy* **2019**, *21*, 618. [[CrossRef](#)]
22. Lapique, L. Recherches quantitatives sur l'excitation électrique des nerfs traitée comme une polarisation. *J. Physiol. Pathol. Génér.* **1907**, *9*, 620–635.
23. Innocenti, G.; Morelli, A.; Genesio, R.; Torcini, A. Dynamical phases of the Hindmarsh-Rose neuronal model: Studies of the transition from bursting to spiking chaos. *Chaos Interdiscip. J. Nonlinear Sci.* **2007**, *17*, 043128. [[CrossRef](#)]
24. Innocenti, G.; Genesio, R. On the dynamics of chaotic spiking-bursting transition in the Hindmarsh–Rose neuron. *Chaos Interdiscip. J. Nonlinear Sci.* **2009**, *19*, 023124. [[CrossRef](#)]
25. Doungmo Goufo, E.F.; Tabi, C.B. On the chaotic pole of attraction for Hindmarsh-Rose neuron dynamics with external current input. *Chaos Interdiscip. J. Nonlinear Sci.* **2019**, *29*, 023104. [[CrossRef](#)]
26. Wang, X.J. Genesis of bursting oscillations in the Hindmarsh-Rose model and homoclinicity to a chaotic saddle. *Phys. D Nonlinear Phenom.* **1993**, *62*, 263–274. [[CrossRef](#)]
27. Storace, M.; Linaro, D.; de Lange, E. The Hindmarsh–Rose neuron model: Bifurcation analysis and piecewise-linear approximations. *Chaos Interdiscip. J. Nonlinear Sci.* **2008**, *18*, 033128. [[CrossRef](#)]
28. Jun, M.; Qing-Yun, W.; Wu-Yin, J.; Ya-Feng, X. Control Chaos in Hindmarsh–Rose Neuron by Using Intermittent Feedback with One Variable. *Chin. Phys. Lett.* **2008**, *25*, 3582–3585. [[CrossRef](#)]
29. Castanedo-Guerra, I.T.; Steur, E.; Nijmeijer, H. Synchronization of coupled Hindmarsh-Rose neurons: Effects of an exogenous parameter\*\*Financially supported by Mexican National Council for Science and Technology (CONACYT). *IFAC-PapersOnLine* **2016**, *49*, 84–89. [[CrossRef](#)]
30. Wang, H.; Zheng, Y.; Lu, Q. Stability and bifurcation analysis in the coupled HR neurons with delayed synaptic connection. *Nonlinear Dyn.* **2017**, *88*, 2091–2100. [[CrossRef](#)]
31. Branciforte, M.; Buscarino, A.; Fortuna, L. A Hyperneuron Model Towards in Silico Implementation. *Int. J. Bifurc. Chaos* **2022**, *32*. [[CrossRef](#)]
32. Brunel, N.; van Rossum, M.C.W. Quantitative investigations of electrical nerve excitation treated as polarization. *Biol. Cybern.* **2007**, *97*, 341–349. [[CrossRef](#)]
33. Racicot, D.M.; Longtin, A. Interspike interval attractors from chaotically driven neuron models. *Phys. D Nonlinear Phenom.* **1997**, *104*, 184–204. [[CrossRef](#)]
34. Hall, G.M.; Bahar, S.; Gauthier, D.J. Experimental control of a chaotic point process using interspike intervals. *Phys. Rev. E* **1998**, *58*, 1685–1689. [[CrossRef](#)]
35. Morena, M.A. Mutual Stabilization of Chaotic Systems through Entangled Cupolets. Ph.D. Thesis, University of New Hampshire, Durham, NH, USA, 2014.
36. Rabinovich, M.; Abarbanel, H.D. The role of chaos in neural systems. *Neuroscience* **1998**, *87*, 5–14. [[CrossRef](#)]

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.