

Article



Neural-Impulsive Pinning Control for Complex Networks Based on V-Stability

Daniel Ríos-Rivera ¹, Alma Y. Alanis ^{1,*} and Edgar N. Sanchez ^{2,*}

- ¹ Computer Sciences Department, Universidad de Guadalajara, Guadalajara 44430, Mexico; daniel.rios8936@alumnos.udg.mx
- ² Electrical Engineering Department, CINVESTAV, Unidad Guadalajara, Zapopan 45017, Mexico
- * Correspondence: alma.alanis@academicos.udg.mx (A.Y.A.); sanchez@gdl.cinvestav.mx (E.N.S.)

Received: 1 July 2020; Accepted: 5 August 2020; Published: 19 August 2020



Abstract: In this work, a neural impulsive pinning controller for a twenty-node dynamical discrete complex network is presented. The node dynamics of the network are all different types of discrete versions of chaotic attractors of three dimensions. Using the V-stability method, we propose a criterion for selecting nodes to design pinning control, in which only a small fraction of the nodes is locally controlled in order to stabilize the network states at zero. A discrete recurrent high order neural network (RHONN) trained with extended Kalman filter (EKF) is used to identify the dynamics of controlled nodes and synthesize the control law.

Keywords: complex networks; V-stability; pinning control; impulsive control; neural control; chaos control

1. Introduction

In recent years, complex networks have been gaining more attention in the scientific community, with a wide range of applications, from computer science to sociology [1]. From the perspective of dynamical systems, it is possible to see a complex network as a group of interconnected systems, each one with their own dynamics and characteristics. For dynamical systems, it is often desirable to force them to follow a desired behavior, which can be performed with a controller. There are many techniques in control theory that have been developed over the years in order to achieve stability around a given equilibrium point [2]. Using a particular method of control depends on the designer or the type of system that is being worked on. A technique developed for the control of complex dynamical networks is pinning control, which consists of controlling only a small fraction of nodes of a complex network [3]. Numerous studies have been published around this pinning control technique, of particular interest to this study is the one of V-stability [4], in which the stability problem for a complex network with different dynamics in its nodes is transformed to verify positive definiteness of an associated matrix.

Another technique not particular to complex networks is impulsive control, which consists of applying a control input in a non-continuous way and is carried out at rather well-defined time instants [5]. The control and synchronization of complex networks through an impulsive input has been studied previously, as in [6], where they propose a control method for a complex network where all node dynamics are the same, similar to [7]. In [8], different node dynamics are proposed, but no pinning control [8].

On the other side, neural networks have long been studied for use in dynamical systems as they have the capacity to reconstruct nonlinear dynamics [9,10] by only having measurements of the system state, which is the case of the neural identifier; then, the output and the model of this neural network can be used to compute a desired control action.

The control and synchronization of complex networks is an active research area. In [11], V-stability

and inverse optimal control were used to control a complex network with different chaotic node dynamics. We use this approach in order to interpret the problem of stability of the network as a linear algebra problem that makes it easier to decide the node and gain selection. A discrete-time neural-network was used in [12,13] to synchronize a continuous-time dynamical complex network with an inverse optimal control and sliding mode approach respectively. This work also makes use of neural identification for the great advantage that it represents, accounting for unmodeled behavior and disturbances, which is a matter of study in this work. In [14], a neural network was used to control the network along with V-stability analysis to determine the pinned node. Most of these previous works deal with continuous-time complex networks [12–14], but not discrete nor impulsive complex networks. We aim to study discrete-time complex networks for their relevance, as any system governed by a computer processor is essentially discrete. A robotic system network may be a good fit to exemplify this case. For impulsive systems, there are few works that deal with discrete systems. In [15], there is a compilation of distinct stability analysis for impulsive systems like the one in [16], where they show stability conditions for a specific type of discrete-time impulsive systems, which is that the impulse dynamics are linear. This is something that often appears in impulse systems [15]. This is not the case of the study presented here, as will be seen in later sections. Impulsive complex network systems are also important for their potential use in biomedicine. Practical stability for impulsive systems was investigated in [17], where they consider impulsive perturbations and *h*-manifolds.

The focus of this paper is based on V-stability and an experimental approach, where we work together the different control schemes mentioned above in order to control a discrete-time complex network with different dynamics in its nodes, with an impulsive control law that is synthesized with the neural model. The controller presented in this paper has characteristics of the different control schemes that conform it, like the unnecessariness of the dynamic model, reducing process cost, and reducing the number of local controllers in the complex network.

The remains of this paper is organized as follows. First, complex networks and their dynamical model are briefly described. Then, the concept of V-stability and the process of selecting nodes to control is described. After that, the parameters used in simulations are introduced. Later, impulsive systems are briefly explained, and we present the model used for the impulsive control simulations. Then, recurrent high order neural networks and the identifier used in this paper are discussed. After that, the results are presented and discussed. Lastly, a conclusion of this work is presented.

2. Some Complex Network Theory

Complex networks can be described as a set of N nodes in which any pair of nodes in the set may be connected by a link. This connection between nodes may be directed or undirected and can also be weighted. The number of connections a node has is called the degree and is often denoted by the letter k. This letter k is of much importance to complex networks since one can assume several properties based on the distribution of node degrees in the network [18]. Degree distribution in form of a power law, for example, is called a scale-free network, and this kind of network shows a few nodes with a large number of connections and a majority of nodes with a small number of these connections [19].

Nodes of a complex network and their connections can have different meanings [20]. Our interest here is to consider the nodes as dynamical systems and the connection between them as communication between the different node dynamics that together form a bigger and more complex system.

Consider then, the next dynamical model of a complex network with *N* nodes and linear and diffusive connections [11].

$$\dot{x}_i = f_i(x_i) + \sum_{j=1}^N c_{ij} a_{ij} \Gamma x_j, \quad i = 1, 2, \dots, N,$$
 (1)

where $x_i = [x_{i1}, x_{i2}, ..., x_{in}]^{\top}$ is the state vector of node *i*. Function $f_i : \mathbb{R}^n \to \mathbb{R}^n$ represents the self-dynamics of node *i*. Constant c_{ij} is the connection strength between node *i* and node *j*. Matrix $A = [a_{ij}] \in \mathbb{R}^{N \times N}$ is an outer coupling matrix that describes the network topology, with $a_{ij} = a_{ji} = 1$ when there is a connection between node *i* and node *j*, $a_{ij} = a_{ji} = 0$ when there is no connection between node *i* and node *j*, and when i = j, the diagonal elements of matrix *A* are $a_{ii} = -k_i$, where k_i is the degree (number of connections) of node *i*. The symmetry of matrix *A* indicates that the connections are undirected. Matrix $\Gamma \in \mathbb{R}^{n \times n}$ is an inner coupling matrix that describes the connections between different components of the state vector.

There are previous works about how to drive system (1) to stability [1]. A very common method is the pinning control approach, in which only a few nodes of the network are controlled (or even just one node) [1,3], the nodes selected for control is still a matter of research but in [3] it was found that for scale-free networks, selecting the largest nodes (nodes with the largest number of connections) is most useful.

3. V-Stability

The theoretical analysis presented in this paper is based on the theorem of V-stability presented in [4], under this premise we must find a common Lyapunov function V(x) for each node dynamics that satisfies:

$$\frac{\partial V(x_i)}{\partial x_i} (f_i(x_i) - \theta_i \Gamma(\overline{x} - x_i)) < 0, \tag{2}$$

for all $x_i \in D$, where *D* is equal to the union of all $D_i = \{x_i : ||x_i - \overline{x}|| < \alpha, \alpha > 0\}$, \overline{x} is the equilibrium point. As long as this condition holds for the nodes' self-dynamics, then the V-stability methodology can be applied to any given network [4]. The constant θ_i is called the passivity degree and is not unique, the theoretical passivity degree is defined to be the largest one, but in practice, any value that satisfies the inequality may be chosen [4]. The physical meaning of this method is that there is an energy-like function that is common to all nodes, such that the energy of said function is decreasing along with the evolution of node dynamics. A positive passivity degree means that the node is stable and that it can provide some energy to stabilize the network. A negative passivity degree, on the other hand, means that the node needs energy from the outside to be stable [4]. This method depends greatly on the function V(x) and constants θ_i , that is why it is called V-stability.

If we use a quadratic Lyapunov function of the form $V(x) = x^{\top}Qx$, where Q is a symmetric and positive definite matrix, then we can change the original stability problem to check if the following inequality is true:

$$C_o = -\Theta + G \le 0, \tag{3}$$

where matrix Θ is a diagonal matrix with passivity degrees θ_i as its elements, and matrix $G = \lfloor g_{ij} \rfloor = \lfloor c_{ij}a_{ij} \rfloor$. The network (1) is V-stable if (3) is true.

For an unstable network, we can stabilize it with a control action. As mentioned before, with pinning control we can control a network by just controlling a small fraction of the nodes which is more practical than controlling every node of the network. For this purpose, we check the eigenvalues of matrix C_o , and if has *m* non-negative eigenvalues then we need at least *m* nodes selected for control [4]. The characteristic matrix of a controlled network is:

$$C = -\theta + G - K,\tag{4}$$

where $K \in \mathbb{R}^{N \times N}$ is a diagonal matrix with *m* elements d_i and the remaining elements equal to zero. Elements d_i must be chosen such that characteristic matrix *C* is negative definite. The values selected can be seen as the feedback gain in the control input u(x) of the form:

$$u(x_i) = -d_i x_i,\tag{5}$$

for all the nodes selected for control.

4. Simulation Preliminaries

Here, we introduce the 20-node network, with some scale-free characteristics, used for the experiments shown in this paper. This network is illustrated in Figure 1.



Figure 1. Twenty-node network used in the experiments.

For coding purposes, we consider a discrete version of the model (1) with a control input for the nodes selected for control:

$$x_i(k+1) = x(k) + T\left(f_i(x_i(k)) + \sum_{j=1}^N c_{ij}a_{ij}\Gamma x_j(k) + u_i(x_i(k))\right),$$
(6)

where *T* is the sampling time. Parameters of the network are:

$$A = \begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix},\tag{7}$$

where

Connection strength c_{ij} is equal to all nodes and is chosen as:

$$c_{ii} = c = 100.$$
 (8)

Matrix Γ is an identity matrix of the order of the state, which is 3:

$$\Gamma = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$
 (9)

Control input is:

$$u_i(x_i(k)) = -d_i x_i(k),$$
 (10)

with constant $d_i > 0$ and only when node *i* is selected control, if not, the control input is zero. Node self-dynamics are determined by the equations of the following chaotic attractors:

$$f_i(x_i(k)) = \begin{bmatrix} 10(x_{i2}(k) - x_{i1}(k)) \\ 28x_{i1}(k) - x_{i1}(k)x_{i3}(k) - x_{i2}(k) \\ x_{i1}(k)x_{i2}(k) - bx_{i3}(k) \end{bmatrix}, \quad i = 1, 6, 11, 16,$$
(11)

which is the Lorenz system with its parameters as they appear in [21].

$$f_i(x_i(k)) = \begin{bmatrix} 35(x_{i2}(k) - x_{i1}(k)) \\ -7x_{i1}(k) - x_{i1}(k)x_{i3}(k) + 28x_{i2}(k) \\ x_{i1}(k)x_{i2}(k) - 3x_{i3}(k) \end{bmatrix}, \quad i = 2, 7, 12, 17,$$
(12)

which is the Chen system with its parameters as they appear in [22].

$$f_i(x_i(k)) = \begin{bmatrix} 36(x_{i2}(k) - x_{i1}(k)) \\ 15x_{i2}(k) - x_{i1}(k)x_{i3}(k) \\ x_{i1}(k)x_{i2}(k) - 3x_{i3}(k) \end{bmatrix}, \quad i = 3, 8, 13, 18,$$
(13)

which is the Lü system with its parameters as they appear in [23].

$$f_i(x_i(k)) = \begin{bmatrix} 10(x_{i2}(k) - x_{i1}(k)) + x_{i2}(k)x_{i3}(k) \\ 25x_{i1}(k) - x_{i1}(k)x_{i3}(k) - x_{i2}(k) \\ x_{i1}(k)x_{i3}(k) - 7x_{i3}(k) \end{bmatrix}, \quad i = 4, 9, 14, 19,$$
(14)

which is the Qi system with its parameters as they appear in [24].

$$f_i(x_i(k)) = \begin{bmatrix} 9.35(x_{i2}(k) - h(x_{i1}(k))) \\ x_{i1}(k) - x_{i2}(k) + x_{i3}(k) \\ -14.79x_{i3}(k) \end{bmatrix}, \quad i = 5, 10, 15, 20,$$
(15)

which is the Chua system with its parameters and:

$$h(x_{i1}(k)) = \frac{2}{7}x_{i1}(k) - \frac{3}{14}(|x_{i1}(k) + 1| - |x_{i1}(k) - 1|),$$
(16)

as they appear in [25].

We first make the continuous analysis of V-stability and then we apply the results of this analysis to the discrete network (6). First, we obtain the passivity degree θ_i for each node using the function $V(x_i) = 0.5x_i^{T}x_i$, the equilibrium point $\overline{x} = 0$, and inequality (2) obtaining.

$$\theta_i < \frac{-38x_{i1}x_{i2} + 10x_{i1}^2 + x_{i1}^2 + \frac{8}{3}x_{i3}^2}{x_{i1}^2 + x_{i2}^2 + x_{i3}^2},$$
(17)

for the Lorenz system,

$$\theta_i < \frac{28x_{i1}x_{i2} + 35x_{i1}^2 - 28x_{i1}^2 + 3x_{i3}^2}{x_{i1}^2 + x_{i2}^2 + x_{i3}^2},$$
(18)

for the Chen system,

$$\theta_i < \frac{-36x_{i1}x_{i2} + 36x_{i1}^2 - 15x_{i1}^2 + 3x_{i3}^2}{x_{i1}^2 + x_{i2}^2 + x_{i3}^2},$$
(19)

for the Lü system,

$$\theta_i < \frac{-38x_{i1}x_{i2} + 10x_{i1}^2 - x_{i1}x_{i2}x_{i3} + x_{i1}^2 + 7x_{i3}^2}{x_{i1}^2 + x_{i2}^2 + x_{i3}^2},$$
(20)

for the Qi system, and

$$\theta_i < \frac{-10.35x_{i1}x_{i2} + 9.35h(x_{i1}) + x_{i1}^2 - x_{i2}x_{i3} + 14 - 79x_{i3}^2}{x_{i1}^2 + x_{i2}^2 + x_{i3}^2},$$
(21)

for the Chua system.

It can be seen that there is no clear value for the passivity degrees, it all depends on the set *D* of interest. Defining an arbitrary set, we obtain the next rough set of values for the passivity degrees shown in Table 1.

Table 1. Passivity degree values for the different chaotic systems.

Chaotic System	Passivity Degree θ_i	
Lorenz	-13.6	
Chen	-28.1	
Lü	-15.1	
Qi	-22.4	
Chua	-5.1	

With these values, we can now construct the diagonal matrix Θ and with this, also matrix C_o and we can obtain its eigenvalues. Only one of the eigenvalues results non-negative, meaning that we need at least one controlled node. We thus construct matrix K with only one non-zero element, we select the element d_3 to be non-zero so we control node 3, based on the number of its connections. If we select $d_3 = 500$ and check for the eigenvalues of C, we can verify that the matrix is now negative definite which means that the network is now stable. A MATLAB[®] simulation of the resulting dynamics using sample time T = 0.001 can be seen in Figure 2.



Figure 2. Node dynamics of the 20-node network with a single controller. From time 0 to 2 there are no connections between the nodes, after that, nodes are connected, at time = 4, control input is applied.

The concept of V-stability refers to a characterization of node stability. The passivity degree expresses how stable or unstable the node in question is. In case the system is stable, we could verify the gain in which the system becomes unstable (in case of such gain exists). In the unstable case, we search for the gain in which the system becomes stable, like we did in this section. The advantage of this characterization is that it defines the stability of a given network with the negativity of matrix (3) (or (4) when it is in a closed loop), this facilitates the node selection by establishing the number of non-negative eigenvalues which is the minimum number of pinned nodes to control.

5. Impulsive Systems

Our goal is not to verify the V-stability method, which gives an interesting result on how many nodes to control, but to apply it while using a neural network as an identifier in order to synthesize control law for the controlled node state, and also verify how it works under and impulsive control scheme and noise conditions.

A time-dependent impulsive system in simple form can be written as [5]:

$$\dot{x}(t) = f_c(x(t)), \quad t \neq t_k,$$

$$\Delta x(t) = f_d(x(t)), \quad t = t_k.$$
(22)

This means that there is a certain dynamic in the system given by $f_c(\cdot)$, but when time gets to t_k , it changes to $f_d(\cdot)$ just for that time instant.

Extrapolating this to our complex network discrete model (6), we can write our system as:

$$x_{i}(k+1) = x(k) + T\left(f_{i}(x_{i}(k)) + \sum_{j=1}^{N} c_{ij}a_{ij}\Gamma x_{j}(k)\right) \qquad Tk \neq t_{k}$$

$$x_{i}(k+1) = x(k) + T\left(f_{i}(x_{i}(k)) + \sum_{j=1}^{N} c_{ij}a_{ij}\Gamma x_{j}(k) + u_{i}(x_{i}(k))\right) \qquad Tk = t_{k}$$
(23)

and $u_i = 0$ if node *i* was not selected for control.

We see that the only thing that changes in the pulse dynamics (when $t = t_k$) is that there is a control input that only appears at time t_k . We define t_k as being every 5T and every 10T, this means that control input appears every five times the sample time and every ten times the sample time, whit sample time T = 0.001, for all simulations.

We cannot be sure the V-stability method works with something like this—that is where the experimentation enters in. Before that, we shall introduce the last concept.

6. Neural Identifier

Artificial neural networks have different applications depending on their configuration [26]. A recurrent high order neural network (RHONN) is good at approximating the behavior of continuous and discrete dynamical systems [9,10].

For a discrete RHONN, we consider the model [10]:

$$x_i(k+1) = w_i^{\mathsf{T}} z_i(x(k), u(k)), \quad i = 1, 2, \dots, n$$
, (24)

where $x_i(k)$ is the state of the *i*th neuron at time step k, n is the dimension of the state, w_i is the adaptable weight vector, and z(x(k), u(k)) is defined as:

-

$$z_{I}(x,u) = \begin{bmatrix} z_{I1} \\ z_{I2} \\ \vdots \\ z_{IL} \end{bmatrix} = \begin{bmatrix} \prod_{j \in I_{1}} y_{j}^{d_{j}(1)} \\ \prod_{j \in I_{2}} y_{j}^{d_{j}(2)} \\ \vdots \\ \prod_{j \in I_{1}} y_{j}^{d_{j}(L)} \end{bmatrix},$$
(25)

where L is the number of high order connections, $\{I_1, I_2, ..., I_L\}$ is a collection of L unordered subsets $\{1, 2, \dots, m + n\}$, and y is a vector formed by the inputs of each neuron defined as:

$$y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \\ y_{n+1} \\ \vdots \\ y_{n+m} \end{bmatrix} = \begin{bmatrix} S(x_1) \\ \vdots \\ S(x_n) \\ u_1 \\ \vdots \\ u_m \end{bmatrix},$$
(26)

vector $u = [u_1, u_2, \dots, u_m]^{\top}$ here is the input vector to the neural network. $S(\cdot)$ is a smooth, monotonically increasing sigmoid function [10,21].

The discrete RHONN can be trained by using the extended Kalman filter (EKF) algorithm. Necessary equations for the EKF algorithm are [10,27]:

$$K(k) = P(k)H^{\top}(k) \left[R(k) + H(k)P(k)H^{\top}(k) \right]^{-1},$$
(27)

$$w(k+1) = w(k) + K(k)[y(k) - \hat{y}(k)],$$
(28)

$$P(k+1) = P(k) - K(k)H(k)P(k) + Q(k),$$
(29)

where $K(k) \in \mathbb{R}^{m \times L}$ is the Kalman gain matrix. $P(k) \in \mathbb{R}^{L \times L}$ represents the error covariance matrix. $H(k) \in \mathbb{R}^{m \times L}$ is the matrix containing the derivatives of each network output with respect to each one of the weights, that is:

$$H_{ij}(k) = \left[\frac{\partial \hat{y}_i(k)}{\partial w_j(k)}\right], \quad i = 1, \dots, m, \quad j = 1, \dots, L,$$
(30)

 $R(k) \in \mathbb{R}^{m \times m}$ is the measurement error covariance matrix. $w \in \mathbb{R}^{L}$ is the weight vector, *L* is the number of weights of the neural network. $y \in \mathbb{R}^{m}$ is the output vector; *m* is the number of outputs, $\hat{y} \in \mathbb{R}^{m}$ is the neural network output, the difference of these two represents the error signal. $Q(k) \in \mathbb{R}^{L \times L}$ is the process error covariance matrix.

The neural identifier used is based on the described model (24) of discrete RHONN and it has the following structure:

$$\begin{aligned} \hat{x}_{i1}(k+1) &= w_{i11}(k)\varphi(x_{i2}(k)) + w_{i21}(k)\varphi(x_{i1}(k)) + w_{i31}(k)\varphi(x_{i2}(k))\varphi(x_{i3}(k)), \\ \hat{x}_{i2}(k+1) &= w_{i12}(k)\varphi(x_{i1}(k)) + w_{i22}(k)\varphi(x_{i1}(k))\varphi(x_{i3}(k)) + w_{i32}(k)\varphi(x_{i2}(k)), \\ \hat{x}_{i3}(k+1) &= w_{i13}(k)\varphi(x_{i1}(k))\varphi(x_{i2}(k)) + w_{i23}(k)\varphi(x_{i3}(k)) + w_{i33}(k)\varphi(x_{i2}(k)), \end{aligned}$$
(31)

where $\varphi(\cdot)$ is the sigmoid function:

$$\varphi(x) = 0.1 \tanh(0.05x).$$
 (32)

The neural network weights $w_{ijl}(k)$, are obtained by training the RHONN with the EKF algorithm. In this case, subscript *j* denotes the weight vector element, subscript *l* denotes the state element to identify, and subscript *i* is the aimed node identify. Weights are computed separately for each state element. Algorithm equations used for training are the ones in (27)–(29), that become:

$$K_{i}(k) = P_{i}(k)H_{ij}^{\mathsf{T}}(k) \Big[R_{i}(k) + H_{ij}(k)P_{i}(k)H_{ij}^{\mathsf{T}}(k)\Big]^{-1},$$
(33)

$$w_i(k+1) = w_i(k) + K_i(k)[x_i(k) - \hat{x}_i(k)],$$
(34)

$$P_i(k+1) = P_i(k) - K_i(k)H_{ij}(k)P_i(k) + Q_{ij}(k),$$
(35)

with initial P(k):

$$P_i(0) = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{bmatrix},$$
(36)

and the rest of the parameters as:

$$H_{i1}^{\mathsf{T}} = \begin{bmatrix} \varphi(x_{i2}(k)) \\ \varphi(x_{i1}(k)) \\ \varphi(x_{i2}(k))\varphi(x_{i3}(k)) \end{bmatrix}, \quad H_{i1}^{\mathsf{T}} = \begin{bmatrix} \varphi(x_{i2}(k)) \\ \varphi(x_{i1}(k)) \\ \varphi(x_{i2}(k))\varphi(x_{i3}(k)) \end{bmatrix}, \quad H_{i1}^{\mathsf{T}} = \begin{bmatrix} \varphi(x_{i2}(k)) \\ \varphi(x_{i1}(k)) \\ \varphi(x_{i2}(k))\varphi(x_{i3}(k)) \end{bmatrix}, \quad (37)$$

$$R_i(k) = R_i = 10,$$
 (38)

$$Q_{i1}(k) = Q_{i2}(k) = Q_{i3} = Q = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.1 \end{bmatrix}.$$
(39)

All these parameters were heuristically determined. Initial weight values are positive random values no bigger than 1.

With this, we can synthesize control law as:

$$u_i = -d_i \hat{x}_i \tag{40}$$

where d_i is estimated by the analysis completed in Section 3 and \hat{x}_i is determined by the network (31).

7. Results

Using the models (6) and (23), with parameters used in Section 4, and neural network parameters mentioned in Section 6, simulations were made in MATLAB[®]. One of the experiments is made with the model (6) as it is, and in the other, random noise is added to the system. We make the simulations for three different control schemes:

- (a) Continuous in discrete time
- (b) Control impulses every 5T
- (c) Control impulses every 10T

Experiments were carried out as follows. All the nodes are unconnected at the beginning of the simulation. At time 2, the nodes connect with each other according to matrix *A*, matrix Γ , and constant *c*, defined in (7), (9), and (8) respectively. At time 4, the synthetized control law appears according to what is observed in (a), (b), and (c). All initial conditions for the node states are random. A general scheme, of the proposed control strategy, is shown in Figure 3.



Figure 3. Control scheme implemented in simulations.

The results of simulations are shown as follows: Figure 4 shows the network dynamics when the system does not have noise for all three control schemes. Figure 5 shows the network dynamics when there is noise in all of the nodes in the network. It is important to note that all of the node self-dynamics are a set of dimensionless chaotic systems.



Figure 4. Complex network node dynamics when there is one controller, there is no noise in the system, control gain $d_3 = 500$, and: (a) Control input is applied at every time step; (b) Control input is applied every 5T; (c) Control input is applied every 10T.



Figure 5. Complex network node dynamics when there is one controller, there is noise in the system, control gain $d_3 = 500$, and: (a) Control input is applied at every time step; (b) Control input is applied every 5T; (c) Control input is applied every 10T.

We can observe that the impulsively controlled systems stabilize, but not as close to zero as the one controlled continuously. If we increase the control gain to $d_3 = 1000$ when the control input is applied every 5T and $d_3 = 1500$ when the control input is applied every 10T we can obtain better results for the case when there is no noise as shown in Figure 6. Doing the same in for the case with noise, we observe in Figure 7 that we can cause more noise in the system.



Figure 6. Complex network node dynamics when there is one controller, there is no noise in the system, and: (a) Control input is applied every 5T with a control gain $d_3 = 1000$; (b) Control input is applied every 10T with a control gain $d_3 = 1500$.



Figure 7. Complex network node dynamics when there one controller, there is noise in the system, and: (a) Control input is applied every 5T with control gain $d_3 = 1000$; (b) Control input is applied every 10T with control gain $d_3 = 1500$.

8. Discussion

As detailed in Section 5, we could not assure that the V-stability method works for the impulsive control schemes, but completing the analysis provides us with a good idea to how many and what nodes to pin, and we also obtain an estimate for the control gain value.

While we achieve stability in zero with the impulsively controlled unnoisy systems, we had to increase the gain to do it. In the case of noise, we observed more noise; therefore, perhaps another solution is more suitable.

In this kind of problem, it is always very tempting to stabilize the network using only one controller for one node. If we go back to the V-stability concept of Section 3, we can control the network described by matrix (7) with at least one pinned node, and we did this by checking the stability of matrix *C* in (4) with a matrix *K* that we designed. We can choose to pin two nodes of the network by making two diagonal elements d_i of *K* different to zero. Hypothetically, if we make $d_3 = d_4 = 200$, that is, we control nodes 3 and 4, we can verify that by doing this, all eigenvalues of *C* are negative, which means that the complex network is stable. Running another simulation of the unnoisy and noisy systems for the different schemes provides us with the results of Figures 8 and 9 respectively.



Figure 8. Complex network node dynamics when there are two controlled nodes, there is no noise in the system, control gains are $d_3 = d_4 = 200$, and: (a) Control input is applied at every time step; (b) Control input is applied every 5T; (c) Control input is applied every 10T.



Figure 9. Complex network node dynamics when there are two controlled nodes, there is noise in the system, control gains are $d_3 = d_4 = 200$, and: (a) Control input is applied at every time step; (b) Control input is applied every 5T; (c) Control input is applied every 10T.



Figure 10. Complex network node dynamics when there are two controlled nodes, there is no noise in the system and: (a) Control input is applied every 5T with control gains $d_3 = d_4 = 700$; (b) Control input is applied every 10T with control gains $d_3 = d_4 = 1200$.



Figure 11. Complex network node dynamics when there are two controlled nodes, there is noise in the system and: (a) Control input is applied every 5T with control gains $d_3 = d_4 = 700$; (b) Control input is applied every 10T with control gains $d_3 = d_4 = 1200$.

Since a large enough gain can provide adequate performance, it can require inputs with a large amount of energy that are not possible to implement, in such case an option is to select more nodes to pin (but not all of the available nodes).

A formal analysis of the V-stability method for discrete and impulsive complex network systems may be a good direction for future work.

The results show that neural network works and adapts well as the same structure was used to identify the dynamics of node 4 and synthesize its control law.

We can reproduce these results with other networks using different parameters (number of nodes, connections, connection strength, etc.) as long as the self-dynamics of the nodes satisfy condition (2) discussed in Section 3, for some passivity degree θ . It is important to note if that we use the same network as before, but with a connection strength reduced to c = 50, constructing matrix C_0 again and checking for its eigenvalues, we still obtain the information that there is only one non-negative

eigenvalue, but setting the same gain $d_3 = 500$ is not sufficient to make matrix *C* negative definite. Figure 12a shows the network node dynamics with a controller with this gain, and we can see that it starts to diverge from zero at the end of the simulation. If we make $d_3 = 800$ matrix *C* will become negative definite, so we use this gain to control the network with a continuous input in discrete time as we see in Figure 12b. Based on the results of Figures 6 and 10, we increase the gain to $d_3 = 1300$ for the case where the control impulses are every 5T, and $d_3 = 1800$ for the case where control impulses are every 10T. These last results are shown in Figure 12c,d respectively. These results were also obtained using the RHONN of past simulations.



Figure 12. Complex network dynamics for the twenty node network with c = 50, there is only one controlled node, there is noise in the system and: (a) Control input is applied at every time step control gain $d_3 = 500$; (b) Control input is applied every time step with control gain $d_3 = 800$; (c) Control input is applied every 5T with control gain $d_3 = 1300$; (d) Control input is applied every 10T with control gain $d_3 = 1800$.

Using a smaller connection strength results in requiring more effort in order to stabilize the network. In fact, if we use a connection strength of c = 1, the analysis tells us that the minimum number of controllers must be 18. If we use the ten-node network described by matrix (41) with node self-dynamics in the same order as past simulations and a connection strength of c = 100, from V-stability analysis on this network, we see that the associated matrix C_o has one non-negative eigenvalue, by pinning the first node, that has seven connections to other nodes, with a gain of $d_1 = 400$ we can make matrix *C* negative definite. As before, we increase this gain to $d_1 = 900$ for the case

×

×

where the control input is applied every 5T and $d_1 = 1400$ when the control input is applied every 10T, these results can be seen in Figure 13. The RHONN used before was also used here.

(c)

Figure 13. Complex network node dynamics of the ten node network when there is one controller, there is no noise in the system and: (a) Control input is applied at every time step with gain $d_1 = -400$; (b) Control input is applied every 5T with gain $d_1 = -900$; (c) Control input is applied every 10T with gain $d_1 = -1400$.

The same analysis was performed to the fifty-node network shown in Figure 14, with a connection strength c = 50, and node self-dynamics appearing in the same order as past networks. V-stability analysis tells us that we need at least one controller for this network, but the bigger the network is, the harder it becomes to determine which node we should select to control. In order to facilitate the

process, we select the three biggest nodes for control (nodes with the largest number of connections), these are 1, 2 and 7. With a gain $d_1 = d_2 = d_7 = 1000$ we can make the associated matrix *C* negative definite. We apply this to the system that is continuous in discrete time and we increase the gain to $d_1 = d_2 = d_7 = 1500$ for the case where the control input is applied every 5T, and $d_1 = d_2 = d_7 = 2000$ for the case where the control input is applied every 5T, and $d_1 = d_2 = d_7 = 2000$ for the system that is polied every 10T. These results are shown in Figure 15. As before, we use the same RHONN structure as in past simulations.



Figure 14. Fifty-node network used in simulations.



Figure 15. Complex network node dynamics of the fifty-node network with three controlled nodes, there is no noise in the system and: (a) Control input is applied at every time step with gains $d_1 = d_2 = d_7 = -1000$; (b) Control input is applied every 5T with gains $d_1 = d_2 = d_7 = -1500$; (c) Control input is applied every 10T with gains $d_1 = d_2 = d_7 = -2000$.

Four additional topologies, illustrated in Figure 16, were simulated. New topologies consist of a five-node network with c = 25, and a 13, 15, and 25 node networks with c = 100. All of the node

self-dynamics appear in the same order as past simulations. V-stability analysis for these topologies indicates that only one node is needed for control in all of them. Gains used are shown in Table 2, where the subscript of *d* indicates the pinned node, gains for the cases where the control input is applied at every time step are determined by the V-stability analysis, gains are increased for the rest of the cases. Starting from where the control input is applied (when time equals 4), to the end of the simulation (when time equals 6), mean square error and variance of all state elements of the network were obtained, and they are also presented in Table 2.



Figure 16. Network topologies used for simulations, consisting of: (**a**) 10 nodes; (**b**) 13 nodes; (**c**) 15 nodes; (**d**) 25 nodes.

Table 2. Passivity degree values for the different chaotic systems.

Topology	Control Scheme	Gain	MSE	Variance
5 nodes	Input applied every time step	$d_3 = 300$	0.1498	0.1465
5 nodes	Input applied every 5T	$d_3 = 800$	4.9434	4.6425
5 nodes	Input applied every 10T	$d_3 = 1300$	2.8084	2.6858
13 nodes	Input applied every time step	$d_3 = 600$	0.3836	0.3451
13 nodes	Input applied every 5T	$d_3 = 1100$	8.2882	5.5867
13 nodes	Input applied every 10T	$d_3 = 1600$	14.1021	9.0395
15 nodes	Input applied every time step	$d_3 = 700$	3.7550	3.2386
15 nodes	Input applied every 5T	$d_3 = 1200$	9.3123	8.5676
15 nodes	Input applied every 10T	$d_3 = 1700$	34.5440	25.5843
25 nodes	Input applied every time step	$d_2 = 600$	1.0844	0.9308
25 nodes	Input applied every 5T	$d_2 = 1100$	3.4638	3.4205
25 nodes	Input applied every 10T	$d_2 = 1600$	9.7282	8.4571

Neural identification was implemented in every case using the same structure as explained in Section 6.

9. Conclusions

The concept of V-stability is studied and applied for a discrete-time complex network with continuous discrete-time and impulsive control schemes. A neural network was implemented as an identifier to synthesize the control law. The different control schemes were able to drive the system to state zero with some modifications made to analyze results in the case of the impulsive schemes. A formal rework of V-stability for impulsive complex networks may be a good research area as it proves to be a useful tool for continuous and continuous in discrete-time dynamical complex networks. One disadvantage of this method is the necessity to check eigenvalues of a given matrix, as the difficulty to do this increases as the network grows in size, although the 20-node network used here and some that are bigger are still manageable for unspecialized computer equipment.

In this work, we applied a novel neural-impulsive pinning control for complex networks for seven different topologies, each one with three different scenarios: input applied every time step, input applied every 5T, input applied every 10T, and the control design was carried out based on V-stability in order to stabilize the network states at zero, and all cases have an excellent evolution, as shown by the results presented in Sections 7 and 8, showing the effectiveness of the proposed methodology for different complex network topologies with different dynamics in each node, without the need to apply control in every node at every time step.

An important area of application of the class of system considered in this paper is for biomedical purposes. Impulsive systems adapt well to how most of the treatments are administrated for different kind of diseases. In particular, a complex network approach, in this case, could represent the interaction between different individuals or a group of these people with specific relation to infectious diseases. These kinds of applications are considered as future work.

Author Contributions: D.R.-R. Studied the state of the art, methodology, implementation the simulation results. A.Y.A. Supervision, project administration, review, editing, funding acquisition. E.N.S. Methodology, funding acquisition. All authors have read and agreed to the published version of the manuscript.

Funding: The authors thank the support of CONACYT Mexico, through Projects CB-2015-256769 and CB-2015-258960 ("Project supported by Fondo Sectorial de Investigación para la Educación").

Acknowledgments: The authors also thank Universidad de Guadalajara and CINVESTAV, Unidad Guadalajara for the support in this research.

Conflicts of Interest: "The authors declare no conflict of interest." "The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results".

References

- 1. Chen, G.; Wang, X.; Li, X. Fundamentals of Complex Networks: Models, Structures and Dynamics, 1st ed.; Wiley: Singapore, 2015.
- 2. Khalil, H. Nonlinear Systems, 3rd ed.; Prentice-Hall: Upper Saddle River, NJ, USA, 2002.
- 3. Li, X.; Wang, X.; Chen, G. Pinning a complex dynamical network to its equilibrium. *IEEE Trans. Circuits Syst.* -*I Regul. Papers* **2004**, *51*, 2074–2087. [CrossRef]
- 4. Xiang, J.; Chen, G. On the V-stability of complex dynamical networks. *Automatica* 2007, 43, 1049–1057. [CrossRef]
- 5. Haddad, W.; Chellaboina, V.; Nersesov, S. *Impulsive and Hybrid Dynamical Systems*, 1st ed.; Princeton University Press: Princeton, NJ, USA, 2006.
- 6. Sun, W.; Lü, J.; Chen, S.; Yu, X. Pinning impulsive control algorithms for complex network. *Chaos* **2014**, 24. [CrossRef] [PubMed]
- Lu, J.; Ho, D.; Cao, J. A unified synchronization criterion for impulsive dynamical networks. *Automatica* 2010, 46, 1215–1221. [CrossRef]
- Zhang, Q.; Lu, J. Control the Complex Networks with Different Dynamical Nodes by Impulse. In Proceedings of the 6th International Symposium on Neural Networks, ISNN 2009, Wuhan, China, 26–29 May 2009. [CrossRef]

- 9. Alanis, A.; Sanchez, E.; Loukianov, A.; Hernandez, E. Discrete-time recurrent high order neural networks for nonlinear identification. *J. Frankl. Inst.* **2010**, *347*, 1253–1265. [CrossRef]
- 10. Sanchez, E.; Alanis, Y. *Redes neuronales: Conceptos fundamentales y aplicaciones a control automático*, 1st ed.; Pearson-Prentice Hall: Madrid, España, 2006.
- 11. Vega, C.; Sanchez, E.; Alzate, R. Inverse optimal pinning control for synchronization of complex networks with nonidentical chaotic nodes. *IFAC-Papers Online* **2018**, *51*, 235–239. [CrossRef]
- 12. Vega, C.; Sanchez, E.; Suarez, O. Synchronization of complex networks with nonidentical nodes via discrete-time neural inverse optimal pinning control. In Proceedings of the 2018 IEEE Latin American Conference on Computational Intelligence (LA-CCI), Guadalajara, Mexico, 7–9 November 2018. [CrossRef]
- Vega, C.; Sanchez, E.; Suarez, O. Discrete-time neural sliding-mode pinning control for synchronization of complex networks. In Proceedings of the 2019 16th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE), Mexico City, Mexico, 11–13 September 2019. [CrossRef]
- 14. Suarez, O.; Vega, C.; Sanchez, E.; Chen, G.; Elvira-Ceja, J.; Rodriguez, D. Neural sliding-mode pinning control for output synchronization for uncertain general complex networks. *Automatica* **2020**, *112*, 108694. [CrossRef]
- 15. Liu, B.; Xu, B.; Zhang, G.; Tong, L. Review of Some Control Theory Results on Uniform Stability of Impulsive Systems. *Mathematics* **2019**, *7*, 1186. [CrossRef]
- 16. Liu, B.; Liu, X. Uniform stability of discrete impulsive systems. Int. J. Syst. Sci. 2008, 39, 181–192. [CrossRef]
- Stamov, G.; Stamova, I.; Li, X.; Gospodinova, E. Practical Stability with Respect to *h*-Manifolds for Impulsive Control Functional Differential Equations with Variable Impulsive Perturbations. *Mathematics* 2019, 7, 656. [CrossRef]
- 18. Newman, J. The Structure and Function of Complex Networks. SIAM Rev. 2003, 45, 167–256. [CrossRef]
- 19. Newman., M.; Barabási, A.; Watts, D. *The Structure and Dynamics of Networks*, 1st ed.; Princeton University Press: Princeton, NJ, USA, 2006.
- 20. Aldana, M. Complex Networks. In Proceedings of the XV Summer School in Physics, Cuernavaca, Mexico, 30 July–11 August 2007.
- 21. Lorenz, E. Deterministic nonperiodic flow. J. Atmos. Sci. 1963, 20, 130-141. [CrossRef]
- 22. Chen, G.; Ueta, T. Yet another chaotic attractor. Int. J. Bifurc. Chaos 1999, 9, 1465–1466. [CrossRef]
- 23. Lü, J.; Chen, G. A new chaotic attractor coined. Int. J. Bifurc. Chaos 2002, 12, 659–661. [CrossRef]
- 24. Qi, G.; Chen, G.; Du, S.; Chen, Z.; Yuan, Z. Analysis of a new chaotic system. *Phys. A Stat. Mech. its Appl.* **2005**, *352*, 295–308. [CrossRef]
- Elhadj, Z.; Clinton, J. The unified chaotic system describing the Lorenz and Chua systems. *Ser. Elec. Energ.* 2010, 23, 345–355. [CrossRef]
- 26. Haykin, S. Neural Networks: A Comprehensive Foundation, 2nd ed.; Pearson Education: Delhi, India, 2005.
- 27. Puskorius, G.; Feldkamp, L. Parameter-Based Kalman Filter Training: Theory and Implementation. In *Kalman Filtering and Neural Networks*, 1st ed.; Haykin, S., Ed.; John Wiley & Sons, Inc.: New York, USA, 2001; pp. 23–68.



© 2020 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 (http://creativecommons.org/licenses/by/4.0/).