Received 11 September 2020; accepted 28 September 2020. Date of publication 5 October 2020; date of current version 22 October 2020.

Digital Object Identifier 10.1109/OJCOMS.2020.3028554

Deep Chaos Synchronization

MAJID MOBINI [1] AND GEORGES KADDOUM [2] (Senior Member, IEEE)

[1] Department of Electrical, Electronics and Communication Engineering, Babol Noshirvani University of Technology, Babol 1634993163, Iran

[2] Département de Génie Éelectrique, University of Québec, École de Technologie Supérieure, Montréal, QC H3C 1K3, Canada

CORRESPONDING AUTHOR: M. MOBINI (e-mail: mobini2002@gmail.com)

This work was supported by the Tier2 Canada research Chair Entitled "Towards a Novel and Intelligent Framework for the Next generations of IoT Networks."

ABSTRACT In this study, we address the problem of chaotic synchronization over a noisy channel by introducing a novel Deep Chaos Synchronization (DCS) system using a Convolutional Neural Network (CNN). Conventional Deep Learning (DL) based communication strategies are extremely powerful but training on large data sets is usually a difficult and time-consuming procedure. To tackle this challenge, DCS does not require prior information or large data sets. In addition, we provide a novel Recurrent Neural Network (RNN)-based chaotic synchronization system for comparative analysis. The results show that the proposed DCS architecture is competitive with RNN-based synchronization in terms of robustness against noise, convergence, and training. Hence, with these features, the DCS scheme will open the door for a new class of modulator schemes and meet the robustness against noise, convergence, and training requirements of the Ultra Reliable Low Latency Communications (URLLC) and Industrial Internet of Things (IIoT).

INDEX TERMS Deep learning, chaotic synchronization, DCS, CNN, Lorenz system, RNN.

I. INTRODUCTION

SYNCHRONIZATION is a fundamental requirement for a wide range of natural phenomena and new industrial technologies [1]. This article focuses on designing a new family of chaos-based receivers that benefit from the advantages of Deep Learning (DL) techniques. Historically, since the early 1980s, chaos-based communication started with the implementation of electronic circuits exhibiting chaotic behavior [2]. The second fundamental step towards chaos-based communication systems was in 1990. Pecora and Carroll discovered that synchronization can be achieved by coupling two chaotic systems with a common signal [3]. After that, several innovative systems have been proposed employing other outstanding features of chaotic waveforms. These waveforms are wide band, nonlinear, noise-like, non-periodic, and hypersensitive to initial conditions [4]. They also have excellent correlation properties and a simple production process. Thus, chaotic waveforms could be used in a vast spectrum of applications. They are major candidates for multi-user spread-spectrum schemes because of their wideband characteristics [5]–[8]. In addition to the spread-spectrum applications, numerous chaos based modulations have been proposed for use in digital wireless communications because of their robustness against the destructive effects of jamming and fading channels [9]. Moreover, many studies have been done on Low Probability of Interception (LPI) features [10] and chaos-based secure communication schemes [11].

In [12], the authors classified secure chaos-based techniques into four generations. The first three generations were based on continuous chaotic synchronization in which the bandwidth efficiency was very low. The fourth generation, called impulsive methods, employed impulsive chaotic synchronization to solve the low bandwidth efficiency drawback. The main design challenge of chaos-based secure schemes is how to deliver a secret message over a public channel using a transceiver structure that shows strong noise rejection capabilities. Several attempts made to design robust digital and analog chaos-based secure communication systems [13], [14]. In general, traditional studies of chaos synchronization rely on the fact that the equations of chaotic systems are known beforehand. This assumption is not realistic for practical chaotic systems in which only one or some observational signals are available [15].

Many recent studies have demonstrated the capability of ML-based approaches for modeling chaotic systems whose equations of motion are unknown. A special approach known as "reservoir computing" caused considerable progress in model-free prediction of chaotic systems [16], [17]. It is currently an active research area that uses natural potentials of Recurrent Neural Networks (RNNs). Reservoir computing has been widely used in the modeling of temporal processes. Also, reservoir computing has employed successfully in several practical applications, e.g., speech recognition, handwriting recognition, robot motor control, and financial forecasting [18]. More importantly, by transmitting just one scalar signal, synchronization can be obtained between well-trained reservoir computers. This technique can also extract Lyapunov exponents and captures the dynamics of a chaotic system [19].

On the other hand, labeling is a difficult and time-consuming process in some practical scenarios, such as chaos-based communications. To tackle this challenge, in our paper we proposed a self-supervised (untrained) structure for a chaos-based receiver. This new design is based on the DIP (Deep Image Prior [20]) approach to achieve a self-supervised configuration. To this aim, we employ a modern version of Generative Adversarial Networks (GANs), known as Deep Convolutional Generative Adversarial Networks (DCGANs), which have a feed-forward structure [21]. Such a design benefits from advantages of conventional DCGANs and self-supervised nature of the DIP approach. Because of these features, the proposed design leads to a high-performance self-supervised structure that relies less on the input data.

The results of this article can be extended to a wide range of applications, from health monitoring, and chaos-based security, to human behavior analysis and pattern studies in dynamic social networks [22]–[25]. The proposed model is flexible for use in a variety of applications with different requirements. It can be used to achieve real-time synchronization in applications that are vulnerable to timing and delay such as control of heart fibrillation. Chaotic conditions of the human heart cause arrhythmia and it can lead to death if left uncontrolled [26]–[28]. To retain a normal heart rhythm, accurate timing can be obtained using the proposed synchronization scheme. On the other hand, there is a lot of evidence that chaotic synchronization provides valuable information and a deeper understanding of physiological mechanisms underlying the brain [29]. When the timing is not the main challenge, de-noising features of the proposed approach can be exploited. Thus, it can be used for noise reduction in practical chaotic signals such as Electroencephalography (EEG), Electromyography (EMG), and other chaotic physiological signals [30]–[32].

A. BASIC DEFINITIONS

Synchronisation gives a conception of high correlations between connected systems. In its elementary definition, synchronisation points out to the tendency to have the same dynamical behaviour. In the context of communications, a local copy of chaotic signals can be recreated in the receiver side using appropriate synchronization circuitry. This is the most popular approach in coherent receivers, for recovering the original chaotic samples from the received noisy signals [33]. These recovered samples enters a correlator as a reference signal for data detection. The Bit Error Rate (BER) performance of these receivers depends on the "closeness" of the reference to the original chaotic samples. Using these receivers have some significant advantages over non-coherent receivers in terms of data rate, noise performance, and bandwidth efficiency. However, all the above advantages can be attained provided that the synchronization is preserved, Otherwise, all these advantages will be lost [34].

To gain a robust chaotic synchronization organization and excellent noise performance, several synchronization policies and various designs for the receivers are reported in the literature. Essential concepts of chaotic synchronization are explained in [3] by finding some effective applications to secure chaos-based communications. In a coupled Lorenz system as a remarkable synchronization scheme [35], [36], two identical chaotic generators are synchronized using a shared drive signal. This type of synchronization is stable and robust to perturbations in the drive signal [37], [38].

In this article, chaotic map selection and other important design decisions are based on the intended applications. These applications are associated with several important issues such as latency, power consumption, and security [39]. The Lorenz map, as a high dimensional chaotic map, has a more complicated mathematical structure and high chaotic complexity. One-dimensional chaotic maps are more suitable for applications with low latency requirements because they need fewer computational operations. On the other hand, simpler chaotic maps suffer from security limitations. This drawback is due to the limited chaotic range, low chaotic complexity, and higher dynamic behavior degradation rate [40]. In a chaotic masking communication scheme, the robustness to channel noise is a determining measure [41]. In this case, the original message is added to a chaotic signal. At the receiver side, the chaotic signal is subtracted from the received signal to retrieve the original data. In order to mask the original message, this signal should be generally much weaker than the state variable of the transmitter's chaotic system, and thus the message signal is vulnerable to channel noise. In this situation with high-security requirements, the Lorenz map can be a good choice. In [42] the authors have compared the Lorenz system with a Rössler system and it is shown that in the Lorenz case, even for rather strong noise, the complete synchronization is preserved.

In this study, we assume that the received signal is corrupted by Additive White Gaussian Noise (AWGN). This assumption has the following advantages:

1) *Tractability:* Noise effects are very important in chaotic synchronization because of the sensitivity of chaotic systems. In fact, a small noise may lead to instability and synchronization error [43]. This assumption

assists in detailed problem tracking by avoiding calculation complexity.

2) *Necessity:* Noise must be considered because it exists always before any other effects of the communication channel. Other destructive effects could be modeled as serial filter blocks after the noise block [33].

3) *Generality:* The relative performance of different communication schemes determined using the AWGN channel model remains valid under real channel models, i.e., systems under fading conditions generally follow the same trend as in the AWGN channels [33], [44], and [45].

B. BACKGROUND AND MOTIVATIONS

To cope with the destructive effects of the noise and distortion, several noise reduction methods have been recommended for chaotic signals, such as the local projection approach and lifting wavelet transform [46], [47]. This issue is even more challenging when the chaotic parameters or initial conditions are unknown. In this case, there are efforts to approximate the noise-free trajectory by estimation of the parameters and initial conditions [48]–[50]. However, in most of the reported works, the chaotic signal cannot be recovered precisely. On the other hand, since noise and chaos have similar spectral behaviour, regular frequency domain filtering degrades the signal of interest [51].

Recently, DL-based techniques have attracted great attention because of effectiveness in data-driven analysis of nonlinear dynamics [52]–[55]. Among these, Recurrent Neural Network (RNN) is frequently used to learn complex dynamics from the input data [16], [56]–[58]. Unfortunately, classical RNN methods suffer from vanishing gradients and tendency to take into account only short-term dependencies [59]. In [60], the authors compared the prediction performance of the RNN and Long-Short Term Memory (LSTM) in the presence of noise. They showed that when LSTM was trained on noisy data, it reduced the noise contribution in the prediction process. This feature can be used for noise reduction goals. However, when LSTM was trained on clean data it became susceptible to perturbations in the input data.

In a surprising new article [20], Deep Image Prior (DIP) method is suggested for image de-noising and inpainting using DCGANs [21]. In DIP, a deep convolutional neural network generator such as DCGAN is initialized with some random weights and these weights should be optimized in such a way that force the network to generate an image that resembles the target image as much as possible. This procedure does not use any prior information, or training on large data sets. Motivated by the potentials of the DIP method [20], this article presents some approaches to take advantages of the DL in order to promote the capabilities of the existing chaos-based communication systems.

C. CONTRIBUTIONS

This article is the first to introduce DL-based chaotic synchronization along with some approaches to improve trainable and untrained systems. In chaotic synchronization and other communication systems, large data sets and labeled data are not available. In addition, for a reasonable performance in regular DNN-based structures, learning and testing steps must be performed for the same channel realization results in a lot of overhead due to learning [61]. Innovative aspects of this article are briefly itemized below:

- As the main achievement, we introduce a Deep Chaos Synchronization system (DCS) that needs no data set. The proposed DCS benefits from the inherent security of the chaotic signals and advantages of DNNs. Moreover, it can offer a low synchronization error, especially for longer sequences. The above-mentioned features make DCS an appealing candidate for use in chaos-based Code Division Multiple Access (CDMA) systems [62], encrypted data transmission [63], [64], Ultra-Reliable Low Latency Communications (URLLC) [65], Industrial Internet of Things (IIoT) [65], and Wireless Sensor Networks (WSNs) [66], [67].
- In addition, we suggest a trainable RNN-based synchronization structure gains de-noising properties of the sequence to sequence RNN, as a special arrangement of the RNNs family for comparison with our proposed DCS approach. We accepted the traditional RNN as a well-known trainable approach since it can give a measure for the de-noising strength of the proposed method. Depending on the intended application, each has its own benefits and can be used.

In order to initial condition estimation, we offer a Genetic Algorithm (GA)-based approach, which gives a suitable reconstruction of the transmitted signals. A correct estimation is vital in chaotic synchronization because of their sensitivity to the initial condition. With accurate initial conditions, the slave system can be synchronized with the master system almost instantaneously, thereby eliminating the problem of transients. This technique can still keep synchronization where the largest conditional Lyapunov exponent is positive and other strategies cannot be synchronized [35].

In order to select a proper map among various chaotic maps, we tested several chaotic maps as the DCS system input. We found a tractable trade-off between processing time and noise robustness. In other words, we show that using the Rössler map [40] and discrete-time Henon map [70], the processing time reduced. The average amplitude error between the recovered drive signal and the original signal shows that the Lorenz map is superior to other maps from a noise robustness point of view.

The rest of this article is organized as follows: In Section II, the basics of the Lorenz map and some other chaotic maps are investigated. Also, the employed initial condition estimation method is described in this Section. Moreover, some basic information about DCGANs, the structure of the proposed transmitter/receiver, and other information related to the DCS system is presented in Section II. In Section III, the RNN-based synchronization system is introduced for comparison with our DCS

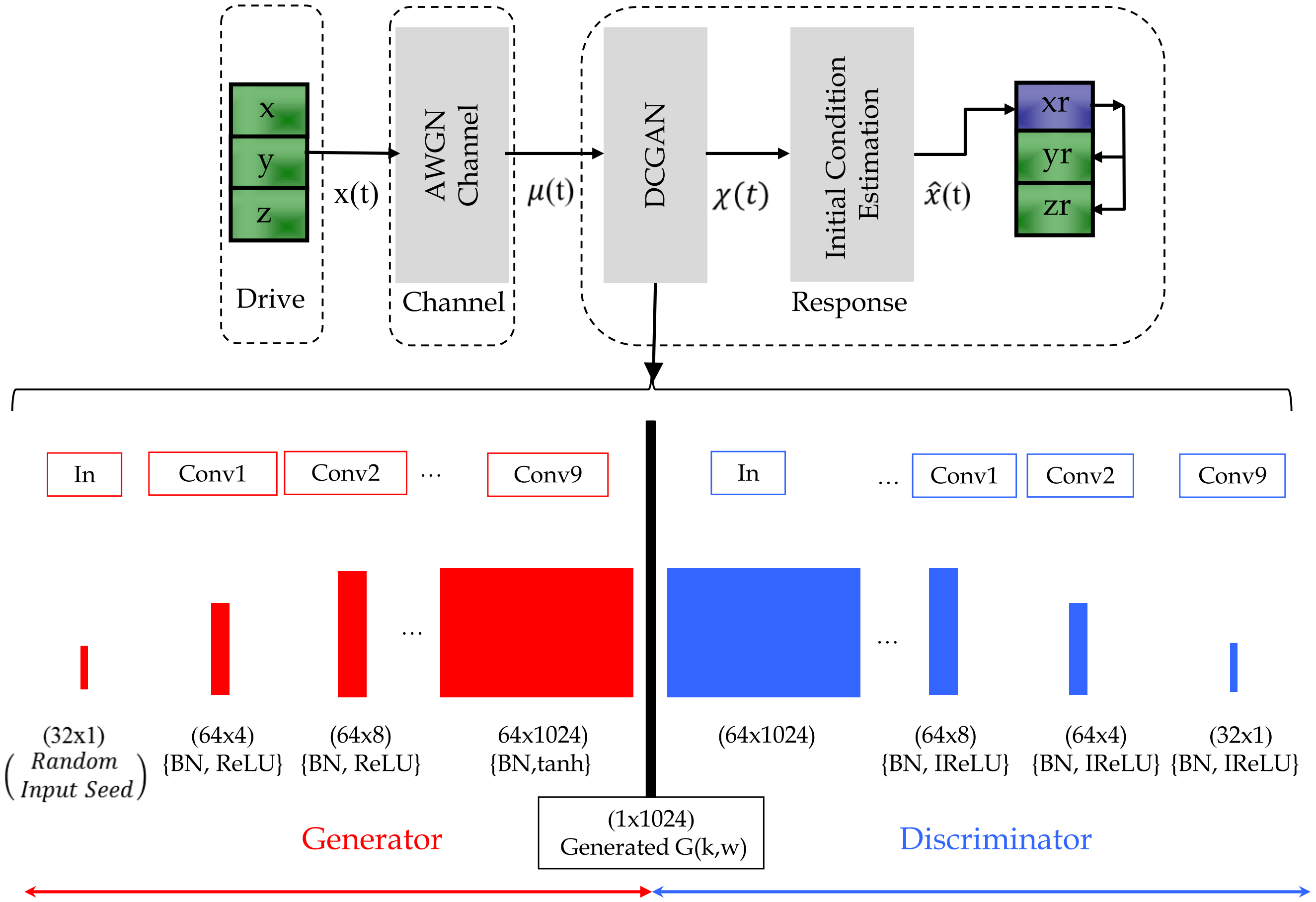


FIGURE 1. DCS model based on DCGAN as the generative model.

method. In Section IV, simulation results and discussions are presented. The conclusions are explained in Section V.

II. DCS MODEL

In this section, the basics of the Lorenz chaotic synchronization system are revisited and then the offered Deep Chaos Synchronization model is presented.

A. CONVENTIONAL LORENZ SYSTEM

Consider the following conventional Lorenz system of Equations [1]:

$$\begin{aligned}\dot{x} &= \rho(y-x),\\ \dot{y} &= rx - y - xz,\\ \dot{z} &= xy - \beta z,\end{aligned} \tag{1}$$

where x, y, and z are the state variables at the drive (Master) subsystem and ρ, r, and β are control parameters. When these parameters are chosen as $\rho = 10$, $r = 28$, and $\beta = 8/3$ the equations lead to chaotic dynamics while the initial conditions shape the chaotic behavior [3]. At the receiver side, the received noisy signal can be written as:

$$\mu(t) = x(t) + \zeta(t), \tag{2}$$

where $\mu(t)$ is a given noisy observation, $\zeta(t)$ is Additive White Gaussian Noise with zero-mean and power spectral density σ_n^2. The response (Slave) uses the $\mu(t)$ as driver, in order to reconstruct Lorenz state variables y_r and z_r:

$$\begin{aligned}\dot{y}_r &= r\mu - y_r - \mu z_r,\\ \dot{z}_r &= \mu y_r - \beta z_r.\end{aligned} \tag{3}$$

We presume that all control parameters ρ, r, and β of the master sub-system are known at the receiver side. The initial values of the master subsystem are (x_0, y_0, z_0) and the slave subsystem is initiated similarly, except that we do not know one of the initial values (x_0). In a Lorenz system starting from two slightly different initial states (x_0, y_0, z_0) and (x_{r0}, y_{r0}, z_{r0}), the trajectories diverge from each other. At the end of this section, a GA-based solution to this challenge is presented.

B. COMPARISON WITH OTHER CHAOTIC MAPS

In addition to the Lorenz system, there are many other chaotic systems that generate chaos and can be utilized in communication systems. For example, Rössler systems can be decomposed into stable drive-response sub-systems to produce a robust synchronization [40]. Moreover, several discrete-time maps, such as the Henon map, are also decomposable into synchronizing sub-systems without the burden of solving differential equations [70]. In Section IV, we discuss the merits and demerits of using these systems. To investigate the effect of different maps on de-noising performance of the DCS system, we first adopt a signal from the Rössler system as input [40]:

$$\begin{aligned}\dot{x} &= \omega y - z,\\ \dot{y} &= \omega x + 0.15y,\\ \dot{z} &= 0.4 + z(x - 8.5),\end{aligned} \tag{4}$$

with $\omega = 0.95$, and another chaotic signal from discrete-time Henon system [70]:

$$\begin{aligned}x[n+1] &= 1 + y[n] - z[n]x^2[n],\\ y[n+1] &= bx[n],\\ z[n+1] &= z[n] - 0.5 + \beta x^2[n],\end{aligned} \tag{5}$$

with $b = 0.25$, and $\beta = 0.279$.

C. PROPOSED DCGAN-BASED RECEIVER

In this study, motivated by the appealing factors of the DIP model [20], the one-dimensional arrangement of DIP is implemented using a DCGAN. In particular, we adapt the input and output of the DIP model and use it as a baseline which we term the Deep Chaos Synchronization. The receiver of the DCS is composed of three stages, as shown in Fig. 1. The implementation steps of the DCS are illustrated in Algorithm 1.

In the first stage, a less noisy signal is achieved from the DIP method. When the objective is to generate data identical to a target distribution, a Generative Adversarial Network (GAN) provides data from a similar distribution efficiently [68]. However, GANs are unstable to train and this regularly generates nonsensical outputs.

DCGAN is one of the attempts to scale up GANs using CNNs. Compared to conventional GANs, spatial pooling functions are replaced with convolutional layers, allowing the DCGAN to learn its own spatial downsampling. Also, DCGAN adopts Batch Normalization (BN) which stabilizes learning by normalizing the input to each unit to have zero mean and unit variance. Moreover, ReLU activation functions are considered for all layers. The RMSProp optimizer is used as a variant of the traditional gradient descent method that adopts a momentum parameter for faster convergence of the training process.

In the second stage, we offer a GA-based approach in order to estimate the initial condition using the achieved clear signal from the previous stage. An accurate initial state estimation results in a proper reconstruction of the original signal.

In the final stage, the estimated initial value enters the response sub-system in order to reconstruct Lorenz attractors. By doing the above mentioned three stages the inherent advantages of a chaotic system are combined with the noise reduction capability of the DIP so as to have remarkable performance. However, the complexity increases in comparison with the conventional Lorenz synchronization system, but compared to RNN-based designs, DCS needs no labeled data or large data set. Moreover, no need to send train signals result in a lot of overhead reduction.

Algorithm 1: DCS Receiver Pseudo-Code

```
1:  Initialize:
    Observed Noisy Sequence $\mu$, Random vector $w$;
2:  DCGAN pseudo-code:
     2-1: Generator ($\mu$)
            - 1D Convolution Transpose ();
            - Batch Normalization ();
            - Relu ();
            - Tanh ();   (At the end of Layers)
     2-2: Discriminator (Input random vector k)
            - 1D Convolution transpose ();
            - Batch Normalization ();
            - lRelu ();
     2-3: DCGAN training loop ()
            - Draw training example from the $\mu$ and $w$;
            - Generate a network $G(w, k)$;
            - Network training using (6):
              $w = RMSprop(G(w, k), LR, MOM)$;
            - Update loss;
              $Output = Network(w)$;
3:  DIP Output ():
            - Solving (6) using GD method;
              $\chi = G^*(w, k)$;
4:  Initial condition estimation ($\chi, \hat{x}$):
            - GA based calculation of the $x_0$;
5:  Reconstruction of $x_r, y_r, z_r$
6:  End
```

As shown in Fig. 1, the DCS receiver is implemented using a DCGAN located at the input of the response subsystem. We aim to tackle the inverse problem of de-noising a drive signal $x(t) \in \mathbb{R}^N$ which has been transmitted on a communication channel with Gaussian noise $\zeta(t) \in \mathbb{R}^N$. Therefore, given a noisy observation $\mu(t)$, the basic idea is to reproduce a signal $\chi(t)$ that is similar to $x(t)$. To implement the DCGAN, We need to configure and organize the three blocks: 1) the generator, 2) the discriminator, and 3) the training process. The discriminator can be implemented by successive convolutional layers, BN layers, and Relu activation functions. In each of the convolutional layers, we downsample the spatial dimension of the input. The generator of the DCGAN consists of a sequence of transpose convolutional layers that upsample the input random sample to generate an artificial signal. A DCGAN can be trained in the same way as a regular GAN. Different from the conventional DCGAN models, in the DIP approach, we optimize the "*random weights*" to force the network to produce an output similar to the target. Therefore, $G(k, w)$ is the output of DCGAN with weights w and latent vector k. In other words, instead of minimizing $\|\chi - \mu\|2$, the network optimization problem of the DCS can be formulated as:

$$w^* = \underset{w}{\operatorname{argmin}} \|G(k, w) - \mu\|^2, \tag{6}$$

where $\mu \in \mathbb{R}^N$ is the observed noisy signal and $G(k, w^*) = \chi$ is the output clean sequence. This non-convex optimization problem can be solved easily using Gradient Descend (GD) method. As an alternative optimization strategy, we can regularize the network using a two stages optimization, which has shown converges better than simple Mean Squared Error (MSE) optimization [69], [71]. Therefore, we first train the adversarial network using a noisy sequence μ, an RMSProp optimizer, and the loss function in (6). Then, this problem can be solved through a simple GD algorithm to obtain the final solution.

D. INITIAL CONDITION ESTIMATION USING GA

The chaotic trajectory reproduction is extremely susceptible to the initial conditions. On the other hand, in most practical scenarios, the initial conditions are hard to measure precisely. So, as a reasonable assumption, we assume that we have only the upper and lower bounds for initial conditions. After an effectively initial condition estimation, we can get a noise-free signal, and the slave system can follow the dynamics of the master system almost instantaneously, thereby eliminating the issue of transients. With these advantages, initial condition estimation is one of the hot topics in chaos-based studies. Some articles have emphasized that when alternative synchronization techniques fail, this approach can achieve synchronization [72]–[76]. For example, in [47], [73] some heuristic algorithms are employed to initial state estimation and robustness of the chaotic synchronization to noise.

In this article, in order to initial condition estimation, we offer a simple GA-based method that provides an excellent reconstruction of the transmitted signal. The GA-based optimization is a convenient tool for non-linear problems that are not appropriate for analytical optimization methods. In addition, GA approach is fit for solving complex multi-variable problems, for example, where we would like to estimate several chaotic parameters and initial conditions. In the literature different methods have been proposed for initial condition estimation, e.g., Differential evolution algorithm-based methods [56], or Hybrid Mehta-heuristic methods [57]. Since the main idea behind this study is to implement an untrained structure for chaos-based receivers and our main focus is not on the precise parameters and initial value estimation, we select the GA to simplify the implementation.

We focus on the simultaneous start property and the security aspects of the initial condition estimation method [71]. The simultaneous start property is important in the synchronization of two coupled discrete-time systems, because it enables us to obtain real-time synchronization without transient time. The receiver can start de-noising process after receiving the first sample. In other words, because the receiver gains deep learning structures, it is inherently predictive, and after receiving the first sample, it can predict the next samples. In the synchronization model presented, if the response sub-system knows the initial conditions of the transmitter, or estimates them accurately, it can make a copy of the transmitted signal without the need for additional signaling. From a security point of view, when the adversary network has access to the wireless medium and public samples x_0, we can secure the DCS system by sharing y_0 and z_0 with legal users as private keys. We can use any of the transmitters outputs x_0, y_0, and z_0 for signal transmission or security purposes. Estimating other variables imposes time and computational load and without loss of generality, we take a single variable case x_0 to decrease the computational complexity.

In our setting, equal parameters ρ, r, and β are considered for the drive and response systems because the synchronization between them is stable provided that the parameters are identical. Here, (x_0, y_0, z_0) and (x_{r0}, y_{r0}, z_{r0}) are the initial states of the drive and response systems, respectively. We have adopted the x component to operate as a drive signal and it is also assumed that the receiver does not know the initial state of the x component (i.e., x_0) correctly and only knows the upper and lower bounds of it. To cope with the above-mentioned trajectory divergence phenomenon, in [73], the authors recommended a piece-wise estimation for each segment of the observed signal. Mathematically, initial state estimation in such a problem reduces to the optimization of the following objective function:

$$\underset{x_0}{\operatorname{argmin}} \left[f_{obj}(x_0) = \frac{1}{T_s} \sum_{t=1}^{T_s} \left(\hat{x}(t) - \chi(t) \right)^2 \right], \tag{7}$$

where x_0 is initial condition of the x component, for the Lorenz map. The idea is to minimize the objective function $f_{obj}(x_0)$ given by the mean-squared error between a candidate chaotic signal solution $\hat{x}(t)$ and the output of the above mentioned neural network $\chi(t)$. If an initial population is determined for (x_0), the solution will be found by an iterative approach such as GA. The initial population can be modified based on some rules such as selection, mutation, and crossover to achieve the optimal solution [77]. In particular, we use a decimal representation of genes in the range [0 0.1], uniform mutation, and one-point crossover. It is necessary to notice that here we determine the accuracy of our estimates by the initial population definition. In other words, the larger the initial population, the higher the accuracy of estimation, and accordingly the reconstructed signal $\hat{x}$ will be closer to the primary signal x. Iterative solving of the system of equations in (1) can give the best solution $\hat{x}(t)$. This can be performed in Python with the help of `scipy.integrate.odeint` function. Ultimately, we can reconstruct Lorenz attractors y_r and z_r at the receiver side using parameters and the estimated initial condition.

III. TRAINABLE RNN-BASED SYNCHRONIZATION MODEL

In this section, we introduce a trainable RNN-based synchronization scheme for comparative investigation. Fig. 2 illustrates the suggested model. It is noted that since the DCS is the first self-supervised chaotic synchronization system, the performance comparison cannot be provided with other

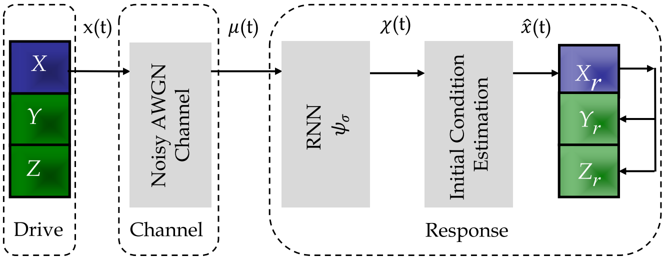


FIGURE 2. Trainable RNN-based synchronization System.

self-supervised systems. Therefore, the comparison with the RNN-based synchronization system can provide a general measure of de-noising power of them, although, they belong to the two different training classes.

A standard RNN includes of a hidden layer **h** and an output χ works on an input sequence **x**$= x(1), \ldots, x(T)$. The hidden state $h(t)$ is updated at each time step as follows:

$$h(t+1) = \psi(x(t), h(t)), \tag{8}$$

where ψ is a non-linear activation function, such as a logistic sigmoid function. An RNN learns a probability distribution over a sequence by being trained to predict the next symbol in a sequence. The output at each time step t is the conditional distribution $p(x(t+1)|x(t), \ldots, x(1))$. From this learned distribution, it can sample a different sequence by iteratively sampling. The RNN is a remarkable tool for learning complex dynamics and, at the same time, has a lot of noise reduction capabilities. It is also proper for Natural Language Processing (NLP) and signal prediction problems. Among these capabilities, we focus on the de-noising potentials of the Encoder-Decoder sequence to sequence RNNs as a special structure of the RNNs family. It consists of two RNNs, one of them encodes a sequence into a fixed-length vector and the second RNN decodes the vector into another sequence [78]. In the RNN-based signal prediction problem, the model predicts the next sequence given a previous sequence of vectors. This will enable us to generate a new sequence, one vector at a time. The architecture of the de-noising problem is similar to the prediction process, except that the signal given to the input of encoder is noisy, and the desired output of the decoder should be a clear signal. We adopt a newer version of dynamic RNN functions, known as Gated Recurrent Unit (GRU), which realizes the RNN for us [79]. GRU has no output gate and incorporates the input and previous gates into an update gate. This new gate manages how much the internal state is combined with a candidate activation.

We select GRU because it requires less computation while provides comparable results with LSTM. In addition, GRU can be easily implemented by `GRUCell` in TensorFlow. Different GRU structures are supported by TensorFlow in the `tf.nn.rnn_cell` module [80], [81]. Fig. 3 illustrates the arrangement of the stacked GRU sequence to sequence RNN.

The G is the value of the hidden output of the encoder in the last time step. We should repeat this value as an input to the further RNN to make it remember the context of the present at all moments when predicting the future. We have now defined the RNN and should train the model using an optimizer. The input and output neuron's dimensions should be $m = d = 1$ because the number of input train and test sequences are one $n_{train} = n_{test} = 1$. We consider a supervised framework and generate two sequences of the noisy chaotic samples with length T as *train* and *test* signals. In the meantime, we keep their noise-free version (or *target*) for final evaluation. In this framework, we train the model showing the *train* sequence and reading a prediction from the output neurons in the test phase. Even with a corrupted *test* signal, this prediction is close to the ground truth [57]. The RNN is trained to suppress the contributions from the noisy observations and minimize the following objective function using an ADAM optimizer [82]:

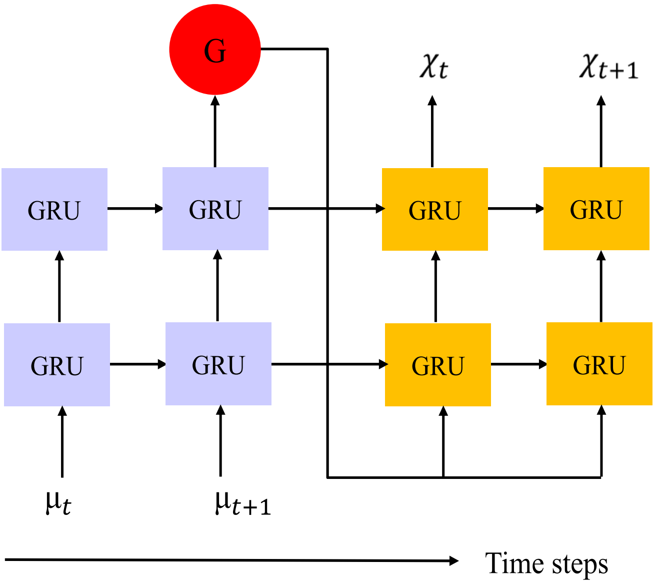


FIGURE 3. The stacked GRU sequence to sequence RNN structure.

$$L = \sum_{t=1}^{N} \frac{1}{2} (\mu(t) - \chi(t))^2. \tag{9}$$

Here,

$$\chi(t+1) = \psi_\sigma(\mu(t+1), G(t)), \tag{10}$$

where $h(t) \in \mathbb{R}^N$ are the internal states, N is the number of memory units, and $\chi(t)$ is the predicted/de-noised signal. ψ_σ denotes the RNN trained against noise with a standard deviation σ. In order to reconstruct Lorenz attractors, χ enters the initial condition estimation block. After initial state estimation, the desired attractors can easily re-produced by the Lorenz circuit.

IV. SIMULATION RESULTS

In this section, a number of simulations are considered for a comparative analysis of the DCS, RNN-based

TABLE 1. List of the Parameters.

Variable	Description	RNN	DCS
N	Size of the RNN	64	-
n_{test}	Number of test signals	1	1
n_{train}	Number of training signals	1	-
T	Length of train / test signals	1024	1024
N_{GF}	Convolutional filter per layer	-	64
Iter	Iterations during the training	800	800
Mom	Momentum of RMSProp	-	0.9
m	Output dimension	1	1
d	Input dimension	1	1
LR	Learning Rate	10^{-4}	10^{-4}
η	Input seed	-	0

TABLE 2. GA Parameters.

Parameter	Value
population size	10000
Number of variables	1
Maximum iterations	10
Crossover fraction	0.1
Mutation fraction	0.1

synchronization, and traditional Lorenz coupled system in (3). The considered parameters are listed in Table 1.

Many methods have been proposed to implement chaotic trajectories in python. We solve the Lorenz equations, using `scipy.integrate.odeint` function according to the method presented in [83]. This function integrates the system of Ordinary Differential Equations (ODE) and returns the solution. We produce 1024 data points with a time step = 0.1. It is presumed that all control parameters of the master sub-system are known at the receiver side. To produce chaos dynamics, the control parameters are selected as $\rho = 10$, $r = 28$, and $\beta = 8/3$. The initial values of the master subsystem are $(x_0, y_0, z_0) = (0.1, 0.1, 0.1)$ and the slave subsystem is the same, except for one of the values (x_0), which is assumed to be unknown and must be estimated. To this aim, we offer the GA-based initial condition estimation method. The parameters of the GA-based approach are listed in Table 2.

A. DE-NOISING PERFORMANCE

We can easily make the RNN model using NumPy and TensorFlow packages to produce a clear signal from a noisy observation. This ability comes from that the RNN tends to weaken the contribution of the noise and captures its own learned dynamics. Fig. 4 shows the convergence curve of the RNN, in which the loss is plotted versus the number of iterations during the training process. In this simulation, the RNN is trained at $\sigma_n^2 = 0.3$ while is tested at $\sigma_n^2 = 0.5$. If σ_n^2 is set at a big value, the RNN might learn nothing but the noise. On the other hand, if the training σ_n^2 is low, the RNN can only learn the clean signal. For a proper training σ_n^2 value must help the RNN to learn both types of samples. The proposed RNN-based system can be trained at a specific and proper noise condition, while operating throughout the range. during the training stage, we employ ADAM optimizer to minimize the loss function presented in (9). One can see that the RNN converges to the best case after several iterations. This demonstrates that the training phase not only imposes computational complexity, but also can be time consuming for the receiver and results in a lot of overhead.

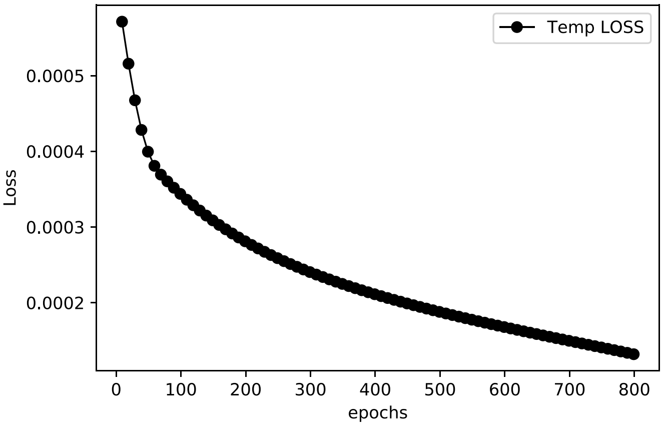
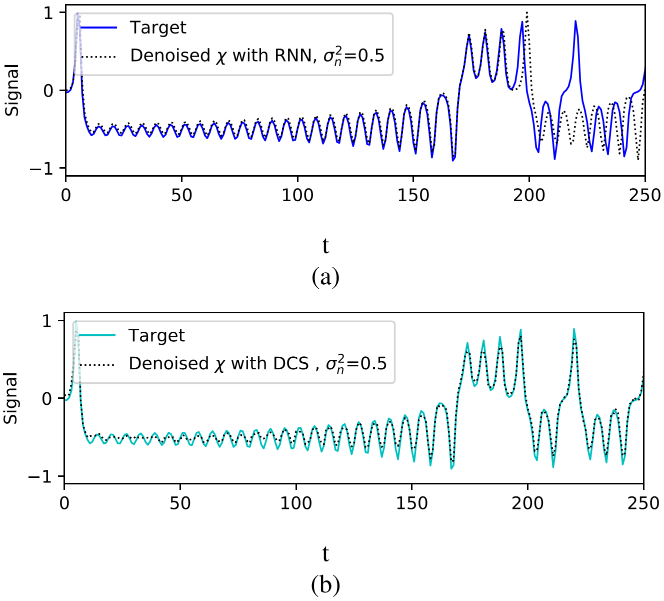


FIGURE 4. Optimizing iterations during the training of RNN.



(a)

(b)

FIGURE 5. (a) Target and de-noised signal (χ) by RNN (b) Target and de-noised signal (χ) by DCS.

Fig. 5 (a) compares the target (x) and de-noised (χ) signals. In order to facilitate visual comparison, the first 250 samples of these signals are plotted. The RNN model follows the target dynamics, but the synchronization is degraded after about 200 time steps. In other words, the RNN is not able to capture the dynamics of the drive signal and its error signal has a large domain. Therefore, it is necessary to re-train the receiver after the above time. Re-training process consumes time and other resources, and results in high computational complexity. In contrast, as shown in Fig. 5 (b), the DCS

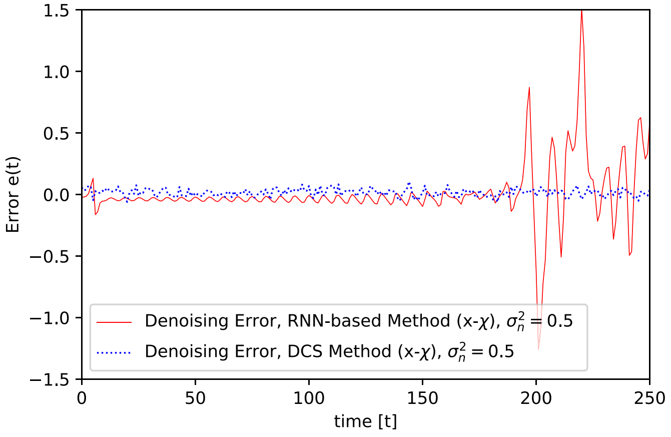


FIGURE 6. **Comparison between de-nosing error $x(t) - \chi(t)$ of the RNN and DCS.**

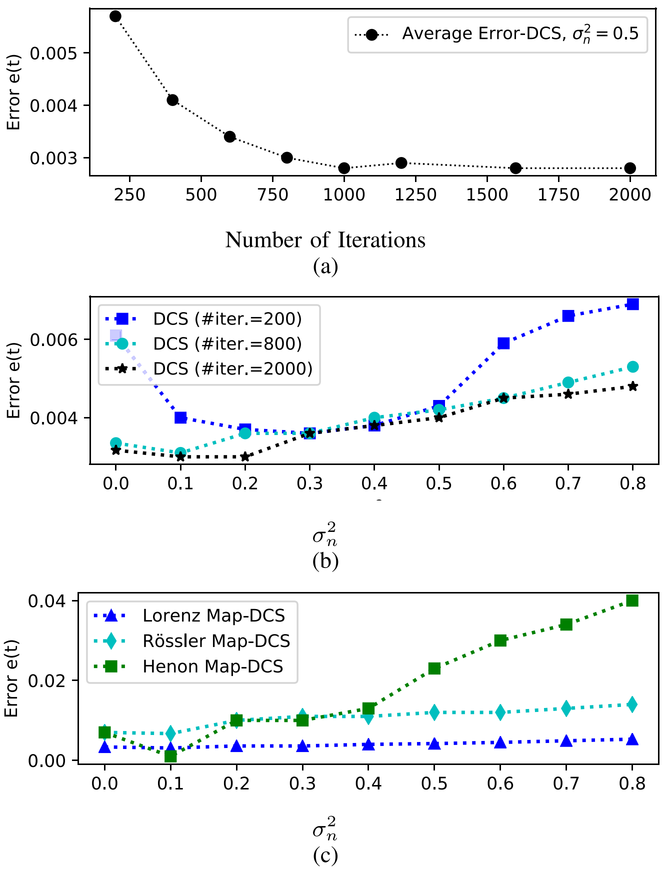


FIGURE 7. **(a) DCS convergence. (b) Effect of increasing the number of iterations. (c) A comparison among the noise performance of the Lorenz, Rössler, and Henon maps.**

model, which is implemented using Torch and TensorFlow libraries, follows the driving dynamics perfectly.

Fig. 6 shows de-noising error, i.e., $(x(t) - \chi(t))$ of the DCS compared with the RNN system in the same scale. We observe that, although the DCS does not use pre-training, it follows dynamics of the target signal and the error amplitude remains fixed over time.

Fig. 7 (a) plots the average amplitude error of the DCS for different number of iterations. The average amplitude error converges to the best case rapidly after about 600 iterations. Fig. 7 (b) shows the average amplitude error between the recovered drive signal and the original signal for different values of σ_n^2. As illustrated in this figure, for large noise values, when the number of iterations increases to 800, the error becomes very small. Increase the number of iterations from 800 to 2000 has a negligible impact on the error performance. In Fig. 7 (c), the robustness of the DCS system is measured for different chaotic maps. Execution time of the DCS algorithm is also measured for each of the maps. The average amplitude error between the recovered signal and the original signal shows that even for a rather strong noise, the Lorenz map is more robust. However, when we use the Rössler and Henon maps presented in (4) and (5), the processing time is reduced by 20% and 25%, respectively. The Lorenz map has a more complicated structure and high chaotic complexity, while the Rössler and Henon maps are faster and more suitable for applications with low latency requirements such as URLLC and IIoT. On the other hand, simpler chaotic maps suffer from security limitations, and the Lorenz map is more suitable for applications such as synchronization in extremely noisy channels and secure communications. According to the above discussion, future systems should make different compromises to meet their requirements.

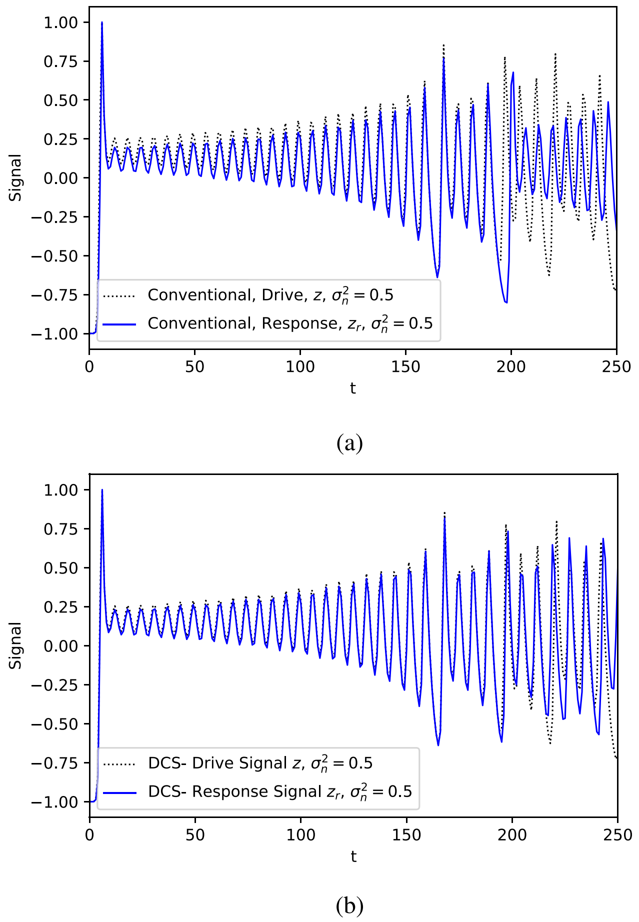
B. SYNCHRONIZATION ERROR COMPARISON

The following simulations deal with the output signals of the response system. Fig. 8 show a comparison between the output of the traditional Lorenz coupled system in (3) and the proposed DCS. Fig. 8 (a) shows the master signal $z(t)$ and reconstructed attractor $z_r(t)$ using the traditional system. The parameters in the transmitter and receiver of the conventional system are the same. We assume that in the receiver, the initial state of (x_0) is a random number in the range [0, 0.1]. Fig. 8 (b) shows the reconstructed attractor $z_r(t)$ by the DCS. The amplitude of the synchronization error is very small because of the noise reduction and initial state estimation in DCS, and the synchronization is more persistent over time.

Fig. 9 presents $z(t) - z_r(t)$ as the synchronization error criterion. To make the visual comparison easier, error signals are plotted over a smaller time span. In comparison with the traditional coupled Lorenz system and the RNN-based synchronization system, the DCS model conserves the dynamics of the master signal while it has a very small error amplitude.

Because the visual comparison may not be accurate, in another experiment we measured the average synchronization error for the above mentioned span $(t = 250)$ and for different values of the noise power. The results are shown in the Table 3. Observe that the DCS method is always superior to the other two methods.

More exactly, consider a scenario, where a chaotic signal with 1024 samples generated, and portions of the signal with specified lengths are randomly selected for transmission and performance evaluation. We repeated this experiment ten times for each determined length and calculated the average synchronization error.



(a)

(b)

FIGURE 8. Master signal z(t) and reconstructed signal in (a) conventional system (b) DCS.

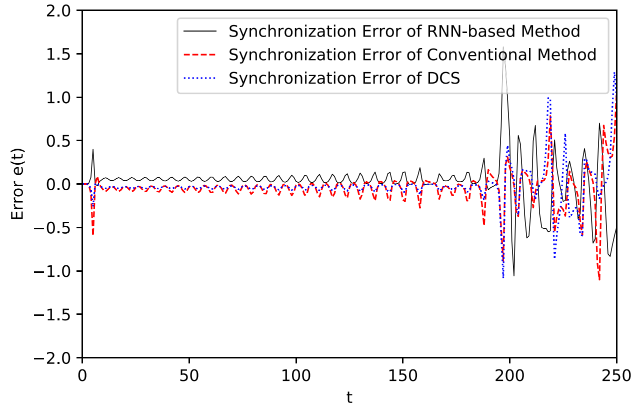


FIGURE 9. Comparison between synchronization error of the different systems for $\sigma_n^2 = 0.5$.

TABLE 3. Average Synchronization Error of Different Methods for Different Noise Values.

	RNN-based Method	Conventional Method	DCS
$\sigma_n^2 = 0.1$	0.05101	0.07180	0.02791
$\sigma_n^2 = 0.2$	0.05111	0.07181	0.02851
$\sigma_n^2 = 0.3$	0.04428	0.07171	0.02779
$\sigma_n^2 = 0.4$	0.05193	0.07192	0.02780
$\sigma_n^2 = 0.5$	0.05195	0.07195	0.02781
$\sigma_n^2 = 0.6$	0.05501	0.07196	0.02781
$\sigma_n^2 = 0.7$	0.06333	0.07197	0.02782

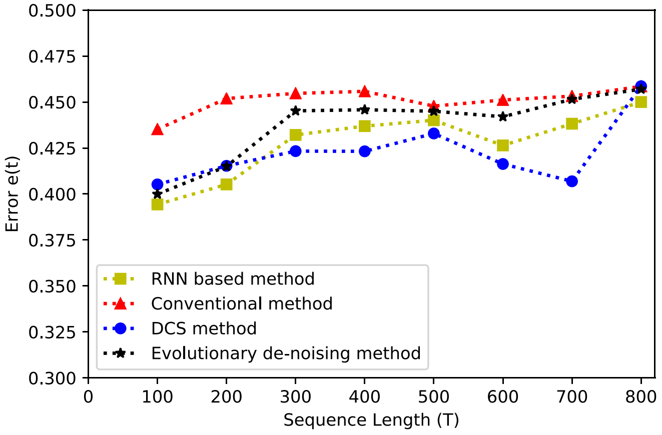


FIGURE 10. Comparison between synchronization error of the different systems for $\sigma_n^2 = 0.5$.

As shown in Fig. 10, the DCS system has a competitive performance with the RNN-based system and even works better for signals with $T > 250$. Furthermore, in this figure, both proposed systems have been compared with the evolutionary de-noising method presented in [73]. To make a fair comparison, GA-based initial condition estimation is used for all of them, according to the values in Table 2. because of the benefits of neural networks, both of the proposed systems are more robust than the evolutionary de-noising method that only uses initial condition estimation, specifically when $T > 250$. It is important to note that we have deliberately selected large noise and high perturbation of the initial values in this experiment so that the performance of the various methods is visually distinguishable. Adjusting parameters like the initial population and mutation rate in GA can counteract these damaging effects and eliminate synchronization errors.

V. CONCLUSION

This article presents the first attempt of using DL in chaos-based communication systems with providing an excellent synchronization between two coupled Lorenz systems. The results showed that the DCS reduces the synchronization error compared to the traditional systems and RNN-based approach. Moreover, the proposed DCS is easy to implement since it does not require training on large data sets. Thus, it can be used in coherent wireless communication scenarios where users or mobile units are generally hard to train. Practical implementations of DCS-based applications, e.g., chaos-based CDMA systems and low latency communications is an interesting topic for future researches.

REFERENCES

[1] D. Eroglu, J. S. Lamb, and T. Pereira, “Synchronisation of chaos and its applications,” *Contemp. Phys.*, vol. 58, no. 3, pp. 207–243, 2017.

[2] L. Chua, J. B. Yu, and Y. Y. Yu. “Negative resistance devices,” *Int. J. Circuit Theory Appl.*, vol. 11, pp. 161–186, Jul. 1983.

[3] L. M. Pecora and T. L. Carroll, “Synchronization in chaotic systems,” *Phys. Rev. Lett.*, vol. 64, p. 821, Feb. 1990.

[4] J. C. Feng, *Reconstruction of Chaotic Signals With Applications to Chaos-Based Communications*. Singapore: World Sci., 2008.

[5] A. P. Kurian, S. Puthusserypady, and S. M. Htut, "Performance enhancement of DS/CDMA system using chaotic complex spreading sequence," *IEEE Trans. Wireless Commun.*, vol. 4, no. 3, pp. 984–989, May 2005.

[6] R. Vali, S. Berber, and S. K. Nguang, "Accurate derivation of chaos-based acquisition performance in a fading channel," *IEEE Trans. Wireless Commun.*, vol. 11, no. 2, pp. 722–731, Feb. 2012.

[7] R. Vali, S. Berber, and S. K. Nguang, "Analysis of chaos-based code tracking using chaotic correlation statistics," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 59, no. 4, pp. 796–805, Apr. 2012.

[8] S. Berber and S. Feng, "Chaos-based physical layer design for WSN applications," in *Proc. Annu. Conf. CIRCOM*, vol. 2, 2013, pp. 157–162.

[9] W. Wei and J. Kim, "Modeling and analysis of chaos-based spread spectrum scheme using irregular LDPC code and non-coherent 16-DCSK under fading and jamming," *Eng. Technol. Appl. Sci. Res.*, vol. 9, no. 6, pp. 5080–5087, 2019.

[10] J. Yu and Y.-D. Yao, "Detection performance of chaotic spreading LPI waveforms," *IEEE Trans. Wireless Commun.*, vol. 4, no. 2, pp. 390–396, Mar. 2005.

[11] V. Lynnyk and S. Celikovský, "On the anti-synchronization detection for the generalized Lorenz system and its applications to secure encryption," *Kybernetika*, vol. 46, no. 1, pp. 1–18, 2010.

[12] T. Yang, "A survey of chaotic secure communication systems," *Int. J. Comput. Cogn.*, vol. 2, no. 2, pp. 81–130, 2004.

[13] L. Kocarev, K. Halle, K. Eckert, U. Parlitz, and L. Chua, "Experimental demonstration of secure communications via chaotic synchronization," *Int. J. Bifurc. Chaos*, vol. 2, no. 3, pp. 709–713, 1992.

[14] I. Pehlivan and Y. Uyaroglu, "Rikitake attractor and it's synchronization application for secure communication systems," *J. Appl. Sci.*, vol. 7, no. 2, pp. 232–236, 2007.

[15] T. Weng, H. Yang, C. Gu, J. Zhang, and M. Small, "Synchronization of chaotic systems and their machine-learning models," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 99, no. 4, 2019, Art. no. 042203.

[16] H. Jaeger and H. Haas, "Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication," *Science*, vol. 304, no. 5667, pp. 78–80, 2004.

[17] T. L. Carroll, "Using reservoir computers to distinguish chaotic signals," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 98, no. 5, 2018, Art. no. 052209.

[18] M. Lukoševičius, H. Jaeger, and B. Schrauwen, "Reservoir computing trends," *KI—Künstliche Intelligenz*, vol. 26, no. 4, pp. 365–371, 2012.

[19] Z. Lu, B. R. Hunt, and E. Ott, "Attractor reconstruction by machine learning," *Chaos*, vol. 28, no. 6, 2018, Art. no. 061104.

[20] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, 2018, pp. 9446–9454.

[21] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015. [Online]. Available: arXiv:1511.06434.

[22] G. Huang, A. Y. Zomaya, F. C. Delicato, and P. F. Pires, "Long term and large scale time synchronization in wireless sensor networks," *Comput. Commun.*, vol. 37, pp. 77–91, Jan. 2014.

[23] R. Shahzadi, S. M. Anwar, F. Qamar, M. Ali, J. J. Rodrigues, and M. Alnowami, "Secure EEG signal transmission for remote health monitoring using optical chaos," *IEEE Access*, vol. 7, pp. 57769–57778, 2019.

[24] A. J. Morales, V. Vavilala, R. M. Benito, and Y. Bar Yam, "Global patterns of synchronization in human communications," *J. Roy. Soc. Interface*, vol. 14, no. 128, p. 1048, 2017.

[25] K. Hirosawa, S. Kittaka, Y. Oishi, F. Kannari, and T. Yanagisawa, "Phase locking in a Nd:YVO$_4$ waveguide laser array using Talbot cavity," *Opt. Exp.*, vol. 21, no. 21, pp. 24952–24961, 2013.

[26] A. S. Karavaev, Y. M. Ishbulatov, V. I. Ponomarenko, B. P. Bezruchko, A. R. Kiselev, and M. D. Prokhorov, "Autonomic control is a source of dynamical chaos in the cardiovascular system," *Chaos*, vol. 29, no. 12, 2019, Art. no. 121101.

[27] F. Lounis, A. Boukabou, and A. Soukkou, "Implementing high-order chaos control scheme for cardiac conduction model with pathological rhythms," *Chaos Solitons Fractals*, vol. 132, Mar. 2020, Art. no. 109581.

[28] S. Scarsoglio, F. Cazzato, and L. Ridolfi, "From time-series to complex networks: Application to the cerebrovascular flow patterns in atrial fibrillation," *Chaos*, vol. 27, no. 9, 2017, Art. no. 093107.

[29] K. Mohanchandra, S. Saha, and K. S. Murthy, "Evidence of chaos in EEG signals: An application to BCI," in *Advances in Chaos Theory and Intelligent Control*. Cham, Switzerland: Springer, pp. 609–625, 2016.

[30] J. E. Jacob, A. Cherian, K. Gopakumar, T. Iype, D. G. Yohannan, and K. P. Divya, "Can chaotic analysis of electroencephalogram aid the diagnosis of encephalopathy?" *Neurol. Res. Int.*, vol. 2018, May 2018, Art. no. 8192820.

[31] W. Wang *et al.*, "Fixed-time synchronization of fractional order memristive MAM neural networks by sliding mode control," *Neurocomputing*, vol. 401, pp. 364–376, Aug. 2020.

[32] M. Wang, Z. Zhou, Z. Li, and Y. Zeng, "An adaptive denoising algorithm for chaotic signals based on improved empirical mode decomposition," *Circuits Syst. Signal Process.*, vol. 38, no. 6, pp. 2471–2488, 2019.

[33] G. Kolumban, M. P. Kennedy, and L. O. Chua, "The role of synchronization in digital communications using chaos. II. Chaotic modulation and chaotic synchronization," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 45, no. 11, pp. 1129–1140, Nov. 1998.

[34] G. Kaddoum, F. Gagnon, and F. D. Richardson, "Design of a secure multi-carrier DCSK system," in *Proc. Int. Symp. Wireless Commun. Syst.*, Paris, France, 2012, pp. 964–968.

[35] A. Griffith, A. Pomerance, and D. J. Gauthier, "Forecasting chaotic systems with very low connectivity reservoir computers," *Chaos*, vol. 29, no. 12, 2019, Art. no. 123108.

[36] K. M. Cuomo, A. V. Oppenheim, and S. H. Strogatz, "Synchronization of Lorenz-based chaotic circuits with applications to communications," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 40, no. 10, pp. 626–633, Oct. 1993.

[37] G. Kaddoum, D. Fournier-Prunaret, P. Chargé, and D. Roviras, "Chaos synchronization between Lorenz systems using two numerical integration methods," in *Proc. Conf. Nonlinear Dyn. Electron. Syst.*, Tokushima, Japan, 2014, pp. 91–94.

[38] A. N. Miliou, I. P. Antoniades, S. G. Stavrinides, and A. N. Anagnostopoulos, "Secure communication by chaotic synchronization: Robustness under noisy conditions," *Nonlinear Anal. Real World Appl.*, vol. 8, no. 3, pp. 1003–1012, 2007.

[39] A. Varghese and D. Tandur, "Wireless requirements and challenges in industry 4.0," in *Proc. Int. Conf. Contemp. Comput. Informat. (IC3I)*, Mysore, India, 2014, pp. 634–638.

[40] M. Alawida, A. Samsudin, and W. H. Alshoura, "Enhancing one-dimensional chaotic map based on bitstream dividing mode," in *Proc. 8th Int. Conf. Softw. Comput. App.*, 2019, pp. 130–134.

[41] C.-J. Cheng, "Robust synchronization of uncertain unified chaotic systems subject to noise and its application to secure communication," *Appl. Math. Comput.*, vol. 219, no. 5, pp. 2698–2712, 2012.

[42] C. Zhou and J. Kurths, "Noise-induced phase synchronization and synchronization transitions in chaotic oscillators," *Phys. Rev. Lett.*, vol. 88, no. 23, 2002, Art. no. 230602.

[43] O. I. Moskalenko, A. E. Hramov, A. A. Koronovskii, and A. A. Ovchinnikov, "Effect of noise on generalized synchronization of chaos: Theory and experiment," *Eur. Phys. J. B*, vol. 82, pp. 69–82, Jun. 2011.

[44] S. S. Haykin, *Communication Systems*, 3rd ed. New York, NY, USA: Wiley, 1994.

[45] J. G. Proakis, *Digital Communications*. Singapore: McGraw-Hill, 1983.

[46] S. Çiçek, A. Ferikoğlu, and I. Pehlivan, "A new 3D chaotic system: Dynamical analysis, electronic circuit design, active control synchronization and chaotic masking communication application," *Optik*, vol. 127, no. 88, pp. 4024–4030, 2016.

[47] D. Wu, H. Du, G. Wen, and J. Liü, "Fixed-time synchronization control for a class of master–slave systems based on homogeneous method," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 66, no. 9, pp. 1547–1551, Sep. 2019.

[48] Y. Liu, L. Li, and Y. Feng, "Finite-time synchronization for high-dimensional chaotic systems and its application to secure communication," *J. Comput. Nonlinear Dyn.*, vol. 11, no. 5, 2016, Art. no. 051028.

[49] J. Kim and M. Jin, "Synchronization of chaotic systems using particle swarm optimization and time-delay estimation," *Nonlinear Dyn.*, vol. 86, no. 3, pp. 2003–2015, 2016.

[50] X. Gao, M. Cheng, L. Deng, M. Zhang, S. Fu, and D. Liu, "Robust chaotic-shift-keying scheme based on electro-optical hybrid feedback system," *Opt. Exp.*, vol. 28, no. 8, pp. 10847–10858, 2020.

[51] T. Schimming, H. Dedieu, and M. Hasler, "Noise filtering in chaos-based communication," in *Chaotic Electronics in Telecommunications*, CRC Press, 2018, pp. 221–253.

[52] X. Cao, L. Liu, Y. Cheng, and X. Shen, "Towards energy-efficient wireless networking in the big data era: A survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 1, pp. 303–332, 1st Quart., 2018.

[53] L. Bariah *et al.*, "A prospective look: Key enabling technologies, applications and open research topics in 6G networks," 2020. [Online]. Available: arXiv:2004.06049.

[54] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *J. Comput. Phys.*, vol. 378, pp. 686–707, Feb. 2019.

[55] K. Yeo and I. Melnyk, "Deep learning algorithm for data-driven simulation of noisy dynamical system," *J. Comput. Phys.*, vol. 376, pp. 1212–1231, Jan. 2019.

[56] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[57] J. Pathak, B. Hunt, M. Girvan, Z. Lu, and E. Ott, "Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach," *Phys. Rev. Lett.*, vol. 120, Jan. 2018, Art. no. 024102.

[58] P. R. Vlachas, W. Byeon, Z. Y. Wan, T. P. Sapsis, and P. Koumoutsakos, "Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks," *Proc. Roy. Soc. A*, vol. 474, May 2018, Art. no. 20170844.

[59] K. Yeo, "Short note on the behavior of recurrent neural network for noisy dynamical system," 2019. [Online]. Available: arXiv:1904.05158.

[60] F. Weninger *et al.*, "Speech enhancement with LSTM recurrent neural networks and its application to noise-robust ASR," in *Proc. Int. Conf. Latent Variable Anal. Sig. Separation*, Liberec, Czechia, 2015, pp. 91–99.

[61] E. Balevi and J. G. Andrews, "Deep learning-based channel estimation for high-dimensional signals," 2019. [Online]. Available: arXiv:1904.09346.

[62] G. Kaddoum, "Wireless chaos-based communication systems: A comprehensive survey," *IEEE Access*, vol. 4, pp. 2621–2648, 2016.

[63] O. Guillen-Fernandez, A. Meléndez-Cano, E. Tlelo-Cuautle, J. C. Núñez-Pérez, and J. de Jesus Rangel-Magdaleno, "On the synchronization techniques of chaotic oscillators and their FPGA-based implementation for secure image transmission," *PLoS ONE*, vol. 14, no. 2, 2019, Art. no. e0209618.

[64] N. A. A. Fataf and M. R. M. Said, "Generalized projective series synchronization between chaotic systems and its application," *J. Fundam. Appl. Sci.*, vol. 9, no. 3S, pp. 294–307, 2017.

[65] M. A. Siddiqi, H. Yu, and J. Joung, "5G ultra-reliable low-latency communication implementation challenges and operational issues with IoT devices," *Electronics*, vol. 8, no. 9, p. 981, 2019.

[66] L. Jin, Y. Zhang, and L. Li, "One-to-many chaotic synchronization with application in wireless sensor network," *IEEE Commun. Lett.*, vol. 17, no. 9, pp. 1782–1785, Sep. 2013.

[67] M. Zhong, Y. Yang, H. Yao, X. Fu, O. A. Dobre, and O. Postolache, "5G and IoT: Towards a new era of communications and measurements," *IEEE Instrum. Meas. Mag.*, vol. 22, no. 6, pp. 18–26, Dec. 2019.

[68] I. J. Goodfellow *et al.*, "Generative adversarial networks," in *Advances in Neural Information Processing Systems*, vol. 3. Red Hook, NY, USA: Curran, 2014, pp. 2672–2680.

[69] V. Veen, A. Jalal, M. Soltanolkotabi, E. Price, S. Vishwanath, and A. G. Dimakis, "Compressed sensing with deep image prior and learned regularization," 2018. [Online]. Available: arXiv:1806.06438.

[70] K. M. Cuomo, A. V. Oppenheim, and S. H. Isabelle, *Spread Spectrum Modulation and Signal Masking Using Synchronized Chaotic Systems*. Fort Belvoir, VA, USA: Defense Techn. Inf. Center, 1992.

[71] J. Liu, Y. Sun, X. Xu, and U. S. Kamilov, "Image restoration using total variation regularized deep image prior," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Brighton, U.K., 2019, pp. 7715–7719.

[72] A. Maybhate, R. E. Amritkar, and D. R. Kulkarni, "Estimation of initial conditions and secure communication," *Int. J. Bifurc. Chaos*, vol. 13, no. 10, pp. 3079–3084, 2003.

[73] D. C. Soriano, R. Attux, J. M. Romano, M. B. Loiola, and R. Suyama, "Denoising chaotic time series using an evolutionary state estimation approach," in *Proc. IEEE Comput. Intell. Control Autom.*, Paris, France, Apr. 2011, pp. 116–122.

[74] B. Peng, L. Bo, Z. Fu-Yi, and L. Wang, "Differential evolution algorithm-based parameter estimation for chaotic systems," *Chaos Solitons Fractals*, vol. 39, no. 5, pp. 2110–2118, 2009.

[75] Y. Peng, K. Sun, S. He, and X. Yang, "Parameter estimation of a complex chaotic system with unknown initial values," *Eur. Phys. J. Plus*, vol. 133, no. 8, p. 305, 2018.

[76] Z. Hua and Y. Zhou, "One-dimensional nonlinear model for producing chaos," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 65, no. 1, pp. 235–246, Jan. 2018.

[77] M. Mobini and M. R. Zahabi, "Multivariable optimisation approach for power allocation in OFDM-DCSK system," *IET Commun.*, vol. 13, no. 13, pp. 1869–1876, Aug. 2019.

[78] G. Chevalier. *Signal forecasting With a Sequence-to-Sequence (seq2seq) RNN Model in TensorFlow*. [Online]. Available: https://github.com/guillaume-chevalier/seq2seq-signal-prediction

[79] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014. [Online]. Available: arXiv:1412.3555.

[80] R. Jozefowicz, W. Zaremba, and I. Sutskever, "An empirical exploration of recurrent network architectures," in *Proc. 32nd Int. Conf. Mach. Learn.*, Jun. 2015, pp. 2342–2350.

[81] S. Abrahams, D. Hafner, E. Erwitt, and A. Scarpinelli, *TensorFlow for Machine Intelligence: A Hands-On Introduction to Learning Algorithms*. Santa Rosa, CA, USA: Bleeding Edge Press, Nov. 2016.

[82] D. P. Kingma and J. L. Ba, "ADAM: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent.*, 2015, pp. 1–15.

[83] M. Madond and T. Gibbons, "Learning and modeling chaos using lstm recurrent neural networks," in *Proc. Midwest Instruct. Comput. Symp. (MICS)*, 2018.

MAJID MOBINI received the M.Sc. degree in communications engineering (with First Rank) from the Department of Electrical and Computer Engineering, Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran, in 2013, and the Ph.D. degree in communications engineering from the Babol Noshirvani University of Technology, in 2019. He was a Visiting Scholar with Cyberspace Research Institute, Shahid Beheshti University, Tehran, from 2013 to 2014. He is currently a Research Associate with the Department of Electrical and Computer Engineering, Babol Noshirvani University of Technology. His current research interests include chaos-based wireless Communication, optimization, artificial intelligence, Internet of Things, cyber-physical systems, machine vision, biomedical instruments, and interdisciplinary researches. He is a Reviewer of the IEEE ACCESS and the *Journal of Soft Computing and Information Theory*.

GEORGES KADDOUM (Senior Member, IEEE) received the bachelor's degree in electrical engineering from the École Nationale Supérieure de Techniques Avancées (ENSTA Bretagne), Brest, France, the M.S. degree in telecommunications and signal processing (circuits, systems, and signal processing) from the Université de Bretagne Occidentale and Telecom Bretagne (ENSTB), Brest, in 2005, and the Ph.D. degree (Hons.) in signal processing and telecommunications from the National Institute of Applied Sciences (INSA), University of Toulouse, Toulouse, France, in 2009. He is currently an Associate Professor and a Tier 2 Canada Research Chair with the École de Technologie Supérieure (ÉTS), Université du Québec, Montréal, Canada. In 2014, he was awarded the ÉTS Research Chair in physical-layer security for wireless networks. Since 2010, he has been a Scientific Consultant in the field of space and wireless telecommunications for several U.S. and Canadian companies. He has published over 200 journal and conference papers and has two pending patents. His recent research activities cover mobile communication systems, modulations, security, and space communications and navigation. He received the Best Papers Awards at the 2014 IEEE International Conference on Wireless and Mobile Computing, Networking, Communications, with three coauthors, and at the 2017 IEEE International Symposium on Personal Indoor and Mobile Radio Communications, with four coauthors. Moreover, he received IEEE Transactions on Communications Exemplary Reviewer Award in 2015, 2017, and 2019. In addition, he received the Research Excellence Award of the Université du Québec in 2018. In the year 2019, he received the Research Excellence Award from the ÉTS in recognition of his outstanding research outcomes. He is currently serving as an Associate Editor for IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY and IEEE COMMUNICATIONS LETTERS.