IDENTIFICATION AND SYNCHRONIZATION OF CHAOTIC AND HYPERCHAOTIC SYSTEMS USING LYAPUNOV STABILITY THEORY

KEVIN HERMAN MURARO GULARTE

TESE DE DOUTORADO EM ENGENHARIA DE SISTEMAS ELETRÔNICOS E DE AUTOMAÇÃO

DEPARTAMENTO DE ENGENHARIA ELÉTRICA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

IDENTIFICATION AND SYNCHRONIZATION OF CHAOTIC AND HYPERCHAOTIC SYSTEMS USING LYAPUNOV STABILITY THEORY

IDENTIFICAÇÃO E SINCRONIZAÇÃO DE SISTEMAS CAÓTICOS E HIPERCAÓTICOS USANDO A TEORIA DE ESTABILIDADE DE LYAPUNOV

KEVIN HERMAN MURARO GULARTE

ORIENTADOR: JOSÉ ALFREDO RUIZ VARGAS, PROF. DR.

TESE DE DOUTORADO EM ENGENHARIA DE
SISTEMAS ELETRÔNICOS E DE AUTOMAÇÃO

PUBLICAÇÃO: PGEA.TD-181/22

BRASÍLIA/DF: DEZEMBRO - 2021

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

IDENTIFICATION AND SYNCHRONIZATION OF CHAOTIC AND HYPERCHAOTIC SYSTEMS USING LYAPUNOV STABILITY THEORY

KEVIN HERMAN MURARO GULARTE

TESE DE DOUTORADO SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA ELÉTRICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR.

APROVADA POR:

Prof. Dr. José Alfredo Ruiz Vargas – ENE/UnB
Orientador

Prof. Dr. Eduardo Stockler Tognetti – ENE/UnB
Membro Interno

Prof. Dr. Elmer Rolando Llanos Villarreal – UFERSA
Membro Externo

Prof. Dr. Bismark Claure Torrico – UFC
Membro Externo

BRASÍLIA, 10 DE DEZEMBRO DE 2021.

FICHA CATALOGRÁFICA

GULARTE, KEVIN HERMAN MURARO
Identification and Synchronization of Chaotic and Hyperchaotic Systems Using Lyapunov Stability Theory [Distrito Federal] 2021.
xvii, 247p., 210 x 297 mm (ENE/FT/UnB, Doutor, Engenharia de Sistemas Eletrônicos e de Automação, 2021).
Tese de doutorado – Universidade de Brasília, Faculdade de Tecnologia.
Departamento de Engenharia Elétrica
1. Teoria de Estabilidade de Lyapunov 2. Sistemas Caóticos
3. Identificação de Sistemas 4. Comunicação Segura
I. ENE/FT/UnB II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

GULARTE, K. H. M. (2021). Identification and Synchronization of Chaotic and Hyperchaotic Systems Using Lyapunov Stability Theory. Tese de doutorado em Engenharia de Sistemas Eletrônicos e de Automação, Publicação PGEA.TD-181/22, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 247p.

CESSÃO DE DIREITOS

AUTOR: Kevin Herman Muraro Gularte
TÍTULO: Identification and Synchronization of Chaotic and Hyperchaotic Systems Using Lyapunov Stability Theory.
GRAU: Doutor ANO: 2021

Kevin Herman Muraro Gularte

Departamento de Engenharia Elétrica (ENE) - FT
Universidade de Brasília (UnB)
Campus Darcy Ribeiro
CEP 70919-970 - Brasília - DF - Brasil

ABSTRACT

Title: Identification and Synchronization of Chaotic and Hyperchaotic Systems Using Lyapunov Stability Theory
Author: Kevin Herman Muraro Gularte
Supervisor: José Alfredo Ruiz Vargas, Prof. Dr.
Graduate Program in Electronic and Automation Systems Engineering
Brasília, December 10th, 2021

The identification and synchronization of chaotic systems are an active topic in the most diverse areas of knowledge. Identification and synchronization schemes of chaos and hyperchaos usually are designed to ensure the convergence of the identification and synchronization errors to a small neighborhood of zero due to the absence of disturbances in the stability analysis. Thus, many works present several recent contributions, but many show possibilities for improvement. Motivated by options for improvement, this doctoral thesis proposes different schemes for identifying and synchronizing chaotic and hyperchaotic systems. Firstly, a method based on neural networks is proposed to identify general chaotic systems. This scheme is unified because it allows parallel and serial-parallel identification models. Based on Lyapunov theory, it is proved that estimation errors are bounded, even in the presence of bounded disturbances. The method allows an adjustment of the transient irrespective of the steady error. A welding and a hyperchaotic financial system were simulated to validate the theoretical. The second contribution is an algorithm for synchronizing a chaotic Lorenz system. The algorithm is based on Lyapunov theory and requires only a scalar control signal for the synchronization, even disturbances are present in all states. The third contribution considers the synchronization of a chaotic system. A scalar control law was designed to synchronize this system based on Lyapunov theory and considering disturbances in all states. It was shown that the synchronization errors converge to a bounded region. Also, a corresponding electronic circuit was implemented to validate the proposed scheme in a realistic situation. In the sequence, the synchronization of a Lü hyperchaotic system is considered. The design of the synchronization scheme is challenging because it is a 4D system. Thus, a reverse engineering procedure determined that only one scalar control signal is suitable. Based on Lyapunov theory, it was proved that a scalar control in the second state allows ensuring a convergent synchronization error for a bounded region. Finally, the projective synchronization of a hyperchaotic system was considered to extend the application to more general situations. This system is particularly challenging due to its structural complexity, and by using reverse engineering, the procedure determined that only two control signals would be necessary. It was possible to show through Lyapunov theory that the synchronization error is bounded in finite-time. Extensive simulations were performed to validate the proposed synchronizations scheme, including secure communication cases.

Keywords: Lyapunov Stability Theory, Chaotic Systems, Systems Identification, Secure Communication.

RESUMO ESTENDIDO

Título: Identificação e Sincronização de Sistemas Caóticos e Hipercaóticos Usando a Teoria de Estabilidade de Lyapunov
Autor: Kevin Herman Muraro Gularte
Orientador: José Alfredo Ruiz Vargas, Prof. Dr.
Programa de Pós-Graduação em Engenharia de Sistemas Eletrônicos e de Automação
Brasília, 10 de dezembro de 2021

A identificação e sincronização de sistemas caóticos têm tido aplicações nas mais diversas áreas do conhecimento, como engenharia, medicina, economia e biologia. Geralmente os esquemas de identificação e sincronização de sistemas caóticos objetivam garantir que os erros de identificação e sincronização tendam a zero, ou que pelo menos sejam limitados. Para tanto, utiliza-se a Teoria da Estabilidade de Lyapunov. Em muitos trabalhos, é comum que algumas considerações simplificadoras sejam feitas para facilitar esta análise de estabilidade. Por exemplo, dentre essas simplificações assume-se a inexistência de distúrbios na análise de convergência e estabilidade. Assim, muitos trabalhos apresentam diversas contribuições recentes, mas muitos deles apresentam possibilidades de melhoria. Na área de identificação, a principal possibilidade de melhoria encontrada foi à falta de um parâmetro de projeto que permitisse controlar a duração do transiente. Na área de sincronização caótica, poucos trabalhos foram encontrados sobre a sincronização subatuada de sistemas caóticos e, nenhum, de sincronização projetiva subatuada de sistemas hipercaóticos que considerassem a presença de distúrbios. Desta forma, nesta tese de doutorado, pretende-se propor esquemas de identificação e sincronização de sistemas caóticos que preencham as lacunas mencionadas. Assim, são apresentadas contribuições originais para o estado da arte na área identificação e sincronização de sistemas caóticos e hipercaóticos subatuados.

No capítulo 1 há a organização do trabalho, motivações, contribuições, objetivos e publicações. Considera-se no capítulo 2 a identificação, uma vez que para se obter um controle com elevado desempenho se requer geralmente bons modelos. Então, é proposto um esquema baseado em redes neurais para a identificação de sistemas caóticos gerais. Este esquema é unificado no sentido que permite a utilização de modelos para identificação paralelos e serie-paralelos. Convêm ressaltar que modelos paralelos são especialmente indicados quando se requer estimação de estados que não são disponíveis para medida. Com base na teoria de estabilidade de Lyapunov é provado que os erros de estimação são limitados e que podem ser reduzidos arbitrariamente a partir de parâmetros de projeto. Adicionalmente, o método permite um ajuste da duração do transiente de forma independente do ajuste do erro em regime. Esta característica torna o esquema especialmente relevante em situações reais onde a qualidade do controle está fortemente relacionada com o ajuste destes parâmetros. Dois sistemas caóticos de grande relevância são utilizados para validar os resultados: um sistema de soldagem e um sistema hipercaótico financeiro.

O capítulo 3 consiste na proposição de um algoritmo para sincronização subatuada do sistema caótico de Lorenz. Esse algoritmo é definido com base na teoria de estabilidade de Lyapunov e se caracteriza por necessitar apenas de um sinal de controle escalar para implementar a sincronização. Outra vantagem comparativa em relação ao que existe na literatura é a consideração da presença de distúrbios em todos os estados para a análise de convergência e estabilidade. Objetivando-se ressaltar a aplicação do algoritmo proposto, considera-se o caso de comunicação segura na presença de distúrbios. O capítulo 4 considera a sincronização de um sistema caótico de forma circuital. Foi projetada uma lei de controle escalar para sincronizar este sistema usando-se a teoria de estabilidade de Lyapunov e considerando distúrbios em todos os estados. Desta forma foi mostrado que os erros de sincronização convergem para uma região limitada cujo raio depende dos distúrbios em todos os estados. Para validar esta proposta em uma situação realista, foi implementado o circuito eletrônico correspondente e testado. Os resultados mostraram que o sincronizador é robusto na presença de distúrbios e que o algoritmo proposto poder ser utilizado com confiabilidade na codificação e decodificação de mensagens.

No capítulo 5 é considerada a sincronização de um sistema hipercaótico de Lü. Esse sistema é interessante, pois tem uma dinâmica mais rica que os sistemas caóticos anteriormente estudados por se tratar de um sistema com dois exponentes de Lyapunov positivos. Logo, esse sistema é mais adequado para comunicação segura. Contudo, por se tratar de um sistema 4D o projeto do esquema de sincronização é desafiador. Desta forma, pelo uso de um procedimento de engenharia reversa, foi determinado que seria necessária somente uma atuação no segundo estado. Definida, então, a forma de atuação e sua localização, com base na teoria de estabilidade de Lyapunov foi provado que um controle escalar no segundo estado permite assegurar um erro de sincronização convergente para uma região limitada cujo raio depende, entre outros, dos distúrbios. O esquema foi testado através de extensas simulações e implementado através de eletrônica analógica. No capítulo 6 objetivando-se estender a aplicação a situações mais gerais, é considerada a sincronização projetiva em tempo finito de um sistema hipercaótico. Este sistema foi proposto em 2019 e é particularmente desafiador devido à sua complexidade estrutural. Com base em um procedimento de engenharia reversa foi determinado que fosse necessária apenas dois sinais de controle. Usando-se dois sinais de controle, no primeiro e quarto estado, foi possível mostrar através da teoria de estabilidade de Lyapunov que o erro de sincronização é convergente em tempo finito para uma região limitada, mesmo com a presença de distúrbios em todos os estados. Extensivas simulações foram realizadas empregando Matlab/Simulink para validar o esquema proposto. No capítulo 7 se estabelece as conclusões e trabalhos futuros. No primeiro apêndice há uma seção de conhecimentos preliminares e no segundo há os códigos utilizados nas simulações.

Palavras-chave: Teoria de Estabilidade de Lyapunov, Sistemas Caóticos, Identificação de Sistemas, Comunicação Segura.

SUMMARY

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

V	: Lyapunov function candidate
W^*	: matrix of optimal weights
$\hat{W}$	: estimation of ideal weight matrix W^*
$\tilde{W}$	: estimation error of ideal weight matrix W^*
e	: synchronization error
θ	: radius of the compact set
x	: n-dimensional state vector
$\hat{x}$	: estimation of the n-dimensional state vector
$\tilde{x}$	: estimation error of the n-dimensional state vector
u	: m-dimensional admissible input or control vector
σ	: regressor vector
$s(.)$	: sigmoidal function
tr	: trace
sup	: supreme
max	: maximum
min	: minimum
λ_{min}	: smallest of the eigenvalues
I	: identity matrix
$\Re$	: set of real numbers

Acronyms

NNs	: Neural Networks
MLP	: Multilayer Perceptron
RBFs	: Radial Basis Function Neural Networks
LPNNs	: Linearly Parameterized Neural Networks
HONNs	: Higher-Order Neural Networks
RMSE	: Root Mean Square of State Errors
GMAW	: Gas Metal Arc Welding

1 INTRODUCTION

1.1 THESIS MOTIVATION

Many works in the area of control of uncertain dynamic systems have been done in linear models. However, this subject has already been extensively studied, and many of these models are unsuitable for generalized applications since nonlinearity is much more common in nature. Furthermore, most linear models are simplifications or approximations that may have applicability limitations. For this reason, nonlinear models have been increasingly demanded. Hence, the first motivation of this work is to study nonlinear models.

The models can be subdivided into white, gray, and black boxes [1]. The white box models are those in which the parameters of a model are known. In the gray box models, not all model parameters are known. In the black box, none of the model parameters are known. Historically, the study of white-box models has been extensively accomplished, but also, in a growing way, the gray and black box models have been considered. The estimation of the system parameters has been frequently used when not all systems information is known [2]. Hence, the area of systems identification has had great relevance and, particularly, identification based on neural networks [3]. A case similar to the identification problem is the observation problem, in which not all states are available for measurement but can be estimated using estimation techniques. Another interesting case to study of control area, where closed-loop identification is performed and, therefore, there is feedback [4]. All these cases have interesting applications that need to be investigated. So, the second motivation of this work lies in the design of identification, observation, and control schemes.

Neural networks are used to approximate unknown nonlinearities in a system. They satisfy the universal approximation condition on a compact domain, allowing unknown maps to be approximated with an arbitrary degree of precision if a suitable structure is provided for the neural model [5]. Neural networks have the advantage of relatively fast implementation and auto-learning, that is, the ability to rely on historical samples to learn. In addition, they are adaptive since neural networks can be used in online applications without needing to have their architecture changed with each update. The problem in these applications is that the residual state error frequently depends on the network structure, which can present a problem. However, Lyapunov stability theory can be used to overcome this drawback since weight adjustment laws based on Lyapunov direct method ensure the boundedness of the estimates. In addition, a suitable choice for identification, observation, and control models, based on the Lyapunov analysis, ensures the convergence of the residual state errors to an arbitrary neighborhood of the origin. The advantage of this framework is that irrespective of how the model is constructed, it is possible to have small residual approximation errors, even in the presence of bounded disturbances. This may arise, for example, due to changes in dynamics due to faults or equipment aging. Therefore, the motivation for using artificial neural networks in this work is that they allow us to approximate unknown nonlinearities and via Lyapunov theory, and it is possible to ensure that the errors are bounded, even in the presence of unknowns.

In the estimation process, there is a moment when the neural networks learning process occurs, called the transient regime, and a moment when the neural network has stabilized, and there are no significant changes in the residual state error, called the permanent regime [6]. Controlling the duration of the transient regime may be helpful for practical purposes since certain controllers have a very short transient regime and may have problems of chattering. Other controllers have a very long transient regime and may have bad performance.

Additionally, once it is possible to estimate the parameters of a system, it is possible to design synchronization schemes between two systems in which the parameters are known, or the values of those parameters have been estimated [7]. An advantage of chaotic systems is that one can have the same structure as the chaotic system that even so, the states trajectories over time can be different depending on the initial conditions. This is because, by definition, chaotic systems are sensitive to initial conditions [8].

Also, chaotic systems exhibiting aperiodic behavior are relatively unpredictable from other systems, so they are usually interesting systems used in a cryptography system. Because they can present the same structure in the system of differential equations, depending on the chaotic system, it is possible to synchronize with the number of control signals less than the number of states in the dynamic system. A consequence of structurally simple controls is to lower costs and facilitate physical implementations. In literature, there have been few studies on this subject. Therefore there is much room for new academic studies.

1.2 THESIS OBJECTIVES

1.2.1 General objective

Based on the Lyapunov stability theory, develop theoretical models of:

1) identification of chaotic systems using artificial neural networks, with the possibility of adjusting the residual state errors and the regime transient duration from different design parameters; and

2) underactuated synchronization of chaotic and hyperchaotic systems, where the syn-

chronization error is bounded.

1.2.2 Specific objectives

- Make the adaptive open-loop identification of a financial hyperchaotic.
- Consider the cases of parallel and semi-parallel identification.
- Extend the synchronizations schemes to the secure communication case, where the message is encoded and decoded.
- Proposed a synchronization scheme to the projective synchronization case.
- Perform validations of the proposed schemes through simulations, applications using analog electronics, or physical implementation.
- Perform a practical application of an adaptive identification proposed scheme in a welding system using the online identifier.
- Perform a comparison of an identification algorithm found in literature and the proposed identification scheme.
- Perform a comparison of a synchronization algorithm found in literature and the proposed identification scheme.

1.3 THESIS CONTRIBUTIONS

1.3.1 Main Contributions

- The proposal of an online chaotic identification scheme that uses artificial neural networks that allows an adjustment of the duration of the transient regime from a design parameter that is not related to the size of the residual state error.
- The proposal of underactuated synchronization schemes of chaotic and hyperchaotic systems in which it is theoretically proven that the synchronization error is bounded even in the presence of bounded external and internal disturbances.

1.3.2 Other Contributions

- The proposal of an online identification scheme that uses artificial neural networks that allows an adjustment of the duration of the transient regime from a design parameter that is not related to the size of the residual state error.

- The proposal of an online identification scheme that uses artificial neural networks that allows an adjustment of the duration of the transient regime from a design parameter that is not related to the size of the residual state error.
- The application of the Lyapunov stability theory to find a neural identifier in which the residual state error relates to some design matrices, so as to allow its convergence to a neighborhood of the origin, even if in presence of bounded disturbances.
- It is proven by using the Lyapunov stability analysis that the convergence of the synchronization error is accomplished in finite time.
- Most proposed control schemes are as simple as possible, with a proportional control signal sufficient to synchronize even hyperchaotic systems. This feature facilitates the use of these synchronization and encryption schemes in practical applications.
- The proposal of the probably first underactuated projective synchronization scheme of a hyperchaotic system in which it is theoretically proven that the synchronization error is bounded even in the presence of bounded external and internal disturbances.

1.4 THESIS OVERVIEW

This is a paper-based doctoral thesis. Thus, chapters 2-6 will contain the developments of the works and the respective introductions with the literature review. The Ph.D. thesis is organized as follows. This chapter presents the introduction, motivation, objective, possible contributions, and structure of the proposed work.

In Chapter 2, by using Lyapunov stability theory, an online adaptive neural identification scheme is proposed for a class of nonlinear systems in the presence of bounded disturbances. The proposed algorithm allows 1) to reduce the residual error of state estimation to small values utilizing design matrices; 2) to control the transient time arbitrarily from a design parameter. Simulations were done to demonstrate the effectiveness and efficiency of the proposed learning algorithm. Simulations were performed for chaotic and hyperchaotic systems to demonstrate the effectiveness and efficiency of the proposed learning algorithm. In these simulations, the size of the residual state error and the choice of the transient time were analyzed. The chapter ends with an application: the neural identification of a welding system with chaotic behavior where the size of the residual state error was analyzed.

In Chapter 3, by using the Lyapunov stability theory, an underactuated synchronization scheme of a Lorenz chaotic system in the presence of bounded disturbances is proposed. An application is also made in secure telecommunication. In Chapter 4, based on the Lyapunov stability theory, a chaotic system synchronization scheme in the presence of bounded disturbances is proposed. In this scheme, the dimension of control inputs is less than states. The case of encoding and decoding messages is also analyzed. A physical application in an electrical circuit is made to validate the scheme.

In Chapter 5, by using the Lyapunov stability theory, an underactuated synchronization scheme for secure communication based on Lü hyperchaotic system is proposed. An application is also made in secure telecommunication. There are also simulations considering applications in electronic circuits. In Chapter 6, a synchronization and secure communication scheme based on a hyperchaotic underactuated projective synchronization is proposed, where there are two inputs and four states. It is proved that the synchronization error is bounded through the Lyapunov stability theory. Simulations considering a circuital implementation project were carried out.

Subsequently, Chapter 7 summarizes the theoretical contributions of the research, the results obtained, and suggestions for future research are also discussed. Appendix 1 describes the theoretical basis of the artificial neural networks, learning algorithms, Lyapunov stability theory, and other technical backgrounds supporting the work chapters. Appendix 2 contains the codes in Matlab language used to generate most of the figures and tables in this thesis.

1.5 PUBLICATIONS

During the doctorate period, several papers were recently published [9–21]. The identifications works were published in

- **K. H. M. Gularte** and J. A. R. Vargas, "Open and Closed Loop Neural Identification of Uncertain Systems with Transient Time Adjustment," *2018 IEEE International Conference on Automation/XXIII Congress of the Chilean Association of Automatic Control (ICA-ACCA)*, pp. 1–7, 2018.
- **K. H. M. Gularte**, J. J. M. Chavez, J. A. R. Vargas, and S. C. A. Alfaro, “An adaptive neural identifier with applications to financial and welding systems,” *International Journal of Control, Automation and Systems*, vol. 19, no. 5, pp. 1976–1987, 2021.

The synchronizations works were published in

- **K. H. M. Gularte**, L. N. C. Rêgo, and J. A. R. Vargas, "Scheme for chaos-based encryption and lyapunov analysis," *International Conference on Automation/XXIII Congress of the Chilean Association of Automatic Control (ICA-ACCA)*, pp. 1–7, 2018.

- **K. H. M. Gularte**, A. Zaiter, R. R. Santos, and J. A. R. Vargas, "Sincronização de um sistema sprott k-caótico subatuado baseado em controle proporcional com ganho variável," *XXII Congresso Brasileiro de Automática, 2018, João Pessoa. Anais do XXII Congresso Brasileiro de Automática*, vol. 1, no. 1, 2019.

- **K. H. M. Gularte**, R. R. Santos, W. A. M. Gabalan, and J. A. R. Vargas, "Proposta para comunicação segura baseada em sincronização projetiva caótica," *ANAIS DO 14º SIMPÓSIO BRASILEIRO DE AUTOMAÇãO INTELIGENTE*, Ouro Preto, pp. 2939–2945, 2019.

- **K. H. M. Gularte**, V. V. Graciano, W. A. M. Gabalan, and J. A. R. Vargas, "Esquema de sincronização projetiva baseado em caos e análise de lyapunov," *ANAIS DO 14ºSIMPÓSIO BRASILEIRO DE AUTOMAÇãO INTELIGENTE*, Ouro Preto, pp. 2852–2857, 2019.

- **K. H. M. Gularte**, L. M. Alves, J. A. R. Vargas, J. P. A. Maranhão, G. C. Carvalho, S. C. A. Alfaro, and J. F. A. Romero, "A Chaotic Synchronization scheme for information security," *2019 13th International Conference on Signal Processing and Communication Systems (ICSPCS)*, pp. 1–10, 2019.

- **K. H. M. Gularte**, L. F. Q. Martins, J. A. R. Vargas, J. P. A. Maranhão, W. A. M. Gabalan, and J. F. A. Romero, "A Scheme for Encryption/Decryption based on Hyperchaotic Systems and Lyapunov Theory," *2019 13th International Conference on Signal Processing and Communication Systems (ICSPCS)*, pp. 1–9, 2019.

- **K. H. M. Gularte**, J. C. G. Gómez, J. A. R. Vargas, and R. R. Dos Santos, "Underactuated Synchronization Scheme of a Hyperchaotic Finance System," *14th IEEE International Conference on Industry Applications (INDUSCON)*, pp. 1335-1339, 2021.

- **K. H. M. Gularte**, J. C. G. Gómez, J. A. R. Vargas, and R. R. Dos Santos, "Projective Synchronization and Antisynchronization of Underactuated Systems," *14th IEEE International Conference on Industry Applications (INDUSCON)*, pp. 1317-1322, 2021.

- **K. H. M. Gularte**, J. C. G. Gómez, J. A. R. Vargas, and R. R. Dos Santos, "Chaos-based Cryptography Using an Underactuated Synchronizer," *14th IEEE International Conference on Industry Applications (INDUSCON)*, pp. 1303-1308, 2021.

- **K. H. M. Gularte**, J. C. G. Gómez, M. E. Vizcarra Melgar, and J. A. R. Vargas, "Chaos Synchronization and its Application in Parallel Cryptography," *2021 IEEE 5th Colombian Conference on Automatic Control (CCAC)*, pp. 198-203, 2021.

- **K. H. M. Gularte**, J. C. G. Gómez, M. E. Vizcarra Melgar, and J. A. R. Vargas, "Underactuated 4D-Hyperchaotic System for Secure Communication in the Presence of

Disturbances," *2021 IEEE 5th Colombian Conference on Automatic Control (CCAC)*, pp. 210-215, 2021.

Paper submitted to a Jornal and accepted for publication:

- **K. H. M. Gularte**, L. M. Alves, J. A. R. Vargas, S. C. A. Alfaro, G. C. de Carvalho, and J. F. A. Romero, "Secure Communication Based on Hyperchaotic Underactuated Projective Synchronization," *IEEE Acess*, pp. 1–12, 2021.

2 TECHNICAL BACKGROUND

2.1 MOTIVATION

This chapter introduces the technical background about Lyapunov Stability Theory, neural networks, their properties, and the notation used throughout this Ph.D. thesis. The aim is to provide basic information about the contents used in this thesis.

2.2 DYNAMIC SYSTEMS

The concepts presented in this section are extracted from [22].

Dynamical systems can be modeled by a finite number of coupled first-order ordinary differential equations

$$\begin{aligned} \dot{x}_1 &= f_1(t, x_1, ..., x_n, u_1, ..., u_p) \\ \dot{x}_2 &= f_2(t, x_1, ..., x_n, u_1, ..., u_p) \\ \vdots \quad & \quad \vdots \\ \dot{x}_n &= f_n(t, x_1, ..., x_n, u_1, ..., u_p) \end{aligned} \tag{2.1}$$

where $\dot{x}_i$ denotes the derivative of x_i, with respect to the time variable t, and $u_1, u_2, ..., u_p$ are specified input variables. We call the variables $x_1, x_2, ..., x_p$ are the state variables. They represent the memory that the dynamic system has of its past. The vector notation to write the above equations compactly is used. Define

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_n \end{bmatrix}, \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_p \end{bmatrix}, f(t, x, u) = \begin{bmatrix} f_1(t, x, u) \\ f_2(t, x, u) \\ \vdots \\ \vdots \\ f_n(t, x, u) \end{bmatrix} \tag{2.2}$$

and rewrite the n first-order differential equations as one n-dimensional first-order vector differential equation

$$\dot{x} = f(t, x, u) \tag{2.3}$$

(2.3) is the state equation, and x is the *state* and u is the *input*. Usually, (2.3) is associate with another equation

$$y = h(t, x, u) \tag{2.4}$$

defines a q-dimensional output vector that comprises variables of particular interest in the analysis of the dynamical system, like variables that can be physically measured or variables required to behave in a specified manner. (2.4) is the output equation and equations (2.3) and (2.4) together are the state-space model, or simply the state model. Mathematical models of finite-dimensional physical systems do not always come in the form of a state-space model. However, physical systems often are modeled in this form by carefully choosing the state variables.

$$\dot{x} = f(t, x) \tag{2.5}$$

Working with an unforced state equation does not necessarily mean that the input to the system is zero. It could be that the input has been specified as a given function of time, $u = \gamma(t)$, a given feedback function of the state, $u = \gamma(x)$, or both, $u = \gamma(t, x)$. Substitution of $u = \gamma$ in (2.3) eliminates u and yields an unforced state equation. A particular case of (2.5) arises when the function f does not depend explicitly on t; that is,

$$\dot{x} = f(x) \tag{2.6}$$

in this case, the system is said to be autonomous or time-invariant. The behavior of an autonomous system is invariant to shifts in the time origin since changing the time variable from t to $\tau = t - a$ does not change the right-hand side of the state equation. If the system is not autonomous, it is called nonautonomous or time-varying.

An important concept in dealing with the state equation is the concept of an equilibrium point. A point $x = x^*$ in the state space is said to be an equilibrium point of (2.5) if it has the property that whenever the state of the system starts at x^* it will remain at x^* for all future time. For the autonomous system (2.6), the equilibrium points are the real roots of the equation

$$f(x) = 0 \tag{2.7}$$

An equilibrium point could be isolated; there are no other equilibrium points in its vicinity, or there could be a continuum of equilibrium points.

2.3 MATHEMATICAL PRELIMINARIES

This section provides some fundamental mathematical concepts that are necessary for the work chapters.

2.3.1 Young inequality

By defining that a and b are positive variables, $p > 1$ and $q > 1$, consider the following Young inequality for products

$$ab \leq \frac{a^p}{p} + \frac{b^q}{q} \tag{2.8}$$

where $\frac{1}{p} + \frac{1}{q} = 1$. Assuming p = 2 and q = 2, then

$$ab \leq \frac{a^2}{2} + \frac{b^2}{2} \tag{2.9}$$

Note that this expression can also be deduced from inequality $(a-b)^2 \geq 0$. This is important because, in the case of $p = 2$ and $q = 2$, (2.9) is also correct for any negative real value of a and b. Rewriting (2.9) results

$$\left(a\sqrt{\sigma}\right)\left(\frac{b}{\sqrt{\sigma}}\right) \leq \frac{a^2\sigma}{2} + \frac{b^2}{2\sigma} \tag{2.10}$$

being $\sigma > 0$. More information can be obtained in [23].

2.3.2 Vector Norms

Definition 1 Let $x \in X \subset \Re^n$ be a n-dimensional vector. The p-norm of f is defined by

$$\|x\|_p = \left(\sum_i |x_i|^p\right)^{1/p}, \ for\ p \in [1, \infty) \tag{2.11}$$

Thus, by denoting $p = 1, 2, \infty$, the corresponding normed spaces are called L_1, L_2, L_∞, respectively. In this thesis usually, it is used the case where $p = 2$:

$$\|x\| = \|x\|_2 = \sqrt{\sum_i |x_i|^2} \tag{2.12}$$

$$\|x\|^2 = x^T x, \ x \in \Re^{1\times n} \tag{2.13}$$

By defining a real constant α, using the vector $x \in \Re^n$, and considering $y \in Y \subset \Re^n$ a n-dimensional vector, the followings properties are true:

$$||x|| \geq 0 \tag{2.14}$$

$$||\alpha{\cdot}x|| = |\alpha|{\cdot}||x|| \tag{2.15}$$

$$||x + y|| \leq ||x|| + ||y|| \tag{2.16}$$

More information can be obtained in [5, 24, 25].

2.3.3 Matrix Norms

Definition 2 Let $A, B \in \Re^{n \times n}$ be n-dimensional matrices, and $c \in \Re$. Then, the following properties are valid

$$\|A\| \ \geq 0 \tag{2.17}$$

$$\|cA\| \ \leq |c| \cdot \|A\| \tag{2.18}$$

$$\|Ax\| \ \leq \|A\|_F \cdot \|x\| \tag{2.19}$$

$$\|A + B\| \ \leq \|A\| \ + \|B\| \tag{2.20}$$

$$tr(A + B) = tr(A) + tr(B) \tag{2.21}$$

The Frobenius norm can be defined as:

$$\|A\|_F^2 \ = tr(A^T A) \tag{2.22}$$

Where $tr(A)$ is the trace function. This function returns the sum of diagonal entries of a square matrix. More information can be obtained in [25].

2.3.4 Lyapunov Stability Theory

We present in this section some concepts about Lyapunov stability theory. The following definitions and theorem were extracted from [24].

2.3.4.1 Concepts of Stability

We consider systems described by ordinary differential equations of the form

$$\dot{x} = f\left(t, x\right), \; x\left(t_0\right) = x_0 \tag{2.23}$$

where $x \in \Re^n, f : \tau \times B\left(r\right), \tau = [t_0, \infty)$, and $B\left(r\right) = \left\{x \in \Re^n | \left\|x\right\| < r\right\}$. We assume that f is of such nature that for every $x_0 \in B\left(r\right)$ and every $t_0 \in \Re^+$, (2.23) possesses one and only one solution $x\left(t; t_0; x_0\right)$.

DEFINITION 3 A state x_e is said to be an **equilibrium state** of the system described by (2.23) if

$$f\left(t, x_e\right) \equiv 0 \;\; for\; all \;\; t \geq t_0 \tag{2.24}$$

DEFINITION 4 An equilibrium state x_e is called an **isolated equilibrium state** if there exists a constant $r > 0$ such that $B\left(x_e, r\right) := \left\{x | \left\|x - x_e\right\| < r\right\}$ contains no equilibrium state of (2.23) other than x_e.

DEFINITION 5 The equilibrium state x_e is said to be **stable(in the sense of Lyapunov)** if for arbitrary t_0 and $\varepsilon > 0$ there exists a $\delta\left(\varepsilon, t_0\right)$ such that $|x_0 - x_e| < \delta$ implies $|x\left(t; t_0; x_0\right) - x_e|$ for all $t \geq t_0$.

DEFINITION 6 The equilibrium state x_e is said to be **uniformly stable (u.s)** if it is stable and if $\delta\left(\varepsilon, t_0\right)$ in Definition 5 does not depend on t_0.

DEFINITION 7 The equilibrium state x_e is said to be **asymptotically stable (a.s)** if (i) it is stable, and (ii) there exists a $\delta\left(t_0\right)$ such that $|x_0 - x_e| < \delta\left(t_0\right)$ implies $lim_{t \to \infty} |x\left(t; t_0; x_0\right) - x_e| = 0$. If condition (ii) is satisfied, then the equilibrium state x_e is said to be **attractive**.

DEFINITION 8 The set of all $x_0 \in \Re^n$ such that $x\left(t; t_0; x_0\right) \to x_e$ as $t \to \infty$ for some $t_0 \geq 0$ is called the **region of attraction** of the equilibrium state x_e.

DEFINITION 9 The equilibrium state x_e is said to be **uniformly asymptotically stable (u.a.s)** if (i) it is uniformly stable, (ii) for every $\varepsilon > 0$ and any $t_0 \in \Re^+$, there exist a $\delta_0 > 0$ independent of t_0 and ε and a $T\left(\varepsilon\right) > 0$ independent of t_0, such that $|x\left(t; t_0; x_0\right) - x_e| < \varepsilon$ for all $t > t_0 + T\left(\varepsilon\right)$ whenever $|x_0 - x_e| < \delta_0$.

Definition 10 The equilibrium state x_e is **exponentially stable (e.s)** if there exists an $\alpha > 0$, and for every $\varepsilon > 0$ there exists a $\delta(\varepsilon) > 0$ such that

$$|x(t; t_0; x_0) - x_e| < \varepsilon e^{-\alpha(t-t_0)} \;\; for\ all \;\; t \geq t_0 \tag{2.25}$$

whenever $|x_0 - x_e| < \delta(\varepsilon)$.

Definition 11 The equilibrium state x_e is said to be **unstable** if it is not stable.

When (2.23) have a unique solution for each $x_0 \in \Re^n$ and $t_0 \in \Re^+$, we need the following definitions for the global characterization of solutions.

Definition 12 A solution $x(t; t_0; x_0)$ of (2.23) is **bounded** if there exists a $\beta > 0$ such that $|x(t; t_0; x_0)| < \beta$ for all $t > t_0$, where β may depend on each solution.

Definition 13 The solutions of (2.23) are **uniformly bounded (u.b)** if for any $\alpha > 0$ and $t_0 \in \Re^+$, there exists a $\beta = \beta(\alpha)$ independent of t_0 such that if $|x_0| < \alpha$, then $|x(t; t_0; x_0)| < \beta$ for all $t > t_0$.

Definition 14 The solutions of (2.23) are **uniformly ultimately bounded (u.u.b)** (with bound B) if there exists a $B > 0$ and if corresponding to any $\alpha \geq 0$ and $t_0 \in \Re^+$, there exists a $T = T(\alpha) > 0$ (independent of t_0) such that $|x_0| < \alpha$ implies $|x(t; t_0; x_0)| < B$ for all $t > t_0 + T$.

Definition 15 If $x(t; t_0; x_0)$ is a solution of $\dot{x} = f(t, x)$, then the trajectory $x(t; t_0; x_0)$ is said to be **stable (u.s., a.s., u.a.s., e.s., unstable)** if the equilibrium point $z_e = 0$ of the differential equation

$$\dot{z} = f(t, z + x(t; t_0; x_0)) - f(t, x(t; t_0; x_0)) \tag{2.26}$$

is **stable (u.s., a.s., u.a.s., e.s., unstable, respectively)**.

2.3.4.2 Lyapunov's Direct Method

The stability properties of the equilibrium state or solution of (2.23) can be studied by using the direct method of Lyapunov (also known as Lyapunov's second method). The objective of this method is to answer questions of stability by using the form of $f(t, x)$ in (2.23) rather than the explicit knowledge of the solutions. We start with the following definitions.

Definition 16 A continuous function $\varphi : [0, r] \rightarrow \Re^+$ (or a continuous function $\varphi : [0, \infty) \rightarrow \Re^+$) is said to belong to **class** K, i.e., $\varphi \in K$, if

(i) $\varphi(0) = 0$.

(ii) φ is strictly increasing on $[0, r]$ (or on $[0, \infty)$).

Definition 17 A continuous function $\varphi : [0, \infty) \rightarrow \Re^+$ is said to belong to **class** KR, i.e., $\varphi \in KR$, if

(i) $\varphi\,(0) = 0$.

(ii) φ is strictly increasing on $[0, \infty)$.

(iii) $lim_{r\rightarrow\infty}\varphi(r) = \infty$.

Definition 18 Two functions $\varphi_1, \varphi_2 \in K$ defined on $[0, r]$ (or on $[0, \infty]$) are said to be **of the same order of magnitude** if there exist positive constants k_1, k_2, such that

$$k_1\varphi_1\left(r_1\right) \leq \varphi_2\left(r_1\right) \leq k_2\varphi_2\left(r_1\right), \forall r_1 \in [0, r]\,(or\ \forall r_1 \in [0, \infty]) \tag{2.27}$$

Definition 19 A function $V\left(t, x\right) : \Re^+ \times B(r) \rightarrow \Re$ with $V\left(t, 0\right) = 0, \forall t \in \Re^+$ is **positive definite** if there exists a continuous function $\varphi \in K$ such that $V\left(t, x\right) \geq \varphi(|x|), \forall t \in \Re^+$, $x \in B(r)$ and some $r > 0$. $V\left(t, x\right)$ is called **negative-definite** if $-V\left(t, x\right)$ is positive definite.

Definition 20 A function $V\left(t, x\right) : \Re^+ \times B(r) \rightarrow \Re$ with $V\left(t, 0\right) = 0, \forall t \in \Re^+$ is said to be **positive(negative) semidefinite** if $V\left(t, x\right) \geq 0(V\left(t, x\right) \leq 0)$, for all $t \in \Re^+$ and $x \in B(r)$ for some $r > 0$.

Definition 21 A function $V\left(t, x\right) : \Re^+ \times B(r) \rightarrow \Re$, with $V\left(t, 0\right) = 0, \forall t \in \Re^+$ is said to be **decrescent** if there exists $\varphi \in K$ such that $|V\left(t, x\right)| \leq \varphi\left(|x|\right), \forall t \geq 0$ and $\forall x \in B\left(r\right)$ for some $r > 0$.

Definition 22 A function $V\left(t, x\right) : \Re^+ \times \Re^r \rightarrow \Re$ with $V\left(t, 0\right) = 0, \forall t \in \Re^+$ is said to be **radially unbounded** if there exists $\varphi \in KR$ such that $V\left(t, x\right) \geq \varphi\left(|x|\right)$ for all $x \in \Re^n$ and $t \in \Re^+$.

It is clear from the Definition (22) that if $V(t, x)$ is radially unbounded, it is also positive definite for all $x \in \Re^n$, but the converse is not true.

Let us assume (without loss of generality) that $x_e = 0$ is an equilibrium point of (2.23) and define $\dot{V}$ to be the time-derivative of the function $V\left(t, x\right)$ along the solution of (2.23), i.e.,

$$\dot{V} = \frac{\partial V}{\partial t} + \left(\nabla V\right)^T f\left(t, x\right) \tag{2.28}$$

where $\nabla V = \left[\frac{\partial V}{\partial x_1}, \frac{\partial V}{\partial x_2}, ..., \frac{\partial V}{\partial x_n}\right]^T$ is the gradient of V with respect to x. The second method of Lyapunov is summarized by the following theorem.

Theorem 2.3.1 Suppose there exists a positive definite function $V\left(t, x\right) : \Re^+ \times B(r) \rightarrow \Re$ for some $r > 0$ with continuous first-order partial derivatives with respect to x, t, and $V\left(t, 0\right) = 0, \forall t \in \Re^+$. Then, the following statements are true:

(i) If $\dot{V} \leq 0$, then $x_e = 0$ is **stable**.

(ii) If V is decrescent and $\dot{V} \leq 0$, then $x_e = 0$ is **uniformly stable**.

(iii) If V is decrescent and $\dot{V} < 0$, then x_e is **uniformly asymptotically stable**.

(iv) If V is decrescent and there exist $\varphi_1, \varphi_2, \varphi_3 \in K$ of the same order of magnitude such that

$$\varphi_1\left(|x|\right) \leq V\left(t,x\right) \leq \varphi_2\left(|x|\right), V\left(t,x\right) \leq -\varphi_3\left(|x|\right) \tag{2.29}$$

for all $x \in B(r)$ and $t \in \Re^+$, then $x_e = 0$ is exponentially stable.

In the above theorem, the state x is restricted to be inside the ball $B(r)$ for some $r > 0$. Therefore, the results (i) to (iv) of Theorem 2.3.1 are referred to as local results.

Theorem 2.3.2 Assume that (2.23) possesses unique solutions for all $x_0 \in \Re^n$. If there exists a function $V(t,x)$ defined on $|x| \geq R$ (where R may be large) and $t \in [0,\infty)$ with continuous first-order partial derivatives with respect to x, t and if there exist $\varphi_1, \varphi_2 \in KR$ such that

(i) $\varphi_1\left(|x|\right) \leq V\left(t,x\right) \leq \varphi_2\left(|x|\right)$

(ii) $\dot{V}\left(t,x\right) \leq 0$ for all $|x| \geq R$ and $t \in [0,\infty)$, then, the solutions of (2.23) are uniformly bounded. If in addition there exists $\varphi_3 \in K$ defined on $[0,\infty)$ and

(iii) $\dot{V}\left(t,x\right) \leq \varphi_3\left(|x|\right)$ for all $|x| \geq R$ and $t \in [0,\infty)$ then, the solutions of (2.23) are uniformly ultimately bounded.

2.4 ARTIFICIAL NEURAL NETWORKS

Initially will be shown some concepts about neural networks biological that will ultimately serve as motivation for the artificial neural network, which will be discussed next. The concepts and figures used in this section were taken from [26].

2.4.1 Biological Neural Networks

Neurons, or nerve cells, are the building blocks of the nervous system. Neurons have unique features and structures that differentiate them from other cells. The neuron has three distinct regions:

- Cell body (or soma): provides the support functions and structure of the cell. It collects and processes information received from other neurons.

- Dendrites: are tube-like extensions that repeatedly branch and form a bushy tree around the cell body. They provide the main path on which the neuron receives coming information.
- Axon: the part of the neuron that extends away from the cell body and provides the path over which information travel to other neurons.

The Figure 2.1 (from [26]) shows a biological neuron.

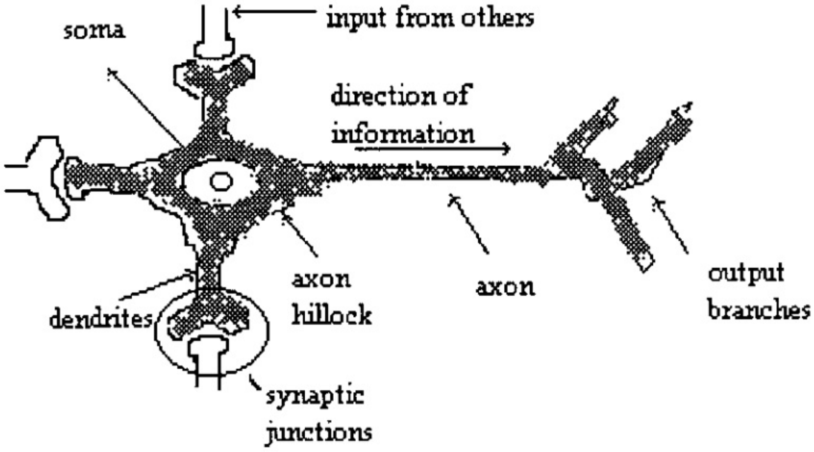

Figure 2.1 – Biological Neuron Scheme.

At the origin of the axon, a nerve impulse is triggered by the cell body in response to the received information. The impulse sweeps along the axon until it reaches the end. The junction point of an axon with a dendrite of another neuron is called a synapse.

2.4.2 Artificial Neural Models

An artificial neural network (ANN) is a massively parallel distributed processor inspired by biological neural networks, storing experimental knowledge and making it available for use. The similarities with the brain are:

- Knowledge is acquired through a learning process.
- Interneuron connectivity named as synaptic weights is used to store this knowledge.

The procedure for the learning process is known as a learning algorithm. Its function is to modify the synaptic weights of the networks in order to attain a prespecified goal. The weights modification provides the traditional method for neural networks design and implementation. The neuron is the fundamental information-processing unit for the operation of a neural network [26, 27]. The Figure 2.2 [27] shows the model of a neuron.

This model has three basic elements:

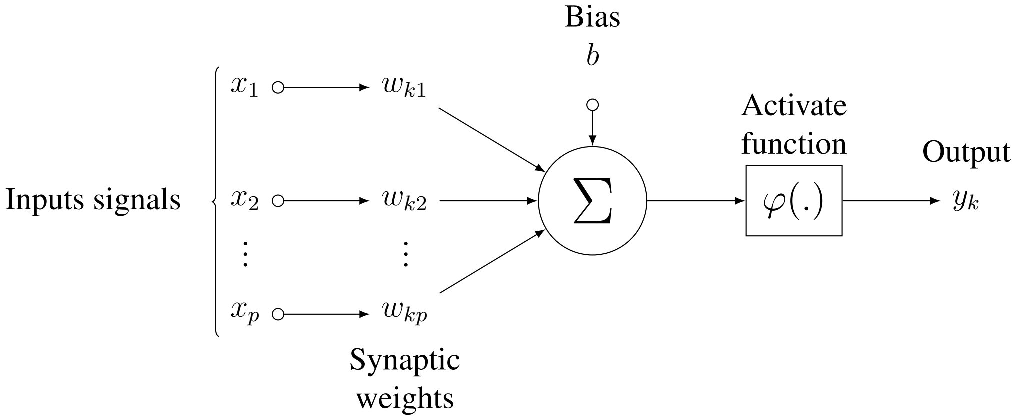

Figure 2.2 – Nonlinear model of a neuron.

- A set of synapses links, each element being characterized by its own weight or strength.
- An adder for summing the inputs signal components, multiplied by the respective synapsis weight.
- A nonlinear activation function transforming the adder output into the output of the neuron.

The neuron scheme presented in Figure 2.2 also includes an externally applied bias or threshold, denoted by b. Bias can increase or decrease the input of the activation function, depending on whether it is positive or negative, respectively.

2.4.3 Linearly Parameterized Neural Networks

Linearly parameterized neural networks (LPNNs) can be expressed mathematically as

$$\rho_{nn}\left(\hat{W}, \zeta\right) = \hat{W}\sigma\left(\zeta\right) \tag{2.30}$$

where $\rho_{nn} : \Re^{L_{nn}} \mapsto \Re^{n}$ is a function, $\hat{W} \in \Re^{n \times L_\rho}$ is a weight matrix, $\zeta \in \Re^{L_\zeta}$ are the inputs of the neural network and $\sigma : \Re^{L_\zeta} \mapsto \Re^{L_\rho}$ is the basis function vector, which can be considered as a nonlinear vector function whose arguments are preprocessed by a scalar function $s(\cdot)$, and $n, L_\rho, L_\zeta, L_{nn}$ are integers strictly positive. Commonly used scalar functions include sigmoid (used in this work), hyperbolic tangent, gaussian, Hardy's, inverse Hardy's multiquadratic [5]. However, this work is only interested in the class of LPNNs for which $\sigma(\cdot)$ is bounded, since in this case results

$$\|\sigma\left(\zeta\right)\| \leq \sigma_0 \tag{2.31}$$

being σ_0 a strictly positive constant.

The class of LPNNs considered in this work includes radial basis function neural networks (RBFs), wavelet networks, high order neural networks (HONNs) [5, 28, 29], and also others linearly parameterized approximators as Takagi-Sugeno fuzzy systems [30]. Universal approximation results in [5, 28–30] indicate that:

PROPERTY 1 Given a constant $\varepsilon_0 > 0$ and a continuous function $f : \Re^{L_\zeta} \mapsto \Re^n$ there exists a weight matrix $\hat{W} = W^*$ and a L_ρ is big enough such that

$$\left\| f(\zeta) - \hat{W}\sigma(\zeta) \right\|_\infty \leq \varepsilon_0 \tag{2.32}$$

where $\hat{W} \in \Gamma$, $\zeta \in \Omega$, $\Gamma = \left\{ \hat{W} | \left\| \hat{W} \right\| \leq \alpha_{\hat{W}} \right\}$, $\alpha_{\hat{W}}$ is a positive constant, $\hat{W}$ is the estimation of W^*, which is an "optimum" matrix, and ε_0 is an approximation, reconstruction or modeling error.

PROPERTY 2 The Output of LPNNs is continuous with respect to their arguments and satisfies the condition of Lipschitz [24], for all $\zeta \in \Omega(\zeta)$, where Ω is a compact set.

2.4.4 Neural Network Structures

The way in which the neurons of a neural network are interconnected determines its structure. The commonly used static neural network structures for system identification are multilayer perceptron, fuzzy systems, radial basis function, and wavelet networks.

2.4.4.1 Multilayer Feedforward Neural Network

They distinguish themselves by the presence of one or more hidden layers whose computation nodes are called hidden neurons. Typically the neurons in each layer have as their inputs the output signals of the preceding layer. If each neuron in each layer is connected to every neuron in the adjacent forward layer, then the neural network is named as fully connected, on the opposite case, it is called partly connected. A multilayer perceptron (MLP) has three distinctive characteristics:

- The activation function of each neuron is smooth as opposed to the hard limit used in the single-layer perceptron. Usually, this nonlinear function is sigmoidal.
- The network contains one or more layers of hidden neurons.
- The network exhibits a high degree of connectivity.

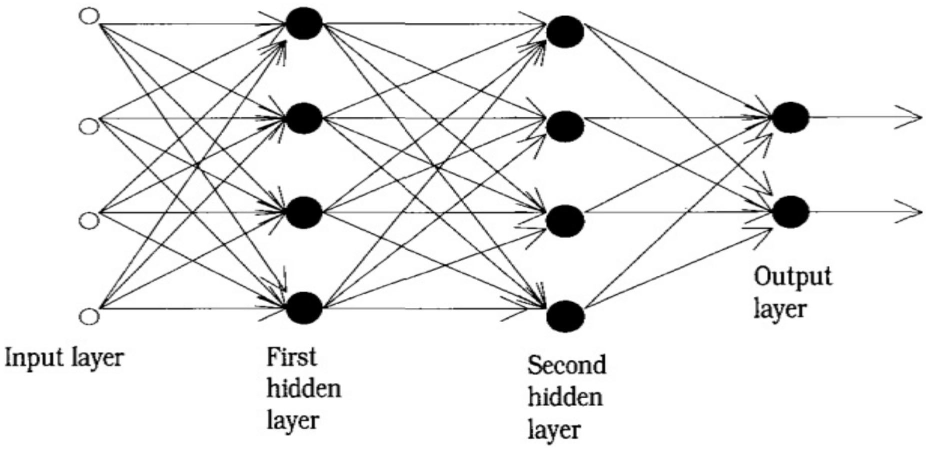

Figure 2.3 – Multilayer perceptron.

In this thesis, in Chapter 2, systems with one hidden layer are used. The Figure 2.3 (from [26]) shows the Multilayer perceptron scheme.

2.4.5 Sigmoidal functions

This section extracts some informations from [31]. The activation functions usually employed in multilayer perceptron are sigmoidal.

Some Sigmoidal functions	
Name	**Formula**
Logistic	$\frac{1}{1+e^{-\gamma x}}, \ \gamma > 0$
Hyperbolic Tangent	$\frac{e^x - e^{-x}}{e^x + e^{-x}}$
Arc-tangent	$arctan(x)$

Table 2.1 – Common Sigmoidal Activation Functions for MLP Networks.

In this work, all activation functions are logistic types.

2.5 CHAOTIC SYSTEMS

The concepts presented in the subsection 2.5.1 were extracted from [8] and the informations presented in subsections 2.5.2-2.5.4 were extracted from [32].

2.5.1 Defining chaos

No definition of the term chaos is universally accepted yet, but almost everyone would agree on the three ingredients used in the following working definition:

Chaos is aperiodic long-term behavior in a deterministic system that exhibits sensitive dependence on initial conditions.

- "Aperiodic long-term behavior" means that there are trajectories which do not settle down to fixed points, periodic orbits, or quasiperiodic orbits as $t \to \infty$. For practical reasons, we should require that such trajectories are not too rare. For instance, we could insist that there be an open set of initial conditions leading to aperiodic trajectories, or perhaps that such trajectories should occur with nonzero probability, given a random initial condition.
- "Deterministic" means that the system has no random or noisy inputs or parameters. The irregular behavior arises from the system's nonlinearity, rather than from noisy driving forces.
- "Sensitive dependence on initial conditions" means that nearby trajectories separate exponentially fast, i.e., the system has a positive Lyapunov exponent.

2.5.2 Lyapunov exponents

Lyapunov exponents are numbers that measure the exponential attraction or separation in time of two adjacent trajectories in phase space with different initial conditions. There are actually "n" different Lyapunov exponents for an n-dimensional system [8]. A positive Lyapanov exponent indicates a chaotic motion in a dynamical system with bounded trajectories.

Let $\delta(t)$ be the distance in time t of two trajectories in phase space that started with an initial distance δ_0 at time t_0. If $\delta(t)$ grows exponentially with the evolution of the system, then it has a sensitive dependency to the initial conditions. In mathematical terms,

$$\delta(t) = \delta_0 e^{\lambda}(t - t_0) \tag{2.33}$$

being λ the Lyapunov exponent.

The sum of the Lyapunov exponents is the time-averaged divergence of the phase space velocity; hence any dissipative dynamical system will have at least one negative exponent, the sum of all exponents is negative, and the post-transient motion of trajectories will occur on a zero volume limit set, an attractor [33]. An attractor for a dissipative system with one or more positive Lyapunov exponents is said to be "strange", or "chaotic" [33].

There are many different methods to estimate the largest Lyapunov exponents. The method proposed by Wolf [33] is most widely used to estimate Lyapunov exponents. However, also the Rosenstein [34] and Kantz [35] methods are usually employed to estimate the largest Lyapunov exponents.

2.5.3 Poincaré section (map)

An old technique for analyzing solutions to differential equations, developed by Poincaré, now assumes greater importance in the modern study of dynamical systems. The Poincaré section is a method to transform a continuous dynamical process in time into a set of difference equations, known in modern as a map. More specifically, the Poincaré section is a sequence of points in phase space generated by the penetration of a trajectory of continuous evolution through a generalized plane or surface in the space. For a periodically forced second-order nonlinear oscillator, a Poincaré map can be obtained by stroboscopically observing the position and velocity at a particular phase of the forcing function. The study of maps obtained from Poincaré sections of flows is based on the theory that certain topological features of the movement in time are preserved in the discrete-time dynamics of the maps.

To illustrate how a Poincaré section is obtained, imagine a system of three first-order differential equations that represent continuous trajectories in Cartesian space (see Figure 2.4). If the solutions are bounded, the solution curve is contained within some finite volume in this space.

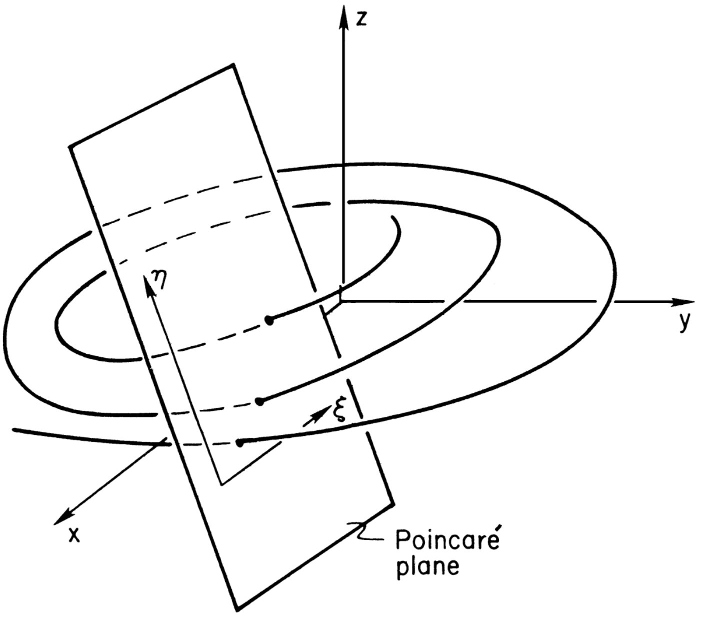

Figure 2.4 – Poincaré section.

Motions whose Poincaré maps have a finite set of points (periodical or subharmonic motion) or a closed curve of points are known as classical attractors. A motion with a set of

Poincaré points that is not a classical attractor and has specific fractal properties is known as a strange attractor. Strange attractor motions are related to chaotic motions.

2.5.4 Bifurcation diagram

Bifurcation denotes the change in the type of long-time dynamical motion when some parameter or set of parameters is varied (for example, as when a rod under a compressive load buckle—one equilibrium state changes to two stable equilibrium states). The study of the changes in the dynamic behavior of systems as parameters is the subject of bifurcation theory. Parameter Values at which the qualitative or topological nature of the motion changes are known as critical or bifurcation values.

2.5.5 Difference of chaotic and hyperchaotic systems

A necessary condition for a system to be chaotic is that at least one Lyapunov exponent be positive [36]. For a continuous dynamic system to be chaotic, it is necessary to have three or more states. On the other hand, a system can exhibit hyperchaos when at least two of their associated Lyapunov exponents are positive, and its states dimension is higher than three [37]. Note that every hyperchaotic system is also a chaotic system.

2.6 SYSTEMS IDENTIFICATION

Initially will be shown some concepts about systems identification. The concepts and Figures used in this section were taken from [1].

2.6.1 Theoretical and Experimental Modeling

In the following figure, there are different kinds of mathematical models.

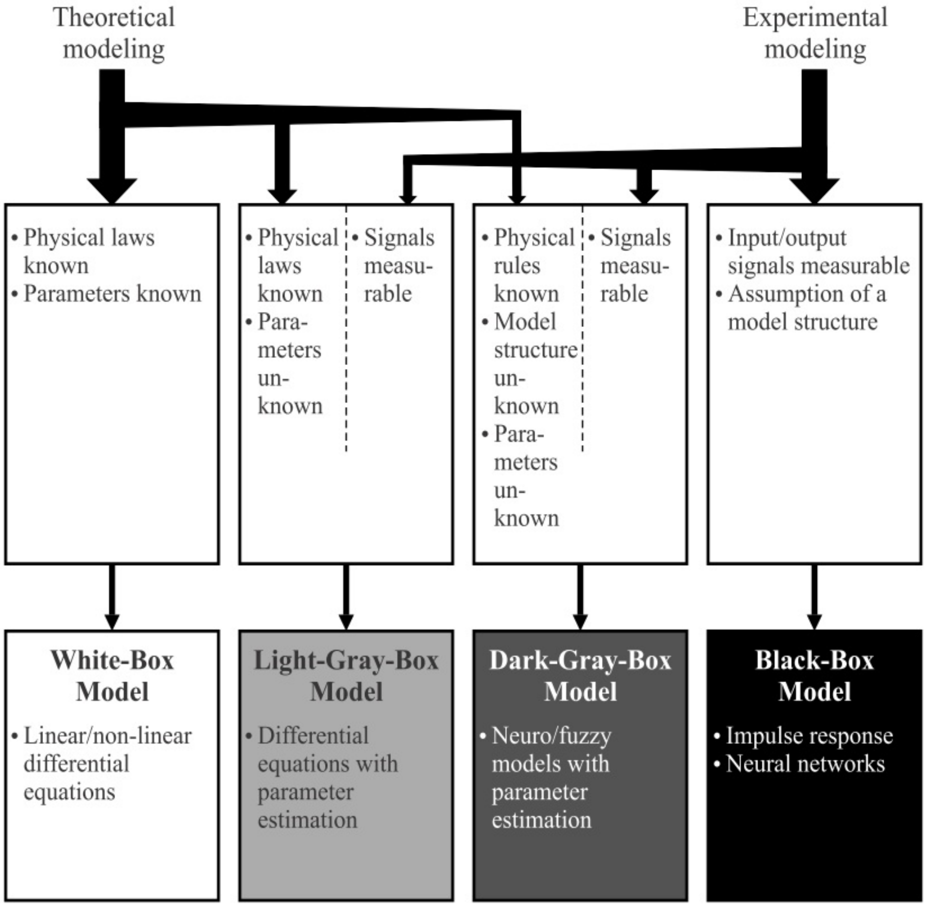

Figure 2.5 – Different kinds of mathematical models.

Although the theoretical analysis can, in principle, deliver more information about the system, provided that the internal behavior is known and can be described mathematically, experimental analysis has found ever-increasing attention over the past 50 years. The main reasons are the following:

- Theoretical analysis can become quite complex even for simple systems.
- Mostly, model coefficients derived from the theoretical considerations are not precise enough.
- Not all actions taking place inside the system are known.
- The actions taking place cannot be described mathematically with the required accuracy.
- Some systems are very complex, making the theoretical analysis too time-consuming.
- Identified models can be obtained in a shorter time with less effort compared to theoretical modeling

The experimental analysis allows the development of mathematical models by measuring the input and output of systems of arbitrary composition. One significant advantage is that the same experimental analysis methods can be applied to diverse and arbitrarily complex systems. By measuring the input and output only, one does, however, only obtain models governing the input-output behavior of the system, i.e., the models will in general not describe the precise internal structure of the system. These input-output models are approximations and are still sufficient for many areas of application. If the system also allows the measurement of internal states, one can obviously also gather information about the internal structure of the system. With the advent of digital computers starting in the 1960s, capable identification methods have started.

2.6.2 Offline and Online Identification

If digital computers are utilized for the identification, then one differentiates between two types of coupling between process and computer, see figure 2.6 (from [1]):

- Offline (indirect coupling)
- Online (direct coupling)

For offline identification, the measured data are first stored (e.g., data storage) and are later transferred to the computer utilized for data evaluation and are processed there.

The online identification is performed parallelly to the experiment. The computer is coupled with the process, and the data points are operated on as they become available. All algorithms used in this work are of online type.

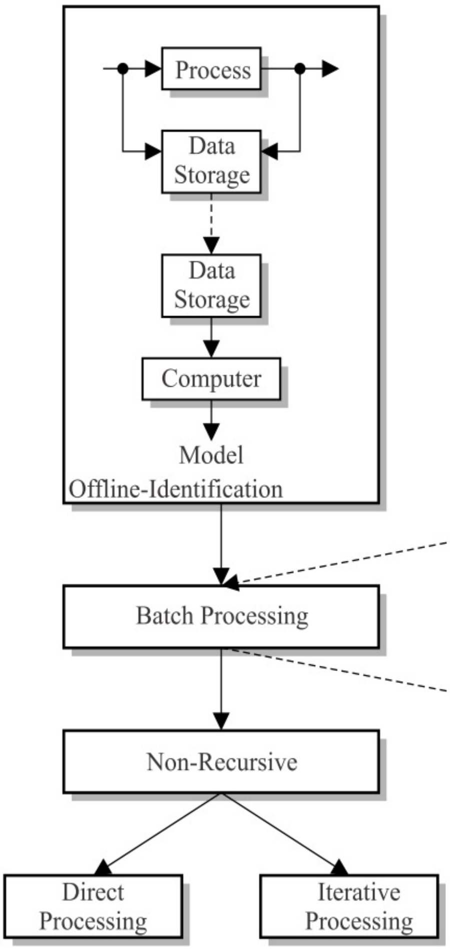

Figure 2.6 – Different setups for the data processing as part of the identification.

2.7 TRANSIENT STATE, STEADY-STATE, AND UNSTEADY-STATE RESPONSE

The following definition were extracted from [6].

Two parts compose a system response in the time domain, transient, steady-state, or unsteady-state. Transient is the immediate system response to an input from an equilibrium state. After the transient state, a system response can assume a steady-state or unsteady-state. In a stable system, the output tends to a constant value when $t \to \infty$. When the system response enters and stays in the threshold around the constant value, the system reaches the steady state. The time the stable system takes to reach the steady-state is the settling time, t_s. On the other hand, if the response never reaches a final value or oscillates, surpassing the threshold when $t \to \infty$ the system is then at unsteady-state. Consequently, the system outputs at unsteady-state vary with time during the on-time interval even induced by an invariable input.

2.8 SYNCHRONIZATION SYSTEMS

Most of the information contained in this section was extracted from [38].

2.8.1 Synchronization concepts

The analysis of synchronization phenomena in the evolution of dynamical systems has been a subject of active investigation for a long time. In the 17th century, Christiaan Huygens discovered that two very loosely coupled pendulum clocks (hanging from the same beam) synchronized in phase. Recently, the search for synchronization schemes has shifted to chaotic systems. In this sense, the emergence of collective (synchronized) dynamics is, in general, not trivial.

In chaotic systems, two trajectories emerging from two different close-by initial conditions separate exponentially over time. Therefore, chaotic systems intrinsically defy synchronization because even two identical systems starting from slightly different initial conditions would evolve in time unsynchronized manner (the systems states differences would grow exponentially). This is a practical problem since initial experimental conditions are never known perfectly. Therefore, the setting of some collective (synchronized) behavior in coupled chaotic systems has great importance and interest. The synchronization of chaos is when two (or many) chaotic systems (either equivalent or nonequivalent) adjust a given property of their motion to a typical behavior due to coupling or forcing. This ranges from a complete agreement of trajectories to the locking of phases.

A global system is formed by two subsystems that realize a drive–response (or master-slave) configuration. This implies that one subsystem evolves freely and drives the evolution of the other. So, the response system is slaved to follow the drive system's dynamics (or a proper function of the dynamics). Instead, it purely acts as an external but chaotic forcing for the response system. In such a case, external synchronization is produced. In other words, synchronization lies in adjusting the dynamic behavior of two dynamic systems, known as master (drive) and slave (response), so their trajectories converge in time. The synchronization of chaotic systems was first introduced in 1990 [39]. Typical examples of synchronization applications are communication with chaos.

2.8.2 Synchronization types

There are many different synchronization types, namely complete or identical synchronization (CS) [39, 40], phase (PS) [41] and lag (LS) synchronization [42], generalized synchronization (GS) [43, 44], intermittent lag synchronization (ILS) [42, 45], imperfect phase synchronization (IPS) [46], almost synchronization (AS) [47], antisynchronization [48], projective synchronization [49], hybrid projective synchronization [50], generalized projective synchronization [51, 52], modified projective synchronization (MPS) [53], function projective synchronization (FPS) [54], Q-S synchronization [55], full state hybrid projective synchronization [56], impulsive synchronization [57], finite-time synchronization [58], and exponential synchronization [59].

CS was the first discovered and is the simplest form of synchronization in chaotic systems. It consists of perfect hooking of the chaotic trajectories of two systems which are achieved utilizing a coupling signal to remain in step with each other over time. This mechanism was first shown to occur when two identical chaotic systems are coupled unidirectionally, provided that the conditional Lyapunov exponents of the subsystem to be synchronized are all negative [39]. GS goes further in using completely different systems and associating the output of one system to a given function of the output of the other system [43, 44]. Coupled nonidentical oscillatory or rotatory systems can reach an intermediate regime (PS), wherein locking of the phases is produced, while correlation in the amplitudes remains weak [41]. The transition to PS for two coupled oscillators has been firstly characterized regarding the Rössler system [41].

LS is a step between PS and CS. It implies the asymptotic boundedness of the difference between the output of one system at time t and the output of the other shifted in time of a lag time τ_{lag} [42]. This implies that the two outputs lock their phases and amplitudes but with a time lag [42]. ILS means that the two systems are most of the time verifying LS. Still, intermittent bursts of local nonsynchronous behavior may occur [42, 45] in correspondence with the passage of the system trajectory in particular attractor regions wherein the local Lyapunov exponent along a globally contracting direction is positive [42, 45]. Analogously, IPS is a situation where phase slips occur within a PS regime [46]. Finally, AS results in the asymptotic boundedness of the difference between a subset of variables of one system and the corresponding subset of variables of the other system [47].

2.9 SECURE COMMUNICATION GENERATIONS

The concepts of this section were extract from [60].

2.9.1 First generation

The first generation was proposed in 1993, known as additive chaos masking [61] and chaotic shift keying [62]. The additive chaos masking scheme consists of two identical chaotic systems in both the transmitter and the receiver. Chaotic shift keying, also known as chaotic switching, was designed to transmit a digital message signal.

The scheme was proved that it could not be used under experimental conditions because of the following drawbacks. Since the message signal is typically 20dB to 30dB weaker than the chaotic mask, this method is susceptible to channel noise and parameter mismatch between the chaotic transmitter and receiver systems. Furthermore, this scheme has a meager degree of security [63] and is very robust to noise and parameter mismatch. However, it has a low degree of security if the chaotic attractors are too far away in the bifurcation space. However, since this is the first scheme of chaotic digital communication systems, many possibilities exist for improving it.

2.9.2 Second generation

The second generation was proposed from 1993 to 1995, known as chaotic modulation. This generation used two different ways to modulate message signals into chaotic carriers. The first method, called chaotic parameter modulation [64], used message signals to change the parameters of the chaotic transmitter. The second method, called chaotic non-autonomous modulation [65], used the message signal to change the phase space of the chaotic transmitter.

The message signal modulates some parameters of the chaotic system in the transmitter such that its trajectories keep changing in different chaotic attractors. The second-generation improved the degree of security but was still found unsatisfactory [66–68].

2.9.3 Third generation

The third generation was proposed in 1997 [69] to improve security to a much higher level than the first two generations. We call this generation a chaotic cryptosystem. In this generation, the combination of the classical cryptographic technique and chaotic synchronization is used to enhance security. So far, this generation has had the highest security in all the chaotic secure communication systems proposed and has not yet been broken.

2.9.4 Fourth generation

The fourth-generation employed impulsive chaotic synchronization to synchronize chaotic transmitters and chaotic receivers. The synchronization signal for a third-order chaotic transmitter only used $\leq 94Hz$ bandwidth. This bandwidth is much smaller than the $30kHz$ bandwidth needed for transmitting synchronization signals in the other three generations. The degree of security of the fourth generation is higher than that of the third generation.

3 AN ADAPTIVE NEURAL IDENTIFIER WITH APPLICATIONS TO FINANCIAL AND WELDING SYSTEMS

The research outcomes of this chapter have been published as a journal paper entitled "An adaptive neural identifier with applications to financial and welding systems" in [16]. *This chapter has extended and improved some parts compared to the original paper.*

Modeling techniques are commonly separated into two groups: i) those based on modeling by physic laws; and ii) those based on identification from data (black-box identification) [70]. System identification lies in the estimation of the parameters of mathematical models from their input and output measurements. In this subject, there are several possible approaches to perform the parameter estimation, in discrete [71, 72] or continuous-time domain [73,74]. Many works have been made using Wiener [75,76] and Hammerstein [77–80] models. The block-oriented models allow for building models from simple blocks to find structures that are useful for several actual applications. Another approach lies in the use of fractional order models [75,81,82], which have been used mainly in the area of electrochemical energy storage systems [82]. For example, in [75], a fractional Wiener system identification was considered. In [78], a nonlinear recursive identification method of Hammerstein ARMAX systems was proposed. An additional approach lies in the use of an Extreme Learning Machine (ELM) [74, 80, 83]. The main feature of the ELM is the random initialization of hidden nodes which remain unchanged during the learning process without iterative tuning [74]. Another area considers the recursive [71] or iterative [84] identification of cascade systems. In [85], an improved nonlinear subspace identification method was performed. As can be seen, the area of system identification is widespread in the literature.

Neural identification models are relevant in predicting the behavior of dynamic systems and providing a parameterization when the model is uncertain. In 1956, the term *system identification* for the problem of parameter estimation of a black-box model from their input-output relationship was first introduced by Zadeh [2]. In the sequence, from 1990, works on identification systems using dynamic neural networks were performed. See, for instance, [3]. In dynamic neural models their weights are usually adjusted using the backpropagation algorithm or their robust modifications [3, 28, 86–90].

Several methods for identification and control based on linearly parameterized neural networks have emerged in the literature. These works are motivated by the simple structure of this kind of neural network, which simplifies the stability analysis. See, for example, the seminal work [91]. LPNN's include a fairly large class of neural networks, such as high order neural networks, radial basis function networks, adaptive fuzzy systems, and wavelet networks [28, 92–94]. Also, neuro-fuzzy identification models have been extensively used. See, for example, [95, 96] and the references therein. Other relevant approaches include the employment of fuzzy wavelet neural networks [95], and sinusoidal rough-neural networks [97]. There are also recent cases of the usage of neural networks in closed-loop identification in which feedback control signals are used [4, 98–100].

It should be noted that most of the neural identification models found in the literature use the series-parallel configuration [9, 71, 73–75, 80, 95, 97, 101–107]. However, for observer-based control problems, the parallel approach is more valuable than the series-parallel model, since the parallel configuration only employs the estimated states in the neural regressor [108]. Also, in most works in the literature, see, for instance, [71, 73, 75, 80, 95, 105–107], the adjustment of transient can not be made irrespective of the residual state error.

On the other hand, welding modeling is a challenging problem mainly owing to the complexity of the welding process, which is, inherently, time-varying, strongly coupled, and chaotic [109–113]. Several factors impact the welding process, such as welding voltage, welding current, welding speed, wire-speed, shielding gas, welding wire, work-piece, movement of the weld pool, and environmental conditions, among others. Hence, it is challenging to obtain a model by using classical approaches. In this case, intelligent approaches, such as neural networks and fuzzy logic, are more applicable [110, 111, 114, 115].

However, the application of these approaches in welding brings the following drawbacks: i) neural networks and fuzzy logic modeling depend on a process of data collection which is both very time-consuming and expensive [111, 116]; ii) rule explosion in fuzzy logic modeling become uncontrollable with the increase of variables [111]; iii) off-line learning with well-known drawbacks (overfitting and local minima [103]) is usually employed; iv) proposed works are, in general, not reusable since they are only valid for a specific droplet transfer mode in which their input/output relationships were acquired [117–121]; and v) proposed approaches for weld geometry prediction suffer from lack of theoretical results related to the stability and convergence of the approximation errors. See, for instance, [116, 122–124].

For example, in [116], a real-time weld geometry prediction is performed by using neural networks optimized by both principal component analysis (PCA) and genetic algorithm. The PCA algorithm preprocessed the sample data to remove the redundant information between model variables and reduce the model complexity. Hence, [116] had a stiff data collection process. In [122], a radial basis function neural network model to predict the weld bead geometry in shielded metal arc welding was proposed. In [123], a neural network model for predicting the depth of penetration and optimizing the welding bead was devised. In [124], a simulation study using multiple regression analysis and adaptive neuro-fuzzy inference system was accomplished. The main aim was to predict the complete weld bead geometry for a cold metal transfer welding process. All of these works presented relevant contributions to the subject of welding prediction. However, the theoretical justification of the stability and convergence of the prediction process was not considered. Furthermore, all of these works are only applicable to a simple welding transfer mode, where the data for training of the algorithms was obtained.

Motivated by the previous facts, a flexible neural identification model, which allows for series-parallel and parallel configurations, is proposed in this chapter. Based on Lyapunov theory and an identification model with suitable feedback terms, an adaptive law for the weight is proposed to guarantee that the state error converges to an arbitrary neighborhood of the origin while the weight error is uniformly bounded. Furthermore, the proposed algorithm allows for the regulation of the identification transient without affecting the residual state error, which is owing to both the enhanced weight law and the identification model. More precisely, the following contributions of this chapter are presented.

1) The proposed scheme allows for the adjustment of the identification transient from a design parameter that is not related to the size of the residual state error, in contrast to [71, 73–75, 80, 95, 97, 101–107];

2) The proposed identification model allows for the use of parallel and serial-parallel configurations, in contrast to [9, 71, 73–75, 80, 95, 97, 101–107];

3) The application of the proposed methodology for online weld geometry prediction over multiple droplet transfer modes is accomplished. It allows for reusability and does not suffer from overfitting or peaking phenomenon, in contrast to [116–124]. To the best of the author knowledge, an online identification algorithm for welding processes over multiple droplet transfer modes and based on Lyapunov theory is not present in the literature.

The chapter is organized as follows. Section 3.1 presents related concepts to linearly parameterized neural networks. In Section 3.2, the problem to be studied and the main assumptions were introduced. The identification model and state estimate error equation are presented in Section 3.3. In Section 3.4, a weight law, which ensures that the residual state error is ultimately bounded, is proposed. The proposed identifier is validated in Section 3.5, where two chaotic systems are considered. Finally, the conclusions of the chapter are presented in Section 3.6.

3.1 LINEARLY PARAMETERIZED NEURAL NETWORKS

Linearly parameterized neural networks can be expressed as

$$\rho_{nn}\left(\hat{W},\zeta\right)=\hat{W}\sigma\left(\zeta\right) \tag{3.1}$$

where $\rho_{nn} : \Re^{L_{nn}} \mapsto \Re^{n}$ is a function, $\hat{W} \in \Re^{n \times L_{\rho}}$ is a weight matrix, $\zeta \in \Re^{L_{\zeta}}$ are the inputs of the neural network and $\sigma : \Re^{L_{\zeta}} \mapsto \Re^{L_{\rho}}$ is a nonlinear vector function whose arguments are preprocessed by a scalar function $s(\cdot)$, and $n, L_{\rho}, L_{\zeta}, L_{nn}$ are integers strictly positive. Normally used scalar functions $s(\cdot)$ include sigmoid (used in this chapter), hyperbolic tangent, gaussian, Hardy's, inverse Hardy's multiquadratic [86]. This chapter is only interested in the class of LPNNs for which $\sigma(\cdot)$ is bounded, that is,

$$\|\sigma(\zeta)\| \leq \sigma_0 \tag{3.2}$$

where σ_0 is a strictly positive constant.

3.2 PROBLEM STATEMENTS

Consider the following nonlinear differential equation:

$$\dot{x} = F(x, u, v, t), \quad x(0) = x_0 \tag{3.3}$$

where $x \in X \subset \Re^{n}$ is a n-dimensional state vector, $u \in U \subset \Re^{m}$ is a m-dimensional admissible input vector, $v \in V \subset \Re^{p}$ is p-dimensional vector of time varying uncertain variables, t is the time, and $F : X \times U \times V \times [0, \infty) \mapsto \Re^{n}$ is a continuous map. It is assumed that X, U, V are compact sets and F is an unknown map and is locally Lipschitzian with respect to x in $X \times U \times V \times [0, \infty)$, so that (3.3) has a unique solution through x_0.

In this chapter, the following is established:

ASSUMPTION 3.2.1 On a region $X \times U \times V \times [0, \infty)$

$$\|h(x, u, v, t)\| \leq h_0 \tag{3.4}$$

$$h(x, u, v, t) = F(x, u, v, t) - f(x, u) \tag{3.5}$$

where h are external or internal disturbances, f is an unknown map and $\bar{h}_0$, such that $\bar{h}_0 > h_0 \geq 0$, is an unknown constant.

REMARK 3.2.1 The Assumption 3.2.1 is standard in the literature. The consideration of disturbances explicitly is interesting in Lyapunov analysis to know the real impact of disturbances on the size of the residual state set.

The main goal is to design an identifier based on neural networks for (3.3) to ensure the convergence of the estimated state to an arbitrary neighborhood of the actual state, even in the presence of approximation error and bounded disturbances.

3.3 IDENTIFICATION MODEL AND STATE ESTIMATE ERROR EQUATION

Consider that $\bar{f}$ is the best known approximation of f, $P \in \Re^{n \times n}$ is a scaling matrix defined as $P = P^T > 0$, $\bar{g} = P^{-1}g$, and $g(x,u) = f(x,u) - \bar{f}(x,u)$. After some mathematical manipulations, the system (3.3) can be rewritten as

$$\dot{x} = \bar{f}\left(x,u\right) + P\bar{g}\left(x,u\right) + h\left(x,u,v,t\right) \tag{3.6}$$

Note that if the designer has no previous knowledge of f, then $\bar{f}$ is assumed as being a zero vector. From (3.6), based on LPNNs, the nonlinear mapping $\bar{g}(x,u)$ can be substituted by the neural parametrization $W^{*}\sigma(x,u)$ plus the reconstruction error term $\varepsilon(x,u)$. Then, (3.6) becomes

$$\dot{x} = \bar{f}\left(x,u\right) + PW^{*}\sigma\left(x,u\right) + P\varepsilon\left(x,u\right) + h\left(x,u,v,t\right) \tag{3.7}$$

where $\sigma\left(x,u\right)$ is a nonlinear vector function whose arguments are preprocessed by a scalar sigmoidal function $s(\cdot)$ and $W^{*} \in \Re^{n \times L}$ is an ideal matrix, only required for analytical purposes, defined as

$$W^{*} := \underset{(\hat{W} \in \Gamma)}{\arg\min} \left\{ \left\| \bar{g}\left(x,u\right) - \hat{W}\sigma\left(x,u\right) \right\|_{\infty} \right\} \tag{3.8}$$

where $x \in X$, $u \in U$, $\Gamma = \left\{ \hat{W} \in \Re^{n \times L} : || \hat{W} ||_F < \alpha_{\hat{w}} \right\}$, $\alpha_{\hat{w}}$ is a strictly positive constant, and $\hat{W}$ is an estimate of W^{*}. The reconstruction error term $\varepsilon\left(x,u\right)$ can be defined as

$$\varepsilon\left(x,u\right) := \bar{g}\left(x,u\right) - W^{*}\sigma\left(x,u\right) \tag{3.9}$$

Since X, U are compact sets and by using (3.2), the following can be assumed.

ASSUMPTION 3.3.1 On a region $X \times U$, the approximation error is upper bounded by

$$||\varepsilon(x,u)|| < \varepsilon_0 \tag{3.10}$$

$$\bar{\varepsilon}_0 > \varepsilon_0 \geq 0 \tag{3.11}$$

where $\bar{\varepsilon}_0$ is an unknown constant.

The Assumption 3.3.1 is standard in identification. Additionally, note that the uniqueness of $||\varepsilon(x,u)||$ is ensured by (3.8), but $\varepsilon(x,u)$ and W^* might be nonunique. The W^* is introduced as being the value of $\hat{W}$ that minimizes the L_∞-norm difference between $\bar{g}(x,u)$ and $\hat{W}\sigma(x,u)$. The scaling matrix P in (3.6) is defined to manipulate the performance of the identification process.

The proposed identification model is

$$\dot{\hat{x}} = -\left(L - \psi\gamma_W I\right)\left(\hat{x} - x\right) + P\hat{W}\hat{\sigma} \tag{3.12}$$

where $\hat{x}$ is the estimated state, $\gamma_W > 0$, $\psi > 0$, $\hat{\sigma} = \sigma\left(x,u\right)$ for the series-parallel and $\hat{\sigma} = \sigma\left(\hat{x},u\right)$ for the parallel configuration [24], $\hat{\sigma}$ is a nonlinear vector function whose arguments are preprocessed by a sigmoidal scalar function $s(\cdot)$, and $L \in \Re^{n\times n}$ is a positive definite feedback gain matrix introduced to attenuate the effect of the perturbations. It will be demonstrated that the identification model (3.12) and the adjustment law for $\hat{W}$, to be proposed in the next section, ensure the convergence of the residual state error to an arbitrary neighborhood of the origin, even in the presence of reconstruction error and bounded disturbances.

Assuming that the state estimation error is $\tilde{x} = \hat{x} - x$, from (3.7) and (3.12), results

$$\begin{aligned}\dot{\tilde{x}} = &-\left(L - \psi\gamma_W I\right)\tilde{x} + P\tilde{W}\hat{\sigma} + PW^*\tilde{\sigma} \\ &- P\varepsilon\left(x,u\right) - h\left(x,u,v,t\right)\end{aligned} \tag{3.13}$$

where $\tilde{W} = \hat{W} - W^*$, $\tilde{\sigma} = 0$ for the series-parallel identification model and $\tilde{\sigma} = \sigma\left(\hat{x},u\right) - \sigma\left(x,u\right)$ for the parallel identification model.

3.4 ADAPTIVE WEIGHT LAW AND STABILITY ANALYSIS

Before presenting the main theorem, some facts are stated, used in the stability analysis.

Fact 3.4.1 In the problem of this chapter, the following equation is valid:

$$tr\left(\tilde{W}^T\tilde{x}\sigma^T\right)=\tilde{x}^T\tilde{W}\sigma \tag{3.14}$$

Fact 3.4.2 Let $W^*, W_0, \hat{W}, \tilde{W} \in \Re^{n\times L}$. Then, with the definition of $\tilde{W}=\hat{W}-W^*$, the following equation is true:

$$2tr\left[\tilde{W}^T\left(\hat{W}-W_0\right)\right]=\left\|\tilde{W}\right\|_F^2+\left\|\hat{W}-W_0\right\|_F^2-\left\|W^*-W_0\right\|_F^2 \tag{3.15}$$

Fact 3.4.3 Let $A\in\Re^{c\times d}$, $b\in\Re^c$, where $c>0$ and $d>0$ are whole numbers. Then, the following expressions are true:

$$tr\left(A^T+A\right)=2tr\left(A\right) \tag{3.16}$$

$$-b^TAb\leq-\lambda_{min}b^T\left(A\right)b \tag{3.17}$$

where $\lambda(A)$ is its eigenvalues

Fact 3.4.4 Consider that a, b and $c\in\Re^+$, so

$$a\left\|\tilde{x}\right\|^2-b\left\|\tilde{x}\right\|-c>0 \tag{3.18}$$

$$\left\|\tilde{x}\right\|^2-\frac{b}{a}\left\|\tilde{x}\right\|+\frac{b^2}{4a^2}>\frac{c}{a}+\frac{b^2}{4a^2} \tag{3.19}$$

$$\left(\left\|\tilde{x}\right\|-\frac{b}{2a}\right)^2>\frac{4ac+b^2}{4a^2} \tag{3.20}$$

$$\left\|\tilde{x}\right\|>\frac{b\pm\sqrt{4ac+b^2}}{2a} \tag{3.21}$$

As $b-\sqrt{4ac+b^2}<0$ and $\|\tilde{x}\|\geq0$, this is an invalid solution, then

$$\left\|\tilde{x}\right\|>\frac{\frac{b}{2}+\sqrt{ac+\left(\frac{b}{2}\right)^2}}{a} \tag{3.22}$$

Fact 3.4.5 Consider that $a, b,$ and $c\in\Re^+$, then

$$m\left(\tilde{x}\right)=-a\left\|\tilde{x}\right\|^2+b\left\|\tilde{x}\right\|+c \tag{3.23}$$

The derivative of equation (3.23) is equal to

$$\dot{m} = -2a\left\|\tilde{x}\right\| + b \tag{3.24}$$

The maximum value of (3.23) occurs when $\dot{m} = 0$, then

$$\left\|\tilde{x}\right\| = \frac{b}{2a} \tag{3.25}$$

Replacing this value in (3.23), then the maximum value of (3.23) is equal to

$$m\left(\tilde{x}\right) = \frac{4ac + b^2}{4a} \tag{3.26}$$

FACT 3.4.6 Let $a_1, a_2, a_3, a_4 \in \Re^+$. Consider the following logistic sigmoid function

$$s(a_4) = \frac{a_1}{a_2 + a_3 \cdot e^{-a_4}} \tag{3.27}$$

Note that regardless of the values of a_4, the value of s is bounded:

$$\frac{a_1}{a_2 + a_3} < s < \frac{a_1}{a_2} \tag{3.28}$$

REMARK 3.4.1 Note that $\tilde{\sigma}\left(\hat{x}(t), x(t), u(t)\right)$ is bounded, in other words, there are a σ_{sup} such that $\sigma_{sup} \geq sup\left\{\tilde{\sigma}\left(\hat{x}(t), x(t), u(t)\right)\right\}$ for all $t \geq 0$.

The main theorem of this chapter is stated and proved in this section.

THEOREM 3.4.1 Consider the class of general nonlinear systems defined in (3.3), which satisfies the Assumptions 3.2.1 and 3.3.1, and the identification model (3.12). If the weight learning law is given by

$$\dot{\hat{W}} = -2\gamma_W\left[\psi\left(\hat{W} - W_0\right) + \tilde{x}\hat{\sigma}^T\right] \tag{3.29}$$

then $\tilde{x}$ and $\tilde{W}$ are uniformly bounded and $\tilde{x}$ is uniformly ultimately bounded with ultimate bound ρ_2, where $\rho_2 = \frac{b}{2\lambda_{min}(Q)} + \frac{\sqrt{\lambda_{min}(Q)c + (\frac{b}{2})^2}}{\lambda_{min}(Q)}$, $b = 2\left(\bar{\varepsilon}_0 + \left\|P^{-1}\right\|_F \bar{h}_0 + \bar{y}\right)$, $c = \psi\left\|W^* - W_0\right\|_F^2$, $Q = L^T P^{-1} + P^{-1}L$, W_0 is a constant matrix, $\bar{y} = 0$ for the series-parallel identification model, $\bar{y} = \sigma_{sup}\left\|W^*\right\|_F$ for the parallel identification model, and $\sigma_{sup} = sup$ $\tilde{\sigma}\left(\hat{x}(t), x(t), u(t)\right)$ for all $t \geq 0$.

Proof.

Consider the following Lyapunov function candidate:

$$V = \tilde{x}^T P^{-1} \tilde{x} + \frac{tr\left(\tilde{W}^T \gamma_W^{-1} \tilde{W}\right)}{2} \tag{3.30}$$

The time-derivative of (3.30) results

$$\dot{V} = \tilde{x}^T P^{-1} \dot{\tilde{x}} + (\tilde{x}^T P^{-1} \dot{\tilde{x}})^T + \gamma_W^{-1} tr\left(\tilde{W}^T \dot{\tilde{W}}\right) \tag{3.31}$$

By replacing (3.13) and (3.29) in (3.31), implies

$$\begin{aligned} \dot{V} = & -\tilde{x}^T \left(P^{-1}L + L^T P^{-1}\right)\tilde{x} - \psi\gamma_W [\tilde{x}^T P^{-1}\tilde{x} \\ & + \left(\tilde{x}^T P^{-1}\tilde{x}\right)^T] + \tilde{x}^T \left(\tilde{W}\hat{\sigma} + W^*\tilde{\sigma} - \varepsilon - P^{-1}h\right) \\ & - 2\psi tr\left[\tilde{W}^T\left(\hat{W} - W_0\right)\right] + [\tilde{x}^T(\tilde{W}\hat{\sigma} + W^*\tilde{\sigma} - \varepsilon \\ & - P^{-1}h)]^T - 2tr\left(\tilde{W}^T\tilde{x}\hat{\sigma}^T\right) \end{aligned} \tag{3.32}$$

Employing the Facts 3.4.1 and 3.4.2 in (3.32) results

$$\begin{aligned} \dot{V} \leq & -\lambda_{min}\left(Q\right)\left\|\tilde{x}\right\|^2 - 2\psi\gamma_W\tilde{x}^T P^{-1}\tilde{x} \\ & + 2\left\|\tilde{x}\right\|\left(\left\|\varepsilon\right\| + \left\|P^{-1}\right\|_F \left\|h\right\| + \bar{y}\right) \\ & - \psi\left(\left\|\tilde{W}\right\|_F^2 + \left\|\hat{W} - W_0\right\|_F^2 - \left\|W^* - W_0\right\|_F^2\right) \end{aligned} \tag{3.33}$$

By using $\left\|\varepsilon\right\| < \bar{\varepsilon}_0$ and $\left\|h\right\| < \bar{h}_0$, (3.33) implies

$$\begin{aligned} \dot{V} \leq & -\left\|\tilde{x}\right\|^2\left[\lambda_{min}\left(Q\right)\right] + \left\|\tilde{x}\right\|\left(2\bar{\varepsilon}_0 + 2\left\|P^{-1}\right\|_F \bar{h}_0 + 2\bar{y}\right) \\ & + \psi\left\|W^* - W_0\right\|_F^2 - \psi\left\|\tilde{W}\right\|_F^2 - \psi\left\|\hat{W} - W_0\right\|_F^2 - 2\psi\gamma_W\tilde{x}^T P^{-1}\tilde{x} \end{aligned} \tag{3.34}$$

By considering that $a = \lambda_{min}\left(Q\right)$, $b = 2(\bar{\varepsilon}_0 + \left\|P^{-1}\right\|_F \bar{h}_0 + \bar{y})$, and $c = \psi\left\|W^* - W_0\right\|_F^2$, where $a > 0$, $b \geq 0$, and $c \geq 0$, results

$$\begin{aligned} \dot{V} \leq & -a\left\|\tilde{x}\right\|^2 + b\left\|\tilde{x}\right\| + c \\ & - \psi\left\|\tilde{W}\right\|_F^2 - \psi\left\|\hat{W} - W_0\right\|_F^2 - 2\psi\gamma_W\tilde{x}^T P^{-1}\tilde{x} \end{aligned} \tag{3.35}$$

Case 1: Boundedness of $\tilde{W}$ and $\tilde{x}$: Based on (3.35), then

$$\dot{V} \leq -a \left\| \tilde{x} \right\|^2 + b \left\| \tilde{x} \right\| + c - \psi \left\| \tilde{W} \right\|_F^2 \tag{3.36}$$

In this way, $\dot{V} < 0$ as long as

$$\left\| \tilde{W} \right\|_F > \sqrt{\frac{4ac + b^2}{4a\psi}} := \rho_1 \tag{3.37}$$

or

$$\left\| \tilde{x} \right\| > \frac{\sqrt{4ac + b^2} + b}{2a} := \rho_2 \tag{3.38}$$

Thus, since ρ_1 and ρ_2 are constants, by using Lyapunov arguments [22], it can be established that $\tilde{W}$ and $\tilde{x}$ are uniformly bounded since $\dot{V} < 0$ outside of compact set Ω, where $\Omega = \{\tilde{W}, \tilde{x} : \left\| \tilde{W} \right\|_F \leq \rho_1 \text{ or } \left\| \tilde{x} \right\| \leq \rho_2\}$. Observe that if, at any time, $\left\| \tilde{W} \right\|_F$ or $\left\| \tilde{x} \right\|$ escape of the residual set Ω, then $\dot{V} < 0$, and it forces the convergence of the weight and state error to the residual set Ω (Figure 3.1).

Case 2: Identification transient: From (3.35), then

$$\begin{aligned} \dot{V} \leq & -\alpha V + \alpha V - a \left\| \tilde{x} \right\|^2 + b \left\| \tilde{x} \right\| + c \\ & - \psi \left\| \tilde{W} \right\|_F^2 - \psi \left\| \hat{W} - W_0 \right\|_F^2 - 2\psi\gamma_W \tilde{x}^T P^{-1} \tilde{x} \end{aligned} \tag{3.39}$$

or

$$\begin{aligned} \dot{V} \leq & -\alpha V - \psi \left\| \hat{W} - W_0 \right\|_F^2 \\ & + \left\| \tilde{W} \right\|_F^2 \left(\frac{\alpha}{2\gamma_W} - \psi \right) + \tilde{x}^T P^{-1} \tilde{x} \left(\alpha - 2\psi\gamma_W \right) \end{aligned} \tag{3.40}$$

By considering that $\alpha = 2\psi\gamma_W$ results

$$\dot{V} \leq -\alpha V \tag{3.41}$$

By using Lemma 3.2.4 [24], it can be established that:

$$V\left(t\right) \leq e^{-\alpha(t-t_0)} V(t_0) \tag{3.42}$$

$$\forall t \geq t_0 \geq 0 \tag{3.43}$$

Assuming that $t_0 = 0$, then

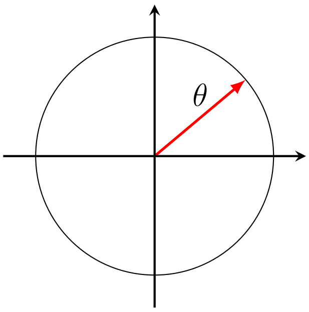

Figure 3.1 – Bounded set.

$$V(t) \leq V(0)e^{-\alpha t} \tag{3.44}$$

In addition, by defining

$$V_M := \sup_{(\tilde{x},\tilde{W})\in\Omega} \left\{ V\left(\tilde{x},\tilde{W}\right) \right\} \tag{3.45}$$

and if $V(t) = V_M$, then a time interval in which $\tilde{x}$ and $\tilde{W}$ reach Ω can easily be obtained, i.e.,

$$ln\left[V_M\right] \leq -\alpha t + ln\left[V(0)\right] \tag{3.46}$$

or

$$t \leq \frac{ln\left[\frac{V(0)}{V_M}\right]}{\alpha} \tag{3.47}$$

□

REMARK 3.4.2 In (3.38), the size of the residual state error is inversely proportional to $\lambda_{min}(Q)$; hence, by changing the values of L and P, the residual state error size can be reduced, even in the presence of bounded disturbances.

REMARK 3.4.3 The identification model and learning law were chosen to allow for the adjustment of the transient independent from the residual state error. Note that t in (3.47) is the maximum time for $\tilde{x}$ and $\tilde{W}$ reach Ω, where the identification transient is inversely proportional to α. Since $\alpha = 2\psi\gamma_W$, the manipulation of the identification transient is possible only by changing γ_W, since the ultimate bound ρ_2 in (3.38) does not depend on γ_W.

REMARK 3.4.4 The uncoupling between identification transient and residual state error was mainly obtained by the introduction of the design parameter γ_W jointly in the identification model (3.12) and weight law (3.29). The employment of this parameter had a positive impact on the Lyapunov analysis, which allowed us the obtainment of (3.38) and, finally, (3.47).

3.5 SIMULATION

In this section, two examples are presented to validate the theoretical results. The first example, Section 3.5.1, shows the independence between the transient and residual state error using a serial-parallel configuration. In Section 3.5.2, a comparison is made between chapter identification algorithm and that in [74]. The second example, an application of the proposed methodology to a challenging problem, is presented in Section 3.5.3, where a parallel configuration was used.

In all simulations to obtain the numerical solutions were used the solver ode45 of Matlab /Simulink®, with variable-step, and relative and absolute tolerance of $1e - 10$, in an AMD Ryzen 7 1700 processor, Windows 10. Figure 3.2 shows a flow diagram with the implementation of the proposed method. Note that the blue blocks show the unknown information (black-box identification), and the other blocks the known information.

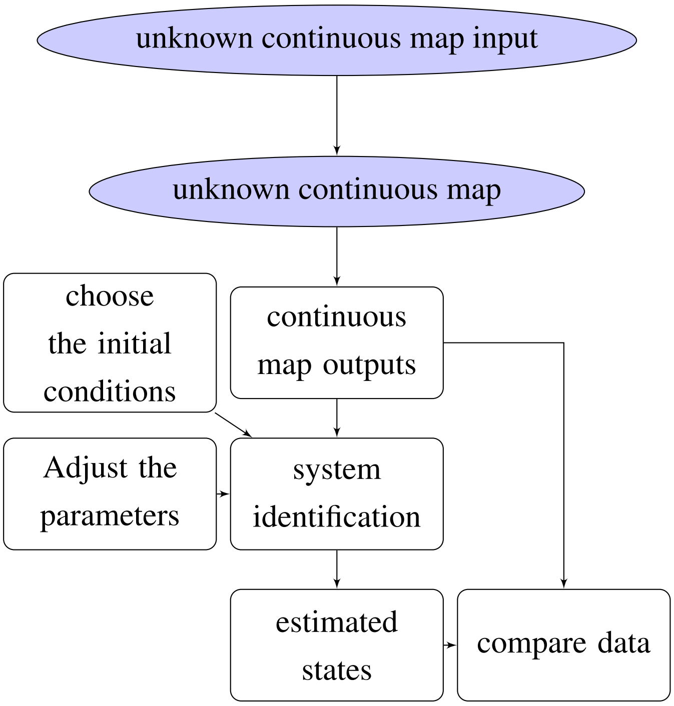

Figure 3.2 – Flow diagram.

3.5.1 Hyperchaotic Finance System

Consider the perturbed hyperchaotic finance system [125, 126], which is described by

$$\begin{aligned}
\dot{x_1} &= x_3 + (x_2 - a)x_1 + x_4 + d_{x_1} \\
\dot{x_2} &= 1 - bx_2 - x_1^2 + d_{x_2} \\
\dot{x_3} &= -x_1 - cx_3 + d_{x_3} \\
\dot{x_4} &= -dx_1x_2 - kx_4 + d_{x_4}
\end{aligned} \tag{3.48}$$

where x_1,x_2, x_3, and x_4 are the interest rate, investment demand, price exponent, and average profit margin, respectively, d_{x_1}, d_{x_2}, d_{x_3}, and d_{x_4} are unknown disturbances, and $a = 0.9$, $b = 0.2$, $c = 1.5$, $d = 0.2$, and $k = 0.17$ are known parameters. The system is autonomous, that is, $u = 0$.

To identify the uncertain system (3.48), the proposed identification model (3.12) and the adaptive law (3.29) were implemented. The initial conditions for the chaotic system and identification model were $x_1(0) = 1$, $x_2(0) = 2$, $x_3(0) = 0.5$, $x_4(0) = 0.5$, $\hat{x}_1(0) = -2$, $\hat{x}_2(0) = -2$, $\hat{x}_3(0) = -2$, $\hat{x}_4(0) = -2$, and $\hat{W}(0) = 0$ to evaluate the performance of the proposed algorithm under adverse initial conditions. The design matrices were chosen as $P = 30I$ and $L = 2I$, where I is the identity matrix. The nonlinear vector function σ is equal to

$$\sigma^T = \begin{matrix} [s(x_1), & s(x_2), & s(x_3), & s(x_4) \\ s^2(x_1), & s^2(x_2), & s^2(x_3), & s^2(x_4)] \end{matrix} \tag{3.49}$$

and $s(\cdot) = \frac{5}{1+e^{-0.5(\cdot)}}$. The design parameters ψ and W_0 were chosen as $\psi = 1$ and $W_0 = \begin{bmatrix} I & | & 0 \end{bmatrix}$.

To verify the robustness of the proposed method, at $t = 5s$ is introduced the presence of disturbances of the form $d_{x_1} = 0.7cos(6t)$, $d_{x_2} = cos(5t)$, $d_{x_3} = 0.9cos(4t)$, and $d_{x_4} = 0.8sin(3t)$. From (3.47), it can be seen that the parameter α affects the transient time. Once $\alpha = 2\psi\gamma_W$, the parameter γ_W can be used to control the transient, since γ_W is not related to the size of the residual state error, as can be seen in (3.12). Three different values of γ_W were used for three different simulations. The three values chosen for γ_W are $\gamma_W = 0.5$, $\gamma_W = 5$, and $\gamma_W = 50$. Figures 3.3 - 3.6 show the performances obtained in the estimation of the states for different values of γ_W. The Table 3.1 shows the root mean square value error (RMSE) for the different values of γ_W.

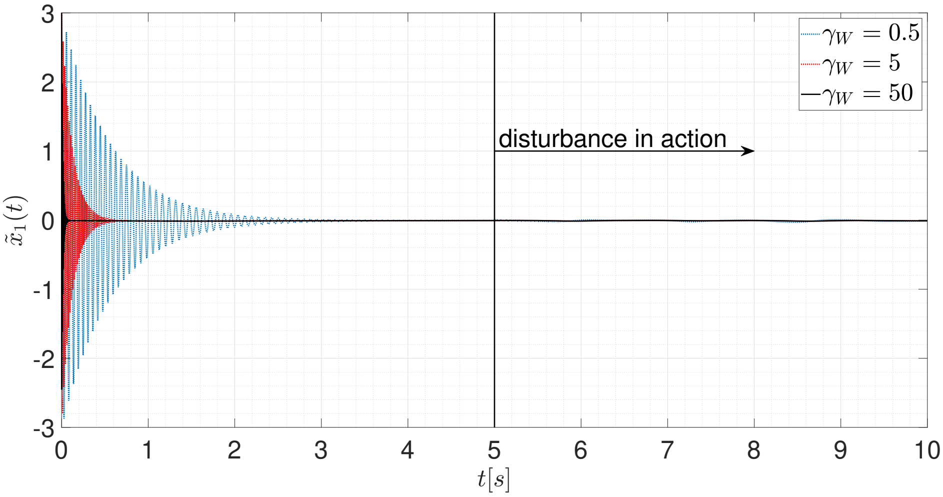

Figure 3.3 – Performance in the estimation of the interest rate (x_1).

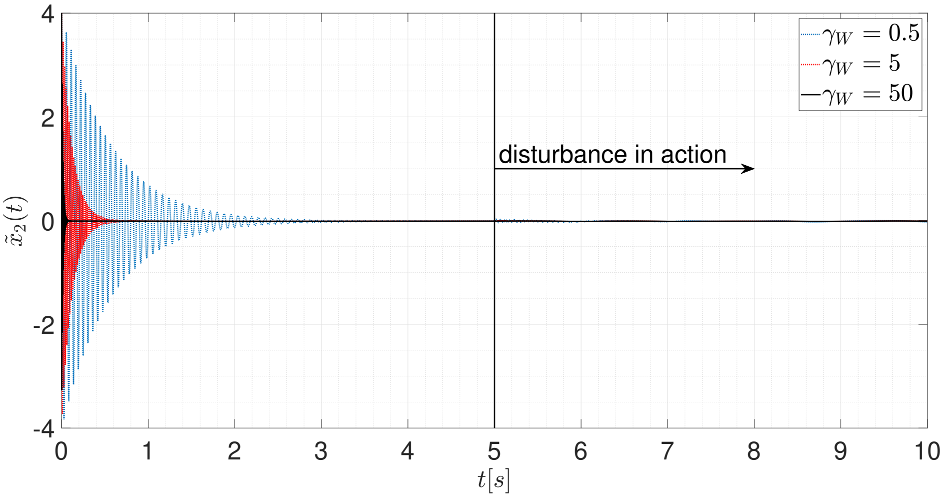

Figure 3.4 – Performance in the estimation of the investment demand (x_2).

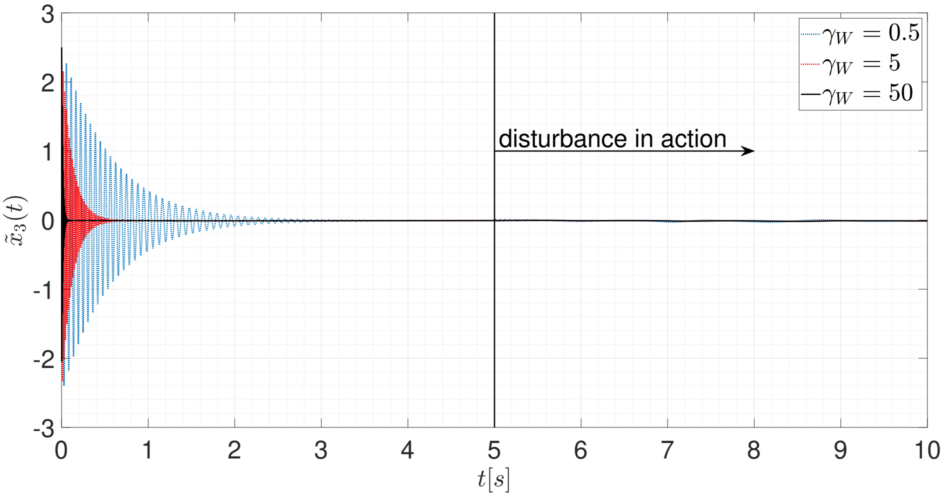

Figure 3.5 – Performance in the estimation of the price exponent (x_3).

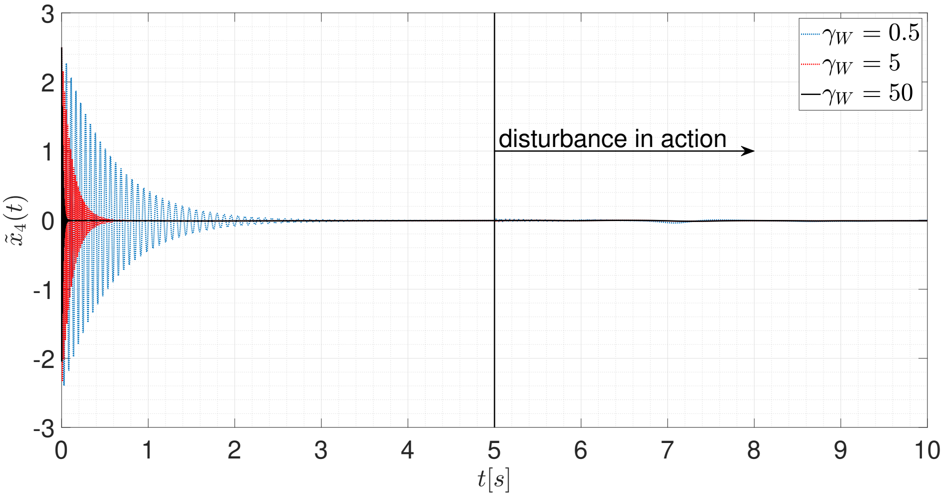

Figure 3.6 – Performance in the estimation of the average profit margin (x_4).

Table 3.1 – Root mean square of state errors for t = [0 4] seconds.

Root Mean Square of State Errors			
γ_W	0.5	5	50
$e_{1_{rms}}$	0.794721	0.650665	0.482688
$e_{2_{rms}}$	1.059300	0.867260	0.643366
$e_{3_{rms}}$	0.662322	0.542299	0.402297
$e_{4_{rms}}$	0.662369	0.542317	0.402315
e_{rms}	0.811025	0.664020	0.492596

Note from Figures 3.3 - 3.6 that the residual state error, after the transient, is approximately null for all situations. These figures also depict that the transients are different depending on the design parameter γ_W. Table 3.1 shows the RMSE for different γ_W value. Small γ_W correspond to large RMSE and transient. Also, large γ_W correspond to small RMSE and transient, as depicted in Table 3.1 and Figure 3.2.

Tables 3.2 - 3.8 show results when changing the parameters of ψ, L and P. Larger values of ψ lead to lower RMSE values in all cases. This is expected because of larger values of ψ lead to a shorter transient duration. In the case of L and P, it is a little more complex. In the case of L, it is observed that when γ_W is small, a larger value of L leads to a smaller transient and consequently a smaller RMSE, but for high values of γ_W, changes in L do not have much effect. In the case of P, one notices that when γ_W is raised, an increase in P leads to a higher RMSE value. The reason for this is that higher values of P collaborate to have a longer transient duration.

Table 3.2 – Root mean square of state errors for t = [0 10] seconds and ψ = 1, L = 2I, and P = 30I.

Root Mean Square of State Errors			
γ_W	0.5	5	50
$e_{1_{rms}}$	0.628738	0.498952	0.312479
$e_{2_{rms}}$	0.838058	0.665049	0.416507
$e_{3_{rms}}$	0.524012	0.415878	0.260504
$e_{4_{rms}}$	0.524047	0.415881	0.260483
e_{rms}	0.641634	0.509201	0.318912

Table 3.3 – Root mean square of state errors for t = [0 10] seconds and ψ = 0.1, L = 2I, and P = 30I.

Root Mean Square of State Errors			
γ_W	0.5	5	50
$e_{1_{rms}}$	0.704385	0.673863	0.548365
$e_{2_{rms}}$	0.939192	0.898458	0.731131
$e_{3_{rms}}$	0.586971	0.561556	0.456977
$e_{4_{rms}}$	0.586990	0.561560	0.456978
e_{rms}	0.718901	0.687751	0.559668

Table 3.4 – Root mean square of state errors for t = [0 10] seconds and ψ = 10, L = 2I, and P = 30I.

Root Mean Square of State Errors			
γ_W	0.5	5	50
$e_{1_{rms}}$	0.436541	0.266011	0.155488
$e_{2_{rms}}$	0.578026	0.352088	0.206001
$e_{3_{rms}}$	0.367531	0.229060	0.134457
$e_{4_{rms}}$	0.366604	0.226480	0.132730
e_{rms}	0.445148	0.272501	0.159446

Table 3.5 – Root mean square of state errors for t = [0 10] seconds and ψ = 1, L = 0.2I, and P = 30I.

Root Mean Square of State Errors			
γ_W	0.5	5	50
$e_{1_{rms}}$	0.739183	0.513046	0.313774
$e_{2_{rms}}$	0.985318	0.683835	0.418233
$e_{3_{rms}}$	0.616103	0.427624	0.261582
$e_{4_{rms}}$	0.616123	0.427628	0.261562
e_{rms}	0.754375	0.523584	0.320233

Table 3.6 – Root mean square of state errors for t = [0 10] seconds and ψ = 1, L = 20I, and P = 30I.

Root Mean Square of State Errors			
γ_W	0.5	5	50
$e_{1_{rms}}$	0.393517	0.408435	0.299869
$e_{2_{rms}}$	0.524287	0.544399	0.399698
$e_{3_{rms}}$	0.327790	0.340448	0.249996
$e_{4_{rms}}$	0.327887	0.340443	0.249974
e_{rms}	0.401429	0.416827	0.306043

Table 3.7 – Root mean square of state errors for t = [0 10] seconds and ψ = 1, L = 2I, and P = 3I.

Root Mean Square of State Errors			
γ_W	0.5	5	50
$e_{1_{rms}}$	0.609060	0.462488	0.283054
$e_{2_{rms}}$	0.812111	0.616407	0.377257
$e_{3_{rms}}$	0.507335	0.385395	0.235916
$e_{4_{rms}}$	0.507487	0.385423	0.235898
e_{rms}	0.621403	0.471930	0.288835

Table 3.8 – Root mean square of state errors for t = [0 10] seconds and ψ = 1, L = 2I, and P = 300I.

Root Mean Square of State Errors			
γ_W	0.5	5	50
$e_{1_{rms}}$	0.635612	0.510440	0.327864
$e_{2_{rms}}$	0.847229	0.680361	0.437013
$e_{3_{rms}}$	0.529808	0.425459	0.273330
$e_{4_{rms}}$	0.529814	0.425460	0.273312
e_{rms}	0.648681	0.520927	0.334615

In conclusion, an increase of γ_W does not affect the residual state error, but the transient is reduced. Inversely, a decrease of γ_W does not affect the residual state error, but the transient is extended since the ultimate bound of the state error does not depend on γ_W. Note that the identification performed well, even in the presence of disturbances at $t = 5s$.

3.5.2 Comparison

The parameters for the comparison are the same as in [74]. Three simulations were performed to verify the performance of the two identification algorithms. The first simulation compares the performance related to the residual state error. The second and third simulations were made to observe differences in the transient period. To identify the uncertain system (3.48), the proposed identification model (3.12) and the adaptive law (3.29) were implemented. In these two simulations, the initial conditions of the identification system were the same as in the Subsection 3.5.1. The nonlinear vector function σ is defined in (3.49) where $s(\cdot) = \frac{5}{1+e^{-0.5(\cdot)}}$ and the design parameters ψ and W_0 were chosen as $\psi = 1$ and $W_0 = \begin{bmatrix} I & \mid & 0 \end{bmatrix}$.

In the first simulation, a high disturbance was selected to better check the potential of the algorithms. The disturbance was chosen to be equal to $d_{x_1} = 8000cos(9t)$, $d_{x_2} = 10000cos(6t)$, $d_{x_3} = 6000sin(15t) + 10000e^{-t}$, and $d_{x_4} = 6000sin(15t) + 4000cos(20t)$. The parameters chosen for this simulation were $L = 140000I$, $P = 15L$, $\psi = 0.00005$, and $\gamma_W = 5$.

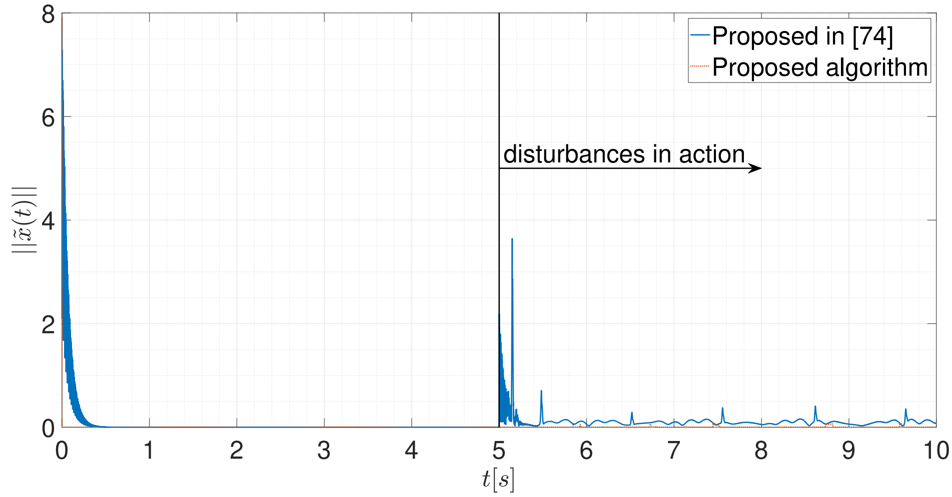

Figure 3.7 – Comparison of the state error norm.

Table 3.9 – Root mean square of state errors for t = [0 10] seconds and consider that $\psi = 5{\cdot}10^{-5}$, $L = 14{\cdot}I{\cdot}10^4$, and $P = 21{\cdot}I{\cdot}10^5$.

Root Mean Square of State Errors			
			e_{rms} in proposed algorithm
ψ	L	P	0.050721
0.1ψ	L	P	0.051067
10ψ	L	P	0.052372
ψ	$0.1L$	P	0.118153
ψ	$10L$	P	0.016807
ψ	L	$0.1P$	0.052867
ψ	L	$10P$	0.043514

As can be seen in Figure 3.7, the performance of both algorithms is satisfactory, and the state error norms are close to zero. This is also because both schemes use LPNNs, but it should be noted that the algorithm proposed in [74] induces a state error far from zero. The proposed algorithm performance was better due to the adjustment of the design parameters.

Table 3.9 shows the different RMSE values for changes in the parameters ψ, L, and P. Changes in the value of ψ do not have much impact on the RMSE values, although it shows a slight increase in the RMSE value as expected in (3.38). Larger values of L and P lead to higher RMSE values, as expected.

In the second and third simulations, the disturbances were chosen as $d_{x_1} = 5cos(9t)x_1$, $d_{x_2} = 15cos(6t)x_2x_3$, $d_{x_3} = x_4[sin(15t) + 10e^{-t}]$, and $d_{x_4} = 3sin(15t) + 2cos(20t)$. The parameters chosen for this simulation were $L = 2I$, $P = 15L$, and $\psi = 1$. In the second simulation $\gamma_W = 0.5$ and in the third $\gamma_W = 50$. The Figures 3.8 and 3.9 shows the results of the simulations.

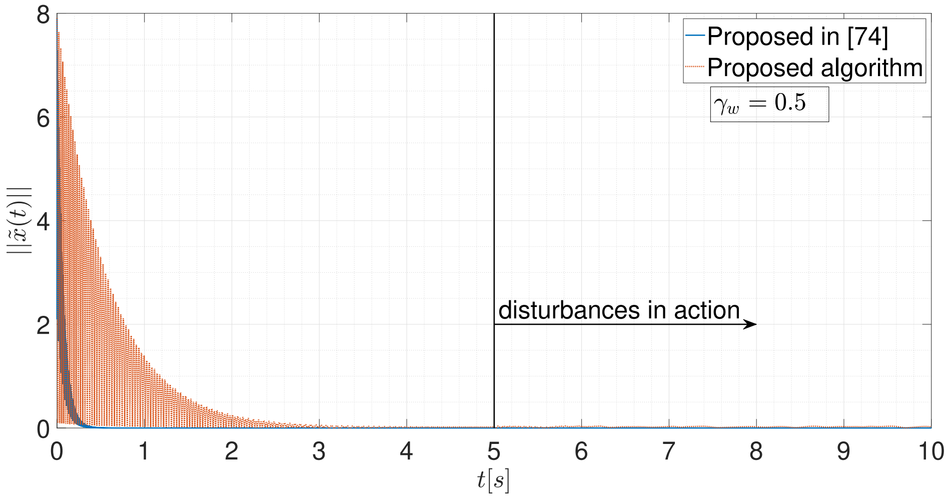

Figure 3.8 – Comparison of the state error norm when $\gamma_w = 0.5$.

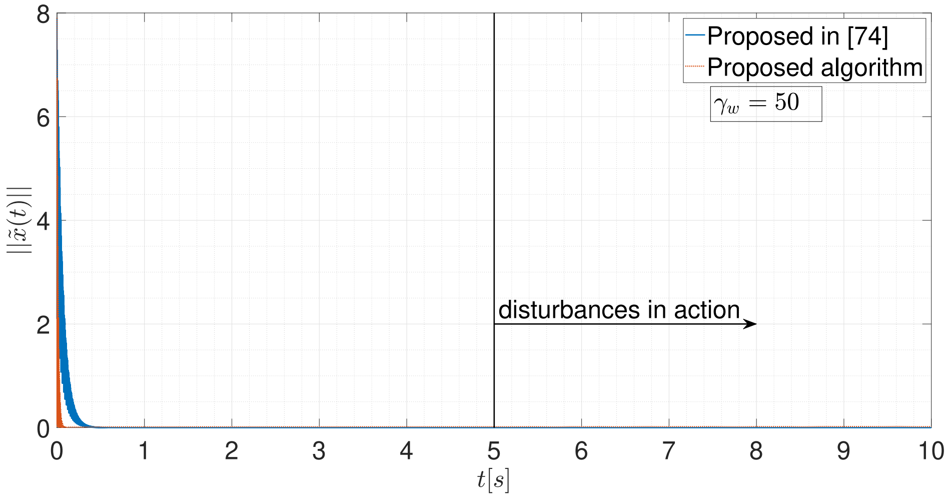

Figure 3.9 – Comparison of the state error norm when $\gamma_w = 50$.

Table 3.10 – Root mean square of state errors for t = [0 10] seconds and consider that $\psi = 1$, $L = 2{\cdot}I$, and $L = 30{\cdot}I$.

Root Mean Square of State Errors in the Proposed Algorithm				
			e_{rms} when $\gamma_W = 0.5$	e_{rms} when $\gamma_W = 50$
ψ	L	P	2.897384	0.610932
0.1ψ	L	P	3.233244	1.606452
10ψ	L	P	1.686113	0.271689
ψ	$0.1L$	P	3.374821	0.614588
ψ	$10L$	P	1.499498	0.579801
ψ	L	$0.1P$	2.909199	0.626446
ψ	L	$10P$	2.645071	0.783904

It should be noted that from the Figures 3.8 and 3.9, the residual state error, after the transient, is approximately null for both algorithms. However, there is an essential difference in both transients. Chapter approach allows for simple manipulation of the transient irrespectively of the residual state error, which can be seen in Figures 3.8 and 3.9. This feature is the main advantage of the chapter scheme compared with that in [74].

Table 3.10 shows results when changing ψ, L, and P. Larger values of ψ and L lead to lower RMSE values in all cases, and larger P values lead to lower RMSE values in most cases. When $\gamma_W = 50$, a higher value of P led to a higher value of RMSE because the transient duration increased. The results of Table 3.10 were different from tables 3.3 - 3.8 because here there were changes not only in the transient, but also in the residual state error. So it was expected that larger values of ψ would lead to larger transients, and larger values of L and P would lead to smaller RMSE.

3.5.3 Welding System

Consider that the GMAW can be modeled by (3.3), where the input vector is defined by $u = [u_i], i = 1, 2, 3$ and the state vector is defined by $x = [x_j], j = 1, 2, 3, 4, 5, 6$. More specifically, u_1 is the nominal wire speed, u_2 is the open-circuit voltage, u_3 is the nominal welding speed, x_1 is the welding current, x_2 is the arc voltage, x_3 is the stick out, x_4 is the width, x_5 is the reinforcement, and x_6 is the penetration of the weld bead. The aim is to obtain a parameterization based on neural networks and identify their weights through stable adaptive laws.

Figure 3.10 shows the experimental setup that was implemented to measure the actual values of the welding current, arc length, arc voltage, and stick out, as well as the geometric parameters of the weld bead, such as width, reinforcement, and penetration. The welding source used was a multiprocess TransPuls Synergic 5000 Fronius; the GMAW welding torch is a Watercooled TTW 4500 Robacta, which uses a welding wire AWS ER 316, 1.2 mm, and the shielding gas was 96% argon plus 4% CO2 with a flow rate of 15 l/min. The test piece was a sheet of 1020 steel with a thickness of 6 mm, width 35 mm, and length of 300 mm, which is moved by a prismatic position table actuated by a ball screw and stepper motor. A program in Labview® was designed to control the speed of the table (welding speed), the firing of the GMAW process, the welding current, and the ending of the process. The NI USB 6009, National Instruments, the ROB 5000, and Fronius were used as communication interfaces with the welding source. A web camera of 2 megapixels was adapted for infrared at an acquisition frequency of 30-50 frames per second (fps). A High pass infrared filter of 1000 nm, an infrared attenuator for wavelength above 800 nm (attenuator and filter high-pass), and a zoom lens of 18-108 mm/F2.5 were also employed.

The infrared camera was fixed at a distance of 0.8 m from the target to view the weld pool laterally. In other words, the angles between the lens axis and the welding were 90°C, and with an inclination of 0°C to the horizontal plane. However, the width of the bead was filmed using an arrangement of mirrors near the torch. The aim was to focus on the width and reinforcement bead during the welding process. The camera position and mirrors allowed for the observation of these variables. On the other hand, to estimate the penetration, it was necessary to longitudinally cut the weld bead and chemically process it to enhance the obtained images (Figures 3.17 - 3.19). All pictures were then recorded on the computer, which controls the camera, to be subsequently synchronized, processed, and analyzed. The experiment was accomplished with a base metal at room temperature (25 °C), welding voltage ranging from 17 to 32 V, welding speed ranging from 6.5 to 13.5 mm/s, and wire-speed ranging from 4.7 to 8.3 m/min, as depicted in Figures 3.11 - 3.13. All parameters are listed in Table 3.11, the obtained measurements are included in Figures 3.14 - 3.19, and the estimated weight norm is shown in Figure 3.20.

Table 3.11 – Welding parameters.

Material	1020 steel of 300 X 35 X 6 mm
Type of gas	96% Argon, 4% CO2
Gas flow	15 L/min
Electrode	Negative
Wire	15 GMAW - AWS Stainless ER 316, 1.2 mm
Camera acquisition rate (frames per second -fps)	50 fps
Plate-electrode distance	3 mm
Open circuit voltage	17 - 32 V
Welding speed	3 mm
Plate-electrode distance	3 mm
Open circuit voltage	17 - 32 V
Welding speed	6.5 to 13.5 mm/s with variable acceleration
Wire speed	4.7 to 8.3 m/min

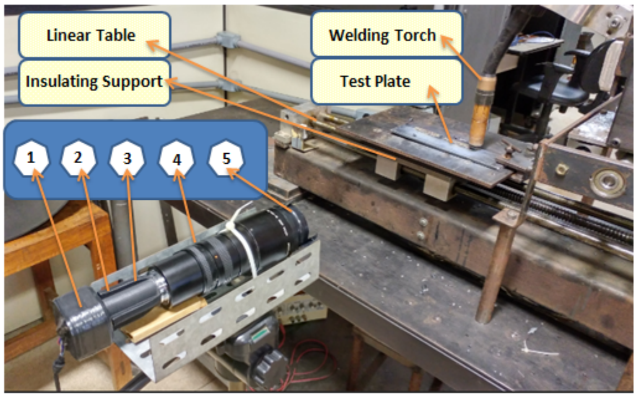

Figure 3.10 – Experimental setup: (1) web camera with 2.0 megapixels, 50 fps, and adapted for infrared, (2) high pass infrared filter of 1000 nm and 1050 nm, (3) infrared attenuator for wavelength above 800 nm (attenuator and filter high-pass), (4) telephoto lens with a zoom of 18 - 108 / 2.5, and (5) polarizer.

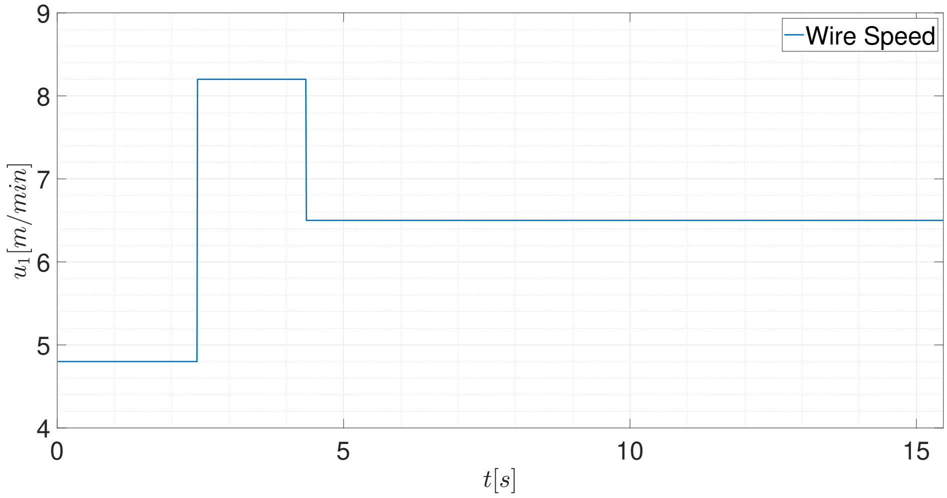

Figure 3.11 – Nominal Wire Speed (m/min).

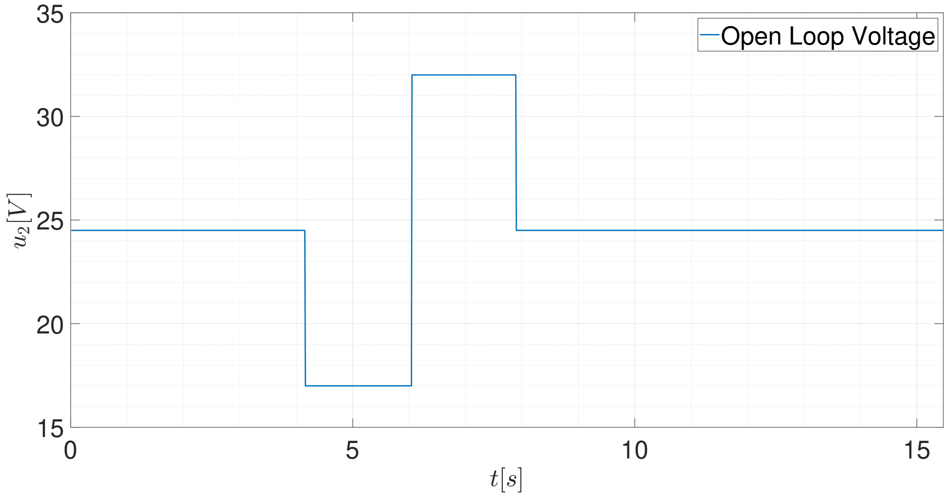

Figure 3.12 – Nominal Open Circuit Voltage (V).

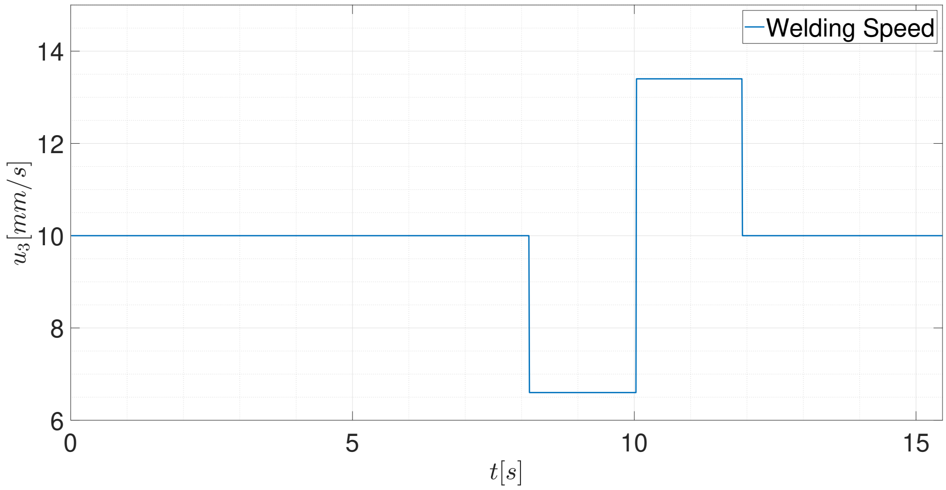

Figure 3.13 – Nominal Welding Speed (m/s).

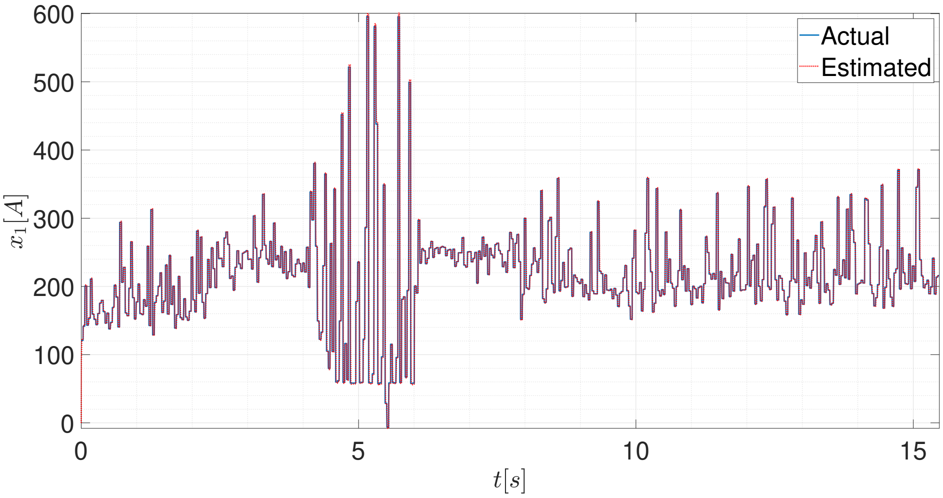

Figure 3.14 – Performance in the estimation of the welding current (x_1).

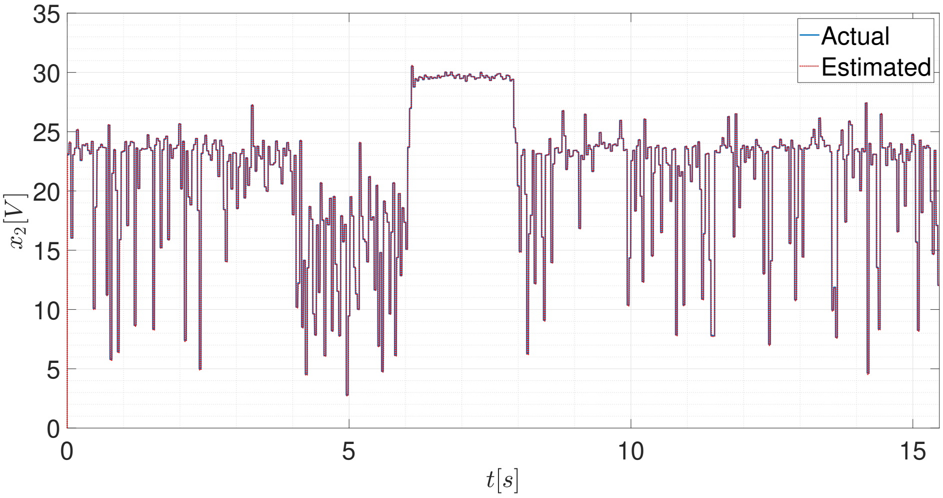

Figure 3.15 – Performance in the estimation of the arc voltage (x_2).

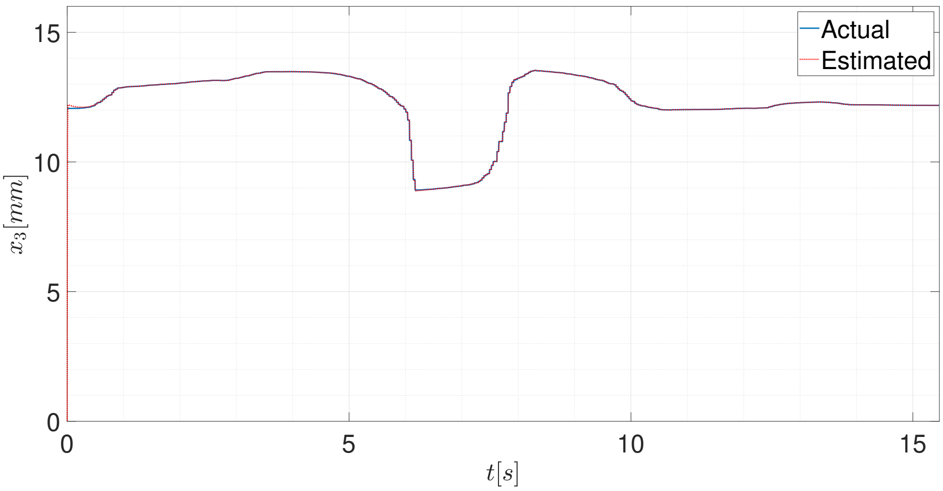

Figure 3.16 – Performance in the estimation of the stick out (x_3).

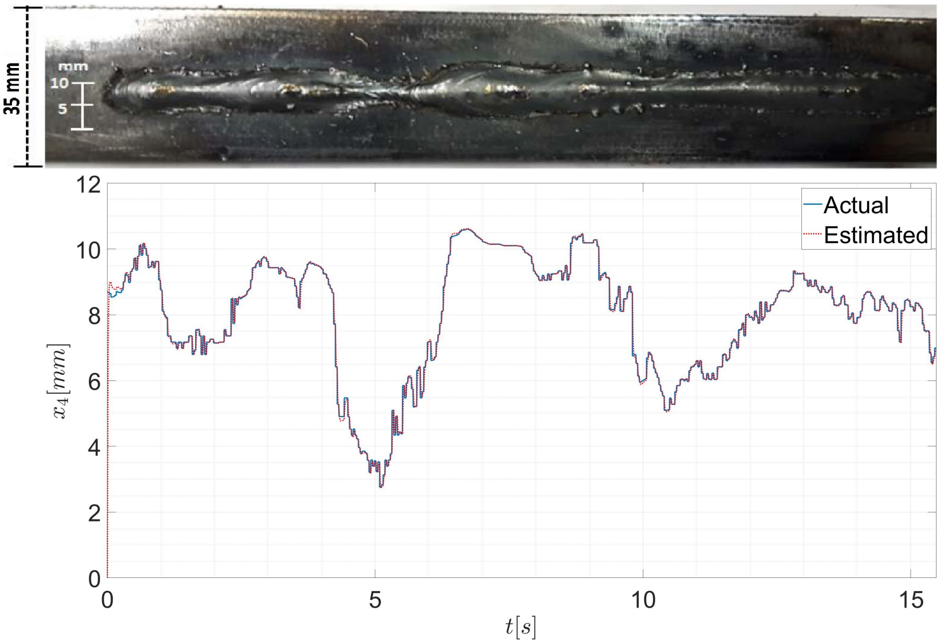

Figure 3.17 – Performance in the estimation of the width (x_4).

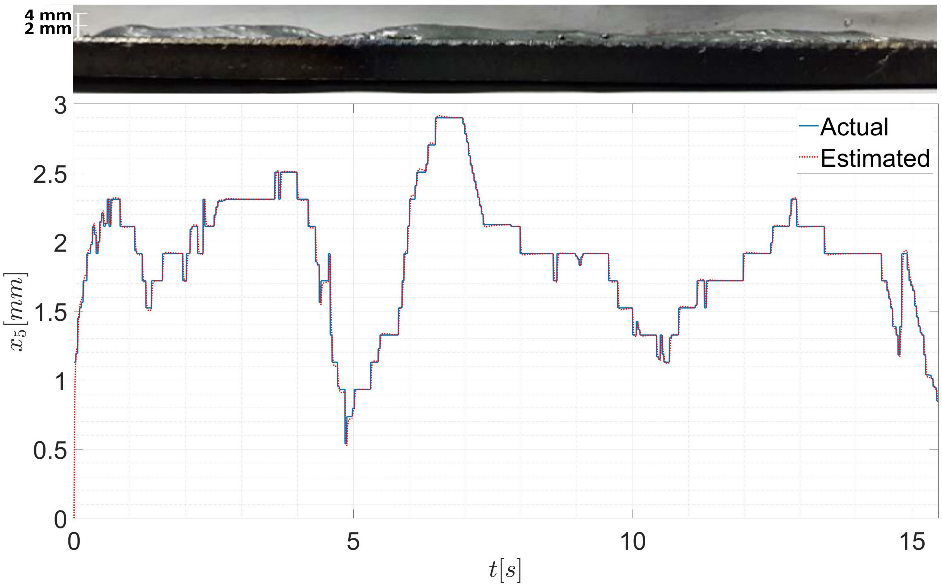

Figure 3.18 – Performance in the estimation of the reinforcement (x_5).

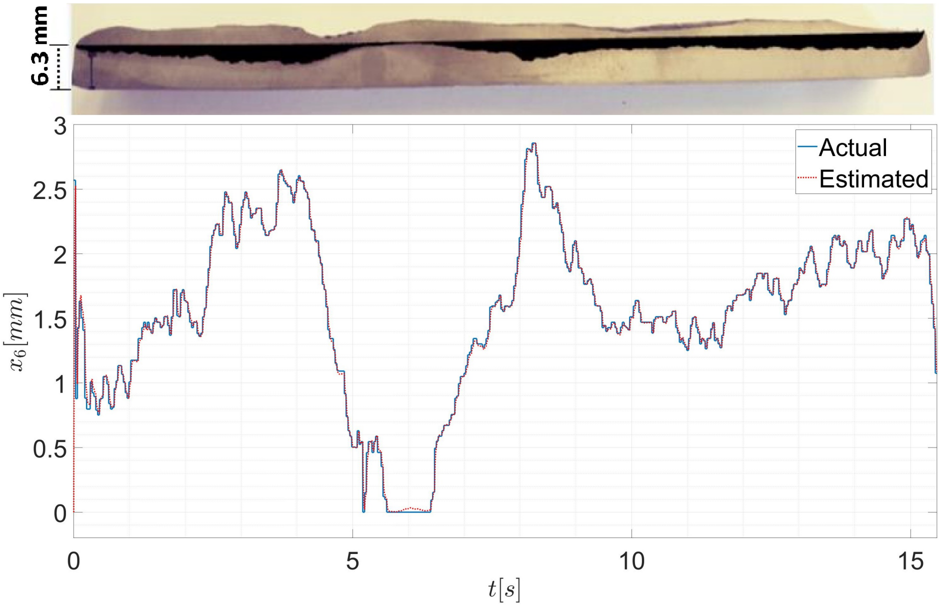

Figure 3.19 – Performance in the estimation of the penetration (x_6).

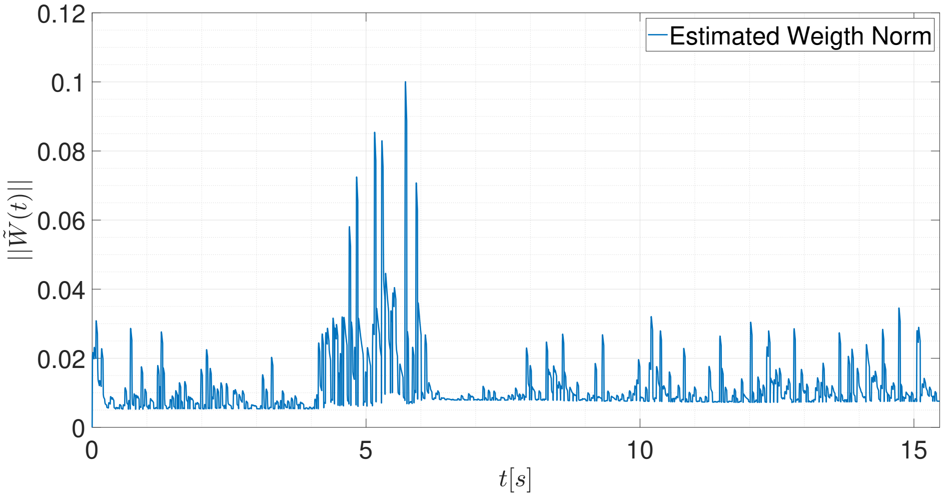

Figure 3.20 – Estimation Weight Norm.

The proposed identification model (3.12) and the adaptive law (3.29) were implemented. The initial conditions for the identification model were $\hat{x}_1(0) = 0$, $\hat{x}_2(0) = 0$, $\hat{x}_3(0) = 0$, $\hat{x}_4(0) = 0.01$, $\hat{x}_5(0) = 0$, $\hat{x}_6(0) = 0$, and $\hat{W}(0) = 0$ to evaluate the performance of the proposed algorithm under adverse initial conditions. The design matrices were chosen as

$$L = P = 20 \begin{bmatrix} 5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.50}$$

The nonlinear vector function σ was selected as

$$\sigma^T = \begin{matrix} [s(\hat{x}_1), & s(\hat{x}_2), & s(\hat{x}_3), & s(\hat{x}_4), & s(\hat{x}_5), & s(\hat{x}_6), \\ s^2(\hat{x}_1), & s^2(\hat{x}_2), & s^2(\hat{x}_3), & s^2(\hat{x}_4), & s^2(\hat{x}_5), & s^2(\hat{x}_6)] \end{matrix} \tag{3.51}$$

The chosen activation function was $s(\cdot) = \frac{5}{1+e^{-0.5(\cdot)}}$. The design parameters ψ and γ_W were chosen as $\psi = 0.001$ and $\gamma_W = 0.001$. W_0 was chosen as $W_0^T = \begin{bmatrix} I \\ 0 \end{bmatrix}$. Figures 3.11 - 3.13 show the inputs, and Figures 3.14 - 3.19 show the estimation performance of the states, and Figure 3.20 show the Frobenius norm of the weight matrix. Observe that the simulation corroborates the theoretical results since the estimated states converge to a neighborhood of the actual states, and the residual state error and transient can be adjusted according to the user requirements.

Table 3.12 shows results when changing ψ, L, P, and γ_W. No significant changes in values from RMSE to changes in ψ were observed. Lower values of RMSE were observed for higher values of L in all cases and larger values of P and γ_W in most cases. The cases in which the RMSE increased with the increase of P must have happened because of small transient appearances after not derivable points in figures 3.18 and 3.19. An increase in γ_W led to lower transient and lower RMSE values.

Table 3.12 – Root mean square of state errors for t = [0 10] seconds and consider that $\psi = 1$, $L = 2{\cdot}I$, and $L = 30{\cdot}I$.

Root Mean Square of State Errors in the Proposed Algorithm									
				e_{rms} of x_1	e_{rms} of x_2	e_{rms} of x_3	e_{rms} of x_4	e_{rms} of x_5	e_{rms} of x_6
ψ	L	P	γ_W	39.000338	1.955200	0.330874	0.397476	0.072766	0.129421
0.1ψ	L	P	γ_W	39.000328	1.955198	0.330874	0.397476	0.072766	0.129421
10ψ	L	P	γ_W	39.000439	1.955217	0.330874	0.397477	0.072767	0.129421
ψ	$0.1L$	P	γ_W	69.034589	3.277834	0.562885	0.704905	0.132829	0.132829
ψ	$10L$	P	γ_W	28.495272	1.433543	0.244221	0.293705	0.054324	0.095736
ψ	L	$0.1P$	γ_W	38.828820	1.954766	0.331411	0.398783	0.073730	0.128943
ψ	L	$10P$	γ_W	37.621412	1.894895	0.333088	0.393300	0.393300	0.132241
ψ	L	P	$0.1\gamma_W$	38.828112	1.953731	0.331417	0.398784	0.073731	0.128944
ψ	L	P	$10\gamma_W$	37.590665	1.896355	0.335465	0.394251	0.072550	0.132494

It should be noted that, resulting from the stimulus used to excite the process (open-circuit voltage, welding speed, and wire-speed), the welding oscillates between different transfer modes (short-circuit, globular, and spray). To the best of the author's knowledge, an identification scheme to tackle different transfer modes of welding in a unified form has not yet been proposed in the literature.

3.6 SUMMARY

In this chapter, a unified method to identify uncertain nonlinear systems has been proposed. Based on series-parallel and parallel configurations, a neural identification model to make both all errors uniformly bounded as well as the state error uniformly ultimately bounded was introduced. It has been shown that the proposed scheme allows for the manipulation of the identification transient without changing the ultimate bound for the state error. Simulations have been performed to validate the theoretical results and depict the peculiarities and application of the proposed method. Future works include the application of the proposed identification scheme to observation and output feedback control problems, like [127, 128], where the states were estimated through an adaptive observer.

4 SCHEME FOR SYNCHRONIZATION AND ENCRYPTION OF A 3D-CHAOTIC LORENZ SYSTEM AND LYAPUNOV ANALYSIS

The research outcomes of this chapter have been published as a conference paper entitled "Scheme for chaos-based encryption and lyapunov analysis" in [10]. *This chapter has extended and improved some parts compared to the original paper.*

Many synchronization schemes of chaotic systems appeared in the literature [129–131], mainly with the purpose of secure telecommunication. The objective of these works is to encrypt confidential information with the use of chaotic circuits. Thereunto, it is necessary to use a transmitter system based on chaotic circuits to encode confidential information and another chaotic system to receive to reconstruct the encoded message. Among all different proposals found in the literature, for example, chaotic masking and parameters modulation [132], it is observed that the first can be implemented with low costs since the system can be built via analog electronics.

However, the proposals found in the literature usually have a lot of common problems, such as the use of chaotic systems completely actuated as receivers. In other words, it is assumed equality of dimensions between the state and input, so the control signal is presented in all the differential equations that define the slave chaotic system. Besides that, the lack of disturbances in the stability analysis is the biggest and most relevant problem that negatively impacts the robustness of the proposed algorithms. Note that disturbances, internal or external, are inevitable to analog electronics implementations once the electronic components have, for example, uncertainties, heating, nonideal behavior. On the other hand, disturbances are inevitable in the transmission lines in real applications.

More specifically, in [133–135] it is proposed schemes for secure telecommunication based on Lyapunov stability theory and in different nonlinear control techniques (Backstepping, sliding mode, adaptive, projective, multi-scroll, and so on [133–139]). However, most papers consider a control dimension equal to the chaotic system order [53, 140, 141] and do not consider the presence of disturbances in the stability analysis. In [142], it is considered the presence of multiplicative disturbances and control with reduced order, but the system requires the use of an observer to estimate the states that are not available to measure, which makes the system complex.

This chapter intends to design a secure telecommunication system that solves the mentioned problems based on the information above. The design is based on Lyapunov stability theory [24]; however, in oppose to [143], and [144] in the analysis, it is considered the presence of bounded disturbances in all the differential equations in the slave chaotic system explicitly. More precisely, a Lorenz system with a simple control that acts in only one differential equation makes the design easy to implement via analog components. More exactly, the synchronization scheme of this chapter presents the following characteristics.

1) One of the first underactuated synchronization schemes for Lorenz systems is proposed.

2) The control is one-dimensional.

3) The synchronization scheme uses a simple proportional control.

4) The proposed scheme considers the presence of disturbances in all states in the stability analysis.

5) The proposed scheme is applied to secure communication.

As far as the author knows, no work prior to this chapter combined all these characteristics simultaneously, which is the main contribution of this chapter.

The chapter is organized as follows. In Section 4.1, the problem and main assumptions are introduced. In Section 4.2, the synchronization error, the control law, and the Lyapunov analysis are presented. In Section 4.3, the secure communication case is analysed. Section 4.4 shows simulations and the application of the proposed method in secure telecommunication. The conclusions of the chapter are done in Section 4.5.

4.1 PROBLEM FORMULATION

Consider the chaotic system [61, 145] described by

$$\begin{aligned}
\dot{x}(t) &= -ax(t) + ay(t) \\
\dot{y}(t) &= bx(t) - y - 4dx(t)z(t) + u \\
\dot{z}(t) &= -cz + dx(t)y(t)
\end{aligned} \tag{4.1}$$

where $a = 16$, $b = 45.6$, $c = 4$, and $d = 5$. Note that this system with two attractors was proposed by Lorenz, where $x(t)$, $y(t)$, and $z(t)$ are the system states and a, b, c and d are real constants. From (4.1), consider the master and slave systems with disturbances:

$$\begin{aligned} \dot{x}_m &= -ax_m + ay_m + h_{1m}(t) \\ \dot{y}_m &= bx_m - y_m - 4dx_m z_m + h_{2m}(t) \\ \dot{z}_m &= -cz_m + dx_m y_m + h_{3m}(t) \end{aligned} \tag{4.2}$$

$$\begin{aligned} \dot{x}_s &= -ax_s + ay_s + h_{1s}(t) \\ \dot{y}_s &= bx_s - y_s - 4dx_s z_s + h_{2s}(t) + u \\ \dot{z}_s &= -cz_s + dx_s y_s + h_{3s}(t) \end{aligned} \tag{4.3}$$

where x_m, y_m, and z_m are the states of the master system; x_s, y_s, and z_s, are the states of the slave system; h_{1m}, h_{2m}, and h_{3m} are the disturbances in the master system, h_{1s}, h_{2s} and h_{3s} are the disturbances in the slave system, and u is the signal control. Then, this chapter aim is to synchronize the systems (4.2) and (4.3) in which the slave system allows only one scalar control signal acting in only one state.

Remark 4.1.1 Once the system (4.1) is chaotic, its behavior is related to the initial conditions and sensitive to changes. Because of that and also by the fact that it has an aperiodic behavior the synchronization of chaotic systems tends to be more challenging than other dynamic systems.

Remark 4.1.2 Note that the synchronizer only has access to information from the actuated master states, that is, from the states where there is a control signal. In the (4.3) system, the control u cannot have the presence of the states x_m and z_m. However, the other states of the master system and all the states of the slave system are available.

Fact 4.1.1 In [145] was proved that (4.1), and consequently (4.2), is chaotic and that the system is dissipative. A consequence of the system being dissipative is also being bounded. With the boundedness of the system (4.1), the following inequalities are true:

$$\begin{aligned} |x_m(t)| &\leq \bar{x} \\ |y_m(t)| &\leq \bar{y} \\ |z_m(t)| &\leq \bar{z} \end{aligned} \tag{4.4}$$

$\forall t \geq 0$, where $\bar{x}$, $\bar{y}$, and $\bar{z}$ are unknown positive constants.

Assumption 4.1.1 The disturbances are bounded. More specifically

$$\begin{aligned} \tilde{h}_1(t) &= h_{1s}(t) - h_{1m}(t) \\ \tilde{h}_2(t) &= h_{2s}(t) - h_{2m}(t) \\ \tilde{h}_3(t) &= h_{3s}(t) - h_{3m}(t) \end{aligned} \tag{4.5}$$

$\forall t \geq 0$, where $|\tilde{h}_1(t)| \leq \bar{h}_1$, $|\tilde{h}_2(t)| \leq \bar{h}_2$, and $|\tilde{h}_3(t)| \leq \bar{h}_3$, with $\bar{h}_1$, $\bar{h}_2$, and $\bar{h}_3$ being unknown constants.

REMARK 4.1.3 The objective of considering the disturbances h in (4.2) and (4.3) is to emphasize that the synchronization is valid even in the presence of unexpected changes in the system dynamics. These changes can occur due to failures, climate changes, or equipment aging.

4.2 SYNCHRONIZATION EQUATION ERROR AND PROPOSED SIGNAL CONTROL

Define the dynamic of the synchronization errors as

$$\begin{cases} \dot{e}_1 &= \dot{x}_s - \dot{x}_m \\ \dot{e}_2 &= \dot{y}_s - \dot{y}_m \\ \dot{e}_3 &= \dot{z}_s - \dot{z}_m \end{cases} \tag{4.6}$$

By using and (4.2) and (4.3) in (4.6), and employing (4.4), results

$$\begin{cases} \dot{e}_1 &= -ae_1 + ae_2 + \tilde{h}_1 \\ \dot{e}_2 &= -e_2 + be_1 - 4d(e_1e_3 + e_1z_m + e_3x_m) + \tilde{h}_2 + u \\ \dot{e}_3 &= -ce_3 + d(e_1e_2 + e_1y_m + e_2x_m) + \tilde{h}_3 \end{cases} \tag{4.7}$$

Once the dynamic of errors is defined, an appropriate control signal is provided so that the slave system synchronizes.

THEOREM 4.2.1 Consider the master and slave systems (4.2) and (4.3), and the proportional control law

$$u = -\psi e_2 \tag{4.8}$$

If,

$$\psi > \delta \tag{4.9}$$

Then, the synchronization error is uniformly ultimately bounded and converges in finite-time to the compact set

$$\Omega = \left\{ e \in \Re^3 \mid ||e|| \leq \theta \right\} \tag{4.10}$$

where $\delta = 0.5\sigma_2\bar{h}_2^2 + 0.5\sigma_5^{-1}(b^2 + a^2\gamma^2 + 16\bar{z}^2) - 1$, $\theta = \sqrt{\frac{\beta}{\rho}}$, $\beta = \beta_u + \beta_n$, $\rho = min\{\rho_1, \rho_2, \rho_3\}$, $\beta_u = 0.5\sigma_2^{-1}$, $\beta_n = 0.5(\gamma\sigma_1^{-1}\bar{h}_1^2 + 4\sigma_3^{-1}\bar{h}_3^2)$, $\rho_1 = a\gamma - 0.5\gamma\sigma_1 - 2d\sigma_4\bar{y}^2 - 0.5\sigma_5$, $\rho_2 = \psi - \delta$, $\rho_3 = 4c - 2\sigma_3 - 2d\sigma_4^{-1}$, $\|e\|^2 = e_1^2 + e_2^2 + e_3^2$, and $\sigma_i, i = 1, ..., 5$ are positive constants.

Proof.

Consider the following Lyapunov function candidate

$$V = \frac{1}{2}(\gamma e_1^2 + e_2^2 + 4e_3^2) \tag{4.11}$$

where $\gamma > 0$. The time-derivative of (4.11) results

$$\dot{V} = \gamma e_1 \dot{e}_1 + e_2 \dot{e}_2 + 4e_3 \dot{e}_3 \tag{4.12}$$

By replacing (4.7) in (4.12) implies

$$\begin{aligned} \dot{V} &= \gamma e_1(-ae_1 + ae_2 + \tilde{h}_1) + e_2[-e_2 + be_1 - 4d(e_1e_3 + e_1z_m + e_3x_m) + \tilde{h}_2 \\ &\quad + u] + 4e_3[-ce_3 + d(e_1e_2 + e_1y_m + e_2x_m) + \tilde{h}_3] \end{aligned} \tag{4.13}$$

Replacing (4.8) in (4.13) results

$$\begin{aligned} \dot{V} &= -a\gamma e_1^2 - e_2^2(\psi + 1) - 4ce_3^2 + \gamma e_1\tilde{h}_1 + e_2\tilde{h}_2 + 4e_3\tilde{h}_3 + 4de_1e_3y_m \\ &\quad + e_2e_1(b + a\gamma - 4z_m) \end{aligned} \tag{4.14}$$

Thus, analyzing when $\dot{V} \leq 0$, by employing inequality of Young, the Assumption 4.1.1, and the Fact 4.1.1, so

$$\begin{aligned} &\gamma e_1\tilde{h}_1 \leq 0.5\gamma(\sigma_1 e_1^2 + \sigma_1^{-1}\bar{h}_1^2) \\ &e_2\tilde{h}_2 \leq 0.5(\sigma_2\bar{h}_2^2 e_2^2 + \sigma_2^{-1}) \\ &4e_3\tilde{h}_3 \leq 2(\sigma_3 e_3^2 + \sigma_3^{-1}\bar{h}_3^2) \\ &4de_1e_3y_m \leq 2d(\sigma_4^{-1}e_1^2\bar{y}^2 + \sigma_4 e_3^2) \\ &e_1e_2(b + a\gamma - 4z_m) \leq 0.5[\sigma_5 e_1^2 + \sigma_5^{-1}e_2^2(b^2 + a^2\gamma^2 + 16\bar{z}^2)] \end{aligned} \tag{4.15}$$

replacing (4.15) and rewritten (4.14) implies

$$\dot{V} \leq -e_1^2\rho_1 - e_2^2\rho_2 - e_3^2\rho_3 + \beta_u + \beta_n \tag{4.16}$$

Note that: 1) there are values of γ and $\sigma_i, i = 1, ..., 5$ that cause $\rho_1 > 0$ and $\rho_3 > 0$; and 2) ψ is chosen by the user so that (4.9) is satisfied, and, consequently, $\rho_2 > 0$. So there is a $\rho > 0$, and (4.16) can be rewritten as

$$\dot{V} \leq -\rho \left|\left|e\right|\right|^2 + \beta \tag{4.17}$$

Hence, $\dot{V} < 0$ as long as $||e|| > \theta$. Since θ is constant, then the synchronization errors are bounded. In the region Ω note that if for any reason $||e||$ is not part of that region Ω, $\dot{V}$ becomes defined negative and forces the synchronization error convergence to the region Ω, according to (4.17). In other words, if $\dot{V} < 0$ the error norm can only become smaller as the time increases. In addition, the convergence to the residual set Ω is in a finite time, due to the particular form of (4.17) [24]. Consequently, the synchronization error is uniformly ultimately bounded and converge to a sphere with radius θ (Figure 4.1). □

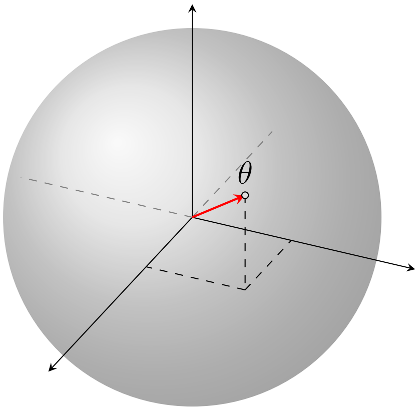

Figure 4.1 – 3D-Bounded set.

REMARK 4.2.1 It should be pointed out that the proposed scheme, unlike the precursor proposed [61], considers the presence of disturbances in all states, which is rarely considered in the literature, as far as the author know.

REMARK 4.2.2 The idea of the proposed method lies in the usage of the system structure and Lyapunov theory to design an adequate control law. Based on a trial-and-error procedure, all possibilities of underactuated control in the analysis was considered and the simplest one was chosen. Inequality of Young was used in the stability analysis in the process to make $\dot{V} < 0$ outside of a small compact set.

REMARK 4.2.3 Defining $D_i(i = 1, ..., 5)$ as the respective domains of $\sigma_i(i = 1, ..., 5)$, results

$$\begin{aligned} D_1 &= \{\sigma_1 \in \Re | 0 < \sigma_1 < 2a\} \\ D_2 &= \left\{\sigma_2 \in \Re | 0 < \sigma_2 < \frac{2(1+\psi)}{\bar{h}_2^2}\right\} \\ D_3 &= \{\sigma_3 \in \Re | 0 < \sigma_3 < 2c\} \\ D_4 &= \left\{\sigma_4 \in \Re | \frac{2d}{4c - 2\sigma_3} < \sigma_4 < \frac{a\gamma}{2d\bar{y}^2}\right\} \\ D_5 &= \left\{\sigma_5 \in \Re | \frac{b^2 + a^2\gamma^2 + 16\bar{z}^2}{2(1+\psi)} < \sigma_5 < 2a\gamma\right\} \end{aligned} \tag{4.18}$$

Observe that the residual synchronization error considered is affected by the control gain ψ, disturbances, and upper bounds for the states of the master system, as can be seen from (4.9). The performance for the actuated states can be arbitrarily enhanced by increasing ψ. For non-actuated states, it can not be guaranteed that a change in the gain of the control will cause the residual synchronization error to decrease, being this the main limitation of underactuated synchronization systems.

4.3 CHAOS-BASED SECURE COMMUNICATION

In order to have a well-posed problem to secure telecommunication case, the following assumption are done:

ASSUMPTION 4.3.1 It is assumed that the messages are bounded. More specifically,

$$|m_i(t)| \leq \bar{m}_i, i = 1, ..., 3 \tag{4.19}$$

$\forall t \geq 0$, where m_1, m_2, and m_3 are the original messages and $\bar{m}_1$, $\bar{m}_2$, and $\bar{m}_3$ are positive constants.

Further, motivated by [7], it can be defined

$$\begin{aligned} \hat{m}_1 &= s_1 - x_s \\ \hat{m}_2 &= s_2 - y_s \\ \hat{m}_3 &= s_3 - z_s \end{aligned} \tag{4.20}$$

where $s_1 = x_m + m_1$, $s_2 = y_m + m_2$, and $s_3 = z_m + m_3$ are the encoded messages; and $\hat{m}_1$, $\hat{m}_2$, and $\hat{m}_3$ are the decoded messages.

By using (4.20) and defining $\tilde{m}_i = \hat{m}_i - m_i, i = 1, ..., 3$, where $\tilde{m}_1$, $\tilde{m}_2$, and $\tilde{m}_3$ are the message errors, it can be concluded that

$$\tilde{m}_i = -e_i, i = 1, ..., 3 \tag{4.21}$$

Remark 4.3.1 It is worth noticing that the quality of the message reconstruction is the same as the synchronization, as shown in (4.21). That is, the boundedness of the message error is assured when the synchronization error is bounded.

Remark 4.3.2 An overview of the secure communication scheme can be seen in Figure 4.2.

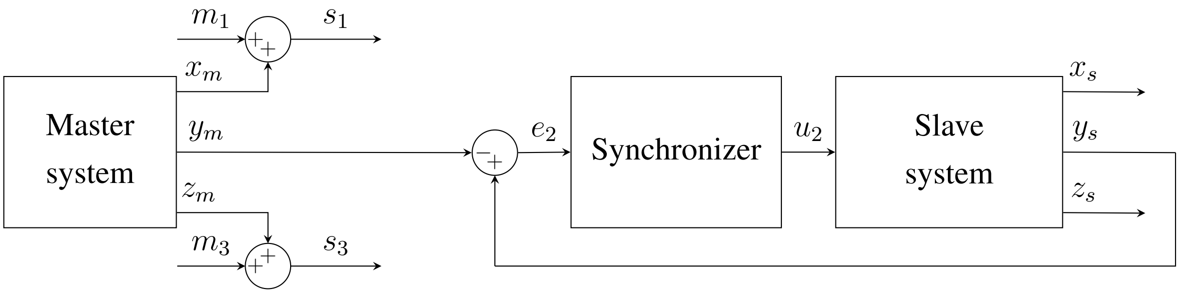

Figure 4.2 – Synchronization and secure communication scheme.

4.4 SIMULATION

To validate the signal control (4.8), computing simulations were made to synchronizae the systems (4.2) and (4.3). The simulations done in Matlab/Simulink 2020b® were made in a Windows 10 platform, with AMD Ryzen 7 1700 processor for all simulations, variable-step algorithm ODE45 solver, and relative and absolute tolerance of 10^{-10}. The simulation done in Multisim used the trapezoidal integration method, truncation error with factor 7, and maximum integration order of 2. The initial conditions for the master system were $x_m(0) = [0.2; -0.3; 0.4]$, and for the slave system were $x_s(0) = [0.1; 0.3; -0.1]$. The control law (4.8) was used to synchronize the master and slave systems, with $\psi = 50$. Disturbances in the slave system were considered, with $h_{1m} = 0$, $h_{2m} = 0$, $h_{3m} = 0$, $h_{1s} = 0.1sin(2t)$, $h_{2s} = 0.2sin(t)$, and $h_{3s} = 0.1sin(4t)$.

Figures 4.3 - 4.8 show the synchronization results obtained in MATLAB. It can be seen by the figures that the synchronization error is close to zero. The figures also show that even having only one control signal in one of the state equations (state y), the synchronization occurs for the three states. Synchronization errors are not exactly zero because of the presence of disturbances.

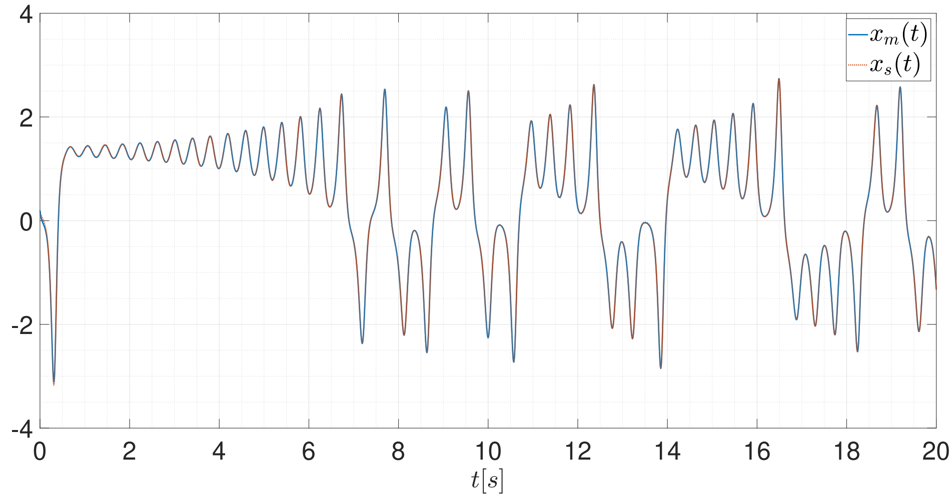

Figure 4.3 – Performance on the synchronization of the first state.

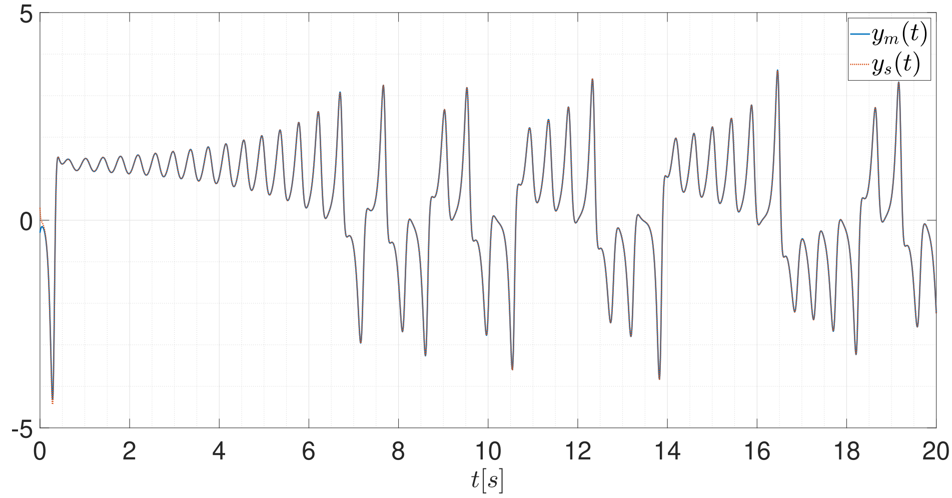

Figure 4.4 – Performance on the synchronization of the second state.

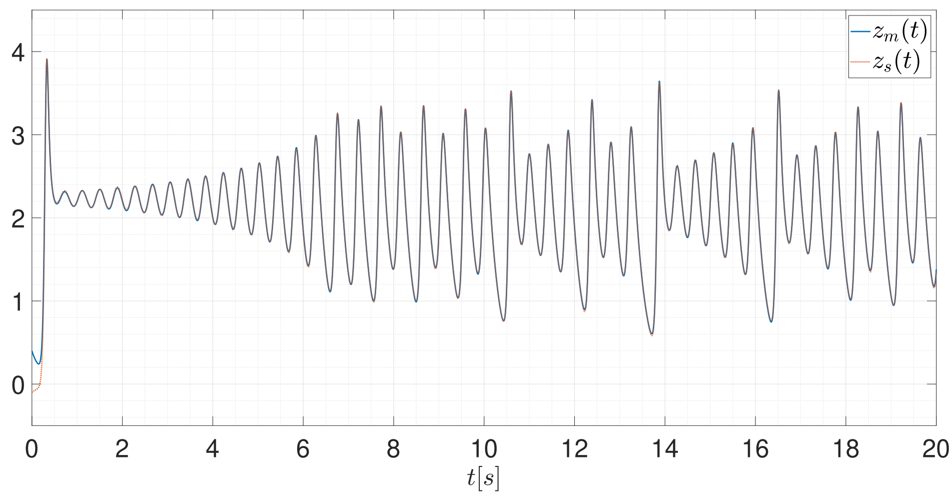

Figure 4.5 – Performance on the synchronization of the third state.

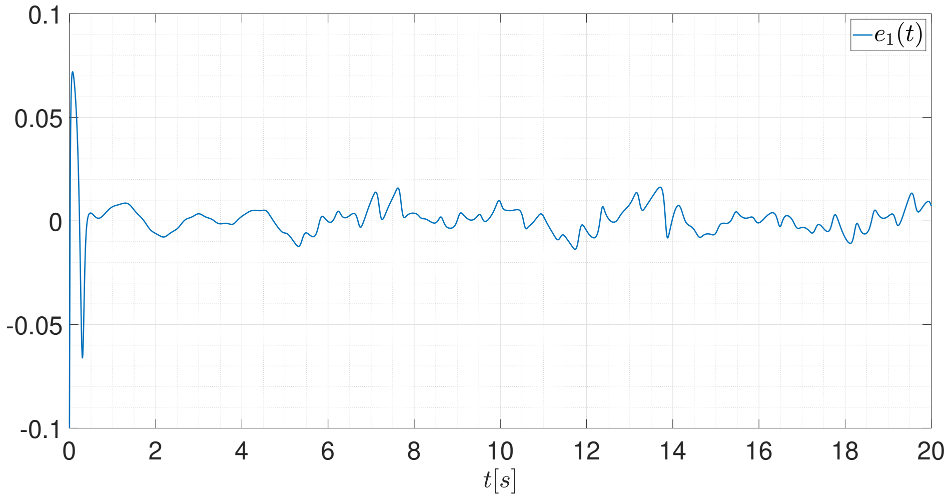

Figure 4.6 – Synchronization error of the first state.

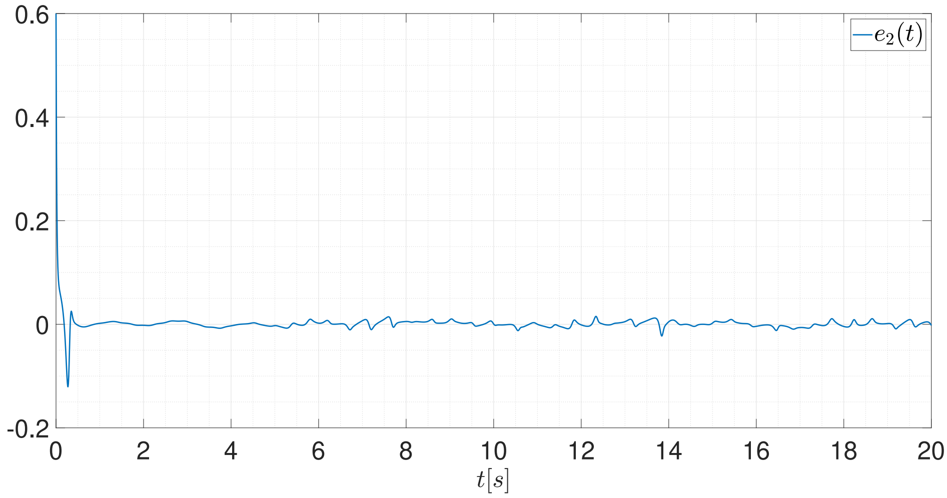

Figure 4.7 – Synchronization error of the second state.

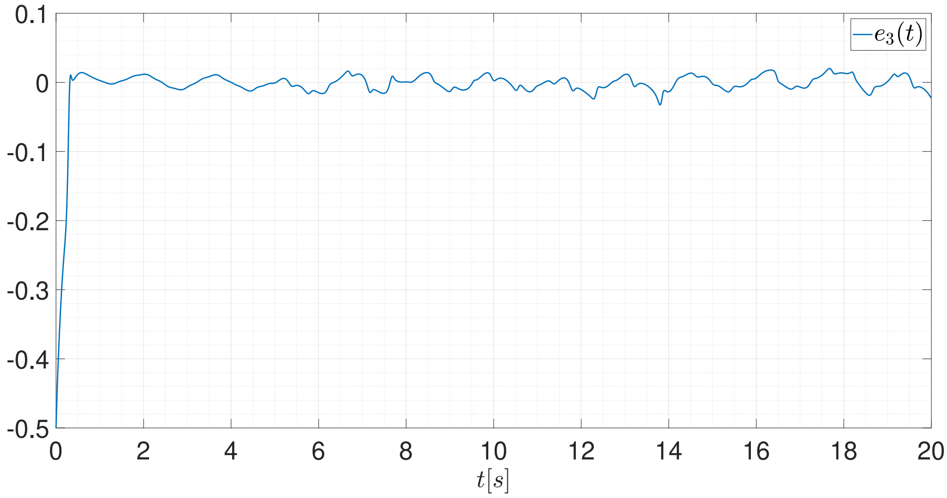

Figure 4.8 – Synchronization error of the third state.

Table 4.1 shows that the optimal control gain value is approximately $\psi = 50$. Smaller, according to theory, values of ψ can lead to larger values of β_u and smaller values of ρ_2, and, consequently, greater values of residual synchronization error. Larger values of ψ theoretically should not increase the synchronization error value. However, in computer simulations, high values of the control gain can lead to greater approximation errors.

Table 4.1 – Root mean square of state errors for t = [0 20] seconds and consider that $\psi = 50$.

Root Mean Square of State Errors in the Proposed Algorithm				
	$e_{1_{rms}}$	$e_{2_{rms}}$	$e_{3_{rms}}$	e_{rms}
0.01ψ	2.058822	2.246861	0.939715	2.984134
0.1ψ	0.578907	0.648567	0.312524	0.865672
ψ	0.009019	0.019277	0.040397	0.042332
10ψ	0.007865	0.015392	0.053506	0.054188
100ψ	0.008793	0.011410	0.058363	0.058797

Figures 4.9 - 4.12 show the secure communication case by using MATLAB. The messages are equal to $m_1 = 0.1sin(25.1328t)$ and $m_3 = 0.05cos(4t) + 0.05cos(t)$. It can be seen in Figures 4.9 - 4.10 that the original message is very different to encoded message and similar to decoded message. Figures 4.11 - 4.12 show the message errors. The errors are small, however are not null because of the presence of disturbances, as expected with the theory.

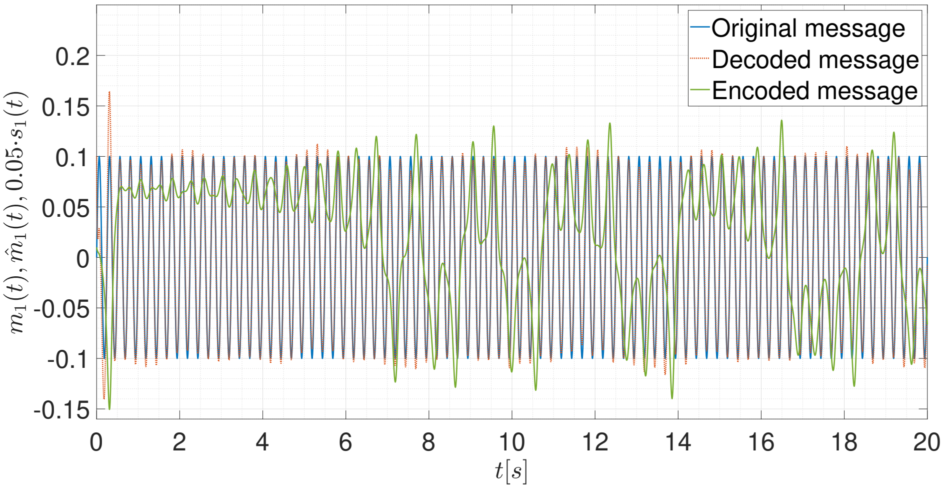

Figure 4.9 – Message introduced in the first state.

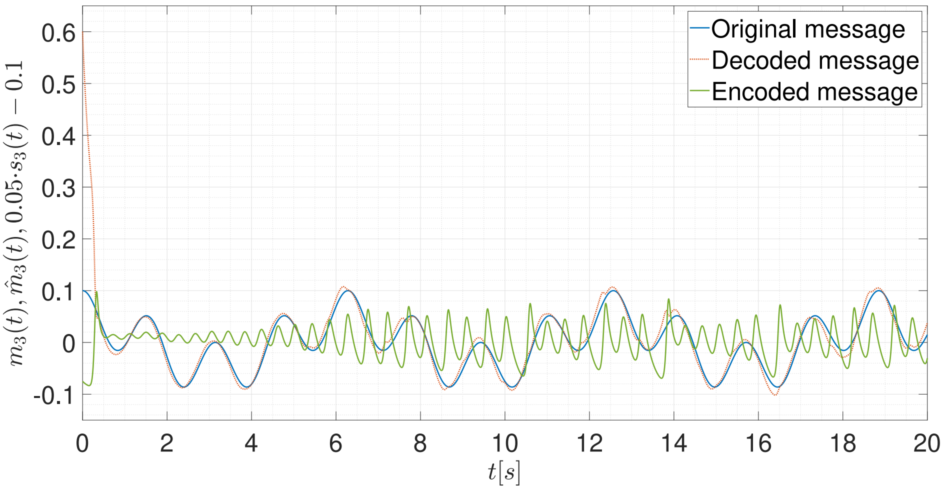

Figure 4.10 – Message introduced in the third state.

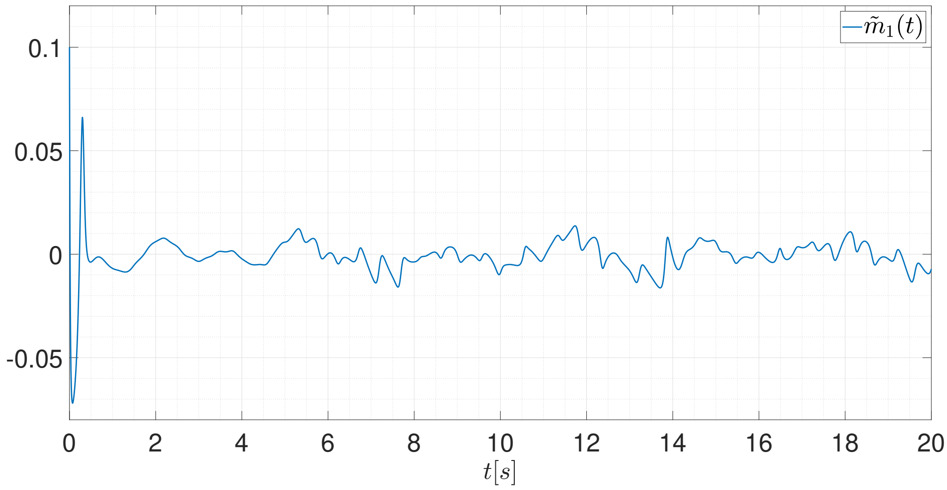

Figure 4.11 – First message error.

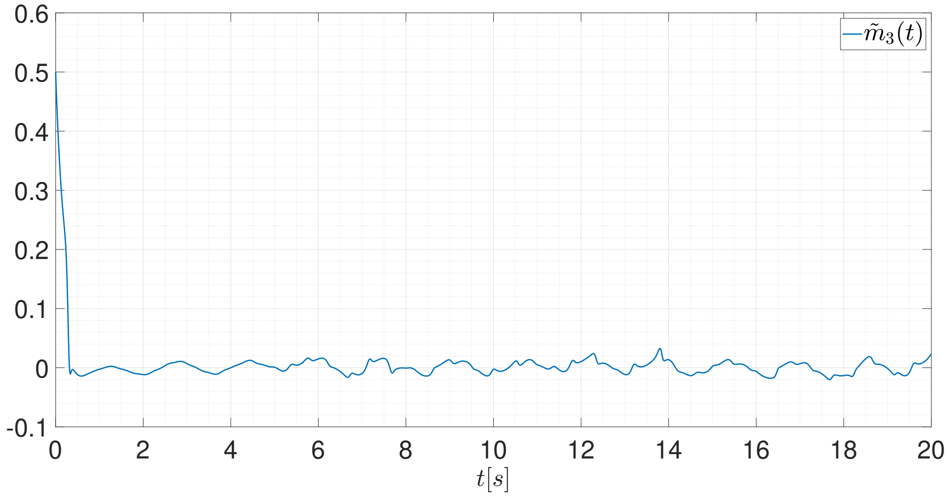

Figure 4.12 – Third message error.

In the sequence, the Multisim software is used to simulate an actual implementation via analog electronics of the master, slave, controller, encode, and decode systems. This can be done using operational amplifiers and the different configurations, such as the summing amplifier or the inverting amplifier. It is important to remark that the message could not be encoded using the state y because the controller would consider the message a disturbance, and it would not be possible to retrieve it. Figures 4.13 - 4.16 show the circuital implementation of those systems. It is good to note that ideal components are utilized in the simulations, so the non-ideal behaviors of the circuits were not considered, such as component heating or tolerances.

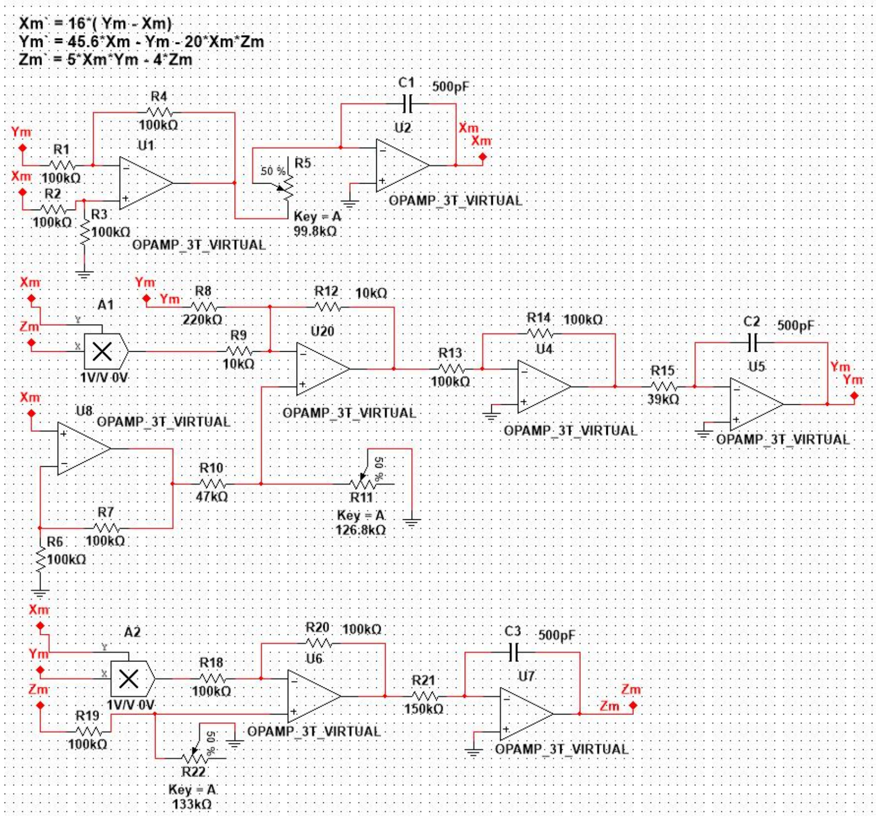

Figure 4.13 – Master circuit.

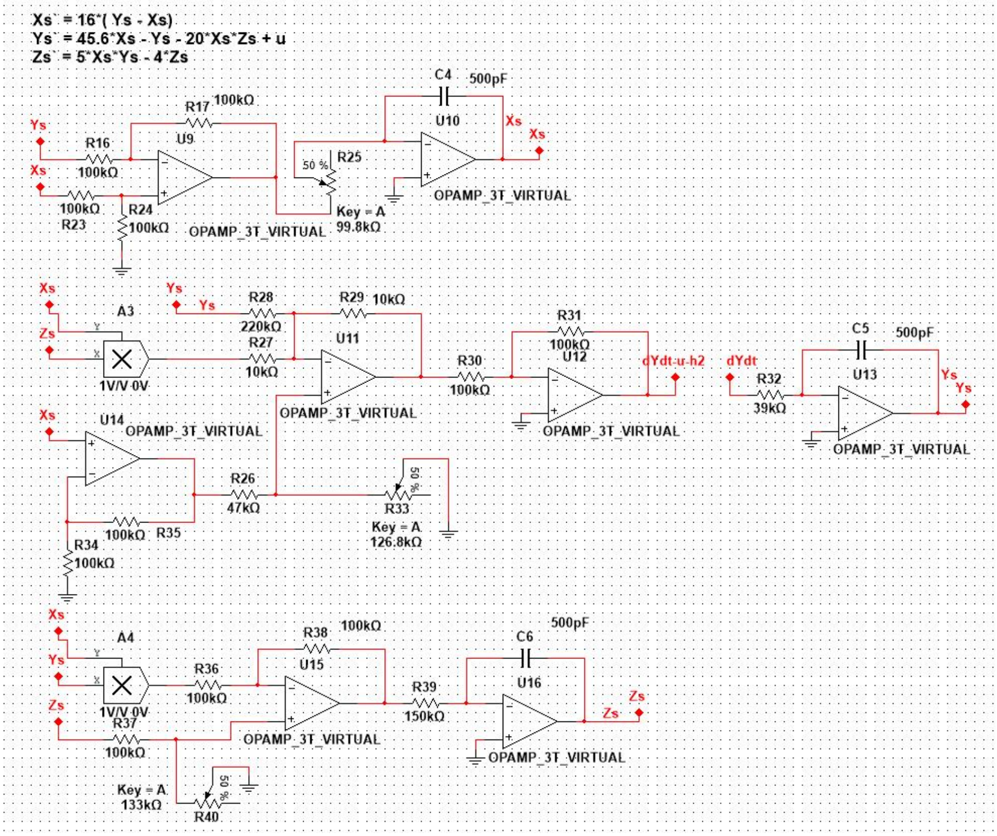

Figure 4.14 – Slave circuit.

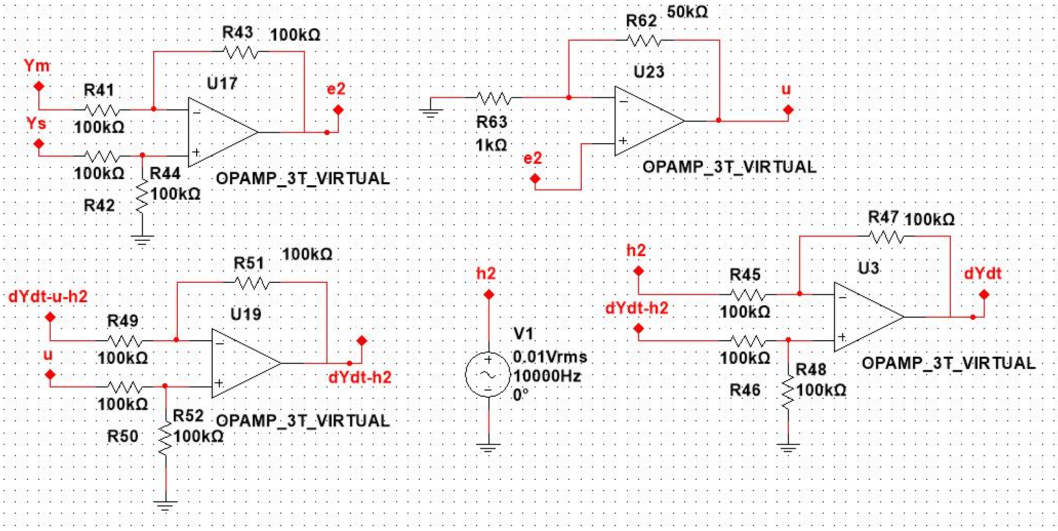

Figure 4.15 – Controller circuit with disturbances included.

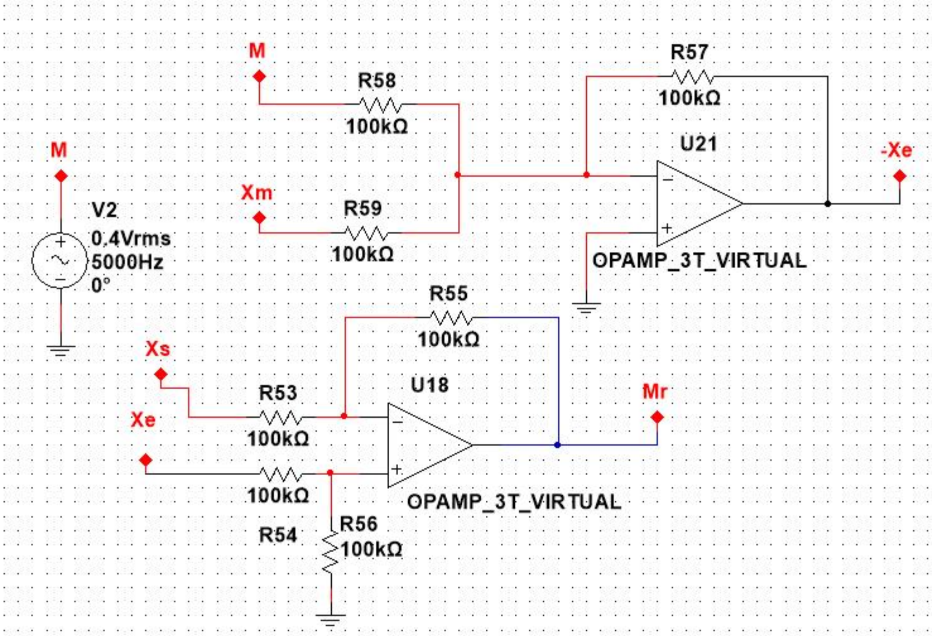

Figure 4.16 – Encode and decode circuits.

Finally, the obtained results from the simulation can be seen in Figures 4.17 - 4.19, where Figure 4.17 shows that the confidential message sent is a sine with amplitude $0.4V_{rms}$ and $5kHz$ of frequency, the Figure 4.18 shows the message encoded that appears to be a random signal and Figure 4.19 shows the retrieved message. In Figures 4.17 - 4.19, the scale is $500\mu s$ in the horizontal axis and 2V on the vertical axis. Some variation between the original message and the retrieved message; however, the decoded message is very similar to the original message, and the efficiency of the presented method can be seen. Note that the disturbances were also considered in the simulation with amplitude $0.01V_{rms}$ and frequency $10kHz$.

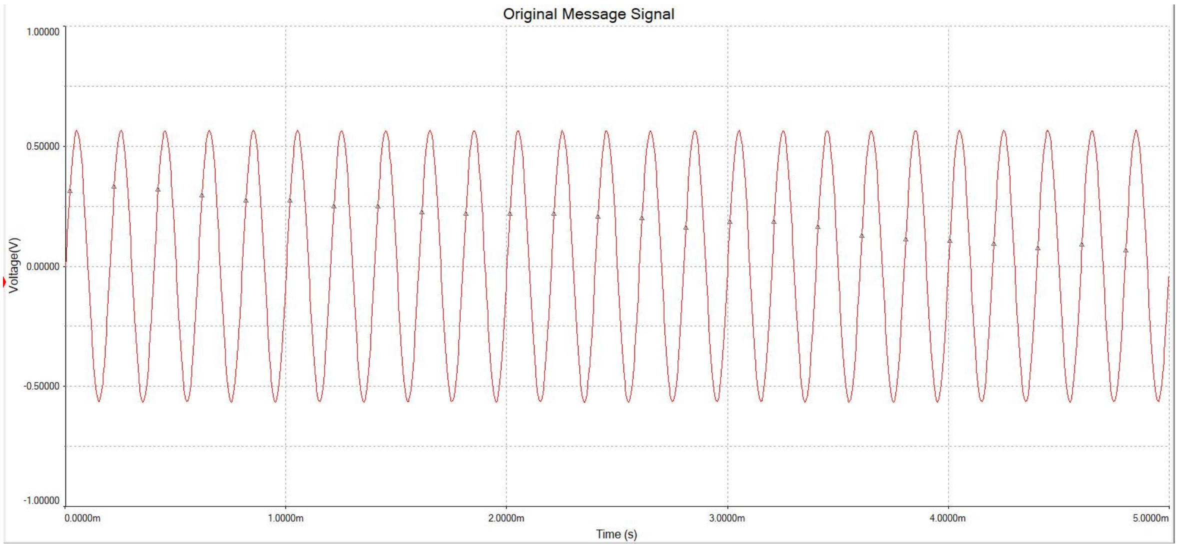

Figure 4.17 – Original message.

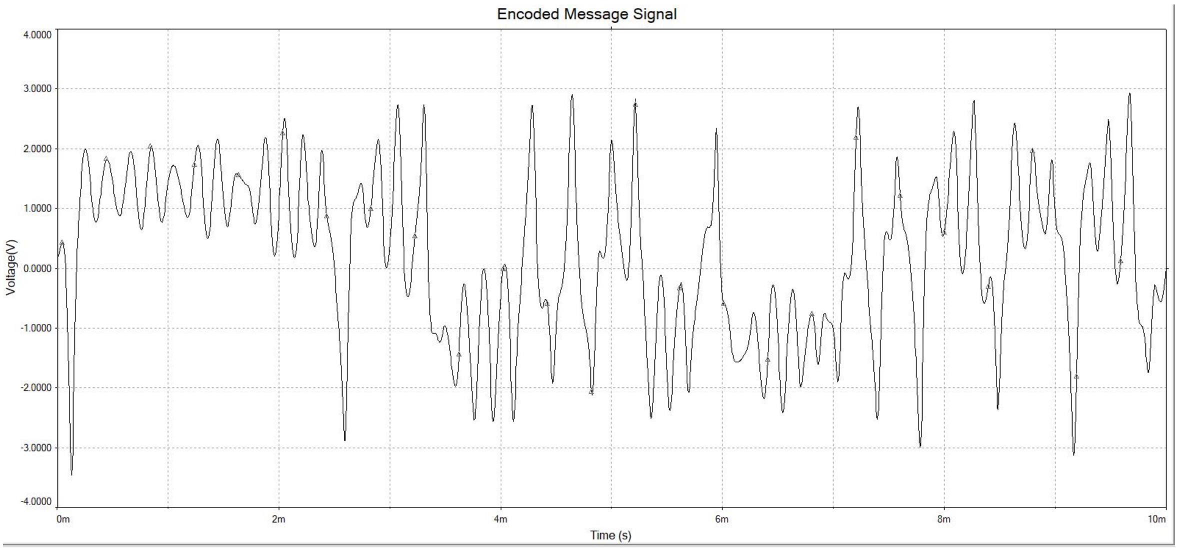

Figure 4.18 – Encoded message.

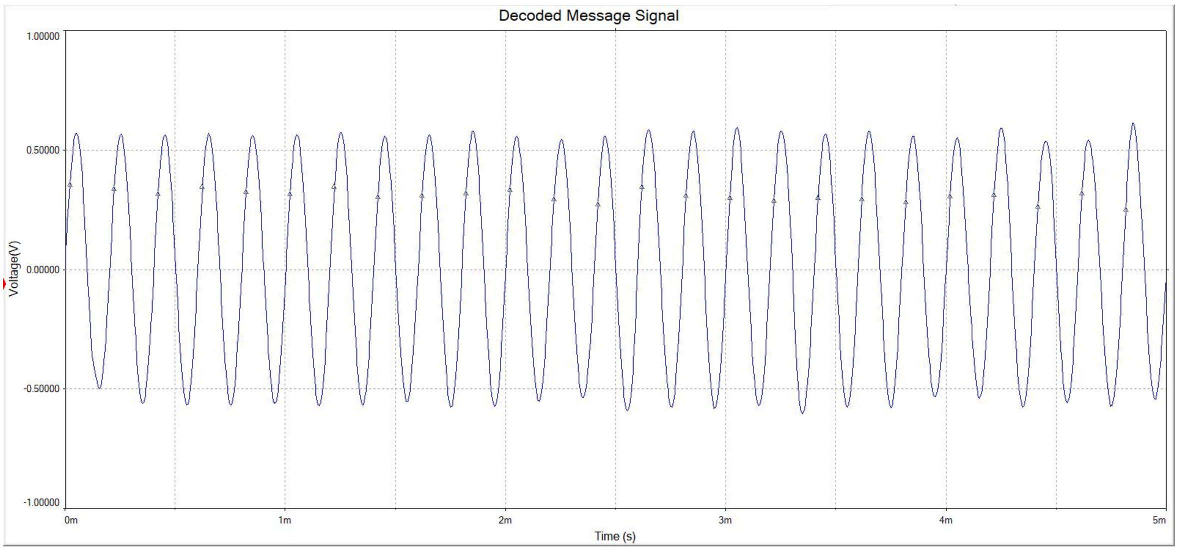

Figure 4.19 – Decoded message.

4.5 SUMMARY

This chapter proposes a synchronization algorithm based on Lyapunov stability theory for an underactuated Lorenz system and subject to bounded disturbances in all states. A signal control was proposed theoretically and simulated to synchronize the master and slave systems. The simulation results were consistent with the theoretical results.

5 A SYNCHRONIZATION SCHEME OF A 3D-CHAOTIC SYSTEM FOR INFORMATION SECURITY

The research outcomes of this chapter have been published as a conference paper entitled "A Chaotic Synchronization scheme for information security" in [11]. *This chapter has extended and improved some parts compared to the original paper.*

Chaotic systems can be found in many fields, such as in engineering, biology, mathematics, physics, among others. Applications of chaotic systems include communication, signal detection, cryptography, wireless sensors, laser system and so forth. [10, 61, 132, 140, 143, 146, 147]. Since the development of Pecora and Carroll's method [39], many types of synchronization have been developed, for example, anti-synchronization [147] and lag synchronization [148]. Its applications extend to other areas, such as biomedical engineering [149] and neural computing [74, 103].

The cryptography circuits found in the literature are based on the synchronization of two chaotic circuits. Chaotic signals are helpful for encoding and decoding signals since, by definition, the response of a chaotic system is complex, being unpredictable. Also, because chaotic systems are sensitive to initial conditions, it is possible that even knowing the structure of the chaotic system does not recognize state trajectories. In this way, chaotic systems are interesting to use in applications with encryption. The reason for disturbances occurring in real situations is due to tolerances and nonideal behaviors of electronic components, among other factors. In order to ensure excellent performance in practical applications, it is necessary to consider the possibility of disturbances in the theoretical analysis.

Although there are many methods of synchronization in the current literature, many of them present disadvantages, such as the need for a control signal in each state equation or, in many cases, the nonconsideration of disturbances [53, 140, 141]. Another recurrent problem in the literature is the relative complexity of the synchronizer. For example, in [133, 134] adaptive synchronizers are introduced. Therefore, the dimension of the synchronizer is larger than the dimension of the system. In [135], the dimension of the synchronizer is the same as that of the system. However, due to the use of Integrator backstepping, the synchronizer is structurally complex. In addition, the synchronizers proposed in [53, 140, 141] assume restrictive hypotheses, such as the equality of dimension between the system and the control [150]. Other works even consider the possibility of having bounded disturbances, but only in some state. For further details, see [143, 144].

It should be noted that, in order to achieve effective encrypting, there must be a superposition between the frequency spectrum of the message, and the signal-to-noise ratio of the message must be smaller than the signal-to-noise ratio of the employed state. If these conditions are disobeyed, the signal can still be appropriately recovered by a high-pass high order filter or by merely accepting the encrypted signal as the message signal, in case the deformation between them is negligible.

Motivated by the previous facts, this chapter presents a simple and robust scheme of synchronizing chaotic systems to overcome the mentioned deficiencies. The design is carried out in the context of Lyapunov stability theory to ensure boundedness and convergence. More specifically, the proposed synchronization scheme has the following characteristics.

1) To operate satisfactorily even in the presence of bounded disturbances that are present in all state equations.

2) For the use of a simple proportional control acting on only one state equation.

3) The synchronization error to be bounded and convergent to a small value that the user can adjust.

4) An application using electronic circuits to show the sophistication of the proposed scheme.

As far as the author is aware, no work prior to this chapter combined all these features simultaneously, which is the main contribution of this chapter.

The chapter is organized as follows. In Section 5.1, the problem and main assumptions are presented. The synchronization error, the control law, and Lyapunov analysis is introduced in Section 5.2. Section 5.3 is concerned with the secure telecommunication case. In Section 5.4, the development of a scaled system and the electronic circuit for the implementation of the proposed method is accomplished. Section 5.5 shown some simulations of the synchronization scheme. Section 5.6 presents the experimental implementation of electrical circuit synchronization. Finally, the conclusions of the chapter are made in Section 5.6.

5.1 PROBLEM STATEMENT

Consider the following chaotic system [151]:

$$\begin{cases} \dot{x}(t) = a(y(t) - x(t)) \\ \dot{y}(t) = cx(t) - y(t) - x(t)z(t) \\ \dot{z}(t) = x(t)y(t) - bz(t) \end{cases} \tag{5.1}$$

Based on the system (5.1), the master system is defined by

$$\begin{cases} \dot{x}_m &= a(y_m - x_m) \\ \dot{y}_m &= cx_m - y_m - x_m z_m \\ \dot{z}_m &= x_m y_m - bz_m \end{cases} \tag{5.2}$$

The slave system is defined by

$$\begin{cases} \dot{x}_s &= a(y_s - x_s) + h_1(t) \\ \dot{y}_s &= cx_s - y_s - x_s z_s + h_2(t) + u \\ \dot{z}_s &= x_s y_s - bz_s + h_3(t) \end{cases} \tag{5.3}$$

where x, y, and z are the states variables of the system (5.1), x_m, y_m, and z_m are the state variables of the master system, and x_s, y_s, and z_s are the state variables of the slave system. The systems parameters are $a = 2.1$, $b = 0.6$, and $c = 30$. The slave system disturbances are $h_1(t)$, $h_2(t)$, and $h_3(t)$. Note that u is the control signal.

This chapter proposes the synchronization of (5.2) and (5.3) by using a control signal only in the second state. Note that synchronization is achieved by any value of initial conditions, even in the presence of disturbances.

REMARK 5.1.1 The limitation of the system being underactuated is that the synchronizer only has access to information from the actuated master states, that is, from the states where there is a control signal. In other words, in the (5.2) system, the control u cannot have the presence of the states x_m, z_m, and w_m. Although, the other states of the master system and all the states of the slave system are available.

FACT 5.1.1 In [151] was proved that (5.1), and consequently (5.2), is chaotic and that the system is dissipative. The consequence of the system being dissipative is also being bounded. With the boundedness of the system (5.1), the following inequalities are true:

$$\begin{aligned} |x_m(t)| &\leq \bar{x} \\ |y_m(t)| &\leq \bar{y} \\ |z_m(t)| &\leq \bar{z} \end{aligned} \tag{5.4}$$

$\forall t \geq 0$, where $\bar{x}$, $\bar{y}$, and $\bar{z}$ are unknown positive constants.

REMARK 5.1.2 Since (5.1) is chaotic, so the system is dependent on its initial conditions. Therefore, a simple change in the values of the initial conditions has a considerable effect on the system behavior. Due to this, it is possible to make a secure telecommunication scheme in which the structure of the master system is similar to the structure of the slave system since it is enough to have different initial conditions so that the states have different behaviors.

ASSUMPTION 5.1.1 We assume that the disturbance is bounded. More specifically

$$\begin{aligned} |h_1(t)| &\leq \bar{h}_1 \\ |h_2(t)| &\leq \bar{h}_2 \\ |h_3(t)| &\leq \bar{h}_3 \end{aligned} \tag{5.5}$$

$\forall t \geq 0$, being $\bar{h}_1$, $\bar{h}_2$, and $\bar{h}_3$ unknown constants.

REMARK 5.1.3 The system (5.3) shows explicit disturbance, which is sometimes unusual in the literature. This allows us to evaluate the influence of these uncertainties over the convergence of the synchronization errors. It is generally interesting to consider the presence of disturbances better to analyze the robustness of the scheme in actual situations.

5.2 SYNCHRONIZATION ERROR AND PROPOSED SIGNAL CONTROL

Synchronization errors of the system are defined by

$$\begin{cases} e_1(t) = x_s(t) - x_m(t) \\ e_2(t) = y_s(t) - y_m(t) \\ e_3(t) = z_s(t) - z_m(t) \end{cases} \tag{5.6}$$

Therefore, dynamic equations of the errors can be obtained using systems (5.2) and (5.3) in the time-derivative of (5.6)

$$\begin{cases} \dot{e}_1 &= a(e_2 - e_1) + h_1 \\ \dot{e}_2 &= ce_1 - e_2 - e_1e_3 - e_1z_m - e_3x_m + h_2 + u \\ \dot{e}_3 &= e_1e_2 + e_1y_m + e_2x_m - be_3 + h_3 \end{cases} \tag{5.7}$$

Once defined the dynamics of errors, the main theorem of this chapter is presented.

THEOREM 5.2.1 Consider the master and slave system described in (5.2) and (5.3) and the proportional control law defined by:

$$u = -\psi e_2 \tag{5.8}$$

If,

$$\psi > \delta \tag{5.9}$$

Then, the synchronization error is uniformly ultimately bounded and converges in finite-time to the compact set

$$\Omega = \left\{ e \in \Re^3 \mid ||e|| \leq \theta \right\} \tag{5.10}$$

where $\delta = 0.5\sigma_2\bar{h}_2^2 + 0.5\sigma_4^{-1}(a^2\gamma^2 + c^2 + \bar{z}^2) - 1$, $\theta = \sqrt{\frac{\beta}{\rho}}$, $\beta = \beta_u + \beta_n$, $\rho = min\{\rho_1, \rho_2, \rho_3\}$, $\beta_u = 0.5\sigma_2^{-1}$, $\beta_n = 0.5(\gamma\sigma_1^{-1}\bar{h}_1^2 + \sigma_3^{-1}\bar{h}_3^2)$, $\rho_1 = a\gamma - 0.5(\gamma\sigma_1 + \sigma_4 + \sigma_5\bar{y}^2)$, $\rho_2 = \psi - \delta$, $\rho_3 = b - 0.5\sigma_3 - 0.5\sigma_5^{-1}$, $\|e\|^2 = e_1^2 + e_2^2 + e_3^2$, and $\sigma_i, i = 1, ..., 5$ are positive constants.

Proof.

Consider the following Lyapunov function candidate

$$V = \frac{1}{2}(\gamma e_1^2 + e_2^2 + e_3^2) \tag{5.11}$$

where $\gamma > 0$. The time-derivative of (5.11) along the trajectories of (5.7), results

$$\begin{aligned} \dot{V} &= \gamma e_1[a(e_2 - e_1) + h_1] \\ &\quad + e_2(ce_1 - e_2 - e_1e_3 - e_1z_m - e_3x_m + h_2 + u) \\ &\quad + e_3(e_1e_2 + e_1y_m + e_2x_m - be_3 + h_3) \end{aligned} \tag{5.12}$$

Replacing (5.8) in (5.12), impies

$$\begin{aligned} \dot{V} &= -a\gamma e_1^2 - e_2^2(\psi + 1) - be_3^2 + \gamma e_1h_1 + e_2h_2 + e_3h_3 \\ &\quad + e_1e_3y_m + e_1e_2(a\gamma + c - z_m) \end{aligned} \tag{5.13}$$

Thus, analyzing when $\dot{V} \leq 0$, by using inequality of Young, the Assumption 5.1.1, and the Fact 5.1.1, then

$$\begin{aligned} &\gamma e_1h_1 \leq 0.5\gamma(\sigma_1e_1^2 + \sigma_1^{-1}\bar{h}_1^2) \\ &e_2h_2 \leq 0.5(\sigma_2\bar{h}_2^2e_2^2 + \sigma_2^{-1}) \\ &e_3h_3 \leq 0.5(\sigma_3e_3^2 + \sigma_3^{-1}\bar{h}_3^2) \\ &e_1[e_2(a\gamma + c - z_m)] \leq 0.5[\sigma_4e_1^2 + \sigma_4^{-1}e_2^2(a^2\gamma^2 + c^2 + \bar{z}^2)] \\ &e_1e_3y_m \leq 0.5(\sigma_5e_1^2\bar{y}^2 + \sigma_5^{-1}e_3^2) \end{aligned} \tag{5.14}$$

Replacing (5.14) in (5.13), results

$$\dot{V} \leq -e_1^2\rho_1 - e_2^2\rho_2 - e_3^2\rho_3 + \beta_u + \beta_n \tag{5.15}$$

Observe that: 1) there are values of γ and $\sigma_i, i = 1, ..., 5$ that cause $\rho_1 > 0$ and $\rho_3 > 0$; and 2) ψ is chosen by the designer so that (5.9) is satisfied, and, consequently, $\rho_2 > 0$. Therefore, there is a $\rho > 0$, and (5.15) can be rewritten as

$$\dot{V} \leq -\rho \left\|e\right\|^2 + \beta \tag{5.16}$$

Based on (5.16), $\dot{V} < 0$ when $\|e\| > \theta$. Since θ is a constant, it can be established that the synchronization error is uniformly ultimately bounded (for further details, see [24]). If under any circumstance $\|e\|$ leaves the residual set Ω, $\dot{V}$ becomes negative definite and forces the convergence of the synchronization error to the residual set Ω. The convergence to the residual set Ω is in finite-time, due to the particular form of 5.16 [24]. Therefore, the synchronization error is uniformly ultimately bounded and converge to a sphere with radius θ (Figure 5.1). □

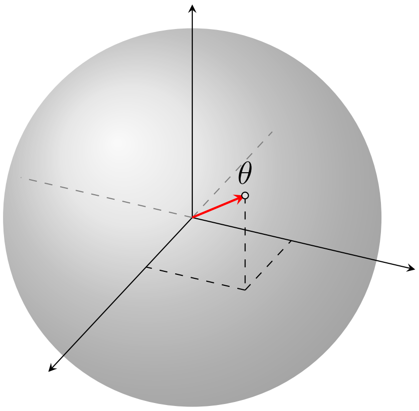

Figure 5.1 – 3D-Bounded set.

REMARK 5.2.1 The proposed method lies in the usage of the system structure and Lyapunov theory to design an valid control law. Based on a trial-and-error procedure, all possibilities of underactuated control in the analysis was considered and the simplest one was chosen. Inequality of Young was used in the stability analysis in the process to make $\dot{V} < 0$ outside of a small compact set.

REMARK 5.2.2 Defining $D_i(i = 1, ..., 5)$ as the respective domains of $\sigma_i(i = 1, ..., 5)$, results

$$
\begin{aligned}
D_1 &= \{\sigma_1 \in \Re | 0 < \sigma_1 < 2a\} \\
D_2 &= \left\{\sigma_2 \in \Re | 0 < \sigma_2 < frac2(1+\psi)\bar{h}_2^2\right\} \\
D_3 &= \{\sigma_3 \in \Re | 0 < \sigma_3 < 2b\} \\
D_4 &= \left\{\sigma_4 \in \Re | \frac{a^2\gamma^2 + c^2 + \bar{z}^2}{2(1+\psi)} < \sigma_4 < 2a\gamma\right\} \\
D_5 &= \left\{\sigma_5 \in \Re | \frac{1}{2b} < \sigma_5 < \frac{2a\gamma}{\bar{y}^2}\right\}
\end{aligned}
\tag{5.17}
$$

The residual synchronization error considered is affected by the control gain ψ, disturbances, and upper bounds for the states of the master system, as can be seen from (5.9). The performance for the actuated states can be arbitrarily enhanced by increasing ψ. For non-actuated states, it can not be guaranteed that a change in the gain of the control will cause the residual synchronization error to decrease, being this the main limitation of underactuated synchronization systems.

5.3 CHAOS-BASED SECURE COMMUNICATION

In addition to the synchronization case, the application of the proposed method to secure telecommunication are also considered. To have a well-posed problem, the following assumption must be imposed.

ASSUMPTION 5.3.1 It is assumed that the messages are bounded. More specifically,

$$|m_i(t)| \leq \bar{m}_i, i = 1, ..., 3 \tag{5.18}$$

$\forall t \geq 0$, where m_1, m_2, and m_3 are the original messages and $\bar{m}_1$, $\bar{m}_2$, and $\bar{m}_3$ are positive constants.

Motivated by [7], it can be defined

$$
\begin{aligned}
\hat{m}_1 &= s_1 - x_s \\
\hat{m}_2 &= s_2 - y_s \\
\hat{m}_3 &= s_3 - z_s
\end{aligned}
\tag{5.19}
$$

where $s_1 = x_m + m_1$, $s_2 = y_m + m_2$, and $s_3 = z_m + m_3$ are the encrypted messages; and $\hat{m}_1$, $\hat{m}_2$, and $\hat{m}_3$ are the decrypted messages.

On the other hand, by using (5.19) and defining $\tilde{m}_i = \hat{m}_i - \delta m_i, i = 1, ..., 3$, where $\tilde{m}_1$, $\tilde{m}_2$, and $\tilde{m}_3$ are the message errors, it can be concluded that

$$\tilde{m}_i = -e_i, i = 1, ..., 3 \tag{5.20}$$

REMARK 5.3.1 Observe that the quality of the message reconstruction is the same as the synchronization, as shown in (5.20). Also, the boundedness of the message error is assured when the synchronization error is bounded.

REMARK 5.3.2 An overview of the secure communication scheme can be seen in Figure 5.2.

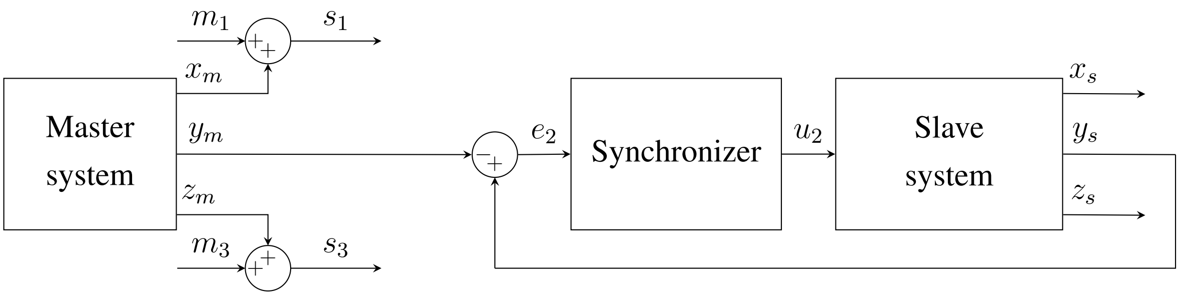

Figure 5.2 – Synchronization and secure communication scheme.

5.4 SCALING

It was found that the solution to these equations had a very high amplitude. Therefore, the amplitude was scaled down by a factor of 20. More exactly,

$$\begin{cases} \dot{x} = 2.1(y - x) \\ \dot{y} = 30x - y - 20xz \\ \dot{z} = 20xy - 0.6z \end{cases} \tag{5.21}$$

Following the scaling, the system was simulated in Matlab/Simulink, including the slave, the synchronizer, and the message. The circuit was then projected so that the slave and the master were as similar as possible, being both composed of 10 blocks, of which 2 were multipliers (xy and xz), 1 was a synchronizer, 3 were integrators, and the remaining 3 were adders. The schematic for this arrangement can be found in Figure 5.3.

In Figure 5.3, the switches identify the circuit as the master or the slave. The position of the switches indicates that the schematic above represents the slave. For the master, $k(t) = m(t)$, $p(t) = -s(t)$, $w(t) = y_s(t)$, and $f(t) = \frac{-u(t)}{10}$; for the slave, $k(t) = -s(t)$, $p(t) = \hat{m}(t)$, $w(t) = \frac{-u(t)}{10}$, and $f(t) = u(t) + y_s(t)$. The resistors R1, R4, R6, R11, R12, R13, R14, R15, R26, and R27 have value 10kΩ; the resistors R2, R3, R5, R21, R22, R23, R24, and R25 have value 100 kΩ; the resistors R7 and R8 have value 4.75kΩ; the resistors R9, R10, and

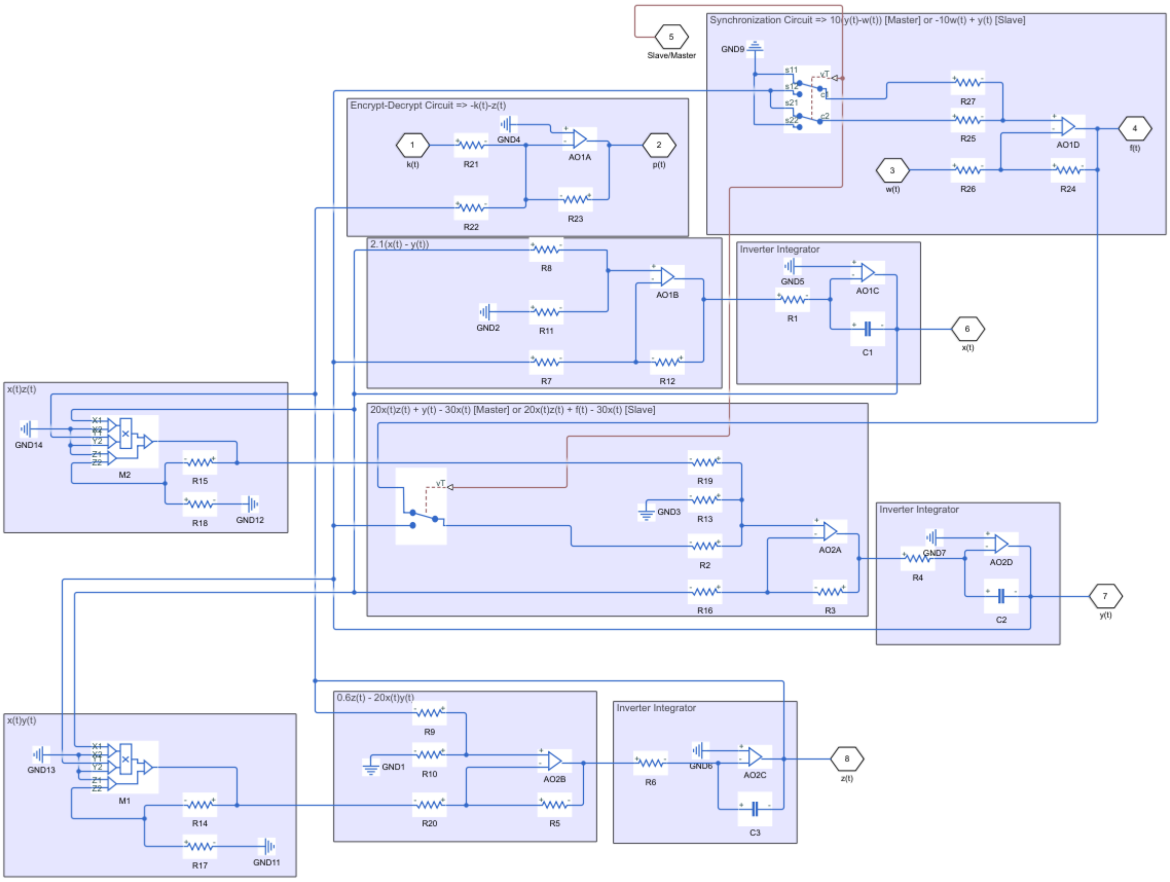

Figure 5.3 – Circuit of master and slave systems.

R16 have value 165kΩ, 4.87kΩ, and 3.32kΩ respectively; R17 and R18 measure 90.9kΩ; R19 and R20, 5.11kΩ. The capacitors have a value of 100 nF; the operational amplifies are LF347 and multipliers are AD33JNZ. The capacitors and resistors have a tolerance of 5% and 1% respectively.

It was noted that the equations from system 5.21 would result in the necessity of unity gain in the integrators or tiny gain in the adders, leading to very large or very small resistance values. Therefore, the system was again scaled in frequency by a factor of 1000, resulting in the system of equations in what follows.

$$\begin{cases} \dot{x}(t) = 1000 \cdot 2.1(y(t) - x(t)) \\ \dot{y}(t) = 1000(30x(t) - y(t) - 20x(t)z(t)) \\ \dot{z}(t) = 1000(20x(t)y(t) - 0.6z(t)) \end{cases} \tag{5.22}$$

The systems of equations (5.23) and (5.24) define, respectively, the master and the slave. $m(t)$ is the message signal, encrypted message is $s(t) = m(t) + z_m(t)$, the recovered signal in the slave is $\hat{m}(t) = s(t) - z_s(t)$, the control signal is $u(t) = -100 \cdot e_y(t)$, and the synchronization error of the y state is $e_y(t) = y_s(t) - y_m(t)$. More precisely,

$$\begin{cases} \dot{x}_m = 1000 \cdot 2.1(y_m - x_m) \\ \dot{y}_m = 1000(30x_m - y_m - 20x_m z_m) \\ \dot{z}_m = 1000(20x_m y_m - 0.6z_m) \end{cases} \tag{5.23}$$

$$\begin{cases} \dot{x}_s = 1000 \cdot 2.1(y_s - x_s) \\ \dot{y}_s = 1000(30x_s - y_s - 20x_s z_s + u) \\ \dot{z}_s = 1000(20x_s y_s - 0.6z_s) \end{cases} \tag{5.24}$$

The 1000 gain that resulted from the frequency scaling was wholly allocated in the integrators. After adopting 100 nF as the valor for the capacitors, it was found that the resistors $R1$, $R4$, and $R6$ should have the following value of resistances: $R1 = R4 = R6 = \frac{1}{1000 \cdot 100 \cdot 10^{-9}} = 10\text{k}\Omega$.

In order to facilitate the implementation and reduce the differences between the master and the slave, the encrypted signal was given by $-s(t)$, that is to say, the encrypting block makes $-m(t) - z_m(t)$ and the decoder block makes $s(t) - z_s(t) = \hat{m}(t)$. Thus, the encoder and decoder blocks are identical, and both are equivalent to an inverting adder without gain. Arbitrarily, it was decided that $R23 = 10\text{k}\Omega$ and, as a result of the absence of gain, $R21 = R22 = R23$. Since the integrator inverts its input signal, the operations in the adders with states must also be reverted. That is to say; summation should be substituted by subtraction and vice versa. Consider $R12 = 10\text{k}\Omega$ and $R3 = R5 = 100\text{k}\Omega$.

For the X state adder, $R8 = R7 = \frac{10000}{2.1} = 4.7619\text{k}\Omega$. For the implementation, the value of 4.75kΩ was used, since that was the nearest resistance value found for a precision resistor. Consider $Z = 2.1 - 2.1 - 1 = -1$. Therefore, in this adder, it is necessary to connect a resistor between the non-inverting input of the operational amplifier and ground, with value $R11 = \frac{10000}{-Z} = 10\text{k}\Omega$. For the Y state adder, $R19 = \frac{100000}{20} = 5\text{k}\Omega$. However, the nearest value commercially available was 5.11kΩ, which was adopted in the implementation. Also, $R2 = \frac{100000}{1} = 100\text{k}\Omega$ and $R16 = \frac{100000}{30} = 3.3333\text{k}\Omega$. For the latter case, the value of 3.321lΩ was used instead due to restrictions in the value of commercially available resistors. Finally, $Z = 20 + 1 - 30 - 1 = -10$, so that a resistor is necessary to connect the non-inverting input of the operational amplifier to ground, with value $R13 = \frac{100000}{10} = 10\text{k}\Omega$. For the Z state adders, $R9 = \frac{100000}{0.6} = 166.6666\text{k}\Omega$. The nearest commercially available value was 165kΩ. Also, $R20 = \frac{100000}{20} = 5\text{k}\Omega$. As in the Y state adder, the adopted value was 5.11kΩ. Finally, due to $Z = 0.6 - 20 - 1 = -20.4$, a resistor must connect the non-inverting input of the op-amp to ground, with value $R10 = \frac{100000}{20.4} = 4.902\text{k}\Omega$. The value used for in the implementation was 4.87kΩ.

The multiplication blocks were built by a simple connection of the AD633JNZ multiplier and a proper association of resistors. This configuration results in $\frac{(X1-X2)(Y1-Y2)}{10}\frac{(Ra+Rb)}{Ra}$. Consider $X2 = Y2 = 0$, $Ra = R14 = R15 = 10\text{k}\Omega$, and $Rb = R17 = R18 = 90\text{k}\Omega$, then $\frac{1}{10}$ multiplying factor can be eliminated and the output of the block is simply $X1 \cdot Y1$. Nonetheless, a precision resistor of 90kΩ could not be found, so 90.9kΩ resistors were employed instead.

For the synchronizers, the desired operations are $-10e_y(t) = 10(y_m(t) - y_s(t))$ and $f(t) = -10 \cdot (-10e_y(t)) + y_s(t) = 100e_y(t) + y_s(t) = -u(t) + y_s(t)$ for the master and the slave respectively. Here, notice that the first difference between the master and slave circuits, which, in Figure 5.3, is indicated by the position of the switches. The switch in the Y state adder selects between $y(t)$ in the master and $f(t)$ in the slave. In the master, the differential equation assumes the form $\dot{y}(t) = 1000(30x(t) - y(t) - 20x(t)z(t))$, while in the slaves it is given by $\dot{y}(t) = 1000(30x(t) - f(t) - 20x(t)z(t)) = 1000(30x(t) - y(t) + u(t) - 20x(t)z(t))$. In order to implement the equations of the synchronization blocks in the master, consider $R24 = 100\text{k}\Omega$, $R26 = R27 = \frac{100000}{10} = 10\text{k}\Omega$, and $Z = 10 - 10 - 1 = -1$. The non-inverting input can then be connected to ground by a resistor $R25 = 100\text{k}\Omega$. For the slave, it suffices to invert resistors R25 and R27. This inversion is represented by changing the position of the switch. In Figure 5.3, the switches are positioned as in the necessary configuration for the slave.

5.5 SIMULATION

The simulations done in Matlab/Simulink 2020b® were made in a Windows 10 platform, with AMD Ryzen 7 1700 processor for all simulations. Simulations corresponding to Figures 5.4 - 5.13 and Table 5.1 used variable-step algorithm ODE45 solver, and relative and absolute tolerance of 10^{-10}. Simulations corresponding to Figures 5.14 - 5.17 and Table 5.2 used variable-step algorithm ODE23t solver, and relative and absolute tolerance of 10^{-8}.

The initial conditions for the master system were $x_m(0) = [0.1; -0.1; 0.1]$, and for the slave system were $x_s(0) = [0.2; 0.1; 0.2]$. The control law (5.8) was used to synchronize the master and slave systems, with $\psi = 100$. Disturbances were considered, with $h_1 = 0.3sin(2t) + 0.1sin(20t)$, $h_2 = 0.4sin(t) + 0.1cos(2t)$, and $h_3 = 0.3square(t) + 0.2sin(4t)$.

Figures 5.4 - 5.9 show the synchronization of the states X, Y, and Z of the master and the slave in terms of differential equations. Figures 5.7 - 5.9 present the synchronization errors. Note that the synchronization result was good, with synchronization errors close to zero even with only one control signal and the presence of disturbances.

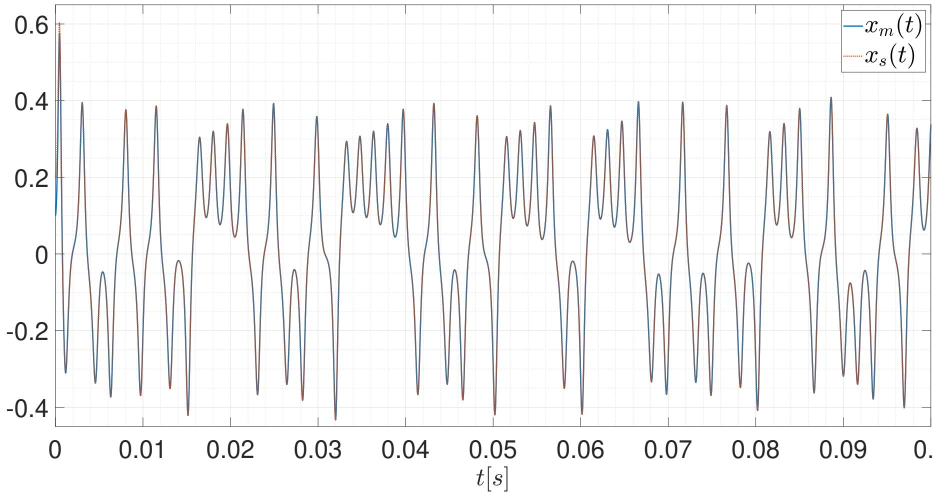

Figure 5.4 – Synchronization of the first state in master and slave systems.

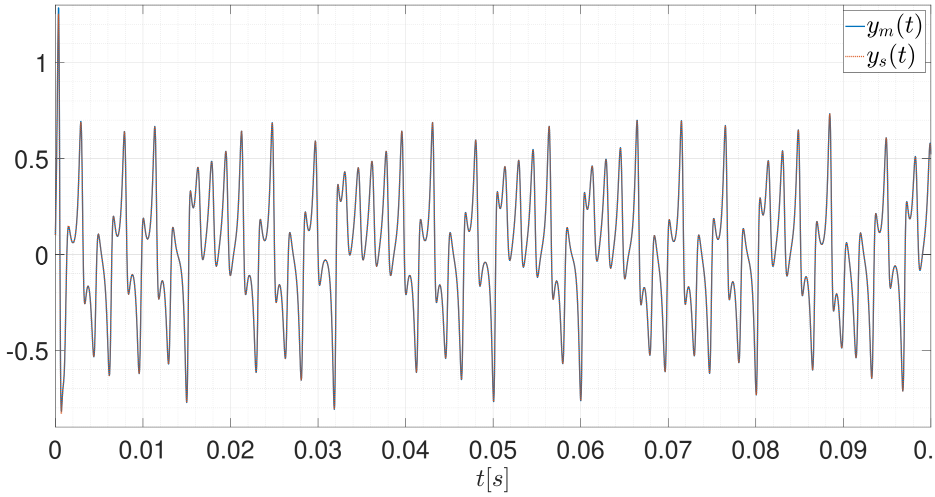

Figure 5.5 – Synchronization of the second state in master and slave systems.

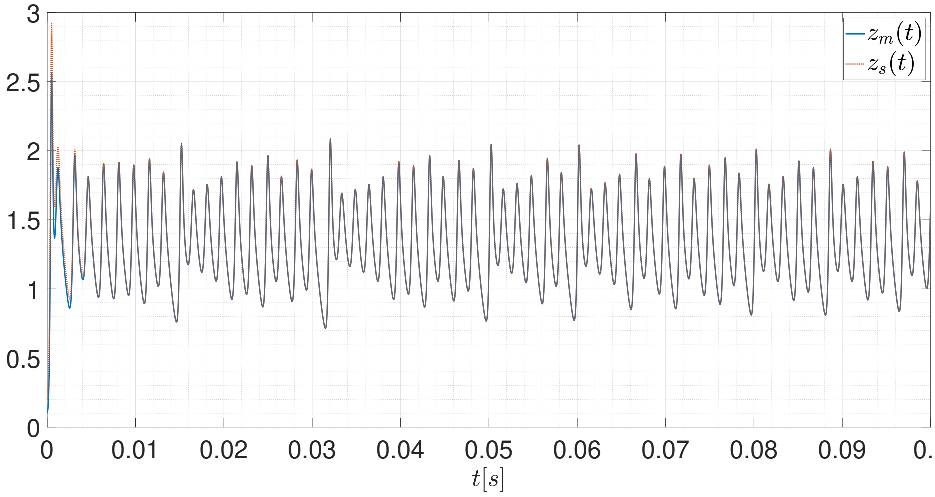

Figure 5.6 – Synchronization of the third state in master and slave systems.

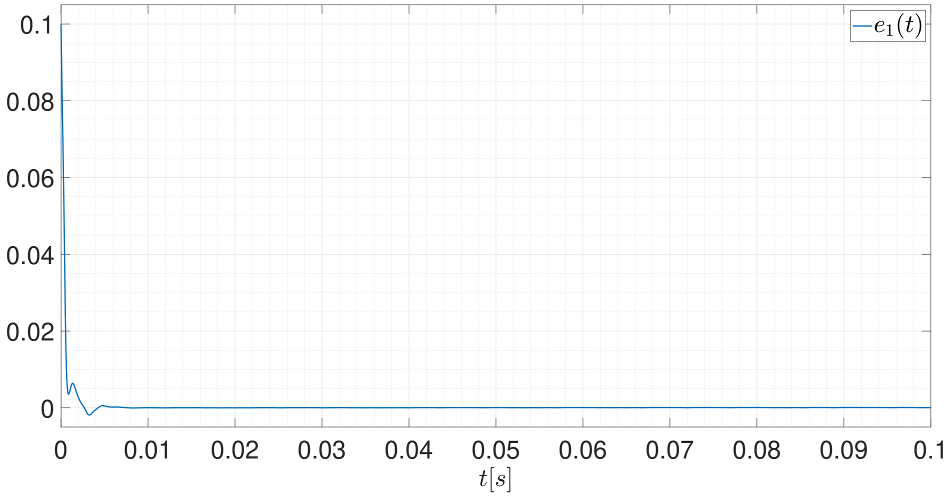

Figure 5.7 – Synchronization error of the first state.

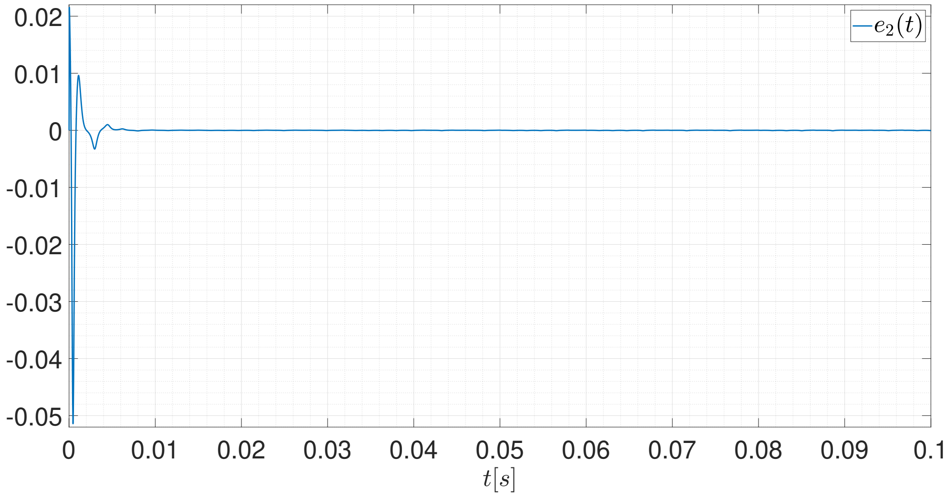

Figure 5.8 – Synchronization error of the second state.

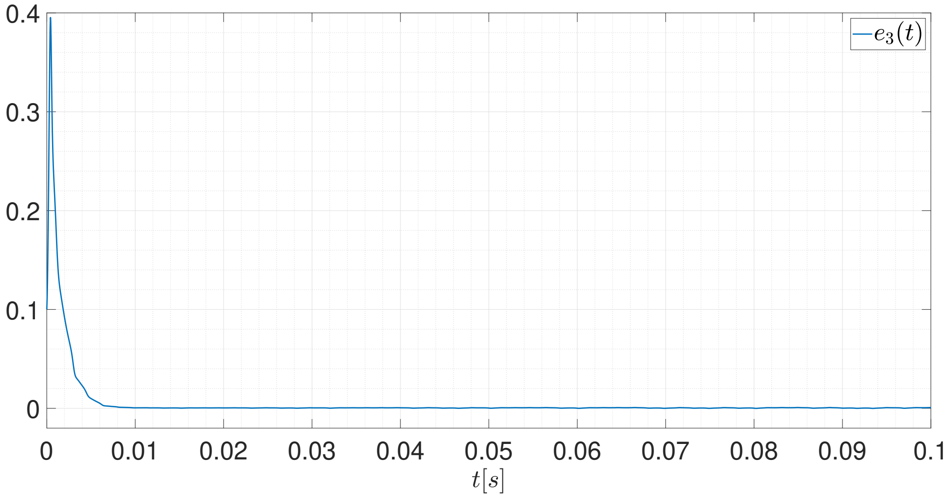

Figure 5.9 – Synchronization error of the third state.

Table 5.1 shows that the optimal control gain value is approximately $\psi = 10$. Smaller values of ψ lead to larger values of β_u and smaller values of ρ_2, consequently, lead to greater synchronization error values. Theoretically, larger values of ψ should not increase the synchronization error value, but higher control gain can lead to greater approximation errors in computer simulations.

Table 5.1 – Root mean square of state errors for t = [0 0.1] seconds and consider that $\psi = 100$.

Root Mean Square of State Errors in the Proposed Algorithm				
	$e_{1_{rms}}$	$e_{2_{rms}}$	$e_{3_{rms}}$	e_{rms}
0.001ψ	0.330024	0.462829	0.419797	0.530140
0.01ψ	0.254716	0.364291	0.426320	0.445616
0.1ψ	0.009402	0.024971	0.038773	0.041080
ψ	0.009720	0.004866	0.054131	0.054499
10ψ	0.008243	0.000489	0.056497	0.056771
100ψ	0.005060	0.000031	0.036659	0.036819

Figures 5.10 - 5.13 present the communication secure case. Messages signals $m_1(t)$ and $m_3(t)$ are introduced, as well as the encrypted signal $s(t)$ and the recovered signal. m_1 is a square wave of frequency 159 Hz and amplitude 0.02 V and m_3 is a sawtooth wave of frequency 159 Hz and amplitude 0.02 V. Observe that the message error is very small, and the encoded message is different from the original message.

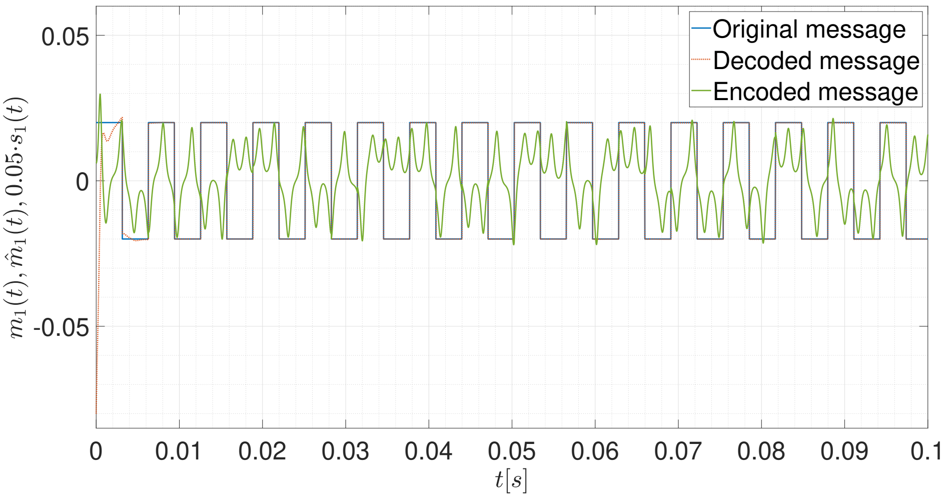

Figure 5.10 – Transmitted message($m(t)$), encrypted message $s(t)$, and recovered message $\hat{m}(t)$ in the first state.

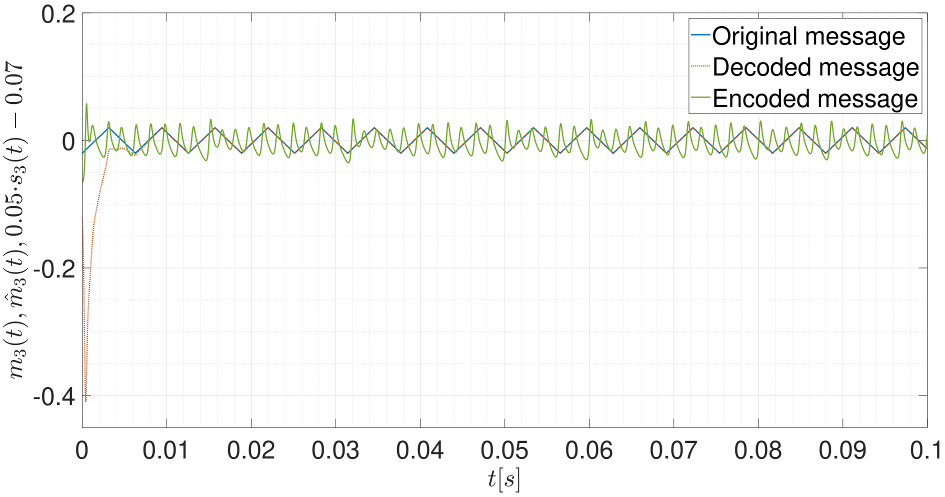

Figure 5.11 – Transmitted message($m(t)$), encrypted message $s(t)$, and recovered message $\hat{m}(t)$ in the third state.

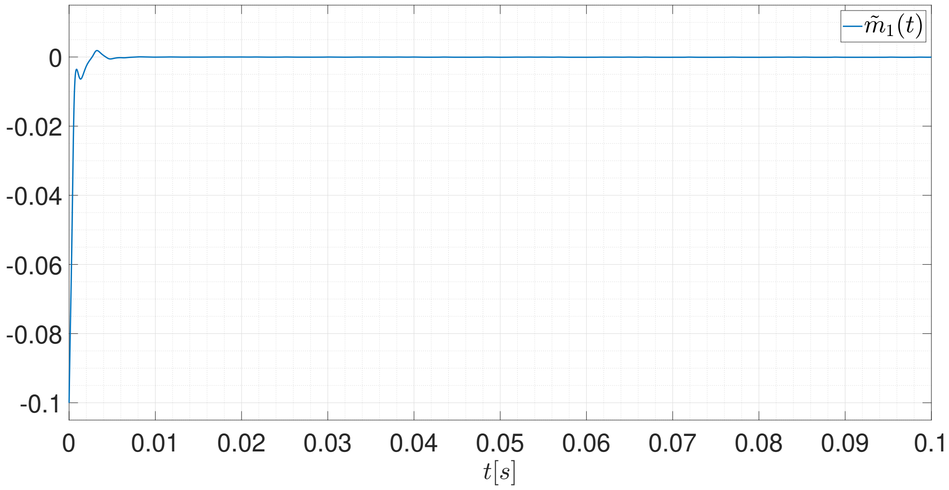

Figure 5.12 – Message error 1.

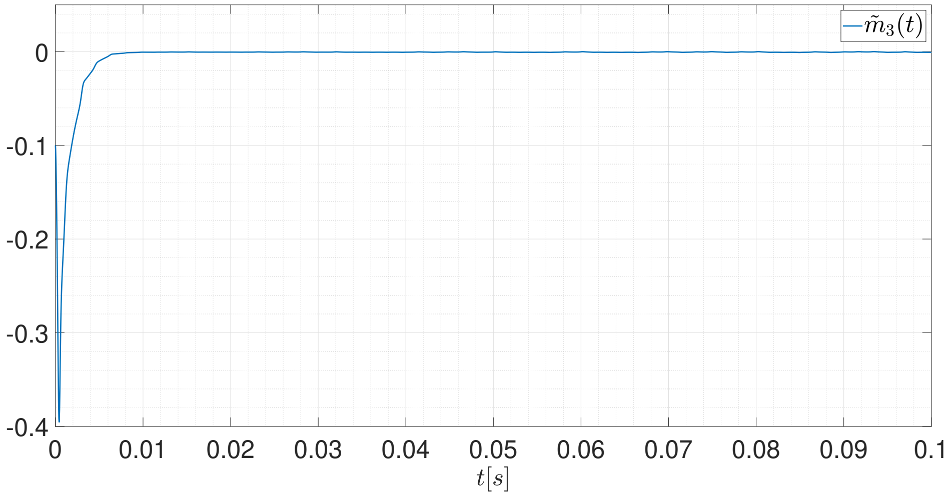

Figure 5.13 – Message error 3.

The complete simulated circuit is not included because it is the circuit presented in Figure 5.3, but duplicated with the switches properly positioned (as in section 5.3), ideal components substituted by real ones, power supplies of +15V and -15V for the integrated circuits, voltage sensors and a square wave generator that produces the message signal.

Figure 5.14 shows the state Z of the master and the slave for the circuit simulation. The initial conditions differed between the simulations. Figure 5.15 shows the synchronization errors. It should be noted that, unlike the simulation in Simulink, the states did not synchronize perfectly during the circuit simulation. This is due to three main factors: noises, the added imperfections in the multipliers and operational amplifiers, and the tolerance of the resistors and capacitors. The simulation would have failed if resistors with 5% of tolerance had been applied. As a result, resistors with tolerance rated as 1% were chosen instead.

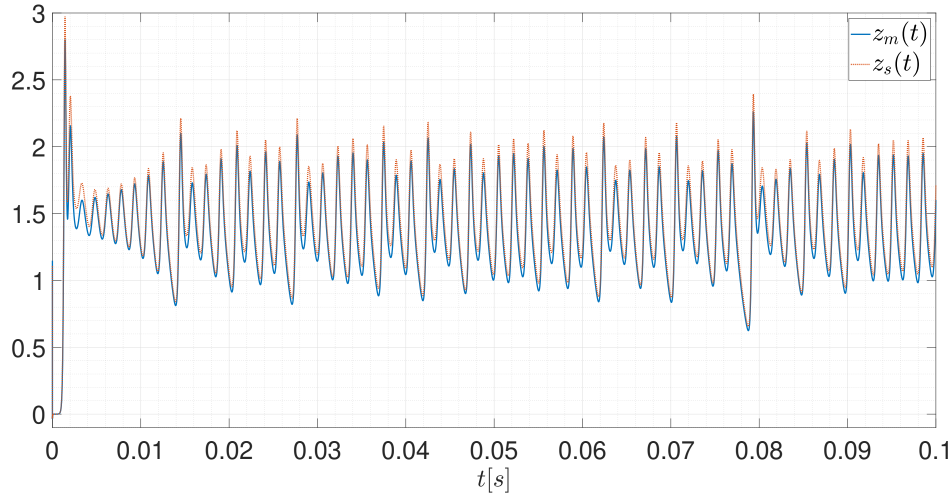

Figure 5.14 – Synchronization of the third state in the circuit simulation.

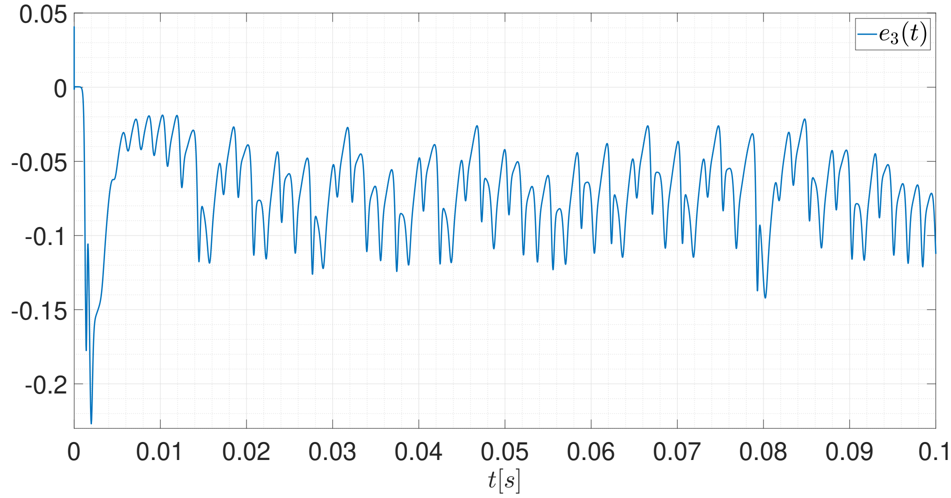

Figure 5.15 – Synchronization error of the third state in the circuit simulation.

Figure 5.16 - 5.17 show the communication secure case. The signal $m(t)$, which, as described above, is a square wave of amplitude 0.4 V and frequency 100 Hz, along with the

encrypted signal $-s(t)$ and the decoded signal $\hat{m}(t)$. Since the states do not synchronize perfectly, the message could not be completely recovered. However, the result is satisfactory since an inevitable synchronization error will be due to the actual components and their inherent disturbances.

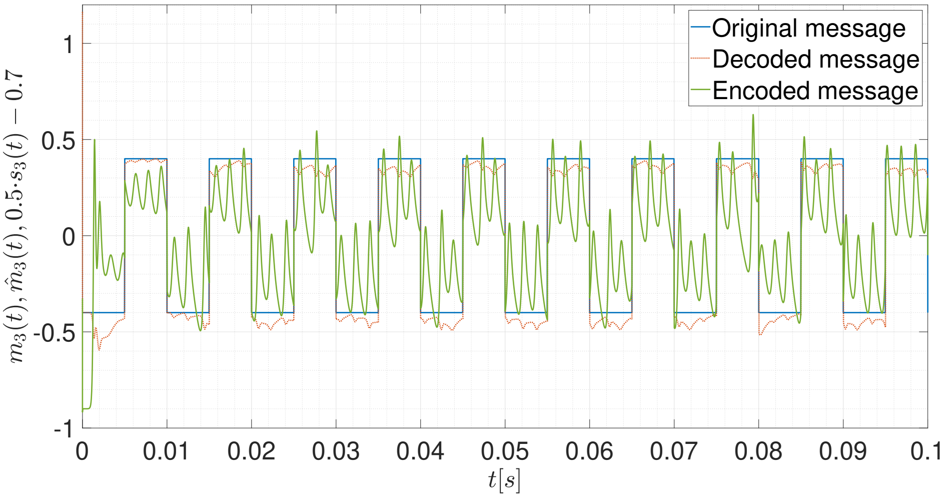

Figure 5.16 – Transmitted message($m(t)$), Encrypted message $s(t)$, and recovered message $\hat{m}(t)$ in the third state in the circuit simulation.

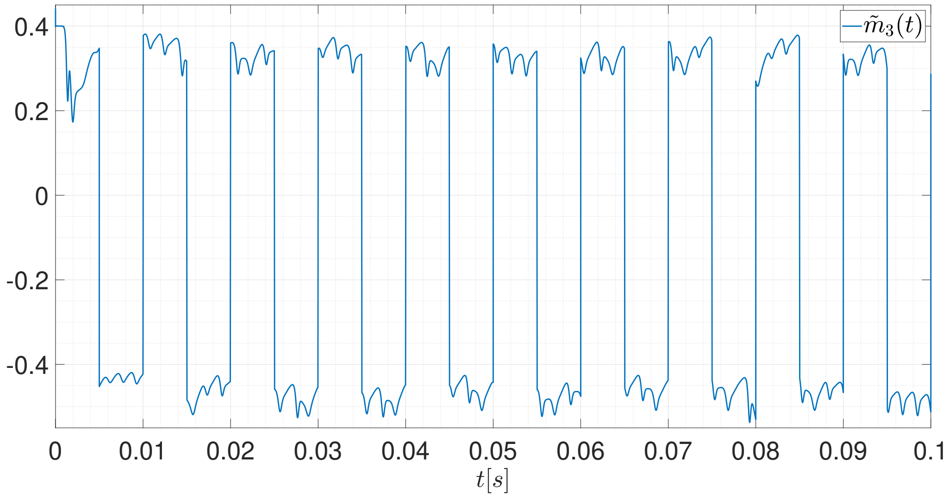

Figure 5.17 – Message error 3 in the circuit simulation.

Table 5.2 – Root mean square of state errors for t = [0 0.1] seconds.

Root Mean Square of State Errors in the Proposed Algorithm			
$e_{1_{rms}}$	$e_{2_{rms}}$	$e_{3_{rms}}$	e_{rms}
0.009312	0.014079	0.078171	0.078179

5.6 EXPERIMENT

The circuit in Figure 5.3 was initially built on a breadboard. However, it was found that the recovered signal contained an intense noise, preventing the proper operation of the circuit. Due to this complication, it was decided that the circuit should be built on a printed circuited board. Fortunately, all of the previous problems were eliminated in the new version. In Figure 5.18 the red marked part is the one that gave origin to the printed board in Figure 5.19. Based in Figure 5.18, Tables 5.3, 5.4, and 5.5 were manually done, that contain how each element of the circuit, Multiplier, Operational Amplifier, Resistor, Capacitor, and Jumpers interconnects, considering the lines with these names. In the Ultiboard, the components were inserted, and then these tables were loaded with that the Ultiboard program generated the printed board.

Figure 5.19 shows the schematic for the printed circuit generated by the software Ultiboard by means of the netlist Tables 5.3, 5.4, and 5.5. In Figure 5.20 is presented the picture of the built circuit. Notably, jumpers were added to the interface of the blocks in 5.3 to isolate them for individual tests. The element P1 in the schematic is external connection pins. From 1 to 10, the pins are positive supplying, ground, negative supplying, state X, state Y, state Z, message signal (master) or encrypted signal (slave), encrypted signal (master) or recovered signal (slave), state Y of the slave (master) or control signal (master), a control signal (master) or unused (slave).

The board contains some jumper wires whose function is not restricted to isolating the appropriated blocks but are also used to determine the nature of the circuit (master or slave) and the used state. Therefore, except for the jumper position, the slave and master boards are identical, and it is possible to choose which states will be employed in each. If jumper J1 is closed, the used state is X; if J2 is closed, the state is Z. For this chapter, the used state was Z, for which the synchronization error was found to be the least among the three states. The slave is obtained by closing J20, J18, and J17, while closing J9, J16, and J19 results in the master. It should be noted that these groups are mutually exclusive since, if both were to be closed, the outputs of the operational amplifiers would be short-circuited.

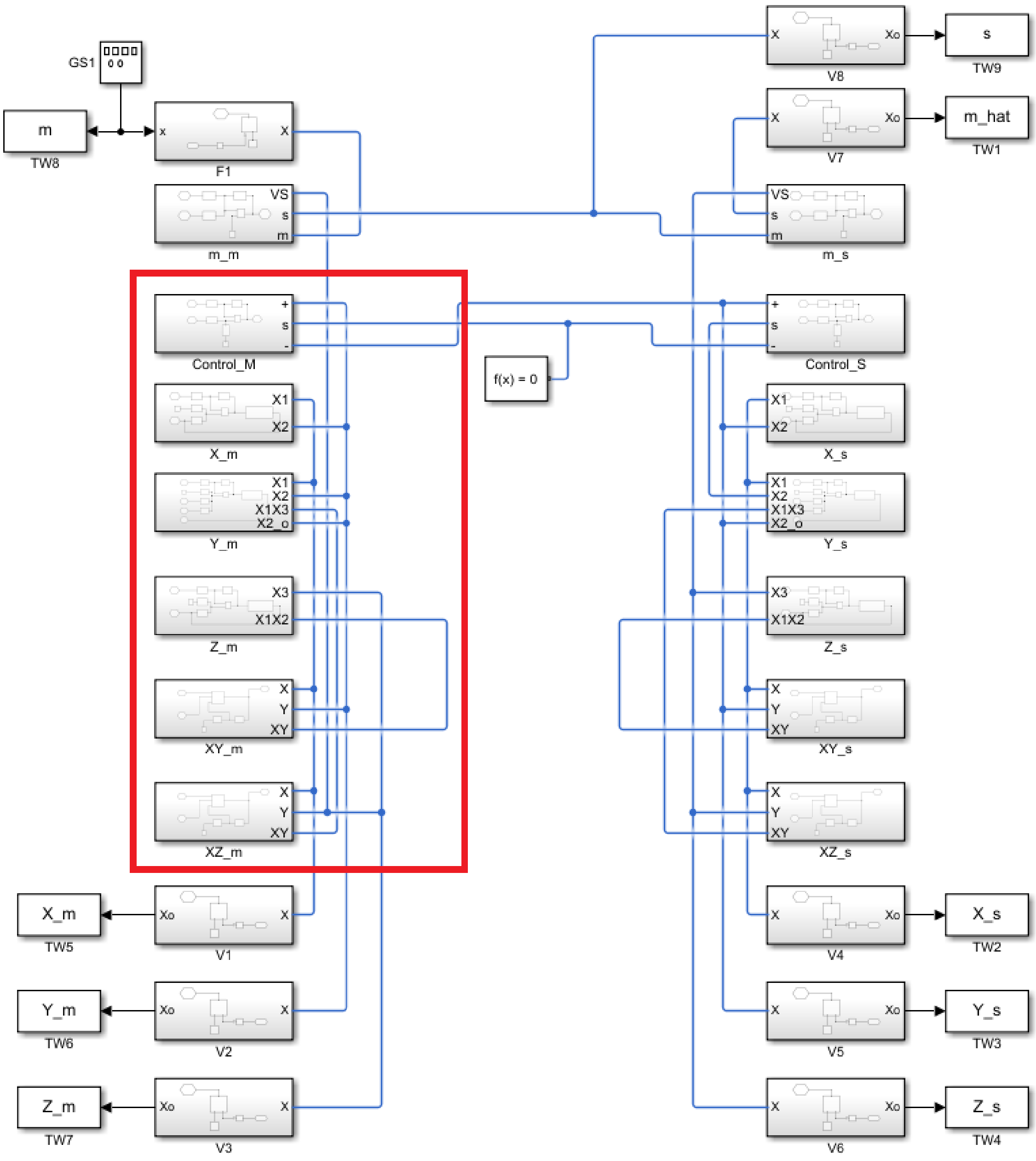

Figure 5.18 – Simulink Scheme of the circuit.

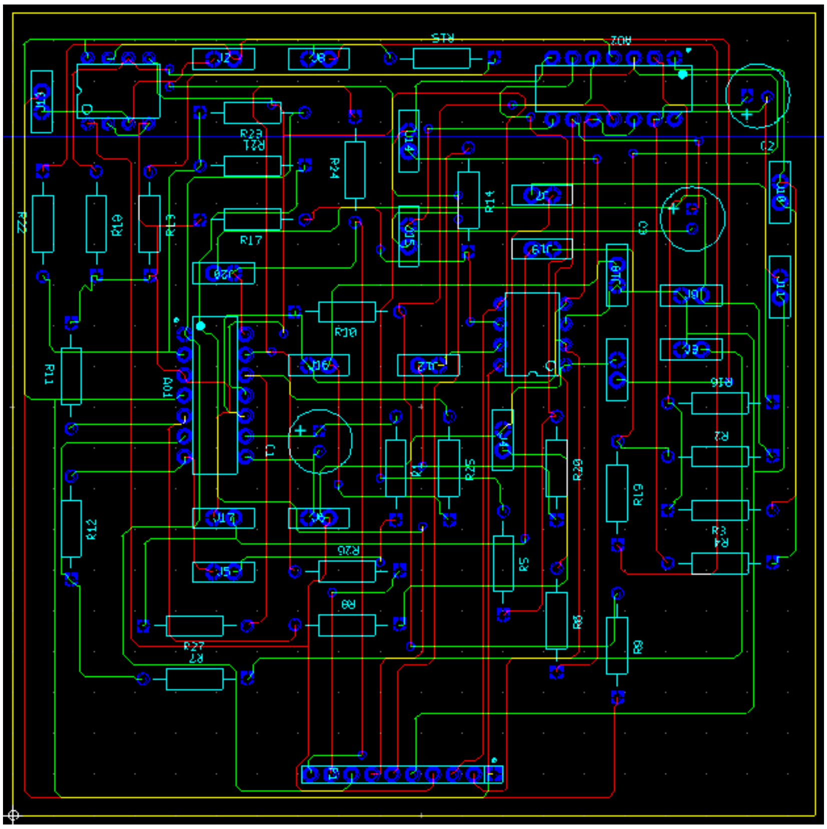

Figure 5.19 – Schematic of the printed circuit. Red: lower trails. Green: upper trails. Dark Blue: welding points. Light blue: component description.

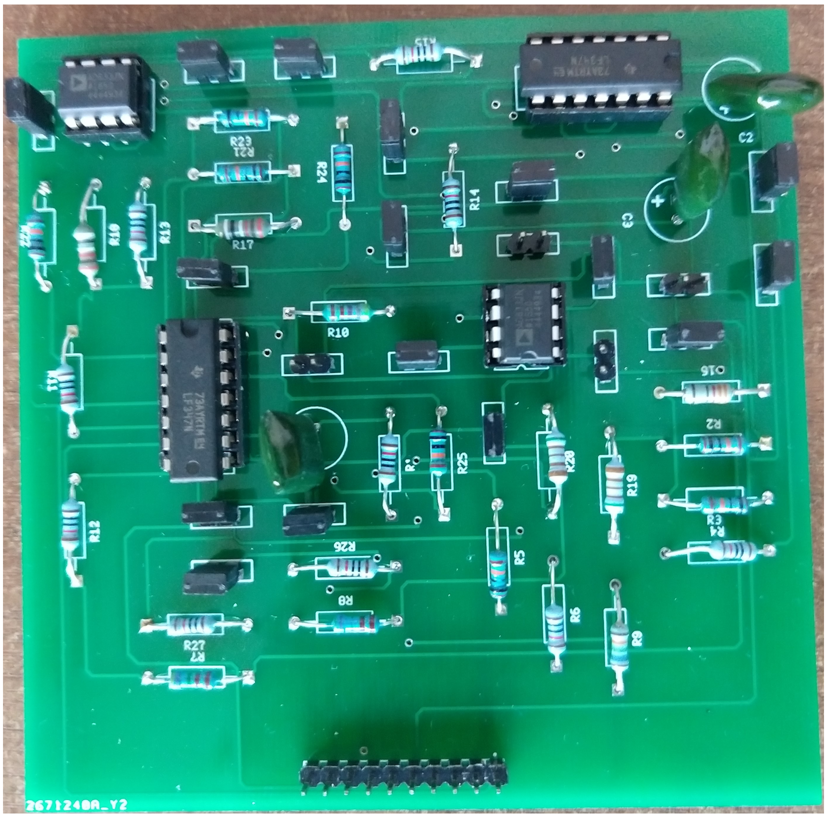

Figure 5.20 – Photo of the circuit.

Table 5.3 – Netlists: Resistors, Capacitors.

Code	**Nominal Value**	**Pin 1**	**Pin 2**
R1	10kΩ	X_L	AO1_9
R2	100kΩ	Y_2	AO2_3
R3	100kΩ	AO2_2	AO2_1
R4	10kΩ	Y_L	AO2_13
R5	100kΩ	AO2_6	AO2_7
R6	10kΩ	Z_L	AO2_9
R7	4.75kΩ	Y_1	AO1_6
R8	4.75kΩ	X_1	AO1_5
R9	165kΩ	Z_3	AO2_5
R10	4.87kΩ	GND	AO2_5
R11	10kΩ	GND	AO1_5
R12	10kΩ	AO1_6	AO1_7
R13	10kΩ	GND	AO2_3
R14	10kΩ	M1_Z	XY
R15	10kΩ	M2_Z	XZ
R16	3.32kΩ	X_2	AO2_2
R17	90.9kΩ	GND	M1_Z
R18	90.9kΩ	GND	M2_Z
R19	5.11kΩ	XZ_2	AO2_3
R20	5.11kΩ	XY_3	AO2_6
R21	100kΩ	M	AO1_2
R22	100kΩ	SV	AO1_2
R23	100kΩ	AO1_2	S
R24	100kΩ	AO1_13	U
R25	100kΩ	C_R1	AO1_12
R26	10kΩ	Y_s	AO1_13
R27	10kΩ	C_R2	AO1_12
C1	100nF	AO1_9	AO1_8
C2	100nF	AO2_13	AO2_14
C3	100nF	AO2_9	AO2_8

Table 5.4 – Netlists: Jumpers.

Code	Pin 1	Pin 2
J1	X	SV
J2	Z	SV
J3	Y	Y_1
J4	X	X_1
J5	AO1_7	X_L
J6	AO1_8	X
J7	X	X_2
J8	XZ	XZ_2
J9	Y	Y_2
J10	AO2_1	Y_L
J11	AO2_14	Y
J12	XY	XY_3
J13	Z	Z_3
J14	AO2_7	Z_L
J15	AO2_8	Z
J16	GND	C_R1
J17	GND	C_R2
J18	Y	C_R1
J19	Y	C_R2
J20	U	Y_2

Table 5.5 – Netlists: Integrated Circuits.

Code	AO1	AO2	M1	M2
IC	LF347	LF347	AD633JNZ	AD633JNZ
Pin 1	S	AO2_1	X	X
Pin 2	AO1_2	AO2_2	GND	GND
Pin 3	GND	AO2_3	Y	Z
Pin 4	VCC+	VCC+	GND	GND
Pin 5	AO1_5	AO2_5	VCC-	VCC-
Pin 6	AO1_6	AO2_6	M1_Z	M2_Z
Pin 7	AO1_7	AO2_7	XY	XZ
Pin 8	AO1_8	AO2_8	VCC+	VCC+
Pin 9	AO1_9	AO2_9		
Pin 10	GND	GND		
Pin 11	VCC-	VCC-		
Pin 12	AO1_12	GND		
Pin 13	AO1_13	AO2_13		
Pin 14	U	AO2_14		

Figure 5.21 shows $x_s(t)$ in channel 1, $x_m(t)$ in channel 2, and $e_x(t) = x_s(t) - x_m(t)$ in MATH (it is a mode in oscilloscope). Figure 5.22 shows $y_s(t)$ in channel 1, $y_m(t)$ in channel 2, and $e_y(t) = y_s(t) - y_m(t)$ in MATH. Finally, Figure 5.23 shows $z_s(t)$ in channel 1, $z_m(t)$ in channel 2, and $e_z(t) = z_s(t) - z_m(t)$ in MATH. It should be noted that the synchronization errors are small, indicating that the systems were able to achieve synchronization, as in the simulation.

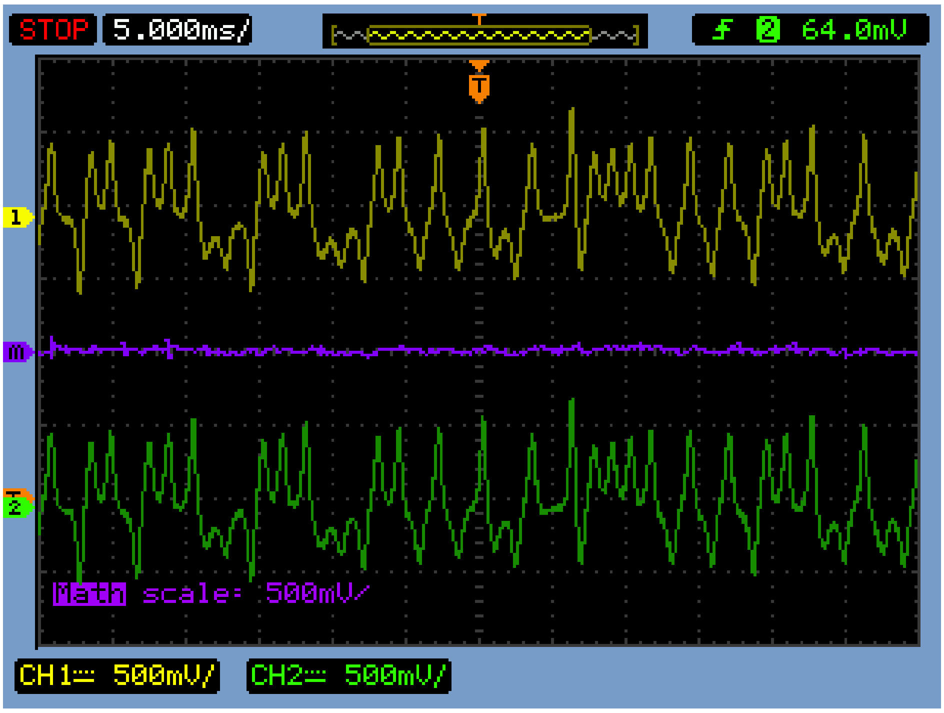

Figure 5.21 – Observation of experimental $x_m(t)$ (Channel 2), $x_s(t)$ (Channel 1), and $e_x(t)$ (MATH).

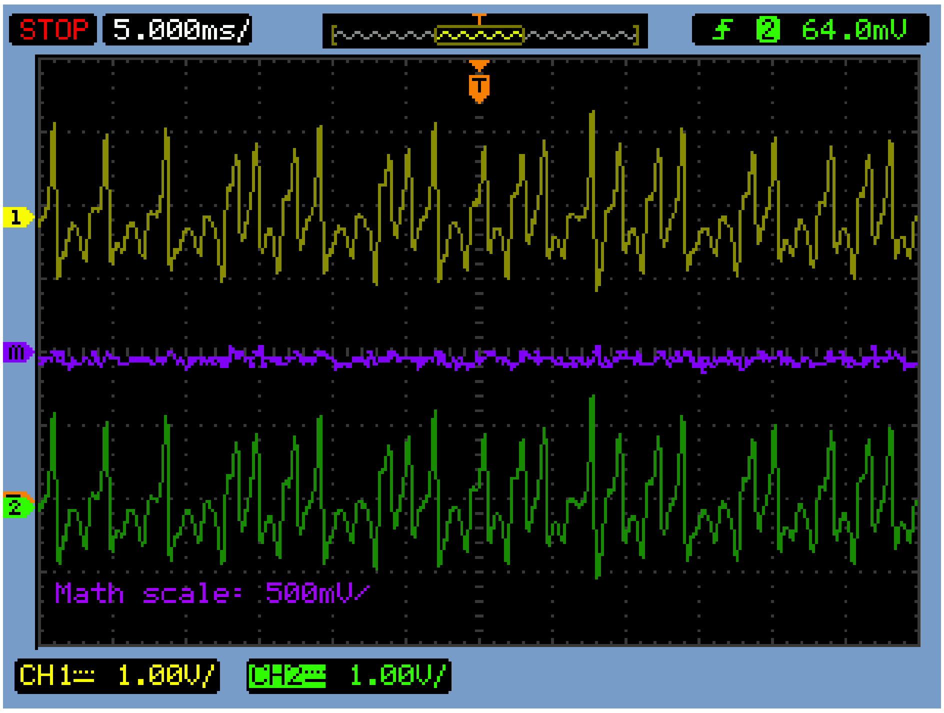

Figure 5.22 – Observation of experimental $y_m(t)$ (Channel 2), $y_s(t)$ (Channel 1), and $e_y(t)$ (MATH).

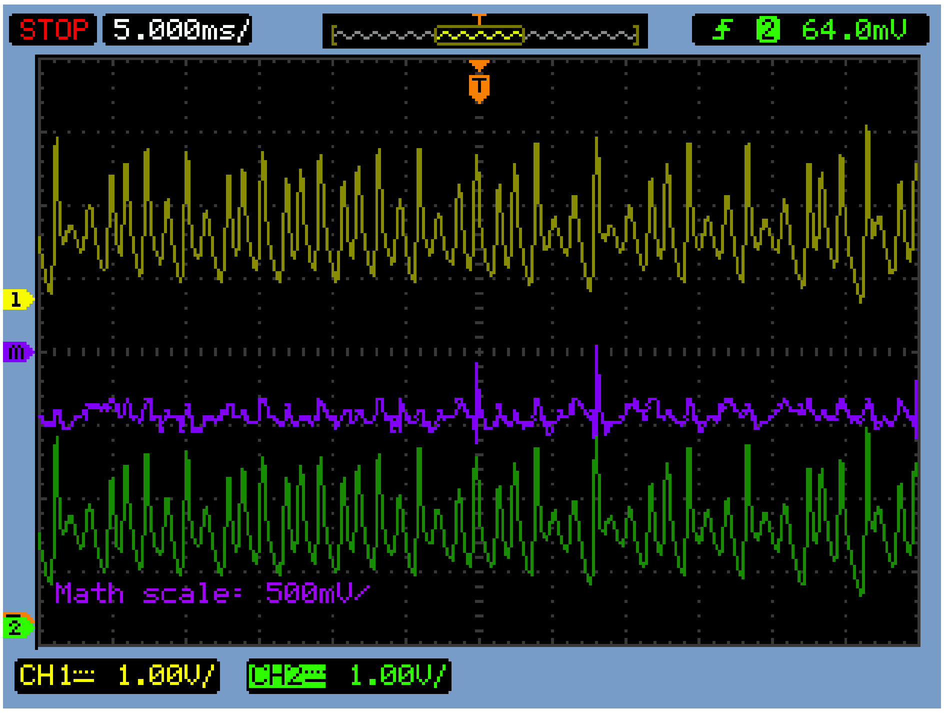

Figure 5.23 – Observation of experimental $z_m(t)$ (Channel 2), $z_s(t)$ (Channel 1), and $e_z(t)$ (MATH).

Figure 5.24 shows the message signal $m(t)$ in channel 1, a square wave of frequency 100 Hz, and amplitude 0.5 V. In channel 2 is presented the recovered signal $\hat{m}(t)$. Comparing both, it is evident that they are very similar. In Figure 5.25 is shown the encrypted signal. It is noticeable that the encrypted signal was gravely distorted even though it still possessed the abrupt transitions typical to square waves. It would be possible for a binary transmission system to employ the encryption directly, using an appropriate filter to invert it and remove its DC component. Nevertheless, there would be significant bit errors in some points.

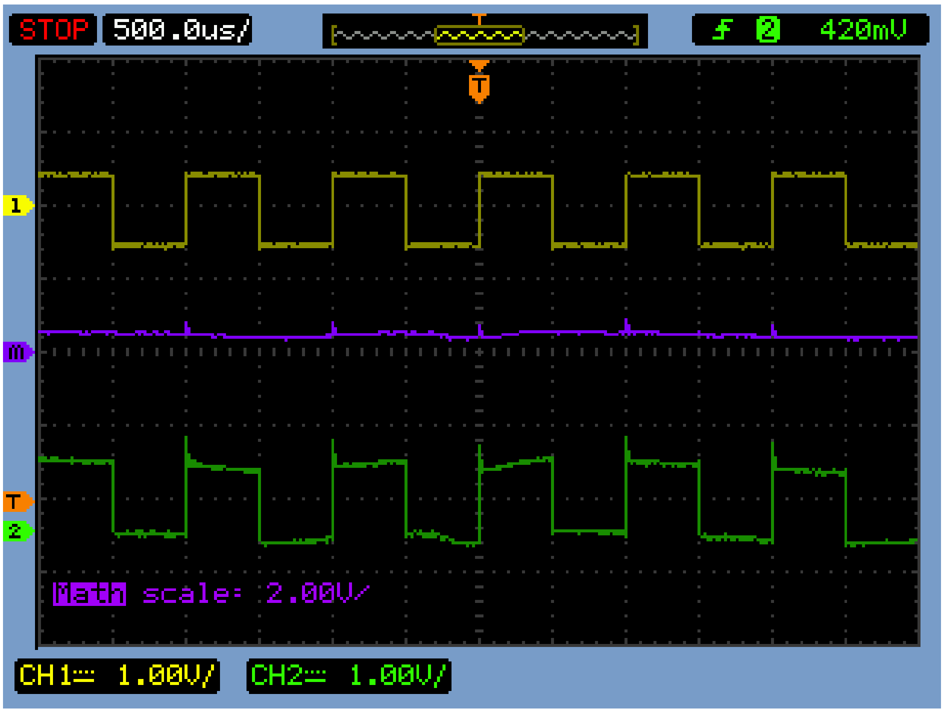

Figure 5.24 – Observation of experimental $m(t)$ (Channel 2), $\hat{m}(t)$ (Channel 1), and $m(t) - \hat{m}(t)$ (MATH) for square wave

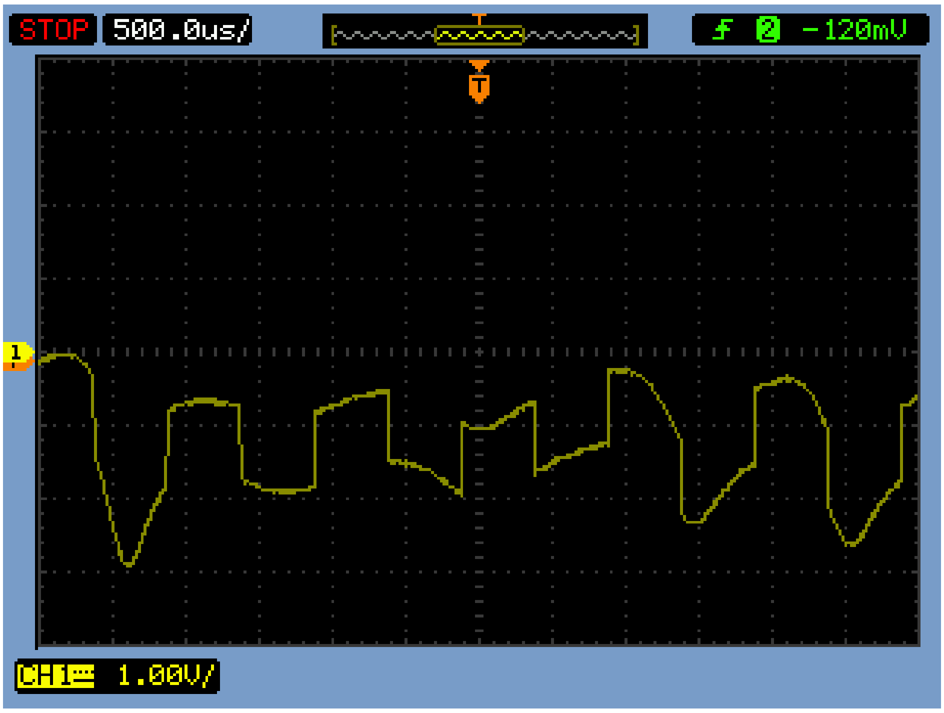

Figure 5.25 – Observation of experimental $-s(t)$ for square wave

Figures 5.26 and 5.27 have the same configuration from Figures 5.24 and 5.25 respectively, but with the message signal changed to a sinusoidal wave. The same applies to Figures 5.28 and 5.29, but for a triangular wave. The encrypted signal for the sinusoidal wave was very similar to the encrypted signal for the triangular wave. It would not be absurd to take one for another. Even then, the behavior of the original waveforms is still noticeable, even if not as evident as for the square wave.

Thus, from the figures, it can be observed that in both simulations and in the circuit implementation of the system, the master system correctly encodes the message and the slave system decodes the message. The performance is good though the system is underactuated and even in the presence of real components, and consequently, even in the presence of disturbances.

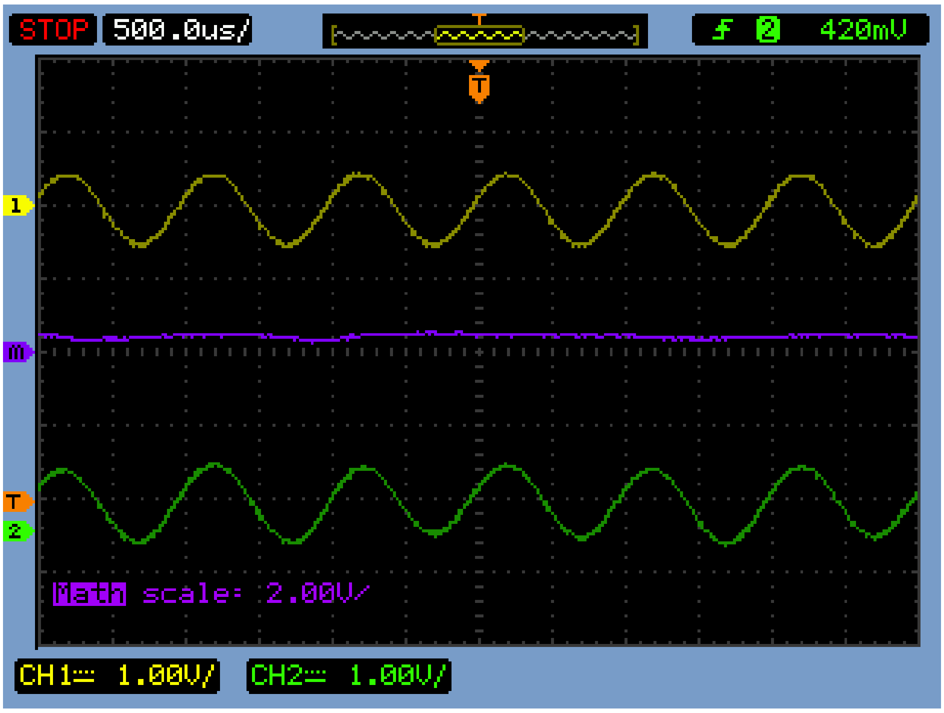

Figure 5.26 – Observation of experimental $m(t)$ (Channel 2), $\hat{m}(t)$ (Channel 1), and $m(t) - \hat{m}(t)$ (MATH) for sinusoidal wave.

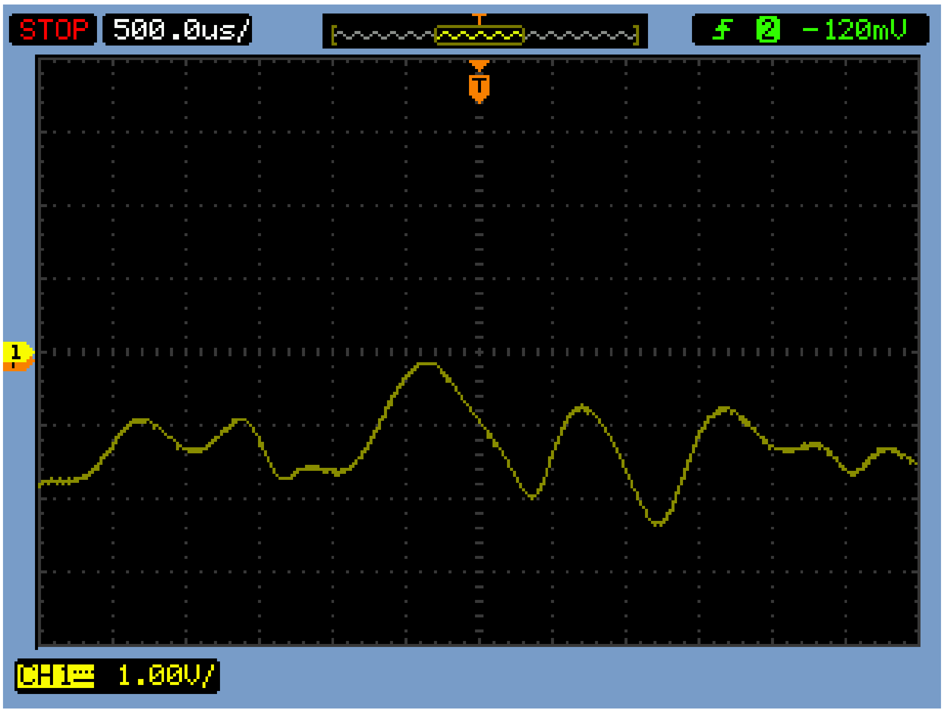

Figure 5.27 – Observation of experimental $-s(t)$ for sinusoidal wave.

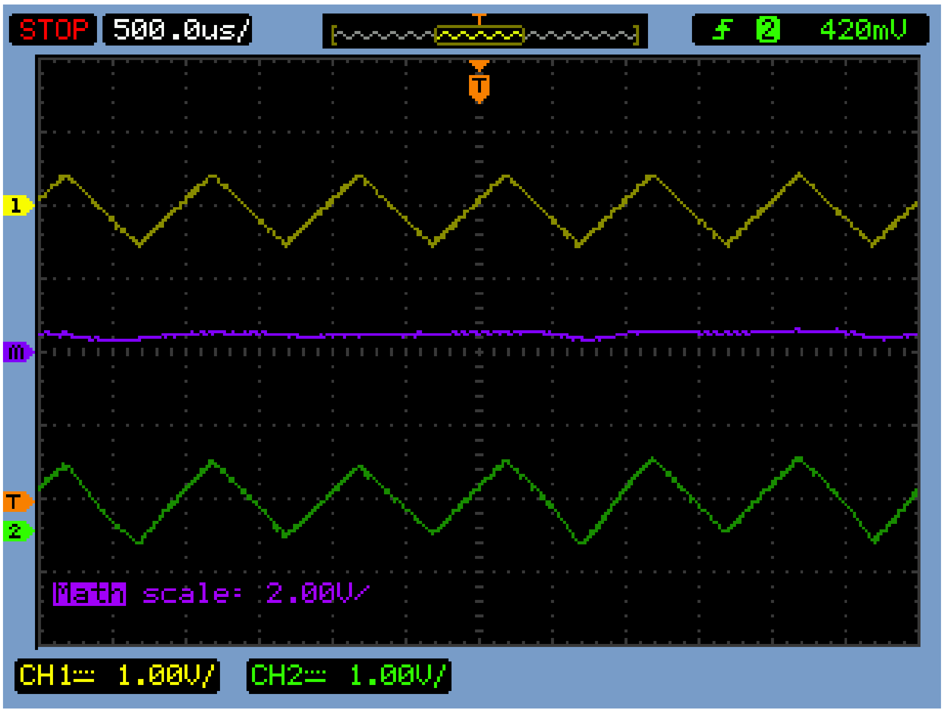

Figure 5.28 – Observation of experimental $m(t)$ (Channel 2), $\hat{m}(t)$ (Channel 1), and $m(t) - \hat{m}(t)$ (MATH) for triangular wave.

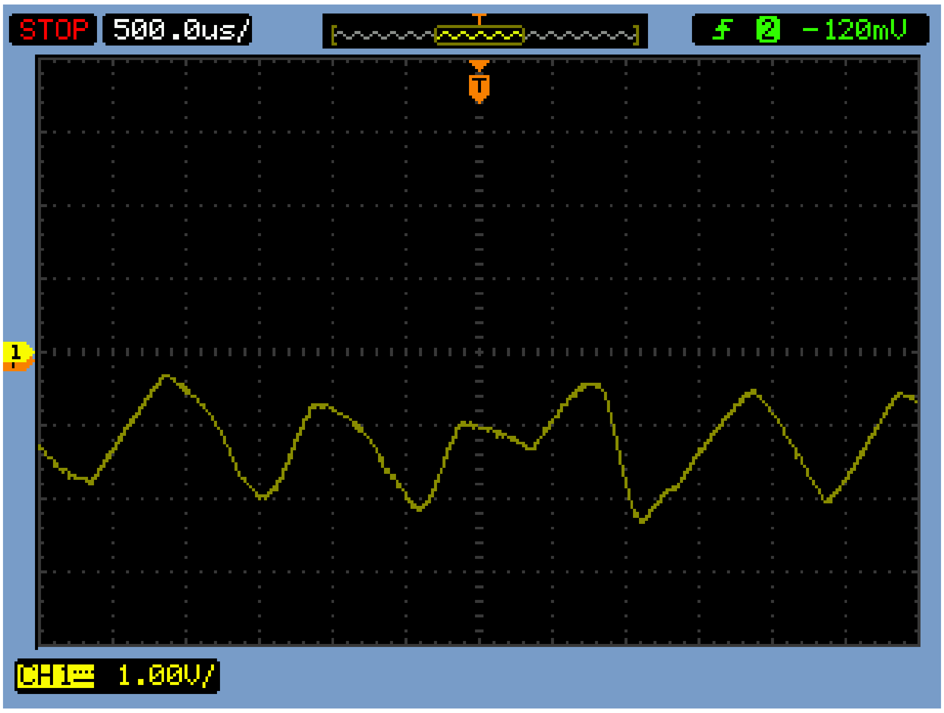

Figure 5.29 – Observation of experimental $-s(t)$ for triangular wave.

5.7 SUMMARY

A simple scheme for the synchronization of a chaotic system based on Lyapunov theory has been proposed. Remarkable features of the proposed approach are stability, robustness, and convergence to a small neighborhood of the origin, even when bounded disturbances are present. The proposed synchronizer has been analogically simulated and applied to secure telecommunication to validate the method. It should be noted that the encrypted signal still holds resemblance to the original signal, as expected, even in the presence of disturbances (for instance, due to the tolerances and nonideal behavior of the electronic devices).

6 A SCHEME FOR SECURE COMMUNICATION BASED ON LÜ HYPERCHAOTIC SYSTEM AND LYAPUNOV THEORY

The research outcomes of this chapter have been published as a conference paper entitled "A Scheme for Encryption/Decryption based on Hyperchaotic Systems and Lyapunov Theory" in [12]. *This chapter has extended and improved some parts compared to the original paper*.

The study of chaos, and especially, the synchronization of chaotic systems, has been intensified over the last few years [10, 146, 147, 150, 152, 153]. Synchronization of chaotic systems is frequently proposed in the literature and has attracted considerable attention, especially to the engineering community, due mainly to its application to secure communication [83, 152, 154–158], encryption [159–161], neural computing [74, 103, 127, 162], economics [126], biological systems [163], and biomedical engineering [76, 149].

Particularly, the synchronization of chaotic systems for secure communication has become popular since 1990 with the introduction of the work of Pecora and Carroll [39]. The Pecora and Carroll method (PC) describes the synchronization of two identical master-slave systems and their application to secure communication. Since then, several types of synchronization have been defined, such as anti-synchronization [147, 164], complete synchronization [153], lag synchronization [148], and others.

Communication systems based on synchronization, such as chaotic signal masking, have been successfully applied to secure communication [165]. Chaotic masking consists of encoding information by adding it to the chaotic signals in the transmitter. The masked information is then transmitted to the receiver, where it is decrypted when the master and slave systems are synchronized [7]. In most studies regarding chaotic systems synchronization, the dimension of the control input is the same as that of the state vector. On the other hand, insofar as the synchronization of chaotic systems is considered, synchronization, where the number of control input is less than the number of state variables, has sometimes been discussed [166, 167].

However, synchronization schemes for underactuated hyperchaotic systems are rarely found in the literature. For instance, in [164], the anti-synchronization problem is addressed by using a sliding mode controller. Numerical simulations were presented to show the effectiveness of the proposed scheme. However, the anti-synchronization was achieved using two inputs described by sophisticated control laws, which make the circuit implementation difficult. In addition, disturbances were not taken into account, and applications to anti-synchronization were not considered. In [168], an adaptive synchronization of a hyperchaotic system was proposed. The synchronization was achieved with a single-input linear feedback controller and in the presence of uncertain parameters. However, disturbances were not considered, and an application for such synchronization was not shown. In [153], a complete synchronization of two identical delay hyperchaotic systems via a single-input control and linear control law was suggested. However, as in [168], disturbances were not considered, and applications for secure telecommunication were not proposed. In [169], a sliding mode control scheme was utilized to achieve the synchronization of a hyperchaotic circuit with a single input, but also, neither disturbances nor applications in secure communications were considered. Also, the complexity of the proposed control law limits further circuit implementation and its consequent application to secure telecommunication.

Because of the statements above, it can be seen that all previously cited works present some of the following drawbacks.

1) The presence of disturbances in the theoretical analysis was not considered. The application of the proposed synchronization for secure communication is scant.

2) The underactuated case was rarely considered.

This chapter considers the synchronization of an underactuated hyperchaotic system and its application to secure communication in virtue of the facts mentioned above. More specifically, the problem of hyperchaotic synchronization based on a single control input is considered. However, in contrast to the literature, disturbances in the stability analysis are considered, with a positive impact on the robustness of the method, and apply the proposed scheme to secure communication. Finally, extensive simulations were performed to show the performance of the proposed approach and its effectiveness in encryption for parallel secure telecommunication. As far as the author knows, this is the first underactuated synchronization scheme of a hyperchaotic system with all these characteristics simultaneously in the literature, which is the main contribution of this chapter.

The chapter is organized as follows. Section 6.1 presents the problem and main assumptions. The synchronization error and control law, which ensures that the synchronization errors are bounded and finite-time convergent, are shown in Section 6.2. In Section 6.3, a study of the proposed method in secure telecommunication is presented. In Section 6.4, simulations are performed to validate the proposed scheme. The conclusions of the chapter are shown in Section 6.5.

6.1 PROBLEM STATEMENT

At first, the hyperchaotic system used in this chapter are defined. Consider the master Lü hyperchaotic system [170]:

$$\begin{cases} \dot{x}_m &= a(y_m - x_m) \\ \dot{y}_m &= cy_m - x_m z_m + w_m \\ \dot{z}_m &= x_m y_m - bz_m \\ \dot{w}_m &= z_m - dw_m \end{cases} \tag{6.1}$$

and the perturbed underactuated slave Lü hyperchaotic system:

$$\begin{cases} \dot{x}_s &= a(y_s - x_s) + h_1(t) \\ \dot{y}_s &= cy_s - x_s z_s + w_s + h_2(t) + u \\ \dot{z}_s &= x_s y_s - bz_s + h_3(t) \\ \dot{w}_s &= z_s - dw_s + h_4(t) \end{cases} \tag{6.2}$$

where x_m, y_m, z_m, and w_m are the state variables of the master system; and x_s, y_s, z_s, and w_s are the state variables of the slave system. The systems parameters are $a = 15$, $b = 5$, $c = 10$, and $d = 1$. The slave system disturbances are $h_1(t)$, $h_2(t)$, $h_3(t)$, and $h_4(t)$, and u is the control signal.

Remark 6.1.1 The restriction of the system being underactuated is that the synchronizer only has access to information from the actuated master states, that is, from the states where there is a control signal. In other words, in the (6.2) system, the control u cannot have the presence of the states x_m, z_m, and w_m. Although, the other states of the master system and all the states of the slave system are available.

Fact 6.1.1 In [170] was proved that (6.1) is hyperchaotic and that the system is dissipative. A consequence of the system being dissipative is also being bounded. With the boundedness of the system (6.1), the following inequalities are true:

$$\begin{aligned} |x_m(t)| &\leq \bar{x} \\ |y_m(t)| &\leq \bar{y} \\ |z_m(t)| &\leq \bar{z} \\ |w_m(t)| &\leq \bar{w} \end{aligned} \tag{6.3}$$

$\forall t \geq 0$, where $\bar{x}$, $\bar{y}$, $\bar{z}$, and $\bar{w}$ are unknown positive constants.

Assumption 6.1.1 Assume that the disturbances are bounded. More precisely,

$$
\begin{aligned}
|h_1(t)| &\leq \bar{h}_1 \\
|h_2(t)| &\leq \bar{h}_2 \\
|h_3(t)| &\leq \bar{h}_3 \\
|h_4(t)| &\leq \bar{h}_4
\end{aligned}
\tag{6.4}
$$

$\forall t \geq 0$, being $\bar{h}_1$, $\bar{h}_2$, $\bar{h}_3$, and $\bar{h}_4$ unknown constants.

Then, by considering Assumption 6.1.1, this chapter proposes the synchronization of (6.2) and (6.3) by using a scalar control signal only in the second state, irrespective of the presence of disturbances, which can affect all the states of the slave system.

Remark 6.1.2 It is interesting to notice that systems (6.2) and (6.3) are different for the sake of the presence of disturbances. Disturbances are inevitable in practical implementations because of the tolerance of the components, environmental conditions, and electromagnetic noise. This fact relaxes the assumption of "identity of structure," which is usually found in the literature (see, for example, [153, 168, 169], and the references therein).

6.2 SYNCHRONIZATION ERROR AND PROPOSED SIGNAL CONTROL

By defining the synchronization errors as

$$
\begin{cases}
e_1(t) = x_s - x_m \\
e_2(t) = y_s - y_m \\
e_3(t) = z_s - z_m \\
e_4(t) = w_s - w_m
\end{cases}
\tag{6.5}
$$

and by employing (6.2) and (6.3) in the time-derivative of (6.5), results

$$
\begin{cases}
\dot{e}_1 &= a(e_2 - e_1) + h_1 \\
\dot{e}_2 &= ce_2 - e_1 e_3 - e_1 z_m - e_3 x_m + e_4 + h_2 + u \\
\dot{e}_3 &= e_1 e_2 + e_1 y_m + e_2 x_m - be_3 + h_3 \\
\dot{e}_4 &= e_3 - de_4 + h_4
\end{cases}
\tag{6.6}
$$

The main result of the chapter is summarized in what follows.

Theorem 6.2.1 Consider the master and slave systems described by (6.2) and (6.3), and the proportional control law defined by

$$u = -\psi e_2 \tag{6.7}$$

If,

$$\psi > \delta \tag{6.8}$$

Then, the synchronization error is uniformly ultimately bounded and converges in finite-time to the compact set

$$\Omega = \left\{ e \in \Re^4 \mid ||e|| \leq \theta \right\} \tag{6.9}$$

where $\delta = c + 0.5[\sigma_2 \bar{h}_2^2 + \sigma_5^{-1}(a^2\gamma^2 + \bar{z}^2) + \sigma_7]$, $\theta = \sqrt{\frac{\beta}{\rho}}$, $\beta = \beta_u + \beta_n$, $\rho = min\{\rho_1, \rho_2, \rho_3, \rho_4\}$, $\beta_u = 0.5\sigma_2^{-1}$, $\beta_n = 0.5(\gamma\sigma_1^{-1}\bar{h}_1^2 + \sigma_3^{-1}\bar{h}_3^2 + \sigma_4^{-1}\bar{h}_4^2)$, $\rho_1 = a\gamma - 0.5(\gamma\sigma_1 + \sigma_5 + \sigma_6\bar{y}^2)$, $\rho_2 = \psi - \delta$, $\rho_3 = b - 0.5(\sigma_3 + \sigma_6^{-1} + \sigma_8)$, $\rho_4 = d - 0.5(\sigma_4 + \sigma_7^{-1} + \sigma_8^{-1})$, $\|e\|^2 = e_1^2 + e_2^2 + e_3^2 + e_4^2$, and $\sigma_i, i = 1, ..., 8$ are positive constants.

Proof.

Consider the following Lyapunov function candidate

$$V = \frac{1}{2}(\gamma e_1^2 + e_2^2 + e_3^2 + e_4^2) \tag{6.10}$$

where $\gamma > 0$. The time-derivative of (6.10) along the trajectories of (6.6), results

$$\begin{aligned} \dot{V} &= \gamma e_1[a(e_2 - e_1) + h_1] + e_2(ce_2 - e_1e_3 - e_1z_m - e_3x_m + e_4 + h_2 + u) \\ &\quad + e_3(e_1e_2 + e_1y_m + e_2x_m - be_3 + h_3) + e_4(e_3 - de_4 + h_4) \end{aligned} \tag{6.11}$$

By replacing (6.7) in (6.11), implies

$$\begin{aligned} \dot{V} &= -\gamma a e_1^2 - e_2^2(\psi - c) - be_3^2 - de_4^2 + \gamma e_1h_1 + e_2h_2 + e_3h_3 + e_4h_4 \\ &\quad + e_1e_2(\gamma a - z_m) + e_1e_3y_m + e_2e_4 + e_3e_4 \end{aligned} \tag{6.12}$$

Analyzing when $\dot{V} \leq 0$, by employing the inequality of Young, the Assumption 6.1.1, and the Fact 6.1.1, then

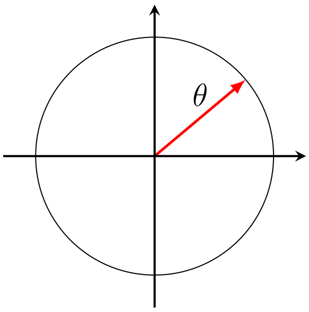

Figure 6.1 – Bounded set.

$$\begin{aligned}
&\gamma e_1 h_1 \leq 0.5\gamma(\sigma_1 e_1^2 + \sigma_1^{-1}\bar{h}_1^2)\\
&e_2 h_2 \leq 0.5(\sigma_2 \bar{h}_2^2 e_2^2 + \sigma_2^{-1})\\
&e_3 h_3 \leq 0.5(\sigma_3 e_3^2 + \sigma_3^{-1}\bar{h}_3^2)\\
&e_4 h_4 \leq 0.5(\sigma_4 e_4^2 + \sigma_4^{-1}\bar{h}_4^2)\\
&e_1[e_2(a\gamma - z_m)] \leq 0.5[\sigma_5 e_1^2 + \sigma_5^{-1} e_2^2(a^2\gamma^2 + \bar{z}^2)]\\
&e_1 e_3 y_m \leq 0.5(\sigma_6 e_1^2 \bar{y}^2 + \sigma_6^{-1} e_3^2)\\
&e_2 e_4 \leq 0.5(\sigma_7 e_2^2 + \sigma_7^{-1} e_4^2)\\
&e_3 e_4 \leq 0.5(\sigma_8 e_3^2 + \sigma_8^{-1} e_4^2)
\end{aligned} \tag{6.13}$$

By using (6.13), (6.12) implies

$$\dot{V} \leq -e_1^2\rho_1 - e_2^2\rho_2 - e_3^2\rho_3 - e_4^2\rho_4 + \beta_u + \beta_n \tag{6.14}$$

Note that: 1) there are values of γ and $\sigma_i, i = 1, ..., 8$ that make $\rho_1 > 0$, $\rho_3 > 0$, and $\rho_4 > 0$; and 2) ψ is chosen by the user so that (6.8) is satisfied, and, consequently, $\rho_2 > 0$. Therefore, there is a $\rho > 0$, and (6.14) can be rewritten as

$$\dot{V} \leq -\rho\,||e||^2 + \beta \tag{6.15}$$

Based on (6.15), $\dot{V} < 0$ when $||e|| > \theta$. Since θ is a positive constant, it can be claimed that the synchronization error is uniformly ultimately bounded [24]. In the region Ω note that if for any reason $||e||$ is not part of that region Ω, $\dot{V}$ becomes defined negative and forces the synchronization error convergence to the region Ω, according to (6.15). In addition, the convergence to the residual set Ω is in a finite time, due to the particular form of (6.15) [24]. Consequently, the synchronization error is uniformly ultimately bounded and converge to a ball with radius θ (Figure 6.1). □

REMARK 6.2.1 From the system structure and Lyapunov theory, an adequate control law is designed. Based on a trial-and-error procedure, all possibilities of underactuated control in the analysis was considered and the simplest one was chosen. Inequality of Young was used in the stability analysis in the process to make $\dot{V} < 0$ outside of a small compact set.

REMARK 6.2.2 Defining $D_i(i = 1, ..., 5)$ as the respective domains of $\sigma_i(i = 1, ..., 5)$, results

$$\begin{aligned}
D_1 &= \{\sigma_1 \in \Re | 0 < \sigma_1 < 2a\} \\
D_2 &= \left\{\sigma_2 \in \Re | 0 < \sigma_2 < \frac{2(\psi - c)}{\bar{h}_2^2}\right\} \\
D_3 &= \{\sigma_3 \in \Re | 0 < \sigma_3 < 2b\} \\
D_4 &= \{\sigma_4 \in \Re | 0 < \sigma_4 < 2d\} \\
D_5 &= \left\{\sigma_5 \in \Re | \frac{a^2 + \gamma^2 + \bar{z}^2}{2(\psi - c)} < \sigma_5 < 2a\gamma\right\} \\
D_6 &= \left\{\sigma_6 \in \Re | \frac{1}{2b} < \sigma_6 < \frac{2a\gamma}{\bar{y}^2}\right\} \\
D_7 &= \left\{\sigma_7 \in \Re | \frac{1}{2d} < \sigma_7 < 2\right\} \\
D_8 &= \left\{\sigma_8 \in \Re | \frac{1}{2d} < \sigma_8 < 2\right\}
\end{aligned} \tag{6.16}$$

The residual synchronization error considered is affected by the control gain ψ, disturbances, and upper bounds for the states of the master system, as can be seen from (6.8). The performance for the actuated states can be arbitrarily enhanced by increasing ψ. For non-actuated states, it can not be guaranteed that a change in the gain of the control will cause the residual synchronization error to decrease (main disadvantage of the scheme).

6.3 CHAOS-BASED SECURE COMMUNICATION

In order to have a well-posed problem, the following assumption is made.

ASSUMPTION 6.3.1 It is assumed that the messages are bounded. More specifically,

$$|m_i(t)| \leq \bar{m}_i, i = 1, ..., 4 \tag{6.17}$$

$\forall t \geq 0$, where m_1, m_2, m_3, and m_4 are the original messages and $\bar{m}_1$, $\bar{m}_2$, $\bar{m}_3$, and $\bar{m}_4$ are positive constants.

Further, motivated by [7], it can be defined

$$
\begin{aligned}
\hat{m}_1 &= s_1 - x_s \\
\hat{m}_2 &= s_2 - y_s \\
\hat{m}_3 &= s_3 - z_s \\
\hat{m}_4 &= s_4 - w_s
\end{aligned}
\tag{6.18}
$$

being $s_1 = x_m + m_1$, $s_2 = y_m + m_2$, $s_3 = z_m + m_3$, and $s_4 = w_m + m_4$ the encrypted messages; and $\hat{m}_1$, $\hat{m}_2$, $\hat{m}_3$, and $\hat{m}_4$ the decrypted messages.

Also, by using (6.18) and defining $\tilde{m}_i = \hat{m}_i - m_i, i = 1, ..., 4$, where $\tilde{m}_1$, $\tilde{m}_2$, $\tilde{m}_3$, and $\tilde{m}_4$ are the message errors, it can be concluded that

$$
\tilde{m}_i = -e_i, i = 1, ..., 4 \tag{6.19}
$$

Remark 6.3.1 Notice that the quality of the message reconstruction is the same as the synchronization, as shown in (6.19). Furthermore, the boundedness of the message error is assured when the synchronization error is bounded.

Remark 6.3.2 An overview of the secure communication scheme can be seen in Figure 6.2.

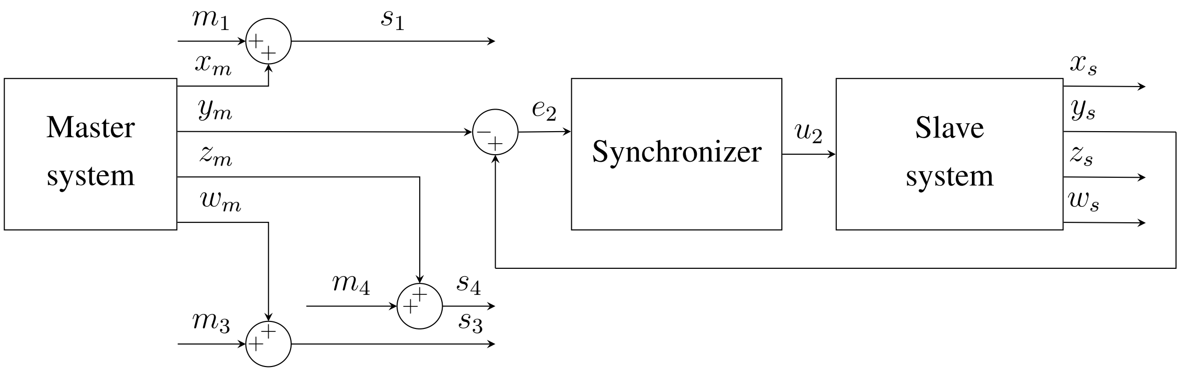

Figure 6.2 – Synchronization and secure communication scheme.

6.4 SIMULATION

Simulations were performed using Matlab 2020b® on a Windows 10 platform, with AMD Ryzen 7 1700 processor for all simulations, variable-step algorithm ODE45 solver, and relative tolerance of 10^{-10}. To make the chaotic systems implementable, the master and slave systems were scaled in size to 4%. The master and slave systems were synchronized by using the control law (6.7), with $\psi = 60$. This control gain was required to obtain adequate performance in all simulations. The initial conditions for the master slave and systems are $x_m(0) = 0.1$, $y_m(0) = 0.1$, $z_m(0) = 0$, $w_m = 0.1$, $x_s(0) = -0.2$, $y_s(0) = -0.2$, $z_s(0) = 0.1$, and $w_s(0) = 0$. Also, to evaluate the robustness of the proposed method in the presence of disturbances, the following scaled disturbances are chosed: $h_1(t) = 0.02sin(5t)$, $h_2(t) = 0.022cos(3t)$, $h_3(t) = 0.016sin(3t)$, and $h_4(t) = 0.002sin(6t)$.

Figures 6.3 - 6.6 show the synchronization performance for all states. Figures 6.7 - 6.10 show the synchronization errors. The synchronization is achieved fast, and the residual synchronization error is close to zero, even with just a proportional control signal. Based on Figures 6.3 - 6.10, the proposed synchronizer is stable and robust in the presence of disturbances.

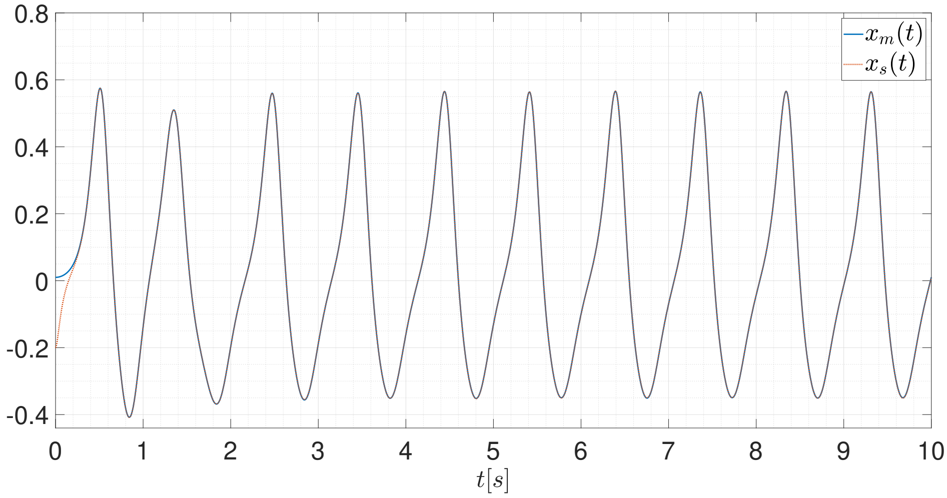

Figure 6.3 – Synchronization result of $x_m(t)$ and $x_s(t)$.

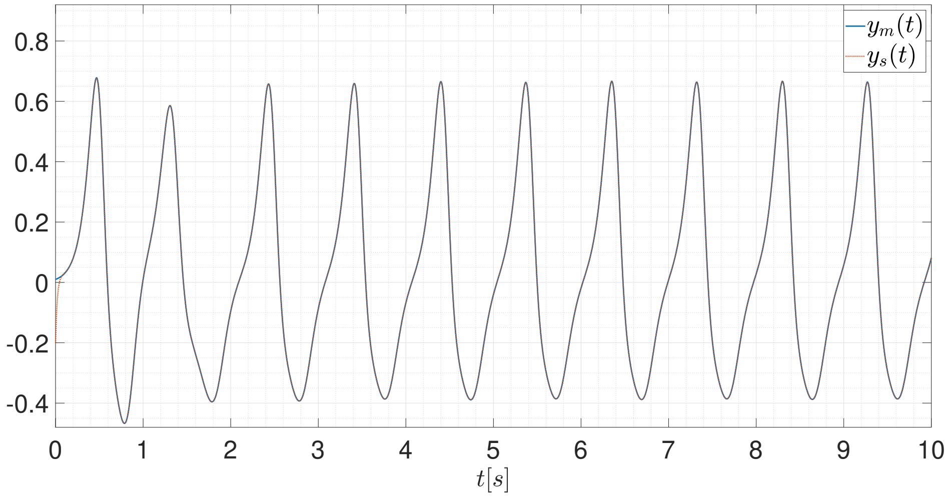

Figure 6.4 – Synchronization result of $y_m(t)$ and $y_s(t)$.

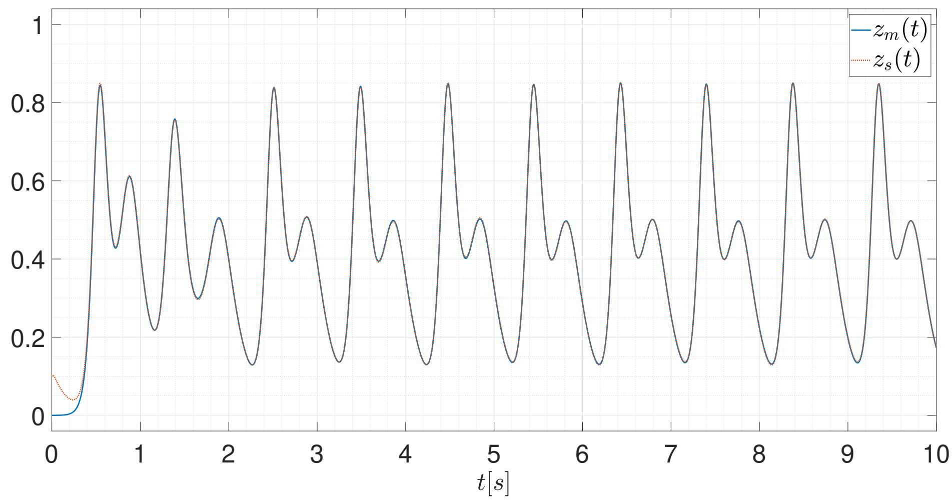

Figure 6.5 – Synchronization result of z_m(t) and z_s(t).

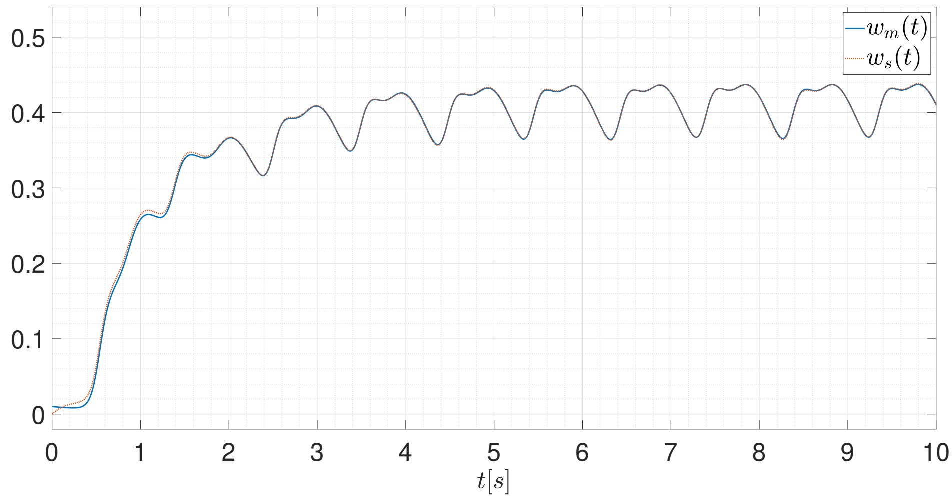

Figure 6.6 – Synchronization result of w_m(t) and w_s(t).

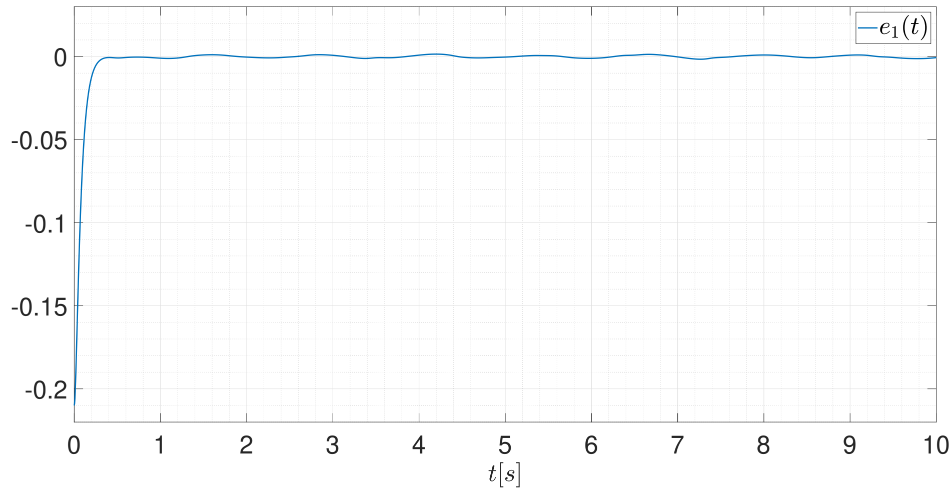

Figure 6.7 – Synchronization error of $x(t)$.

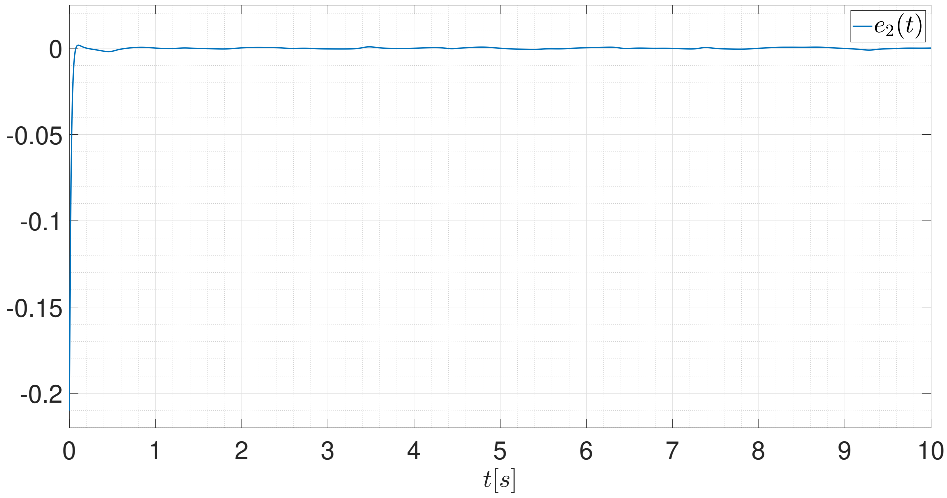

Figure 6.8 – Synchronization error of $y(t)$.

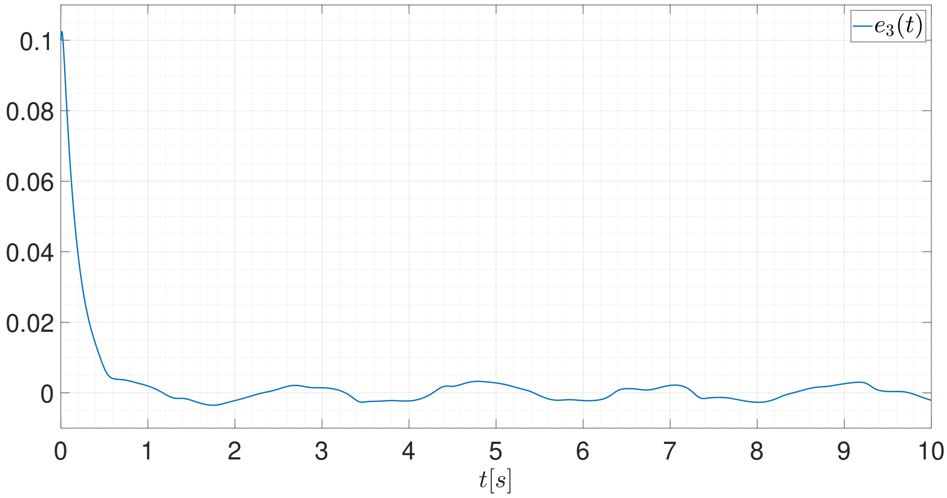

Figure 6.9 – Synchronization error of $z(t)$.

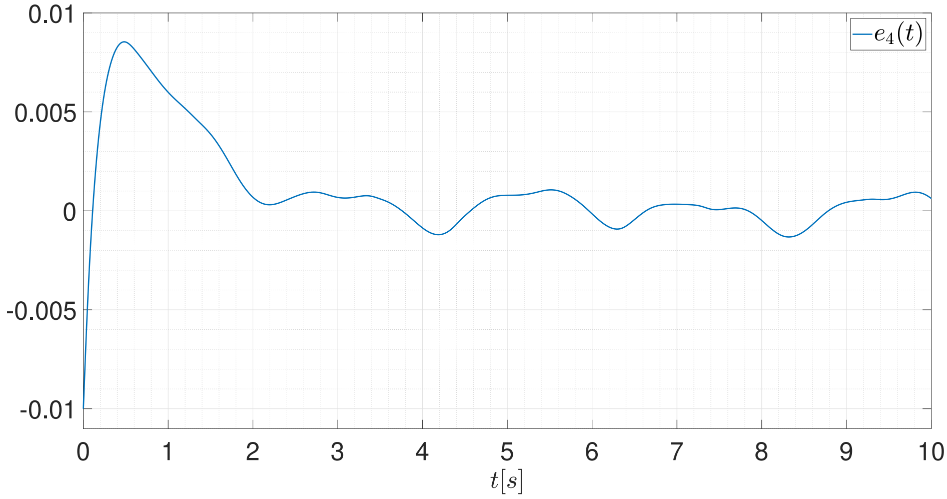

Figure 6.10 – Synchronization error of $w(t)$.

Table 6.1 shows that the optimal control gain value is approximately $\psi = 60$. Smaller values of ψ lead to larger values of β and smaller values of ρ, consequently, lead to greater synchronization error values. Theoretically, larger values of ψ should not increase the synchronization error value, but higher control gain can lead to greater approximation errors in computer simulations.

Table 6.1 – Root mean square of state errors for t = [0 10] seconds and consider that $\psi = 60$.

Root Mean Square of State Errors in the Proposed Algorithm					
	$e_{1_{rms}}$	$e_{2_{rms}}$	$e_{3_{rms}}$	e_{rms}	
0.01ψ	0.345177	0.378929	0.233219	0.057706	0.057706
0.1ψ	0.064951	0.072758	0.072758	0.005701	0.094601
ψ	0.014854	0.006642	0.011160	0.002588	0.018651
10ψ	0.021911	0.006746	0.013454	0.002563	0.025515
100ψ	0.021345	0.005004	0.013449	0.002575	0.024802

In the sequence, the application of the proposed synchronization scheme for secure communication is shown in Figures 6.11 - 6.22. The encryption/decryption of three messages is deemed, where the original messages were $m_1(t) = 0.3sin(3t) + 0.06cos(20t)$, $m_3(t) = 0.3sin(2t) + 0.1cos(15t) + 0.02cos(30t)$, and $m_4(t) = 0.06sin(5t) + 0.03sin(15t)$. It is noteworthy that the synchronization scheme and decryption algorithm only use available signals on the receiver.

Figures 6.11 - 6.16 show the performance of the synchronization of the slave and master systems in the presence of the introduced messages and disturbances. Figures 6.17 - 6.19 depict the performance in the encryption/decryption of the messages. Message errors are plotted in Figures 6.20 - 6.22. Note that original messages are very different from coded messages and very close to coded messages. Additionally, message errors are tiny.

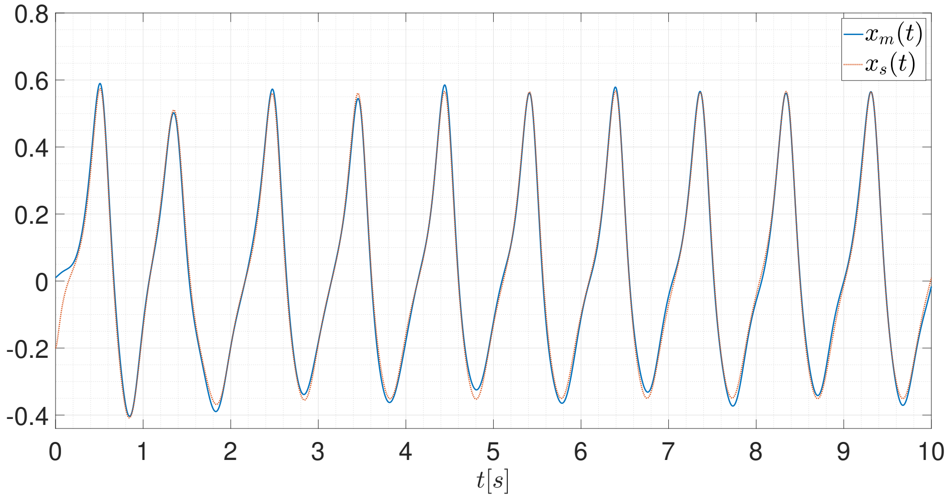

Figure 6.11 – Synchronization performance of $x_m(t)$ and $x_s(t)$.

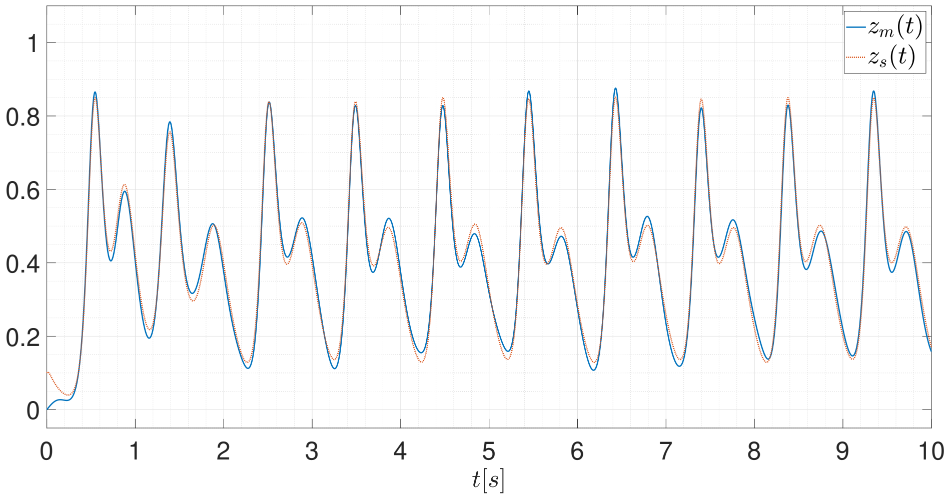

Figure 6.12 – Synchronization performance of $z_m(t)$ and $z_s(t)$.

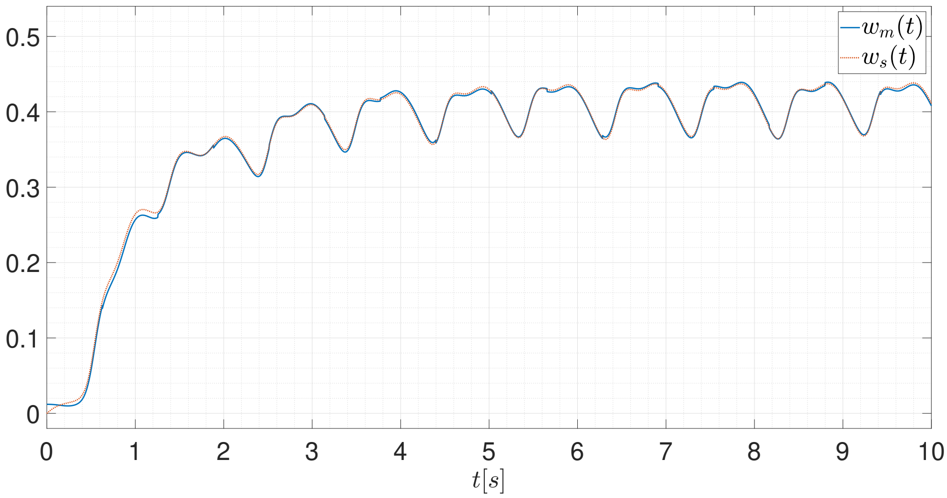

Figure 6.13 – Synchronization performance of $w_m(t)$ and $w_s(t)$.

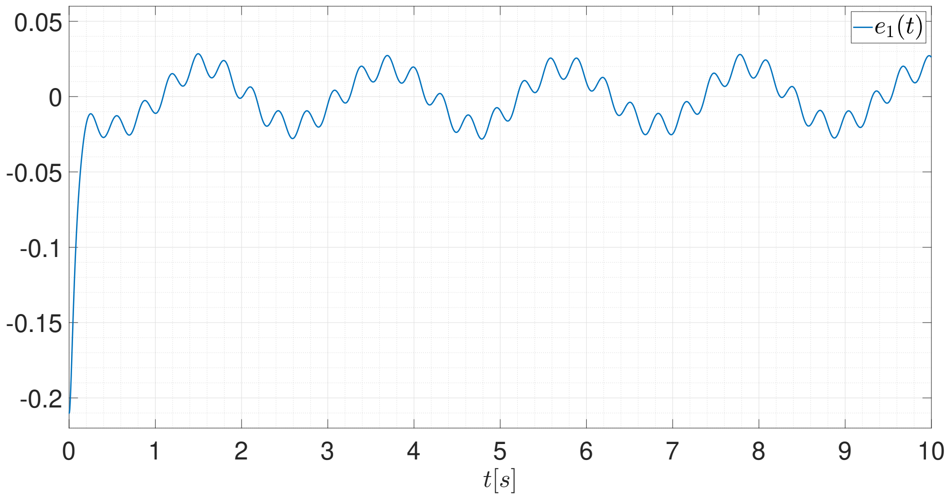

Figure 6.14 – Synchronization error of $x(t)$.

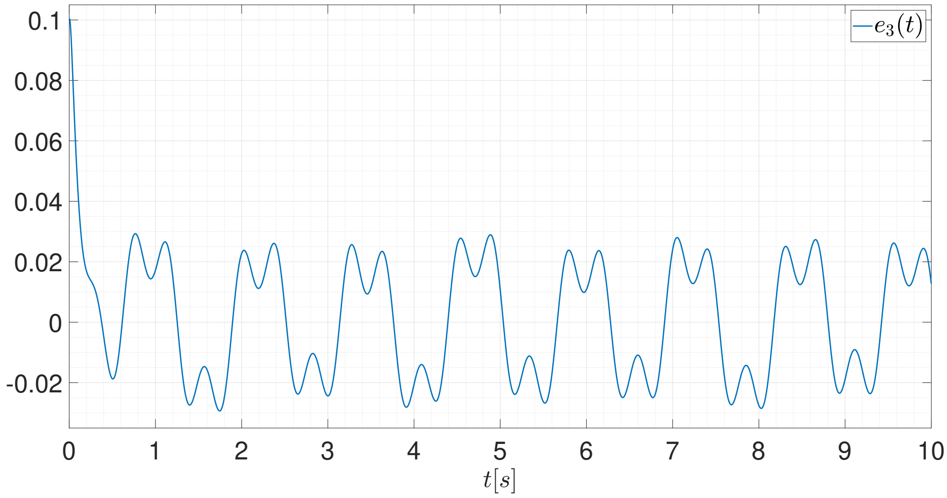

Figure 6.15 – Synchronization error of $z(t)$.

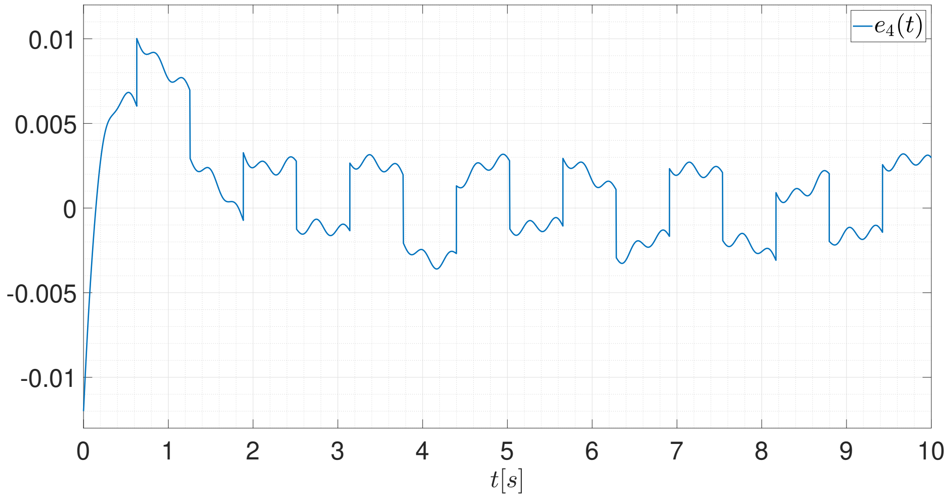

Figure 6.16 – Synchronization error of $w(t)$.

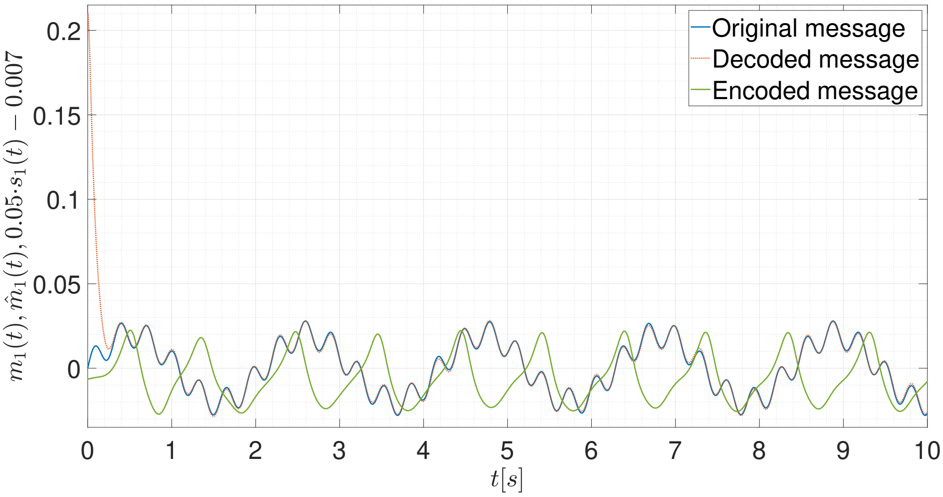

Figure 6.17 – Original message 1 (blue), decrypted message 1 (red), and encrypted message 1 (green).

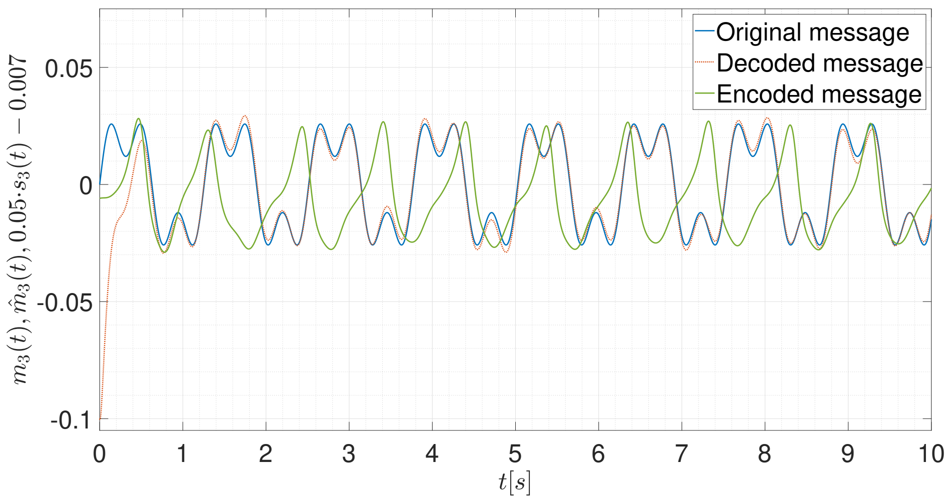

Figure 6.18 – Original message 3 (blue), decrypted message 3 (red), and encrypted message 3 (green).

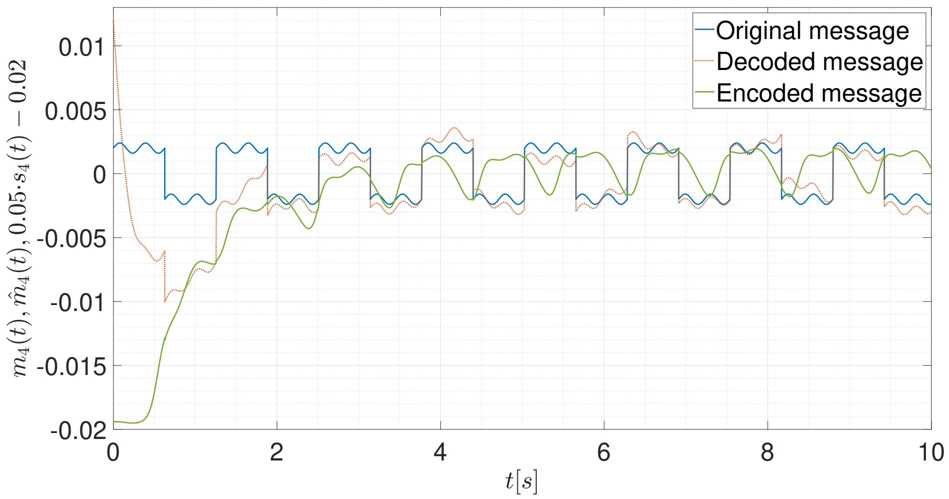

Figure 6.19 – Original message 4 (blue), decrypted message 4 (red), and encrypted message 4 (green).

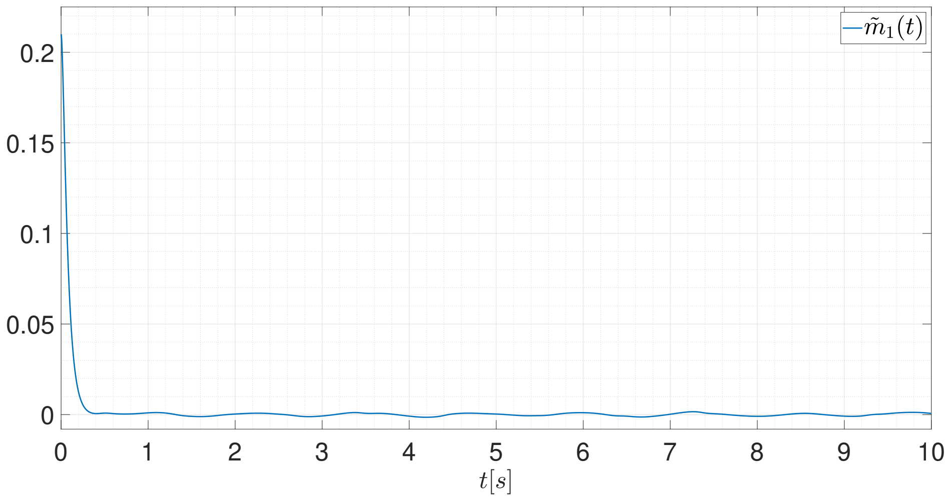

Figure 6.20 – Message error 1.

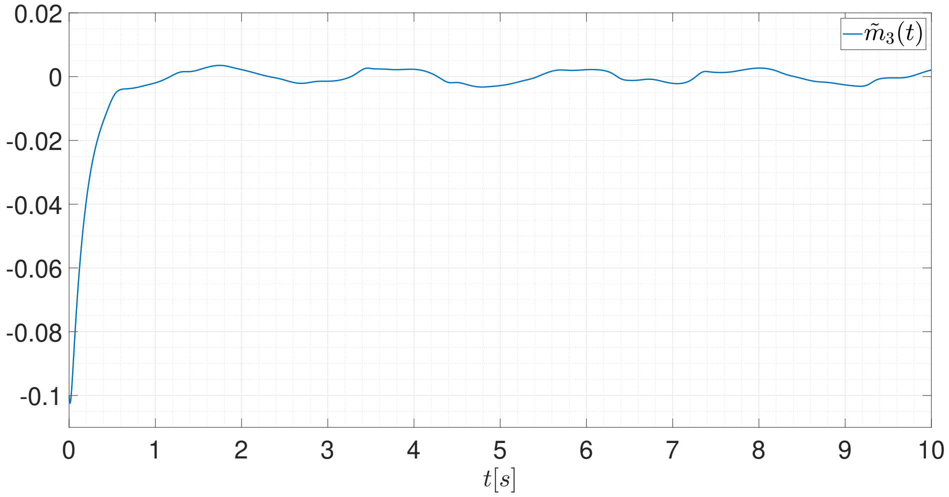

Figure 6.21 – Message error 2.

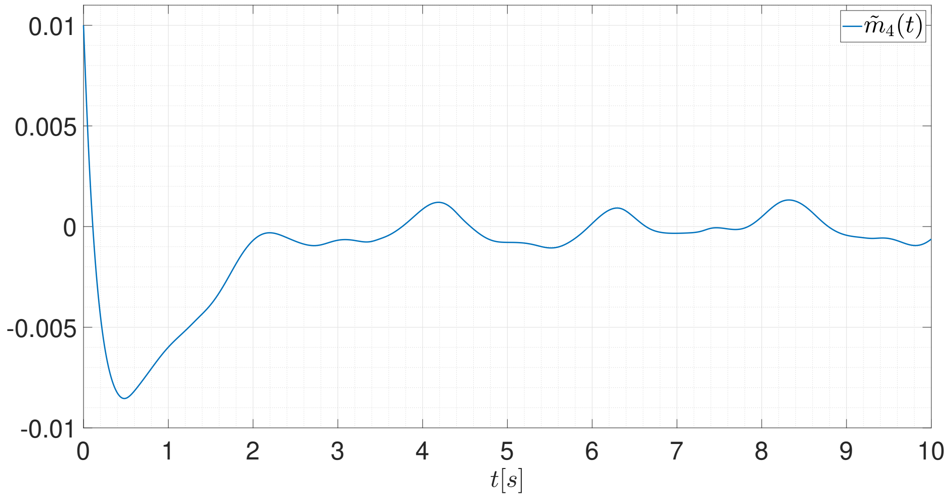

Figure 6.22 – Message error 3.

Based on Figures 6.3 - 6.22, observe that the proposed synchronization scheme is stable, practical, and easy to apply. On the other hand, it can also be applied to secure communication systems. However, the application performance depends on the power of the information signal and disturbances, which have to be sufficiently lower than the power of the chaotic carrier and message, respectively.

6.5 SUMMARY

A simple scheme for synchronization of a class of underactuated hyperchaotic systems has been proposed. Based on Lyapunov theory, a simple control law, which only acts on one state, has been devised to ensure the finite-time convergence of the synchronization error to a bounded region, despite the presence of bounded disturbances. Exhaustive simulations using Matlab ® have been accomplished to validate the theoretical results and show their application. Besides, the proposed methodology has been successfully applied to the encryption/decryption of messages.

7 SECURE COMMUNICATION BASED ON HYPERCHAOTIC UNDERACTUATED PROJECTIVE SYNCHRONIZATION

The research outcomes of this chapter have been submitted to the IEEE Access journal, and the paper was accepted for publication. This paper was done in joint work with other authors: Lucas Martins Alves, José Alfredo Ruiz Vargas, Sadek Crisóstomo Absi Alfaro, Guilherme Caribe de Carvalho, Jesus Franklin Andrade Romero. The name of the submitted work is "Secure Communication Based on Hyperchaotic Underactuated Projective Synchronization". *This chapter has extended and improved some parts compared to the original paper.*

In the past few years, significant progress has been made in the study of chaotic systems. Chaotic systems are deterministic nonlinear systems that show sensitive dependence on initial conditions and have an aperiodic behavior [8]. A necessary condition for a system to be chaotic lies in that at least one Lyapunov exponent be positive [36]. The chaotic motion on a strange attractor was first discovered in the 60s by Lorenz [145]. In the following years, some relevant chaotic models, such as those by Rössler [171], Chen [172], Sprott [173], and Lü [174], were introduced. More recently, many works have been proposed in the literature. See, for example, [146, 175–177].

On the other hand, a system can exhibit hyperchaos when at least two of their associated Lyapunov exponents are positive, and its dimension is higher than three [37]. Hyperchaos was initially introduced in 1979 by Rössler [178]. Since then, other important models have been proposed. Refer to [36, 155, 179–190] and the references therein to quote a few.

Chaotic and hyperchaotic systems have been used in most diverse contexts, including nonlinear identification [16, 74, 103], observation and control [191], [101], economy [192, 193], welding [194], [112], and secure communication [177, 187, 195–201]. In particular, chaos-based cryptography is a very active research topic in the literature, which is motivated by the pseudo-random behavior observed in chaotic systems. Typical applications encompass, for example, the generation of pseudo-random numbers for encryption and decryption of messages [202–206]. The main technologies used for implementation are analog electronics [184, 187, 207], field-programmable gate array (FPGA) [202, 208], microcontrollers [209, 210], and digital signal processing (DSP) [211]. It should be noted that chaos has a less complex and unpredictable behavior than hyperchaos, as claimed in [212]. Hence, the use of hyperchaos can sometimes be more suitable than chaos for secure communication [213, 214].

Synchronization lies in adjusting the dynamic behavior of two dynamic systems, known as master (drive) and slave (response), so that their trajectories converge in time. The synchronization of chaotic systems was first introduced in 1990 [39]. Several classes of synchronization have been proposed since then: antisynchronization (AS) [215], lag synchronization (LS) [148], projective synchronization (PS) [53, 181, 216–219], modified projective synchronization (MPS) [53], and function projective synchronization FPS [220]. In general, the synchronization type depends on a scaling factor. For instance, PS is characterized by a constant proportional synchronization between the master and slave systems. Hence, identical and AS are particular cases of this kind of synchronization with scaling factors 1 or -1. Matrix and functional scaling factors define MPS and FPS, respectively. However, most of the synchronization works above are only valid under either fully-actuated control or matching condition [221].

Also, interesting contributions have been proposed in [155, 169, 177, 182, 185, 189, 190, 216, 222, 223]. However, in these works, the usage of disturbances in the stability and convergence analysis was not considered.

In summary, the synchronization of either chaos or hyperchaos is characterized by two main hypotheses: the control dimension and system order are equal [53, 126, 146, 150, 155, 182, 185, 199, 219, 222] and unknowns are not considered in the stability analysis [155, 169, 177, 182, 185, 189, 190, 216, 222, 223]. The former is related to the complexity of the synchronization scheme and the latter to the robustness of the method. Synchronization of underactuated hyperchaotic systems is rarely found in the literature. Also, to the best of the author knowledge, the robust projective synchronization of underactuated hyperchaotic systems is not present in the literature. It should be noted that underactuation is a condition defined by a higher number of independent variables than control signals [224]. Thus, underactuated systems have fewer actuators than degrees of freedom [225–228], i.e., the control dimension is lesser than the state dimension. The main significance of the underactuated projective synchronization is the reduction of actuators in diverse applications.

Motivated by the previous facts, this chapter presents a robust scheme for projective synchronization of a hyperchaotic system to overcome the aforementioned drawbacks. Hence the proposed approach is based on both Lyapunov theory (to ensure boundedness and finite-time convergence) and underactuated control law (to simplify the application). More specifically, this chapter presents the following contributions.

1) An underactuated projective synchronization scheme for a perturbed Zhou et al. hyperchaotic system [185] is proposed. The proposed synchronization has a simple structure, in contrast to [53, 126, 146, 150, 155, 182, 185, 188, 189, 199, 212, 219, 229].

2) Neither matching condition nor fully-actuated control is assumed, i.e., the analysis

considers that disturbances are present in all states, even in those without actuation, and all states are not used in the proposed synchronization mechanism. In contrast to [155,169,177, 182,185,189,190,216,222,223].

3) The proposed scheme is applied to secure communication, and it was implemented using electrical circuits. In contrast to [53,126,150,155,166,182,185,189,190,219,229].

It should be noted that a simple structure leads to easier implementations. The consideration of disturbances in the stability analysis aim at robustness against disturbances which are inevitable in actual applications. To the best of author knowledge, this is the first time that a robust underactuated projective synchronization method for hyperchaotic systems is proposed in the literature.

The chapter is organized as follows. In Section 7.1, the problem and main assumptions are introduced. The synchronization error is presented in Section 7.2. In Section 7.3, a control law, which ensures that the synchronization errors are bounded and finite-time convergent, is proposed. Section 7.4 is concerned with application of the proposed method in secure telecommunication. In Section 7.5, the development of an electronic circuit for the implementation of the proposed method is accomplished, and a comparison study with another work in the literature is performed. Finally, the conclusions of the chapter are drawn in Section 7.6.

7.1 PROBLEM STATEMENT

Consider the following master hyperchaotic system [185]:

$$\begin{cases} \dot{x}_m = a(y_m - x_m) - w_m \\ \dot{y}_m = bx_m - x_m z_m - y_m \\ \dot{z}_m = x_m y_m - cz_m \\ \dot{w}_m = dx_m z_m - kw_m \end{cases} \tag{7.1}$$

Based on (7.1), a perturbed slave hyperchaotic system can be defined as

$$\begin{cases} \dot{x}_s = a(y_s - x_s) - w_s + h_1 + u_1 \\ \dot{y}_s = bx_s - x_s z_s - y_s + h_2 \\ \dot{z}_s = x_s y_s - cz_s + h_3 \\ \dot{w}_s = dx_s z_s - kw_s + h_4 + u_4 \end{cases} \tag{7.2}$$

where x_m, y_m, z_m, and w_m are the state variables of the master system (7.1); x_s, y_s, z_s, and

w_s are the state variables of the slave system (7.2); h_1, h_2, h_3, and h_4 are the disturbances; u_1 and u_4 are the control signals. The system parameters satisfy $a \in [10, 25]$, $b \in [10, 50]$, $c \in [1, 3]$, $d \in [1, 2]$, and $k \in [-1, 2]$ [185].

The aim of this chapter lies in the synchronization of (7.1) and (7.2) by using an underactuated control scheme, for any initial condition, even in the presence of unmatched disturbances.

REMARK 7.1.1 Important clarification of the problem of the system being underactuated is that the synchronizer only has access to information from the actuated master states, that is, from the states where there is a control signal. In other words, in the (7.2) system, the controls u_1 and u_4 cannot have the presence of the states y_m and z_m. However, the other states of the master system and all the states of the slave system are available.

FACT 7.1.1 In [185] was proved that (7.1) is hyperchaotic and that the system is dissipative. A consequence of the system being dissipative is also being bounded. With the boundedness of the system (7.1), the following inequalities are true:

$$\begin{aligned} |x_m(t)| &\leq \bar{x} \\ |y_m(t)| &\leq \bar{y} \\ |z_m(t)| &\leq \bar{z} \\ |w_m(t)| &\leq \bar{w} \end{aligned} \tag{7.3}$$

$\forall t \geq 0$, where $\bar{x}$, $\bar{y}$, $\bar{z}$, and $\bar{w}$ are unknown positive constants.

ASSUMPTION 7.1.1 It is assumed that the disturbances in (7.2) are bounded. More specifically,

$$\begin{aligned} |h_1(t)| &\leq \bar{h}_1 \\ |h_2(t)| &\leq \bar{h}_2 \\ |h_3(t)| &\leq \bar{h}_3 \\ |h_4(t)| &\leq \bar{h}_4 \end{aligned} \tag{7.4}$$

$\forall t \geq 0$, where $\bar{h}_1$, $\bar{h}_2$, $\bar{h}_3$, and $\bar{h}_4$ are unknown positive constants.

REMARK 7.1.2 It is noteworthy that systems (7.1) and (7.2) are, in general, different, due to the presence of disturbances. Besides, the chapter approach assumes that all these disturbances are not in the control span. Hence, the control signal can not be used straightforwardly to tackle with them. However, this chapter exploits the master boundedness and the particular structure of (7.1) - (7.2), in the next sections, to devise a simple and robust control law.

REMARK 7.1.3 Note that Equation (7.1) does not have a disturbance term. That is no loss of generality because any disturbance in (7.1) would join with those in (7.2) in the stability analysis.

7.2 PROJECTIVE SYNCHRONIZATION ERROR

In this section, the main error associated with the synchronization problem is defined.

The projective synchronization error is defined as

$$\begin{aligned} e_1 &= x_s - \delta x_m \\ e_2 &= y_s - \delta y_m \\ e_3 &= z_s - \delta z_m \\ e_4 &= w_s - \delta w_m \end{aligned} \tag{7.5}$$

where δ is a nonzero constant defined by the user.

Based on (7.1) - (7.2), the time-derivative of (7.5) results

$$\begin{aligned} \dot{e}_1 &= -ae_1 + ae_2 - e_4 + h_1 + u_1 \\ \dot{e}_2 &= -e_2 + be_1 - e_1e_3 - \delta z_m e_1 - \delta x_m e_3 - (\delta^2 - \delta)x_m z_m + h_2 \\ \dot{e}_3 &= -ce_3 + e_1e_2 + \delta y_m e_1 + \delta x_m e_2 + (\delta^2 - \delta)x_m y_m + h_3 \\ \dot{e}_4 &= -ke_4 + de_1e_3 + d\delta z_m e_1 + d\delta x_m e_3 + (\delta^2 - \delta)dx_m z_m + h_4 + u_4 \end{aligned} \tag{7.6}$$

7.3 LYAPUNOV STABILITY ANALYSIS

After the formulation of the synchronization error equations, the next step is selecting a control law. In what follows, this chapter considers a standard Lyapunov function candidate V, which hinges upon the synchronization error, and chooses a control law to make $\dot{V}$ lower than zero outside a compact region at the origin Ω. The key drivers for the design are then the peculiar structure of (7.2), boundedness of (7.1), and an enlargement process for constraining $\dot{V}$ to be negative definite outside Ω

Theorem 1: Consider the master and slave systems (7.1) - (7.1), Assumption 3.2.1, and the control laws

$$\begin{aligned} u_1 &= -\psi_1 e_1 - \psi_2 e_1 e_4^2 \\ u_4 &= -\psi_3 e_4 \end{aligned} \tag{7.7}$$

If,

$$\begin{aligned} \psi_1 &> \delta_1 \\ \psi_2 &> \frac{d^2}{2} \\ \psi_3 &> \delta_2 \end{aligned} \tag{7.8}$$

Then, the synchronization error is ultimately bounded. It converges in finite-time to the compact set

$$\Omega = \left\{ e \in \Re^4 \mid \ \|e\| \leq \theta < \xi \right\} \tag{7.9}$$

where the convergence time holds

$$t_{max} = \begin{cases} 0 & \text{if } e\left(t\right) \in \Omega \\ \dfrac{1}{\rho} ln \left[\dfrac{V\left(0\right)}{\frac{\xi^2}{2} - \frac{\beta}{\rho}} \right] & \text{otherwise} \end{cases} \tag{7.10}$$

and $\delta_1 = 0.5[\sigma_1 \bar{h}_1^2 + \sigma_5(a^2 + b^2 + \delta^2 \bar{z}^2) + \sigma_6 \delta^2 \bar{y}^2 + \sigma_7(d\delta\bar{x} + 1)] - a$, $\delta_2 = 0.5\Big(\sigma_4 \bar{h}_4^2 + \frac{d\delta\bar{x}+1}{\sigma_7} + \frac{d^2\delta^2\bar{x}^2}{\sigma_8} + \sigma_{11}\Big) - k$, $\theta = \sqrt{\frac{2\beta}{\rho}}$, $\xi > 0$, $\beta = \beta_u + \beta_n + \beta_\delta$, $\frac{\rho}{2} = min\left\{\rho_1, \rho_2, \rho_3, \rho_4\right\}$, $\beta_u = 0.5\left(\frac{1}{\sigma_1} + \frac{1}{\sigma_4}\right)$, $\beta_n = 0.5\left(\frac{\bar{h}_2^2}{\sigma_2} + \frac{\bar{h}_3^2}{\sigma_3}\right)$, $\beta_\delta = 0.5\left(\delta^2 - \delta\right)^2\left(\frac{\bar{x}^2\bar{z}^2}{\sigma_9} + \frac{\bar{x}^2\bar{y}^2}{\sigma_{10}} + \frac{\bar{x}^2\bar{z}^2}{\sigma_{11}}\right)$, $\rho_1 = \psi_1 - \delta_1$, $\rho_2 = 1 - 0.5\left(\sigma_2 + \frac{1}{\sigma_5} + \sigma_9\right)$, $\rho_3 = c - 0.5\left(\sigma_3 + \frac{1}{\sigma_6} + \sigma_8 + \sigma_{10} + 1\right)$, $\rho_4 = \psi_3 - \delta_2$, $\|e\|^2 = e_1^2 + e_2^2 + e_3^2 + e_4^2$, and $\sigma_i, i = 1, ..., 11$ are positive constants.

Proof.

Consider the following Lyapunov function candidate

$$V = \frac{\|e\|^2}{2} \tag{7.11}$$

The time-derivative of (7.11) results

$$\dot{V} = e_1\dot{e}_1 + e_2\dot{e}_2 + e_3\dot{e}_3 + e_4\dot{e}_4 \tag{7.12}$$

By replacing (7.6) in (7.12) implies

$$\begin{aligned}\dot{V} &= e_1(-ae_1 + ae_2 - e_4 + h_1 + u_1) \\ &\quad + e_2[-e_2 + be_1 - e_1e_3 - \delta z_m e_1 - \delta x_m e_3 - (\delta^2 - \delta)x_m z_m + h_2] \\ &\quad + e_3[-ce_3 + e_1e_2 + \delta y_m e_1 + \delta x_m e_2 + (\delta^2 - \delta)x_m y_m + h_3] \\ &\quad + e_4[-ke_4 + de_1e_3 + d\delta z_m e_1 + d\delta x_m e_3 + (\delta^2 - \delta)dx_m z_m + h_4 + u_4]\end{aligned} \tag{7.13}$$

Also, by employing (7.7) in (7.13), then

$$\begin{aligned}\dot{V} &= -(\psi_1 + a)e_1^2 - e_2^2 - ce_3^2 - (\psi_3 + k)e_4^2 - \psi_2 e_1^2 e_4^2 + h_1e_1 + h_2e_2 + h_3e_3 \\ &\quad + h_4e_4 + e_1e_2(a + b - \delta z_m) + \delta y_m e_1 e_3 + e_1e_4(d\delta z_m - 1) + d\delta x_m e_3 e_4 \\ &\quad - (\delta^2 - \delta)x_m z_m e_2 + (\delta^2 - \delta)x_m y_m e_3 + (\delta^2 - \delta)x_m z_m e_4 + de_1e_3e_4\end{aligned} \tag{7.14}$$

On the other hand, from Fact 7.1.1, Assumption 3.2.1, and Young's inequality, it follows that

$$\begin{aligned} & e_1h_1 \leq \frac{\sigma_1 e_1^2 \bar{h}_1^2}{2} + \frac{1}{2\sigma_1}; e_2h_2 \leq \frac{\sigma_2 e_2^2}{2} + \frac{\bar{h}_2^2}{2\sigma_2} \\ & e_3h_3 \leq \frac{\sigma_3 e_3^2}{2} + \frac{\bar{h}_3^2}{2\sigma_3}; e_4h_4 \leq \frac{\sigma_4 e_4^2 \bar{h}_4^2}{2} + \frac{1}{2\sigma_4} \\ & e_1e_2(a + b - \delta z_m) \leq \frac{\sigma_5(a^2 + b^2 + \delta^2 \bar{z}^2)e_1^2}{2} + \frac{e_2^2}{2\sigma_5} \\ & \delta y_m e_1 e_3 \leq \frac{\sigma_6 \delta^2 \bar{y}^2 e_1^2}{2} + \frac{e_3^2}{2\sigma_6} \\ & e_1e_4(d\delta x_m - 1) \leq \frac{\sigma_7 e_1^2(d\delta\bar{x} + 1)}{2} + \frac{e_4^2(d\delta\bar{x} + 1)}{2\sigma_7} \\ & d\delta x_m e_3 e_4 \leq \frac{\sigma_8 e_3^2}{2} + \frac{d^2\delta^2\bar{x}^2 e_4^2}{2\sigma_8} \\ & -(\delta^2 - \delta)x_m z_m e_2 \leq \frac{\sigma_9 e_2^2}{2} + \frac{(\delta^2 - \delta)^2\bar{x}^2\bar{z}^2}{2\sigma_9} \\ & (\delta^2 - \delta)x_m y_m e_3 \leq \frac{\sigma_{10} e_3^2}{2} + \frac{(\delta^2 - \delta)^2\bar{x}^2\bar{y}^2}{2\sigma_{10}} \\ & (\delta^2 - \delta)x_m z_m e_4 \leq \frac{\sigma_{11} e_4^2}{2} + \frac{(\delta^2 - \delta)^2\bar{x}^2\bar{z}^2}{2\sigma_{11}} \\ & de_1e_3e_4 \leq \frac{e_3^2}{2} + \frac{d^2e_1^2e_4^2}{2}\end{aligned} \tag{7.15}$$

Based on (7.15), and analyzing when $\dot{V} \leq 0$, (7.14) implies

$$\begin{aligned}\dot{V} \leq & -e_1^2(\psi_1 - \delta_1) - e_2^2\rho_2 - e_3^2\rho_3 - e_4^2(\psi_3 - \delta_2) + \beta_u \\ & + \beta_n + \beta_\delta - \left(\psi_2 - \frac{d^2}{2}\right) e_1^2 e_4^2\end{aligned} \tag{7.16}$$

By using (7.8), (7.16) can be written as

$$\dot{V} \leq -\rho V + \beta \tag{7.17}$$

Then, one can see that $\dot{V} < 0$ in (7.17) when $e \in \Omega^C$. Since Ω is a compact set, the errors starting inside Ω will remain there forever. In case that the errors start outside Ω, it can be seen that $\dot{V} < 0$, then V and, consequently, $\|e\|$ will decrease monotonically until the errors enter Ω at some finite time t_{max}.

To determine t_{max}, it should be noted that (7.17) implies ([24], Lemma 3.2.4)

$$V(t) \leq \left[V(0) - \frac{\beta}{\rho}\right] \exp(-\rho t) + \frac{\beta}{\rho} \tag{7.18}$$

which further yields

$$t_{max} = \frac{1}{\rho} ln \left\{ \frac{V(0)}{\frac{\xi^2}{2} - \frac{\beta}{\rho}} \right\} \tag{7.19}$$

Therefore, (7.19) shows that the synchronization error converges to the compact set Ω at least in a finite-time t_{max} and, hence ([230]), that the this error is uniformly ultimately bounded and converge to a ball with radius θ (Figure 7.1). □

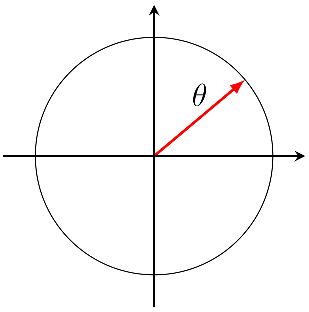

Figure 7.1 – Bounded set.

REMARK 7.3.1 The main idea of the proposed method lies in the usage of the system structure and Lyapunov theory to design an adequate control law. Based on a trial-and-error procedure, all possibilities of underactuated control in the analysis are considered, and the simplest one was chosen. Inequality of Young was used in the stability analysis in the process to make $\dot{V} < 0$ outside of a small compact set, whose size can be decreased by increasing the control gains, even in the presence of bounded unmatched disturbances.

Remark 7.3.2 Integrator backstepping [231,232] and sliding mode [190,233] are also used in the control of underactuated systems. However, most works based on backstepping suffer from "the explosion of complexity," and, in general, the presence of matched disturbances is assumed. See, for example, [231, 232]. On the other hand, sliding mode control suffers from chattering, and the disturbances are also supposed to be matched [232]. Besides, most of synchronizers based on hyperchaotic systems found in the literature employ complete actuation [175, 212, 234, 235]. Then, the main peculiarity of this chapter, in contrast to the literature, thus lies in that neither matching condition nor fully-actuated control is assumed.

Remark 7.3.3 By replacing the parameters in (7.8) and considering the case of identical synchronization ($\delta = 1$), (7.8) can be rewritten as

$$\begin{aligned}
&\psi_1 > 0.5[\sigma_1 \bar{h}_1^2 + \sigma_5(400 + 1024 + \bar{z}^2) + \sigma_6 \bar{y}^2 \\
&\quad + \sigma_7(\bar{x} + 1)] - 20 \\
&\psi_2 > 0.5 \\
&\psi_3 > 0.5\left(\sigma_4 \bar{h}_4^2 + \frac{\bar{x} + 1}{\sigma_7} + \frac{\bar{x}^2}{\sigma_8} + \sigma_{11}\right) + 1
\end{aligned} \tag{7.20}$$

In addition, one can select conservative values for the bounds and other parameters as $\bar{x} = 27$, $\bar{y} = 38$, $\bar{z} = 63$, $\bar{w} = 240$, $\bar{h}_1 = 2.7$, $\bar{h}_2 = 3.8$, $\bar{h}_3 = 6.3$, $\bar{h}_4 = 24$, $\sigma_1 = 1$, $\sigma_4 = \frac{1}{2}$, $\sigma_5 = 1$, $\sigma_6 = 1$, $\sigma_7 = 1$, $\sigma_8 = 1$, and $\sigma_{11} = \frac{1}{2}$. Then $\psi_1 > 3429.29$ and $\psi_3 > 523.75$. It should be noted that high gains do not prevent application since these gains can be decreased by amplitude scaling of the hyperchaotic systems leading to lower bounds for the states, as can be seen in Section 7.5.

Remark 7.3.4 Defining $D_i (i = 1, ..., 11)$ as the respective domains of $\sigma_i (i = 1, ..., 11)$, results

$$
\begin{aligned}
D_1 &= \left\{ \sigma_1 \in \Re | 0 < \sigma_1 < \frac{2(\psi_1 + a)}{\bar{h}_1^2} \right\} \\
D_2 &= \{ \sigma_2 \in \Re | 0 < \sigma_2 < 2 \} \\
D_3 &= \{ \sigma_3 \in \Re | 0 < \sigma_3 < 2c \} \\
D_4 &= \left\{ \sigma_4 \in \Re | 0 < \sigma_4 < \frac{2(\psi_3 + k)}{\bar{h}_4^2} \right\} \\
D_5 &= \left\{ \sigma_5 \in \Re | \frac{1}{2} < \sigma_5 < \frac{2(\psi + a)}{a^2 + b^2 + \delta^2 \bar{z}^2} \right\} \\
D_6 &= \left\{ \sigma_6 \in \Re | \frac{1}{2c} < \sigma_6 < \frac{\psi_1}{\delta^2 \bar{y}^2} \right\} \\
D_7 &= \left\{ \sigma_7 \in \Re | \frac{2\psi_3}{d\delta\bar{x} + 1} < \sigma_7 < \frac{\psi_1}{d\delta\bar{x} + 1} \right\} \\
D_8 &= \left\{ \sigma_8 \in \Re | \frac{2\psi_3}{d^2\delta^2\bar{x}^2} < \sigma_8 < 2c - 1 \right\} \\
D_9 &= \left\{ \sigma_9 \in \Re | 0 < \sigma_9 < \frac{1}{2} \right\} \\
D_{10} &= \{ \sigma_{10} \in \Re | 0 < \sigma_{10} < 2c - 1 \} \\
D_{11} &= \{ \sigma_{11} \in \Re | 0 < \sigma_{11} < 2\psi_3 \}
\end{aligned}
\tag{7.21}
$$

Note that the performance of the proposed method, as far as the residual synchronization error is considered, is affected by the control gains ψ_1 and ψ_3, scaling factor δ, disturbances, and upper bounds for the states of the master system, as can be seen from (7.8) and the definitions below (7.10). In general, the performance for the actuated states can be arbitrarily enhanced by increasing ψ_1 and ψ_3. For non-actuated states, although there is a complex relationship between β and ρ in (7.9), in some cases, the performance can also be indirectly controlled through ψ_1 and ψ_3 since higher control gains can allow for higher ρ in (7.9), as can be seen from the stability analysis.

7.4 CHAOS-BASED SECURE COMMUNICATION

In addition to the projective synchronization case, the application of the proposed method to secure telecommunication is also considered. To have a well-posed problem, the following assumption must be imposed.

Assumption 7.4.1 It is assumed that the messages are bounded. More specifically,

$$|m_i(t)| \leq \bar{m}_i, i = 1, ..., 4 \tag{7.22}$$

$\forall t \geq 0$, where m_1, m_2, m_3, and m_4 are the original messages and $\bar{m}_1$, $\bar{m}_2$, $\bar{m}_3$, and $\bar{m}_4$ are positive constants.

Further, motivated by [7], it can be defined

$$\begin{aligned} \hat{m}_1 &= \delta s_1 - x_s \\ \hat{m}_2 &= \delta s_2 - y_s \\ \hat{m}_3 &= \delta s_3 - z_s \\ \hat{m}_4 &= \delta s_4 - w_s \end{aligned} \tag{7.23}$$

where $s_1 = x_m + m_1$, $s_2 = y_m + m_2$, $s_3 = z_m + m_3$, and $s_4 = w_m + m_4$ are the encoded messages; and $\hat{m}_1$, $\hat{m}_2$, $\hat{m}_3$, and $\hat{m}_4$ are the decoded messages.

On the other hand, by using (7.23) and defining $\tilde{m}_i = \hat{m}_i - \delta m_i, i = 1, ..., 4$, where $\tilde{m}_1$, $\tilde{m}_2$, $\tilde{m}_3$, and $\tilde{m}_4$ are the message errors, it can be concluded that

$$\tilde{m}_i = -e_i, i = 1, ..., 4 \tag{7.24}$$

Remark 7.4.1 It is worth noticing that the quality of the message reconstruction is the same as the synchronization, as shown in (7.24). Furthermore, the boundedness of the message error is assured when the synchronization error is bounded.

Remark 7.4.2 An overview of the secure communication scheme can be seen in Figure 7.2.

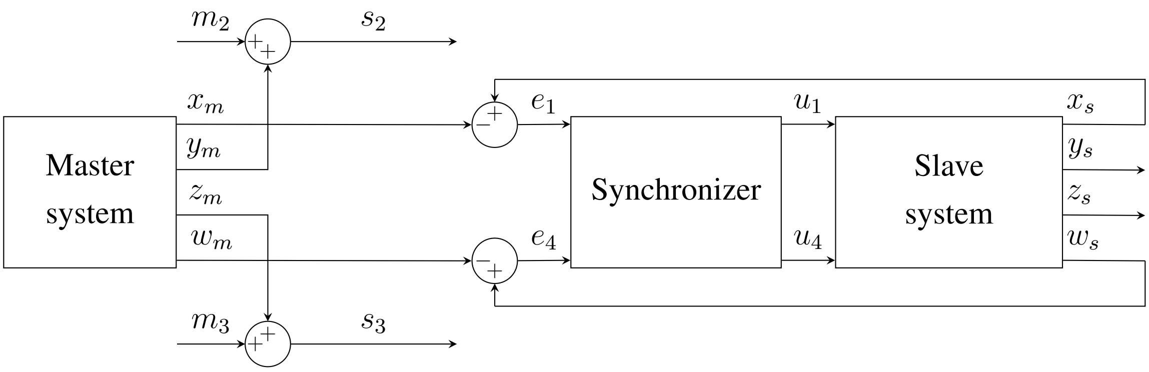

Figure 7.2 – Synchronization and secure communication scheme.

7.5 SIMULATIONS

Simulations were performed using Matlab/Simulink 2020b® on a Windows 10 platform, with AMD Ryzen 7 1700 processor for all simulations, variable-step algorithm ODE15s solver, and relative tolerance of 10^{-8}. All the scripts to reproduce the results of this chapter are available with the author under request. In all simulations, it was considered that $\delta = 1$.

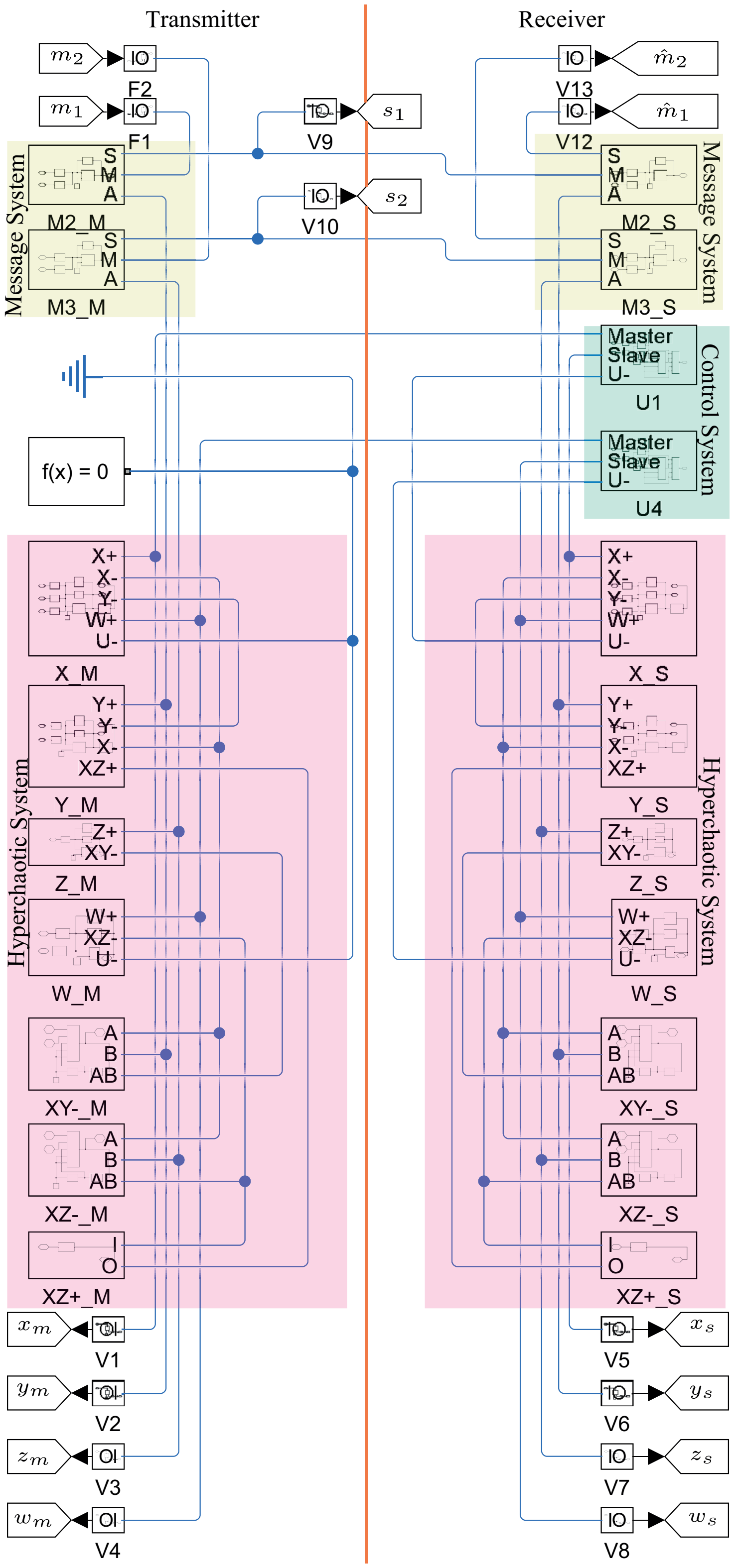

Figure 7.3 – Circuit diagram. M2_M, M3_M, M2_S, M3_S are the encryption/ decryption blocks (Figure 7.6). U1 and U4 are generated by the CONTROL block (Figure 7.3). X_M and X_S are the X state blocks (Figure 7.7). Y_M and Y_S are the Y state blocks (Figure 7.8). Z_M and Z_S are the Z state blocks (Figure 7.9). W_M and W_S are the W state blocks (Figure 7.10). +XZ_M and +XZ_S are the inversion blocks (Figure 7.4). -XZ_M, -XZ_S,-XY_M, and -XY_S are the multiplication blocks (Figure 7.5). V1 to V12 are voltage to signal converters and the S1 and S2 are signal to voltage converters of the Simulink.

7.5.1 Implementation Example

For implementation purposes, consider that $a = 10$, $b = 47$, $c = 1$, $d = 1$, and $k = 2$. It was also necessary to scale both the frequency and amplitude of the hyperchaotic systems (7.1) - (7.2) to decrease the transient and hold the operating voltage of the devices. The system amplitude was then decreased up to 20 times, and the system rate was increased by a factor of 1000. Hence, the scaled systems were rewritten in a condensed form as

$$\begin{cases} \dot{X} = 10^3[10(Y - X) - X + I_* u_1] + h_1 \\ \dot{Y} = 10^3[47X - 20XZ - Y] + h_2 \\ \dot{Z} = 10^3[20XY - Z] + h_3 \\ \dot{W} = 10^3[20XZ - 2W + I_* u_4] + h_4 \end{cases} \tag{7.25}$$

where $I_* = \{0, 1\}$, u_1 and u_4 are defined as in (7.7) being $\psi_1 = \psi_3 = 100$ and $\psi_2 = 10000$. Figure 7.3 shows the circuit diagram designed by using Simscape/Simulink being their blocks detailed in Figures 7.4 - 7.11.

For the sake of conciseness, (7.25) was considered as being the transmitter when $I_* = 0$ (X, Y, Z, and W must be substituted by X_m, Y_m, Z_m, and W_m) and the receiver when $I_* = 1$ (in which case X, Y, Z, and W must be substituted by X_s, Y_s, Z_s, and W_s).

The encrypted signals are defined as $s_2 = m_2 + Y_m$, $s_3 = m_3 + Z_m$ and the decrypted signals as $\hat{m}_2 = s_2 - Y_s$, $\hat{m}_3 = s_3 - Z_s$. This definition was adopted to use the same circuital structure for simple encryption and decryption. The considered messages were m_2 as being a sequence of bits and $m_3 = a_0 + 0.5 \cdot \sum_{i=1}^{8}[a_i \cdot \cos(i \cdot w \cdot t) + b_i \cdot \sin(i \cdot w \cdot t)]$, where $w = 941.7$, $a_0 = -0.1009055$, $a_1 = 0.09614$, $b_1 = -0.08111$, $a_2 = -0.002126$, $b_2 = -0.002561$, $a_3 = 0.01418$, $b_3 = 0.03685$, $a_4 = 0.0004152$, $b_4 = -0.002264$, $a_5 = -0.01999$, $b_5 = -0.0121$, $a_6 = 7.883 \cdot 10^{-5}$, $b_6 = -0.0008134$, $a_7 = 0.002927$, $b_7 = -0.0009047$, $a_8 = -3.586 \cdot 10^{-5}$, and $b_8 = -0.0008205$.

In all simulations, analog multipliers AD633JNZ and operational amplifiers OPA228 are used. Nominal voltage limits, slew rate, bandwidth, input impedance, output impedance, offset voltage, and polarization current from their datasheets were also considered. Further, all capacitors and resistors used here were non-ideal with a tolerance of 0.1% to better reproduce a more realistic scenario.

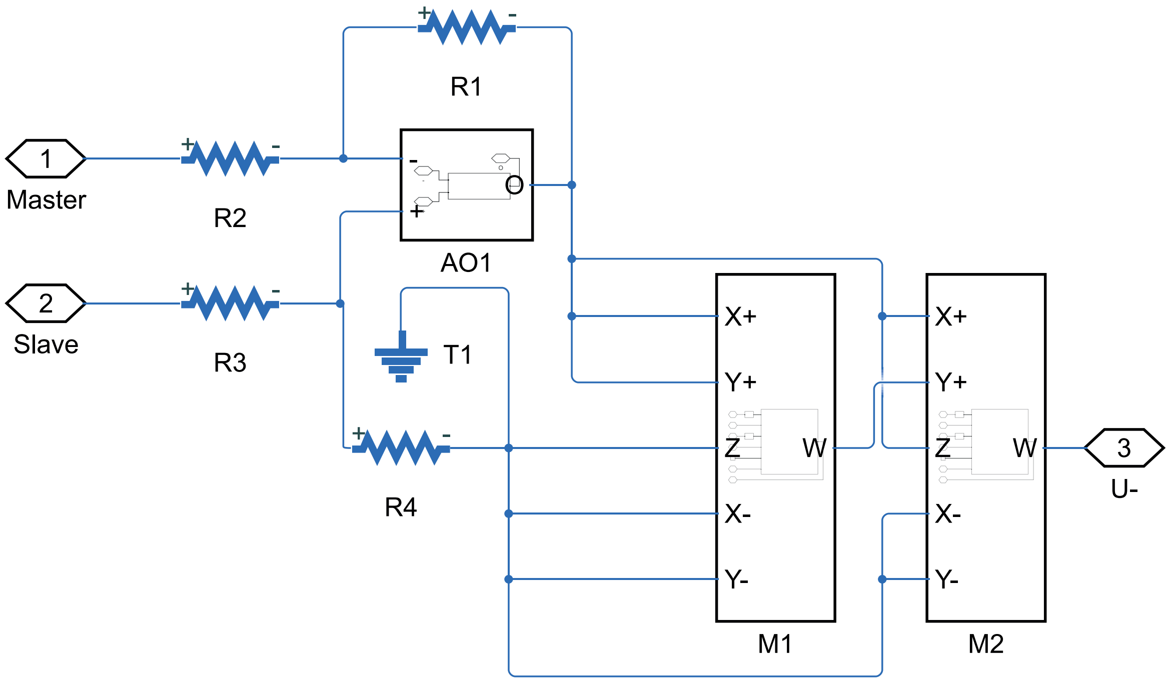

Figure 7.4 – CONTROL block circuit. R1 = R4 = R5 = R8 = 100kΩ and R2 = R3 = R6 = R7 = 1kΩ with tolerance of 0.1%. The blocks OA1 and OA2 are operational amplifiers OPA228. The blocks M1 and M2 are analog multipliers AD633JNZ.

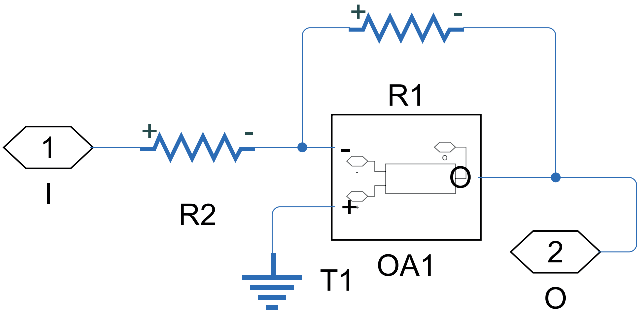

Figure 7.5 – Inverter block circuit. R1 = R2 = 10kΩ with tolerance of 0.1%. The block OA1 is an operational amplifier OPA228.

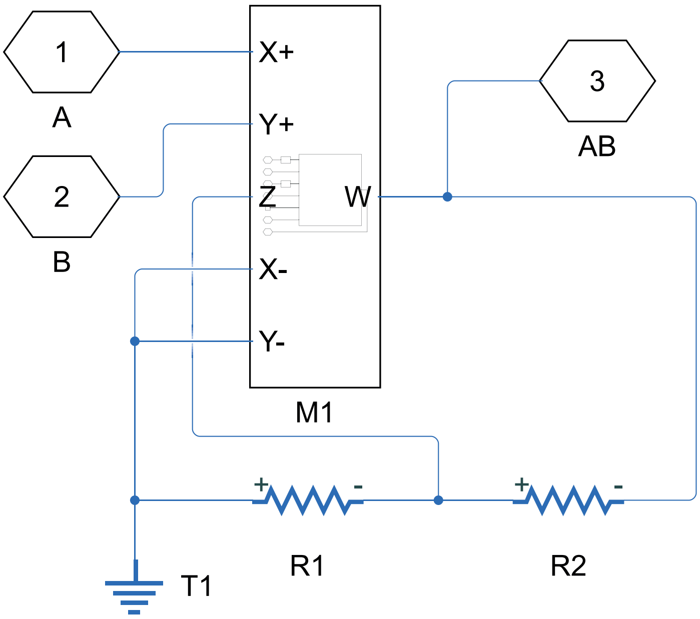

Figure 7.6 – Multiplier block circuit. R1 = R2 = 10kΩ with tolerance of 0.1%. The block M1 is an analog multiplier AD633JNZ.

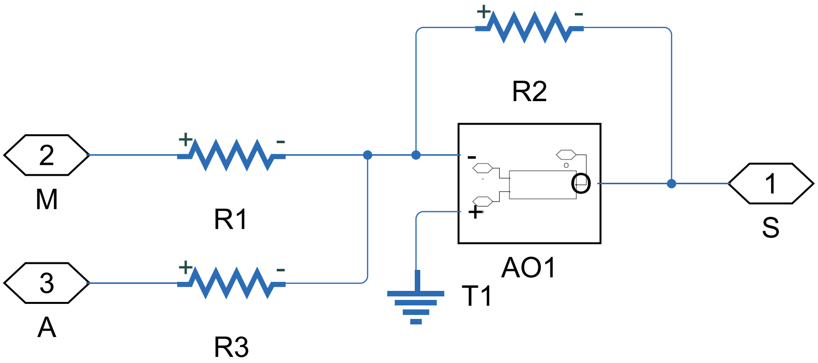

Figure 7.7 – Encryption/decryption block circuit. R1 = R2 = R3 = 100kΩ with tolerance of 0.1%. The block OA1 is an operational amplifier OPA228.

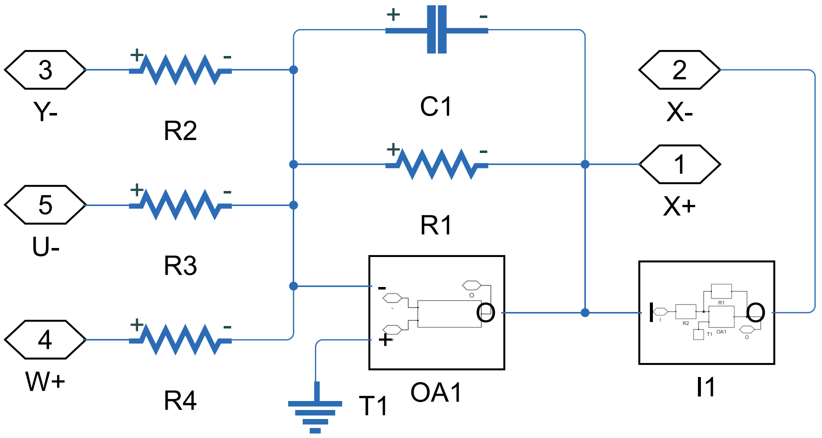

Figure 7.8 – State X block circuit. C1 = 10nF, R1 = R2 = 10kΩ, and R3 = R4 = 100kΩ. Resistors and capacitors have a tolerance of 0.1%, OA1 is an operational amplifier OPA228 and I1 is an inversion block (Figure 7.5).

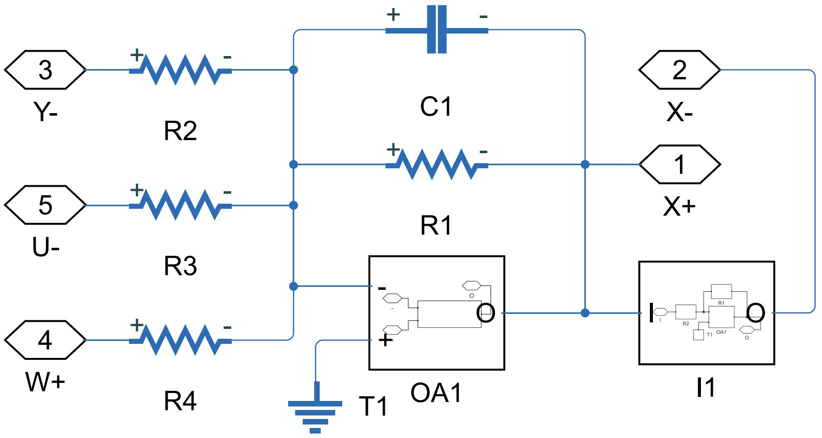

Figure 7.9 – State Y block circuit. C1 = 10nF, R1 = 100kΩ, R2 = 2.1kΩ, and R3 = 1kΩ. Resistors and capacitors have a tolerance of 0.1%, OA1 is an operational amplifier OPA228 and I1 is an inversion block (Figure 7.5).

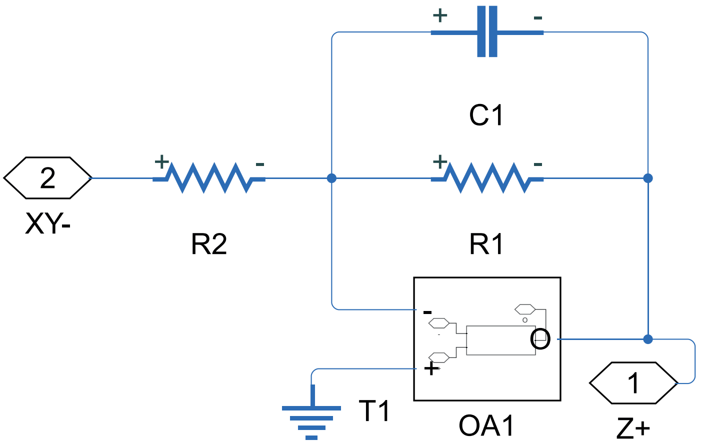

Figure 7.10 – State Z block circuit. C1 = 10nF, R1 = 100kΩ, and R2 = 1kΩ. Resistors and capacitors have a tolerance of 0.1% and OA1 is an operational amplifier OPA228.

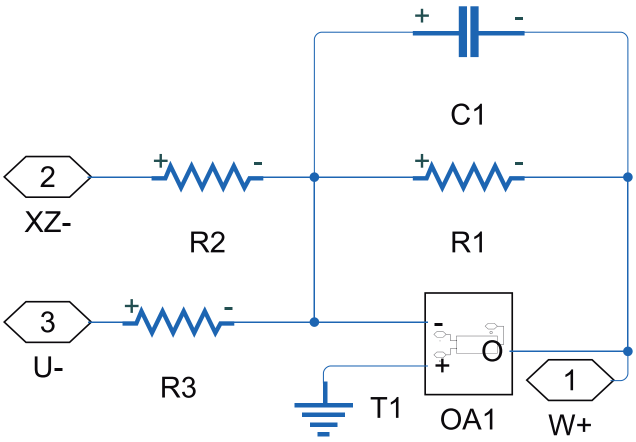

Figure 7.11 – State W block circuit. C1 = 10nF, R1 = 50kΩ, R2 = 1kΩ, and R3 = 100kΩ. Resistors and capacitors have a tolerance of 0.1% and OA1 is an operational amplifier OPA228.

Figure 7.12 - 7.19 depicts a typical performance of the synchronization using electronic circuits, and Table 7.1 shows the root mean square of synchronization errors. The synchronization errors are small even when underactuated control and unmatched perturbations resulting from the non-ideal behavior of the devices are considered. Based on the simulations, the proposed method is easily implementable by analog electronics and has good performance, as far as the synchronization error is considered.

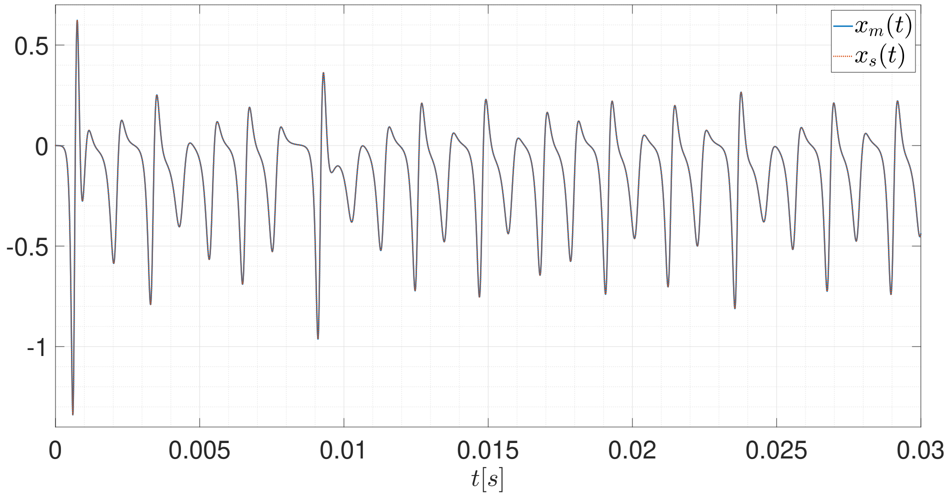

Figure 7.12 – Synchronization performance of $x_m(t)$ and $x_s(t)$ using Matlab. The states are in volts.

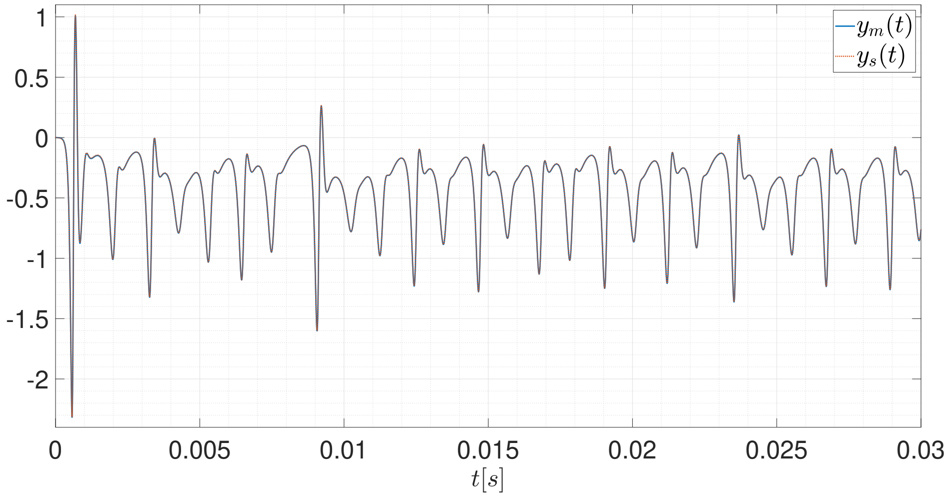

Figure 7.13 – Synchronization performance of $y_m(t)$ and $y_s(t)$ using Matlab. The states are in volts.

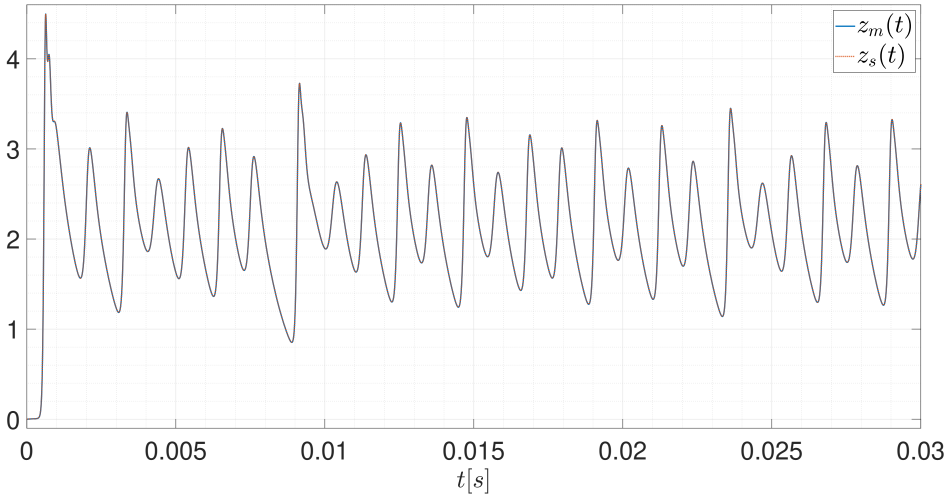

Figure 7.14 – Synchronization performance of $z_m(t)$ and $z_s(t)$ using Matlab. The states are in volts.

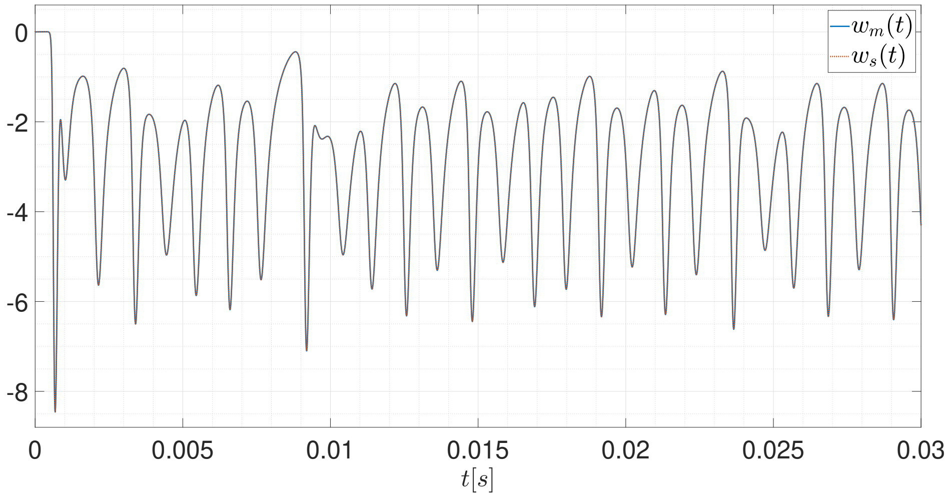

Figure 7.15 – Synchronization performance of $w_m(t)$ and $w_s(t)$ using Matlab. The states are in volts.

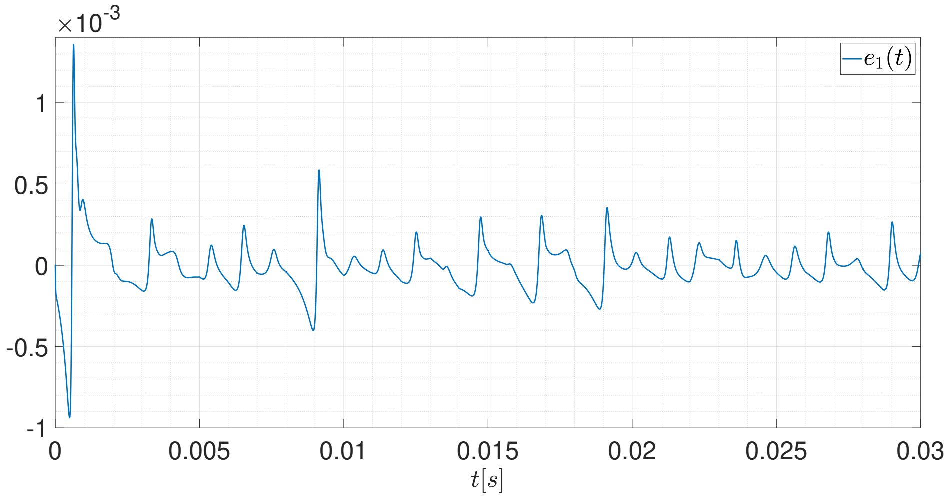

Figure 7.16 – Synchronization error of $x(t)$. The error is in volts.

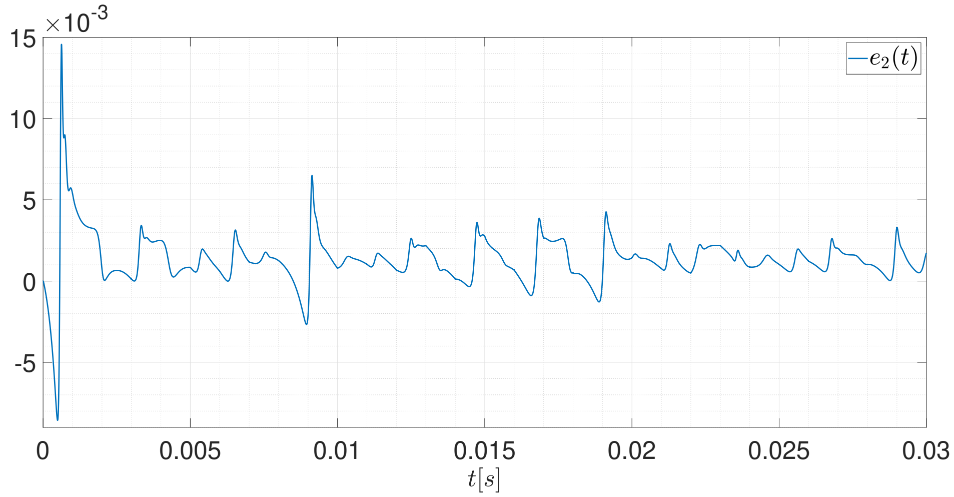

Figure 7.17 – Synchronization error of $y(t)$. The error is in volts.

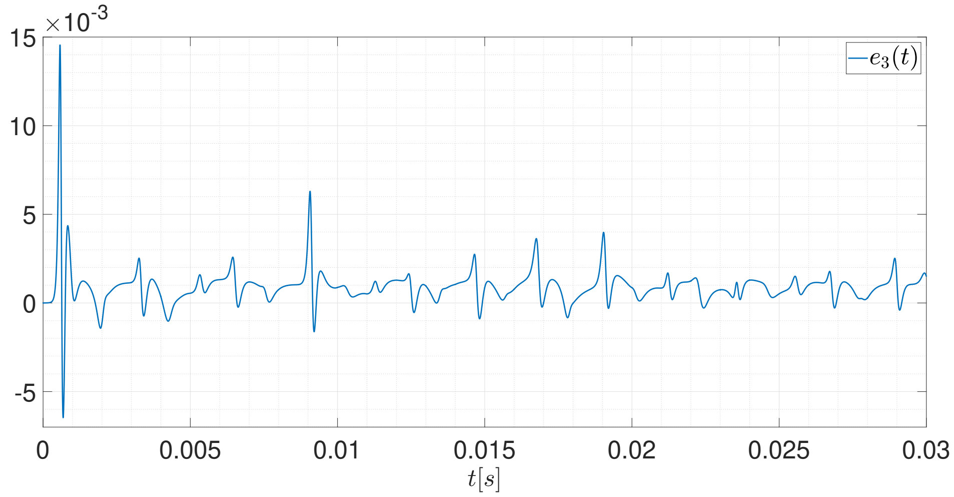

Figure 7.18 – Synchronization error of $z(t)$. The error is in volts.

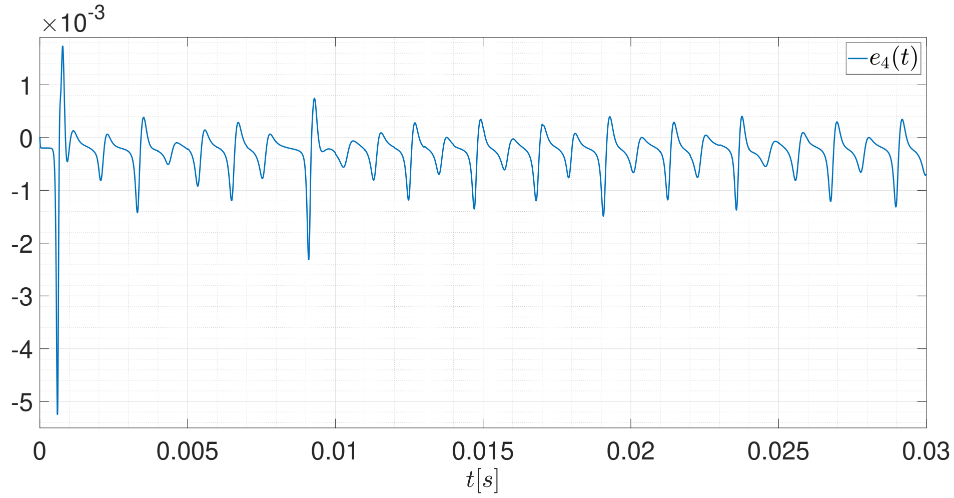

Figure 7.19 – Synchronization error of $w(t)$. The error is in volts.

Table 7.1 – Root mean square of synchronization errors for t = [0 0.03] seconds.

Root Mean Square of Synchronization Errors in the Proposed Algorithm				
$e_{1_{rms}}$	$e_{2_{rms}}$	$e_{3_{rms}}$	$e_{4_{rms}}$	e_{rms}
0.000198	0.002452	0.001767	0.000632	0.002527

Figure 7.20 - 7.23 shows the performance of the encryption of two message signals using the proposed synchronization method. As expected from theoretical analysis, the recovered signals converge over time close to the original ones, even in the presence of internal or external perturbations, which are due to the usage of non-ideal devices in a realistic scenario in the simulations.

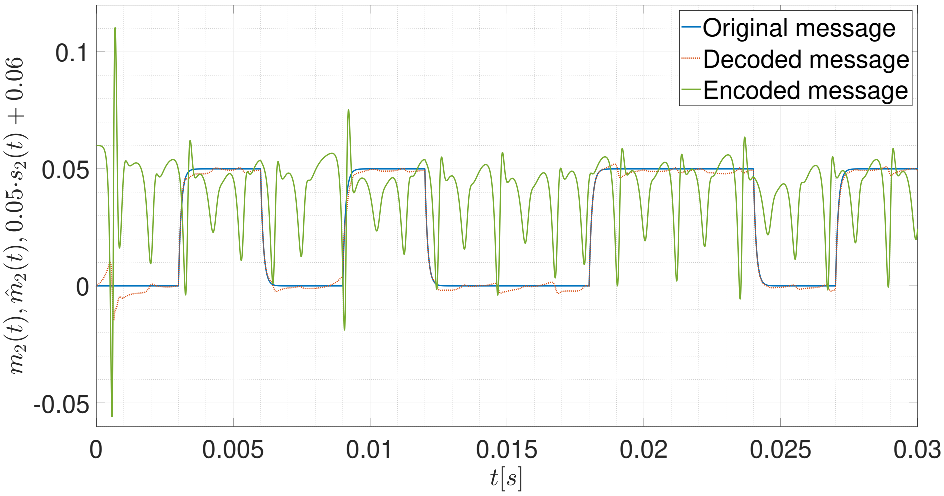

Figure 7.20 – Performance of the secure communication of the second state. All signals are in volts.

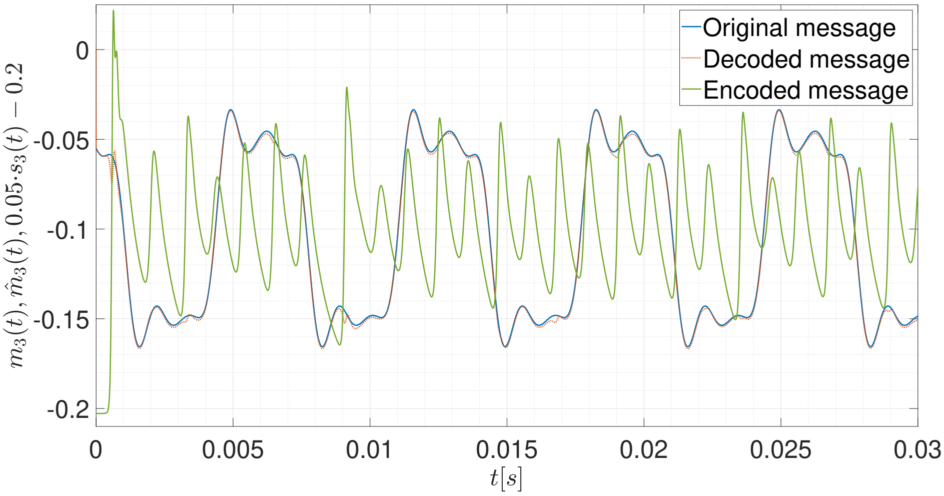

Figure 7.21 – Performance of the secure communication of the third state. All signals are in volts.

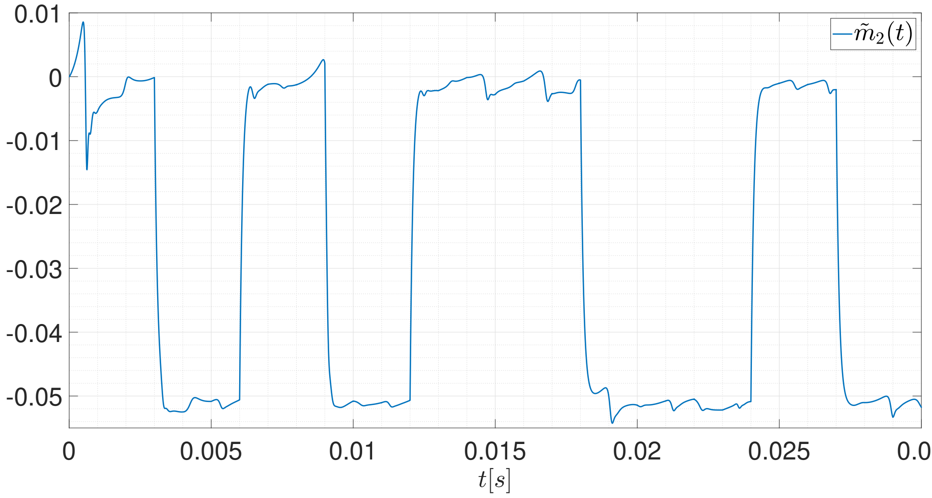

Figure 7.22 – Message error of the second state.

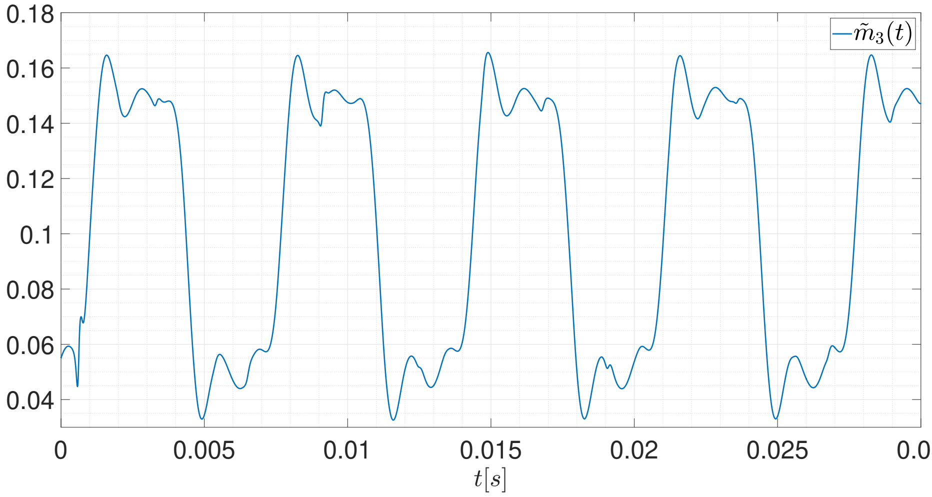

Figure 7.23 – Message error of the third state.

7.5.2 Comparison with [185]

To better show the performance of the proposed scheme, a comparison between the chapter synchronizer performance with that in [185] in the presence of disturbances was realized. In [185], a novel synchronizer for hyperchaotic Lorenz system was introduced. Although finite-time convergence was accomplished, neither external disturbances in the stability analysis nor a simple underactuated control were considered.

The parameters used in the simulation were $a = 20$, $b = 32$, $c = 3$, $d = 1$, $k = -1$, $\psi_1 = \psi_3 = 100$, and $\psi_2 = 10000$. The initial conditions for the master and slave systems in both cases were $x_m(0) = 1$, $y_m(0) = 1$, $z_m(0) = 1$, $w_m(0) = 1$, $x_s(0) = 20$, $y_s(0) = 15$, $z_s(0) = 15$, and $w_s(0) = -200$. Also, an unmatched disturbance was introduced to check the robustness of the proposed approach when it is compared with that in [185]. The used disturbance was defined as

$$
\begin{aligned}
h_1 &= 0.3 \cdot \exp(5 \cdot 10^{-5} \cdot x_s^2) \\
h_2 &= 0.2 \cdot \exp(10^{-5} \cdot y_s^2) \\
h_3 &= 3sin(3t) + cos(20t) \\
h_4 &= 5sin(10t) + 10cos(t)
\end{aligned} \tag{7.26}
$$

Three systems were simulated: the master (7.1), slave (7.2), using an underactuated control, and the slave system in [185], in which a full-actuated control is employed. From simulations shown in Figure 7.24 - 7.31, it can be seen that the performance of the proposed scheme is similar to that in [185]. However, the proposed approach is simpler than that in [185] since the proposed control is only used in the x_s and w_s channels.

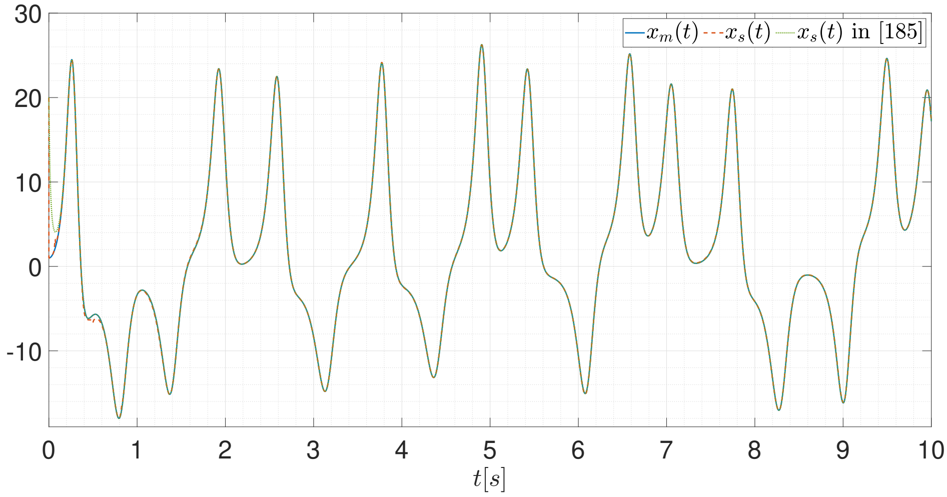

Figure 7.24 – Synchronization error of $x(t)$.

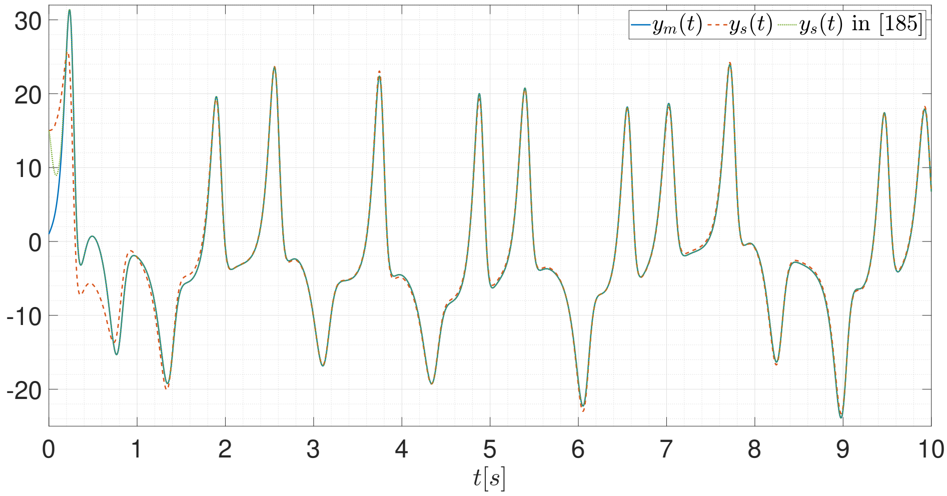

Figure 7.25 – Synchronization error of $y(t)$.

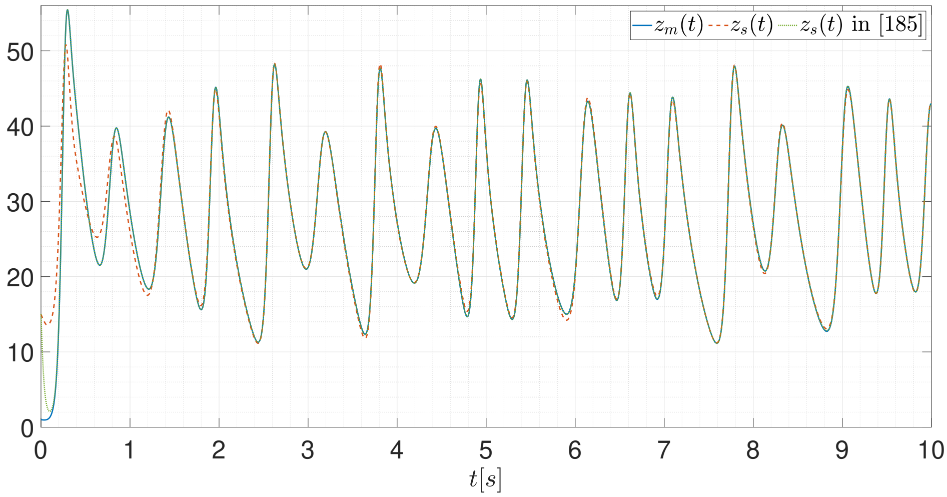

Figure 7.26 – Synchronization error of $z(t)$.

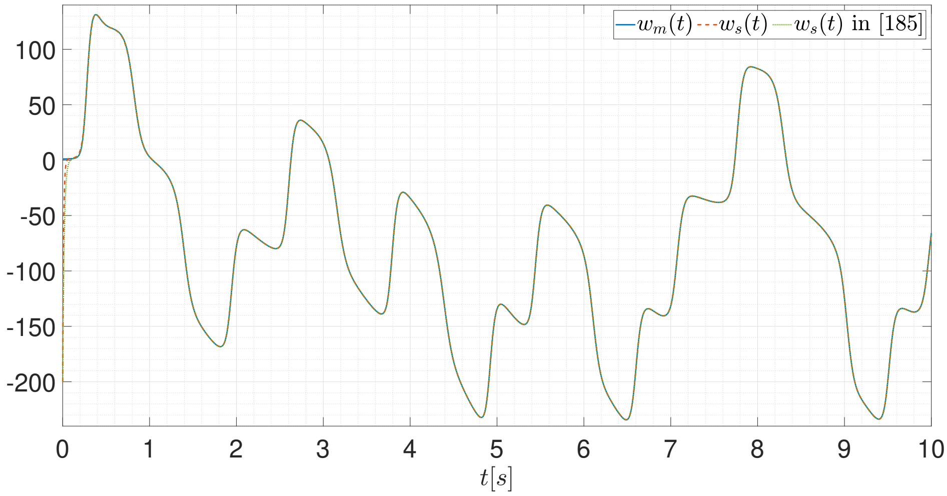

Figure 7.27 – Synchronization error of $w(t)$.

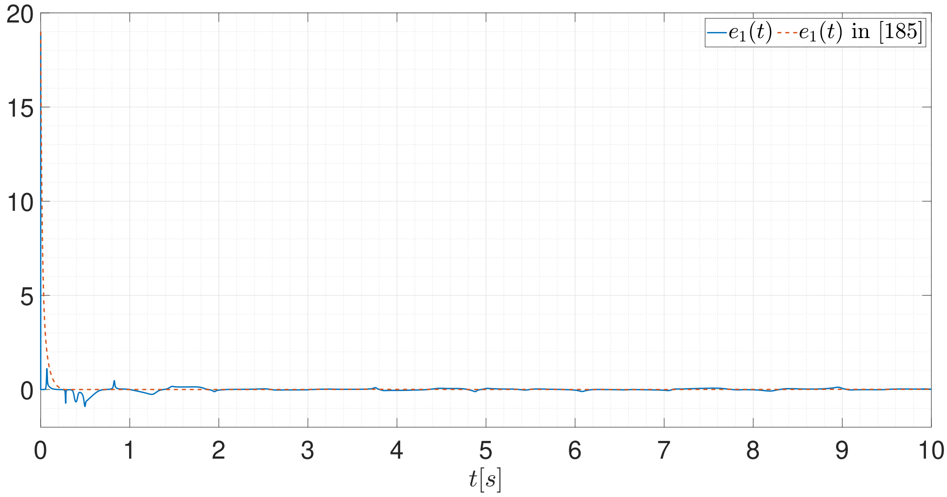

Figure 7.28 – Synchronization error of $x(t)$.

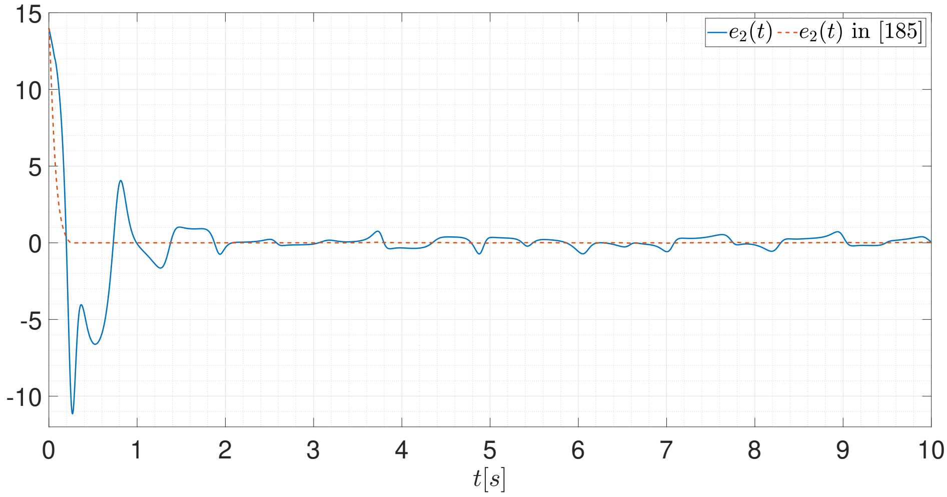

Figure 7.29 – Synchronization error of $y(t)$.

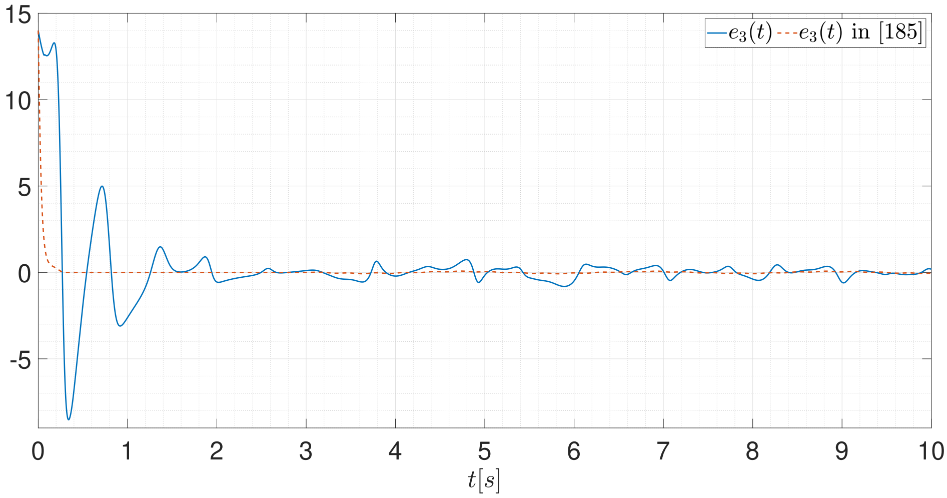

Figure 7.30 – Synchronization error of $z(t)$.

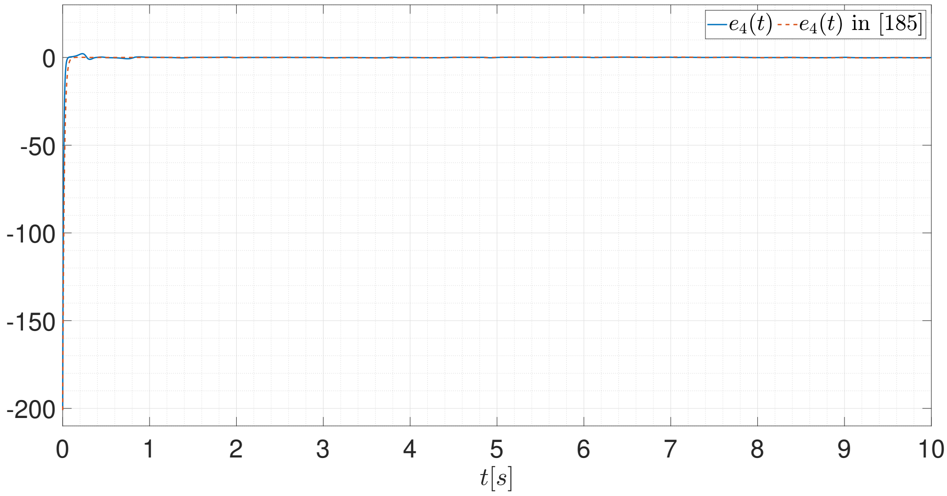

Figure 7.31 – Synchronization error of $w(t)$.

Table 7.2 shows that the optimal control gain value is approximately $\psi_1 = 1$, $\psi_2 = 10^6$, and $\psi_3 = 100$. Smaller values of these parameters lead to larger values of β and smaller values of ρ, consequently, lead to greater synchronization error values. Theoretically, larger values of these parameters should not increase the synchronization error value, but higher control gain can lead to greater approximation errors in computer simulations.

Table 7.2 – Root mean square of state errors for t = [0 10] seconds and consider that $\psi_1 = \psi_3 = 100$, and $\psi_2 = 10000$.

Root Mean Square of State Errors in the Proposed Algorithm							
			$e_{1_{rms}}$	$e_{2_{rms}}$	$e_{3_{rms}}$	$e_{4_{rms}}$	e_{rms}
ψ_1	ψ_2	ψ_3	1.493294	4.721932	4.641420	38.558155	38.780224
$0.01\psi_1$	ψ_2	ψ_3	1.429751	4.484119	4.419726	36.490260	36.698627
$0.1\psi_1$	ψ_2	ψ_3	1.436064	4.462597	4.410235	36.719126	36.928169
$10\psi_1$	ψ_2	ψ_3	1.493636	4.337775	4.374533	38.841132	39.064938
$100\psi_1$	ψ_2	ψ_3	1.519841	4.150333	4.289482	39.527566	39.754720
ψ_1	$0.01\psi_2$	ψ_3	1.575117	4.434670	4.543989	39.686591	39.913421
ψ_1	$0.1\psi_2$	ψ_3	1.520767	4.514290	4.536734	39.008991	39.233795
ψ_1	$10\psi_2$	ψ_3	1.377317	4.620323	4.497149	36.115979	36.329537
ψ_1	$100\psi_2$	ψ_3	1.306059	4.572268	4.432235	34.662253	34.867462
ψ_1	ψ_2	$0.01\psi_3$	1.563147	3.839121	3.962410	187.732569	187.735909
ψ_1	ψ_2	$0.1\psi_3$	1.422606	3.782394	3.973009	41.596886	41.846137
ψ_1	ψ_2	$10\psi_3$	1.557796	4.895822	4.976434	39.282801	39.510939
ψ_1	ψ_2	$100\psi_3$	1.603156	4.883210	5.200149	39.229397	39.457287

7.6 SUMMARY

An underactuated scheme for projective synchronization of hyperchaotic systems based on Lyapunov theory has been proposed in this chapter. The main advantages of the proposed synchronization method are its simplicity, low cost of implementation, and its ability to tackle unmatched disturbances. It has been further employed in chaos-based secure communication to depict the performance of a typical application. With the use of electronic circuits, the ease of implementation has been shown as well. It should be noted that the decoded signal is close to the original, as expected, even in the presence of unmatched disturbances and underactuated control.

Potential drawbacks of the scheme must mention the need to use two control signals instead of just one. This is due to the topology of the considered hyperchaotic systems. Future works will include the underactuated synchronization of hyperchaotic systems of high dimension by using online approximators such as neural networks and fuzzy systems.

8 CONCLUSIONS

This work has studied synchronization and identification schemes of nonlinear systems based on the Lyapunov theory.

8.1 IDENTIFICATION CASE

Initially, all issues related to identification based on neural networks and Lyapunov methods relevant to this work have been considered. In the sequence, a scheme about the issues mentioned earlier based on Lyapunov arguments has been proposed to relate the state to independent design parameters to decouple the transient and steady-error performances. Although several works in the literature consider the identification and control based on neural networks, it is noteworthy that the decoupling of the transient and steady-error performances in these problems has rarely been investigated. In particular, secure communication based on analog chaos and control of welding systems is two topics that have motivated enormous technological and scientific interest in the last years. Hence, financial hyperchaotic and welding systems have been employed to validate the identification scheme in this work.

On the other hand, all simulations were considered the presence of disturbances to assess the robustness of the proposed algorithms. Several classes of disturbances have been used to show that the proposed schemes corroborate the theoretical results, which are that the algorithms are stable. The residual errors converge to an arbitrary neighborhood of the origin, where the transient and steady-error performances can be adjusted irrespective.

Exhaustive simulations were carried out to evaluate the influence of the design parameters on the performance of the algorithms. The independence of the transient and steady-error performances has been fully confirmed. In particular, the identification and control of a welding system have been accomplished, which showed that the proposed schemes could be used successfully in this case, where the transient and steady-state can be adjusted according to any desired geometric parameters of the weld bead.

8.2 SYNCHRONIZATION CASE

In reviewing previous work on synchronization, it was found that in most of the works found in the literature, the dimension of the control input is equal to the number of states in the dynamic system. In addition, many works did not consider the presence of disturbances in all states of the system. In this way, it was decided in this doctoral thesis to propose some underactuated synchronization schemes based on chaotic systems and that theoretically considered the presence of disturbances in all state signs.

Initially, some synchronization systems based on chaotic 3-D systems were considered and then extended to hyperchaotic 4-D systems. Additionally, a synchronization scheme was generalized for the case of projective synchronization. All the proposed synchronization schemes have been extended to the case of secure communication. Most secure communication systems were also designed in analog electronics. Several simulations were carried out to validate the proposed theoretical schemes.

8.3 CHAPTERS CONCLUSIONS

Chapter 2 considered the online identification problem of uncertain systems. Based on parallel and series-parallel configurations with feedback and Lyapunov arguments, a unified identification algorithm was introduced to ensure the boundedness of all associated errors and convergence of the state estimation error to an arbitrary neighborhood of the origin. The main peculiarity of the proposed algorithm lies in allowing the adjustment of the identification transient by using parameters that are not related to the residual state error. Two examples were deemed to validate the theoretical results and show the relevance of applying the proposed methodology for online weld geometry prediction.

In Chapter 3 it was proposed a scheme for secure telecommunication based on the synchronization of an underactuated chaotic Lorenz system, Lyapunov analysis, and analog electronics. Unlike most of the schemes usually found in the literature, the proposed scheme only requires that the control signal act in one of the state equations of the slave system. The proposed scheme has the advantages of being robust against disturbances (internal and external) and simple, which is essential since it leads to important cost reductions when implemented using analog electronics. Computational simulations using Matlab and Multisim were performed to validate the proposed robustness and simplicity of the scheme.

Chapter 4 proposed a scheme for secure telecommunication based on chaotic oscillators and Lyapunov theory. The presence of internal and external disturbances to increase the robustness of the method is considered in the Lyapunov design. Another advantage of the proposed approach is that only one control input is required for synchronizing the master system with the slave system, making it relatively easy for application. Additionally, implementation via analog circuits of the proposed model is performed to validate the theoretical analysis (this is the main advantage of this scheme about the scheme proposed in chapter 3).

Chapter 5 was concerned with the encryption and decryption problem based on the synchronization of hyperchaotic systems (the main advantage of this scheme about the scheme proposed in Chapters 3 and 4). Basically, by using Lyapunov arguments, a synchronization algorithm was introduced to allow the encryption/decryption of information signals. In contrast to what is found in the literature, the proposed scheme uses hyperchaotic underactuated oscillators (transmitter/receiver), which synchronize even in the presence of bounded perturbations in all states. Simulations were provided to validate the theoretical results.

Chapter 6 proposed a scheme for secure telecommunication based on hyperchaotic systems and the Lyapunov theory. The main advantage of this scheme about the schemes proposed in chapters 3, 4, and 5 is the projective synchronization approach. The design of the projective synchronizer considers the presence of disturbances to increase the robustness of the method. The main advantage of the proposed approach lies in that only two control inputs are required to synchronize the master and slave systems. Hence, the control structure is simple, which simplifies the applications. Based on Lyapunov theory, the proposed approach ensures the finite-time convergence of the synchronization error to a bounded region, even in the presence of disturbances in all states. A comparison study and secure communication applications in which messages can be encrypted and decoded were also addressed to validate the theory.

8.4 FUTURE WORKS

For future works, the following research lines are suggested:

- It is well-known that linearly parameterized neural networks suffer from the "curse of dimensionality" as considered in this work. Therefore, a natural sequence to alleviate the drawback mentioned earlier lies in nonlinearly parameterized neural networks. In this sense, the identifier in [150] can be used as a starting point.
- The application of the proposed identification scheme to the output feedback control problem. Like [128], where the states were estimated through an adaptive observer.
- To implement the circuital scheme experimentally of synchronization of a hyperchaotic system.
- The proposal of an underactuated synchronization system and secure telecommunication that works in hyperchaotic systems that there are more than four states.
- The proposal of an underactuated synchronization system in which a neural network constitutes the slave system.
- Propose an adaptive underactuated chaotic synchronization system.

- In secure communication applications find ways to protect messages against attacks, such as using chaotic auxiliary systems to encode and decode messages.

REFERENCES

[1] R. Isermann and M. Münchhof, *Identification of Dynamic Systems - An Introduction with Applications*, 1st ed. Springer, Verlag Berlin Heidelberg, 2011.

[2] L. Zadeh, "On the identification problem," *IRE Transactions on Circuit Theory*, vol. 3, no. 4, pp. 277–281, 1956.

[3] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Transactions on neural networks*, vol. 1, no. 1, pp. 4–27, 1990.

[4] Y. Zhang, Y. Liu, and L. Liu, "Adaptive finite-time NN control for 3-DOF active suspension systems with displacement constraints," *IEEE Access*, vol. 7, pp. 13 577–13 588, 2019.

[5] S. Ge, C. Hang, T. Lee, and T. Zhang, *Stable Adaptive Neural Network Control*, 1st ed. Springer, Kluwer academic publishers, 2002.

[6] R. Antunes, V. A. GonzÁlez, and K. Walsh, "Identification of Repetitive Processes at Steady- and Unsteady-state: Transfer Function," *Research Gate Conference Paper Perth, Australia*, pp. 793–802, 2015.

[7] B. Jovic, *Synchronization techniques for chaotic communication systems.* Springer Science & Business Media, 2011.

[8] S. H. Strogatz, *Nonlinear dynamics and chaos with student solutions manual: With applications to physics, biology, chemistry, and engineering*. CRC press, 2018.

[9] K. H. M. Gularte and J. A. R. Vargas, "Open and Closed Loop Neural Identification of Uncertain Systems with Transient Time Adjustment," *2018 IEEE International Conference on Automation/XXIII Congress of the Chilean Association of Automatic Control (ICA-ACCA)*, pp. 1–7, 2018.

[10] K. H. M. Gularte, L. N. C. Rêgo, and J. A. R. Vargas, "Scheme for chaos-based encryption and lyapunov analysis," *International Conference on Automation/XXIII Congress of the Chilean Association of Automatic Control (ICA-ACCA)*, pp. 1–7, 2018.

[11] K. H. M. Gularte, L. M. Alves, J. A. R. Vargas, J. P. A. Maranhão, G. C. Carvalho, S. C. A. Alfaro, and J. F. A. Romero, "A Chaotic Synchronization scheme for information security," *2019 13th International Conference on Signal Processing and Communication Systems (ICSPCS)*, pp. 1–10, 2019.

[12] K. H. M. Gularte, L. F. Q. Martins, J. A. R. Vargas, J. P. A. Maranhão, W. A. M. Gabalan, and J. F. A. Romero, “A Scheme for Encryption/Decryption based on Hyperchaotic Systems and Lyapunov Theory,” *2019 13th International Conference on Signal Processing and Communication Systems (ICSPCS)*, pp. 1–9, 2019.

[13] K. H. M. Gularte, R. R. Santos, W. A. M. Gabalan, and J. A. R. Vargas, “Proposta para comunicação segura baseada em sincronização projetiva caótica,” *ANAIS DO 14º SIMPóSIO BRASILEIRO DE AUTOMAçãO INTELIGENTE, 2019, Ouro Preto*, pp. 2939–2945, 2019.

[14] K. H. M. Gularte, V. V. Graciano, W. A. M. Gabalan, and J. A. R. Vargas, “Esquema de sincronização projetiva baseado em caos e análise de lyapunov,” *ANAIS DO 14º SIMPóSIO BRASILEIRO DE AUTOMAçãO INTELIGENTE, 2019, Ouro Preto*, pp. 2852–2857, 2019.

[15] K. H. M. Gularte, A. Zaiter, R. R. Santos, and J. A. R. Vargas, “Sincronização de um sistema sprott k-caótico subatuado baseado em controle proporcional com ganho variável,” *XXII Congresso Brasileiro de Automática, 2018, João Pessoa. Anais do XXII Congresso Brasileiro de Automática*, vol. 1, no. 1, 2019.

[16] K. H. M. Gularte, J. J. M. Chavez, J. A. R. Vargas, and S. C. A. Alfaro, “An adaptive neural identifier with applications to financial and welding systems,” *International Journal of Control, Automation and Systems*, vol. 19, no. 5, pp. 1976–1987, 2021.

[17] K. H. Gularte, J. C. Gómez, J. A. Vargas, and R. R. Dos Santos, “Underactuated synchronization scheme of a hyperchaotic finance system,” in *2021 14th IEEE International Conference on Industry Applications (INDUSCON)*. IEEE, 2021, pp. 1335–1339.

[18] ——, “Projective synchronization and antisynchronization of underactuated systems,” in *2021 14th IEEE International Conference on Industry Applications (INDUSCON)*. IEEE, 2021, pp. 1317–1322.

[19] ——, “Chaos-based cryptography using an underactuated synchronizer,” in *2021 14th IEEE International Conference on Industry Applications (INDUSCON)*. IEEE, 2021, pp. 1303–1308.

[20] K. H. Gularte, J. C. Gómez, M. E. Vizcarra Melgar, and J. A. Vargas, “Chaos synchronization and its application in parallel cryptography,” in *2021 IEEE 5th Colombian Conference on Automatic Control (CCAC)*, 2021, pp. 198–203.

[21] K. H. M. Gularte, J. C. G. Gómez, M. E. Vizcarra Melgar, and J. A. Vargas, “Underactuated 4d-hyperchaotic system for secure communication in the presence of dis-

turbances," in *2021 IEEE 5th Colombian Conference on Automatic Control (CCAC)*, 2021, pp. 210–215.

[22] H. K. Khalil, *Nonlinear systems*. Upper Saddle River, 2002.

[23] D. S. Mitrinovic, J. Pecaric, and A. M. Fink, *Classical and new inequalities in analysis*. Springer Science & Business Media, 2013, vol. 61.

[24] P. A. Ioannou and J. Sun, *Robust adaptive control*. Prentice Hall Inc., Upper Saddle River, New Jersey, 1995.

[25] K. B. Petersen and M. S. Pedersen, *The Matrix Cookbook*. Technical University of Denmark, 2012.

[26] A. S. Poznyak, E. N. Sanchez, and W. Yu, *Differential Neural Networks for Robust Nonlinear Control - Identification, State Estimation and Trajectory Tracking*, 1st ed. World Scientific, 2001.

[27] S. Haykin, *Neural networks and learning machines*, 3rd ed. Pearson Prentice Hall, 2008.

[28] E. B. Kosmatopoulos, M. M. Polycarpou, M. A. Christodoulou, and P. A. Ioannou, "High-order neural network structures for identification of dynamical systems," *IEEE transactions on Neural Networks*, vol. 6, no. 2, pp. 422–431, 1995.

[29] Q. Zhang and A. Benveniste, "Wavelet Networks," *IEEE Transactions on Neural Networks*, vol. 3, no. 6, pp. 889–898, 1992.

[30] L. Wang, *Adaptive Fuzzy Systems and Control: Design and Stability Analysis*. Pearson Prentice Hall, 1994.

[31] J. Han and C. Morag, "The influence of the sigmoid function parameters on the speed of backpropagation learning," *Springer - Natural to Artificial Neural Computation*, p. 195–201, 1995.

[32] F. Moon, "Nonlinear dynamics," in *Encyclopedia of Physical Science and Technology (Third Edition)*, 3rd ed., R. A. Meyers, Ed. New York: Academic Press, 2003, pp. 523–535. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B0122274105004841

[33] A. Wolf, J. B. Swift, H. L. Swinney, and J. A. Vastano, "Determining lyapunov exponents from a time series," *Physica D: nonlinear phenomena*, vol. 16, no. 3, pp. 285–317, 1985.

[34] M. T. Rosenstein, J. J. Collins, and C. J. De Luca, “A practical method for calculating largest lyapunov exponents from small data sets,” *Physica D: Nonlinear Phenomena*, vol. 65, no. 1-2, pp. 117–134, 1993.

[35] H. Kantz, “A robust method to estimate the maximal lyapunov exponent of a time series,” *Physics letters A*, vol. 185, no. 1, pp. 77–87, 1994.

[36] T. Kapitaniak and L. O. Chua, “Hyperchaotic attractors of unidirectionally-coupled chua’s circuits,” *International Journal of Bifurcation and Chaos*, vol. 4, no. 2, pp. 477–482, 1994.

[37] J. C. Sprott, *Elegant chaos: algebraically simple chaotic flows.* World Scientific, 2010.

[38] S. Boccaletti, J. Kurths, G. Osipov, D. L. Valladares, and C. S. Zhou, “The synchronization of chaotic systems,” *Physics reports*, vol. 366, no. 1-2, pp. 1–101, 2002.

[39] L. M. Pecora and T. L. Carroll, “Synchronization in chaotic systems,” *Physical review letters*, vol. 64, no. 8, p. 821, 1990.

[40] H. Fujisaka and T. Yamada, “Stability theory of synchronized motion in coupled-oscillator systems,” *Progress of theoretical physics*, vol. 69, no. 1, pp. 32–47, 1983.

[41] M. G. Rosenblum, A. S. Pikovsky, and J. Kurths, “Phase synchronization of chaotic oscillators,” *Physical review letters*, vol. 76, no. 11, p. 1804, 1996.

[42] ——, “From phase to lag synchronization in coupled chaotic oscillators,” *Physical Review Letters*, vol. 78, no. 22, p. 4193, 1997.

[43] N. F. Rulkov, M. M. Sushchik, L. S. Tsimring, and H. D. Abarbanel, “Generalized synchronization of chaos in directionally coupled chaotic systems,” *Physical Review E*, vol. 51, no. 2, p. 980, 1995.

[44] L. Kocarev and U. Parlitz, “Generalized synchronization, predictability, and equivalence of unidirectionally coupled dynamical systems,” *Physical review letters*, vol. 76, no. 11, p. 1816, 1996.

[45] S. Boccaletti and D. Valladares, “Characterization of intermittent lag synchronization,” *Physical Review E*, vol. 62, no. 5, p. 7497, 2000.

[46] M. A. Zaks, E.-H. Park, M. G. Rosenblum, and J. Kurths, “Alternating locking ratios in imperfect phase synchronization,” *Physical review letters*, vol. 82, no. 21, p. 4228, 1999.

[47] R. Femat and G. Solıs-Perales, "On the chaos synchronization phenomena," *Physics Letters A*, vol. 262, no. 1, pp. 50–60, 1999.

[48] M. M. Al-sawalha, A. Alomari, S. Goh, and M. Nooran, "Active anti-synchronization of two identical and different fractional-order chaotic systems," *International Journal of Nonlinear Science*, vol. 11, no. 3, pp. 267–274, 2011.

[49] G. Si, Z. Sun, Y. Zhang, and W. Chen, "Projective synchronization of different fractional-order chaotic systems with non-identical orders," *Nonlinear Analysis: Real World Applications*, vol. 13, no. 4, pp. 1761–1771, 2012.

[50] C.-M. Chang and H.-K. Chen, "Chaos and hybrid projective synchronization of commensurate and incommensurate fractional-order chen–lee systems," *Nonlinear Dynamics*, vol. 62, no. 4, pp. 851–858, 2010.

[51] S. Shao, "Controlling general projective synchronization of fractional order rossler systems," *Chaos, Solitons & Fractals*, vol. 39, no. 4, pp. 1572–1577, 2009.

[52] P. Zhou, F. Kuang, and Y.-M. Cheng, "Generalized projective synchronization for fractional order chaotic systems," *Chinese Journal of Physics*, vol. 48, no. 1, pp. 49–56, 2010.

[53] G. Li, "Modified projective synchronization of chaotic system," *Chaos, Solitons & Fractals*, vol. 32, pp. 1786–1790, 2007.

[54] P. Zhou and W. Zhu, "Function projective synchronization for fractional-order chaotic systems," *Nonlinear Analysis: Real World Applications*, vol. 12, no. 2, pp. 811–816, 2011.

[55] Y. Chai, L. Chen, R. Wu, and J. Dai, "Qs synchronization of the fractional-order unified system," *Pramana*, vol. 80, no. 3, pp. 449–461, 2013.

[56] H. Feng, Y. Yang, and S. P. Yang, "A new method for full state hybrid projective synchronization of different fractional order chaotic systems," in *Applied Mechanics and Materials*, vol. 385. Trans Tech Publ, 2013, pp. 919–922.

[57] H. Xi, S. Yu, R. Zhang, and L. Xu, "Adaptive impulsive synchronization for a class of fractional-order chaotic and hyperchaotic systems," *Optik*, vol. 125, no. 9, pp. 2036–2040, 2014.

[58] M. P. Aghababa, "Finite-time chaos control and synchronization of fractional-order nonautonomous chaotic (hyperchaotic) systems using fractional nonsingular terminal sliding mode technique," *Nonlinear Dynamics*, vol. 69, no. 1, pp. 247–261, 2012.

[59] K. Mathiyalagan, J. H. Park, and R. Sakthivel, "Exponential synchronization for fractional-order chaotic systems with mixed uncertainties," *Complexity*, vol. 21, no. 1, pp. 114–125, 2015.

[60] T. Yang, "A survey of chaotic secure communication systems," *International journal of computational cognition*, vol. 2, no. 2, pp. 81–130, 2004.

[61] K. M. Cuomo, A. V. Oppenheim, and S. H. Strogatz, "Synchronization of Lorenz-based chaotic circuits with applications to communications," *Chaos, Solitons & Fractals*, vol. 40, no. 10, pp. 626–633, 1993.

[62] H. Dedieu, M. P. Kennedy, and M. Hasler, "Chaos shift keying: modulation and demodulation of a chaotic carrier using self-synchronizing chua's circuits," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 40, no. 10, pp. 634–642, 1993.

[63] K. M. Short, "Steps toward unmasking secure communications," *International Journal of Bifurcation and Chaos*, vol. 4, no. 04, pp. 959–977, 1994.

[64] T. Yang and L. O. Chua, "Secure communication via chaotic parameter modulation," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 43, no. 9, pp. 817–819, 1996.

[65] C. W. Wu and L. O. Chua, "A simple way to synchronize chaotic systems with applications to secure communication systems," *International Journal of Bifurcation and Chaos*, vol. 3, no. 06, pp. 1619–1627, 1993.

[66] K. M. Short, "Unmasking a modulated chaotic communications scheme," *International Journal of Bifurcation and Chaos*, vol. 6, no. 02, pp. 367–375, 1996.

[67] T. Yang, L.-B. Yang, and C.-M. Yang, "Application of neural networks to unmasking chaotic secure communication," *Physica D: Nonlinear Phenomena*, vol. 124, no. 1-3, pp. 248–257, 1998.

[68] ——, "Breaking chaotic secure communication using a spectrogram," *Physics Letters A*, vol. 247, no. 1-2, pp. 105–111, 1998.

[69] T. Yang, C. W. Wu, and L. O. Chua, "Cryptography based on chaotic systems," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 44, no. 5, pp. 469–472, 1997.

[70] L. Ljung, *System identification: Theory for the user*, 2nd ed. Prentice-Hall International, New Jersey, USA, 1999.

[71] J. Vörös, "Online system identification for lifetime diagnostic of supercapacitors with guaranteed stability," *Nonlinear Dynamics*, vol. 87, no. 2, pp. 1427–1434, 2017.

[72] T. A. Tutunji, "Parametric system identification using neural networks," *Applied Soft Computing*, vol. 47, pp. 251–261, 2016.

[73] H. Chaoui, A. E. Mejdoubi, A. Oukaour, and H. Gualous, "Online system identification for lifetime diagnostic of supercapacitors with guaranteed stability," *IEEE Transactions on Neural Networks*, vol. 24, no. 6, pp. 2094–2102, 2016.

[74] E. Grzeidak, J. A. R. Vargas, and S. C. A. Alfaro, "ELM with guaranteed performance for online approximation of dynamical systems," *Nonlinear Dynamics*, vol. 91, no. 3, pp. 1587–1603, 2018.

[75] L. Sersour, T. Djamah, and M. Bettayeb, "Nonlinear system identification of fractional Wiener models," *Nonlinear Dynamics*, vol. 92, no. 4, pp. 1493–1505, 2018.

[76] A. Xiong, X. Zhao, J. Han, and G. Liug, "Application of the chaos theory in the analysis of EMG on patients with facial paralysis," *Robot Intelligence Technology and Applications 2*, pp. 805–819, 2014.

[77] N. I. Chaudhary and M. A. Z. Raja, "Identification of Hammerstein nonlinear ARMAX systems using nonlinear adaptive algorithms," *Nonlinear Dynamics*, vol. 79, no. 2, pp. 1385–1397, 2015.

[78] L. Ma and X. Liu, "A nonlinear recursive instrumental variables identification method of Hammerstein ARMAX system," *Nonlinear Dynamics*, vol. 79, no. 2, pp. 1601–1613, 2015.

[79] S. Boubaker, "Identification of nonlinear Hammerstein system using mixed integer-real coded particle swarm optimization: application to the electric daily peak-load forecasting," *Nonlinear Dynamics*, vol. 90, no. 2, pp. 797–814, 2017.

[80] S. Wang, W. Wang, F. Liu, Y. Tang, and X. Guan, "Identification of chaotic system using Hammerstein-ELM model," *Nonlinear Dynamics*, vol. 81, no. 3, pp. 1081–1095, 2015.

[81] A. K. Mani, M. D. Narayanan, and M. Sen, "Parametric identification of fractional-order nonlinear systems," *Nonlinear Dynamics*, vol. 93, no. 2, pp. 945–960, 2018.

[82] B. Wang, S. E. Li, H. Peng, and Z. Liu, "Fractional-order modeling and parameter identification for lithium-ion batteries," *Journal of Power Sources*, vol. 293, pp. 151–161, 2015.

[83] Z. B. Li, X. F. Zhang, W. M. Wang, and G. Zhang, “Nonlinear synchronization of a new lorenz system and its application in secure communication,” *2018 IEEE International Conference on Mechatronics, Robotics and Automation (ICMRA)*, pp. 100–104, 2018.

[84] J. Vörös, “Iterative identification of nonlinear dynamic systems with output backlash using three-block cascade models,” *Nonlinear Dynamics*, vol. 79, no. 3, pp. 2187–2195, 2014.

[85] S. Wei, Z. K. Peng, X. J. Dong, and W. M. Zhang, “A nonlinear subspace-prediction error method for identification of nonlinear vibrating structures,” *Nonlinear Dynamics*, vol. 91, no. 3, pp. 1605–1617, 2018.

[86] S. S. Ge, C. C. Hang, T. H. Lee, and T. Zhang, *Stable adaptive neural network control*. Springer Science & Business Media, 2013, vol. 13.

[87] A. Y. Alanis, E. N. Sanchez, A. G. Loukianov, and E. A. Hernandez, “Discrete-time recurrent high order neural networks for nonlinear identification,” *Journal of the Franklin Institute*, vol. 347, no. 7, pp. 1253–1265, 2010.

[88] I. Chairez, “Adaptive neural network nonparametric identifier with normalized learning laws,” *IEEE transactions on neural networks and learning systems*, vol. 28, no. 5, pp. 1216–1227, 2016.

[89] Z. Man, H. R. Wu, S. Liu, and X. Yu, “A new adaptive backpropagation algorithm based on lyapunov stability theory for neural networks,” *IEEE Transactions on Neural Networks*, vol. 17, no. 6, pp. 1580–1591, 2006.

[90] R. Kumar, S. Srivastava, and J. R. P. Gupta, “Diagonal recurrent neural network based identification of nonlinear dynamical systems with Lyapunov stability based adaptive learning rates,” *Neurocomputing*, vol. 287, pp. 102–117, 2018.

[91] M. M. Polycarpou and P. A. Ioannou, *Identification and control of nonlinear systems using neural network models: Design and stability analysi*. University of Southern Calif, 1991.

[92] L.-X. Wang, “Stable adaptive fuzzy control of nonlinear systems,” *IEEE Transactions on fuzzy systems*, vol. 1, no. 2, pp. 146–155, 1993.

[93] Z. J. Fu, W. F. Xie, and J. Na, “Robust adaptive nonlinear observer design via multi-time scales neural network,” *Neurocomputing*, vol. 190, pp. 217–225, 2016.

[94] J. Zhang, G. G. Walter, Y. Miao, and W. N. W. Lee, “Wavelet neural networks for function learning,” *IEEE transactions on Signal Processing*, vol. 43, no. 6, pp. 1485–1497, 1995.

[95] L. Dou, R. Ji, and J. Gao, “Identification of nonlinear aeroelastic system using fuzzy wavelet neural network,” *Neurocomputing*, vol. 214, pp. 935–943, 2016.

[96] J. Tavoosi, A. A. Suratgar, and M. B. Menhaj, “Stable ANFIS2 for nonlinear system identification,” *Neurocomputing*, vol. 182, pp. 235–246, 2016.

[97] G. Ahmadi and M. Teshnehlab, “Designing and implementation of stable sinusoidal rough-neural identifier,” *IEEE transactions on neural networks and learning systems*, vol. 28, no. 8, pp. 1774–1786, 2016.

[98] Y. Zhang, Y. Liu, and L. Liu, “Minimal learning parameters-based adaptive neural control for vehicle active suspensions with input saturation,” *Neurocomputing*, vol. 396, no. 5, pp. 153–161, 2019.

[99] Y. J. Liu, Q. Zeng, S. Tong, C. P. Chen, and L. Liu, “Adaptive neural network control for active suspension systems with time-varying vertical displacement and speed constraints,” *IEEE Transactions on Industrial Electronics*, vol. 66, no. 12, pp. 9458–9466, 2019.

[100] R. Kumar, S. Srivastava, and J. R. P. Gupta, “Diagonal recurrent neural network based adaptive control of nonlinear dynamical systems using lyapunov stability criterion,” *ISA transactions*, vol. 67, pp. 407–427, 2017.

[101] J. A. R. Vargas, K. H. M. Gularte, and E. M. Hemerly, “On-Line Neuro Identification of Uncertain Systems Based on Scaling and Explicit Feedback,” *Journal of Control, Automation and Electrical Systems*, vol. 24, no. 6, pp. 753–763, 2013.

[102] S. Dadashi, J. Feaster, J. Bayandor, F. Battaglia, and A. J. Kurdila, “Identification and adaptive control of history dependent unsteady aerodynamics for a flapping insect wing,” *Nonlinear Dynamics*, vol. 85, no. 3, pp. 1405–1421, 2016.

[103] J. A. R. Vargas, W. Pedrycz, and E. M. Hemerly, “Improved learning algorithm for two-layer neural networks for identification of nonlinear systems,” *Neurocomputing*, vol. 329, pp. 86–96, 2019.

[104] X.-L. Li, C. Jia, K. Wang, and J. Wang, “Trajectory tracking of nonlinear system using multiple series-parallel dynamic neural networks,” *Neurocomputing*, vol. 168, no. 30, pp. 1–12, 2015.

[105] A. Bolourchi, S. F. Masri, and O. J. Aldraihem, “Development and application of computational intelligence approaches for the identification of complex nonlinear systems,” *Nonlinear Dynamics*, vol. 79, no. 2, pp. 765–786, 2015.

[106] Y. Yang, Z. K. Peng, X. J. Dong, W. M. Zhang, and G. Meng, "Nonlinear time-varying vibration system identification using parametric time–frequency transform with spline kernel," *Nonlinear Dynamics*, vol. 85, no. 3, pp. 1679–1694, 2016.

[107] J.-C. Sun, N. Wang, M. J. E, and Y.-C. Liu, "Extreme learning control of surface vehicles with unknown dynamics and disturbances," *Neurocomputing*, vol. 167, pp. 535–542, 2015.

[108] J. A. R. Vargas and E. M. Hemerly, "Nonlinear adaptive observer design for uncertain dynamical systems," in *Proceedings of the 39th IEEE Conference on Decision and Control (Cat. No. 00CH37187)*, vol. 2. IEEE, 2000, pp. 1307–1308.

[109] G. A. Bestard, R. C. Sampaio, J. A. Vargas, and S. C. A. Alfaro, "Sensor fusion to estimate the depth and width of the weld bead in real time in gmaw processes," *Sensors*, vol. 18, no. 4, p. 962, 2018.

[110] S. B. Chen and N. Lv, "Research evolution on intelligentized technologies for arc welding process," *Journal of Manufacturing Processes*, vol. 16, no. 1, pp. 109–122, 2014.

[111] S.-B. Chen and J. Wu, *Intelligentized methodology for arc welding dynamical processes*. Springer Berlin Heidelberg, 2009.

[112] L. Zhiyong, Z. Qiang, L. Yan, Y. Xiaocheng, and T. S. Srivatsan, "An analysis of gas metal arc welding using the Lyapunov exponent," *Materials and Manufacturing Processes*, vol. 28, no. 2, pp. 213–219, 2013.

[113] Y. Huang, K. Wang, J. Fang, and X. Zhou, "Multifractal analysis for gas metal arc welding," *The International Journal of Advanced Manufacturing Technology*, vol. 94, no. 5-8, pp. 1903–1910, 2018.

[114] Y. Liu, W. Zhang, and Y. Zhang, "Dynamic neuro-fuzzy-based human intelligence modeling and control in GTAW," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 1, pp. 324–335, 2013.

[115] Y. Liu and Y. Zhang, "Iterative local ANFIS-based human welder intelligence modeling and control in pipe GTAW process: A data-driven approach," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 3, pp. 1079–1088, 2014.

[116] Z. Lei, J. Shen, Q. Wang, and Y. Chen, "Real-time weld geometry prediction based on multi-information using neural network optimized by PCA and GA during thin-plate laser welding," *Journal of Manufacturing Processes*, vol. 43, pp. 207–217, 2019.

[117] T. Ç. Akinci, H. S. Noğay, and G. Gökmen, “Determination of optimum operation cases in electric arc welding machine using neural network,” *Journal of mechanical science and technology*, vol. 25, no. 4, pp. 1003–1010, 2011.

[118] P. Sathiya, K. Panneerselvam, and M. Y. A. Jaleel, “Optimization of laser welding process parameters for super austenitic stainless steel using artificial neural networks and genetic algorithm,” *Materials & Design (1980-2015)*, vol. 36, pp. 490–498, 2012.

[119] M. Luo and Y. C. Shin, “Estimation of keyhole geometry and prediction of welding defects during laser welding based on a vision system and a radial basis function neural network,” *The International Journal of Advanced Manufacturing Technology*, vol. 81, no. 1-4, pp. 263–276, 2015.

[120] I.-S. Kim, J.-S. Son, C.-E. Park, I.-J. Kim, and H.-H. Kim, “An investigation into an intelligent system for predicting bead geometry in GMA welding process,” *Journal of Materials Processing Technology*, vol. 159, no. 1, pp. 113–118, 2005.

[121] D. Katherasan, J. V. Elias, P. Sathiya, and A. N. Haq, “Simulation and parameter optimization of flux cored arc welding using artificial neural network and particle swarm optimization algorithm,” *Journal of Intelligent Manufacturing*, vol. 25, no. 1, pp. 67–76, 2012.

[122] A. N. Ahmed, C. M. Noor, M. F. Allawi, and A. El-Shafie, “RBF-NN-based model for prediction of weld bead geometry in Shielded Metal Arc Welding (SMAW),” *Neural Computing and Applications*, vol. 29, no. 3, pp. 889–899, 2018.

[123] R. Sudhakaran, V. V. Murugan, P. S. Sivasakthivel, and M. Balaji, “Prediction and optimization of depth of penetration for stainless steel gas tungsten arc welded plates using artificial neural networks and simulated annealing algorithm,” *Neural Computing and Applications*, vol. 22, no. 3-4, pp. 637–649, 2013.

[124] B. Girinath, N. S. Shanmugam, and K. Sankaranarayanasamy, “Weld bead graphical prediction of cold metal transfer weldment using ANFIS and MRA model on Matlab platform,” *SIMULATION*, vol. 95, no. 8, pp. 725–736, 2018.

[125] H. Yu, G. Cai, and Y. Li, “Dynamic analysis and control of a new hyperchaotic finance system,” *Nonlinear Dynamics*, vol. 67, no. 3, pp. 2171–2182, 2012.

[126] J. A. R. Vargas, E. Grzeidak, and E. M. Hemerly, “Robust adaptive synchronization of a hyperchaotic finance system,” *Nonlinear Dynamics*, vol. 80, no. 1-2, pp. 239–248, 2015.

[127] J. A. R. Vargas, K. H. M. Gularte, and E. M. Hemerly, "Adaptive observer design based on scaling and neural networks," *IEEE Latin America Transactions*, vol. 11, no. 4, pp. 989–994, 2013.

[128] J. Qiu, K. Sun, T. Wang, and H. Ga, "Observer-based fuzzy adaptive event-triggered control for pure-feedback nonlinear systems with prescribed performance," *IEEE Transactions on Fuzzy systems*, vol. 27, no. 11, pp. 2152–2162, 2019.

[129] O. S. Onma, O. I. Olusola, and A. N. Njah, "Synchronization of uncertain chaotic systems via complete-adaptive-impulsive controls," *Journal of Nonlinear Dynamics*, 2014.

[130] G. Li, L. Zeng, L. Zhang, and Q. M. J. Wu, "State identification of Duffing oscillator based on extreme learning machine," *IEEE Signal Processing Letters*, vol. 25, no. 1, pp. 25–29, 2017.

[131] J. Yan and T. Liao, "Discrete sliding mode control for hybrid synchronization of continuous Lorenz systems with matched/unmatched disturbances," *Transactions of the Institute of Measurement and Control*, vol. 40, no. 5, pp. 1417–1424, 2018.

[132] M. Feki, "An adaptive chaos synchronization scheme applied to secure communication," *Chaos, Solitons & Fractals*, vol. 18, pp. 141–148, 2003.

[133] H. Tirandaz, M. Ahmadnia, and H. Tavakoli, "Adaptive Projective Lag Synchronization of T and Lu Chaotic Systems," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 7, pp. 3446–3453, 2017.

[134] J. Yan, M. Hung, T. Chiang, and Y. Yang, "Robust synchronization of chaotic systems via adaptive sliding mode control," *Physics Letters A*, vol. 356, pp. 220–225, 2006.

[135] J. Zhang, C. Li, H. Zhang, and J. Yu, "Chaos synchronization using single variable feedback based on backstepping method," *Chaos, Solitons & Fractals*, vol. 21, pp. 1183–1193, 2004.

[136] Y. Wang, Y. Xia, and P. Zhou, "Fuzzy-model-based sampled-data control of chaotic systems: A fuzzy time-dependent lyapunov-krasovskii functional approach," *IEEE Transactions on fuzzy systems*, vol. 25, no. 6, pp. 1672–1684, 2017.

[137] B. Aguirre-Hernández, E. Campos-Cantón, J. A. López-Rentería, and E. C. Diaz-González, "A polynomial approach for generating a monoparametric family of chaotic attractors via switched linear systems," *Chaos, Solitons & Fractals*, vol. 71, pp. 100–106, 2015.

[138] E.-C. Díaz-González, J. A. López-Rentería, E. Campos-Cantón, and B. Aguirre-Hernández, “Maximal unstable dissipative interval to preserve multi-scroll attractors via multi-saturated functions,” *Journal of Nonlinear Science*, 2016.

[139] E.-C. Díaz-González, B. Aguirre-Hernández, J. A. López-Rentería, E. Campos-Cantón, and C. A. Loredo-Villalobos, “Stability and multiscroll attractors of control systems via the abscissa,” *Mathematical Problems in Engineering*, 2016.

[140] M. M. El-Dessoky, E. O. Alzahrany, and N. A. Almohammadi, “Function Projective Synchronization for Four scroll Attractor by Nonlinear Control,” *Applied Mathematical Sciences*, vol. 11, pp. 1247–1259, 2017.

[141] X. Wu, G. Chen, and J. Cai, “Chaos synchronization of the master-slave generalized Lorenz systems via linear state error feedback control,” *Physica D: Nonlinear Phenomena*, vol. 229, pp. 52–80, 2007.

[142] J. Yang and F. Zhu, “Synchronization for chaotic systems and chaos-based secure communications via both reduced-order and step-by-step sliding mode observers,” *Communications in Nonlinear Science and Numerical Simulation*, vol. 18, pp. 926–937, 2013.

[143] N. Aguila-Camacho, M. A. Duarte-mermoud, and E. Delgado-Aguilera, “Adaptive synchronization of fractional lorenz systems using a reduced number of control signals and parameters,” *Chaos, Solitons & Fractals*, vol. 87, pp. 1–11, 2016.

[144] J. Sun and Y. Shen, “Adaptive anti-synchronization of chaotic complex systems and chaotic real systems with unknown parameters,” *Journal of Vibration and Control*, vol. 22, p. 2992–3003, 2016.

[145] E. N. Lorenz, “Deterministic nonperiodic flow,” *Journal of the atmospheric sciences*, vol. 20, no. 2, pp. 130–141, 1963.

[146] S. Mobayen, S. T. Kingni, V.-T. Pham, F. Nazarimehr, and S. Jafari, “Analysis, synchronisation and circuit design of a new highly nonlinear chaotic system,” *International Journal of Systems Science*, vol. 49, no. 3, pp. 617–630, 2018.

[147] D. Liu, S. Zhu, and K. Sun, “Global anti-synchronization of complex-valued memristive neural networks with time delays,” *IEEE transactions on cybernetics*, vol. 49, no. 5, pp. 1735–1747, 2018.

[148] G. Al-Mahbashi and M. M. Noorani, “Finite-Time Lag Synchronization of Uncertain Complex Dynamical Networks With Disturbances via Sliding Mode Control,” *IEEE Access*, vol. 7, pp. 7082–7092, 2019.

[149] C.-K. Chen, C.-L. Lin, S.-L. Lin, Y.-M. Chiu, and C.-T. Chiang, "A chaotic theoretical approach to ECG-based identity recognition [application notes]," *IEEE Computational Intelligence Magazine*, vol. 9, no. 1, pp. 53–63, 2014.

[150] J. A. Vargas, E. Grzeidak, K. H. M. Gularte, and S. C. A. Alfaro, "An adaptive scheme for chaotic synchronization in the presence of uncertain parameter and disturbances," *Neurocomputing*, vol. 174, pp. 1038–1048, 2016.

[151] G. Tigan and D. Opriş, "Analysis of a 3D chaotic system," *Chaos, Solitons & Fractals*, vol. 36, no. 5, pp. 1315–1319, 2008.

[152] D. Su, W. Bao, J. Liu, and C. Gong, "An efficient simulation of the fractional chaotic system and its synchronization," *Journal of the Franklin Institute*, vol. 355, no. 18, pp. 9072–9084, 2018.

[153] X. Shi and Z. Wang, "Complete synchronization of delay hyperchaotic Lü system via a single linear input," *Nonlinear Dynamics*, vol. 69, no. 4, pp. 2245–2253, 2012.

[154] M. E. Sahin, Z. C. Taskiran, H. Guler, and S. E. Hamamci, "Simulation and implementation of memristive chaotic system and its application for communication systems," *Sensors and Actuators A: Physical*, vol. 290, pp. 107–118, 2019.

[155] T. Wang, D. Wang, and K. Wu, "Chaotic adaptive synchronization control and application in chaotic secure communication for industrial internet of things," *IEEE Access*, vol. 6, pp. 8584–8590, 2018.

[156] B. Naderi, H. Kheiri, and A. Heydari, "Secure communication based on synchronization of three chaotic systems," *International Journal of Nonlinear Science*, vol. 27, no. 1, pp. 53–64, 2019.

[157] J. Yang, Y. Chen, and F. Zhu, "Singular reduced order observer-based synchronization for uncertain chaotic systems subject to channel disturbance and chaos-based secure communication," *Applied Mathematics and Computation*, vol. 229, no. 1, pp. 227–238, 2014.

[158] Ü. Çavuşoğlu, A. Akgül, S. Kaçar, İ. Pehlivan, and A. Zeng, "A novel chaos-based encryption algorithm over tcp data packet for secure communication," *Security and Communication Networks*, vol. 9, no. 11, pp. 1285–1296, 2016.

[159] Ü. Çavuşoğlu, S. Kaçar, I. Pehlivan, and A. Zengin, "Secure image encryption algorithm design using a novel chaos based s-box," *Chaos, Solitons & Fractals*, vol. 95, pp. 92–101, 2017.

[160] A. Anees, A. M. Siddiqui, and F. Ahmed, “Chaotic substitution for highly autocorrelated data in encryption algorithm,” *Communications in Nonlinear Science and Numerical Simulation*, vol. 19, no. 9, pp. 3106–3118, 2014.

[161] G. Xu, Y. Shekofteh, A. Akgül, C. Li, and S. Panahi, “A new chaotic system with a self-excited attractor: entropy measurement, signal encryption, and parameter estimation,” *Entropy*, vol. 20, no. 2, p. 86, 2018.

[162] B. Qiu, L. Li, H. Peng, and Y. Yang, “Asymptotic and finite-time synchronization of memristor-based switching networks with multi-links and impulsive perturbation,” *Neural Computing and Applications*, vol. 31, no. 8, pp. 4031–4047, 2019.

[163] F. T. Arecchi, R. Meucci, A. D. Garbo, and E. Allaria, “Homoclinic chaos in a laser: synchronization and its implications in biological systems,” *Optics and lasers in Engineering*, vol. 39, no. 3, pp. 293–304, 2003.

[164] S. Singh and B. B. Sharma, “Sliding mode control based anti-synchronization scheme for hyperchaotic Lu system,” *2011 International Conference on Communication Systems and Network Technologies*, pp. 382–386, 2011.

[165] T. L. Liao and S. H. Tsai, “Adaptive synchronization of chaotic systems and its application to secure communications,” *Chaos, Solitons & Fractals*, vol. 11, no. 9, pp. 1387–1396, 2000.

[166] A. Loría, “A linear time-varying controller for synchronization of Lü chaotic systems with one input,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 56, no. 8, pp. 674–678, 2009.

[167] H. N. Agiza, “Chaos synchronization of lü dynamical system,” *Nonlinear Analysis: Theory, Methods & Applications*, vol. 58, no. 1-2, pp. 11–20, 2004.

[168] C. C. Yang, “Adaptive synchronization of lü hyperchaotic system with uncertain parameters based on single-input controller,” *Nonlinear Dynamics*, vol. 63, no. 3, pp. 447–454, 2011.

[169] S. Singh, “Single input sliding mode control for hyperchaotic lü system with parameter uncertainty,” *International Journal of Dynamics and Control*, vol. 4, no. 4, pp. 504–514, 2016.

[170] E. M. Elabbasy, H. N. Agiza, and M. M. El-Dessoky, “Adaptive synchronization of a hyperchaotic system with uncertain parameter,” *Chaos, Solitons & Fractals*, vol. 30, no. 5, pp. 1133–1142, 2006.

[171] O. E. Rössler, “An equation for continuous chaos,” *Physics Letters A*, vol. 57, p. 397–398, 1976.

[172] G. Chen and T. Ueta, "Yet another chaotic attractor," *International Journal of Bifurcation and Chaos*, vol. 9, no. 7, p. 1465–1466, 1999.

[173] J. C. Sprott, "Simple chaotic systems and circuits," *American Journal of Physics*, vol. 68, no. 8, pp. 2171–2182, 2000.

[174] J. Lü, G. Chen, and D. Chen, "Bridge the gap between the lorenz system and the chen system," *Int. J. Bifur. Chaos*, vol. 12, no. 12, pp. 2917–2926, 2002.

[175] Z. Wang, C. Volos, S. T. Kingni, A. T. Azar, and V. T. Pham, "Four-wing attractors in a novel chaotic system with hyperbolic sine nonlinearity," *Optik*, vol. 131, pp. 1071–1078, 2012.

[176] S. Vaidyanathan, O. A. Abba, G. Betchewe, and M. Alidou, "A new three-dimensional chaotic system: its adaptive control and circuit design," *International Journal of Automation and Control*, vol. 13, no. 1, pp. 101–121, 2019.

[177] C. Nwachioma, J. H. Pérez-Cruz, A. Jiménez, M. Ezuma, and R. Rivera-Blas, "A new chaotic oscillator—properties, analog implementation, and secure communication application," *IEEE Access*, vol. 7, pp. 7510–7521, 2019.

[178] O. E. Rossler, "An equation for hyperchaos," *Physics Letters A*, vol. 71, no. 2-3, pp. 155–157, 1979.

[179] C. Li, J. C. Sprott, W. Thio, and H. Zhu, "A new piecewise linear hyperchaotic circuit," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 61, no. 12, pp. 977–981, 2014.

[180] Y. Li, W. K. S. Tang, and G. Chen, "Generating hyperchaos via state feedback control," *International Journal of Bifurcation and Chaos*, vol. 15, no. 10, p. 3367–3375, 2012.

[181] Q. Jia, "Hyperchaos generated from the lorenz chaotic system and its control," *Physics Letters A*, vol. 366, p. 217–222, 2007.

[182] K. Rajagopal, S. Vaidyanathan, A. Karthikeyan, and A. Srinivasan, "Complex novel 4d memristor hyperchaotic system and its synchronization using adaptive sliding mode control," *Alexandria Engineering Journal*, vol. 57, no. 2, pp. 683–694, 2018.

[183] J. P. Singh and B. K. Roy, "Hidden attractors in a new complex generalised lorenz hyperchaotic system, its synchronisation using adaptive contraction theory, circuit validation and application," *Nonlinear Dynamics*, vol. 92, no. 2, pp. 373–394, 2018.

[184] I. Ahmad, B. Srisuchinwong, and W. San-Um, “On the first hyperchaotic hyperjerk system with no equilibria: A simple circuit for hidden attractors,” *IEEE Access*, vol. 6, pp. 35 449–35 456, 2018.

[185] C. Zhou, C. Yang, D. Xu, and C. Chen, “Dynamic Analysis and Finite-Time Synchronization of a New Hyperchaotic System With Coexisting Attractors,” *IEEE Access*, vol. 7, pp. 52 896–52 902, 2019.

[186] B. A. Mezatio, M. T. Motchongom, B. R. W. Tekam, R. Kengne, R. Tchitnga, and A. Fomethe, “A novel memristive 6D hyperchaotic autonomous system with hidden extreme multistability,” *Chaos, Solitons & Fractals*, vol. 120, pp. 100–115, 2019.

[187] W. Yu, J. Wang, J. Wang, H. Zhu, M. Li, Y. Li, and D. Jiang, “Design of a new seven-dimensional hyperchaotic circuit and its application in secure communication,” *IEEE Access*, vol. 7, pp. 125 586–125 608, 2019.

[188] P. Li, J. Du, S. Li, and Y. Zheng, “Modulus synchronization of a novel hyperchaotic real system and its corresponding complex system,” *IEEE Access*, vol. 7, pp. 109 577–109 584, 2019.

[189] X.Wang, S. Vaidyanathan, C. Volos, V. T. Pham, and T. Kapitaniak, “Dynamics, circuit realization, control and synchronization of a hyperchaotic hyperjerk system with coexisting attractors,” *Nonlinear Dynamics*, vol. 89, no. 3, pp. 1673–1687, 2017.

[190] Y. Feng, Z. Wei, U. E. Kocamaz, A. Akgül, and I. Moroz, “Synchronization and electronic circuit application of hidden hyperchaos in a four-dimensional self-exciting homopolar disc dynamo without equilibria,” *Complexity*, vol. 2017, 2017.

[191] Y. Zhao, X. Li, and P. Duan, “Observer-based sliding mode control for synchronization of delayed chaotic neural networks with unknown disturbance,” *Neural Networks*, vol. 117, pp. 268–273, 2019.

[192] A. Yousefpour, H. Jahanshahi, J. M. Munoz-Pacheco, S. Bekiros, and Z. Wei, “A fractional-order hyper-chaotic economic system with transient chaos,” *Chaos, Solitons & Fractals*, vol. 130, p. 109400, 2020.

[193] H. Wang, C. Weng, Z. Song, and J. Cai, “Research on the law of spatial fractional calculus diffusion equation in the evolution of chaotic economic system,” *Chaos, Solitons & Fractals*, vol. 131, p. 109462, 2020.

[194] G. A. Bestard, R. C. Sampaio, J. A. Vargas, and S. C. A. Alfaro, “Sensor fusion to estimate the depth and width of the weld bead in real time in gmaw processes,” *Sensors*, vol. 18, no. 4, p. 962, 2018.

[195] H.-P. Ren, M. S. Baptista, and C. Grebogi, “Wireless communication with chaos,” *Physical Review Letters*, vol. 110, no. 18, p. 184101, 2013.

[196] J. Wang, W. Yu, J. Wang, Y. Zhao, J. Zhang, and D. Jiang, “A new six-dimensional hyperchaotic system and its secure communication circuit implementation,” *International Journal of Circuit Theory and Applications*, vol. 47, no. 5, pp. 702–717, 2019.

[197] P. Liu, R. Xi, P. Ren, J. Hou, and X. Li, “Analysis and implementation of a new switching memristor scroll hyperchaotic system and application in secure communication,” *Complexity*, 2018.

[198] B. Naderi and H. Kheiri, “Exponential synchronization of chaotic system and application in secure communication,” *Optik*, vol. 127, no. 5, pp. 2407–2412, 2016.

[199] N. Smaoui, A. Karouma, and M. Zribi, “Secure communications based on the synchronization of the hyperchaotic chen and the unified chaotic systems,” *Communications in Nonlinear Science and Numerical Simulation*, vol. 16, no. 8, pp. 3279–3293, 2011.

[200] W. Yan and Q. Ding, “A new matrix projective synchronization and its application in secure communication,” *IEEE Access*, vol. 7, pp. 112 977–112 984, 2019.

[201] A. Ouannas, A. Karouma, G. Grassi, V.-T. Pham *et al.*, “A novel secure communications scheme based on chaotic modulation, recursive encryption and chaotic masking,” *Alexandria Engineering Journal*, vol. 60, no. 1, pp. 1873–1884, 2021.

[202] F. Yu, L. Li, B. He, L. Liu, S. Qian, Y. Huang, and J. Jin, “Design and FPGA implementation of a pseudorandom number generator based on a four-wing memristive hyperchaotic system and Bernoulli map,” *IEEE Access*, vol. 7, pp. 181 884–181 898, 2019.

[203] Q. Liu and L. Liu, “Color Image Encryption Algorithm Based on DNA Coding and Double Chaos System,” *IEEE Access*, vol. 8, pp. 83 596–83 610, 2020.

[204] J. Xu, P. Li, F. Yang, and H. Yan, “High Intensity Image Encryption Scheme Based on Quantum Logistic Chaotic Map and Complex Hyperchaotic System,” *IEEE Access*, vol. 7, pp. 167 904–167 918, 2019.

[205] X. Wang and L. Liu, “Image Encryption Based on Hash Table Scrambling and DNA Substitution,” *IEEE Access*, vol. 8, pp. 68 533–68 547, 2020.

[206] S. Zhu and C. Zhu, “Plaintext-related image encryption algorithm based on block structure and five-dimensional chaotic map,” *IEEE Access*, vol. 7, pp. 147 106–147 118, 2019.

[207] X. Min, X. Wang, P. Zhou, S. Yu, and H. H. C. Iu, "An Optimized Memristor-Based Hyperchaotic System With Controlled Hidden Attractors," *IEEE Access*, vol. 7, pp. 124 641–124 646, 2019.

[208] F. Yu, H. Shen, L. Liu, Z. Zhang, Y. Huang, B. He, and Q. Xu, "CCII and FPGA realization: a multistable modified fourth-order autonomous Chua's chaotic system with coexisting multiple attractors," *Complexity*, 2020.

[209] B. Bao, Q. Yang, L. Zhu, H. Bao, Q. Xu, Y. Yu, and M. Chen, "Chaotic bursting dynamics and coexisting multistable firing patterns in 3d autonomous Morris–lecar model and microcontroller-based validations," *International Journal of Bifurcation and Chaos*, vol. 29, no. 10, p. 1950134, 2019.

[210] H. Takhi, K. Kemih, L. Moysis, and C. Volos, "Passivity based control and synchronization of perturbed uncertain chaotic systems and their microcontroller implementation," *International Journal of Dynamics and Control*, pp. 1–18,, 2012.

[211] S. He, K. Sun, H. Wang, X. Mei, and Y. Sun, "Generalized synchronization of fractional-order hyperchaotic systems and its DSP implementation," *Nonlinear Dynamics*, vol. 92, no. 1, pp. 85–96, 2018.

[212] X. Wu, Z. Fu, and J. Kurths, "A secure communication scheme based generalized function projective synchronization of a new 5D hyperchaotic system," *Physica Scripta*, vol. 90, no. 4, p. 045210, 2015.

[213] G. Pérez and H. A. Cerdeira, "Extracting messages masked by chaos," *Physical Review Letters*, vol. 74, no. 11, p. 1970, 1995.

[214] L. Cao, "A four-dimensional hyperchaotic finance system and its control problems," *Journal of Control Science and Engineering*, 2018.

[215] D. Liu, S. Zhu, and K. Sun, "Global anti-synchronization of complex-valued memristive neural networks with time delays," *IEEE transactions on cybernetics*, vol. 49, no. 5, pp. 1735–1747, 2018.

[216] R. Mainieri and J. Rehacek, "Projective synchronization in three-dimensional chaotic systems," *Physical Review Letters*, vol. 82, no. 15, p. 3042, 1999.

[217] J. Yan and C. Li, "Generalized projective synchronization of a unified chaotic system," *Chaos, Solitons & Fractals*, vol. 26, no. 4, pp. 1119–1124, 2005.

[218] S. Chen and J. Cao, "Projective synchronization of neural networks with mixed time-varying delays and parameter mismatch," *Nonlinear Dynamics*, vol. 67, no. 2, pp. 1397–1406, 2012.

[219] R. Kumar, S. Sarkar, S. Das, and J. Cao, "Projective Synchronization of Delayed Neural Networks With Mismatched Parameters and Impulsive Effects," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 4, pp. 1211–1221, 2019.

[220] H. Du, Q. Zeng, and C. Wang, "Function projective synchronization of different chaotic systems with uncertain parameters," *Physics Letters A*, vol. 372, no. 33, pp. 5402–5410, 2008.

[221] M. Krstic, P. V. Kokotovic, and I. Kanellakopoulos, *Nonlinear and adaptive control design*. John Wiley & Sons, Inc., 1995.

[222] A. Zarei and S. Tavakoli, "Synchronization of Quadratic Chaotic Systems Based on Simultaneous Estimation of Nonlinear Dynamics," *Journal of Computational and Nonlinear Dynamics*, vol. 13, no. 8, 2018.

[223] L. N. Nguenjou, G. Kom, J. M. Pone, J. Kengne, and A. Tiedeu, "A window of multistability in genesio-tesi chaotic system, synchronization and application for securing information," *AEU-International Journal of Electronics and Communications*, vol. 99, pp. 201–214, 2019.

[224] J. Moreno-Valenzuela and C. Aguilar-Avelar, *Motion control of underactuated mechanical systems*. Springer, 2018, vol. 1.

[225] J. Moreno-Valenzuela, J. Montoya-Cháirez, and V. Santibáñez, "Robust trajectory tracking control of an underactuated control moment gyroscope via neural network–based feedback linearization," *Neurocomputing*, vol. 403, pp. 314–324, 2020.

[226] I. Fantoni, R. Lozano, and S. Sinha, "Non-linear control for underactuated mechanical systems," *Appl. Mech. Rev.*, vol. 55, no. 4, pp. B67–B68, 2002.

[227] S. Rudra, R. K. Barai, and M. Maitra, *Block backstepping design of nonlinear state feedback control law for underactuated mechanical systems*. Springer, 2017.

[228] B. Monga and J. Moehlis, "Supervised learning algorithms for controlling underactuated dynamical systems," *Physica D: Nonlinear Phenomena*, vol. 412, p. 132621, 2020.

[229] A. Sabaghian and S. Balochian, "Parameter estimation and synchronization of hyper chaotic lu system with disturbance input and uncertainty using two under-actuated control signals," *Transactions of the Institute of Measurement and Control*, vol. 41, no. 6, pp. 1729–1739, 2019.

[230] E. Lavretsky and K. A. Wise, "Robust adaptive control," in *Robust and adaptive control*. Springer, 2013, pp. 317–353.

[231] M. K. Shukla and B. Sharma, “Backstepping based stabilization and synchronization of a class of fractional order chaotic systems,” *Chaos, Solitons & Fractals*, vol. 102, pp. 274–284, 2017.

[232] S. Ha, H. Liu, S. Li, and A. Liu, “Backstepping-based adaptive fuzzy synchronization control for a class of fractional-order chaotic systems with input saturation,” *International Journal of Fuzzy Systems*, vol. 21, no. 5, pp. 1571–1584, 2019.

[233] Q. Yao, “Synchronization of second-order chaotic systems with uncertainties and disturbances using fixed-time adaptive sliding mode control,” *Chaos, Solitons & Fractals*, vol. 142, p. 110372, 2021.

[234] V. F. Signing, J. Kengne, and L. Kana, “Dynamic analysis and multistability of a novel four-wing chaotic system with smooth piecewise quadratic nonlinearity,” *Chaos, Solitons & Fractals*, vol. 113, pp. 263–274, 2018.

[235] S. Vaidyanathan, A. Sambas, and M. Mamat, “A new chaotic system with axe-shaped equilibrium, its circuit implementation and adaptive synchronization,” *Archives of Control Sciences*, vol. 28, 2018.

APPENDIX

A CODES

A.1 CODES FOR SIMULATIONS IN CHAPTER 3

A.1.1 Simulink plant used for simulations corresponding to Figures 3.3 - 3.6 and Tables 3.1 - 3.8

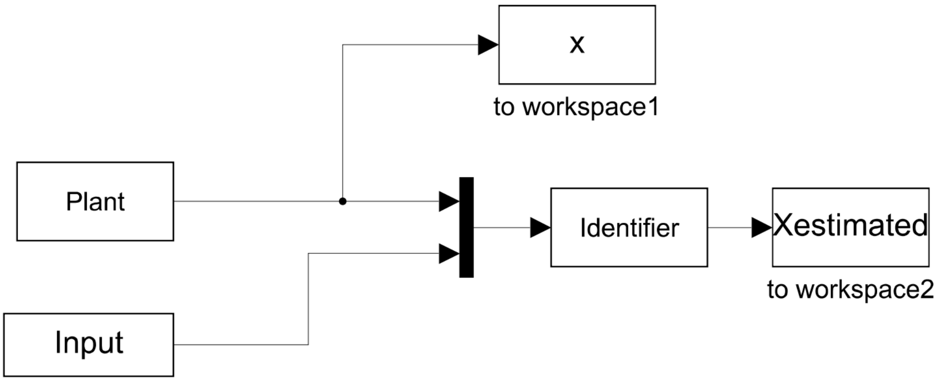

Figure A.1 – Simulink of finance system.

A.1.2 Codes used for simulations corresponding to Figures 3.3 - 3.6 and Tables 3.1 - 3.8

Listing A.1 – Main.m

```
close all
%clear all
clc

format1 = 'jpeg';
format2 = 'epsc';
GAMAW1 = 0.5;
GAMAW2 = 5;
GAMAW3 = 50;

fSize = 38;
axesSize = 38;
lSize = 2;
dvlsize = 2;
dhlsize = 2;

addpath('./Data/');

Simulation_all
```

```
Graphs

fileID = fopen(’./Tables/Table.txt’,’w’);
fprintf(fileID,’Root Mean Square of State Errors\n’);
fprintf(fileID,’gamma_W \t 0.5 \t\t 5 \t\t 50\n’);
fprintf(fileID,’e_1_{rms} \t %f \t %f \t %f\n’, Table(1,1), Table(2,1),
    Table(3,1));
fprintf(fileID,’e_2_{rms} \t %f \t %f \t %f\n’, Table(1,2), Table(2,2),
    Table(3,2));
fprintf(fileID,’e_3_{rms} \t %f \t %f \t %f\n’, Table(1,3), Table(2,3),
    Table(3,3));
fprintf(fileID,’e_4_{rms} \t %f \t %f \t %f\n’, Table(1,4), Table(2,4),
    Table(3,4));
fprintf(fileID,’e_{rms} \t %f \t %f \t %f\n’, Table(1,5), Table(2,5),
    Table(3,5));
fclose(fileID);

save ./Saved_Data/Data.mat
clc
```

Listing A.2 – Simulation_all.m

```
model = ’./Data/Identifier_System.mdl’;

Input = [0 GAMAW1];
open_system(model,’loadonly’); %load simulink model

modelname = ’Identifier_System’;
set_param(modelname,’SolverType’,’Variable-step’...
                   ,’Solver’,’ode45’...
                   ,’RelTol’,’1e-10’...
                   ,’AbsTol’,’1e-10’...
                   ,’MaxStep’,’0.1’...
                   ,’StopTime’,’10’...
                   ,’ScreenColor’,’white’...
                   ,’TimeSaveName’,’t’);
save_system(’./Data/Identifier_System.mdl’);
newsys = ’./Mdl_2018a/Identifier_System.mdl’;
save_system(model,newsys,’ExportToVersion’,’R2018A_MDL’);
sim(model);
clc
close_system(model, 0); %Close the simulink model, 1 to save, 0 to not
    save

gt1=t;
gx1=x;
gxest1=Xestimated;
```

```
aux=size(x);
error1 = zeros(aux(1),4);
error1(:,1) = x(:,1) - Xestimated(:,1);
error1(:,2) = x(:,2) - Xestimated(:,2);
error1(:,3) = x(:,3) - Xestimated(:,3);
error1(:,4) = x(:,4) - Xestimated(:,4);

Input = [0 GAMAW2];
open_system(model,'loadonly'); %load simulink model
sim(model);
close_system(model, 0);
gt2=t;
gx2=x;
gxest2=Xestimated;
aux=size(x);
error2 = zeros(aux(1),4);
error2(:,1) = x(:,1) - Xestimated(:,1);
error2(:,2) = x(:,2) - Xestimated(:,2);
error2(:,3) = x(:,3) - Xestimated(:,3);
error2(:,4) = x(:,4) - Xestimated(:,4);

Input = [0 GAMAW3];
open_system(model,'loadonly'); %load simulink model
sim(model);
close_system(model, 0);
gt3=t;
gx3=x;
gxest3=Xestimated;
aux=size(x);
error3 = zeros(aux(1),4);
error3(:,1) = x(:,1) - Xestimated(:,1);
error3(:,2) = x(:,2) - Xestimated(:,2);
error3(:,3) = x(:,3) - Xestimated(:,3);
error3(:,4) = x(:,4) - Xestimated(:,4);

serror1 = size(error1);
serror2 = size(error2);
serror3 = size(error3);

Table = zeros(3,5);
Table(1,:) = [norm(error1(:,1)) norm(error1(:,2)) norm(error1(:,3)) norm(
   error1(:,4)) norm(error1)/2]/sqrt(serror1(1));
Table(2,:) = [norm(error2(:,1)) norm(error2(:,2)) norm(error2(:,3)) norm(
   error2(:,4)) norm(error2)/2]/sqrt(serror2(1));
Table(3,:) = [norm(error3(:,1)) norm(error3(:,2)) norm(error3(:,3)) norm(
   error3(:,4)) norm(error3)/2]/sqrt(serror3(1));
```

Listing A.3 – Plant.m

```
function [sys,x0,str,ts] = Plant(t,x,u,flag)

%System extract from Yu, H., Cai, G. and Li, Y., "Dynamic analysis and
%control of a new hyperchaotic ?nance control system", Nonlinear Dyn,
%Volume 67, Issue 3, Pages 2171-2182, 2012.

a=0.9; %Constants
b=0.2;
c=1.5;
d=0.2;
k=0.17;

switch flag,
   %%%%%%%%%%%%%%%%%%
   % Initialization %
   %%%%%%%%%%%%%%%%%%
case 0,
  sizes = simsizes;
    sizes.NumContStates = 4; %Number of Constant States
    sizes.NumDiscStates = 0; %Number of Discret States
    sizes.NumOutputs = 4;    %Number of Outputs
    sizes.NumInputs = 0;     %Number of Inputs
    sizes.DirFeedthrough = 1;
    sizes.NumSampleTimes = 1;
    sys = simsizes(sizes);
    x0=[1 2 0.5 0.5]; %Initial Conditions
  str=[];
  ts=[0 0];
   %%%%%%%%%%%%%%%
   % Directives  %
   %%%%%%%%%%%%%%%
case 1,     %Hyperchaotic System
    sys= [ x(3)+(x(2)-a)*x(1)+x(4);
         1-b*x(2)-(x(1))^2;
         -x(1)-c*x(3);
           -d*x(1)*x(2)-k*x(4)]+disturb(x,u,t);
   %%%%%%%%%%%
   % Outputs %
   %%%%%%%%%%%
case 3,
   sys = x;
   %%%%%%%%
   % End  %
   %%%%%%%%
case {2,4,9},
    sys = []; % do nothing
```

```
 otherwise
   error(['unhandled flag = ',num2str(flag)]);
end

function disturb = disturb(x,u,t) %disturb
if t>=5
    n=4;
    disturb=n*[0.7*cos(6*t);cos(5*t);0.9*cos(4*t);0.8*sin(3*t)];
else
    disturb=[0 ; 0; 0; 0];
end
```

Listing A.4 – Identifier.m

```
function [sys,x0,str,ts] = Identifier(t,x,u,flag)

%Controller and its parameters
L = 2*[1 0 0 0; 0 1 0 0;0 0 1 0; 0 0 0 1];
P = 30*[1 0 0 0; 0 1 0 0;0 0 1 0; 0 0 0 1];
PSI=1;
G=1;
W01=G*[1 0 0 0 0 0 0 0]'; %W zero
W02=G*[0 1 0 0 0 0 0 0]';
W03=G*[0 0 1 0 0 0 0 0]';
W04=G*[0 0 0 1 0 0 0 0]';

switch flag,
   %%%%%%%%%%%%%%%%%%%
   % Initialization  %
   %%%%%%%%%%%%%%%%%%%
case 0,

  sizes = simsizes;
    sizes.NumContStates  = 36;  %Number of Constant States
    sizes.NumDiscStates  = 0;   %Number of Discret States
    sizes.NumOutputs     = 5;   %Number of Outputs
    sizes.NumInputs      = 5;   %Number of Inputs
    sizes.DirFeedthrough = 1;
    sizes.NumSampleTimes = 1;
    sys = simsizes(sizes);

    x0=zeros(36,1); %
    x0(1)=-2;
    x0(2)=-2;
    x0(3)=-2;
    x0(4)=-2;
        str=[];
  ts=[0 0];
```

```
   %%%%%%%%%%%%%%%
   % Directives  %
   %%%%%%%%%%%%%%%
case 1,
      %Identification Model
      sys = [-L*[x(1)-u(1);x(2)-u(2);x(3)-u(3);x(4)-u(4)] - u(5)*PSI*[x
   (1)-u(1);x(2)-u(2);x(3)-u(3);x(4)-u(4)] + P*[x(5:12)’;x(13:20)’;x
   (21:28)’;x(29:36)’]*S(x,u);
         %Learning Law
         -2*u(5)*(PSI*(x(5:12)-W01) + (x(1)-u(1))*S(x,u));
         -2*u(5)*(PSI*(x(13:20)-W02) + (x(2)-u(2))*S(x,u));
         -2*u(5)*(PSI*(x(21:28)-W03) + (x(3)-u(3))*S(x,u));
       -2*u(5)*(PSI*(x(29:36)-W04) + (x(4)-u(4))*S(x,u))];
   %%%%%%%%%%%
   % Outputs %
   %%%%%%%%%%%
  case 3,
    sys = [x(1:4);
           norm([x(5:12)’;x(13:20)’;x(21:28)’;x(29:36)’],’fro’)];
case {2,4,9},
    sys = [];

 otherwise
   error([’unhandled flag = ’,num2str(flag)]);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function S = S(x,u)      %Regressors
S=[1*(z(u(1)));
   1*(z(u(2)));
   1*(z(u(3)));
   1*(z(u(4)));
   1*(z(u(1))^2);
   1*(z(u(2))^2);
   1*(z(u(3))^2);
   1*(z(u(4))^2)];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function z = z(uu)  %Sigmoidal Function
lambda=0;
alfa=5;
beta=.5;
z=alfa/(exp(-beta*uu)+1)+lambda;
```

Listing A.5 – Graphs.m

```
addpath(’./Figures/’);

nome_1 = ’./Figures/FIG_2_3’;
nome_2 = ’./Figures/FIG_2_4’;
```

```
nome_3 = './Figures/FIG_2_5';
nome_4 = './Figures/FIG_2_6';

set(0,'DefaultAxesFontSize',axesSize);

fig=figure('visible','off', 'DefaultAxesPosition', [0.064, 0.135, 0.921,
    0.845]);
plot(gt1,error1(:,1),':',gt2,error2(:,1),'r:',gt3,error3(:,1),'k','
    LineWidth',lSize);
grid on
grid minor
YL = get(gca, 'ylim');          %plot the vertical line
YR = YL(2) - YL(1);
YL = [YL(1) - 1000 * YR, YL(2) + 1000 * YR];
line([5, 5], YL, 'YLimInclude', 'off', 'Color','k','LineWidth',dvlsize);
pa = annotation('arrow');       % store the arrow information in pa
pa.Parent = gca;                % associate the arrow the the current
    axes
pa.X = [5 8];                   % the location of arrow
pa.Y = [1 1];
pa.LineWidth  = dhlsize;        % make the arrow bolder for the figure
pa.HeadWidth  = 20;
pa.HeadLength = 20;
text(5.05,1.2,'disturbance in action','Fontsize',fSize) % write a text on
     top of the arrow
h=legend({'$$\gamma_{W} = 0.5$$','$$\gamma_{W} = 5$$','$$\gamma_{W} = 50
    $$'});
set(h,'Interpreter','Latex','FontSize',fSize);
ylabel('$$\tilde{x}_{1}(t)$$','Interpreter','Latex','Fontsize',fSize)
xlabel('$t[s]$','Interpreter','Latex','Fontsize',fSize);
set(gcf,'units','normalized','outerposition',[0 0 1 1]);
saveas(gcf, nome_1, format1);
saveas(gcf, nome_1, format2);
close(fig)

fig=figure('visible','off', 'DefaultAxesPosition', [0.064, 0.135, 0.921,
    0.845]);
plot(gt1,error1(:,2),':',gt2,error2(:,2),'r:',gt3,error3(:,2),'k','
    LineWidth',lSize);
grid on
grid minor
YL = get(gca, 'ylim');          %plot the vertical line
YR = YL(2) - YL(1);
YL = [YL(1) - 1000 * YR, YL(2) + 1000 * YR];
line([5, 5], YL, 'YLimInclude', 'off', 'Color','k','LineWidth',dvlsize);
pa = annotation('arrow');       % store the arrow information in pa
pa.Parent = gca;                % associate the arrow the the current
```

```
    axes
pa.X = [5 8];                   % the location of arrow
pa.Y = [1 1];
pa.LineWidth  = dhlsize;        % make the arrow bolder for the figure
pa.HeadWidth  = 20;
pa.HeadLength = 20;
text(5.05,1.3,'disturbance in action','Fontsize',fSize) % write a text on
     top of the arrow
h=legend({'$$\gamma_{W} = 0.5$$','$$\gamma_{W} = 5$$','$$\gamma_{W} = 50
    $$'});
set(h,'Interpreter','Latex','FontSize',fSize);
ylabel('$$\tilde{x}_{2}(t)$$','Interpreter','Latex','Fontsize',fSize)
xlabel('$t[s]$','Interpreter','Latex','Fontsize',fSize);
set(gcf,'units','normalized','outerposition',[0 0 1 1]);
saveas(gcf, nome_2, format1);
saveas(gcf, nome_2, format2);
close(fig)

fig=figure('visible','off', 'DefaultAxesPosition', [0.064, 0.135, 0.921,
    0.845]);
plot(gt1,error1(:,3),':',gt2,error2(:,3),'r:',gt3,error3(:,3),'k','
    LineWidth',lSize);
grid on
grid minor
YL = get(gca, 'ylim');          %plot the vertical line
YR = YL(2) - YL(1);
YL = [YL(1) - 1000 * YR, YL(2) + 1000 * YR];
line([5, 5], YL, 'YLimInclude', 'off', 'Color','k','LineWidth',dvlsize);
pa = annotation('arrow');       % store the arrow information in pa
pa.Parent = gca;                % associate the arrow the the current
    axes
pa.X = [5 8];                   % the location of arrow
pa.Y = [1 1];
pa.LineWidth  = dhlsize;        % make the arrow bolder for the figure
pa.HeadWidth  = 20;
pa.HeadLength = 20;
text(5.05,1.2,'disturbance in action','Fontsize',fSize) % write a text on
     top of the arrow
h=legend({'$$\gamma_{W} = 0.5$$','$$\gamma_{W} = 5$$','$$\gamma_{W} = 50
    $$'});
set(h,'Interpreter','Latex','FontSize',fSize);
ylabel('$$\tilde{x}_{3}(t)$$','Interpreter','Latex','Fontsize',fSize)
xlabel('$t[s]$','Interpreter','Latex','Fontsize',fSize);
set(gcf,'units','normalized','outerposition',[0 0 1 1]);
saveas(gcf, nome_3, format1);
saveas(gcf, nome_3, format2);
close(fig)
```

```
fig=figure('visible','off', 'DefaultAxesPosition', [0.064, 0.135, 0.921,
    0.845]);
plot(gt1,error1(:,4),':',gt2,error2(:,4),'r:',gt3,error3(:,4),'k','
    LineWidth',lSize);
grid on
grid minor
YL = get(gca, 'ylim');          %plot the vertical line
YR = YL(2) - YL(1);
YL = [YL(1) - 1000 * YR, YL(2) + 1000 * YR];
line([5, 5], YL, 'YLimInclude', 'off', 'Color','k','LineWidth',dvlsize);
pa = annotation('arrow');       % store the arrow information in pa
pa.Parent = gca;                % associate the arrow the the current
    axes
pa.X = [5 8];                   % the location of arrow
pa.Y = [1 1];
pa.LineWidth  = dhlsize;        % make the arrow bolder for the figure
pa.HeadWidth  = 20;
pa.HeadLength = 20;
text(5.05,1.2,'disturbance in action','Fontsize',fSize) % write a text on
     top of the arrow
h=legend({'$$\gamma_{W} = 0.5$$','$$\gamma_{W} = 5$$','$$\gamma_{W} = 50
    $$'});
set(h,'Interpreter','Latex','FontSize',fSize);
ylabel('$$\tilde{x}_{4}(t)$$','Interpreter','Latex','Fontsize',fSize)
xlabel('$t[s]$','Interpreter','Latex','Fontsize',fSize);
set(gcf,'units','normalized','outerposition',[0 0 1 1]);
saveas(gcf, nome_4, format1);
saveas(gcf, nome_4, format2);
close(fig)
```

A.1.3 Simulink plant used for simulations corresponding to Figures 3.7 - 3.9 and Tables 3.9 - 3.10

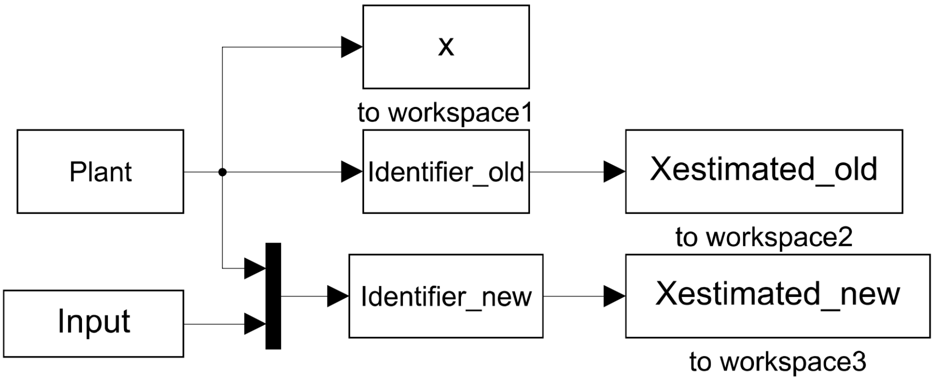

Figure A.2 – Simulink of comparison.

A.1.4 Codes used for simulations corresponding to Figures 3.7 - 3.9 and Tables 3.9 - 3.10

Listing A.6 – Plant.m

```
function [sys,x0,str,ts] = Plant(t,x,u,flag)

%Hypercaotic Finance System

%Constants for the hyperchaotic system
a=0.9;
b=0.2;
c=1.5;
d=0.2;
k=0.17;

switch flag,
   %%%%%%%%%%%%%%%%%%
   % Initialization %
   %%%%%%%%%%%%%%%%%%
case 0,
    sizes = simsizes;
    sizes.NumContStates  = 4;   %Number of continuous states
    sizes.NumDiscStates  = 0;   %Number of discret states
    sizes.NumOutputs     = 4;   %Number of outputs
    sizes.NumInputs      = 0;   %Number of inputs
    sizes.DirFeedthrough = 1;
    sizes.NumSampleTimes = 1;
    sys = simsizes(sizes);
    x0=[4 0 -0.5 2.5];          %Initial conditions
    str=[];
    ts=[0 0];
   %%%%%%%%%%%%%%%
   % Directives  %
   %%%%%%%%%%%%%%%
case 1,
   %Unified chaotic system implementation, the chosen constant generates
   %a Chen system.
   sys = [x(3) + (x(2) - a)*x(1) + x(4);
      1 - b*x(2) - x(1)*x(1);
     -x(1) - c*x(3);
     -d*x(1)*x(2) - k*x(4)] + disturb(x,u,t);

   %%%%%%%%%%
   % Output %
   %%%%%%%%%%
case 3,
   sys = x;
   %%%%%%%%%%%%%
   %    End    %
   %%%%%%%%%%%%%
case {2,4,9},
    sys = [];

 otherwise
   error(['unhandled flag = ',num2str(flag)]);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Summary: Return the system's perturbations
%Arguments: state vector, input vector and time
function disturb = disturb(x,u,t)
if t>=5
    disturb= 1000*[8*cos(9*t);
             10*cos(6*t);
       6*sin(15*t) + 10*exp(-t);
       6*sin(15*t) + 4*cos(20*t)];
else
    disturb=0; %Until t=5 secs the disturb is null
end
```

Listing A.7 – Identifier_old.m

```
%Project description: Online Identification using hidden layer neural
   networks
%                      with adaptive laws
%Authors: Jose Alfredo Ruiz Vargas and Emerson Grzeidak
%Date: 04/2015          Local: University of Brasilia
function [sys,x0,str,ts] = Identifier_old(t,x,u,flag)

%Diagonal matrix A with negative elements
A = [-30 0 0 0; 0 -29.5 0 0; 0 0 -32 0; 0 0 0 -30];

%Diagonal matrix A with positive elements
B = 150*[1.2 0 0 0; 0 0.9 0 0; 0 0 1.1 0; 0 0 0 0.8];

%Constant parameters for the learning laws
gammaW=0.01;

gamma0 = 0.1;
gamma1 = 1;
gamma2 = 0.0001;

%Initial weight to the output layer
W0 = [0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];
```

```
%Random generated matrix used for this simulation
%The matrix was generated using V0 = randn(4, 9)
%and the kept fixed for the simulation
V0 = [0.5377 0.3188 0.5784 0.7254 0.1241 0.6715 0.4889 0.2939 0.0689;
      0.8339 0.3077 0.7694 0.0631 0.4897 0.2075 0.0347 0.7873 0.8095;
      0.2588 0.4336 0.3499 0.7147 0.4090 0.7172 0.7269 0.8884 0.9443;
      0.8622 0.3426 0.0349 0.2050 0.4172 0.6302 0.3034 0.1471 0.4384];

%Positive definite matrix P and Matrix K
P = 60*[0.001 0 0 0; 0 0.001 0 0; 0 0 0.001 0; 0 0 0 0.001];
K = P + P';

%Parameters for the sigmoidal function
alpha=200;
beta=1;

switch flag,
   %%%%%%%%%%%%%%%%%%
   % Initialization %
   %%%%%%%%%%%%%%%%%%

case 0,

  sizes = simsizes;
    sizes.NumContStates  = 20;   %Number of continuous states
    sizes.NumDiscStates  = 0;    %Number of discrete states
    sizes.NumOutputs     = 25;   %Number of outputs
    sizes.NumInputs      = 4;    %Number of inputs
    sizes.DirFeedthrough = 1;
    sizes.NumSampleTimes = 1;
    sys = simsizes(sizes);
    x0=zeros(20,1);              %Initial conditions
    x0(1:4)=[-2 -2 -2 -2];     %Initial conditions for the estimated
   states
        str=[];
  ts=[0 0];
   %%%%%%%%%%%%%%%
   % Directives  %
   %%%%%%%%%%%%%%%

case 1,

    % Identification model and learning laws implementation
      sys = [A*x(1:4) + B*[x(5:8)'; x(9:12)'; x(13:16)'; x(17:20)']*Sig(x
   , u, alpha, beta, V0)-parameter_l(x, u, gamma0, gamma1, gamma2, K, t);
             -gammaW*(norm(x_error(x, u))*W_error(x, W0, 1) + w_term1(x,
   u, B, K, 1, alpha, beta, V0));
             -gammaW*(norm(x_error(x, u))*W_error(x, W0, 2) + w_term1(x,
   u, B, K, 2, alpha, beta, V0));
             -gammaW*(norm(x_error(x, u))*W_error(x, W0, 3) + w_term1(x,
   u, B, K, 3, alpha, beta, V0));
             -gammaW*(norm(x_error(x, u))*W_error(x, W0, 4) + w_term1(x,
   u, B, K, 4, alpha, beta, V0))];

   %%%%%%%%%%%%%%%
   % Outputs     %
   %%%%%%%%%%%%%%%

  case 3,
       sys = [x(1:20);
              norm(x(5:20));
              norm((u(1)-x(1)));
              norm((u(2)-x(2)));
              norm((u(3)-x(3)));
              norm((u(4)-x(4)))];
case {2,4,9},
    sys = [];
otherwise
   error(['unhandled flag = ',num2str(flag)]);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Summary: Return the line of the matrix of estimation errors for the
   parameter W
%Arguments: state vector, initial weights for W and the desired line

function W_error = W_error(x, W0, line)

%The state vector is in column format, your transposed is needed
%to mount the estimation matrix W
temp = [x(5:8)’; x(9:12)’; x(13:16)’; x(17:20)’] - W0;

if line == 1
    W_error = temp(1,:)’;
end
if line == 2
    W_error = temp(2,:)’;
end
if line == 3
    W_error = temp(3,:)’;
end
if line == 4
    W_error = temp(4,:)’;
```

```
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Summary: Return the vector function l
%Arguments: state vector, system inputs, matrix K, time t and control
%parameters: gamma0, gamma1, gamma2
function parameter_l = parameter_l(x, u, gamma0, gamma1, gamma2, K, t)

denominator = (min(eig(K))*(norm(x_error(x,u))+gamma1*exp(-gamma2*t)));
parameter_l = - (gamma0*x_error(x,u))/denominator;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Summary: Return a vector with estimated state's errors
%Arguments: state vector, system inputs

function x_error = x_error(x, u)

X = [x(1); x(2); x(3); x(4)];
U = [u(1); u(2); u(3); u(4)];

x_error = X - U;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Summary: Return the NN's nonlinear regressor vector
%Arguments: state vector, system inputs and sigmoidal function parameters

function Sig = Sig(x,u, alpha, beta, V0)     %Regressor

%Parameters for the activation function
V0Z = V0*[u(1); u(2); u(3); u(4); u(1)^2; u(2)^2; u(3)^2; u(4)^2; 1];

%A sigmoidal function is used, we pass each of the elements of the vector
Sig=[(z(V0Z(1), alpha, beta));
     (z(V0Z(2), alpha, beta));
     (z(V0Z(3), alpha, beta));
     (z(V0Z(4), alpha, beta))];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Summary: Sigmoidal fucntion that returns the results of the regressor
%Arguments: the product VZ and parameters for the sigmoidal function

function z = z(arg, alpha, beta)  %Sigmoidal activation function

z=alpha/(1+exp(-beta*arg));   %Sigmoidal Function

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Summary: Return the line of B*K*Xerror*S(VZ)
```

```
%Arguments: state vector, system inputs, matrix B and K,
%desired line of the matrix and parameters of sigmoidal function

function w_term1 = w_term1(x, u, B, K, line, alpha, beta, V0)

%term B*K*Xerror*S(VZ) 3x3
temp = B*K*x_error(x,u)*(Sig(x,u, alpha, beta, V0)’);

if line == 1
    w_term1 = temp(1, :)’;
end
if line == 2
    w_term1 = temp(2, :)’;
end
if line == 3
    w_term1 = temp(3, :)’;
end
if line == 4
    w_term1 = temp(4, :)’;
end
```

A.1.5 Codes used for simulations corresponding to Figure 3.7 and Table 3.9

Listing A.8 – Main.m

```
close all
%clear all
clc

format1 = ’jpeg’;
format2 = ’epsc’;
GAMAW1 = 0.5;
GAMAW2 = 5;
GAMAW3 = 50;

fSize = 38;
axesSize = 38;
lSize = 2;
dvlsize = 2;
dhlsize = 2;

addpath(’./Data/’);

Simulation_all

Graphs

serror1 = size(nerror2);
```

```
e_rms_1 = norm(nerror2)/sqrt(serror1(1));

fileID = fopen('./Tables/Table.txt','w');
fprintf(fileID,'Root Mean Square of State Errors\n');
fprintf(fileID,'e_{rms} proposed in paper\t %f\n', e_rms_1);
fclose(fileID);

save ./Saved_Data/Data.mat
clc
```

Listing A.9 – Simulation_all.m

```
model = './Data/Identifier_System.mdl';

Input = [0 GAMAW2];
open_system(model,'loadonly'); %load simulink model
modelname = 'Identifier_System';
set_param(modelname,'SolverType','Variable-step'...
                   ,'Solver','ode45'...
                   ,'RelTol','1e-10'...
                   ,'AbsTol','1e-10'...
                   ,'MaxStep','0.1'...
                   ,'StopTime','10'...
                   ,'ScreenColor','white'...
                   ,'TimeSaveName','t');
save_system('./Data/Identifier_System.mdl');
newsys = './Mdl_2018a/Identifier_System.mdl';
save_system(model,newsys,'ExportToVersion','R2018A_MDL');
sim(model);
clc
close_system(model, 0); %Close the simulink model, 1 to save, 0 to not
    save

gt2=t;
aux=size(t);
aux_aqui=size(t);
error2 = zeros(aux(1),4);
nerror2 = zeros(aux(1),1);
error2(:,1) = x(:,1) - Xestimated_new(:,1);
error2(:,2) = x(:,2) - Xestimated_new(:,2);
error2(:,3) = x(:,3) - Xestimated_new(:,3);
error2(:,4) = x(:,4) - Xestimated_new(:,4);

error_comp = zeros(aux(1),4);
nerror_comp = zeros(aux(1),1);
error_comp(:,1) = x(:,1) - Xestimated_old(:,1);
error_comp(:,2) = x(:,2) - Xestimated_old(:,2);
error_comp(:,3) = x(:,3) - Xestimated_old(:,3);
error_comp(:,4) = x(:,4) - Xestimated_old(:,4);

for i = 1:aux(1)
    a1 = error2(i,1);
    a2 = error2(i,2);
    a3 = error2(i,3);
    a4 = error2(i,4);
    a_M = [a1 a2 a3 a4];
    nerror2(i) = norm(a_M);
end

for i = 1:aux(1)
    a1 = error_comp(i,1);
    a2 = error_comp(i,2);
    a3 = error_comp(i,3);
    a4 = error_comp(i,4);
    a_M = [a1 a2 a3 a4];
    nerror_comp(i) = norm(a_M);
end
```

Listing A.10 – Identifier_new.m

```
function [sys,x0,str,ts] = Identifier_new(t,x,u,flag)

%Controller and its parameters
L = 70000*2*[1 0 0 0; 0 1 0 0;0 0 1 0; 0 0 0 1];
P = 70000*30*[1 0 0 0; 0 1 0 0;0 0 1 0; 0 0 0 1];
PSI=0.00005; %0.00005
G=1;
W01=G*[1 0 0 0 0 0 0 0]'; %W zero
W02=G*[0 1 0 0 0 0 0 0]';
W03=G*[0 0 1 0 0 0 0 0]';
W04=G*[0 0 0 1 0 0 0 0]';

switch flag,
   %%%%%%%%%%%%%%%%%%%
   % Initialization  %
   %%%%%%%%%%%%%%%%%%%
case 0,

  sizes = simsizes;
    sizes.NumContStates   = 36;  %Number of Constant States
    sizes.NumDiscStates   = 0;   %Number of Discret States
    sizes.NumOutputs      = 5;   %Number of Outputs
    sizes.NumInputs       = 5;   %Number of Inputs
    sizes.DirFeedthrough = 1;
    sizes.NumSampleTimes = 1;
    sys = simsizes(sizes);
    x0=zeros(36,1); %
    x0(1)=-2;
    x0(2)=-2;
    x0(3)=-2;
    x0(4)=-2;
        str=[];
  ts=[0 0];
   %%%%%%%%%%%%%%%
   % Directives  %
   %%%%%%%%%%%%%%%
case 1,
      %Identification Model
      sys = [-L*[x(1)-u(1);x(2)-u(2);x(3)-u(3);x(4)-u(4)] - u(5)*PSI*[x
   (1)-u(1);x(2)-u(2);x(3)-u(3);x(4)-u(4)] + P*[x(5:12)';x(13:20)';x
   (21:28)';x(29:36)']*S(x,u);
         %Learning Law
         -2*u(5)*(PSI*(x(5:12)-W01) + (x(1)-u(1))*S(x,u));
         -2*u(5)*(PSI*(x(13:20)-W02) + (x(2)-u(2))*S(x,u));
         -2*u(5)*(PSI*(x(21:28)-W03) + (x(3)-u(3))*S(x,u));
       -2*u(5)*(PSI*(x(29:36)-W04) + (x(4)-u(4))*S(x,u))];
   %%%%%%%%%%%
   % Outputs %
   %%%%%%%%%%%
  case 3,
    sys = [x(1:4);
           norm([x(5:12)';x(13:20)';x(21:28)';x(29:36)'],'fro')];
case {2,4,9},
    sys = [];

 otherwise
   error(['unhandled flag = ',num2str(flag)]);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function S = S(x,u)     %Regressors
S=[1*(z(u(1)));
   1*(z(u(2)));
   1*(z(u(3)));
   1*(z(u(4)));
   1*(z(u(1))^2);
   1*(z(u(2))^2);
   1*(z(u(3))^2);
   1*(z(u(4))^2)];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function z = z(uu)  %Sigmoidal Function
lambda=0;
alfa=5;
```

```
beta=.5;
z=alfa/(exp(-beta*uu)+1)+lambda;
```

Listing A.11 – Graphs.m

```
addpath('./Figures/');

nome_1 = './Figures/FIG_2_7';
set(0,'DefaultAxesFontSize',axesSize);

fig=figure('visible','off', 'DefaultAxesPosition', [0.055, 0.135, 0.93,
    0.845]);
plot(gt2,nerror_comp(:,1),gt2,nerror2(:,1),':','LineWidth',lSize);
grid on
grid minor
set(0,'DefaultAxesFontSize',axesSize);
YL = get(gca, 'ylim');          %plot the vertical line
YR = YL(2) - YL(1);
YL = [YL(1) - 1000 * YR, YL(2) + 1000 * YR];
line([5, 5], YL, 'YLimInclude', 'off', 'Color','k','LineWidth',dvlsize);
pa = annotation('arrow');       % store the arrow information in pa
pa.Parent = gca;                % associate the arrow the the current
    axes
pa.X = [5 8];                   % the location of arrow
pa.Y = [5 5];
pa.LineWidth  = dhlsize;        % make the arrow bolder for the figure
pa.HeadWidth  = 20;
pa.HeadLength = 20;
text(5.05,5.3,'disturbances in action','Fontsize',fSize) % write a text
    on top of the arrow
h=legend('Proposed in [74]','Proposed algorithm','Location','northeast');
set(h,'FontSize',fSize);
xlabel('$t[s]$','Interpreter','Latex','Fontsize',fSize);
ylabel('$$||\tilde{x}(t)||$$','Interpreter','Latex','Fontsize',fSize)
set(gcf,'units','normalized','outerposition',[0 0 1 1]);
saveas(gcf, nome_1, format1);
saveas(gcf, nome_1, format2);
close(fig)
```

A.1.6 Codes used for simulations corresponding to Figures 3.8 - 3.9 and Table 3.10

Listing A.12 – Main.m

```
close all
%clear all
clc
```

```
format1 = 'jpeg';
format2 = 'epsc';
GAMAW1 = 0.5;
GAMAW2 = 5;
GAMAW3 = 50;

fSize = 38;
axesSize = 38;
lSize = 2;
dvlsize = 2;
dhlsize = 2;

addpath('./Data/');

Simulation_all

Graphs

serror1 = size(nerror1);
serror3 = size(nerror3);

e_rms_1 = norm(nerror1)/sqrt(serror1(1));
e_rms_2 = norm(nerror3)/sqrt(serror3(1));

fileID = fopen('./Tables/Table.txt','w');
fprintf(fileID,'Root Mean Square of State Errors in the proposed
   algorithm\n');
fprintf(fileID,'e_{rms} in proposed in paper when gamma_W = 0.5 \t %f\n',
     e_rms_1);
fprintf(fileID,'e_{rms} in proposed in paper when gamma_W = 50 \t %f\n',
   e_rms_2);
fclose(fileID);

save ./Saved_Data/Data.mat
clc
```

Listing A.13 – Simulation_all.m

```
model = './Data/Identifier_System.mdl';

Input = [0 GAMAW1];
open_system(model,'loadonly'); %load simulink model
modelname = 'Identifier_System';
set_param(modelname,'SolverType','Variable-step'...
                   ,'Solver','ode45'...
                   ,'RelTol','1e-10'...
                   ,'AbsTol','1e-10'...
                   ,'MaxStep','0.1'...
```

```
                    ,'StopTime','10'...
                    ,'ScreenColor','white'...
                    ,'TimeSaveName','t');
save_system('./Data/Identifier_System.mdl');
newsys = './Mdl_2018a/Identifier_System.mdl';
save_system(model,newsys,'ExportToVersion','R2018A_MDL');
sim(model);
clc
close_system(model, 0); %Close the simulink model, 1 to save, 0 to not
    save

gt1=t;
aux=size(t);
aux_aqui=size(t);
error1 = zeros(aux(1),4);
nerror1 = zeros(aux(1),1);
error1(:,1) = x(:,1) - Xestimated_new(:,1);
error1(:,2) = x(:,2) - Xestimated_new(:,2);
error1(:,3) = x(:,3) - Xestimated_new(:,3);
error1(:,4) = x(:,4) - Xestimated_new(:,4);

error_comp1 = zeros(aux(1),4);
nerror_comp1 = zeros(aux(1),1);
error_comp1(:,1) = x(:,1) - Xestimated_old(:,1);
error_comp1(:,2) = x(:,2) - Xestimated_old(:,2);
error_comp1(:,3) = x(:,3) - Xestimated_old(:,3);
error_comp1(:,4) = x(:,4) - Xestimated_old(:,4);

for i = 1:aux(1)
    a1 = error1(i,1);
    a2 = error1(i,2);
    a3 = error1(i,3);
    a4 = error1(i,4);
    a_M = [a1 a2 a3 a4];
    nerror1(i) = norm(a_M);

    a1 = error_comp1(i,1);
    a2 = error_comp1(i,2);
    a3 = error_comp1(i,3);
    a4 = error_comp1(i,4);
    a_M = [a1 a2 a3 a4];
    nerror_comp1(i) = norm(a_M);
end

Input = [0 GAMAW3];
open_system(model,'loadonly'); %load simulink model
sim(model);
```

```
close_system(model, 0);

gt3 = t;
aux = size(t);
error3 = zeros(aux(1),4);
nerror3 = zeros(aux(1),1);
error3(:,1) = x(:,1) - Xestimated_new(:,1);
error3(:,2) = x(:,2) - Xestimated_new(:,2);
error3(:,3) = x(:,3) - Xestimated_new(:,3);
error3(:,4) = x(:,4) - Xestimated_new(:,4);

error_comp3 = zeros(aux(1),4);
nerror_comp3 = zeros(aux(1),1);
error_comp3(:,1) = x(:,1) - Xestimated_old(:,1);
error_comp3(:,2) = x(:,2) - Xestimated_old(:,2);
error_comp3(:,3) = x(:,3) - Xestimated_old(:,3);
error_comp3(:,4) = x(:,4) - Xestimated_old(:,4);
for i = 1:aux(1)
    a1 = error3(i,1);
    a2 = error3(i,2);
    a3 = error3(i,3);
    a4 = error3(i,4);
    a_M = [a1 a2 a3 a4];
    nerror3(i) = norm(a_M);

    a1 = error_comp3(i,1);
    a2 = error_comp3(i,2);
    a3 = error_comp3(i,3);
    a4 = error_comp3(i,4);
    a_M = [a1 a2 a3 a4];
    nerror_comp3(i) = norm(a_M);
end
```

Listing A.14 – Identifier_new.m

```
function [sys,x0,str,ts] = Identifier_new(t,x,u,flag)

%Controller and its parameters
L = 2*[1 0 0 0; 0 1 0 0;0 0 1 0; 0 0 0 1];
P = 30*[1 0 0 0; 0 1 0 0;0 0 1 0; 0 0 0 1];
PSI=1;
G=1;
W01=G*[1 0 0 0 0 0 0 0]'; %W zero
W02=G*[0 1 0 0 0 0 0 0]';
W03=G*[0 0 1 0 0 0 0 0]';
W04=G*[0 0 0 1 0 0 0 0]';

switch flag,
   %%%%%%%%%%%%%%%%%%%
   % Initialization  %
   %%%%%%%%%%%%%%%%%%%
case 0,

  sizes = simsizes;
    sizes.NumContStates  = 36;  %Number of Constant States
    sizes.NumDiscStates  = 0;   %Number of Discret States
    sizes.NumOutputs     = 5;   %Number of Outputs
    sizes.NumInputs      = 5;   %Number of Inputs
    sizes.DirFeedthrough = 1;
    sizes.NumSampleTimes = 1;
    sys = simsizes(sizes);

    x0=zeros(36,1); %
    x0(1)=-2;
    x0(2)=-2;
    x0(3)=-2;
    x0(4)=-2;
        str=[];
  ts=[0 0];
   %%%%%%%%%%%%%%%
   % Directives  %
   %%%%%%%%%%%%%%%
case 1,
      %Identification Model
      sys = [-L*[x(1)-u(1);x(2)-u(2);x(3)-u(3);x(4)-u(4)] - u(5)*PSI*[x
    (1)-u(1);x(2)-u(2);x(3)-u(3);x(4)-u(4)] + P*[x(5:12)';x(13:20)';x
    (21:28)';x(29:36)']*S(x,u);
         %Learning Law
         -2*u(5)*(PSI*(x(5:12)-W01) + (x(1)-u(1))*S(x,u));
         -2*u(5)*(PSI*(x(13:20)-W02) + (x(2)-u(2))*S(x,u));
         -2*u(5)*(PSI*(x(21:28)-W03) + (x(3)-u(3))*S(x,u));
       -2*u(5)*(PSI*(x(29:36)-W04) + (x(4)-u(4))*S(x,u))];
   %%%%%%%%%%%
   % Outputs %
   %%%%%%%%%%%
  case 3,
    sys = [x(1:4);
           norm([x(5:12)';x(13:20)';x(21:28)';x(29:36)'],'fro')];
case {2,4,9},
    sys = [];

 otherwise
   error(['unhandled flag = ',num2str(flag)]);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
function S = S(x,u)     %Regressors
S=[1*(z(u(1)));
   1*(z(u(2)));
   1*(z(u(3)));
   1*(z(u(4)));
   1*(z(u(1))^2);
   1*(z(u(2))^2);
   1*(z(u(3))^2);
   1*(z(u(4))^2)];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function z = z(uu)  %Sigmoidal Function
lambda=0;
alfa=5;
beta=.5;
z=alfa/(exp(-beta*uu)+1)+lambda;
```

Listing A.15 – Graphs.m

```
addpath('./Figures/');

nome_1 = './Figures/FIG_2_8';
nome_2 = './Figures/FIG_2_9';
set(0,'DefaultAxesFontSize',axesSize);

fig=figure('visible','off', 'DefaultAxesPosition', [0.055, 0.135, 0.93,
    0.845]);
plot(gt1,nerror_comp1(:,1),gt1,nerror1(:,1),':','LineWidth',lSize);
grid on
grid minor
YL = get(gca, 'ylim'); %plot the vertical line
YR = YL(2) - YL(1);
YL = [YL(1) - 1000 * YR, YL(2) + 1000 * YR];
line([5, 5], YL, 'YLimInclude', 'off', 'Color','k','LineWidth',dvlsize);
pa = annotation('arrow');  % store the arrow information in pa
pa.Parent = gca;           % associate the arrow the the current axes
pa.X = [5 8];             % the location of arrow
pa.Y = [2 2];
pa.LineWidth  = dhlsize;         % make the arrow bolder for the figure
pa.HeadWidth  = 20;
pa.HeadLength = 20;
text(5.05,2.3,'disturbances in action','Fontsize',fSize) % write a text
    on top of the arrow
h=legend('Proposed in [74]','Proposed algorithm','Location','northeast');
set(h,'FontSize',fSize);
xlabel('$t[s]$','Interpreter','Latex','Fontsize',fSize);
ylabel('$$||\tilde{x}(t)||$$','Interpreter','Latex','Fontsize',fSize)
dim = [.75 .43 .4 .4];
str = '$$\gamma_{w} = 0.5$$';
```

```
annotation('textbox',dim,'String',str,'Interpreter','Latex','Fontsize',
    fSize,'FitBoxToText','on');
set(gcf,'units','normalized','outerposition',[0 0 1 1]);
saveas(gcf, nome_1, format1);
saveas(gcf, nome_1, format2);
close(fig)

fig=figure('visible','off', 'DefaultAxesPosition', [0.055, 0.135, 0.93,
    0.845]);
plot(gt3,nerror_comp3(:,1),gt3,nerror3(:,1),':','LineWidth',lSize);
grid on
grid minor
YL = get(gca, 'ylim'); %plot the vertical line
YR = YL(2) - YL(1);
YL = [YL(1) - 1000 * YR, YL(2) + 1000 * YR];
line([5, 5], YL, 'YLimInclude', 'off', 'Color','k','LineWidth',dvlsize);
pa = annotation('arrow');  % store the arrow information in pa
pa.Parent = gca;           % associate the arrow the the current axes
pa.X = [5 8];            % the location of arrow
pa.Y = [2 2];
pa.LineWidth  = dhlsize;          % make the arrow bolder for the figure
pa.HeadWidth  = 20;
pa.HeadLength = 20;
text(5.05,2.3,'disturbances in action','Fontsize',fSize) % write a text
    on top of the arrow
h=legend('Proposed in [74]','Proposed algorithm','Location','northeast');
set(h,'FontSize',fSize);
xlabel('$t[s]$','Interpreter','Latex','Fontsize',fSize);
ylabel('$$||\tilde{x}(t)||$$','Interpreter','Latex','Fontsize',fSize)
dim = [.75 .43 .4 .4];
str = '$$\gamma_{w} = 50$$';
annotation('textbox',dim,'String',str,'Interpreter','Latex','Fontsize',
    fSize,'FitBoxToText','on');
set(gcf,'units','normalized','outerposition',[0 0 1 1]);
saveas(gcf, nome_2, format1);
saveas(gcf, nome_2, format2);
close(fig)
```

A.1.7 Simulink plant used for simulations corresponding to Figures 3.11 - 3.20 and Table 3.12

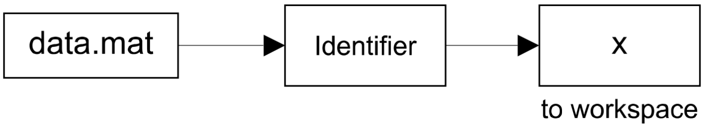

Figure A.3 – Simulink of welding system.

A.1.8 Codes used for simulations corresponding to Figures 3.10 - 3.20 and Table 3.12

Listing A.16 – Main.m

```
close all
%clear all
clc

addpath('./Data/');
addpath('./Figures/');

valor = 1547;

data = xlsread('Table.xlsx');
vect = zeros(valor,3);
vect(:,1) = data(1:valor,3); %Velocidade de arame (m/min} (entrada)
vect(:,2) = data(1:valor,5); %tensão malha aberta (V) (entrada)
vect(:,3) = data(1:valor,4); %velocidade de soldagem (mm/s) (entrada)

y = zeros(valor,8);
y(:,1) = data(1:valor,2)/1000; %tempo (s)
y(:,2) = data(1:valor,6); %corrente (A)
y(:,3) = data(1:valor,8); %Longitude de arco (mm)
y(:,4) = data(1:valor,7); %tensão de arco (V)
y(:,5) = data(1:valor,9); %stick out (mm)
y(:,6) = data(1:valor,10); %largura(mm)
y(:,7) = data(1:valor,11); %reforço(mm)
y(:,8) = data(1:valor,12); %penetracao(mm)

z = y';
save ./Data/data.mat z

format1 = 'jpeg';
format2 = 'epsc';
fSize = 38;
axesSize = 38;
lSize = 2;

model = './Data/Welding_Idenfication_System.mdl';

open_system(model,'loadonly'); %load simulink model
modelname = 'Welding_Idenfication_System';
set_param(modelname,'SolverType','Variable-step'...
                   ,'Solver','ode45'...
                   ,'RelTol','1e-10'...
                   ,'AbsTol','1e-10'...
                   ,’MaxStep’,’0.1’...
                   ,’StopTime’,’15.47’...
                   ,’ScreenColor’,’white’...
                   ,’TimeSaveName’,’t’);
save_system(’./Data/Welding_Idenfication_System.mdl’);
newsys = ’./Mdl_2018a/Welding_Idenfication_System.mdl’;
save_system(model,newsys,’ExportToVersion’,’R2018A_MDL’);
clc
close_system(model, 0); %Close the simulink model, 1 to save, 0 to not
    save

Simulation

Graphs

srror1 = size(error(:,1));
serror2 = size(error(:,2));
serror3 = size(error(:,3));
serror4 = size(error(:,4));
serror5 = size(error(:,5));
serror6 = size(error(:,6));
serror7 = size(error(:,7));

e_rms1 = norm(error(:,1))/sqrt(serror1(1));
e_rms2 = norm(error(:,2))/sqrt(serror2(1));
e_rms3 = norm(error(:,3))/sqrt(serror3(1));
e_rms4 = norm(error(:,4))/sqrt(serror4(1));
e_rms5 = norm(error(:,5))/sqrt(serror5(1));
e_rms6 = norm(error(:,6))/sqrt(serror6(1));

fileID = fopen(’./Tables/Table.txt’,’w’);
fprintf(fileID,’Root Mean Square of State Errors\n’);
fprintf(fileID,’e_{rms} proposed in paper (A)\t %f\n’, e_rms1);
fprintf(fileID,’e_{rms} proposed in paper (V)\t %f\n’, e_rms2);
fprintf(fileID,’e_{rms} proposed in paper (mm)\t %f\n’, e_rms3);
fprintf(fileID,’e_{rms} proposed in paper (mm)\t %f\n’, e_rms4);
fprintf(fileID,’e_{rms} proposed in paper (mm)\t %f\n’, e_rms5);
fprintf(fileID,’e_{rms} proposed in paper (mm)\t %f\n’, e_rms6);
fclose(fileID);

save ./Saved_Data/Data.mat
clc
```

Listing A.17 – Simulation.m

```
open_system(model,’loadonly’); %load simulink model
sim(model);
close_system(model, 0); %Close the simulink model, 1 to save, 0 to not
```

```
   save

aux=size(x);
error = zeros(aux(1),7);
error(:,1) = x(:,8) - x(:,1);
%error(:,2) = x(:,9) - x(:,2);
error(:,2) = x(:,10) - x(:,3);
error(:,3) = x(:,11) - x(:,4);
error(:,4) = x(:,12) - x(:,5);
error(:,5) = x(:,13) - x(:,6);
error(:,6) = x(:,14) - x(:,7);
```

Listing A.18 – Identifier.m

```
function [sys,x0,str,ts] = Identifier(t,x,u,flag)

%Identifier and its parameters
L = 100*[5 0 0 0 0 0 0; 0 4 0 0 0 0 0; 0 0 10 0 0 0 0;...
    0 0 0 4 0 0 0; 0 0 0 0 1.1 0 0; 0 0 0 0 0 1 0; 0 0 0 0 0 0 1];
P = L;
GAMAW = 0.001;
PSI = 0.001;
G = 1;
W01 = G*[1 0 0 0 0 0 0 0 0 0 0 0 0 0]'; %W zero
W02 = G*[0 1 0 0 0 0 0 0 0 0 0 0 0 0]';
W03 = G*[0 0 1 0 0 0 0 0 0 0 0 0 0 0]';
W04 = G*[0 0 0 1 0 0 0 0 0 0 0 0 0 0]';
W05 = G*[0 0 0 0 1 0 0 0 0 0 0 0 0 0]';
W06 = G*[0 0 0 0 0 1 0 0 0 0 0 0 0 0]';
W07 = G*[0 0 0 0 0 1 0 0 0 0 0 0 0 0]';

switch flag,
   %%%%%%%%%%%%%%%%%%
   % Initialization %
   %%%%%%%%%%%%%%%%%%
case 0,

  sizes = simsizes;
    sizes.NumContStates  = 105;  %Number of Constant States
    sizes.NumDiscStates  = 0;   %Number of Discret States
    sizes.NumOutputs     = 15;   %Number of Outputs
    sizes.NumInputs      = 7;   %Number of Inputs
    sizes.DirFeedthrough = 1;
    sizes.NumSampleTimes = 1;
    sys = simsizes(sizes);
    x0=zeros(105,1); %Initial Conditions
    x0(4,1) = 0.01;
        str=[];
  ts=[0 0];
   %%%%%%%%%%%%%%%
   % Directives  %
   %%%%%%%%%%%%%%%
case 1,
        %Identifier Model
      sys = [-L*[x(1)-u(1);x(2)-u(2);x(3)-u(3);x(4)-u(4);x(5)-u(5);x(6)-u
    (6);x(7)-u(7)]...
          - GAMAW*PSI*[x(1)-u(1);x(2)-u(2);x(3)-u(3);x(4)-u(4);x(5)-u(5);
   x(6)-u(6);x(7)-u(7)]...
          + P*[x(8:21)';x(22:35)';x(36:49)';x(50:63)';x(64:77)';x(78:91)
   ';x(92:105)']*S(x,u);
         %Learning Law
         -2*GAMAW*(PSI*(x(8:21)-W01) + (x(1)-u(1))*S(x,u));
         -2*GAMAW*(PSI*(x(22:35)-W02) + (x(2)-u(2))*S(x,u));
         -2*GAMAW*(PSI*(x(36:49)-W03) + (x(3)-u(3))*S(x,u));
         -2*GAMAW*(PSI*(x(50:63)-W04) + (x(4)-u(4))*S(x,u));
         -2*GAMAW*(PSI*(x(64:77)-W05) + (x(5)-u(5))*S(x,u));
         -2*GAMAW*(PSI*(x(78:91)-W06) + (x(6)-u(6))*S(x,u));
         -2*GAMAW*(PSI*(x(92:105)-W07) + (x(6)-u(6))*S(x,u))];
   %%%%%%%%%%%
   % Outputs %
   %%%%%%%%%%%
  case 3,
    sys = [u(1:7);x(1:7);
           norm([x(8:21)';x(22:35)';x(36:49)';x(50:63)';x(64:77)';x
    (78:91)';x(92:105)'],'fro')];
case {2,4,9},
    sys = [];

 otherwise
   error(['unhandled flag = ',num2str(flag)]);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function S = S(x,u)     %Regressors
S=[1*(z(x(1)));
   1*(z(x(2)));
   1*(z(x(3)));
   1*(z(x(4)));
   1*(z(x(5)));
   1*(z(x(6)));
   1*(z(x(7)));
   1*z(x(1))^2;
   1*z(x(2))^2;
   1*z(x(3))^2;
   1*z(x(4))^2;
   1*z(x(5))^2;
   1*z(x(6))^2
   1*z(x(7))^2];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function z = z(uu)   %Sigmoidal Function
lamda=0;
alfa=5;
beta=.5;
z=alfa/(exp(-beta*uu)+1)+lamda;
```

Listing A.19 – Graphs.m

```
addpath('./Figures/');

nome_1 = './Figures/FIG_2_11';
nome_2 = './Figures/FIG_2_12';
nome_3 = './Figures/FIG_2_13';
nome_4 = './Figures/FIG_2_14';
nome_5 = './Figures/FIG_2_15';
nome_6 = './Figures/FIG_2_16';
nome_7 = './Figures/FIG_2_17';
nome_8 = './Figures/FIG_2_18';
nome_9 = './Figures/FIG_2_19';
nome_10 = './Figures/FIG_2_20';
set(0,'DefaultAxesFontSize',axesSize);

fig=figure('visible','off', 'DefaultAxesPosition', [0.056, 0.135, 0.941,
   0.845]);
plot(y(:,1),vect(:,1),'LineWidth',lSize);
grid on
grid minor
h=legend('Wire Speed','Location','northeast');
set(h,'FontSize',fSize);
ylabel('$u_1[m/min]$','Interpreter','Latex','Fontsize',fSize);
xlim([0, 15.47]);
xlabel('$t[s]$','Interpreter','Latex','Fontsize',fSize);
set(gcf,'units','normalized','outerposition',[0 0 1 1]);
saveas(gcf, nome_1, format1);
saveas(gcf, nome_1, format2);
close(fig)

fig=figure('visible','off', 'DefaultAxesPosition', [0.069, 0.135, 0.928,
   0.845]);
plot(y(:,1),vect(:,2),'LineWidth',lSize);
grid on
grid minor
h=legend('Open Loop Voltage','Location','northeast');
set(h,'FontSize',fSize);
ylabel('$u_2[V]$','Interpreter','Latex','Fontsize',fSize);
```

```
xlim([0, 15.47]);
xlabel('$t[s]$','Interpreter','Latex','Fontsize',fSize);
set(gcf,'units','normalized','outerposition',[0 0 1 1]);
saveas(gcf, nome_2, format1);
saveas(gcf, nome_2, format2);
close(fig)

fig=figure('visible','off', 'DefaultAxesPosition', [0.069, 0.135, 0.927,
    0.845]);
plot(y(:,1),vect(:,3),'LineWidth',lSize);
grid on
grid minor
h=legend('Welding Speed','Location','northeast');
set(h,'FontSize',fSize);
ylabel('$u_3[mm/s]$','Interpreter','Latex','Fontsize',fSize);
xlim([0 15.47]);
ylim([6 15]);
xlabel('$t[s]$','Interpreter','Latex','Fontsize',fSize);
set(gcf,'units','normalized','outerposition',[0 0 1 1]);
saveas(gcf, nome_3, format1);
saveas(gcf, nome_3, format2);
close(fig)

fig=figure('visible','off', 'DefaultAxesPosition', [0.085, 0.135, 0.912,
    0.845]);
plot(t,x(:,1),t, x(:,8),'r:','LineWidth',lSize);
grid on
grid minor
h=legend('Actual','Estimated','Location','northeast');
set(h,'FontSize',fSize);
ylabel('$x_1[A]$','Interpreter','Latex','Fontsize',fSize);
xlim([0 15.47]);
xlabel('$t[s]$','Interpreter','Latex','Fontsize',fSize);
set(gcf,'units','normalized','outerposition',[0 0 1 1]);
saveas(gcf, nome_4, format1);
saveas(gcf, nome_4, format2);
close(fig)

fig=figure('visible','off', 'DefaultAxesPosition', [0.069, 0.135, 0.928,
    0.845]);
plot(t,x(:,3),t, x(:,10),'r:','LineWidth',lSize);
grid on
grid minor
h=legend('Actual','Estimated','Location','northeast');
set(h,'FontSize',fSize);
ylabel('$x_2[V]$','Interpreter','Latex','Fontsize',fSize);
xlim([0 15.47]);
```

```
xlabel(’$t[s]$’,’Interpreter’,’Latex’,’Fontsize’,fSize);
set(gcf,’units’,’normalized’,’outerposition’,[0 0 1 1]);
saveas(gcf, nome_5, format1);
saveas(gcf, nome_5, format2);
close(fig)

fig=figure(’visible’,’off’, ’DefaultAxesPosition’, [0.069, 0.135, 0.928,
    0.845]);
plot(t,x(:,4),t, x(:,11),’r:’,’LineWidth’,lSize);
grid on
grid minor
h=legend(’Actual’,’Estimated’,’Location’,’northeast’);
set(h,’FontSize’,fSize);
ylabel(’$x_3[mm]$’,’Interpreter’,’Latex’,’Fontsize’,fSize);
xlim([0 15.47]);
ylim([0 16]);
xlabel(’$t[s]$’,’Interpreter’,’Latex’,’Fontsize’,fSize);
set(gcf,’units’,’normalized’,’outerposition’,[0 0 1 1]);
saveas(gcf, nome_6, format1);
saveas(gcf, nome_6, format2);
close(fig)

fig=figure(’visible’,’off’, ’DefaultAxesPosition’, [0.069, 0.135, 0.927,
    0.845]);
plot(t,x(:,5),t, x(:,12),’r:’,’LineWidth’,lSize);
grid on
grid minor
h=legend(’Actual’,’Estimated’,’Location’,’northeast’);
set(h,’FontSize’,fSize);
ylabel(’$x_4[mm]$’,’Interpreter’,’Latex’,’Fontsize’,fSize);
xlim([0 15.47]);
xlabel(’$t[s]$’,’Interpreter’,’Latex’,’Fontsize’,fSize);
set(gcf,’units’,’normalized’,’outerposition’,[0 0 1 1]);
saveas(gcf, nome_7, format1);
saveas(gcf, nome_7, format2);
close(fig)

fig=figure(’visible’,’off’, ’DefaultAxesPosition’, [0.077, 0.135, 0.921,
    0.845]);
plot(t,x(:,6),t, x(:,13),’r:’,’LineWidth’,lSize);
grid on
grid minor
h=legend(’Actual’,’Estimated’,’Location’,’northeast’);
set(h,’FontSize’,fSize);
ylabel(’$x_5[mm]$’,’Interpreter’,’Latex’,’Fontsize’,fSize);
xlim([0 15.47]);
xlabel(’$t[s]$’,’Interpreter’,’Latex’,’Fontsize’,fSize);
```

```
set(gcf,'units','normalized','outerposition',[0 0 1 1]);
saveas(gcf, nome_8, format1);
saveas(gcf, nome_8, format2);
close(fig)

fig=figure('visible','off', 'DefaultAxesPosition', [0.077, 0.135, 0.921,
    0.845]);
plot(t,x(:,7),t, x(:,14),'r:','LineWidth',lSize);
grid on
grid minor
h=legend('Actual','Estimated','Location','northeast');
set(h,'FontSize',fSize);
ylabel('$x_6[mm]$','Interpreter','Latex','Fontsize',fSize);
xlim([0 15.47]);
ylim([-0.2 3]);
xlabel('$t[s]$','Interpreter','Latex','Fontsize',fSize);
set(gcf,'units','normalized','outerposition',[0 0 1 1]);
saveas(gcf, nome_9, format1);
saveas(gcf, nome_9, format2);
close(fig)

fig=figure('visible','off', 'DefaultAxesPosition', [0.096, 0.135, 0.9015,
     0.845]);
plot(t,x(:,15),'LineWidth',lSize);
grid on
grid minor
h=legend('Estimated Weigth Norm','Location','northeast');
set(h,'FontSize',fSize);
ylabel('$||\tilde{W}(t)||$','Interpreter','Latex','Fontsize',fSize);
xlim([0 15.47]);
xlabel('$t[s]$','Interpreter','Latex','Fontsize',fSize);
set(gcf,'units','normalized','outerposition',[0 0 1 1]);
saveas(gcf, nome_10, format1);
saveas(gcf, nome_10, format2);
close(fig)
```

A.2 CODES FOR SIMULATIONS IN CHAPTER 4

A.2.1 Codes used for simulations corresponding to Figures 4.3 - 4.12 and Table 4.1

Listing A.20 – Main.m

```
clear   %limpa variáveis anteriores
clc     %limpa o que estava escrito no terminal
%Condições iniciais, única regra essencial é que os estados do mestre e
    escravo sejam diferentes.
```

```
x0_mestre =  [0.2 -0.3 0.4];
x0_escravo = [0.1 0.3 -0.1];
psi = 50;      %Ganho do controle subatuado
%IMPORTANTE: Coloque a mensagem igual a zero nos estados atuados, pois
   esses estados
%são importantes para a correta sincronização dos sistemas mestre e
   escravo
t_fim = 20;     %Tempo em que a simulação termina

options = odeset('RelTol', 1e-10,...    %configurações da simulação
                 'AbsTol', 1e-10,...
                 'MaxStep',0.001);
t_ciclo = [0 t_fim ];   %coloca-se o tempo inicial e final da simulação
   em um vetor
x0(1) = x0_mestre (1);  %repassando as condições iniciais para variáveis
   escalonadas
x0(2) = x0_mestre (2);
x0(3) = x0_mestre (3);
x0(4) = x0_escravo(1);
x0(5) = x0_escravo(2);
x0(6) = x0_escravo(3);

[t, x] = ode45(@Esquema, t_ciclo, x0, options, psi);
%realiza a simulação e salva os resultados
aux = size(t);
msg = zeros(aux(1), 3);

for i = 1:aux(1)
    tempo = t(i,1);
    msg_aux = Mensagens(tempo);
    msg(i,:) = msg_aux';
end
Xmaster_sem_msg = x(:, 1:3);
Xmaster = x(:, 1:3) + msg(:, 1:3);
Xslave = x(:, 4:6);
clearvars -except t Xmaster_sem_msg Xmaster Xslave msg

e1 = Xslave(:,1) - Xmaster_sem_msg(:,1);
e2 = Xslave(:,2) - Xmaster_sem_msg(:,2);
e3 = Xslave(:,3) - Xmaster_sem_msg(:,3);

merror1 = Xmaster(:,1) - Xslave(:,1) - msg(:,1);
merror2 = Xmaster(:,2) - Xslave(:,2) - msg(:,2);
merror3 = Xmaster(:,3) - Xslave(:,3) - msg(:,3);

%Graphs
```

```
e = [e1 e2 e3];
serror1 = size(e1);
serror2 = size(e2);
serror3 = size(e3);
serror = size(e);

fileID = fopen('./Tables/Table.txt','w');
fprintf(fileID,'Root Mean Square of State Errors\n\n');
fprintf(fileID,'e_1_{rms} \t %f \n', norm(e1)/sqrt(serror1(1)));
fprintf(fileID,'e_2_{rms} \t %f \n', norm(e2)/sqrt(serror2(1)));
fprintf(fileID,'e_3_{rms} \t %f \n', norm(e3)/sqrt(serror3(1)));
fprintf(fileID,'e_{rms} \t %f \n\n', norm(e)/sqrt(serror(1)));
fclose(fileID);

save ./Saved_Data/Data.mat
clc

function y = Esquema(t, x, psi)
eq_mestre =  Sistema(x(1), x(2), x(3));    %equação do sistema mestre
eq_escravo = Sistema(x(4), x(5), x(6));   %equação sistema escravo
y(1:3, 1) = eq_mestre;
y(4:6, 1) = eq_escravo + controle(x, psi) + disturb(t);
%O resultado dos sistema mestre e escravo
end

function controle = controle(x, psi)
controle = [0;                            %estrutura do controle
            -psi*(x(5) - x(2));
            0];
end

function disturb = disturb(t)   %distúrbios, caso queira coloca-los
if t>=0                         %Início dos distúrbios a partir de 5 segundos
    disturb = [0.1*sin(2*t);
               0.2*sin(t);
               0.1*sin(4*t)];
else
    disturb = 0; %Até t=10 segundos os distúrbios são nulos
end
end
```

Listing A.21 – Sistema.m

```
function equation = Sistema(x, y, z)

a = 16; %Constantes
b = 45.6;
```

```
c = 4;
d = 5;

%coloque aqui a estrutura do seu sistema dinâmico
equation =  [a*(y - x);
             b*x - y - 4*d*x*z;
             -c*z + d*x*y];
end
```

Listing A.22 – Mensagens.m

```
function msg = Mensagens(t)
msg = [0.1*sin(8*3.1416*t);   %coloque aqui a estrutura do seu sistema
   dinâmico
       0; %0.15*square(0.5*t) + 0.05*sin(4*t)
       0.05*cos(4*t) + 0.05*cos(t)];
end
%Mensagens enviadas (devem ser no maximo 5% do valor maximo atingido pelo
    estado)
%Entenda como msg1 uma mensagem presente no primeiro estado, por exemplo
%Não coloque mensagens em estados onde o controle esta presente

%IMPORTANTE: Coloque a mensagem igual a zero nos estados atuados, pois
   esses estados
%são importantes para a correta sincronização dos sistemas mestre e
   escravo
```

Listing A.23 – Graphs.m

```
format1 = 'jpeg';
format2 = 'epsc';

fSize = 38;
axesSize = 38;
lSize = 2;
dvlsize = 2;
dhlsize = 2;
fonte = 38;
largura_linha = 2;
color1 = [0 0.4470 0.7410];
color2 = [0.8500 0.3250 0.0980];
color3 = [0.4660 0.6740 0.1880];

nome_1 = './Figures/FIG_3_3';
nome_2 = './Figures/FIG_3_4';
nome_3 = './Figures/FIG_3_5';
nome_4 = './Figures/FIG_3_6';
nome_5 = './Figures/FIG_3_7';
```

```
nome_6 = ’./Figures/FIG_3_8’;
nome_7 = ’./Figures/FIG_3_9’;
nome_8 = ’./Figures/FIG_3_10’;
nome_9 = ’./Figures/FIG_3_11’;
nome_10 = ’./Figures/FIG_3_12’;

set(0,’DefaultAxesFontSize’,axesSize);

fig=figure(’visible’,’off’, ’DefaultAxesPosition’, [0.033, 0.135, 0.952,
    0.845]);
plot(t,Xmaster_sem_msg(:,1),t, Xslave(:,1),’:’,’LineWidth’,lSize);
grid on
grid minor
h=legend(’$x_m(t)$’,’$x_s(t)$’,’Location’,’northeast’);
set(h,’Interpreter’,’Latex’,’FontSize’,fSize);
xlabel(’$t[s]$’,’Interpreter’,’Latex’,’Fontsize’,fSize);
ylim([-4 4])
set(gcf,’units’,’normalized’,’outerposition’,[0 0 1 1]);
saveas(gcf, nome_1, format1);
saveas(gcf, nome_1, format2);
close(fig)

fig=figure(’visible’,’off’, ’DefaultAxesPosition’, [0.033, 0.135, 0.952,
    0.845]);
plot(t,Xmaster_sem_msg(:,2),t, Xslave(:,2),’:’,’LineWidth’,lSize);
grid on
grid minor
h=legend(’$y_m(t)$’,’$y_s(t)$’,’Location’,’northeast’);
set(h,’Interpreter’,’Latex’,’FontSize’,fSize);
xlabel(’$t[s]$’,’Interpreter’,’Latex’,’Fontsize’,fSize);
ylim([-5 5])
set(gcf,’units’,’normalized’,’outerposition’,[0 0 1 1]);
saveas(gcf, nome_2, format1);
saveas(gcf, nome_2, format2);
close(fig)

fig=figure(’visible’,’off’, ’DefaultAxesPosition’, [0.025, 0.135, 0.96,
    0.845]);
plot(t,Xmaster_sem_msg(:,3),t, Xslave(:,3),’:’,’LineWidth’,lSize);
grid on
grid minor
h=legend(’$z_m(t)$’,’$z_s(t)$’,’Location’,’northeast’);
set(h,’Interpreter’,’Latex’,’FontSize’,fSize);
xlabel(’$t[s]$’,’Interpreter’,’Latex’,’Fontsize’,fSize);
ylim([-0.5 4.5])
set(gcf,’units’,’normalized’,’outerposition’,[0 0 1 1]);
saveas(gcf, nome_3, format1);
saveas(gcf, nome_3, format2);
close(fig)

fig=figure(’visible’,’off’, ’DefaultAxesPosition’, [0.068, 0.135, 0.917,
    0.845]);
plot(t,e1,’LineWidth’,lSize);
grid on
grid minor
h=legend(’$e_1(t)$’,’Location’,’northeast’);
set(h,’Interpreter’,’Latex’,’FontSize’,fSize);
xlabel(’$t[s]$’,’Interpreter’,’Latex’,’Fontsize’,fSize);
set(gcf,’units’,’normalized’,’outerposition’,[0 0 1 1]);
saveas(gcf, nome_4, format1);
saveas(gcf, nome_4, format2);
close(fig)

fig=figure(’visible’,’off’, ’DefaultAxesPosition’, [0.054, 0.135, 0.931,
    0.845]);
plot(t,e2,’LineWidth’,lSize);
grid on
grid minor
h=legend(’$e_2(t)$’,’Location’,’northeast’);
set(h,’Interpreter’,’Latex’,’FontSize’,fSize);
xlabel(’$t[s]$’,’Interpreter’,’Latex’,’Fontsize’,fSize);
set(gcf,’units’,’normalized’,’outerposition’,[0 0 1 1]);
saveas(gcf, nome_5, format1);
saveas(gcf, nome_5, format2);
close(fig)

fig=figure(’visible’,’off’, ’DefaultAxesPosition’, [0.054, 0.135, 0.931,
    0.845]);
plot(t,e3,’LineWidth’,lSize);
grid on
grid minor
h=legend(’$e_3(t)$’,’Location’,’northeast’);
set(h,’Interpreter’,’Latex’,’FontSize’,fSize);
xlabel(’$t[s]$’,’Interpreter’,’Latex’,’Fontsize’,fSize);
set(gcf,’units’,’normalized’,’outerposition’,[0 0 1 1]);
saveas(gcf, nome_6, format1);
saveas(gcf, nome_6, format2);
close(fig)

fig=figure(’visible’,’off’, ’DefaultAxesPosition’, [0.1, 0.135, 0.885,
    0.845]);
aux = Xmaster(:,1) - Xslave(:,1);
plot(t,msg(:,1),’-’,’Color’,color1,’LineWidth’,largura_linha);
grid on
grid minor
hold on;
plot(t, aux,’:’,’Color’,color2,’LineWidth’, largura_linha);
plot(t, 0.05*Xmaster(:,1),’-’,’Color’,color3,’LineWidth’, largura_linha);
ylim([-0.16 0.25])
h=legend(’Original message’, ’Decoded message’, ’Encoded message’,’
    Location’,’northeast’);
set(h,’FontSize’,fonte);
xlabel(’$$t[s]$$’,’Interpreter’,’Latex’,’Fontsize’,fonte);
ylabel(’$$m_1(t), \hat{m}_1(t), 0.05{\cdot}s_1(t)$$’,’Interpreter’,’Latex
    ’,’Fontsize’,fonte);
set(gcf,’units’,’normalized’,’outerposition’,[0 0 1 1]);
saveas(gcf, nome_7, format1);
saveas(gcf, nome_7, format2);
close(fig)

fig=figure(’visible’,’off’, ’DefaultAxesPosition’, [0.086, 0.135, 0.899,
    0.845]);
aux = Xmaster(:,3) - Xslave(:,3);
plot(t,msg(:,3),’-’,’Color’,color1,’LineWidth’,largura_linha);
grid on
grid minor
hold on;
plot(t, aux,’:’,’Color’,color2,’LineWidth’, largura_linha);
plot(t, 0.05*Xmaster(:,3) - 0.1,’-’,’Color’,color3,’LineWidth’,
    largura_linha);
ylim([-0.15 0.65])
h=legend(’Original message’, ’Decoded message’, ’Encoded message’,’
    Location’,’northeast’);
set(h,’FontSize’,fonte);
xlabel(’$$t[s]$$’,’Interpreter’,’Latex’,’Fontsize’,fonte);
ylabel(’$$m_3(t), \hat{m}_3(t), 0.05{\cdot}s_3(t) - 0.1$$’,’Interpreter’,
    ’Latex’,’Fontsize’,fonte);
set(gcf,’units’,’normalized’,’outerposition’,[0 0 1 1]);
saveas(gcf, nome_8, format1);
saveas(gcf, nome_8, format2);
close(fig)

fig=figure(’visible’,’off’, ’DefaultAxesPosition’, [0.068, 0.135, 0.917,
    0.845]);
plot(t,merror1,’LineWidth’, largura_linha);
grid on
grid minor
ylim([-0.08 0.12])
h=legend(’$$\tilde{m}_1(t)$$’,’Interpreter’,’Latex’,’Location’,’northeast
    ’);
set(h,’FontSize’,fonte);
```

```
xlabel(’$$t[s]$$’,’Interpreter’,’Latex’,’Fontsize’,fonte);
set(gcf,’units’,’normalized’,’outerposition’,[0 0 1 1]);
saveas(gcf, nome_9, format1);
saveas(gcf, nome_9, format2);
close(fig)

fig=figure(’visible’,’off’, ’DefaultAxesPosition’, [0.053, 0.135, 0.932,
    0.845]);
plot(t,merror3,’LineWidth’, largura_linha);
grid on
grid minor
ylim([-0.1 0.6])
h=legend(’$$\tilde{m}_3(t)$$’,’Interpreter’,’Latex’,’Location’,’northeast
    ’);
set(h,’FontSize’,fonte);
xlabel(’$$t[s]$$’,’Interpreter’,’Latex’,’Fontsize’,fonte);
set(gcf,’units’,’normalized’,’outerposition’,[0 0 1 1]);
saveas(gcf, nome_10, format1);
saveas(gcf, nome_10, format2);
close(fig)
```

A.3 CODES FOR SIMULATIONS IN CHAPTER 5

A.3.1 Codes used for simulations corresponding to Figures 5.4 - 5.13 and Table 5.1

Listing A.24 – Main.m

```
clear   %limpa variáveis anteriores
clc     %limpa o que estava escrito no terminal
%Condições iniciais, única regra essencial é que os estados do mestre e
    escravo sejam diferentes.

%Escalonamento de Amplitude
x_fator = 1;%20;       %\bar{x} = x/x_fator , sendo "\bar{x}" o novo
    valor e "x" o valor antigo
y_fator = 1;%20;       %\bar{x} = x/x_fator , sendo "\bar{x}" o novo
    valor e "x" o valor antigo
z_fator = 1;%20;       %\bar{x} = x/x_fator , sendo "\bar{x}" o novo
    valor e "x" o valor antigo
%Escalonamento Frequência
freq_fator = 1;    %Quanto maior, mais rapido acontece a simulação

x0_mestre =  [0.1 0.1 0.1]; %x0_mestre =  [0.1 0.1 0.1];
x0_escravo = [0.2 0.1 0.2];
psi = 100;      %Ganho do controle subatuado
%IMPORTANTE: Coloque a mensagem igual a zero nos estados atuados, pois
```

```
    esses estados
%são importantes para a correta sincronização dos sistemas mestre e
    escravo
t_fim = 0.1;      %Tempo em que a simulação termina

options = odeset('RelTol', 1e-10,...    %configurações da simulação
                 'AbsTol', 1e-10,...
                 'MaxStep',0.001);
t_ciclo = [0 t_fim ];   %coloca-se o tempo inicial e final da simulação
    em um vetor

amp_f(1) = x_fator;
amp_f(2) = y_fator;
amp_f(3) = z_fator;

x0(1) = x0_mestre (1)/amp_f(1);  %repassando as condições iniciais para
    variáveis escalonadas
x0(2) = x0_mestre (2)/amp_f(2);
x0(3) = x0_mestre (3)/amp_f(3);
x0(4) = x0_escravo(1)/amp_f(1);
x0(5) = x0_escravo(2)/amp_f(2);
x0(6) = x0_escravo(3)/amp_f(3);

[t, x] = ode45(@Esquema, t_ciclo, x0, options, psi, amp_f, freq_fator);
%realiza a simulação e salva os resultados
aux = size(t);
msg = zeros(aux(1), 3);

for i = 1:aux(1)
    tempo = t(i,1);
    msg_aux = Mensagens(tempo);
    msg(i,:) = msg_aux';
end
Xmaster_sem_msg = x(:, 1:3);
Xmaster = x(:, 1:3) + msg(:, 1:3);
Xslave = x(:, 4:6);
clearvars -except t Xmaster_sem_msg Xmaster Xslave msg

e1 = Xslave(:,1) - Xmaster_sem_msg(:,1);
e2 = Xslave(:,2) - Xmaster_sem_msg(:,2);
e3 = Xslave(:,3) - Xmaster_sem_msg(:,3);

merror1 = Xmaster(:,1) - Xslave(:,1) - msg(:,1);
merror2 = Xmaster(:,2) - Xslave(:,2) - msg(:,2);
merror3 = Xmaster(:,3) - Xslave(:,3) - msg(:,3);

Graphs
```

```
e = [e1 e2 e3];
serror1 = size(e1);
serror2 = size(e2);
serror3 = size(e3);
serror = size(e);

fileID = fopen('./Tables/Table.txt','w');
fprintf(fileID,'Root Mean Square of State Errors\n\n');
fprintf(fileID,'e_1_{rms} \t %f \n', norm(e1)/sqrt(serror1(1)));
fprintf(fileID,'e_2_{rms} \t %f \n', norm(e2)/sqrt(serror2(1)));
fprintf(fileID,'e_3_{rms} \t %f \n', norm(e3)/sqrt(serror3(1)));
fprintf(fileID,'e_{rms} \t %f \n\n', norm(e)/sqrt(serror(1)));
fclose(fileID);

save ./Saved_Data/Data.mat
clc

function y = Esquema(t, x, psi, amp_f, freq_fator)
eq_mestre =  Sistema(amp_f(1)*x(1),...
                     amp_f(2)*x(2),...
                     amp_f(3)*x(3));    %equação do sistema mestre
eq_escravo =  Sistema(amp_f(1)*x(4),...
                      amp_f(2)*x(5),...
                      amp_f(3)*x(6));   %equação sistema escravo
eq_mestre(1) = freq_fator*eq_mestre(1)/amp_f(1);
eq_mestre(2) = freq_fator*eq_mestre(2)/amp_f(2);
eq_mestre(3) = freq_fator*eq_mestre(3)/amp_f(3);

eq_escravo(1) = eq_escravo(1)/amp_f(1);
eq_escravo(2) = eq_escravo(2)/amp_f(2);
eq_escravo(3) = eq_escravo(3)/amp_f(3);
y(1:3, 1) = eq_mestre;
y(4:6, 1) = freq_fator*eq_escravo + 1000*controle(x, psi) + disturb(t);
%O resultado dos sistema mestre e escravo
end

function controle = controle(x, psi)
controle = [0;                            %estrutura do controle
            -psi*(x(5) - x(2));
            0];
end

function disturb = disturb(t)   %distúrbios, caso queira coloca-los
if t>=0                         %Início dos distúrbios a partir de 5
   segundos
    disturb = [0.3*sin(2*t) + 0.1*sin(20*t);
```

```
                0.4*sin(t) + 0.1*cos(2*t);
                0.3*square(t) + 0.2*sin(4*t)];
else
    disturb = 0; %Até t=10 segundos os distúrbios são nulos
end
end
```

Listing A.25 – Sistema.m

```
function equation = Sistema(x, y, z)

a = 2.1;  %Constantes
b = 0.6;
c = 30;

equation =  1000*[a*(y - x);    %coloque aqui a estrutura do seu sistema
   dinâmico
                  c*x - y - 20*x*z;
                  -b*z + 20*x*y];
end
```

Listing A.26 – Mensagens.m

```
function msg = Mensagens(t)
tt = 1000*t;
msg = 0.2*[0.1*square(tt);   %coloque aqui a estrutura do seu sistema din
   âmico
            0; %0.2*sin(tt);% + 0.05*sin(4*tt)
            0.1*sawtooth(tt,0.5)];%sin(4*tt) + 0.05*sin(tt)];
end
%Mensagens enviadas (devem ser no maximo 5% do valor maximo atingido pelo
    estado)
%Entenda como msg1 uma mensagem presente no primeiro estado, por exemplo
%Não coloque mensagens em estados onde o controle esta presente

%IMPORTANTE: Coloque a mensagem igual a zero nos estados atuados, pois
   esses estados
%são importantes para a correta sincronização dos sistemas mestre e
   escravo
```

Listing A.27 – Graphs.m

```
format1 = 'jpeg';
format2 = 'epsc';

fSize = 38;
axesSize = 38;
lSize = 2;
dvlsize = 2;
```

```
dhlsize = 2;
fonte = 38;
largura_linha = 2;
color1 = [0 0.4470 0.7410];
color2 = [0.8500 0.3250 0.0980];
color3 = [0.4660 0.6740 0.1880];

nome_1 = './Figures/FIG_4_4';
nome_2 = './Figures/FIG_4_5';
nome_3 = './Figures/FIG_4_6';
nome_4 = './Figures/FIG_4_7';
nome_5 = './Figures/FIG_4_8';
nome_6 = './Figures/FIG_4_9';
nome_7 = './Figures/FIG_4_10';
nome_8 = './Figures/FIG_4_11';
nome_9 = './Figures/FIG_4_12';
nome_10 = './Figures/FIG_4_13';

set(0,'DefaultAxesFontSize',axesSize);

fig=figure('visible','off', 'DefaultAxesPosition', [0.053, 0.135, 0.932,
    0.86]);
plot(t,Xmaster_sem_msg(:,1),t, Xslave(:,1),':','LineWidth',lSize);
grid on
grid minor
h=legend('$x_m(t)$','$x_s(t)$','Location','northeast');
set(h,'Interpreter','Latex','FontSize',fSize);
xlabel('$t[s]$','Interpreter','Latex','Fontsize',fSize);
ylim([-0.45 0.65]);
set(gcf,'units','normalized','outerposition',[0 0 1 1]);
saveas(gcf, nome_1, format1);
saveas(gcf, nome_1, format2);
close(fig)

fig=figure('visible','off', 'DefaultAxesPosition', [0.053, 0.135, 0.932,
    0.86]);
plot(t,Xmaster_sem_msg(:,2),t, Xslave(:,2),':','LineWidth',lSize);
grid on
grid minor
h=legend('$y_m(t)$','$y_s(t)$','Location','northeast');
set(h,'Interpreter','Latex','FontSize',fSize);
xlabel('$t[s]$','Interpreter','Latex','Fontsize',fSize);
ylim([-0.9 1.3]);
set(gcf,'units','normalized','outerposition',[0 0 1 1]);
saveas(gcf, nome_2, format1);
saveas(gcf, nome_2, format2);
close(fig)
```

```
fig=figure(’visible’,’off’, ’DefaultAxesPosition’, [0.045, 0.135, 0.94,
    0.845]);
plot(t,Xmaster_sem_msg(:,3),t, Xslave(:,3),’:’,’LineWidth’,lSize);
grid on
grid minor
h=legend(’$z_m(t)$’,’$z_s(t)$’,’Location’,’northeast’);
set(h,’Interpreter’,’Latex’,’FontSize’,fSize);
xlabel(’$t[s]$’,’Interpreter’,’Latex’,’Fontsize’,fSize);
ylim([0 3]);
set(gcf,’units’,’normalized’,’outerposition’,[0 0 1 1]);
saveas(gcf, nome_3, format1);
saveas(gcf, nome_3, format2);
close(fig)

fig=figure(’visible’,’off’, ’DefaultAxesPosition’, [0.06, 0.135, 0.925,
    0.86]);
plot(t,e1,’LineWidth’,lSize);
grid on
grid minor
h=legend(’$e_1(t)$’,’Location’,’northeast’);
set(h,’Interpreter’,’Latex’,’FontSize’,fSize);
xlabel(’$t[s]$’,’Interpreter’,’Latex’,’Fontsize’,fSize);
ylim([-0.005 0.105]);
set(gcf,’units’,’normalized’,’outerposition’,[0 0 1 1]);
saveas(gcf, nome_4, format1);
saveas(gcf, nome_4, format2);
close(fig)

fig=figure(’visible’,’off’, ’DefaultAxesPosition’, [0.068, 0.135, 0.917,
    0.86]);
plot(t,e2,’LineWidth’,lSize);
grid on
grid minor
h=legend(’$e_2(t)$’,’Location’,’northeast’);
set(h,’Interpreter’,’Latex’,’FontSize’,fSize);
xlabel(’$t[s]$’,’Interpreter’,’Latex’,’Fontsize’,fSize);
ylim([-0.052 0.022]);
set(gcf,’units’,’normalized’,’outerposition’,[0 0 1 1]);
saveas(gcf, nome_5, format1);
saveas(gcf, nome_5, format2);
close(fig)

fig=figure(’visible’,’off’, ’DefaultAxesPosition’, [0.045, 0.135, 0.94,
    0.845]);
plot(t,e3,’LineWidth’,lSize);
grid on
grid minor
h=legend(’$e_3(t)$’,’Location’,’northeast’);
set(h,’Interpreter’,’Latex’,’FontSize’,fSize);
xlabel(’$t[s]$’,’Interpreter’,’Latex’,’Fontsize’,fSize);
ylim([-0.02 0.4]);
set(gcf,’units’,’normalized’,’outerposition’,[0 0 1 1]);
saveas(gcf, nome_6, format1);
saveas(gcf, nome_6, format2);
close(fig)

fig=figure(’visible’,’off’, ’DefaultAxesPosition’, [0.1, 0.135, 0.885,
    0.86]);
aux = Xmaster(:,1) - Xslave(:,1);
plot(t,msg(:,1),’-’,’Color’,color1,’LineWidth’,largura_linha);
grid on
grid minor
hold on;
plot(t, aux,’:’,’Color’,color2,’LineWidth’, largura_linha);
plot(t, 0.05*Xmaster(:,1),’-’,’Color’,color3,’LineWidth’, largura_linha);
ylim([-0.085 0.06])
h=legend(’Original message’, ’Decoded message’, ’Encoded message’,’
    Location’,’northeast’);
set(h,’FontSize’,fonte);
xlabel(’$$t[s]$$’,’Interpreter’,’Latex’,’Fontsize’,fonte);
ylabel(’$$m_1(t), \hat{m}_1(t), 0.05{\cdot}s_1(t)$$’,’Interpreter’,’Latex
    ’,’Fontsize’,fonte);
set(gcf,’units’,’normalized’,’outerposition’,[0 0 1 1]);
saveas(gcf, nome_7, format1);
saveas(gcf, nome_7, format2);
close(fig)

fig=figure(’visible’,’off’, ’DefaultAxesPosition’, [0.086, 0.135, 0.899,
    0.845]);
aux = Xmaster(:,3) - Xslave(:,3);
plot(t,msg(:,3),’-’,’Color’,color1,’LineWidth’,largura_linha);
grid on
grid minor
hold on;
plot(t, aux,’:’,’Color’,color2,’LineWidth’, largura_linha);
plot(t, 0.05*Xmaster(:,3) - 0.07,’-’,’Color’,color3,’LineWidth’,
    largura_linha);
ylim([-0.45 0.2])
h=legend(’Original message’, ’Decoded message’, ’Encoded message’,’
    Location’,’northeast’);
set(h,’FontSize’,fonte);
xlabel(’$$t[s]$$’,’Interpreter’,’Latex’,’Fontsize’,fonte);
ylabel(’$$m_3(t), \hat{m}_3(t), 0.05{\cdot}s_3(t) - 0.07$$’,’Interpreter’
    ,'Latex','Fontsize',fonte);
set(gcf,'units','normalized','outerposition',[0 0 1 1]);
saveas(gcf, nome_8, format1);
saveas(gcf, nome_8, format2);
close(fig)

fig=figure('visible','off', 'DefaultAxesPosition', [0.068, 0.135, 0.917,
    0.845]);
plot(t,merror1,'LineWidth', largura_linha);
grid on
grid minor
h=legend('$$\tilde{m}_1(t)$$','Interpreter','Latex','Location','northeast
    ');
set(h,'FontSize',fonte);
xlabel('$$t[s]$$','Interpreter','Latex','Fontsize',fonte);
ylim([-0.105 0.015]);
set(gcf,'units','normalized','outerposition',[0 0 1 1]);
saveas(gcf, nome_9, format1);
saveas(gcf, nome_9, format2);
close(fig)

fig=figure('visible','off', 'DefaultAxesPosition', [0.053, 0.135, 0.932,
    0.86]);
plot(t,merror3,'LineWidth', largura_linha);
grid on
grid minor
ylim([-0.4 0.05]);
h=legend('$$\tilde{m}_3(t)$$','Interpreter','Latex','Location','northeast
    ');
set(h,'FontSize',fonte);
xlabel('$$t[s]$$','Interpreter','Latex','Fontsize',fonte);
set(gcf,'units','normalized','outerposition',[0 0 1 1]);
saveas(gcf, nome_10, format1);
saveas(gcf, nome_10, format2);
close(fig)
```

A.3.2 Codes used for simulations corresponding to Figures 5.14 - 5.17 and Table 5.2

Listing A.28 – Main.m

```
clc     %limpa o que estava escrito no terminal
model = 'Main_system.slx';

%proj = openProject("D:/Doutorado/Tese de doutorado/
    Tese_Doutorado_180007220_2021/Chapter_4/Simulation2/");
open_system(model,'loadonly'); %load simulink model
```

```
modelname = 'Main_system';
set_param(modelname,'SolverType','Variable-step'...
                   ,'Solver','ode23t'...
                   ,'RelTol','1e-8'...
                   ,'AbsTol','1e-8'...
                   ,'MaxStep','0.01'...
                   ,'StopTime','0.1'...
                   ,'ScreenColor','white'...
                   ,'TimeSaveName','t');
save_system('Main_system.slx');
%sim(model); %Descomente para simular
close_system(model, 0); %Close the simulink model, 1 to save, 0 to not
    save

e1 = X_m - X_s;
e2 = Y_m - Y_s;
e3 = Z_m - Z_s;

merror = Z_m - Z_s - m;

e = [e1 e2 e3];
serror1 = size(e1);
serror2 = size(e2);
serror3 = size(e3);
serror = size(e);

fileID = fopen('./Tables/Table.txt','w');
fprintf(fileID,'Root Mean Square of State Errors\n\n');
fprintf(fileID,'e_1_{rms} \t %f \n', norm(e1)/sqrt(serror1(1)));
fprintf(fileID,'e_2_{rms} \t %f \n', norm(e2)/sqrt(serror2(1)));
fprintf(fileID,'e_3_{rms} \t %f \n', norm(e3)/sqrt(serror3(1)));
fprintf(fileID,'e_{rms} \t %f \n\n', norm(e)/sqrt(serror(1)));
fclose(fileID);

save ./Saved_Data/Data.mat
Graphs

clc
%realiza a simulação e salva os resultados
```

Listing A.29 – Graphs.m

```
format1 = 'jpeg';
format2 = 'epsc';

fSize = 38;
axesSize = 38;
lSize = 2;
```

```
dvlsize = 2;
dhlsize = 2;
fonte = 38;
largura_linha = 2;
color1 = [0 0.4470 0.7410];
color2 = [0.8500 0.3250 0.0980];
color3 = [0.4660 0.6740 0.1880];

nome_1 = ’./Figures/FIG_4_14’;
nome_2 = ’./Figures/FIG_4_15’;
nome_3 = ’./Figures/FIG_4_16’;
nome_4 = ’./Figures/FIG_4_17’;

set(0,’DefaultAxesFontSize’,axesSize);

fig=figure(’visible’,’off’, ’DefaultAxesPosition’, [0.045, 0.135, 0.94,
    0.845]);
plot(t,Z_m,t, Z_s,’:’,’LineWidth’,lSize);
grid on
grid minor
h=legend(’$z_m(t)$’,’$z_s(t)$’,’Location’,’northeast’);
set(h,’Interpreter’,’Latex’,’FontSize’,fSize);
xlabel(’$t[s]$’,’Interpreter’,’Latex’,’Fontsize’,fSize);
ylim([-0.1 3]);
set(gcf,’units’,’normalized’,’outerposition’,[0 0 1 1]);
saveas(gcf, nome_1, format1);
saveas(gcf, nome_1, format2);
close(fig)

fig=figure(’visible’,’off’, ’DefaultAxesPosition’, [0.07, 0.135, 0.915,
    0.845]);
plot(t,e3,’LineWidth’,lSize);
grid on
grid minor
h=legend(’$e_3(t)$’,’Location’,’northeast’);
set(h,’Interpreter’,’Latex’,’FontSize’,fSize);
xlabel(’$t[s]$’,’Interpreter’,’Latex’,’Fontsize’,fSize);
ylim([-0.23 0.05]);
set(gcf,’units’,’normalized’,’outerposition’,[0 0 1 1]);
saveas(gcf, nome_2, format1);
saveas(gcf, nome_2, format2);
close(fig)

fig=figure(’visible’,’off’, ’DefaultAxesPosition’, [0.088, 0.135, 0.897,
    0.86]);
aux = Z_m + m;
plot(t, m,’-’,’Color’,color1,’LineWidth’,largura_linha);
```

```
grid on
grid minor
hold on;
plot(t, m_hat,':','Color',color2,'LineWidth', largura_linha);
plot(t, 0.5*aux - 0.7,'-','Color',color3,'LineWidth', largura_linha);
ylim([-1 1.2])
h=legend('Original message', 'Decoded message', 'Encoded message','
    Location','northeast');
set(h,'FontSize',fonte);
xlabel('$t[s]$','Interpreter','Latex','Fontsize',fonte);
ylabel('$m_3(t), \hat{m}_3(t), 0.5{\cdot}s_3(t) - 0.7$','Interpreter','
    Latex','Fontsize',fonte);
set(gcf,'units','normalized','outerposition',[0 0 1 1]);
saveas(gcf, nome_3, format1);
saveas(gcf, nome_3, format2);
close(fig)

fig=figure('visible','off', 'DefaultAxesPosition', [0.055, 0.135, 0.93,
    0.86]);
plot(t,merror,'LineWidth', largura_linha);
grid on
grid minor
h=legend('$\tilde{m}_3(t)$','Interpreter','Latex','Location','northeast')
    ;
set(h,'FontSize',fonte);
xlabel('$t[s]$','Interpreter','Latex','Fontsize',fonte);
ylim([-0.55 0.45]);
set(gcf,'units','normalized','outerposition',[0 0 1 1]);
saveas(gcf, nome_4, format1);
saveas(gcf, nome_4, format2);
close(fig)
```

A.3.3 Codes used for simulations corresponding to Figures 6.3 - 6.22 and Table 6.1

Listing A.30 – Main.m

```
clear   %limpa variáveis anteriores
clc     %limpa o que estava escrito no terminal

x0_mestre =  [0.01 0.01 0 0.01];
x0_escravo = [-0.2 -0.2 0.1 0];

psi = 60;      %Ganho do controle subatuado
%IMPORTANTE: Coloque a mensagem igual a zero nos estados atuados, pois
    esses estados
%são importantes para a correta sincronização dos sistemas mestre e
```

```
   escravo
t_fim = 10;      %Tempo em que a simulação termina

options = odeset('RelTol', 1e-10,...    %configurações da simulação
                 'AbsTol', 1e-10,...
                 'MaxStep',0.001);

t_ciclo = [0 t_fim];   %coloca-se o tempo inicial e final da simulação em
     um vetor

x0(1) = x0_mestre (1);  %repassando as condições iniciais para variáveis
   escalonadas
x0(2) = x0_mestre (2);
x0(3) = x0_mestre (3);
x0(4) = x0_mestre (4);
x0(5) = x0_escravo(1);
x0(6) = x0_escravo(2);
x0(7) = x0_escravo(3);
x0(8) = x0_escravo(4);

[t, x] = ode45(@Esquema, t_ciclo, x0, options, psi);
%realiza a simulação e salva os resultados
aux = size(t);
msg = zeros(aux(1), 4);

for i = 1:aux(1)
    tempo = t(i,1);
    msg_aux = Mensagens(tempo);
    msg(i,:) = msg_aux';
end
Xmaster_sem_msg = x(:, 1:4);
Xmaster = x(:, 1:4) + msg(:, 1:4);
Xslave = x(:, 5:8);
clearvars -except t Xmaster_sem_msg Xmaster Xslave msg

e1 = Xslave(:,1) - Xmaster_sem_msg(:,1);
e2 = Xslave(:,2) - Xmaster_sem_msg(:,2);
e3 = Xslave(:,3) - Xmaster_sem_msg(:,3);
e4 = Xslave(:,4) - Xmaster_sem_msg(:,4);

e1_m = Xslave(:,1) - Xmaster(:,1);
e2_m = Xslave(:,2) - Xmaster(:,2);
e3_m = Xslave(:,3) - Xmaster(:,3);
e4_m = Xslave(:,4) - Xmaster(:,4);

merror1 = Xmaster(:,1) - Xslave(:,1) - msg(:,1);
merror2 = Xmaster(:,2) - Xslave(:,2) - msg(:,2);
```

```
merror3 = Xmaster(:,3) - Xslave(:,3) - msg(:,3);
merror4 = Xmaster(:,4) - Xslave(:,4) - msg(:,4);
%Repassou-se os resultados para vetores mais simples de serem impressos
   em gráficos
%Xmaster_sem_msg como o nome sugere é o estado mestre sem a presença da
   mensagem
%Xmaster é a mensagem codificada ou criptografada (com a presença da
   mensagem)

e = [e1 e2 e3 e4];
serror1 = size(e1);
serror2 = size(e2);
serror3 = size(e3);
serror4 = size(e4);
serror = size(e);

fileID = fopen('./Tables/Table.txt','w');
fprintf(fileID,'Root Mean Square of State Errors\n\n');
fprintf(fileID,'e_1_{rms} \t %f \n', norm(e1)/sqrt(serror1(1)));
fprintf(fileID,'e_2_{rms} \t %f \n', norm(e2)/sqrt(serror2(1)));
fprintf(fileID,'e_3_{rms} \t %f \n', norm(e3)/sqrt(serror3(1)));
fprintf(fileID,'e_4_{rms} \t %f \n', norm(e4)/sqrt(serror4(1)));
fprintf(fileID,'e_{rms} \t %f \n\n', norm(e)/sqrt(serror(1)));
fclose(fileID);

%clearvars -except t Xmaster_sem_msg Xmaster Xslave msg
save ./Saved_Data/Data.mat
%Graphs
clc

function y = Esquema(t, x, psi)

eq_mestre =  Sistema(x(1),...
                     x(2),...
                     x(3),...
                     x(4));    %equação do sistema mestre
eq_escravo =  Sistema(x(5),...
                      x(6),...
                      x(7),...
                      x(8));   %equação sistema escravo
y(1:4, 1) = eq_mestre;
y(5:8, 1) = eq_escravo + controle(x, psi) + disturb(t);
%O resultado dos sistema mestre e escravo

end

function controle = controle(x, psi)
```

```
controle = [0;                            %estrutura do controle
            -1*(psi*(x(6) - x(2)));
            0;
            0];
end

function disturb = disturb(t)   %distúrbios, caso queira coloca-los
if t>=0                        %Início dos distúrbios a partir de 5
   segundos
    disturb = 0.02*[sin(5*t);
                    1.1*cos(3*t);
                    0.8*sin(3*t);
                    0.1*sin(6*t)];
end
end
```

Listing A.31 – Sistema.m

```
function equation = Sistema(x, y, z, w)
a = 15;
b = 5;
c = 10;
d = 1;
equation =  [a*(y - x);    %coloque aqui a estrutura do seu sistema dinâ
   mico
             c*y - 25*x*z + w;
             25*x*y - b*z;
             z - d*w];
end
```

Listing A.32 – Mensagens.m

```
function msg = Mensagens(t)
msg = 0.04*[0.5*sin(3*t) + 0.2*sin(20*t);   %coloque aqui a estrutura do
   seu sistema dinâmico
            0; %0.6*sin(2*t) + 0.2*cos(15*t) + 0.1*cos(30*t)
            0.6*sin(5*t) + 0.3*sin(15*t);
            0.05*square(5*t) + 0.01*sin(20*t)];
end
%Mensagens enviadas (devem ser no maximo 5% do valor maximo atingido pelo
    estado)
%Entenda como msg1 uma mensagem presente no primeiro estado, por exemplo
%Não coloque mensagens em estados onde o controle esta presente

%IMPORTANTE: Coloque a mensagem igual a zero nos estados atuados, pois
   esses estados
%são importantes para a correta sincronização dos sistemas mestre e
   escravo
```

Listing A.33 – Graphs.m

```
format1 = 'jpeg';
format2 = 'epsc';

fSize = 38;
axesSize = 38;
lSize = 2;
dvlsize = 2;
dhlsize = 2;
fonte = 38;
largura_linha = 2;
color1 = [0 0.4470 0.7410];
color2 = [0.8500 0.3250 0.0980];
color3 = [0.4660 0.6740 0.1880];

nome_1 = './Figures/FIG_5_2';
nome_2 = './Figures/FIG_5_3';
nome_3 = './Figures/FIG_5_4';
nome_4 = './Figures/FIG_5_5';
nome_5 = './Figures/FIG_5_6';
nome_6 = './Figures/FIG_5_7';
nome_7 = './Figures/FIG_5_8';
nome_8 = './Figures/FIG_5_9';
nome_9 = './Figures/FIG_5_10';
nome_10 = './Figures/FIG_5_11';
nome_11 = './Figures/FIG_5_12';
nome_12 = './Figures/FIG_5_13';
nome_13 = './Figures/FIG_5_14';
nome_14 = './Figures/FIG_5_15';
nome_15 = './Figures/FIG_5_16';
nome_16 = './Figures/FIG_5_17';
nome_17 = './Figures/FIG_5_18';
nome_18 = './Figures/FIG_5_19';
nome_19 = './Figures/FIG_5_20';
nome_20 = './Figures/FIG_5_21';

set(0,'DefaultAxesFontSize',axesSize);

fig=figure('visible','off', 'DefaultAxesPosition', [0.055, 0.135, 0.93,
   0.845]);
plot(t,Xmaster_sem_msg(:,1),t, Xslave(:,1),':','LineWidth',lSize);
grid on
grid minor
h=legend('$x_m(t)$','$x_s(t)$','Location','northeast');
set(h,'Interpreter','Latex','FontSize',fSize);
xlabel('$t[s]$','Interpreter','Latex','Fontsize',fSize);
ylim([-0.44 0.8]);
```

```
set(gcf,’units’,’normalized’,’outerposition’,[0 0 1 1]);
saveas(gcf, nome_1, format1);
saveas(gcf, nome_1, format2);
close(fig)

fig=figure(’visible’,’off’, ’DefaultAxesPosition’, [0.055, 0.135, 0.93,
    0.86]);
plot(t,Xmaster_sem_msg(:,2),t, Xslave(:,2),’:’,’LineWidth’,lSize);
grid on
grid minor
h=legend(’$y_m(t)$’,’$y_s(t)$’,’Location’,’northeast’);
set(h,’Interpreter’,’Latex’,’FontSize’,fSize);
xlabel(’$t[s]$’,’Interpreter’,’Latex’,’Fontsize’,fSize);
ylim([-0.48 0.92]);
set(gcf,’units’,’normalized’,’outerposition’,[0 0 1 1]);
saveas(gcf, nome_2, format1);
saveas(gcf, nome_2, format2);
close(fig)

fig=figure(’visible’,’off’, ’DefaultAxesPosition’, [0.045, 0.135, 0.94,
    0.86]);
plot(t,Xmaster_sem_msg(:,3),t, Xslave(:,3),’:’,’LineWidth’,lSize);
grid on
grid minor
h=legend(’$z_m(t)$’,’$z_s(t)$’,’Location’,’northeast’);
set(h,’Interpreter’,’Latex’,’FontSize’,fSize);
xlabel(’$t[s]$’,’Interpreter’,’Latex’,’Fontsize’,fSize);
ylim([-0.04 1.04]);
set(gcf,’units’,’normalized’,’outerposition’,[0 0 1 1]);
saveas(gcf, nome_3, format1);
saveas(gcf, nome_3, format2);
close(fig)

fig=figure(’visible’,’off’, ’DefaultAxesPosition’, [0.045, 0.135, 0.94,
    0.86]);
plot(t,Xmaster_sem_msg(:,4),t, Xslave(:,4),’:’,’LineWidth’,lSize);
grid on
grid minor
h=legend(’$w_m(t)$’,’$w_s(t)$’,’Location’,’northeast’);
set(h,’Interpreter’,’Latex’,’FontSize’,fSize);
xlabel(’$t[s]$’,’Interpreter’,’Latex’,’Fontsize’,fSize);
ylim([-0.02 0.54]);
set(gcf,’units’,’normalized’,’outerposition’,[0 0 1 1]);
saveas(gcf, nome_4, format1);
saveas(gcf, nome_4, format2);
close(fig)
```

```
fig=figure('visible','off', 'DefaultAxesPosition', [0.07, 0.135, 0.915,
    0.845]);
plot(t,e1,'LineWidth',lSize);
grid on
grid minor
h=legend('$e_1(t)$','Location','northeast');
set(h,'Interpreter','Latex','FontSize',fSize);
xlabel('$t[s]$','Interpreter','Latex','Fontsize',fSize);
ylim([-0.22 0.03]);
set(gcf,'units','normalized','outerposition',[0 0 1 1]);
saveas(gcf, nome_5, format1);
saveas(gcf, nome_5, format2);
close(fig)

fig=figure('visible','off', 'DefaultAxesPosition', [0.07, 0.135, 0.915,
    0.86]);
plot(t,e2,'LineWidth',lSize);
grid on
grid minor
h=legend('$e_2(t)$','Location','northeast');
set(h,'Interpreter','Latex','FontSize',fSize);
xlabel('$t[s]$','Interpreter','Latex','Fontsize',fSize);
ylim([-0.22 0.025]);
set(gcf,'units','normalized','outerposition',[0 0 1 1]);
saveas(gcf, nome_6, format1);
saveas(gcf, nome_6, format2);
close(fig)

fig=figure('visible','off', 'DefaultAxesPosition', [0.061, 0.135, 0.924,
    0.86]);
plot(t,e3,'LineWidth',lSize);
grid on
grid minor
h=legend('$e_3(t)$','Location','northeast');
set(h,'Interpreter','Latex','FontSize',fSize);
xlabel('$t[s]$','Interpreter','Latex','Fontsize',fSize);
ylim([-0.01 0.11]);
set(gcf,'units','normalized','outerposition',[0 0 1 1]);
saveas(gcf, nome_7, format1);
saveas(gcf, nome_7, format2);
close(fig)

fig=figure('visible','off', 'DefaultAxesPosition', [0.085, 0.135, 0.90,
    0.845]);
plot(t,e4,'LineWidth',lSize);
grid on
grid minor
```

```
h=legend(’$e_4(t)$’,’Location’,’northeast’);
set(h,’Interpreter’,’Latex’,’FontSize’,fSize);
xlabel(’$t[s]$’,’Interpreter’,’Latex’,’Fontsize’,fSize);
ylim([-0.011 0.01]);
set(gcf,’units’,’normalized’,’outerposition’,[0 0 1 1]);
saveas(gcf, nome_8, format1);
saveas(gcf, nome_8, format2);
close(fig)

fig=figure(’visible’,’off’, ’DefaultAxesPosition’, [0.055, 0.135, 0.93,
    0.845]);
plot(t,Xmaster(:,1),t, Xslave(:,1),’:’,’LineWidth’,lSize);
grid on
grid minor
h=legend(’$x_m(t)$’,’$x_s(t)$’,’Location’,’northeast’);
set(h,’Interpreter’,’Latex’,’FontSize’,fSize);
xlabel(’$t[s]$’,’Interpreter’,’Latex’,’Fontsize’,fSize);
ylim([-0.44 0.8]);
set(gcf,’units’,’normalized’,’outerposition’,[0 0 1 1]);
saveas(gcf, nome_9, format1);
saveas(gcf, nome_9, format2);
close(fig)

fig=figure(’visible’,’off’, ’DefaultAxesPosition’, [0.045, 0.135, 0.94,
    0.86]);
plot(t,Xmaster(:,3),t, Xslave(:,3),’:’,’LineWidth’,lSize);
grid on
grid minor
h=legend(’$z_m(t)$’,’$z_s(t)$’,’Location’,’northeast’);
set(h,’Interpreter’,’Latex’,’FontSize’,fSize);
xlabel(’$t[s]$’,’Interpreter’,’Latex’,’Fontsize’,fSize);
ylim([-0.05 1.1]);
set(gcf,’units’,’normalized’,’outerposition’,[0 0 1 1]);
saveas(gcf, nome_10, format1);
saveas(gcf, nome_10, format2);
close(fig)

fig=figure(’visible’,’off’, ’DefaultAxesPosition’, [0.045, 0.135, 0.94,
    0.86]);
plot(t,Xmaster(:,4),t, Xslave(:,4),’:’,’LineWidth’,lSize);
grid on
grid minor
h=legend(’$w_m(t)$’,’$w_s(t)$’,’Location’,’northeast’);
set(h,’Interpreter’,’Latex’,’FontSize’,fSize);
xlabel(’$t[s]$’,’Interpreter’,’Latex’,’Fontsize’,fSize);
ylim([-0.02 0.54]);
set(gcf,’units’,’normalized’,’outerposition’,[0 0 1 1]);
```

```
saveas(gcf, nome_11, format1);
saveas(gcf, nome_11, format2);
close(fig)

fig=figure('visible','off', 'DefaultAxesPosition', [0.068, 0.135, 0.917,
    0.86]);
plot(t,e1_m,'LineWidth',lSize);
grid on
grid minor
h=legend('$e_1(t)$','Location','northeast');
set(h,'Interpreter','Latex','FontSize',fSize);
xlabel('$t[s]$','Interpreter','Latex','Fontsize',fSize);
ylim([-0.22 0.06]);
set(gcf,'units','normalized','outerposition',[0 0 1 1]);
saveas(gcf, nome_12, format1);
saveas(gcf, nome_12, format2);
close(fig)

fig=figure('visible','off', 'DefaultAxesPosition', [0.07, 0.135, 0.915,
    0.86]);
plot(t,e3_m,'LineWidth',lSize);
grid on
grid minor
h=legend('$e_3(t)$','Location','northeast');
set(h,'Interpreter','Latex','FontSize',fSize);
xlabel('$t[s]$','Interpreter','Latex','Fontsize',fSize);
ylim([-0.035 0.105]);
set(gcf,'units','normalized','outerposition',[0 0 1 1]);
saveas(gcf, nome_13, format1);
saveas(gcf, nome_13, format2);
close(fig)

fig=figure('visible','off', 'DefaultAxesPosition', [0.085, 0.135, 0.9,
    0.86]);
plot(t,e4_m,'LineWidth',lSize);
grid on
grid minor
h=legend('$e_4(t)$','Location','northeast');
set(h,'Interpreter','Latex','FontSize',fSize);
xlabel('$t[s]$','Interpreter','Latex','Fontsize',fSize);
ylim([-0.013 0.012]);
set(gcf,'units','normalized','outerposition',[0 0 1 1]);
saveas(gcf, nome_14, format1);
saveas(gcf, nome_14, format2);
close(fig)

fig=figure('visible','off', 'DefaultAxesPosition', [0.092, 0.135, 0.893,
    0.86]);
aux = Xmaster(:,1) - Xslave(:,1);
plot(t,msg(:,1),’-’,’Color’,color1,’LineWidth’,largura_linha);
grid on
grid minor
hold on;
plot(t, aux,’:’,’Color’,color2,’LineWidth’, largura_linha);
plot(t, 0.05*Xmaster(:,1) - 0.007,’-’,’Color’,color3,’LineWidth’,
    largura_linha);
ylim([-0.035 0.215])
h=legend(’Original message’, ’Decoded message’, ’Encoded message’,’
    Location’,’northeast’);
set(h,’FontSize’,fonte);
xlabel(’$$t[s]$$’,’Interpreter’,’Latex’,’Fontsize’,fonte);
ylabel(’$$m_1(t), \hat{m}_1(t), 0.05{\cdot}s_1(t) - 0.007$$’,’Interpreter
    ’,’Latex’,’Fontsize’,fonte);
set(gcf,’units’,’normalized’,’outerposition’,[0 0 1 1]);
saveas(gcf, nome_15, format1);
saveas(gcf, nome_15, format2);
close(fig)

fig=figure(’visible’,’off’, ’DefaultAxesPosition’, [0.1, 0.135, 0.885,
    0.86]);
aux = Xmaster(:,3) - Xslave(:,3);
plot(t,msg(:,3),’-’,’Color’,color1,’LineWidth’,largura_linha);
grid on
grid minor
hold on;
plot(t, aux,’:’,’Color’,color2,’LineWidth’, largura_linha);
plot(t, 0.05*Xmaster(:,2) - 0.007,’-’,’Color’,color3,’LineWidth’,
    largura_linha);
ylim([-0.105 0.075])
h=legend(’Original message’, ’Decoded message’, ’Encoded message’,’
    Location’,’northeast’);
set(h,’FontSize’,fonte);
xlabel(’$$t[s]$$’,’Interpreter’,’Latex’,’Fontsize’,fonte);
ylabel(’$$m_3(t), \hat{m}_3(t), 0.05{\cdot}s_3(t) - 0.007$$’,’Interpreter
    ’,’Latex’,’Fontsize’,fonte);
set(gcf,’units’,’normalized’,’outerposition’,[0 0 1 1]);
saveas(gcf, nome_16, format1);
saveas(gcf, nome_16, format2);
close(fig)

fig=figure(’visible’,’off’, ’DefaultAxesPosition’, [0.115, 0.135, 0.87,
    0.86]);
aux = Xmaster(:,4) - Xslave(:,4);
plot(t,msg(:,4),’-’,’Color’,color1,’LineWidth’,largura_linha);
```

```
grid on
grid minor
hold on;
plot(t, aux,’:’,’Color’,color2,’LineWidth’, largura_linha);
plot(t, 0.05*Xmaster(:,4) - 0.02,’-’,’Color’,color3,’LineWidth’,
    largura_linha);
ylim([-0.02 0.013])
h=legend(’Original message’, ’Decoded message’, ’Encoded message’,’
    Location’,’northeast’);
set(h,’FontSize’,fonte);
xlabel(’$$t[s]$$’,’Interpreter’,’Latex’,’Fontsize’,fonte);
ylabel(’$$m_4(t), \hat{m}_4(t), 0.05{\cdot}s_4(t) - 0.02$$’,’Interpreter’
    ,’Latex’,’Fontsize’,fonte);
set(gcf,’units’,’normalized’,’outerposition’,[0 0 1 1]);
saveas(gcf, nome_17, format1);
saveas(gcf, nome_17, format2);
close(fig)

fig=figure(’visible’,’off’, ’DefaultAxesPosition’, [0.06, 0.135, 0.925,
    0.86]);
plot(t,merror1,’LineWidth’, largura_linha);
grid on
grid minor
h=legend(’$$\tilde{m}_1(t)$$’,’Interpreter’,’Latex’,’Location’,’northeast
    ’);
set(h,’FontSize’,fonte);
xlabel(’$$t[s]$$’,’Interpreter’,’Latex’,’Fontsize’,fonte);
ylim([-0.008 0.225]);
set(gcf,’units’,’normalized’,’outerposition’,[0 0 1 1]);
saveas(gcf, nome_18, format1);
saveas(gcf, nome_18, format2);
close(fig)

fig=figure(’visible’,’off’, ’DefaultAxesPosition’, [0.068, 0.135, 0.917,
    0.845]);
plot(t,merror3,’LineWidth’, largura_linha);
grid on
grid minor
h=legend(’$$\tilde{m}_3(t)$$’,’Interpreter’,’Latex’,’Location’,’northeast
    ’);
set(h,’FontSize’,fonte);
xlabel(’$$t[s]$$’,’Interpreter’,’Latex’,’Fontsize’,fonte);
ylim([-0.11 0.02]);
set(gcf,’units’,’normalized’,’outerposition’,[0 0 1 1]);
saveas(gcf, nome_19, format1);
saveas(gcf, nome_19, format2);
close(fig)
```

```
fig=figure('visible','off', 'DefaultAxesPosition', [0.085, 0.135, 0.9,
    0.86]);
plot(t,merror4,'LineWidth', largura_linha);
grid on
grid minor
h=legend('$$\tilde{m}_4(t)$$','Interpreter','Latex','Location','northeast
    ');
set(h,'FontSize',fonte);
xlabel('$$t[s]$$','Interpreter','Latex','Fontsize',fonte);
ylim([-0.01 0.011]);
set(gcf,'units','normalized','outerposition',[0 0 1 1]);
saveas(gcf, nome_20, format1);
saveas(gcf, nome_20, format2);
close(fig)
```

A.3.4 Simulink plant used for simulations corresponding to Figures 7.24 - 7.31 and Table 7.2

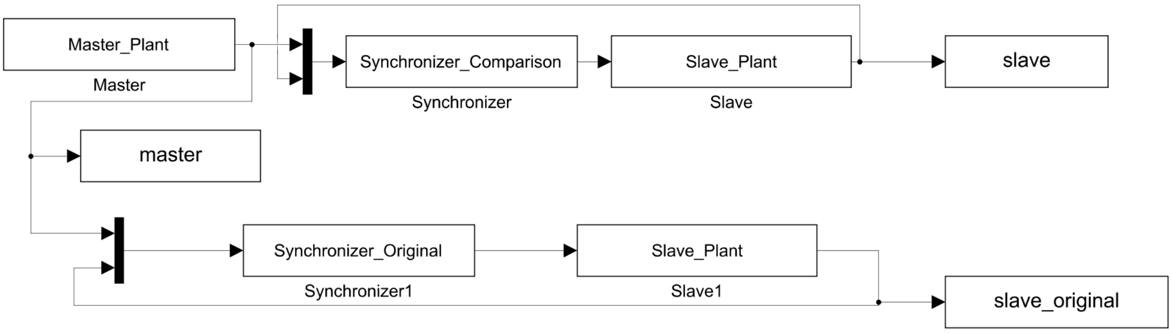

Figure A.4 – Simulink of comparison.

A.3.5 Codes used for simulations corresponding to Figures 7.12 - 7.23 and Table 7.1

Listing A.34 – Main.m

```
clc     %limpa o que estava escrito no terminal
addpath('./Circuit_Simulation/');

proj = openProject("D:/Doutorado/Tese de doutorado/
    Tese_Doutorado_180007220_2021/Chapter_6/Simulations5");
model = 'CircuitSimulation.slx';
open_system(model,'loadonly'); %load simulink model
modelname = 'CircuitSimulation';
set_param(modelname,'SolverType','Variable-step'...
                   ,'Solver','ode15s'...
                   ,'RelTol','1e-8'...
```

```
                    ,'AbsTol','1e-8'...
                    ,'MaxStep','0.01'...
                    ,'StopTime','0.03'...
                    ,'ScreenColor','white'...
                    ,'TimeSaveName','t');
save_system('CircuitSimulation.slx');
sim(model); %Descomente para simular
close_system(model, 0); %Close the simulink model, 1 to save, 0 to not
   save

merror2 = master(:,2) - slave(:,2) - m2;
merror3 = master(:,3) - slave(:,3) - m3;

e1 = slave(:,1) - master(:,1);
e2 = slave(:,2) - master(:,2);
e3 = slave(:,3) - master(:,3);
e4 = slave(:,4) - master(:,4);

e = [e1 e2 e3 e4];
serror1 = size(e1);
serror2 = size(e2);
serror3 = size(e3);
serror4 = size(e4);
serror = size(e);

fileID = fopen('./Tables/Table.txt','w');
fprintf(fileID,'Root Mean Square of State Errors\n\n');
fprintf(fileID,'e_1_{rms} \t %f \n', norm(e1)/sqrt(serror1(1)));
fprintf(fileID,'e_2_{rms} \t %f \n', norm(e2)/sqrt(serror2(1)));
fprintf(fileID,'e_3_{rms} \t %f \n', norm(e3)/sqrt(serror3(1)));
fprintf(fileID,'e_4_{rms} \t %f \n', norm(e4)/sqrt(serror4(1)));
fprintf(fileID,'e_{rms} \t %f \n\n', norm(e)/sqrt(serror(1)));
fclose(fileID);

save ./Saved_Data/Data.mat
Graphs

%close(proj);
clc
```

Listing A.35 – LoadWorkspace.m

```
clear;
load('Workspace.mat');
clc;
```

Listing A.36 – MessageGenerator2.m

```
function [sys,x0,str,ts] = MessageGenerator2(t,x,u,flag)

x_out = 0;

switch flag
   %%%%%%%%%%%%%%%%%
   % Inicialização %
   %%%%%%%%%%%%%%%%%
case 0
  sizes = simsizes;
    sizes.NumContStates = 0; %Número de estados constantes
    sizes.NumDiscStates = 0; %Número de estados discretos
    sizes.NumOutputs = 1;    %Número de saídas
    sizes.NumInputs = 0;     %Número de entradas
    sizes.DirFeedthrough = 1;
    sizes.NumSampleTimes = 1;
    sys = simsizes(sizes);
    x0=[]; %Condições iniciais
  str=[];
  ts=[0 0];
case 3
   if(t>=0.000) && (t<0.003)
       x_out = 0;
   end
   if(t>=0.003) && (t<0.006)
       x_out = 0.05;
   end
   if(t>=0.006) && (t<0.009)
       x_out = 0.0;
   end
   if(t>=0.009) && (t<0.012)
       x_out = 0.05;
   end
   if(t>=0.012) && (t<0.015)
       x_out = 0;
   end
   if(t>=0.015) && (t<0.018)
       x_out = 0;
   end
   if(t>=0.018) && (t<0.021)
       x_out = 0.05;
   end
   if(t>=0.021) && (t<0.024)
       x_out = 0.05;
   end
   if(t>=0.024) && (t<0.027)
       x_out = 0;
   end
   if(t>=0.027) && (t<0.030)
       x_out = 0.05;
   end
   sys = x_out;
   %%%%%%%%%
   %  End  %
   %%%%%%%%%
case {1,2,4,9}
    sys = []; % Não faz nada
 otherwise
   error(['unhandled flag = ',num2str(flag)]);
end
```

Listing A.37 – MessageGenerator3.m

```
function [sys,x0,str,ts] = MessageGenerator3(t,x,u,flag)

       a0 =   -0.001811;
       a1 =     0.09614;
       b1 =    -0.08111;
       a2 =   -0.002126;
       b2 =   -0.002561;
       a3 =     0.01418;
       b3 =     0.03685;
       a4 =   0.0004152;
       b4 =   -0.002264;
       a5 =    -0.01999;
       b5 =     -0.0121;
       a6 =   7.883e-05;
       b6 =  -0.0008134;
       a7 =    0.002927;
       b7 =  -0.0009047;
       a8 =  -3.586e-05;
       b8 =  -0.0008205;
       w =   0.15*6278;

switch flag
   %%%%%%%%%%%%%%%%%
   % Inicialização %
   %%%%%%%%%%%%%%%%%
case 0
  sizes = simsizes;
    sizes.NumContStates = 0; %Número de estados constantes
    sizes.NumDiscStates = 0; %Número de estados discretos
    sizes.NumOutputs = 1;    %Número de saídas
    sizes.NumInputs = 0;     %Número de entradas
    sizes.DirFeedthrough = 1;
    sizes.NumSampleTimes = 1;
    sys = simsizes(sizes);
    x0=[]; %Condições iniciais
  str=[];
  ts=[0 0];
case 3
   x_out = 0.5*(-0.2 + 1*(a0 + a1*cos(t*w) + b1*sin(t*w) + a2*cos(2*t*w)
   + b2*sin(2*t*w) + a3*cos(3*t*w) + b3*sin(3*t*w) + a4*cos(4*t*w) + b4*
   sin(4*t*w) + a5*cos(5*t*w) + b5*sin(5*t*w) + a6*cos(6*t*w) + b6*sin(6*
   t*w) + a7*cos(7*t*w) + b7*sin(7*t*w) + a8*cos(8*t*w) + b8*sin(8*t*w)))
   ;
   sys = x_out;
   %%%%%%%%%
   %  End  %
   %%%%%%%%%
case {1,2,4,9}
    sys = []; % Não faz nada
 otherwise
   error(['unhandled flag = ',num2str(flag)]);
end
```

Listing A.38 – Graphs.m

```
format1 = 'jpeg';
format2 = 'epsc';

fSize = 38;
axesSize = 38;
lSize = 2;
dvlsize = 2;
dhlsize = 2;
fonte = 38;
largura_linha = 2;
color1 = [0 0.4470 0.7410];
color2 = [0.8500 0.3250 0.0980];
color3 = [0.4660 0.6740 0.1880];

nome_1 = './Figures/FIG_6_11';
nome_2 = './Figures/FIG_6_12';
nome_3 = './Figures/FIG_6_13';
nome_4 = './Figures/FIG_6_14';
nome_5 = './Figures/FIG_6_15';
nome_6 = './Figures/FIG_6_16';
nome_7 = './Figures/FIG_6_17';
nome_8 = './Figures/FIG_6_18';
nome_9 = './Figures/FIG_6_19';
nome_10 = './Figures/FIG_6_20';
nome_11 = './Figures/FIG_6_21';
```

```
nome_12 = ’./Figures/FIG_6_22’;

set(0,’DefaultAxesFontSize’,axesSize);

fig=figure(’visible’,’off’, ’DefaultAxesPosition’, [0.053, 0.135, 0.92,
    0.86]);
plot(t, master(:,1),t, slave(:,1),’:’,’LineWidth’,lSize);
grid on
grid minor
h=legend(’$x_m(t)$’,’$x_s(t)$’,’Location’,’northeast’);
set(h,’Interpreter’,’Latex’,’FontSize’,fSize);
xlabel(’$t[s]$’,’Interpreter’,’Latex’,’Fontsize’,fSize);
ylim([-1.4 0.7]);
set(gcf,’units’,’normalized’,’outerposition’,[0 0 1 1]);
saveas(gcf, nome_1, format1);
saveas(gcf, nome_1, format2);
close(fig)

fig=figure(’visible’,’off’, ’DefaultAxesPosition’, [0.055, 0.135, 0.919,
    0.86]);
plot(t,master(:,2),t, slave(:,2),’:’,’LineWidth’,lSize);
grid on
grid minor
h=legend(’$y_m(t)$’,’$y_s(t)$’,’Location’,’northeast’);
set(h,’Interpreter’,’Latex’,’FontSize’,fSize);
xlabel(’$t[s]$’,’Interpreter’,’Latex’,’Fontsize’,fSize);
ylim([-2.4 1.1]);
set(gcf,’units’,’normalized’,’outerposition’,[0 0 1 1]);
saveas(gcf, nome_2, format1);
saveas(gcf, nome_2, format2);
close(fig)

fig=figure(’visible’,’off’, ’DefaultAxesPosition’, [0.025, 0.135, 0.948,
    0.86]);
plot(t,master(:,3),t, slave(:,3),’:’,’LineWidth’,lSize);
grid on
grid minor
h=legend(’$z_m(t)$’,’$z_s(t)$’,’Location’,’northeast’);
set(h,’Interpreter’,’Latex’,’FontSize’,fSize);
xlabel(’$t[s]$’,’Interpreter’,’Latex’,’Fontsize’,fSize);
ylim([-0.1 4.6]);
set(gcf,’units’,’normalized’,’outerposition’,[0 0 1 1]);
saveas(gcf, nome_3, format1);
saveas(gcf, nome_3, format2);
close(fig)

fig=figure(’visible’,’off’, ’DefaultAxesPosition’, [0.032, 0.135, 0.942,
    0.86]);
plot(t,master(:,4),t, slave(:,4),':','LineWidth',lSize);
grid on
grid minor
h=legend('$w_m(t)$','$w_s(t)$','Location','northeast');
set(h,'Interpreter','Latex','FontSize',fSize);
xlabel('$t[s]$','Interpreter','Latex','Fontsize',fSize);
ylim([-8.8 0.6]);
set(gcf,'units','normalized','outerposition',[0 0 1 1]);
saveas(gcf, nome_4, format1);
saveas(gcf, nome_4, format2);
close(fig)

fig=figure('visible','off', 'DefaultAxesPosition', [0.055, 0.135, 0.92,
    0.795]);
plot(t,e1,'LineWidth',lSize);
grid on
grid minor
h=legend('$e_1(t)$','Location','northeast');
set(h,'Interpreter','Latex','FontSize',fSize);
xlabel('$t[s]$','Interpreter','Latex','Fontsize',fSize);
ylim([-0.001 0.0014]);
set(gcf,'units','normalized','outerposition',[0 0 1 1]);
saveas(gcf, nome_5, format1);
saveas(gcf, nome_5, format2);
close(fig)

fig=figure('visible','off', 'DefaultAxesPosition', [0.035, 0.135, 0.94,
    0.795]);
plot(t,e2,'LineWidth',lSize);
grid on
grid minor
h=legend('$e_2(t)$','Location','northeast');
set(h,'Interpreter','Latex','FontSize',fSize);
xlabel('$t[s]$','Interpreter','Latex','Fontsize',fSize);
ylim([-0.009 0.015]);
set(gcf,'units','normalized','outerposition',[0 0 1 1]);
saveas(gcf, nome_6, format1);
saveas(gcf, nome_6, format2);
close(fig)

fig=figure('visible','off', 'DefaultAxesPosition', [0.035, 0.135, 0.94,
    0.795]);
plot(t,e3,'LineWidth',lSize);
grid on
grid minor
h=legend('$e_3(t)$','Location','northeast');
```

```
set(h,’Interpreter’,’Latex’,’FontSize’,fSize);
xlabel(’$t[s]$’,’Interpreter’,’Latex’,’Fontsize’,fSize);
ylim([-0.007 0.015]);
set(gcf,’units’,’normalized’,’outerposition’,[0 0 1 1]);
saveas(gcf, nome_7, format1);
saveas(gcf, nome_7, format2);
close(fig)

fig=figure(’visible’,’off’, ’DefaultAxesPosition’, [0.033, 0.135, 0.942,
    0.795]);
plot(t,e4,’LineWidth’,lSize);
grid on
grid minor
h=legend(’$e_4(t)$’,’Location’,’northeast’);
set(h,’Interpreter’,’Latex’,’FontSize’,fSize);
xlabel(’$t[s]$’,’Interpreter’,’Latex’,’Fontsize’,fSize);
ylim([-0.0055 0.0019]);
set(gcf,’units’,’normalized’,’outerposition’,[0 0 1 1]);
saveas(gcf, nome_8, format1);
saveas(gcf, nome_8, format2);
close(fig)

fig=figure(’visible’,’off’, ’DefaultAxesPosition’, [0.1, 0.135, 0.875,
    0.86]);
aux = master(:,2) + m2;
plot(t, m2,’-’,’Color’,color1,’LineWidth’,largura_linha);
grid on
grid minor
hold on;
plot(t, m2_hat,’:’,’Color’,color2,’LineWidth’, largura_linha);
plot(t, 0.05*aux + 0.06,’-’,’Color’,color3,’LineWidth’, largura_linha);
ylim([-0.06 0.12])
h=legend(’Original message’, ’Decoded message’, ’Encoded message’,’
    Location’,’northeast’);
set(h,’FontSize’,fonte);
xlabel(’$t[s]$’,’Interpreter’,’Latex’,’Fontsize’,fonte);
ylabel(’$m_2(t), \hat{m}_2(t), 0.05{\cdot}s_2(t) + 0.06$’,’Interpreter’,’
    Latex’,’Fontsize’,fonte);
set(gcf,’units’,’normalized’,’outerposition’,[0 0 1 1]);
saveas(gcf, nome_9, format1);
saveas(gcf, nome_9, format2);
close(fig)

fig=figure(’visible’,’off’, ’DefaultAxesPosition’, [0.1, 0.135, 0.875,
    0.86]);
aux = master(:,3) + m3;
plot(t, m3,’-’,’Color’,color1,’LineWidth’,largura_linha);
```

```
grid on
grid minor
hold on;
plot(t, m3_hat,':','Color',color2,'LineWidth', largura_linha);
plot(t, 0.05*aux - 0.2,'-','Color',color3,'LineWidth', largura_linha);
ylim([-0.21 0.025])
h=legend('Original message', 'Decoded message', 'Encoded message','
    Location','northeast');
set(h,'FontSize',fonte);
xlabel('$t[s]$','Interpreter','Latex','Fontsize',fonte);
ylabel('$m_3(t), \hat{m}_3(t), 0.05{\cdot}s_3(t) - 0.2$','Interpreter','
    Latex','Fontsize',fonte);
set(gcf,'units','normalized','outerposition',[0 0 1 1]);
saveas(gcf, nome_10, format1);
saveas(gcf, nome_10, format2);
close(fig)

fig=figure('visible','off', 'DefaultAxesPosition', [0.07, 0.135, 0.905,
    0.845]);
plot(t,merror2,'LineWidth', largura_linha);
grid on
grid minor
h=legend('$\tilde{m}_2(t)$','Interpreter','Latex','Location','northeast')
    ;
set(h,'FontSize',fonte);
xlabel('$t[s]$','Interpreter','Latex','Fontsize',fonte);
ylim([-0.055 0.01]);
set(gcf,'units','normalized','outerposition',[0 0 1 1]);
saveas(gcf, nome_11, format1);
saveas(gcf, nome_11, format2);
close(fig)

fig=figure('visible','off', 'DefaultAxesPosition', [0.06, 0.135, 0.915,
    0.845]);
plot(t,merror3,'LineWidth', largura_linha);
grid on
grid minor
h=legend('$\tilde{m}_3(t)$','Interpreter','Latex','Location','northeast')
    ;
set(h,'FontSize',fonte);
xlabel('$t[s]$','Interpreter','Latex','Fontsize',fonte);
ylim([0.03 0.18]);
set(gcf,'units','normalized','outerposition',[0 0 1 1]);
saveas(gcf, nome_12, format1);
saveas(gcf, nome_12, format2);
close(fig)
```

A.3.6 Codes used for simulations corresponding to Figures 7.24 - 7.31 and Table 7.2

Listing A.39 – Main.m

```
clc     %limpa o que estava escrito no terminal

model = 'Comparison.slx';
open_system(model,'loadonly'); %load simulink model
modelname = 'Comparison';
set_param(modelname,'SolverType','Variable-step'...
                   ,'Solver','ode15s'...
                   ,'RelTol','1e-8'...
                   ,'AbsTol','1e-8'...
                   ,'MaxStep','0.1'...
                   ,'StopTime','10'...
                   ,'ScreenColor','white'...
                   ,'TimeSaveName','t');
save_system('Comparison.slx');
sim(model); %Descomente para simular
close_system(model, 0); %Close the simulink model, 1 to save, 0 to not
    save

e1 = slave(:,1) - master(:,1);
e2 = slave(:,2) - master(:,2);
e3 = slave(:,3) - master(:,3);
e4 = slave(:,4) - master(:,4);

e1_c = slave_original(:,1) - master(:,1);
e2_c = slave_original(:,2) - master(:,2);
e3_c = slave_original(:,3) - master(:,3);
e4_c = slave_original(:,4) - master(:,4);

e = [e1 e2 e3 e4];
serror1 = size(e1);
serror2 = size(e2);
serror3 = size(e3);
serror4 = size(e4);
serror = size(e);

fileID = fopen('./Tables/Table.txt','w');
fprintf(fileID,'Root Mean Square of State Errors\n\n');
fprintf(fileID,'e_1_{rms} \t %f \n', norm(e1)/sqrt(serror1(1)));
fprintf(fileID,'e_2_{rms} \t %f \n', norm(e2)/sqrt(serror2(1)));
fprintf(fileID,'e_3_{rms} \t %f \n', norm(e3)/sqrt(serror3(1)));
fprintf(fileID,'e_4_{rms} \t %f \n', norm(e4)/sqrt(serror4(1)));
fprintf(fileID,'e_{rms} \t %f \n\n', norm(e)/sqrt(serror(1)));
```

```
fclose(fileID);

save ./Saved_Data/Data.mat
Graphs
clc
```

Listing A.40 – Master_Plant.m

```
function [sys,x0,str,ts] = Master_Plant(t,x,u,flag)

%Extract from [20]
%[20] C. Zhou, C. Yang, D. Xu , and C. Chen, "Dynamic Analysis and Finite
    -
%Time Synchronization of a New Hyperchaotic System With Coexisting
%Attractors," IEEE Access, vol. 7, pp. 52896-52902, Apr. 2019.

a = 20;
b = 32;
c = 3;
d = 1;
k = -1;

switch flag
case 0
  sizes = simsizes;
    sizes.NumContStates = 4;
    sizes.NumDiscStates = 0;
    sizes.NumOutputs = 4;
    sizes.NumInputs = 0;
    sizes.DirFeedthrough = 1;
    sizes.NumSampleTimes = 1;
    sys = simsizes(sizes);
  x0=[1 1 1 1]; %initial conditions
    str=[];
  ts=[0 0];
case 1     %System
      sys = [a*(x(2) - x(1)) - x(4);
             b*x(1) - x(1)*x(3) - x(2);
             x(1)*x(2) - c*x(3);
             d*x(1)*x(3) - k*x(4)];
case 3
   sys = x;
case {2,4,9}
    sys = [];
 otherwise
   error(['unhandled flag = ',num2str(flag)]);
end
```

Listing A.41 – Slave_Plant.m

```
function [sys,x0,str,ts] = Slave_Plant(t,x,u,flag)

a = 20;
b = 32;
c = 3;
d = 1;
k = -1;
beta = 1;

switch flag
case 0
  sizes = simsizes;
    sizes.NumContStates = 4;
    sizes.NumDiscStates = 0;
    sizes.NumOutputs = 4;
    sizes.NumInputs = 4;
    sizes.DirFeedthrough = 1;
    sizes.NumSampleTimes = 1;
    sys = simsizes(sizes);
    x0=[20 15 15 -200]; %initial conditions
  str=[];
  ts=[0 0];
case 1
    h = 0;
    if(t > 3)
       h = beta;
    end
      sys = [a*(x(2)-x(1)) - x(4) + u(1);   %System
             b*x(1) - x(1)*x(3) - x(2)  + u(2);
             x(1)*x(2) - c*x(3) + u(3);
             d*x(1)*x(3) - k*x(4) + u(4)]...
             + h*[0.3*exp(5*10^(-5)*x(1)^2); %disturbances
                  0.2*exp(10^(-5)*x(2)^2);
                  3*sin(3*t) + cos(20*t);
                  5*sin(10*t) + 10*cos(t)];
case 3
   sys = x;
case {2,4,9}
    sys = [];
 otherwise
   error(['unhandled flag = ',num2str(flag)]);
end
```

Listing A.42 – Synchronizer_Comparison.m

```
function [sys,x0,str,ts] = Synchronizer_Comparison(t,x,u,flag)
```

```
psi1 = 100;
psi2 = 10000;
psi3 = 100;

switch flag
case 0
  sizes = simsizes;
    sizes.NumContStates  = 4;
    sizes.NumDiscStates  = 0;
    sizes.NumOutputs     = 4;
    sizes.NumInputs      = 8;
    sizes.DirFeedthrough = 1;
    sizes.NumSampleTimes = 1;
    sys = simsizes(sizes);
    x0=zeros(4,1);
    str=[];
  ts=[0 0];
case 1
      sys = [0;
             0;
             0;
             0];
case 3
      sys = [- psi1*(u(5) - u(1)) - psi2*(u(5) - u(1))*(u(8) - u(4))^2;
             0;
             0;
             - psi3*(u(8) - u(4))];
case {2,4,9}
    sys = [];
otherwise
   error(['unhandled flag = ',num2str(flag)]);
end
```

Listing A.43 – Synchronizer_Original.m

```
function [sys,x0,str,ts] = Synchronizer_Original(t,x,u,flag)

switch flag
case 0
  sizes = simsizes;
    sizes.NumContStates  = 4;
    sizes.NumDiscStates  = 0;
    sizes.NumOutputs     = 4;
    sizes.NumInputs      = 8;
    sizes.DirFeedthrough = 1;
    sizes.NumSampleTimes = 1;
    sys = simsizes(sizes);
    x0=zeros(4,1);
    str=[];
  ts=[0 0];
case 1
      sys = [0;
             0;
             0;
             0];
case 3
      sys =  [56*(u(1) - u(5)) + (abs(u(1) - u(5))^0.5)*sign(u(1) - u(5))
   ;
              26*(u(2) - u(6)) + (abs(u(2) - u(6))^0.5)*sign(u(2) - u(6))
   ;
              43*(u(3) - u(7)) + (abs(u(3) - u(7))^0.5)*sign(u(3) - u(7))
   ;
              56*(u(4) - u(8)) + (abs(u(4) - u(8))^0.5)*sign(u(4) - u(8))
   ];
case {2,4,9}
    sys = [];
otherwise
   error(['unhandled flag = ',num2str(flag)]);
end
```

Listing A.44 – Graphs.m

```
format1 = 'jpeg';
format2 = 'epsc';

fSize = 38;
axesSize = 38;
lSize = 2;
dvlsize = 2;
dhlsize = 2;
fonte = 38;
largura_linha = 2;
color1 = [0 0.4470 0.7410];
color2 = [0.8500 0.3250 0.0980];
color3 = [0.4660 0.6740 0.1880];

nome_1 = './Figures/FIG_6_23';
nome_2 = './Figures/FIG_6_24';
nome_3 = './Figures/FIG_6_25';
nome_4 = './Figures/FIG_6_26';
nome_5 = './Figures/FIG_6_27';
nome_6 = './Figures/FIG_6_28';
nome_7 = './Figures/FIG_6_29';
nome_8 = './Figures/FIG_6_30';
```

```
set(0,'DefaultAxesFontSize',axesSize);

fig=figure('visible','off', 'DefaultAxesPosition', [0.045, 0.135, 0.94,
    0.845]);
plot(t, master(:,1),'-','Color',color1,'LineWidth',lSize);
grid on
grid minor
hold on;
plot(t, slave(:,1),'--','Color',color2,'LineWidth',lSize);
plot(t, slave_original(:,1),':','Color',color3,'LineWidth',lSize);
h=legend("$x_m(t)$","$x_s(t)$","$x_s(t)$ in [185]",'Interpreter','latex',
    'Location','northeast','Orientation','horizontal');
xlabel('$t[s]$','Interpreter','Latex','Fontsize',fSize);
ylim([-19 30]);
set(gcf,'units','normalized','outerposition',[0 0 1 1]);
saveas(gcf, nome_1, format1);
saveas(gcf, nome_1, format2);
close(fig)

fig=figure('visible','off', 'DefaultAxesPosition', [0.045, 0.135, 0.94,
    0.86]);
plot(t, master(:,2),'-','Color',color1,'LineWidth',lSize);
grid on
grid minor
hold on;
plot(t, slave(:,2),'--','Color',color2,'LineWidth',lSize);
plot(t, slave_original(:,2),':','Color',color3,'LineWidth',lSize);
h=legend("$y_m(t)$","$y_s(t)$","$y_s(t)$ in [185]",'Interpreter','latex',
    'Location','northeast','Orientation','horizontal');
xlabel('$t[s]$','Interpreter','Latex','Fontsize',fSize);
ylim([-25 32]);
set(gcf,'units','normalized','outerposition',[0 0 1 1]);
saveas(gcf, nome_2, format1);
saveas(gcf, nome_2, format2);
close(fig)

fig=figure('visible','off', 'DefaultAxesPosition', [0.037, 0.135, 0.948,
    0.86]);
plot(t, master(:,3),'-','Color',color1,'LineWidth',lSize);
grid on
grid minor
hold on;
plot(t, slave(:,3),'--','Color',color2,'LineWidth',lSize);
plot(t, slave_original(:,3),':','Color',color3,'LineWidth',lSize);
h=legend("$z_m(t)$","$z_s(t)$","$z_s(t)$ in [185]",'Interpreter','latex',
    'Location','northeast','Orientation','horizontal');
xlabel('$t[s]$','Interpreter','Latex','Fontsize',fSize);
```

```
ylim([0 56]);
set(gcf,’units’,’normalized’,’outerposition’,[0 0 1 1]);
saveas(gcf, nome_3, format1);
saveas(gcf, nome_3, format2);
close(fig)

fig=figure(’visible’,’off’, ’DefaultAxesPosition’, [0.06, 0.135, 0.925,
    0.86]);
plot(t, master(:,4),’-’,’Color’,color1,’LineWidth’,lSize);
grid on
grid minor
hold on;
plot(t, slave(:,4),’--’,’Color’,color2,’LineWidth’,lSize);
plot(t, slave_original(:,4),’:’,’Color’,color3,’LineWidth’,lSize);
h=legend("$w_m(t)$","$w_s(t)$","$w_s(t)$ in [185]",’Interpreter’,’latex’,
    ’Location’,’northeast’,’Orientation’,’horizontal’);
xlabel(’$t[s]$’,’Interpreter’,’Latex’,’Fontsize’,fSize);
ylim([-240 140]);
set(gcf,’units’,’normalized’,’outerposition’,[0 0 1 1]);
saveas(gcf, nome_4, format1);
saveas(gcf, nome_4, format2);
close(fig)

fig=figure(’visible’,’off’, ’DefaultAxesPosition’, [0.038, 0.135, 0.947,
    0.845]);
plot(t, e1,’-’,’Color’,color1,’LineWidth’,lSize);
grid on
grid minor
hold on;
plot(t, e1_c,’--’,’Color’,color2,’LineWidth’,lSize);
h=legend("$e_1(t)$","$e_1(t)$ in [185]",’Interpreter’,’latex’,’Location’,
    ’northeast’,’Orientation’,’horizontal’);
xlabel(’$t[s]$’,’Interpreter’,’Latex’,’Fontsize’,fSize);
ylim([-2 20]);
set(gcf,’units’,’normalized’,’outerposition’,[0 0 1 1]);
saveas(gcf, nome_5, format1);
saveas(gcf, nome_5, format2);
close(fig)

fig=figure(’visible’,’off’, ’DefaultAxesPosition’, [0.045, 0.135, 0.94,
    0.845]);
plot(t, e2,’-’,’Color’,color1,’LineWidth’,lSize);
grid on
grid minor
hold on;
plot(t, e2_c,’--’,’Color’,color2,’LineWidth’,lSize);
h=legend("$e_2(t)$","$e_2(t)$ in [185]",’Interpreter’,’latex’,’Location’,
    'northeast','Orientation','horizontal');
xlabel('$t[s]$','Interpreter','Latex','Fontsize',fSize);
ylim([-12 15]);
set(gcf,'units','normalized','outerposition',[0 0 1 1]);
saveas(gcf, nome_6, format1);
saveas(gcf, nome_6, format2);
close(fig)

fig=figure('visible','off', 'DefaultAxesPosition', [0.035, 0.135, 0.95,
    0.845]);
plot(t, e3,'-','Color',color1,'LineWidth',lSize);
grid on
grid minor
hold on;
plot(t, e3_c,'--','Color',color2,'LineWidth',lSize);
h=legend("$e_3(t)$","$e_3(t)$ in [185]",'Interpreter','latex','Location',
    'northeast','Orientation','horizontal');
xlabel('$t[s]$','Interpreter','Latex','Fontsize',fSize);
ylim([-9 15]);
set(gcf,'units','normalized','outerposition',[0 0 1 1]);
saveas(gcf, nome_7, format1);
saveas(gcf, nome_7, format2);
close(fig)

fig=figure('visible','off', 'DefaultAxesPosition', [0.06, 0.135, 0.925,
    0.86]);
plot(t, e4,'-','Color',color1,'LineWidth',lSize);
grid on
grid minor
hold on;
plot(t, e4_c,'--','Color',color2,'LineWidth',lSize);
h=legend("$e_4(t)$","$e_4(t)$ in [185]",'Interpreter','latex','Location',
    'northeast','Orientation','horizontal');
xlabel('$t[s]$','Interpreter','Latex','Fontsize',fSize);
ylim([-210 30]);
set(gcf,'units','normalized','outerposition',[0 0 1 1]);
saveas(gcf, nome_8, format1);
saveas(gcf, nome_8, format2);
close(fig)
```