Centro de Investigación Científica y de Educación Superior de Ensenada, Baja California

Doctorado en Ciencias
en Electrónica y Telecomunicaciones
con orientación en Instrumentación y Control

Implementación de osciladores caóticos en sistemas embebidos y aplicaciones

Tesis
para cubrir parcialmente los requisitos necesarios para obtener el grado de
Doctor en Ciencias

Presenta:
Rodrigo Daniel Méndez Ramírez

Ensenada, Baja California, México
2018

Tesis defendida por

Rodrigo Daniel Méndez Ramírez

y aprobada por el siguiente Comité

Dr. César Cruz Hernández
Director de tesis

Dr. Rafael de Jesús Kelly Martínez

Dr. Vassili Spirine

Dr. Esteban Tlelo Cuautle

Dr. Hazael Serrano Guerrero

Dr. Adrian Arellano Delgado

Dr. Daniel Sauceda Carvajal
Coordinador del Programa de Posgrado en
Electrónica y Telecomunicaciones

Dra. Rufina Hernández Martínez
Directora de Estudios de Posgrado

Resumen de la tesis que presenta **Rodrigo Daniel Méndez Ramírez** como requisito parcial para la obtención del grado de Doctor en Ciencias en Electrónica y Telecomunicaciones con orientación en Instrumentación y Control.

Implementación de osciladores caóticos en sistemas embebidos y aplicaciones

Resumen aprobado por:

_____

Dr. César Cruz Hernández
Director de tesis

En este trabajo de tesis se aborda la Implementación de osciladores caóticos en sistemas embebidos y aplicaciones. Se presentan 4 estudios utilizando como base sistemas caóticos de naturaleza discreta y sistemas caóticos en tres dimensiones en sus versiones continuas y discretas. El primer estudio, reporta un nuevo sistema caótico tridimensional y su implementación electrónica. El segundo estudio, contiene un nuevo sistema criptográfico digital basado en caos utilizando el protocolo de comunicación SPI implementado en microcontroladores dsPIC. El tercer estudio, se lleva a cabo en una primera etapa utilizando los algoritmos numéricos de Euler, Heun y Runge Kutta de 4º orden, para obtener la versión discreta de un conjunto de cinco sistemas caóticos tridimensionales, en el cual, se incluye el nuevo sistema caótico, posteriormente en una segunda etapa se calculan los exponentes de Lyapunov del conjunto de cinco sistemas caóticos, con el propósito de obtener el intervalo donde se conservan las dinámicas caóticas. Posteriormente, se realiza un estudio para reproducir los algoritmos numéricos en versión discreta de los cinco sistemas caóticos propuestos, en un sistema embebido representado en 4 versiones utilizando microcontroladores PIC de 8 bits, dsPIC de 16 bits, PIC32 de 32 bits y un FPGA Altera, respectivamente. Finalmente, se incluyen resultados analíticos, numéricos y experimentales en cada uno de estos estudios reportados.

Palabras clave: **Nuevo sistema caótico, Sistema criptográfico digital, Microcontroladores PIC, dsPIC y PIC32, Sistema embebido, Protocolo SPI, FPGA, Degradación del caos.**

Abstract of the thesis presented by **Rodrigo Daniel Méndez Ramírez** as a partial requirement to obtain the Doctor in Science degree in Sciences in Electronics and Telecomunications with orientation in Instrumentation and Control.

Implementation of chaotic oscillators in embedded systems and its applications

Abstract approved by:

_____

Dr. César Cruz Hernández
Thesis Director

In this Ph.D. thesis, the implementation of chaotic oscillators in embedded systems and their applications are studied. Four studies are presented using chaotic systems of a discrete nature and chaotic systems in three dimensions in their continuous, and discretized versions. The first study presents a new three-dimensional chaotic system and its electronic implementation. The second study presents a new digital cryptosystem based on chaos using the SPI protocol implemented in dsPIC microcontrollers. The third study is conducted in the first stage using the numerical algorithms of Euler, Heun, and Runge Kutta of 4th order to obtain the discrete version of a set of five three-dimensional chaotic systems where the new chaotic system is included, later in a second stage, the Lyapunov exponents of the set of five chaotic systems are calculated to obtain the interval where the chaotic dynamics are conserved. Finally, a study is carried out to reproduce the numerical algorithms in discrete version of the five chaotic systems proposed in an embedded system represented in 4 versions using 8-bit PIC microcontrollers, 16-bit dsPIC, 32-bit PIC32, and an Altera FPGA; respectively. Finally, analytical, numerical and experimental results are presented in each of these studies.

Keywords: **New Chaotic system, Digital cryptosystem, PIC, dsPIC and PIC32 microcontrollers, Embedded System, SPI protocol, FPGA, Chaos degradation.**

"Y todo lo que hacéis, sea de palabra o de hecho, hacedlo todo en el nombre del Señor Jesús, dando gracias a Dios Padre por medio de él." – Colosenses 3:17.

A mis seres amados que ahora están en el reino de Dios, mis abuelos Enoc y Antonia, mi tía Aurora y mi hermana Karina quien partió mientras realicé mi estancia en el doctorado.

A mis hijos Santiago Maximiliano y Elías Benjamín. A mis sobrinos Fernanda, Alonso, Diego y Sofía. A mis padres Laura y Daniel, y mi hermana Priscila y Fabián. En general, a toda mi familia Ramírez en Chile y Román en México y Estados Unidos.

Finalmente, a mi amada esposa Carolina quien me ha acompañado incondicionalmente en todo este proceso.

Agradecimiento

A Dios por haberme permitido vivir grandes y difíciles momentos en este proceso. Sin su bendición, voluntad y misericordia, nunca podría haber terminado esta tesis.

Al núcleo de profesores e investigadores del Centro de Investigación Científica y de Educación Superior de Ensenada (CICESE) que fueron parte importante de mi formación académica.

A mi director de tesis, el Dr. César Cruz Hernández, quien me guio con mucha paciencia a lo largo de este trabajo. Sus consejos y enseñanzas fueron invaluables y me inculcaron mi interés por la investigación.

A los miembros de mi comité de tesis, Dr. Rafael de Jesús Kelly Martínez, Dr. Vassili Spirine, Dr. Hazael Serrano Guerrero, Dr. Esteban Tlelo Cuatle, Dr. Adrián Arellano Delgado por el tiempo, sugerencias y aportaciones que me permitieron realizar un trabajo de calidad.

Al Consejo Nacional de Ciencia y Tecnología (CONACyT) a través del proyecto de investigación en ciencia básica entre instituciones, Ref. 166654 ”Sincronización de sistemas complejos y algunas aplicaciones”.

Al Consejo Nacional de Ciencia y Tecnología (CONACyT) por brindarme el apoyo económico necesario para realizar mis estudios de doctorado.

Al Dr. Claudio Urrea, Dr. John Kern, al grupo de sistemas complejos en especial a la Dra. Rosa Martha López, Dr. Fausto Abundiz y Dr. Miguel Murillo, quienes colaboraron en mi proceso como estudiante y candidato a doctor en ciencias.

A mi familia, en especial a Don Antonio y Sra. Tina, Sra. Silvia y familia, Tía Carmelita y familia, Juan y Karen, Tío Hernán y Olfa, Tío Juan y Tina, Tío Esteban y familia, Tío Nino y familia, mis primos hermanos Susana, Brenda, Paula, Pamela, Claudia, Carola, Francisco, Leonardo, Andrés, Angiolo y Juan Pablo quienes me han apoyado continuamente en este proceso.

Al constante apoyo de mis queridos amigos Lenny y Margot, Alejandro y Paola, Luis y Paola, Yolanda y Mario, Don Héctor y Sra. Teresa, Víctor Hugo y Marcela, Jaime, mi gran Amiguito Luis, Marcela, Karina, Abril, José, Jorge y familia, Gerson, Fernanda, Juan Claudio y familia, Herman, Gaby y familia, Juan Carlos y familia, David y familia, Caio, Oscar, Stephanie y familia y en general a toda la gente que me aprecia.

Al Pastor Manuel Ulloa, Pastor David Horta y a Iglesia Metodista Independiente, quienes nos han apoyado en todos los momentos difíciles junto a la gracia de nuestro Señor Jesucristo.

Finalmente, a mis queridos amigos y compañeros del CICESE: Daniel, Marcelo, Diana, Roilhi, Leonardo, Abimael, Reinaldo, Eduardo, Oscar, Eliana, Héctor, Topacio, Efrén, Miriam, Eliana, Verónica, Juan José, Alberto, Wilbert, Rigoberto, Armando, Don René, Gustavo, Adán, Heberto, Zail, Manuel, Rolando, Rossy, Iván, Leobardo y Jonathan.

Tabla de contenido

Lista de figuras

Lista de tablas

Capítulo 1. Introducción

En este capítulo se proporciona una introducción, los objetivos generales y particulares de este trabajo de tesis, así mismo los alcances y limitaciones de la misma. Se pretende dar al lector un panorama general de este trabajo.

Los sistemas caóticos han llamado la atención de la comunidad científica, debido a sus potenciales aplicaciones en varias ramas de la ciencia e ingeniería. Se han reportado un gran número de trabajos en diferentes áreas con el fin de abordar algunos problemas interesantes, como la determinación de daños estructurales en materiales (Cates *et al.,* 2002) o en sistemas de comunicaciones seguras, ver por ejemplo (Cuomo *et al*., 1993; Juang *et al*., 2000; Gámez-Guzmán *et al.,* 2009; Uchida *et al*., 2012; Arellano-Delgado *et al.,* 2013).

De acuerdo con la *Real Academia Española*, en física y matemáticas se define el *caos* como: "*comportamiento aparentemente errático e impredecible de algunos sistemas dinámicos deterministas con gran sensibilidad a las condiciones iniciales", también se define como "estado amorfo e indefinido que se supone anterior a la ordenación del cosmos" y también como "confusión, desorden*".

En 1963, Lorenz propuso un sistema tridimensional de dos enrollamientos; este atractor se reconoce como el primer sistema caótico reportado (Lorenz, 1963). Desde entonces, muchos sistemas caóticos fueron reportados en la literatura (Rössler, 1976; Chua *et al.,* 1986; Sprott, 1994; Lü y Chen, 2002; Liu *et al.,* 2005; Wang y Luo, 2006), entre otros. Actualmente, existen sistemas caóticos que poseen características interesantes por ejemplo, menor número de puntos de equilibrios (Huang *et al*., 2015; Pham *et al.,* 2016; Wang *et al.*, 2016; Azar *et al.*, 2017), múltiples enrollamientos (Anishchenko *et al*., 1994; Brucoli *et al*. 1996 y Gámez-Guzmán *et al*., 2009), de orden fraccionario ver por ejemplo (Petras, 2011), dimensión infinita (Farmer, 1982) o por ejemplo, la versión discretizada del sistema MACM (Méndez-Ramírez *et al.*, 2017) posee un tamaño de paso alto, que permite conservar mayormente su dinámica caótica en comparación con una familia de sistemas caóticos clásicos como Lorenz, Rössler, Chen y Liu y Chen, etc.

1.1. Antecedentes

Para comprobar la existencia de caos en sistemas continuos o discretos, se realizan pruebas analíticas, numéricas y experimentales. La mayoría de los trabajos se enfocan en sistemas caóticos en tiempo continuo, sin embargo, en ocasiones es deseable que estos sistemas estén representados en tiempo discreto y puedan utilizarse en tiempo real ver por ejemplo (Mohamed *et al.,* 2013). En la implementación electrónica de sistemas caóticos continuos, se utilizan transistores o amplificadores operacionales (Buscarino *et al.,* 2014). Una propiedad de un sistema caótico es la sensibilidad a las condiciones iniciales, y esta propiedad es inherente en un sistema analógico, el cual tiene offsets presentes (voltajes o corrientes no son cero al alimentarlo) que se aprovechan para que un sistema autónomo pueda evolucionar, mientras que en un sistema digital es difícil cambiar las condiciones iniciales a rangos o resoluciones que admiten los sistemas analógicos. Además, no siempre se pueden obtener la medición de todas las variables de estado, dado que en algunos sistemas se trata de una intensidad de corriente baja que dificulta su medición (Cuomo *et al*., 1993; Sprott, 1997; 2010; Chua *et al.,* 1986; Cruz-Hernández*,* 2004; Kennedy *et al.,* 1995; Buscarino *et al.,* 2014). Para realizar pruebas en simulación numérica, se utilizan herramientas de software como Matlab o Labview (Yang *et al.,* 2005). En la implementación y representación de sistemas caóticos continuos en versión discretizada, se utilizan métodos de discretización, cuya finalidad es obtener la menor degradación posible en las dinámicas de los sistemas originales, en este caso, de caos posible con respecto al sistema continuo. Hoy en día, se utilizan herramientas digitales por ejemplo FPGAs, DSPs, entre otras para implementar los comportamientos de estos sistemas en un sistema embebido ver por ejemplo (Di Jasio, 2008; Koyuncu *et. al,* 2014; Tlelo-Cuatle *et al*., 2015; 2016; Azzaz *et al*., 2013; Muthuswamy y Banerjee, 2014). Desafortunadamente, los costos de software y de plataformas de hardware para el tratamiento digital de sistemas caóticos en versión discreta usando FPGAs y DSPs, siguen siendo muy elevados (BDTI Industry reports, 2006).

1.2. Justificación

La motivación principal y que incito a la realización de este trabajo de tesis doctoral, es la creciente investigación en el campo de sistemas caóticos en sistemas embebidos y su continuo desarrollo y aplicación en los diferentes campos de estudio. Actualmente los sistemas caóticos implementados electrónicamente en sus versiones discretas, han tenido gran éxito debido a la gran cantidad de aplicaciones, en las cuales, se utilizan los sistemas embebidos conformados principalmente por DSP,

microcontroladores en distintas gammas y procesadores más robustos como FPGA. Una de las aplicaciones con mayor auge es la de encriptado de información basado en caos, conociendo las características y ventajas de los sistemas caóticos, podemos inferir que utilizándolos e implementándolos en sistemas embebidos, tal encriptado será más eficiente dependiendo de la información que se desee ocultar.

La implementación electrónica de los sistemas caóticos en su versión continua, utilizando amplificadores operacionales está limitada principalmente por hardware, para sus versiones normalizadas generalmente los sistemas caóticos en versión continua, están sujetos a trabajar con amplitudes previamente atenuadas y las no linealidades se implementan generalmente con multiplicadores, con lo cual las operaciones algebraicas están previamente establecidas por los fabricantes, y en su conjunto la implementación electrónica está limitada a sus costos de implementación. Otra limitación de los sistemas caóticos implementados en su versión continua utilizando electrónica analógica, es que los componentes electrónicos externos como resistencias, bobinas y condensadores están sujetos a desgaste, por lo general su energización necesita tensiones con fuentes externas duales (positivas y negativas) y para aplicaciones más complejas como sincronización, encriptación e implementación de sistemas de redes complejas, se requieren grandes construcciones de los circuitos electrónicos en protoboards.

En este trabajo de tesis doctoral se reportan: un nuevo atractor caótico, un nuevo sistema criptográfico digital y un estudio de degradación de una familia de sistemas caóticos y todas las consideraciones necesarias para implementarlos electrónicamente utilizando sistemas embebidos. A diferencia de otros trabajos previos, en este estudio se proponen pruebas analíticas, numéricas y experimentales para determinar todas las condiciones necesarias para implementar el mapa de Hénon, como parte de un sistema criptográfico digital aplicado a un protocolo de comunicación para señales de audio. Se propone un estudio de degradación caótica para la versión discretizada de los sistemas caóticos continuos de 3 dimensiones (3D) de Lorenz, Rössler, Chen, Liu y Chen. Se incluye además, el nuevo atractor caótico propuesto por Méndez-Ramírez *et al.* (2017); esta familia de sistemas caóticos son implementados en sus versiones discretas, la diferencia con otros trabajos, es que se obtienen los límites de las dinámica caóticas del conjunto de los 5 sistemas caóticos mencionados utilizando los algoritmos numéricos de Euler, Heun y Rungue Kutta de 4to orden (RK4). Posteriormente, el resultado se valida e implementa en un sistema embebido presentado en 4 versiones, el cual, contempla el desempeño utilizando microcontroladores de gama baja, media, alta (microcontroladores PIC de 8 bits, dsPIC de 16 bits y PIC32 de 32 bits) y un FPGA (Altera Cyclone IV GX).

Una de las ventajas de este estudio, por ejemplo es determinar experimentalmente el tiempo de complejidad de un criptosistema digital utilizando el mapa de Hénon, y del conjunto de los 5 sistemas

caóticos Lorenz, Rössler, Chen, Liu y Chen y MACM en sus versiones discretizadas implementados en sistemas embebidos, donde la parte central son microcontroladores de 8, 16, y 32 bits y un FPGA. Esto le permite al lector determinar y utilizar cualquiera de estos sistemas caóticos en sus versiones discretizadas, para implementaciones electrónicas por ejemplo para aplicaciones de encriptación de información como texto, audio, imagen, etc.

1.3. Hipótesis

Es posible implementar osciladores caóticos en sistemas embebidos de gama baja, media y alta, en las cuales, sus dinámicas caóticas se conserven lo más posible y determinar todas las condiciones para su implementación digital, para realizar algunas aplicaciones de interés.

1.4. Objetivos

Con la realización de la presente tesis de doctorado, se pretendió alcanzar el siguiente objetivo general:

Determinar las principales condiciones que permiten implementar osciladores caóticos en sistemas embebidos (donde la parte central son microcontroladores de gama baja, media y alta de 8, 16 y 32 bits como los PIC, dsPIC, PIC32, respectivamente y en FPGA) y realizar algunas aplicaciones.

Objetivos particulares:

- Comparar los métodos que se utilizan en la literatura, para caracterizar e implementar los sistemas caóticos en plataformas embebidas.
- Verificar la representación de los sistemas caóticos donde se conserve el caos en sus dinámicas discretizadas con pruebas numéricas y experimentales.
- Diseñar un sistema caótico y su implementación electrónica.
- Realizar un estudio de los métodos que permiten simular e implementar los sistemas caóticos en su versión discretizada.
- Diseñar un sistema embebido, que permita comprobar la representación de los sistemas caóticos discretizados.

Entre las contribuciones de este trabajo de tesis se encuentran los siguientes artículos:

Revistas indizadas:

1. **Méndez-Ramírez, R.**, Cruz-Hernández, C., Arellano-Delgado, A. y Martínez-Clark, R. (2017). A new simple chaotic Lorenz-type system and its digital realization using a TFT touch-screen display embedded system. *Complexity*, **2017**(6820492): 1–13. Factor de impacto: 4.671.
2. **Méndez-Ramírez, R.**, Arellano-Delgado, A., Cruz-Hernández, C., Abúndiz-Pérez, F. y Martínez-Clark, R. (2017). Chaotic Digital Cryptosystem by using SPI Protocol and its dsPICs Implementation. Frontiers of Information Technology & Electronic Engineering, ZUSC-D-16-01346. Artículo en impression, http://dx.doi.org/10.1631/FITEE.1601346
3. **Méndez-Ramírez, R.**, Arellano-Delgado, A., Murillo-Escobar, M. y Cruz-Hernández, C. (2018). Degradation study of five 3-dimensional Chaotic Systems and its Digital Implementation in Embedded Systems. Artículo por someter.

Capítulo de libro:

4. **Méndez-Ramírez, R.**, Arellano-Delgado, A., Murillo-Escobar, M. y Cruz-Hernández, C. (2018). *Multimedia contents encryption using the chaotic MACM system on a smart-display*. Cryptographic and Information Security Approaches for Images and Videos. CRC Press Taylor & Francis Group. Capítulo aceptado para publicación.

Congresos:

5. **Méndez-Ramírez, R.**, Cruz-Hernández, C., Arellano-Delgado, A., Cardoza-Avendaño, L., López-Gutiérrez, R. M. y Aranda-Bricaire, E. (2015). Implementación del circuito hipercaótico de Chua en un sistema embebido de bajo costo. *Congreso Nacional de Control Automático, AMCA 2015,* Cuernavaca, Morelos, del 14 al 16 de Octubre, México, :171–176.
6. **Méndez-Ramírez, R.**, Cruz-Hernández, C., Arellano-Delgado, A. y López-Gutiérrez, R. M. (2016). Degradation Analysis of Generalized Chua's Circuit Generator of Multi-Scroll Chaotic Attractors and its Implementation on PIC32. *IEEE Future Technologies Conference 2016 (FTC 2016)*, San Fco, CA; USA. Diciembre 6-7, 2016.

1.5. Organización del manuscrito de tesis

Este trabajo de tesis se organiza de la siguiente manera. En el capítulo 2 se describe una breve reseña de los sistemas caóticos y su naturaleza. En el capítulo 3 se presenta un nuevo sistema caótico, sus propiedades y su implementación análoga. En el capítulo 4 se presenta el diseño e implementación de un nuevo criptosistema digital utilizando el protocolo SPI e implementado en dsPIC. En el capítulo 5 se presentan los algoritmos numéricos de Euler, Heun y RK4 que se utilizan para la representación en versión discreta de los sistemas caóticos en 3-D propuestos en esta tesis. En el capítulo 6 tópicos de sistemas embebidos con microcontroladores de gama baja, media y alta y FPGA. En el capítulo 7 se muestran los experimentos y los resultados que se obtuvieron de manera experimental en el estudio de degradación de una familia de 5 atractores caóticos en 3-D en comparación al nuevo atractor caótico propuesto donde se presenta en un sistema embebido implementado en 4 versiones. Finalmente, en el capítulo 8 se mencionan las principales conclusiones y el trabajo futuro que se desprende de la tesis.

Capítulo 2. Conceptos básicos de sistemas caóticos

En este capítulo, se describen los principales conceptos básicos de caos, los sistemas caóticos de naturaleza discreta y continua que se utilizarán en este trabajo de tesis y su posterior implementación digital en sistemas embebidos, que se mostrarán en capítulos posteriores. Primero se describen algunos de los sistemas caóticos de naturaleza discreta y posteriormente se describen los sistemas caóticos continuos en particular de 3 dimensiones.

Existe una gran variedad en la literatura de sistemas caóticos, en particular, si el lector está interesado en profundizar en el tópico, se recomienda por ejemplo consultar las siguientes referencias (Parker y Chua, 1989), (Strogatz, 1994), (Alligood *et al.*, 1996), (Bertuglia y Vaio, 2005), (Zhang *et al.*, 2009) y (Sprott, 2010).

2.1. Reseña histórica del caos

Desde que el hombre adquirió conciencia, existió en su interior profunda curiosidad por el conocimiento de la naturaleza. Ha tratado de obtener una respuesta determinada y única ante cualquier proceso natural, es decir, una respuesta determinista. El diccionario de la *Real Academia Española* (RAE) define el *determinismo* como la doctrina según la cual todos los acontecimientos, y en particular las acciones humanas, están unidos y determinados por la cadena de acontecimientos anteriores. Otra definición de determinismo la propone Bertrand Russell, en la cual, un sistema es determinista exactamente si sus estados previos determinan sus estados posteriores, en el sentido exacto en que el argumento de una función determina sus valores (Russell, 1953; Guerrero, 2000). Esto implica entender la realidad como la consecuencia directa de una causa. Se puede aplicar la idea de determinismo en distintos ámbitos. Por ejemplo, en biología, la idea de determinismo hace referencia a la explicación de la conducta de los organismos vivos según las características de sus genes. Esto quiere decir, que los seres humanos y los animales actúan de acuerdo a su adaptación evolutiva y a lo que dicta la genética en sus comportamientos. Pero como casi siempre ha pasado en la historia de la ciencia, la investigación de algunos sistemas naturales no presentan comportamientos que el hombre es capaz de determinar. Actualmente, esto puede estar pasando por ejemplo con el famoso experimento de los neutrinos, en el cual, el experimento aún se encuentra en etapa de experimentación y recopilación de información (Fermilab, 2017).

Actualmente el caos ya no es extraño para muchos estudiosos de los sistemas dinámicos. En cambio, la teoría y la tecnología del caos se han convertido gradualmente en un campo de investigación prometedor, con un impacto significativo en un número cada vez mayor de aplicaciones novedosas y potencialmente atractivas en la ingeniería y otras ciencias. Hay suficiente evidencia científica y razones prácticas para estudiar y utilizar el control de caos. En un sistema donde las respuestas irregulares son indeseables o incluso dañinas, el caos debe reducirse tanto como sea posible o incluso suprimirse completamente (ejemplo), mientras que si la dinámica caótica es beneficiosa y útil, la sincronización y generación del caos se vuelven deseables, por ejemplo cuando se desea sincronizar dos sistemas caóticos donde uno de los sistemas adopte las dinámica caótica al otro sistema (Pecora y Carroll, 1990; Sira-Ramírez y Cruz-Hernández, 2001; López-Mancilla y Cruz-Hernández, 2004; Cruz-Hernández, 2004; Stefanski y Kapitaniak, 2003; Posadas-Castillo *et al*., 2007; Cruz-Hernández *et al*., 2010; Serrano-Guerrero *et al*., 2010), otro ejemplo donde la dinámica caótica es deseable es en la encriptación de información (Murillo-Escobar *et al*., 2015; 2017). Se han propuesto muchos métodos para controlar el caos, es decir, para estabilizar los sistemas dinámicos, cuando la dinámica caótica no es deseable (Moon, 1992; Chen y Dong, 1998; El Nashie, 1996; Kapitaniak, 1992). El caos es importante en las comunicaciones seguras, procesamiento de la información, mezcla de líquidos, en sistemas biológicos, etc. (Stefanski *et al*., 2003; Kapitaniak, 1995; Lakshmanan y Murali, 1996). Para este propósito, transformar un sistema dinámico no caótico o impedir (o mejorar) la dinámica caótica en un sistema caótico se llama *anticontrol de caos* (Chen, 1997; Chen y Lai, 1998), en la literatura se reporta, por ejemplo, un método simple para contrarrestar el caos en el movimiento de un cuerpo rígido (Chen y Lee, 2004).

Jules Henri Poincaré (1854-1912) matemático, físico teórico y filósofo de la ciencia es reconocido como el primero en vislumbrar la posibilidad del caos. Cuando estudió las propiedades de estabilidad del sistema solar a finales del siglo XIX, Poincaré descubrió que incluso en el caso de tres cuerpos que se movían bajo la ley de atracción de Newton, podían exhibir un comportamiento dinámico muy complicado (Robinson, 2004). Este tipo de movimiento depende sensiblemente de las condiciones iniciales, lo que imposibilita la predicción de las trayectorias a largo plazo si no se conocen exactamente las condiciones iniciales (Diacu y Holmes, 1996). Birkhoff desarrolló métodos geométricos creados por Poincaré y encontró muchos tipos de comportamientos limitantes a largo plazo. Durante la primera mitad del siglo XX, los osciladores no lineales se estudiaron principalmente debido a su papel vital en el desarrollo de tecnologías como la radio, el radar, los bucles de fase bloqueada y los láseres. Especialmente, en la década de 1950, Cartwright, Littlewood y Levinson revelaron un tipo de complejidad desconocida para mostrar que cierto oscilador no lineal forzado no lineal presentaba un número infinito de períodos diferentes.

Posteriormente en 1963, el meteorólogo estadounidense Edward Lorenz realizó una importante contribución, utilizó una computadora para estudiar las dinámicas de un grupo de ecuaciones diferenciales no lineales ordinarias. Estas ecuaciones se obtuvieron a partir de las ecuaciones diferenciales parciales que describían el movimiento turbulento de la atmósfera. Lorenz descubrió que un pequeño cambio en las condiciones iniciales conducía a resultados muy diferentes en un tiempo relativamente corto; esta propiedad se conoce como *dependencia sensible a las condiciones iniciales* (Lorenz, 1963). Este fue realmente un gran descubrimiento, ya que los científicos tradicionales en ese momento, creían que dos trayectorias emitidas desde puntos iniciales cercanos siempre estarían próximas cuando el tiempo transcurre. Lorenz usó la frase "*efecto mariposa*" para referirse al fenómeno de la dependencia sensible a condiciones iniciales de un sistema caótico. Explicó que el fenómeno significa que una mariposa batiendo sus alas en Australia podría afectar el clima en los Estados Unidos un mes después. No fue sino hasta la década de 1970 que el trabajo de Lorenz se conoció en la comunidad matemática más teórica. Si bien Lorenz no predecía correctamente el clima, pero Lorenz entregaba una probabilidad aceptable del clima.

La palabra *caos* generalmente se refiere a un fenómeno que es desordenado e irregular. Sin embargo, en la terminología científica moderna se refiere a *caos* como un fenómeno pseudoaleatorio generado en un sistema determinista. En la década de 1970 el caos obtuvo alta relevancia. En 1971, Ruelle y Takens propusieron una nueva teoría para el inicio de la turbulencia en los fluidos, basada en consideraciones abstractas sobre atractores extraños (Ruelle y Takens, 1971). Alrededor de 1975, después de tres siglos de estudio, los científicos en gran número alrededor del mundo se dieron cuenta de que hay un tercer tipo de movimiento, un movimiento de tipo C, que ahora llamamos *caos*. La palabra *caos* fue introducida por primera vez por Li y Yorke en 1975 para designar sistemas que tienen un comportamiento aperiódico más complicado que el movimiento de equilibrio, periódico o cuasiperiódico (Li y Yorke, 1975), este trabajo es un caso especial del teorema obtenido por Sharkovskii en 1964 (Sharkovskii, 1964), que por razones políticas no era conocido por los matemáticos occidentales durante mucho tiempo. El nuevo movimiento denominado *caos* es errático, pero no simplemente cuasiperiódico con una gran cantidad de períodos, y no necesariamente debido a una gran cantidad de partículas que interactúan, es un tipo de comportamiento que se encuentra en sistemas incluso muy simples (Alligood *et al.*, 1996). En 1976, May mostró cómo surge el caos en mapas no lineales que modelan la dinámica poblacional, y escribió un influyente artículo que enfatizaba la importancia pedagógica de estudiar sistemas no lineales simples, para contrarrestar la enseñanza de los sistema lineales (May, 1976). Posteriormente el trabajo del físico Feigenbaum establece un vínculo entre el caos y las transiciones de fase, y atrajo a una generación de físicos al estudio de la dinámica caótica, Feigenbaum descubrió que existen ciertas leyes universales que

gobiernan la transición del comportamiento regular al caótico; en términos generales, sistemas completamente diferentes pueden volverse caóticos de la misma manera (Feigenbaum, 1978).

A principios de la década de 1980, los computadores se convierten en una herramienta poderosa en el estudio del caos, los investigadores realizaron pruebas numéricas en los computadores y obtuvieron complicadas estructuras de los atractores extraños que generan (Enns y McGuire, 1997; Tucker, 2002). Desde otro punto de vista, las simulaciones computacionales son una herramienta poderosa para ganar intuición sobre los sistemas no lineales y para explorar el terreno emocionante de la dinámica caótica, pero tienen sus limitaciones (Parker y Chua, 1989).

Benoit Mandelbrot construyó la teoría de la *geometría fractal* a finales de la década de 1970 y dibujó la primera imagen de un conjunto de Mandelbrot (Mandelbrot, 1983). La teoría de la geometría fractal generaliza el concepto de dimensión de enteros a números reales y se convirtió en una herramienta poderosa para caracterizar las estructuras complicadas de atractores extraños. Desde mediados de la década de 1980, cada vez más investigadores han prestado atención a la forma de controlar el caos y sus aplicaciones como sincronización y control de caos. Durante mucho tiempo se pensó que los sistemas caóticos eran impredecibles e incontrolables, dado que para una pequeña perturbación en las condiciones o en los parámetros iniciales del sistema caótico, se produce cambios drásticos en el movimiento original que crecen exponencialmente; incontrolables, porque las pequeñas perturbaciones generalmente conducen a otros estados caóticos opuestos a un movimiento estable y regular (Dyson, 1988). Sin embargo, el control de caos es de interés para aplicaciones en la industria y también para fines de investigación académica. En la literatura se reporta el primer trabajo sobre el control de caos en 1989 (Hübler, 1989), pero el método propuesto por Ott-Grebogi-Yorke (GOY) fue el trabajo que atrajo más la atención en la comunidad científica (Ott y Yorke, 1990).

Por otra parte, en 1990 Pecora y Carroll introducen otro campo de aplicación importante, que es la *sincronización de caos* (Pecora y Carroll, 1990). Varios años después, Chen y Lai propusieron un método para generar caos en un sistema de tiempo discreto no caótico (Chen y Lai, 1996). A partir de entonces, surgió un nuevo tópico de investigación, *anticontrol de caos*. Hasta ahora, se han propuesto, desarrollado y probado numerosos métodos de sincronización de caos, ver por ejemplo (Cruz-Hernández y Nijmeijer, 2000; Sira-Ramírez y Cruz-Hernández, 2001; Boccaletti *et al*., 2002; Wang y Zhang, 2002). Muchos experimentos demostraron que los sistemas físicos caóticos responden bastante bien a las estrategias de control recientemente desarrolladas, convencionales o novedosas, simples o sofisticadas. Las aplicaciones del control y sincronización de caos se proponen en áreas tan diversas de investigación como biología,

medicina, fisiología, epidemiología, ingeniería, química, física de láser, sistemas de energía eléctrica, mecánica de fluidos, aerodinámica, electrónica, comunicaciones, sistemas numéricos, sistemas mecánicos, etc. El control de caos se ha convertido en un proceso que maneja la dinámica de un sistema no lineal en una escala más amplia, con la esperanza de que se puedan derivar más beneficios (Zhang *et al.*, 2009).

2.2. Características principales de los sistemas caóticos

No hay una *definición de caos* universalmente aceptada hasta el momento, aunque de manera general, el mundo estaría de acuerdo con tres aspectos utilizados en la siguiente definición; se puede definir un sistema caótico como un sistema determinístico, regido por ecuaciones diferenciales o en diferencias no lineales, que presenta un comportamiento dinámico aparentemente aleatorio y que es sensible a condiciones iniciales. La teoría del caos puede definirse como el estudio cualitativo del comportamiento dinámico aperiódico mostrado por algunos sistemas deterministas no lineales. A continuación se mencionan algunas características principales que identifican a los sistemas caóticos (Devaney y Siegel, 1992; Schuster y Just, 2005; Zamorano, 2012):

- **Determinista**. Significa que los sistemas caóticos no tienen entradas o parámetros aleatorios o ruidosos. El comportamiento irregular (aperiódico) surge de la no linealidad del sistema, más que de las fuerzas motrices ruidosas.
- **Múltiples órbitas periódicas**. Significa que existen trayectorias que no convergen a puntos fijos, y el conjunto de órbitas periódicas u órbitas cuasiperiódicas *originan* que los sistemas caóticos manifiesten un espectro de frecuencia característico, similar al de una señal de ruido cuando el tiempo tiende a infinito. Por ejemplo, en la figura 1 se muestra el espectro de frecuencias de la implementación experimental de láseres caóticos de DFB y FP (Cardoza-Avendaño *et al.*, 2011).

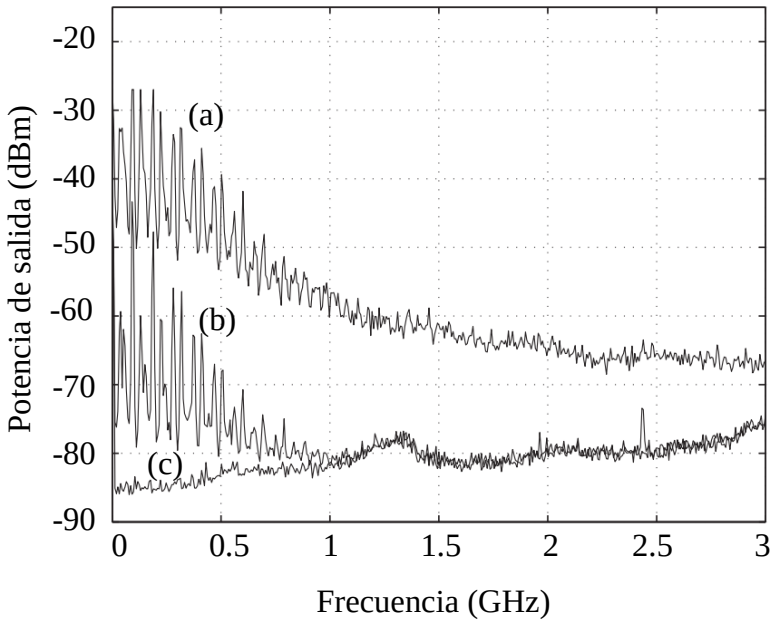
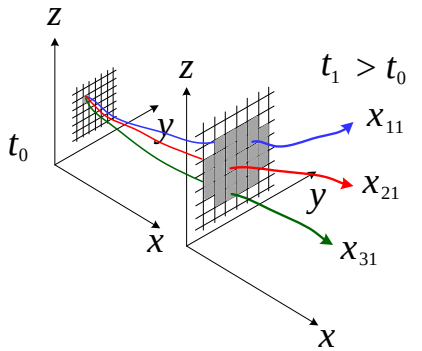
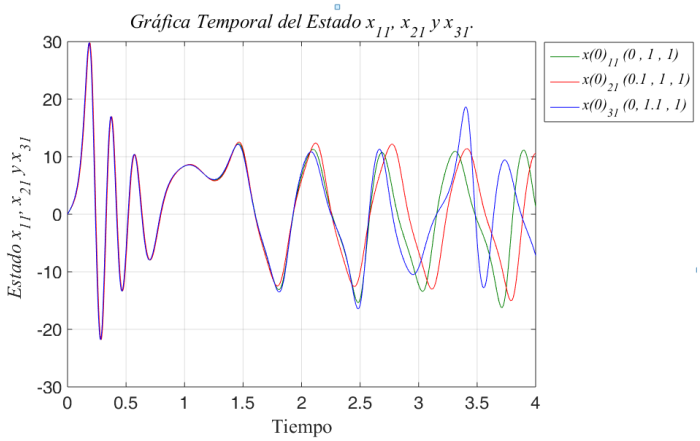


Figura 1. Espectros de frecuencias de la implementación experimental de láseres caóticos de FP: (a) con filtro adicional, intensidad de retroalimentación de 0.006%, (b) sin filtro, intensidad de retroalimentación de 0.006% y (c) sin filtro y retroalimentación externa (Cardoza-Avendaño *et al*., 2011).

- **Dependencia sensible de las condiciones iniciales**. A partir de condiciones iniciales diferentes, aunque muy cercanas una de otras, las trayectorias correspondientes que se producen tienden a ser distintas o a diverger exponencialmente conforme el tiempo transcurre, sin existir correlación alguna entre dichas trayectorias. En la figura 2 se muestra la evolución de las trayectorias del sistema caótico de Chen con condiciones iniciales similares, donde en su evolución temporal las trayectorias divergen (Chen, 1999).



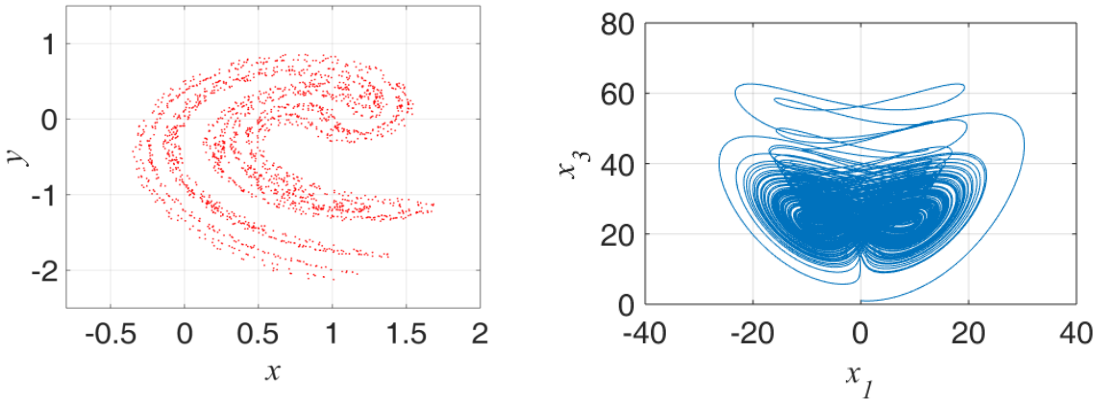
(a) Divergencia de las trayectorias bajo condiciones iniciales muy cercanas (Moon, 1992).



(b) Trayectorias temporales con diferentes condiciones iniciales.

Figura 2. Evolución temporal del sistema caótico de Chen inicializado bajo condiciones muy semejantes.

- **Generación de atractores extraños**. Otra característica propia de los sistemas caóticos es la aparición, en su diagrama de fase de estructuras geométricas con formas poco usuales (como punto fijo o ciclo límite) y sin ningún parecido a objetos geométricos clásicos, son los llamados *atractores extraños*, nombre dado a las estructuras asintóticas a donde evolucionan las órbitas de un sistema caótico. Dentro de las características mencionadas anteriormente, los atractores caóticos atraen los puntos vecinos al atractor y las órbitas de puntos vecinos entre si divergen. Un ejemplo de atractor extraño se presenta en la figura 3a, en la cual, se muestra el atractor caótico de Ikeda en tiempo discreto (Ikeda, 1979) y en la figura 3b se muestra un atractor extraño del de sistema caótico de Chen en tiempo continuo (Chen, 1999).



(a) Atractor caótico del mapa de Ikeda. (b) Atractor caótico del sistema de Chen.

Figure 3. Atractores extraños generados por sistemas caóticos: (a) mapa y (b) continuo.

- **Exponentes de Lyapunov positivos**. La característica crucial de los espacios con tres o más dimensiones que permite comportamiento caótico es la capacidad de las trayectorias a permanecer dentro de alguna región limitada sin intersecarse y sin repetirse de forma exacta. Dado que una de las manifestaciones más característica del comportamiento caótico es su sensibilidad a cambios en las condiciones iniciales del sistema, resulta lógico buscar la forma de medir el grado de sensibilidad de trayectorias vecinas a perturbaciones en sus condiciones iniciales, con vistas a caracterizar la "*caoticidad*" de un sistema. El concepto de divergencia exponencial de órbitas cercanas se formula con los exponentes de Lyapunov.

Los exponentes de Lyapunov se utilizan como medios para cuantificar la expansión y contracción de trayectorias vecinas en un sistema dinámico, es decir, dan una medida de la proporción exponencial, en la cual, órbitas cercanas se van apartando o acercando. En algún

sentido, determinan la “complejidad” de un sistema no lineal. Los exponentes de Lyapunov son una generalización de los valores propios en un sistema lineal.

Matemáticamente, los exponentes de Lyapunov de un sistema continuo se definen en términos de la siguiente expresión (Chen y Dong, 1998),

$$\lambda = \lim_{k\to\infty} \frac{1}{k} \ln \left| \frac{df^k(x_0)}{dx_0} \right|, \tag{1}$$

si el límite existe, siendo λ el *exponente de Lyapunov* de la trayectoria $x(k) = f^k(x_0)$, o bien del mapa f iniciando en el punto x_0. Existen algoritmos numéricos que permiten calcular los exponentes de Lyapunov en sus versiones continuas y discreta, uno de los algoritmos numéricos propuesto por (Wolf *et al.*, 1985; Briggs, 1990), el cual, es utilizado para el estudio de los sistemas caóticos continuos y discretos aludidos en esta tesis.

La convergencia de trayectorias a lo largo de una dirección en el espacio de estados, corresponde a un exponente de Lyapunov *negativo*. Mientras que la divergencia de trayectorias se caracteriza por un exponente de Lyapunov *positivo*. Mientras, una dirección neutra, sin converger ni diverger, corresponde a un exponente de Lyapunov con valor *cero*. El valor absoluto de estos exponentes cuantifica la velocidad de convergencia de las trayectorias. Por tanto, si un sistema tiene un exponente de Lyapunov positivo se dice que el sistema es *caótico* (condición necesaria) y si un sistema presenta dos o más exponentes de Lyapunov positivos, se dice que el sistema es *hipercaótico*. En la figura 4 se muestra la evolución temporal de los exponentes de Lyapunov del sistema caótico continuo MACM (Méndez-Ramírez *et al.*, 2017), se muestra que L_1 correspondiente a un exponente de Lyapunov positivo.

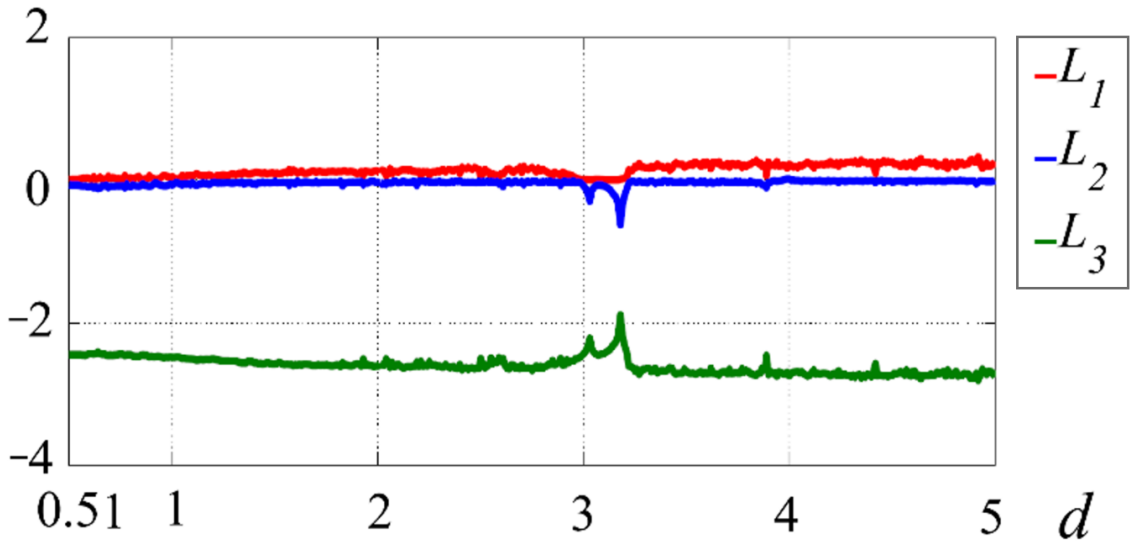
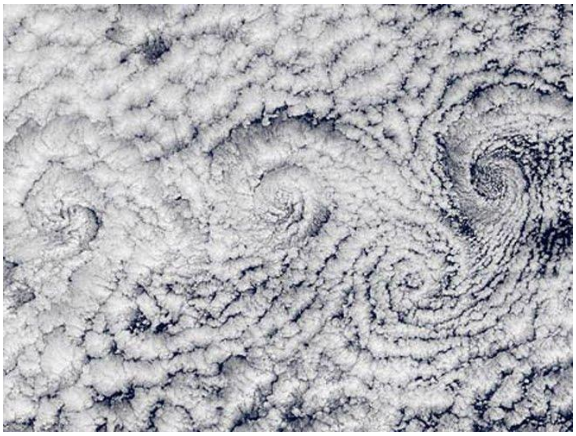


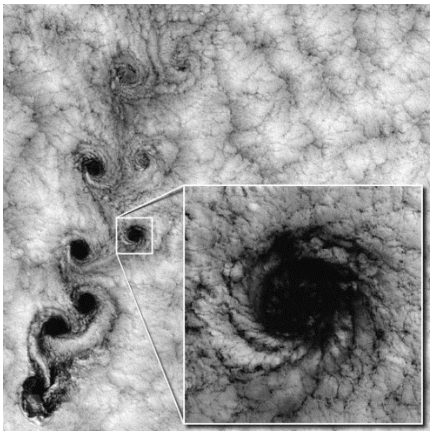
Figura 4. Exponentes de Lyapunov del sistema caótico MACM.

- **Dimensión fraccionaria**.

Los fenómenos del caos están descritos por ejemplo en la matemática fractal, que captura la infinita complejidad de la naturaleza. Reconocer la naturaleza fractal de nuestro mundo puede darnos una nueva percepción, poder y sabiduría (Barnsley, 2014). Muchos objetos naturales exhiben propiedades fractales, incluyendo paisajes, nubes, árboles, órganos, ríos, entre otras. Por ejemplo, la figura 5 muestra zonas meteorológicas de nubes en calma evolucionando con pautas fractales (vórtex), donde el campo de fuerzas atmosféricas puede incidir en comportamientos de orden y caos que están entretejidos por una bella armonía entre ellos, a través de los fractales se puede estudiar la forma de las nubes, cuando nos acercamos visualmente a una nube, observamos que está formada por muchos fragmentos de nubes y estos fragmentos a su vez en otros y así sucesivamente (Díaz-Brecia, 2009).



(a) Vórtex de nubes evolucionando con formas fractales.

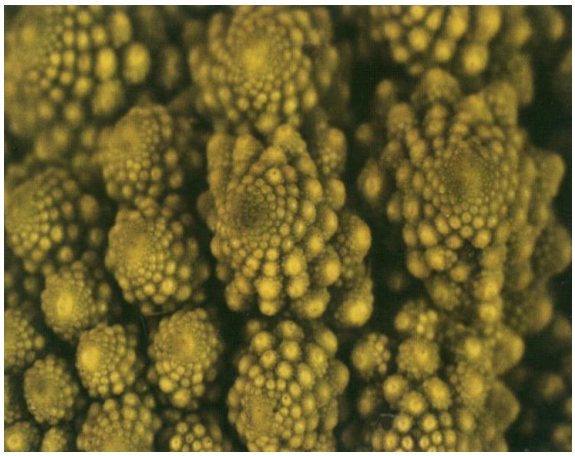


(b) Zonas de caos nubes con comportamientos atmosféricos caóticos

Figura 5. Representación de nubes.

En la figura 6 se muestra otro ejemplo de fractales presentes en la naturaleza, se trata de una variedad hibrida de coliflor y brécol italiano de la familia de las brasicaceas denominada romanesco; es una especie de coliflor que se asemeja a un vegetal de cierto experimento raro (es una de las especies de vegetales más llamativas). Está especie presenta características de geometría fractal en su estructura de forma natural; en donde su cabeza se compone de cabezas más pequeñas que imitan la forma de la cabeza más grande, y cada una de esas cabezas más pequeñas y el patrón progresivamente se repite sucesivamente. La cantidad de inflorescencias que compone el brecol romanesco es un número de Fibonacci (Glenn, 2002).

(a) Romanesco.



(b) Acercamiento del Romanesco.

Figura 6. Romanesco con inflorescencias de geometría fractal.

La *Real Academia Española* define a los fractales como: “Figura plana o espacial, compuesta de infinitos elementos, que tiene como propiedad que su aspecto y distribución estadística no cambian cualquiera que sea la escala con que se observe”. Los fractales tienen tres propiedades definitorias (Torres, 2005):

- **Autosimilaridad**. Sus partes tienen la misma forma o estructura que el todo,
- **Autorreferencia.** Se determina que el propio objeto aparece en la definición de sí mismo.
- **Dimensión fraccionaria.** Como figura geométrica no tiene una dimensión entera, como sucede con los puntos, segmentos de líneas, superficies planas, etc.

Por ejemplo, desde el punto de vista topológico se sabe que una circunferencia y un segmento rectilíneo son la misma curva y encierran el mismo tipo de superficie (pues es posible transformar una en la otra mediante una deformación continua es decir, sin que sea preciso someter a ninguna de las dos a manipulaciones “no topológicas”), pero desde un punto de vista métrico no son la misma curva ya que la circunferencia encierra un área finita el círculo y el segmento a pesar de ser finito, no encierra con su borde un área finita. Aparece entonces la necesidad de intentar clasificar los objetos y analizar cómo integrarlos en el mundo de los entes matemáticos.

La *dimensión topológica* D, es un término que introdujo Henri Poincaré para discernir sobre cuestiones de este tipo y la definición inductiva al introducir este concepto fue la siguiente (Poincaré, 1900):

- Un conjunto vacío posee dimensión topológica D = -1.
- Un punto posee dimensión topológica D = 0.
- Un segmento de línea posee dimensión topológica D = 1.
- Un cuadrado posee dimensión topológica D = 2.
- Un cubo posee dimensión topológica D = 3.
- Un hipercubo posee dimensión topológica D = 4.

La topología general ha evolucionado a una elaborada disciplina que interactúa con casi cualquier otra disciplina matemática. Conceptos básicos como límite, aplicaciones continuas, conexión y compacidad, se han convertido en piezas fundamentales de muchas clases de estructuras matemáticas. Estas ideas revolucionarias fueron propuestas por Felix Hausdorff en 1919, y dan lugar a los *espacios de dimensión no entera*, los cuales son aplicados en muchas áreas

importantes, incluyendo la teoría geométrica, la teoría de sistemas dinámicos y en la descripción de *fractales* (Hausdorff, 1919),

$$D = \frac{\log N}{\log\left(\frac{l}{p}\right)}, \tag{2}$$

en la cual, D describe la dimensión fractal, N la cantidad de unidades que forman el objeto, l la altura del objeto (proyección) y p la altura de las unidades que forman el objeto. Posteriormente, esta teoría es readaptada Besicovish (dimensión de Hausdorff-Besicovich); (Besicovitch, 1929). Generalizando, se establece que al obtener desde a números de copias semejantes a la original, con s como el factor de ampliación que se debe aplicar para obtener la figura original y por último la dimensión fractal y que corresponde a una simplificación del concepto de dimensión que utilizó Haussdorf, se verifica la siguiente relación,

$$a = \frac{1}{s^D}, \tag{3}$$

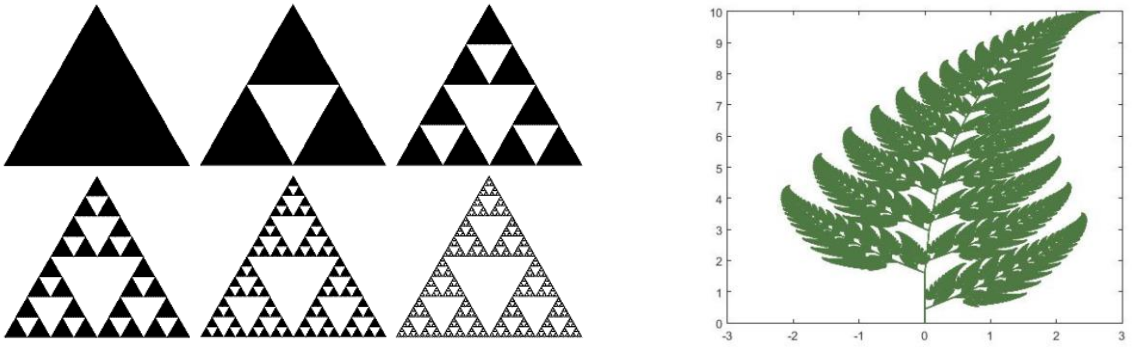
que es equivalente a,

$$D = \frac{\log a}{\log \frac{1}{s}}. \tag{4}$$

En términos de dimensión topológica, la dimensión fraccionaria de Hausdorff-Besicovich coincide por semejanza en el sentido de Poincaré o Devlin, la cual siempre es un número entero, pero hay ciertos objetos geométricos que no adoptan esta dimensión, a estos objetos geométricos los denominaremos *fractales* utilizando la terminología de Benoit Mandelbrot (Mandelbrot, 1983).

Dentro de la geometría fractal se puede distinguir dos tipos de fractales: Objetos construidos existen fractales regulares construidos a partir de objetos a partir de copias exactas y posteriormente utilizando reglas de transformación (escaladas) de sí mismos que reciben en nombre de *fractales regulares* y los objetos que son autosemejantes, pero no están construidos solo a partir de copias exactas de sí mismos donde su dimensión fractal es difícil de determinar o en algunos casos desconocida y reciben el nombre de *fractales irregulares*. En la figura 7a se

muestra un ejemplo de un fractal regular y la figura 7b muestra un fractal generado mediante algoritmo del helecho sintético creado por computadora de Barnsley.



(a) Triangulo de Sierpinski fractal regular. (b) Fractal sintético (irregular), helecho de Bansley.

Figura 7. Geometría fractal.

En el figura 8 se muestra un ejemplo de los fractales presentes en los sistemas caóticos donde se compara el conjunto de Mandelbrot con respecto al mapa logístico, dentro de la zona caótica del mapa logístico se presenta una alternancia de caos con ventanas de orden exponencial y periódico a diferentes escalas de observación, y esto concuerda con la propiedad de a*utosimilitud* que define a los objetos fractales (Peitgen *et al.*, 2004).

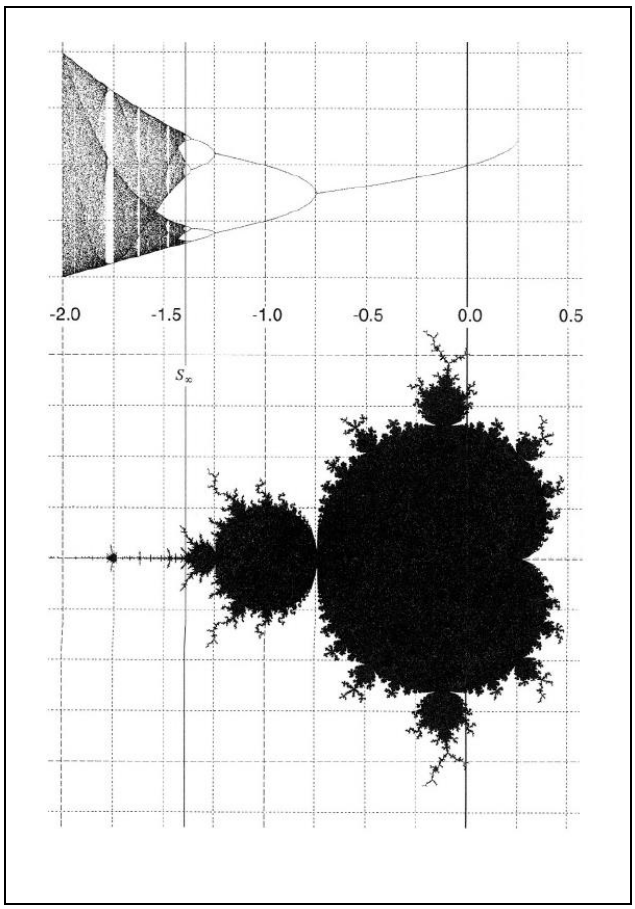


Figura 8. Diagrama de bifurcación de un mapa logístico caótico en comparación con el conjunto de Mandelbrot.

2.3. Determinación del caos

Para la comprobación de la existencia de caos en un sistema dinámico, existen 3 formas:

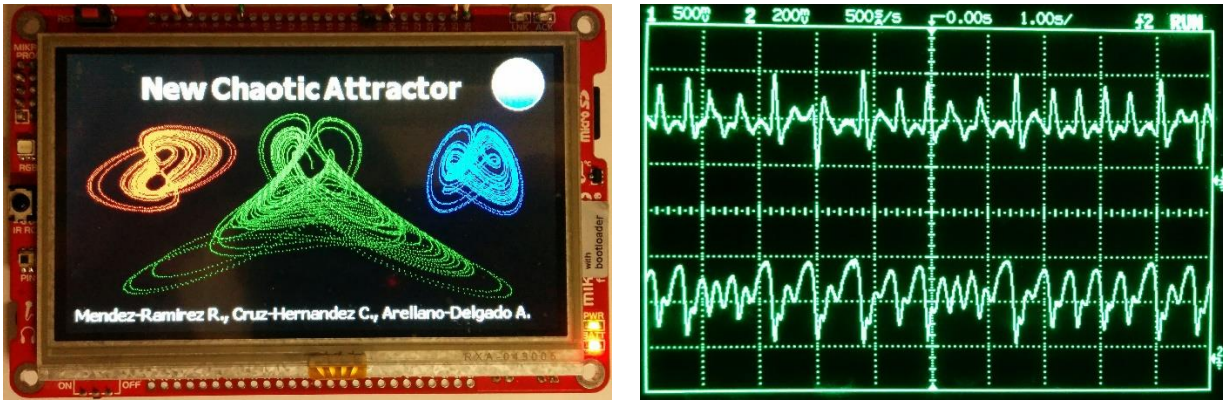
- **Analítica**. Mediante el teorema de Melnikov (Melnikov, 1963).
- **Numérica**. Principalmente, a través del cálculo de los exponentes de Lyapunov, en la literatura se reportan algunos algoritmos numéricos, por ejemplo (Wolf *et al.*, 1985) y (Briggs, 1990). También se pueden establecer zonas acotadas por la presencia de caos mediante los planos de Poincaré (Poincaré, 1899), determinación de la dimensión fractal, entre otros.
- **Experimental**. La prueba de 0-1 de Gottwald-Melbourne se aplica a cualquier sistema dinámico determinista. Al contrario del método usual de calcular el máximo exponente de Lyapunov, este método se aplica directamente a la serie de tiempo y no requiere reconstrucción espacio-fase.

Además, la dimensión del sistema dinámico y la forma de sus ecuaciones son irrelevantes. La entrada son los datos de la serie de tiempo y la salida se encuentra en el rango entre 0 y 1, dependiendo si la dinámica es caótica o no, siendo 0 no caótica y 1 totalmente caótica, ver detalles de la prueba en (Gottwald y Melbourne, 2004). Un ejemplo es utilizar este test para comprobar si el algoritmo las trayectorias generadas para un robot móvil, son efectivamente caóticas (Cetina, 2017), o por ejemplo en la emergencia de caos y sincronización de redes con osciladores periódicos discretos (Arellano-Delgado *et al*., 2017). En la implementación experimental de láseres caóticos de DFB y FP (Cardoza-Avendaño *et al*., 2011).

2.4. Aplicaciones del caos en la ingeniera

El papel de la teoría de caos es el de explicar un fenómeno que se presenta naturalmente, el cual no puede ser entendido por técnicas convencionales. Los sistemas caóticos han llamado la atención de la comunidad científica, debido a sus potenciales aplicaciones en varias ramas de la ciencia e ingeniería. Las características del caos son muy atractivas para incorporarse en algunos sistemas y aplicarlo por ejemplo en comunicaciones seguras, otras aplicaciones son por ejemplo el diseño de antenas o la planeación de trayectorias. La mayoría de estas aplicaciones involucran el análisis de series de datos obtenidos en los experimentos para buscar por ejemplo, un atractor extraño o calcular la dimensión fraccionaria y los exponentes de Lyapunov positivos. A continuación se presentan algunos casos en los que se aplica la teoría de caos en sistemas de ingeniería:

- **Sistema caótico en 3-D implementado en una pantalla inteligente Mikroe Plus PIC32MX7**. Este sistema embebido contiene un microcontrolador PIC32 de gama alta como parte central donde se reproduce el algoritmo basado en un sistema caótico de 3 dimensiones, en el cual, la evolución de sus trayectorias se reproducen en tiempo real (Méndez-Ramírez *et al.*, 2017). Un ejemplo interesante de esta aplicación, donde se reproducen los planos de fases, atractores y las trayectorias del sistema caótico se muestran en la figura 9.



(a) Planos de fase del nuevo atractor caótico MACM sobre pantalla táctil.

(b) Trayectorias de los estados utilizando DACs externos.

Figura 9. Reproducción del nuevo sistema caótico estilo Lorenz (MACM) utilizando el sistema embebido "Smart-display PIC32MX7".

- **Sistemas criptográficos basados en caos**: Algunas de las características del caos se encuentran implementadas experimentalmente en sistemas embebidos, donde es deseable, por ejemplo que los algoritmos de los *sistemas criptográficos* basado en caos sean de menor orden, en el cual el tiempo de complejidad permita que el sistema criptográfico soporte y procese grandes cantidades de información. La sensibilidad a las condiciones es una característica altamente deseable, por ejemplo al variar levemente los valores de los parámetros definidos en una llave secreta, ocasionaría que la encriptación de la información sea altamente segura y difícil de desencriptar. Los sistemas criptográficos basados en caos, pueden ser aplicados en comunicaciones seguras (Murillo-Escobar *et al*., 2015; Méndez-Ramírez *et al.*, 2017). Los sistemas caóticos basados en caos, pueden aplicarse en comunicaciones seguras, sistemas biométricos (Abúndiz-Pérez *et al*., 2016) y en sistemas de medicina a distancia (Murillo-Escobar *et al*., 2017).
- **Trayectorias impredecibles en robot móviles**: Adicionalmente, también se encuentran características del caos presentes en implementaciones en sistemas embebidos más robustos, donde cuentan con sistema operativo basado en Linux para controlar un robot móvil diferencial Khepera III y generar trayectorias caóticas impredecibles y se inducen comportamientos caóticos mediante sincronización de las velocidades lineales y angulares del robot con aplicaciones de vigilancia y patrullaje, ver por ejemplo (Cetina, 2017; Martins-Filho *et al*., 2004; 2005; 2007; Volos *et al*., 2012).

Por otra parte, se han reportado un gran número de trabajos en diferentes áreas científicas con el fin de resolver algunos problemas interesantes en la ingeniería, y algunos de estos ejemplos se mencionan brevemente a continuación:

- **Redes neuronales**. Se presentan técnicas de ingeniería para un nuevo tipo de procesamiento distribuido en paralelo utilizando un modelo de redes neuronales con comportamientos caóticos en modelos síncronos y asincrónicos, con cálculos complementarios con implementaciones analógicas y digitales (Aihara, 2002; Liao *et al*., 2001).
- **Diseño de radares**. Los sistemas caóticos poseen baja predictibilidad lo cual es deseable en el diseño de radares. La teoría del caos se utiliza para procesar la señal de regreso, generar códigos binarios, implementar sistemas codificados e implementar radares de señal ruidosa (NSR por sus siglas en ingles). Ejemplo de esto se puede encontrar en Leung y To, 1993; Strogatz, 1994; Liu *et al*., 2007 y Lin y Liu, 2004.
- **Antenas MIMO**. Diseño de antenas de múltiples entradas y múltiples salidas (MIMO) donde su algoritmo está basado en caos y permite optimizar la comunicación en un conjunto de antenas, ver por ejemplo (Fu *et al.*, 2014).
- **Quimiometria**. Una revisión del caos en la química y quimiometría es posible encontrar en la literatura donde se muestran análisis de complejidad a través de paradigmas alternativos de la química experimental en fractales, con presencia de caos en atractores extraños, en sistemas dinámicos, etc. Ver por ejemplo (Cramer y Booksh., 2006).
- **Antenas fractales**. Diseño de antenas fractales utilizando las entropías de Shannon, Rényi y Kolmogorov, las cuales, analizan y comparan la geometría fractal de la antena y el rendimiento físico junto con sus principales propiedades, ver por ejemplo (Figueroa-Torres *et al*., 2016; 2017; Guariglia, 2016).
- **Localización de anomalías en multitudes complejas**. Métodos para detectar y localizar anomalías complejas en el comportamiento de multitudes de personas, donde el comportamiento de las multitudes de personas describen trayectorias caóticas en sus dinámicas (Wu *et al.*, 2010).
- **Resonadores de microondas**. El caos se manifiesta en resonadores de microondas donde su comportamiento caótico imita al comportamiento de una bola en una mesa de billar y se establece experimentalmente un vínculo entre una función de la mecánica cuántica y las trayectorias de onda clásicas (Stockman y Ferry, 2006).

2.5. Sistemas caóticos

Los sistemas caóticos están representados por ecuaciones diferenciales no lineales, que son difíciles de resolver por métodos analíticos.

2.5.1. Sistemas caóticos de naturaleza discreta

Estos son sistemas cuya evolución se realiza por eventos, se describen por ecuaciones no lineales en diferencias. Para mapas unidimensionales los sistemas se pueden representar genéricamente por,

$$x_{n+1} = f\left(x_n\right), \tag{5}$$

donde cada estado actual x_n determina el siguiente estado x_{n+1} y las n iteraciones de la función f son órbitas dinámicas. Se asume el valor inicial x_0, la órbita será caótica si conduce a un estado aperiódico. Existen en la literatura una diversidad de sistemas caóticos de naturaleza discreta. Por ejemplo, en 1976, se reporta el mapa logístico como una ecuación de primer orden, el cual, tiene excelentes propiedades, dado que es de estructura simple (May, 1976). El Mapa logístico está descrito por

$$x_{n+1} = ax_n\left(1 - x_n\right), \tag{6}$$

en el cual, a corresponde al parámetro de control, que determina el grado de no linealidad del mapa logístico. Utilizando el valor inicial x_0 en el intervalo $0 \leq x_0 \leq 1$ y el valor del parámetro a en $0 \leq a \leq 4$, el mapa logístico (6) produce caos, en la figura 10 se muestra un ejemplo del diagrama de bifurcación del mapa logístico (6).

También en la literatura, se reportan otros sistemas caóticos de naturaleza discreta como el mapa de Hénon (Hénon, 1976), mapa de Ikeda (Ikeda, 1979), mapa de Tinkerbell (Alligood *et al.*, 1996), entre otros.

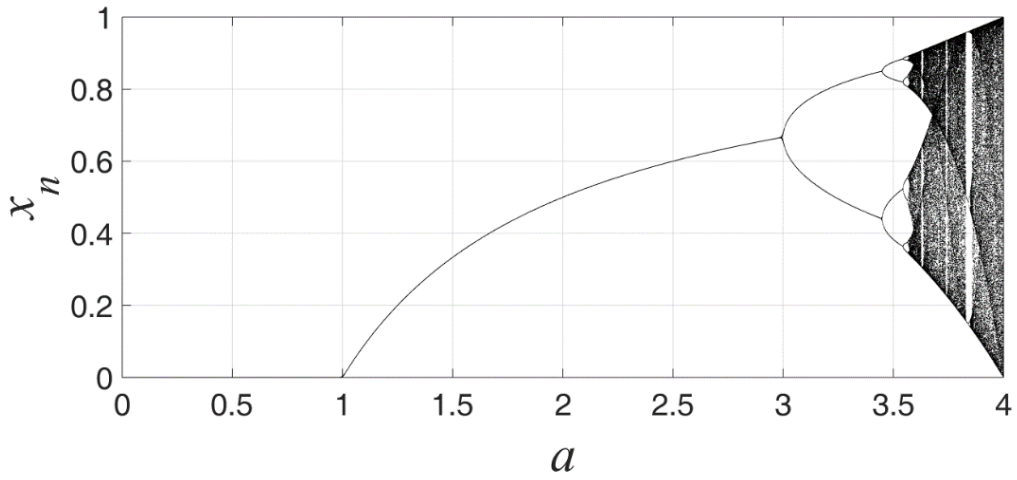


Figura 10. Diagrama de bifurcación del mapa logístico (6).

2.5.2. Sistemas caóticos continuos

Es estos sistemas, la evolución viene dada por la siguiente ecuación diferencial,

$$\dot{\mathbf{x}} = f(\mathbf{x}), \tag{7}$$

donde $\mathbf{x} \in \mathbb{R}^n$ representa el vector de estado del sistema dinámico con la condición inicial $\mathbf{x}(0)$ y $f : \mathbb{R}^n \to \mathbb{R}^n$ es una función no lineal. Como un ejemplo particular, el sistema de Lorenz se representa por el siguiente sistema de ecuaciones diferenciales no lineales (Lorenz, 1963),

$$\begin{cases} \dot{x} = \sigma(y - x), \\ \dot{y} = rx - xz - y, \\ \dot{z} = xy - bz. \end{cases} \tag{8}$$

Los valores standard de los parámetros del sistema caótico de Lorenz son: $\sigma = 10$, $r = 28$, $b = 8/3$. En la figura 11 se muestra la proyección del atractor caótico en 3 dimensiones del sistema caótico de Lorenz (8). También en la literatura especializada en el tópico, se reportan otros sistemas caóticos de naturaleza continua por ejemplo, el sistema de Rössler introducido en 1976 (Rössler, 1976), similarmente el sistema de Chen es introducido en (Chen, 1999), como un sistema dual al sistema de Lorenz, por otra parte el sistema de Liu y Chen fue reportado en 2002 (Liu y Chen, 2002). También existen sistemas caóticos de 4 dimensiones que presentan comportamientos hipercaóticos (Xiaohong y Zhiguang, 2013; Zhou, *et al.*, 2016; Arellano-Delgado *et al*., 2017), entre otros.

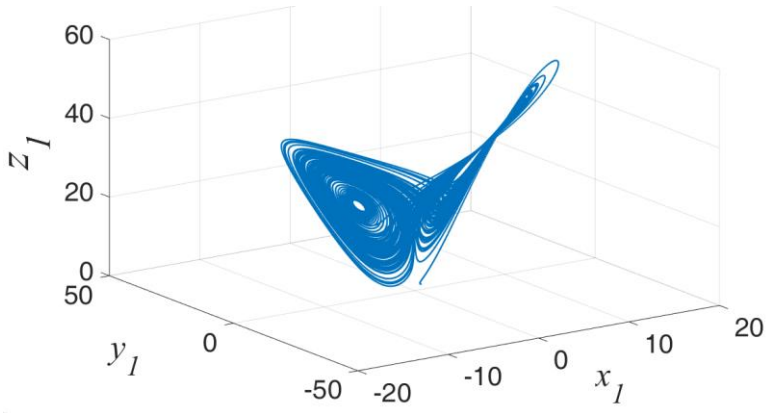


Figura 11. Proyección del atractor caótico del sistema de Lorenz en 3 dimensiones.

2.6. Conclusiones del capítulo

Este capítulo se concentró en describir las principales características y propiedades del caos. Se describió el Mapa Logístico de naturaleza discreta y el sistema caótico de 3 dimensiones de Lorenz de naturaleza continua. En particular, en los próximos capítulos se estudiará y se implementará el sistema caótico de naturaleza discreta de Hénon de 2 dimensiones y las versiones normalizadas de los sistemas de caóticos continuos en 3 dimensiones de Lorenz, Rössler, Chen, Liu y Chen y MACM, los cuales se estudiarán e implementarán experimentalmente en su representación discreta.

Capítulo 3. Construcción del sistema caótico MACM

En este capítulo, se presenta un nuevo sistema caótico autónomo tridimensional, que se deriva del sistema de Lorenz (Lorenz, 1963). La novedad del sistema caótico propuesto es que presenta la combinación de diferentes características: dos parámetros críticos, solo dos no linealidades, implementación electrónica de bajo costo; también es flexible y robusto con respecto a algunos atractores reportados en la literatura, ver por ejemplo (Chen y Ueta, 1999; Liu y Chen, 2002). Se realizan estudios numéricos y analíticos de las propiedades dinámicas para generar caos en el nuevo sistema MACM, recientemente reportado por Méndez-Ramírez *et al.*, (2017), publicación derivada de este trabajo de tesis doctoral. Finalmente, la versión continua del nuevo sistema caótico MACM se implementa mediante un circuito electrónico utilizando amplificadores operacionales.

3.1. Sistema caótico MACM

En esta primera sección se describen las ecuaciones de estado del sistema caótico MACM. Posteriormente se proporcionan las pruebas para verificar la existencia de caos. El nuevo sistema caótico MACM se construye a partir de la inspección y modificación del sistema caótico de Lorenz (1963). El sistema caótico MACM se describe por las siguientes ecuaciones diferenciales no lineales (Méndez-Ramírez *et al*., 2017),

$$\begin{cases} \dot{x} = -ax - byz, \\ \dot{y} = -x + cy, \\ \dot{z} = d - y^2 - z. \end{cases} \tag{9}$$

El sistema propuesto tiene siete términos, dos no linealidades y cuatro parámetros, para el cual, con valores $a = 2$, $b = 2$, $c = 0.5$, $d = 4$ se demostrará que genera dinámicas caóticas mediante análisis y pruebas reportados más adelante en este capítulo. Además, este sistema caótico tiene dos parámetros críticos de bifurcación b y d.

3.1.1 Análisis del sistema caótico MACM

El sistema dinámico MACM (9) es simétrico respecto al eje z, debido a su invariancia bajo la transformación de coordenadas $(x, y, z)\rightarrow(-x, -y, z)$. La simetría no está asociada con los parámetros a, b, c y d. La divergencia para el sistema dinámico MACM (9), se encuentra definida por

$$\nabla V = \frac{\partial \dot{x}}{\partial x} + \frac{\partial \dot{y}}{\partial y} + \frac{\partial \dot{z}}{\partial z} = -a + c - 1 = -2.5 < 0. \tag{10}$$

Por tanto, el análisis anterior prueba que el sistema dinámico MACM (9) es disipativo. La tasa de contracción exponencial del sistema se calcula por la siguiente expresión,

$$\frac{dy}{dx} = (\nabla V)V \rightarrow V = V_0 e^{-2.5t}. \tag{11}$$

Cada volumen que contiene la trayectoria del sistema se contrae a cero cuando $t\rightarrow\infty$ a una tasa exponencial de $-2.5t$. Las órbitas del sistema dinámico MACM (9) se limitan en última instancia a un conjunto límite específico de volumen cero y el movimiento asintótico se instala en un atractor. De este modo, se demuestra la existencia del atractor.

El acotamiento de las trayectorias caóticas del sistema dinámico MACM (9) se demuestra a través del el siguiente teorema. Trabajos similares están reportados por ejemplo en la literatura, ver (Yang y Chen, 2008; Nik y Golchaman, 2014).

Teorema 1 (Méndez-Ramírez *et al.*, 2017): Asuma que los parámetros a, b, c y d del sistema dinámico MACM (9) son positivos. Entonces, las órbitas del sistema (9) incluyendo las órbitas caóticas están confinadas en una región acotada.

Prueba: Considérese la siguiente función candidata a función de Lyapunov

$$V(x, y, z) = \frac{1}{2}\left(x^2 + y^2 + z^2\right). \tag{12}$$

La derivada temporal de la función $V(x, y, z)$ a lo largo de las trayectorias del sistema dinámico MACM (9), queda expresada como

$$\dot{V}(x,y,z) = x\dot{x} + y\dot{y} + z\dot{z}. \tag{13}$$

$$\dot{V}(x,y,z) = -ax^2 - bxyz + cy^2 - xy + dz - zy^2 - z^2$$

$$\dot{V}(x,y,z) = -\left(\frac{byz}{2\sqrt{a}} + \sqrt{a}x\right)^2 + \left(\frac{x}{2\sqrt{c}} - \sqrt{c}y\right)^2 - \frac{x^2}{4c} - \left(z - \frac{d}{2}\right)^2 + \frac{d^2}{4} + \left(\frac{byz}{2\sqrt{a}} - \frac{\sqrt{a}y}{b}\right)^2 - \frac{ay^2}{b^2}.$$

Sea R_0 la región suficientemente grande para que todas las trayectorias (x, y, z) satisfagan que $V(x, y, z) = R$ para $R > R_0$ con la condición,

$$\left(\frac{byz}{2\sqrt{a}} + \sqrt{a}x\right)^2 + \left(z - \frac{d}{2}\right)^2 + \frac{ay^2}{b^2} + \frac{x^2}{4c} > \left(\frac{x}{2\sqrt{c}} - \sqrt{c}y\right)^2 + \left(\frac{byz}{2\sqrt{a}} - \frac{\sqrt{a}y}{b}\right)^2 + \frac{d^2}{4}. \tag{14}$$

En consecuencia, en la superficie $\{(x, y, z)/V(x, y, z)\} = R$. Puesto que, $R > R_0$ podemos escribir, $\dot{V}(x, y, z) < 0$, o como el conjunto $\{(x, y, z)/V(x, y, z)\} \leq R$ es una región confinada para todas las trayectorias caóticas del sistema dinámico MACM (9).

3.1.2 Puntos de equilibrio

Los puntos de equilibrio y sus estabilidades determinan el comportamiento dinámico del sistema MACM (9), estos puntos se pueden encontrar estableciendo $\dot{x} = \dot{y} = \dot{z} = 0$ y $a, b, c, d > 0$. El sistema propuesto (9) tiene cinco puntos fijos: $P_0(0,0,0)$, P_1, P_2, P_3, $P_4\left(\pm c\sqrt{d + \frac{ac}{b}}, \pm\sqrt{d + \frac{ac}{b}}, -\frac{ac}{b}\right)$. La matriz jacobiana del sistema dinámico MACM (9) se define por,

$$J_{VC} = \begin{pmatrix} -a & -bz & -by \\ -1 & c & 0 \\ 0 & -2y & -1 \end{pmatrix}, \tag{15}$$

el polinomio característico de la matriz jacobiana (15) viene dado por

$$det(\lambda I - J) = \lambda^3 + (a - c + 1)\lambda^2 + (a - c - ac - bz)\lambda + 2by^2 - ac - bz = 0. \tag{16}$$

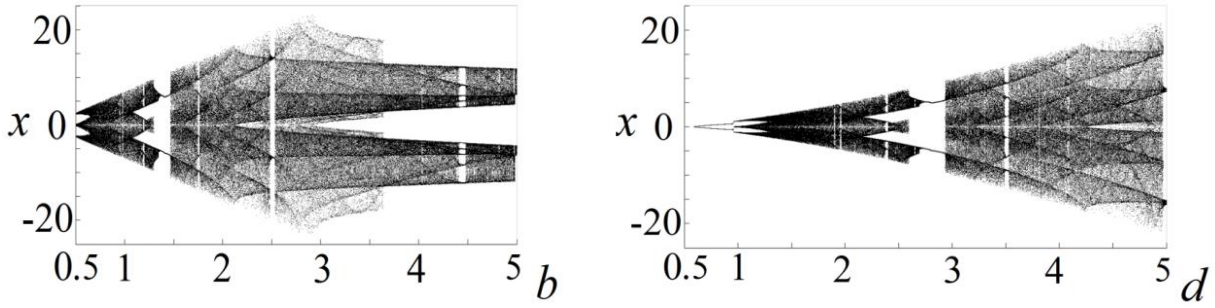
Evaluando los parámetros $a = 2$, $b = 2$, $c = 0.5$ y $d = 4$ en el polinomio característico (16), se estudió la estabilidad de los puntos de equilibrio $P_{0,1,2,3,4}$ (los puntos $P_{1,2,3,4}$ son los mismos). La tabla 1 muestra los resultados de la estabilidad de los puntos de equilibrio del sistema dinámico MACM (9), los cuales son de tipo silla inestable.

Tabla 1. Estabilidad de los puntos de equilibrio del sistema dinámico MACM (9).

Punto de equilibrio	Valores propios	Estabilidad
P_0	$\lambda_1 = -2$, $\lambda_2 = -1$, $\lambda_3 = 0.5$	λ_1 , $\lambda_2 < 0$ y $\lambda_3 > 0$, es un punto silla inestable
$P_{1,2,3,4}$	$\lambda_1 = -3.52387$ $\lambda_2 = 0.51193 + 2.20134i$ $\lambda_3 = 0.51193 + 2.20134i$	$\lambda_1 < 0$ y la parte real de λ_2, $\lambda_3 > 0$, son puntos de silla inestable

3.1.3 Diagramas de Bifurcación

El diagrama de bifurcación se construye con el propósito de visualizar las transiciones entre los comportamientos periódicos y caóticos generados por el sistema dinámico MACM (9), el cual, se construye a partir de la variación de los valores de los parámetros críticos b o d; para mayor información del algoritmo numérico empleado para construir los diagramas de bifurcación, ver por ejemplo (Feigenbaum, 1980). Las condiciones iniciales son $x(0) = y(0) = z(0) = 1$ y el conjunto de los parámetros $a = 2$, $b = 2$, $c = 0.5$ y $d = 4$ son escogidos para todas las pruebas numéricas y experimentales. Para construir uno de los diagramas de bifurcación del sistema MACM se considera el parámetro $d = 4$ fijo y el parámetro b es escogido como el parámetro de bifurcación, ver figura 12a. Mientras, en la figura 12b se ilustra el otro diagrama de bifurcación, obtenido al considerar $a = 2$, $b = 2$ y $c = 0.5$ y d es variado. Como consecuencia de la figura 12, podemos concluir que el sistema dinámico MACM (9) es flexible y robusto porque en las zonas de los parámetros de bifurcación b y d, se tienen comportamientos dinámicos: punto fijo, ciclo límite y atractor extraño.



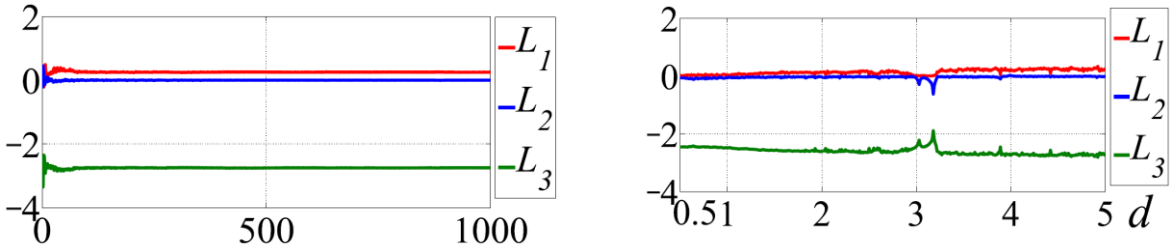
(a) Variación del parámetro b = [0.5, 5]. (b) Variación del parámetro d = [0.5, 5].

Figura 12. Diagrama de bifurcación de los parámetros b y d, versus la variable de estado x del sistema MACM (9) utilizando las condiciones iniciales $x(0)= y(0)= z(0)=1$.

3.1.4 Exponentes de Lyapunov

Para probar numéricamente la presencia de caos en el sistema dinámico MACM (9), se calculan los exponentes de Lyapunov, en este trabajo utilizamos el método reportado en (Wolf *et al.*, 1985) y (Briggs, 1990).

La figura 13 muestra los exponentes de Lyapunov. La figura 13a muestra la evolución de los exponentes de Lyapunov considerando 1000 unidades de tiempo, donde se obtuvieron los siguientes valores: $L_1 = 0.24914$, $L_2 = 0$ y $L_3 = -2.7497$. La figura 13b muestra la evolución de los exponentes de Lyapunov considerando la variación del parámetro de bifurcación d = [0.5, 5].



(a) Parámetro de bifurcación d = 4 considerando 1000 unidades de tiempo. (b) Variación del parámetro de bifurcación d = [0.5, 5].

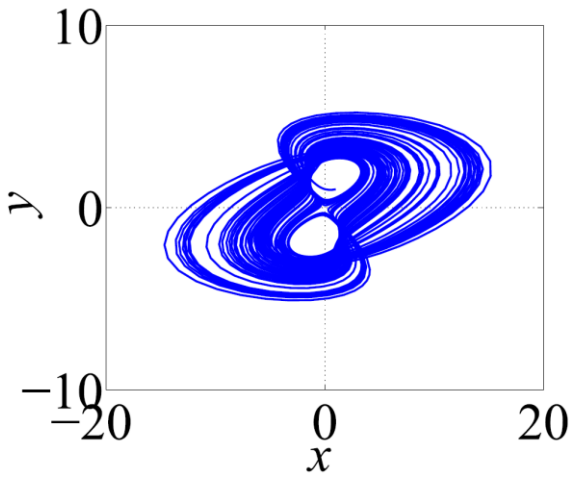
Figura 13. Exponentes de Lyapunov del sistema caótico MACM (9).

3.1.5 Diagramas de Kaplan-Yorke

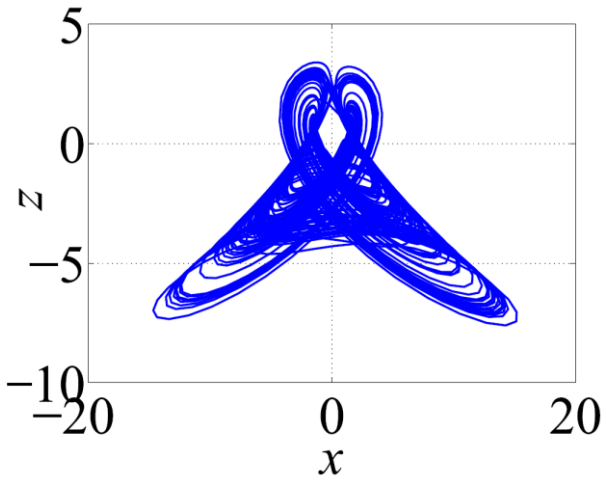
Otro instrumento numérico de determinación de caos la presencia de caos en un sistema dinámico, es recurrir a la dimensión fractal de Kaplan-Yorke D_{KY}, obtenida mediante la siguiente expresión,

$$D_{KY} = j + \frac{1}{|L_{j+1}|}\sum_{i=1}^{j} L_i = 2 + \frac{L_1 + L_2}{|L_3|} = 2.0908. \tag{17}$$

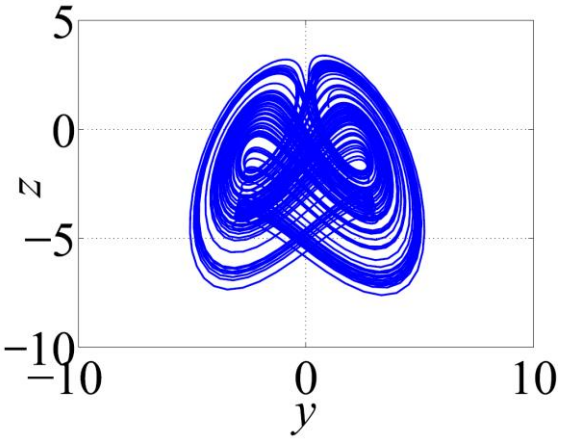
Finalmente, en la figura 14 se muestran los atractores caóticos generados por el sistema MACM (9), proyectados en 2D en la figura 14a-14c y en 3D en la figura 14d.



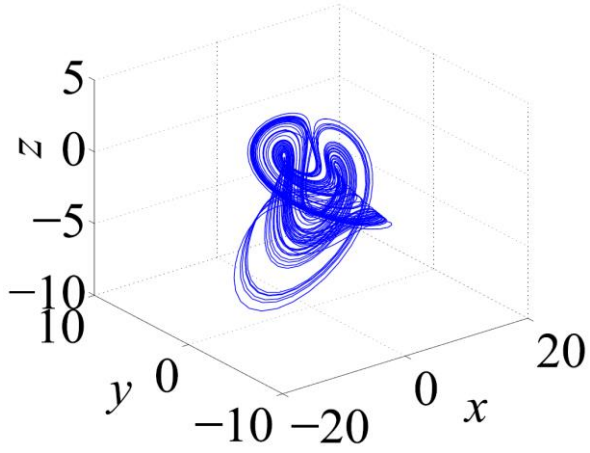
(a) Plano de fase x versus y.



(b) Plano de fase x versus z.



(c) Plano de fase y versus z.



(d) Proyección del atractor x versus y versus z.

Figura 14. Atractor caótico generado por el nuevo sistema MACM (9).

3.2. Implementación electrónica del sistema caótico MACM

Para la implementación electrónica del sistema en versión continua del sistema caótico MACM (9), se empleó el factor de atenuación de 20 para cada variable de estado del sistema, es decir: $x = 20u$, $y = 20v$, $z = 20w$. Al reemplazar las nuevas variables en el sistema caótico MACM (9), obtenemos el siguiente sistema,

$$\begin{cases} \dot{u} = -2u - 40vw, \\ \dot{v} = -u + 0.5v, \\ \dot{w} = 0.2 - 20v^2 - w. \end{cases} \tag{18}$$

Reemplazando las variables de estado $x = u$, $y = v$, $z = w$ en el sistema (18), entonces la representación del circuito electrónico es,

$$\begin{cases} \dot{x} = \dfrac{1}{RC_1}\left(-\dfrac{R}{R_1}x - \dfrac{R}{10R_2}yz\right), \\ \dot{y} = \dfrac{1}{RC_2}\left(-x + \dfrac{R}{R_5}y\right), \\ \dot{z} = \dfrac{1}{RC_3}\left(\dfrac{R}{R_9}b - \dfrac{R}{10R_8}y^2 - z\right), \end{cases} \tag{19}$$

en el cual, los componentes electrónicos son: 3 amplificadores operacionales TL084, 2 multiplicadores AD633, 3 condensadores $C_1 = C_2 = C_3 = 100$ pF, resistencias: $R_1 = 500$ kΩ, $R_2 = 47$ kΩ, $R = R_{10} = 1$ MΩ, $R_3 = 2$ MΩ, $R_6 = 100$ kΩ, $R_4 = R_5 = R_8 = 10$ kΩ, $R_9 = R_{12} = R_{13} = 10$ kΩ y $R_7 = 5$ MΩ, el parámetro de bifurcación se fija en el valor $d = 4$ con el valor de la resistencia $R_{11} = 287$ kΩ, y el circuito fue energizado con los voltajes $+V_{cc} = 18$ V y $-V_{cc} = -18$ V. Para ver los cambios dinámicos se recomienda utilizar una resistencia variable $R_{11(VAR)} = [0, 1\ \text{MΩ}]$ y este voltaje es definido como V_d. En la figura 15 se muestra el circuito electrónico equivalente del sistema caótico MACM (19).

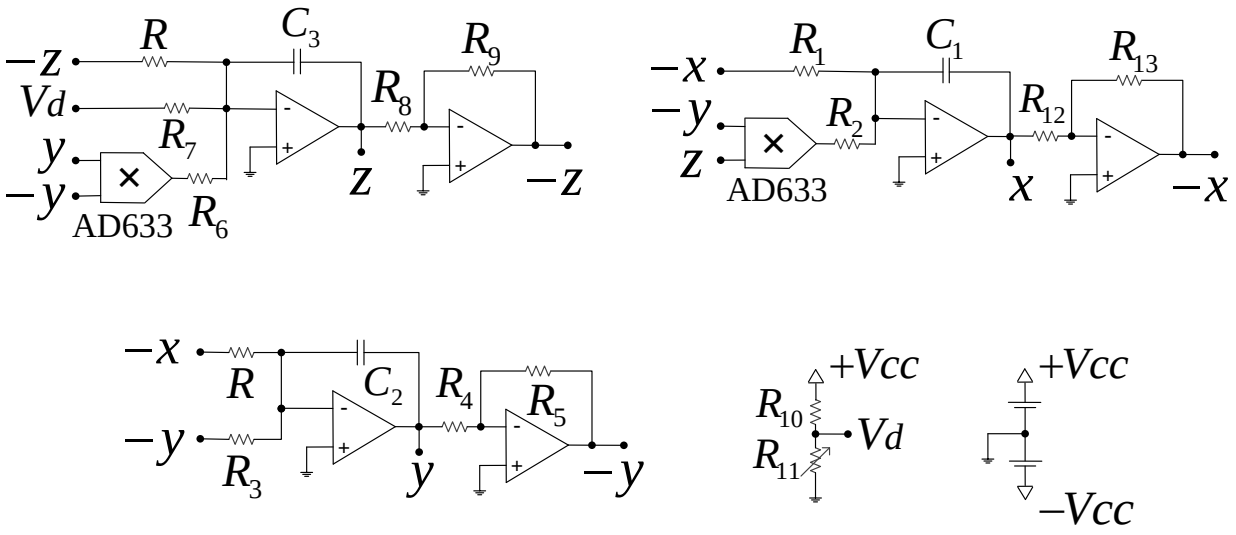
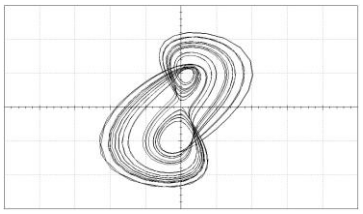
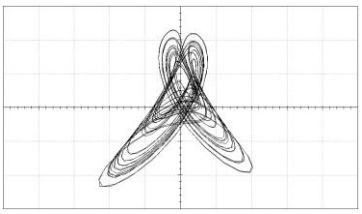


Figura 15. Diagrama esquemático del circuito electrónico equivalente del sistema caótico MACM (19).

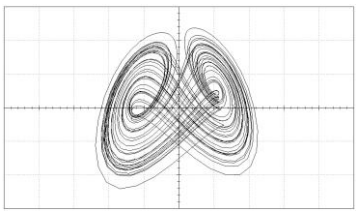
La figura 16 muestra la comparación de los planos de fase del sistema caótico MACM (19) entre la simulación (utilizando el software de simulación de circuitos electrónicos Multisim) y la implementación del circuito electrónico, donde los planos de fase son similares con las simulaciones representadas en la figura 14.



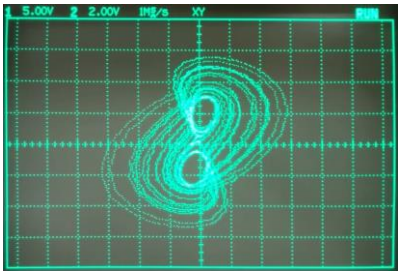
(a) Plano de fase x versus y.



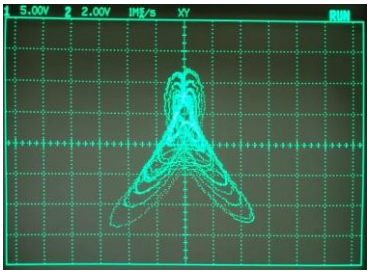
(b) Plano de fase x versus z.



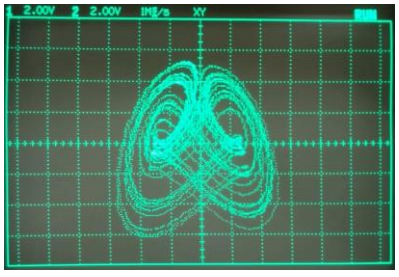
(c) Plano de fase y versus z.



(d) Plano de fase x versus y.



(e) Plano de fase x versus z.



(f) Plano de fase y versus z.

Figura 16. Comparación de los planos de fase simulación utilizando Multisim figuras (a), (b) y (c) y la implementación electrónica figuras (e), (f) y (g) del sistema caótico MACM (19).

3.3. Conclusiones del capítulo

En este capítulo se reportó la propuesta del sistema caótico MACM, capaz de generar dinámicas caóticas variando dos parámetros de bifurcación. Se presentaron estudios analíticos y numéricos para confirmar la generación de caos para la versión continua del sistema MACM. Los resultados mostraron que el sistema caótico MACM es flexible y robusto, lo que permite obtener una riqueza en comportamientos caóticos. Como consecuencia de todas las características que presenta el sistema MACM, resulta de alta facilidad de implementación y puede ser de gran interés en aplicaciones de ingeniería, por ejemplo en criptografía, sistemas biométricos, telemedicina y comunicaciones seguras, ver por ejemplo (Murillo-Escobar *et al*., 2015; 2017; 2017; Mendez-Ramírez *et al*., 2017). Adicionalmente, el sistema caótico MACM se simuló numéricamente y se implementó electrónicamente la versión continua con amplificadores operacionales. En los próximos capítulos de la tesis, se estudiaran las propiedades del nuevo atractor caótico en su versión discreta.

El material de este capítulo se recoge del artículo publicado en (Méndez-Ramírez, *et al.*, 2017).

Capítulo 4. Sistema criptográfico digital basado en caos utilizando el protocolo SPI e implementado sobre dsPICs

En este capítulo se describe el diseño e implementación de un algoritmo de encriptación caótico, aplicado al protocolo de comunicación SPI mediante el uso de microcontroladores dsPIC. El protocolo de interfaz periférica serie (SPI) comúnmente se utiliza por los fabricantes de sistemas embebidos y circuitos integrados para aplicaciones en áreas como las comunicaciones por cable e inalámbricas. El diseño del algoritmo del sistema criptográfico se basa en caos, utilizando el mapa caótico de Hénon y su versión depurada, para el cual, los métodos de difuminación y permutación (en combinación con secuencias de ADN). El protocolo SPI se configura en 16 bits para sincronizar transmisor y receptor, para el cual, se considera una clave secreta simétrica. Los resultados se comprueban experimentalmente en un sistema embebido, donde la parte central consta de dos microcontroladores dsPIC de bajo costo. Para procesar, adquirir y reconstruir mensajes confidenciales, se utilizan las propiedades para el procesamiento de señales digitales y se implementa un convertidor digital-analógico (DAC) con interface SPI. Por último, la seguridad del criptograma se verifica a través de pruebas estadísticas. La capacidad de procesamiento digital del algoritmo se valida por el microcontroladores dsPIC.

4.1. Descripción del sistema criptográfico caótico embebido

En esta sección se reportan las propiedades del protocolo SPI, la descripción del hardware del sistema embebido, la generación de secuencias seudo aleatorias basadas en un mapa caótico discreto y la prueba de sensibilidad con respecto a la definición de la llave secreta.

4.1.1. Breve descripción de la criptografía moderna en sistemas embebidos

La criptografía moderna reporta diferentes técnicas de encriptación, por ejemplo, para la encriptación se utilizan algoritmos como el Triple DES (Estándar de Encriptación de Datos, conocido como TDES o 3DES), el Estándar Avanzado de Encriptación (AES, por sus siglas en inglés) o el algoritmo Internacional de Cifrado de Datos (IDEA, por sus siglas en inglés) (Muhaya *et al*., 2009). Además, en la literatura se reportan el uso de mapas caóticos y sistemas hipercaóticos para cifrar información para diferentes aplicaciones, como la

comunicación segura de correo electrónico (Aguilar-Bustos *et al*., 2010), para sistemas criptográficos basados en caos utilizando PDS (Guglielmi *et al*., 2009; Rhouma y Belghith, 2006), por ejemplo en la sincronización experimental de redes por fibra óptica de plástico (Arellano-Delgado *et al*., 2013), para la sincronización robusta de sistemas caóticos mediante el control por modos deslizantes y control por retroalimentación (Li-li *et al*., 2014) y recientemente para algoritmos de cifrado de imágenes RGB basados en caos (Murillo-Escobar *et al*., 2015, Liu *et al*., 2016), sistemas biométricos (Abúndiz-Pérez *et al*., 2016), encriptado para información médica en telemedicina (Murillo-Escobar *et al*., 2017), y comunicaciones ópticas (López Gutiérrez *et al*., 2009; Cardoza-Avendaño *et al*., 2010; 2012; Juang *et al*., 2000; Uchida *et al*., 2012; Arellano-Delgado *et al*., 2013), entre otros.

Además, para implementar sistemas caóticos en un sistema embebido, se necesitan procesadores robustos basados en chips para el procesamiento digital de señales como DSP o FPGA porque sus arquitecturas son robustas y estos microprocesadores permiten mayor capacidad de procesamiento (Azzaz *et al*., 2013). En este sentido, los microcontroladores en los sistema embebidos representan una alternativa económica para aplicaciones de cifrado utilizando caos para generar secuencias pseudoaleatorias (Murillo-Escobar *et al*., 2015). Los microcontroladores dsPIC son una interesante alternativa para la implementación de sistema embebidos de bajo costo, la arquitectura y las propiedades para procesamiento digital de señales permiten realizar cálculos matemáticos con precisión. Estos microcontroladores reúnen condiciones suficientes como parte principal de un sistema embebido (Di Jasio, 2007). Preliminarmente, se reportan algunos estudios de sistemas embebidos utilizando microcontroladores dsPIC, por ejemplo: el procesamiento de mensajes múltiples usando dsPIC y DACs (Siddiqui *et al*., 2015) y generador de números pseudoaleatorios para el tratamiento de *tinnitus* (término médico para el hecho de "escuchar" ruidos en los oídos, cuando no hay una fuente sonora externa) implementado en un dsPIC (Uriz *et al*., 2016), entre otros.

4.1.2. Protocolo de comunicación SPI

El protocolo SPI se utiliza para la comunicación de los sistemas embebidos porque tiene una configuración sencilla, transmisión rápida en serie de bus de datos y líneas de baja cantidad para conectar otros periféricos (circuitos integrados o dispositivos). Muchos fabricantes de microcontroladores y microprocesadores adoptan el protocolo de comunicación SPI para implementarse directamente en hardware de uno o tres puertos designados. El protocolo de comunicación de transmisión de datos es

dúplex completo y los modos de operación son maestro y esclavo. El protocolo SPI especifica 4 cables conectados mediante el uso de pasadores externos. Una de estas clavijas es la salida maestra a entrada esclava (MOSI), se utiliza para conectar desde un maestro a dispositivos esclavos, también este pin conocido como salida de datos en serie (SDO). El terminal en modo mastro-entrada a esclavo-salida (MISO) se utiliza para conectar el maestro a los dispositivos esclavos, además, este terminal se conoce como entrada de datos en serie (SDI). El reloj serial descrito en el pin (SCK) se utiliza para sincronizar la transferencia de datos desde un dispositivo maestro a un dispositivo esclavo (o varios dispositivos en modo esclavo). El selector esclavo (SS) es el pin que selecciona de maestro a dispositivo esclavo (Motorola Inc., 2003).

Los microcontroladores dsPICs cuentan con un módulo SPI configurado mediante el uso de registros internos que consisten principalmente en un registro de desplazamiento de 16 bits. Este registro de desplazamiento (SPIεSR, donde ε indica el número de módulo SPI) se utiliza para desplazar datos dentro y fuera del registro de memoria intermedia (SPIεBUF). El registro de control (SPIεCON) configura el módulo SPI. El registro estadístico (SPIεSTAT) muestra las condiciones de estado de las operaciones del módulo SPI, en el cual, la bandera SPIRBF permite verificar la recepción de la palabra con 16 bits de otro dispositivo externo se ha realizado (Microchip Technology Inc., 2004).

En este estudio, hemos propuesto la configuración del protocolo SPI con 16 bits mediante el uso de dos dsPICs y DACs externos. Para programar los dsPICs usamos el compilador Mikroc Pro para dsPIC (Mikroelektronika, 2017). La figura 17 muestra el diagrama de bloques completo del sistema criptográfico caótico embebido para transmitir y recibir mensajes cifrados usando el protocolo SPI. El transmisor dsPIC (configurado en modo maestro) procesa primero el mensaje original $m(t)$, este mensaje se encripta y se transmite mediante el protocolo SPI. El otro dsPIC se considera principalmente como receptor pero su función es de transceptor (la configuración del protocolo SPI para este dsPIC cambia de modo esclavo a modo maestro), porque primero recibe y desencripta el mensaje original $m(t)$ y después transmite el mensaje recuperado $m'(t)$. Finalmente, utilizamos un DAC externo (configurado como modo esclavo) para reconstruir el mensaje $m'(t)$.

4.1.3. Descripción del hardware del sistema embebido implementado

Los beneficios de los microcontroladores dsPIC se utilizan para obtener y probar los resultados utilizando el protocolo de comunicación SPI con aplicaciones DSP. La tabla 2 muestra la descripción de hardware asignada a cada unidad del sistema embebido propuesto para conectar los circuitos integrados a través del protocolo SPI. Los resultados del procesamiento de los algoritmos para encriptar y desencriptar se implementan en el transmisor U1 y el receptor U2.

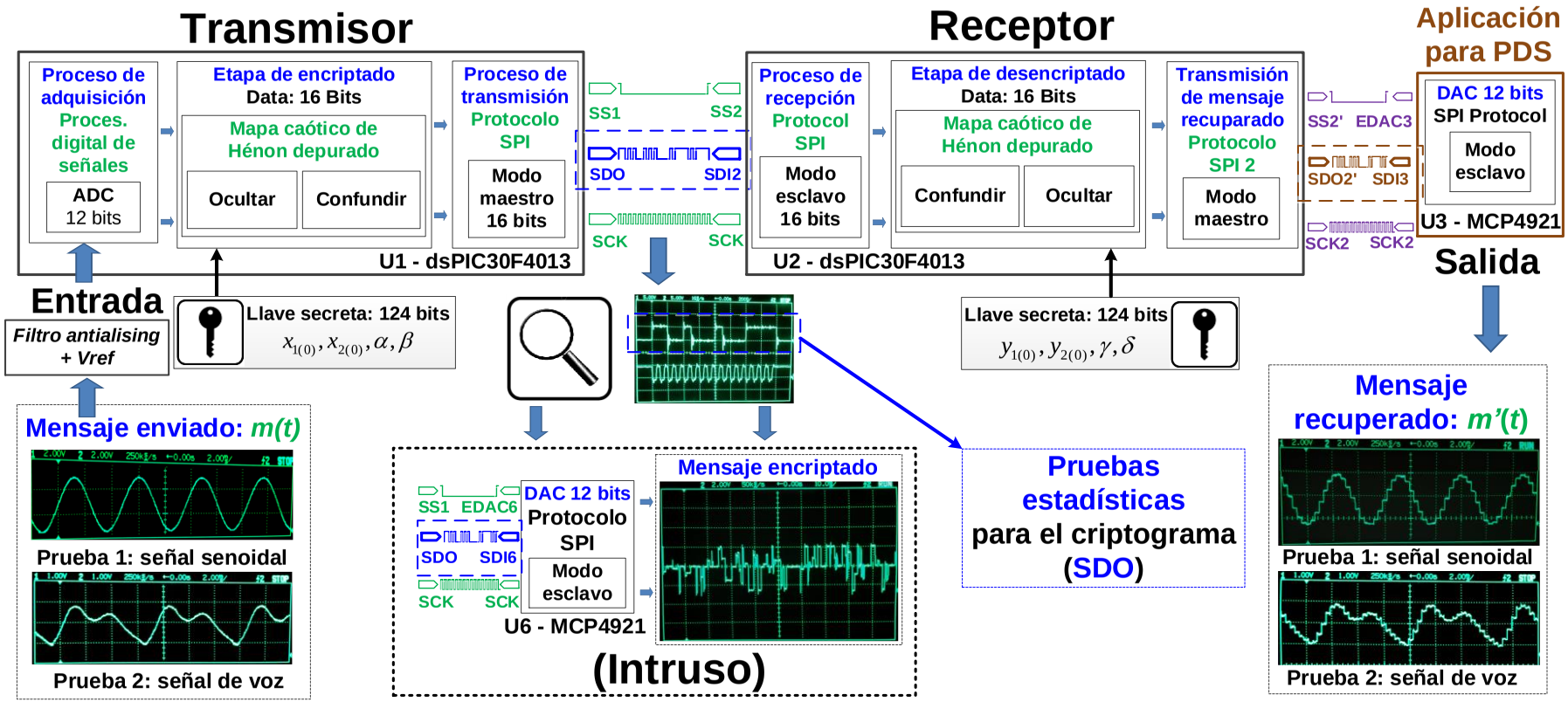


Figura 17. Diagrama de bloques del sistema criptográfico caótico embebido propuesto.

Tabla 2. Descripción de hardware del sistema criptográfico caótico embebido.

N° de Unidad	Descripción de Hardware	Comentario
U1	Maestro, microcontrolador dsPIC30F4013	Transmisor
U2	Esclavo 1, microcontrolador dsPIC30F4013	Receptor (tranceptor)
U3	Esclavo 2, DAC MCP4921	Representación de estado $x_{1(n)}$
U4	Esclavo 3, DAC MCP4921	Representación de estado $x_{2(n)}$
U5	Esclavo 4, DAC MCP4921	Representación de mensaje recuperado $m'(t)$
U6	Esclavo 5, DAC MCP4921	Representación del criptograma

4.2. Generación de secuencias seudoaleatorias

Con el fin de probar los resultados experimentales, la figura 18 muestra el diseño de un sistema embebido para reproducir electrónicamente un sistema caótico discreto de 2 dimensiones, trabajos similares están reportados en Méndez-Ramírez *et al.*, (2015; 2016; 2017). El protocolo de comunicación SPI del microcontroladores dsPIC U1 está configurado en modo maestro, los DACs U3 y U4 están en modo esclavo.

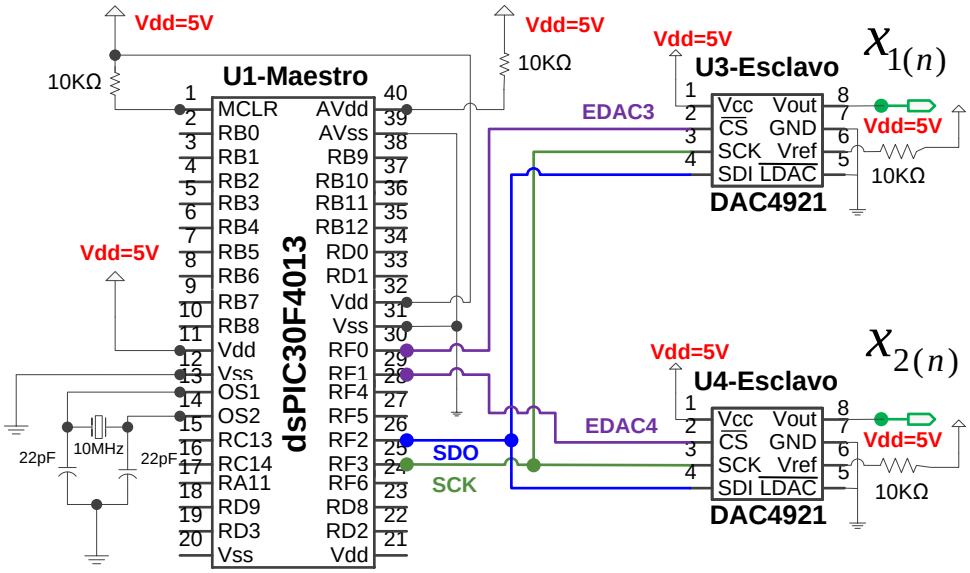


Figura 18. Circuito para representar los estados $x_{1(t)}$ y $x_{2(t)}$ del mapa caótico discreto.

4.2.1. Mapa caótico de Hénon depurado

Para generar secuencias seudoaleatorias consideramos el mapa caótico de Hénon descrito por (Hénon, 1976):

$$\begin{cases} x_{1(n+1)} = 1 - \alpha x_{1(n)}^2 + x_{2(n)}, \\ x_{2(n+1)} = \beta x_{2(n)}. \end{cases} \tag{20}$$

Se utilizan los parámetros $\alpha = 0.5$, $\beta = 0.3$ y las condiciones iniciales $x_{1(0)} = 0.5$ y $x_{2(0)} = 0.1$. El mapa de Hénon (20) se implementa en un sistema embebido y se procesa en el sistema embebido mostrado en la figura 18. El plano de fase de los estados $x_{1(n)}$ versus $x_{2(n)}$ se muestran en la figura 19, desafortunadamente la concentración de los datos y la dispersión del atractor extraño es homogénea.

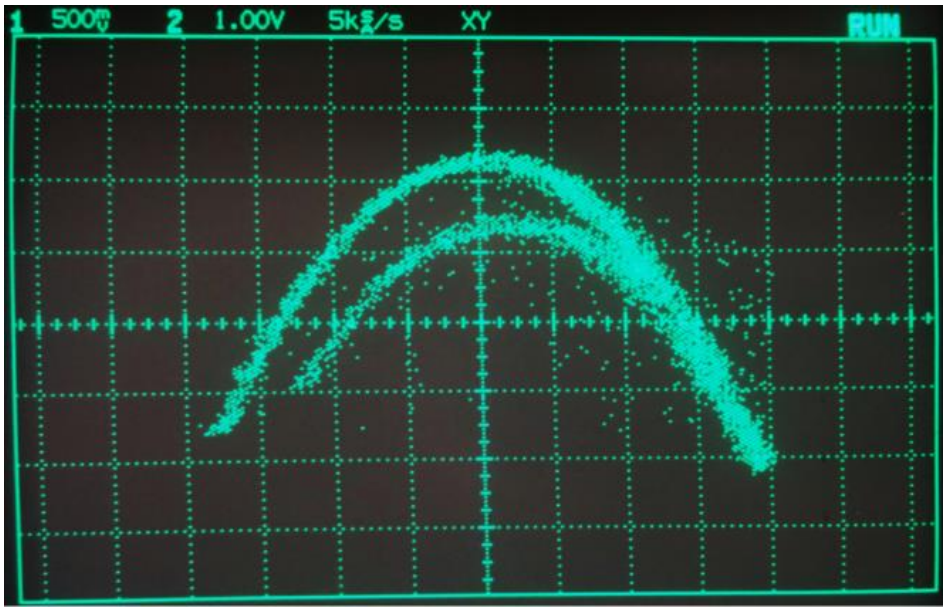


Figura 19. Representación del plano de fase $x_{2(n)}$ versus $x_{1(n)}$ del mapa caótico de Hénon (20).

Para mejorar la heterogeneidad en la concentración y dispersión de datos, se recurre a la construcción e implementación en un sistema embebido de un mapa de *Hénon depurado*, en el cual los valores de los estados $x_{1(n)}$ y $x_{2(n)}$ son convertidos a valores entre 0 y 1. Los primeros números enteros en los estados $x_{1(n)}$ y $x_{2(n)}$ son removidos para dejar solamente los números decimales en el mapa de Hénon (20), con lo cual, se obtiene el mapa de Hénon depurado (Méndez-Ramírez *et al.*, 2017),

$$\begin{cases} x_{1m(n)} = \left[\left(1.3 + x_{1(n)}\right)1000 - floor\left(\left(1.3 + x_{1(n)}\right)1000\right)\right], \\ x_{2m(n)} = \left[\left(0.5 + x_{2(n)}\right)10000 - floor\left(\left(0.5 + x_{2(n)}\right)10000\right)\right], \end{cases} \quad (21)$$

en el cual, $x_{1m(n)}$, $x_{2m(n)} \in [0, 1]$ son los estados del mapa caótico depurado, el subíndice *m* denota que el mapa caótico discreto estándar Hénon (SHM) (20) fue modificado, mapas caóticos depurados con características similares están reportados en (Murillo-Escobar *et al.*, 2015) y (Liu *et al.*, 2016). La figura 20 muestra el plano de fase $x_{1m(n)}$ versus $x_{2m(n)}$ con el atractor depurado implementado en U1, donde hay un gran cambio en la dispersión de los datos del mapa caótico de Hénon depurado (DHM) (21) con $n = 20000$ y en la figura 21 se muestran los histogramas de los mapas de Hénon original (20) y depurado (21), en las cuales se muestran las distribuciones de datos arrojados.

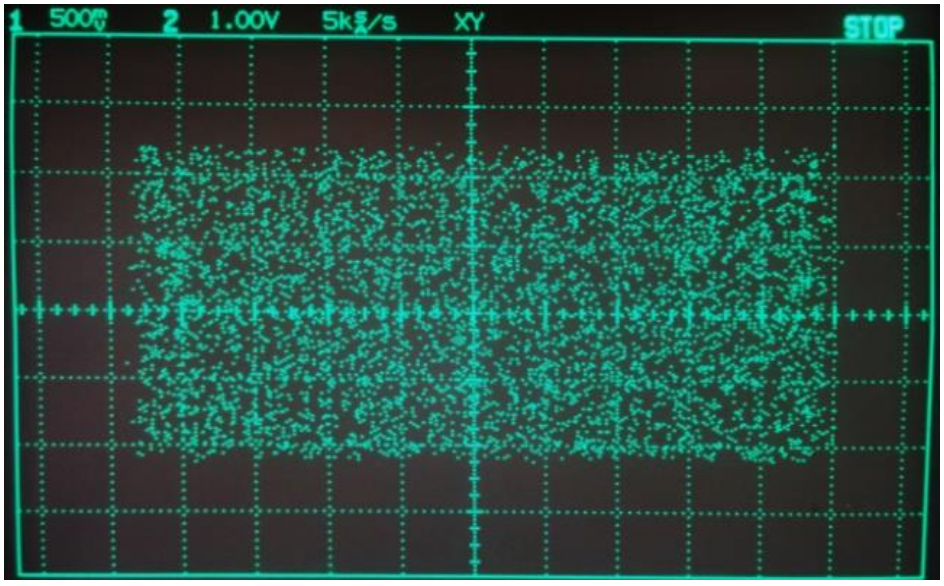
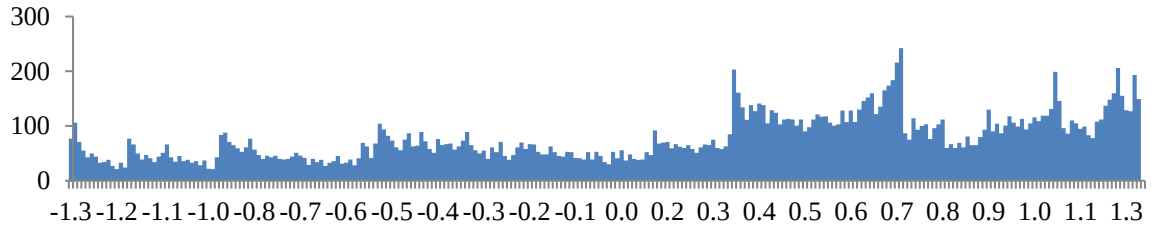
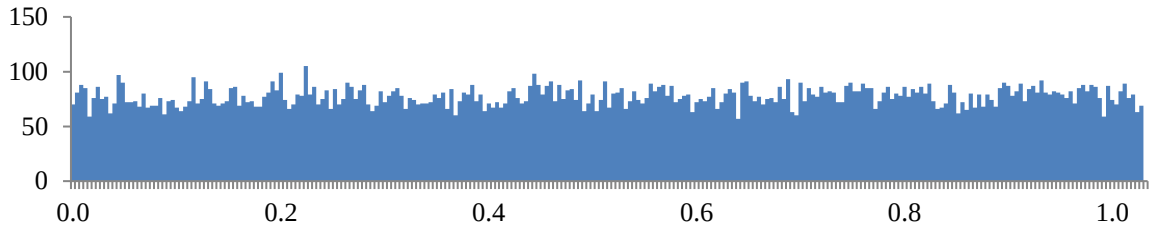


Figura 20. Representación del plano de fase $x_{2m(t)}$ versus $x_{1m(t)}$ mapa de Hénon depurado (21).



(a) Estado $x_{1(n)}$ del mapa estandar de Hénon (20).



(b) Estado $x_{1m(n)}$ del mapa depurado de Hénon (21).

Figura 21. Histogramas de los mapas de Hénon original (20) y depurado (21).

4.3. Definición de clave secreta

La clave secreta propuesta es simétrica y está definida por las condiciones iniciales y los parámetros utilizados en el del mapa estandar de Hénon (20) y del mapa depurado de Hénon (21). La llave debe estar segura; debe estar sujeta a un valor mayor que (Alvarez y Li, 2006), es decir

$$2^{100} = 1267650600228229401496703205376.$$

Utilizamos el estándar IEEE-754 con extensión AN575 a 32 bits para la representación numérica de los dsPICs, reportado por Microchip Technology Inc. (1997). La tabla 3 muestra la representación numérica de formatos de punto flotante con intervalos operativos que los microcontroladores dsPIC usan como estándar, en el cual eb es un exponente polarizado, e es el exponente o característico, f^* es la fracción o mantisa y $|A^*|$ es el número decimal.

Tabla 3. Estándar IEEE-754 Microchip 32 Bits.

	eb	***e***	$\|A^*\|f^*$	**Decimal**
MAX	0x00	128	7FFFFF	6.805646930E+38
MIN	0x01	-128	000000	1.17549435E-38

De acuerdo con los intervalos de la tabla 3, hay un máximo 0xFF7FFFFF y mínimo 0x01000000 denotado en base hexadecimal. El intervalo máximo representado en la base decimal es 4286578687 lo que equivale a restar 2^{32} menos 2^{23}, con resultado 2^{31}. Al realizar para los cálculos se obtiene representación de 4 números,

$$2^{31}2^{31}2^{31}2^{31} = 2^{124} > 2^{100}. \tag{22}$$

Se puede construir la clave secreta en 124 bits y su representación está sujeta a un espacio de llaves con notación hexadecimal,

Llave secreta 124 bits: FF7F FFFF FF7F FFFF FF7F FFFF FF7F FFFF.

La construcción del espacio de llaves se basa en la notación del estándar Microchip IEEE-754 para 32 bits, se utilizan los parámetros y las condiciones iniciales de SHM (20). La tabla 4 muestra la construcción de la clave secreta de 124 bits.

Tabla 4. Construcción de llave secreta.

Notación de la llave secreta	**Parámetros y c.i. del mapa de Hénon**			
	α	β	$x_{1(0)}$	$x_{2(0)}$
Flotante decimal	1.4	0.3	1.1121212	0.4654655
Flotante de 32 bits (IEEE)	3F B3 33 33	3E 99 99 9A	3F 8E 59 FD	3E EE 51 7E

La tabla 5 muestra el conjunto de tres llaves secretas usando los parámetros y condiciones iniciales diferentes del SHM (20). Se obtiene una clave secreta construida con una ligera variación de los parámetros y condiciones iniciales de SHM (20).

Tabla 5. Pruebas de sensibilidad de llaves secretas.

Llave	**Parámetros y c.i. del mapa de Hénon (20)**			
	α	β	$x_{1(0)}$	$x_{2(0)}$
K_1	1.4	0.3	1.1121213	0.4654655
	3F B3 33 33	3E 99 99 9A	3F 8E 59 FE	3E EE 51 7E
K_2	1.4	0.3	1.1121212	0.4654659
	3F B3 33 33	3E 99 99 9A	3F 8E 59 FD	3E EE 51 8E
K_3	1.4	0.298046886920929	1.1121212	0.4654655
	3F B3 33 33	3E 98 99 9A	3F 8E 59 FD	3E EE 51 7E

4.4. Proceso de transmisión

Esta sección describe el proceso de transmisión de mensajes confidenciales $m(t)$, el cual, se divide en tres etapas: etapa de adquisición, etapa de cifrado y transmisión de mensajes cifrados. La figura 22 ilustra la descripción del proceso de transmisión.

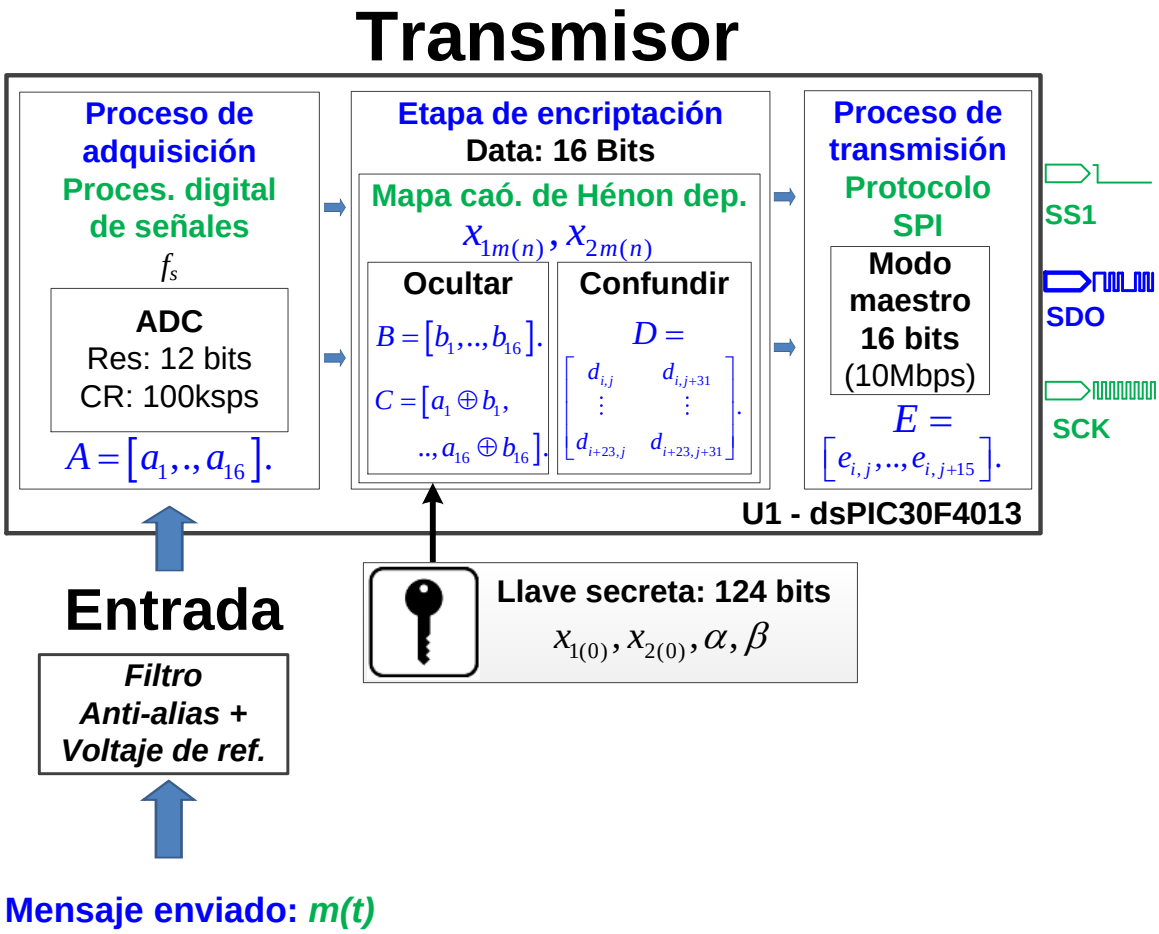


Figura 22. Diagrama de bloques del transmisor.

4.4.1 Proceso de adquisición

Las características técnicas del ADC interno de U1, permiten la adquisición de señales digitales o continuas dependiendo de su configuración (Microchip Technology Inc., 2004). Este dsPIC U1 tiene 13 canales ADC con una tasa de conversión (CR) de 100K muestras por segundo y una resolución de 12 bits para cada canal. El resultado de conversión del mensaje $m(t)$ está contenido en el registro ADCBUF0 de 16 bits, en el cual 12 bits que corresponden a los resultados ADC y los 4 bits restantes se utilizan como

registros de propósito general (RPG). Se descompone el vector $\boldsymbol{A}$ como equivalente del resultado obtenido en la conversión contenido en el registro ADCBUF0, donde cada elemento se expresa como a_k, con k = 1,2,..., 16,

$$\boldsymbol{A} = \left[a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, a_{14}, a_{15}, a_{16} \right], \tag{23}$$

donde $a_k \in [\ 0,\ 1\ ]$. Los elementos $a_{1\text{-}12}$ representan el resultado de conversión analógico usando el ADC de U1 y los elementos $a_{13\text{-}16}$ se consideran como los RPG que se pueden utilizar para la comunicación de canales, instrucciones de control, etc. La tabla 6 muestra la conversión de bits del registrador ADCBUF0 y la representación del vector $\boldsymbol{A}$, en el cual, $m(t)$ está contenido.

Tabla 6. Registros ADCBUF0 para representar los elementos del vector $\boldsymbol{A}$.

RPG bits 15-12	**Bits 11-0**
Elementos $a_{13\text{-}16}$	Los elementos de $a_{1\text{-}12}$ contienen el mensaje $m(t)$

Para la reconstrucción de mensajes, se utiliza el procesamiento digital de señales en el sistema embebido. Para ello, se utiliza el teorema de Nyquist-Shannon que establece que el muestreo uniforme de una señal, debe ser al menos dos veces (Nyquist, 1928; Shannon, 1949). El tamaño del mensaje $m(t)$ depende directamente de la capacidad que el algoritmo de U1 puede procesar y está relacionado directamente con la frecuencia máxima f_{max} y de la frecuencia de muestreo f_s que el algoritmo de U1 soporta. Estas frecuencias están relacionadas por la relación,

$$f_s \geq 2\, f_{max}. \tag{24}$$

Se requiere un filtro anti aliasing en las entradas de los canales analógicos de U1, en el cual, la frecuencia de corte de este filtro f_c se fija tal que

$$f_{max} < f_c < f_s. \tag{25}$$

4.4.2 Etapa de encriptación

El cifrado de mensajes $m(t)$ considera dos subprocesos, los métodos de difusión y confusión. Describimos estos métodos utilizando los estados del mapa de Hénon depurado (DHM) (21).

4.4.2.1. Método de difusión

El método difuso consiste en ocultar la información del mensaje $m(t)$ utilizando una operación lógica sobre los elementos del vector $\boldsymbol{A}$. Se utiliza el estado $x_{1m(n)}$ del mapa caótico depurado (DHM) (21) con un arreglo numérico para generar valores de 0-65535, el resultado es descrito por la expresión,

$$O_{ex(\mathrm{U1})} = x_{1m(n)}2^k = \left[\left(1.3 + x_{1(n)}\right)1000 - floor\left(\left(1.3 + x_{1(n)}\right)1000\right)\right]2^k, \tag{26}$$

También la expresión (26) se puede definir como vector $\boldsymbol{B}$ con 16 elementos equivalentes como sigue,

$$\boldsymbol{B} = \left[b_1, b_2, b_3, b_4, ..., b_{16}\right] \tag{27}$$

Se calcula la operación lógica OR-EX con cada elemento de los vectores $\boldsymbol{A}$ y $\boldsymbol{B}$. La información del vector $\boldsymbol{A}$ se oculta y los resultados se definen mediante el vector $\boldsymbol{C}$,

$$\boldsymbol{C} = \left[a_1 \oplus b_1, a_2 \oplus b_2, ..., a_{16} \oplus b_{16}\right] = \left[c_1, c_2, ..., c_{16}\right]. \tag{28}$$

4.4.2.2. Método de confusión

Este método consiste en construir una matriz con números seudoaleatorios extraídos de una base de datos de cadenas de ácido desoxirribonucleico (ADN). La construcción de la matriz de ADN se describe en el apéndice A. La matriz ($\boldsymbol{A1}$) muestra las secuencias de ADN con las coordenadas de las posiciones; esta matriz tiene dimensión $r \times s$, donde r son las filas y s son las columnas. Los elementos muestran posiciones desordenadas desde 1 al mayor valor de k; estos elementos desordenados son considerados una secuencia. Para cada fila r de la matriz ($\boldsymbol{A1}$), la secuencia de k elementos se repite dos veces. La matriz de secuencia de ADN descrita en el apéndice A de esta tesis.

Para este proceso se consideran los mismos elementos de la matriz ($\boldsymbol{A1}$) descritos en la matriz $\boldsymbol{D}$. Esta matriz consta de r = 24 secuencias de s = 32 elementos,

$$\boldsymbol{D} = \begin{bmatrix} d_{1,1} & d_{1,2} & \cdots & d_{1,s} \\ d_{2,1} & d_{2,2} & \cdots & d_{2,s} \\ \vdots & \vdots & \ddots & \vdots \\ d_{r,1} & d_{r,2} & \cdots & d_{r,s} \end{bmatrix}. \tag{29}$$

El algoritmo de confusión consiste en permutar las posiciones de los k = 16 elementos del vector $\boldsymbol{C}$. Calculamos una disposición numérica a partir de estados del mapa caótico depurado (21), estos estados definen las coordenadas iniciales para obtener de la matriz $\boldsymbol{D}$ (29) un nuevo vector con 16 nuevas posiciones. Las coordenadas iniciales se describen utilizando,

$$\begin{cases} i = x_{1m(n)} r = \left[\left(1.3 + x_{1(n)}\right)1000 - floor\left(\left(1.3 + x_{1(n)}\right)1000\right)\right] r, \\ j = x_{2m(n)} \dfrac{s}{2} = \left[\left(0.5 + x_{2(n)}\right)10000 - floor\left(\left(0.5 + x_{2(n)}\right)10000\right)\right] \dfrac{s}{2}, \end{cases} \tag{30}$$

donde i, j son enteros, los subíndices i = { $0 \leq x_{1m(n)} \leq 1$ / $i \in [\ 1, r\ ]$ } y j = { $0 \leq x_{2m(n)} \leq 1$ / $j \in [\ 1, s\ ]$ } definen las filas y columnas de la matriz $\boldsymbol{D}$, respectivamente. Las coordenadas iniciales de $d_{i,j}$ permiten obtener la posición inicial de una secuencia hasta completar 16 posiciones con el elemento $d_{i,j+15}$ de la matriz $\boldsymbol{D}$ propuesta,

$$\boldsymbol{D} = \begin{bmatrix} d_{i,j} & d_{i,j+1} & \cdots & d_{i,j+31} \\ d_{i+1,j} & d_{i+1,j+1} & \cdots & d_{i+1,j+31} \\ \vdots & \vdots & \ddots & \vdots \\ d_{i+23,j} & d_{i+23,j+1} & \cdots & d_{i+23,j+31} \end{bmatrix}. \tag{31}$$

Finalmente, el proceso de confusión se calcula utilizando (31) y el resultado es contenido en el nuevo vector fila $\boldsymbol{E}$. El primer elemento e_j del vector $\boldsymbol{E}$ contiene el primer elemento c_1 del vector $\boldsymbol{C}$, determinado por la primera posición del elemento $d_{i,j}$ del vector $\boldsymbol{D}$. El proceso se completa cuando el último elemento c_{16} corresponde al elemento e_{j+15} determinado por la posición $d_{i,j+15}$ de la matriz $\boldsymbol{D}$, esto es

$$\boldsymbol{E} = \left[c_1, c_2, ..., c_{16}\right] = \left[d_{i,j}, d_{i,j+1}, ..., d_{i,j+15}\right] = \left[e_j, e_{j+1}, ..., e_{j+15}\right]. \tag{32}$$

4.4.3. Transmisión de mensajes cifrados

El algoritmo de encriptación es implementado en U1. La información cifrada de $m(t)$ es contenida en el vector fila $\boldsymbol{E}$, donde los 16 elementos son equivalentes a una palabra de 16 bits, cuando esta palabra se transmite, corresponde a una iteración n. El protocolo SPI en U1 se configura en modo maestro y las terminales de U1 están conectados con U2 de acuerdo al protocolo: SS1 está conectado con SS2, SDO está conectado con SDI2 y SCK que es el reloj común para sincronizar los dsPICs. Finalmente, la palabra contenida en el vector $\boldsymbol{E}$ se transmite desde U1 a U2 iteración por iteración n.

4.5. Criptograma

En esta sección, se describen las pruebas estadísticas para probar la hipótesis que el mensaje $m(t)$ es seguro desde el punto de vista del observador. Para probar si el algoritmo propuesto reproduce secuencias PR, realizamos las pruebas estadísticas para el criptograma extraído de U1.

La figura 23 muestra el diagrama de bloques del sistema embebido, en el cual, un intruso interviene el protocolo de comunicación SPI utilizando el DAC externo U6. U1 se conecta con U6 utilizando las terminales: SS1 conectado con EDAC6, SDO conectado con SDI6 y SCK es el reloj común para sincronizar el dsPIC con el DAC externo. Se construye el mensaje cifrado o criptograma $m(t)$ y el intruso no puede descifrar los datos del mensaje original.

4.5.1. Pruebas estadísticas de seguridad

Se realizan las pruebas estadísticas para comprobar que el algoritmo de encriptación propuesto permite generar secuencias seudoaleatorias. Para esta prueba se considera una secuencia binaria encriptada con una longitud de n = 20000 bits generadas desde U1 donde la secuencia binaria es leída y almacenada por un computador. Las pruebas estadísticas constan de 5 pruebas numéricas y se realizan utilizando Matlab para verificar si el algoritmo encriptador genera secuencias seudoaleatorias lo suficientemente seguras (Menezes *et al.*, 1996; Fúster, 2004; Yalcin *et al.*, 2004). La tabla 7 muestra los resultados de las cinco pruebas numéricas. Estos resultados demuestran que el algoritmo de cifrado propuesto pasa con éxito todas las pruebas estadísticas y el algoritmo puede generar secuencias seudoaleatorias.

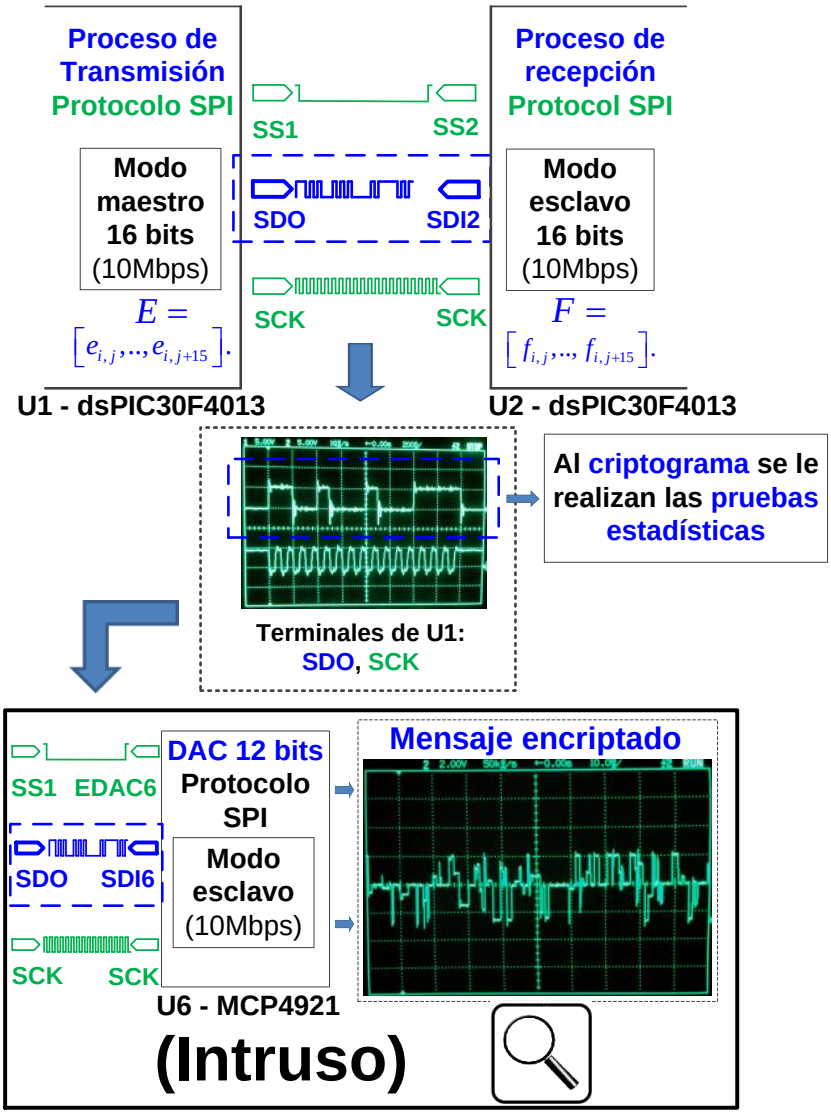


Figura 23. Canal de transmisión con mensaje encriptado contenido en la señal SDO (criptograma), señal de reloj SCK del protocolo SPI y señal obtenida por el intruso al utilizar DAC U6 sobre el criptograma.

Tabla 7. Resultados estadísticos.

Prueba numérica	Sistema propuesto	Valores requeridos	Parámetros
Frequency test X_1	1.6562	< 3.8415	-
Serial test X_2	3.0873	< 5.9915	-
Poker test X_3	10.87008	< 14.0671	$m = 3$
	17.2160	< 24.9958	$m = 4$
	40.5280	< 44.9853	$m = 5$
	71.750	< 82.5287	$m = 6$
Runs test X_4	6.1505	< 9.4877	$k = 3$
	6.9181	< 12.5916	$k = 4$
	8.7414	< 15.5073	$k = 5$
	11.9391	< 18.3070	$k = 6$
Autocorrelación test X_5	-0.3960	$-1.96 < X_5 < 1.96$	

4.6. Proceso de recepción

Esta sección describe el proceso de recepción para la reconstrucción de $m'(t)$. El proceso de recepción es el mismo que el proceso de transmisión reportado en sección 4.4.2 y los procesos de difusión y confusión son reproducidos en etapas invertidas. En la figura 24 se muestran los procesos divididos en 3 etapas: recepción del mensaje cifrado, proceso de descifrado y la retransmisión externa del mensaje recuperado utilizando un DAC externo. El algoritmo de desencriptación contenido en U2 se configura para comprobar constantemente si la palabra de 16 bits de U1 fue recibida exitosamente. Para ello, se considera la configuración del módulo SPI en U2. La bandera SPIRBF del registro interno SPI1STAT, permite verificar si la recepción de una palabra con 16 bits de U1 fue recibida exitosamente. Finalmente, el vector $\boldsymbol{F}$ contiene la recepción del mensaje encriptado,

$$\boldsymbol{F} = \left[f_j, f_{j+2}, \ldots, f_{j+15} \right]. \tag{33}$$

4.6.1. Etapa de descifrado

Se utilizan los mismos métodos con las mismas características propuestas en la etapa de encriptación de la sección 4.2.2, pero en orden inverso. La clave secreta es simétrica, por lo tanto es obligatorio que el algoritmo U2 contenga la misma condición inicial y los mismos valores de parámetros del mapa caótico depurado (21). Estas subsecciones se describen a continuación.

4.6.1.1. Método de confusión inversa

Para generar secuencias seudoaleatorias se usan los mapas de Hénon estándar (20) y el mapa de Hénon depurado (21), pero la representación de variables de estados se cambió para diferenciar las etapas de transmisión y recepción. Se definen las variables de estado $y_{1(n)}$, $y_{2(n)}$ y los parámetros γ, δ del mapa de Hénon utilizado en esta etapa como sigue,

$$\begin{cases} y_{1(n+1)} = 1 - \gamma y_{1(n)}^2 + y_{2(n)}, \\ y_{2(n+1)} = \delta y_{1(n)}. \end{cases} \tag{34}$$

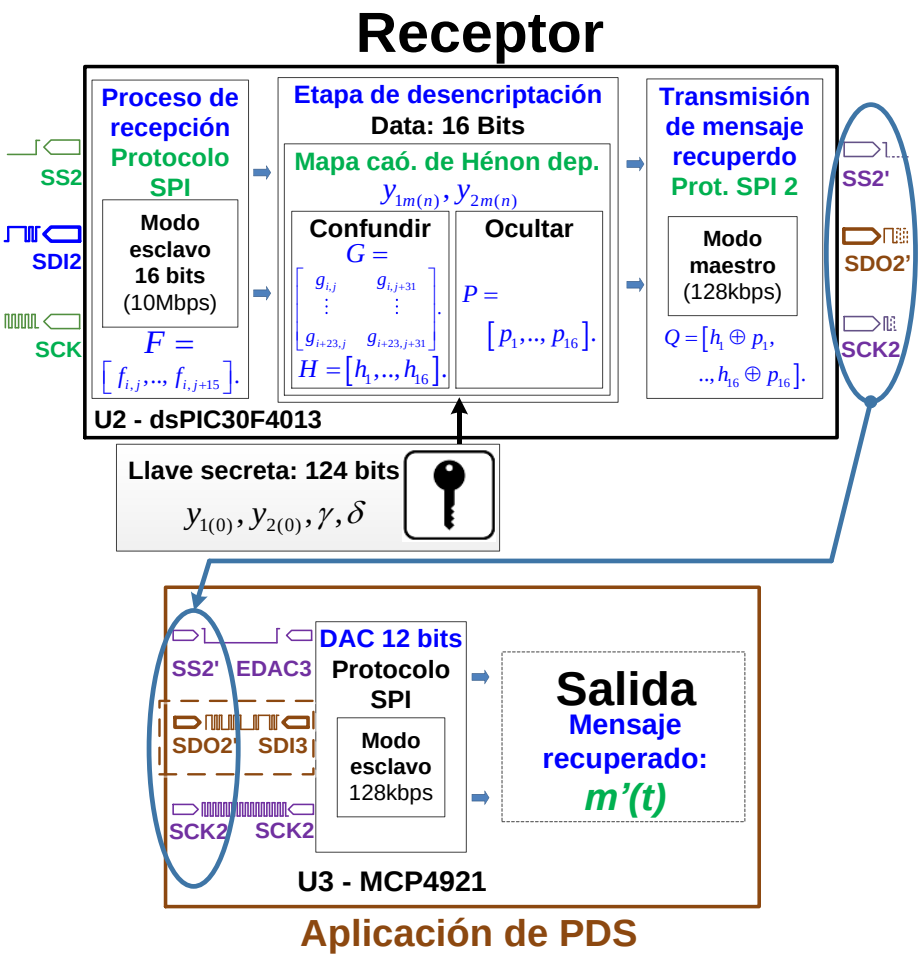


Figura 24. Diagrama de bloques del receptor con mensaje recuperado por U3 utilizando el protocolo SPI.

Los estados depurados se representan por $y_{1m(n)}$ y $y_{2m(n)}$. Las coordenadas iniciales se describen como sigue,

$$\begin{cases} l = y_{1m(n)} v = \left[\left(1.3 + y_{1(n)}\right)1000 - floor\left(\left(1.3 + y_{1(n)}\right)1000\right)\right] v, \\ u = y_{2m(n)} \dfrac{w}{2} = \left[\left(0.5 + y_{2(n)}\right)10000 - floor\left(\left(0.5 + y_{2(n)}\right)10000\right)\right] \dfrac{w}{2}, \end{cases} \tag{35}$$

donde l, u son números enteros, los subíndices $l = \{ 0 \le y_{1m(n)} \le 1 \,/\, l \in [\,1, v\,] \}$ y $u = \{ 0 \le y_{2m(n)} \le 1 \,/\, u \in [\,1, w\,] \}$ definen las filas y columnas respectivamente para determinar las posiciones de los elementos para la nueva matriz $\boldsymbol{G}$ de ADN. Esta matriz contiene los mismos elementos de la matriz $\boldsymbol{D}$ (31). Las coordenadas iniciales $g_{l,u}$ permiten obtener la posición inicial de una secuencia hasta completar 16 posiciones y la última posición denominada $g_{l,u+15}$,

$$\boldsymbol{G}=\begin{bmatrix} g_{l,u} & g_{l,u+1} & \cdots & g_{l,u+31} \\ g_{l+\mathbf{1},u} & g_{l+1,u+1} & \cdots & g_{l+1,u+31} \\ \vdots & \vdots & \ddots & \vdots \\ g_{l+23,u} & g_{i+23,\,j+1} & \cdots & g_{l+23,u+31} \end{bmatrix}. \tag{36}$$

El vector $\boldsymbol{F}$ contiene el mensaje $m(t)$ encriptado, el proceso de confusión fue calculado usando (34)-(36). Se obtiene el nuevo vector fila $\boldsymbol{H}$. El primer elemento h_u del vector $\boldsymbol{H}$ contiene el primer elemento f_j del vector $\boldsymbol{F}$ y el último elemento f_{j+15} es determinado por h_{u+15},

$$\boldsymbol{H}=\left[\, f_j, f_{j+2},..., f_{j+15}\right]=\left[g_{l,u}, g_{l,u+1},..., g_{l,u+15}\right]=\left[h_u, h_{u+1},..., h_{u+15}\right]. \tag{37}$$

4.6.1.2. Método de difusión inversa

Se utiliza el estado $y_{1m(n)}$ del mapa de Hénon depurado (35) y se calculó un arsenal numérico para generar valores de 0-65535 y el resultado fue definido por,

$$O_{ex(\mathrm{U1})}=y_{1m(n)}2^k=\left[\left(1.3+y_{1(n)}\right)1000-floor\left(\left(1.3+y_{1(n)}\right)1000\right)\right]2^k, \tag{38}$$

El vector $\boldsymbol{P}$ define los 16 elementos equivalentes,

$$\boldsymbol{P}=\left[\, p_1, p_2,..., p_{16}\right]. \tag{39}$$

La operación lógica OR-EX se realizó sobre cada elemento del vector $\boldsymbol{H}$ con respecto al vector $\boldsymbol{P}$. Finalmente, se construye el mensaje descifrado $m'(t)$ y la información está contenida en el vector $\boldsymbol{Q}$,

$$\boldsymbol{Q}=\left[h_1\oplus p_1, h_2\oplus p_2,..., h_{16}\oplus p_{16}\right]=\left[q_1, q_2,..., q_{16}\right]. \tag{40}$$

4.6.2. Transmisión del mensaje recuperado

Una vez que se recuperó el mensaje descifrado en el receptor, configuramos el segundo protocolo SPI usando U2 para transmitir $m'(t)$ para un DAC externo U6. El hardware de dsPIC U2 tiene sólo un puerto SPI que permite enlazar U1 y U2 en modo esclavo. Es necesario habilitar otro puerto SPI en modo maestro para vincular U2 con U5. La biblioteca Soft SPI, de Mikroc Pro para el compilador dsPIC, permite habilitar otro puerto SPI mediante la configuración de software y hardware. Definamos la conexión de pines usando el hardware de U2 a U5, donde SS2' estaba conectado con EDAC3, SDO2 estaba conectado con SDI3 y SCK2 es el reloj común para sincronizar los dsPICs con el DAC. Ahora, el mensaje descifrado $m'(t)$ se puede transmitir de U2 a U5 utilizando el nuevo puerto SPI. La figura 14 muestra el segundo protocolo SPI de U2 conectado con el DAC externo U5.

4.7. Resultados experimentales

En esta sección se presenta los resultados experimentales para transmitir y recuperar mensajes confidenciales de U1 a U2, el mensaje se envía a través del criptograma. Se desarrollaron tres tipos de pruebas para obtener experimentalmente el rendimiento del sistema embebido propuesto: sensibilidad de las claves secretas, señal analógica reproducida por un generador de funciones y grabación de voz. La figura 25 muestra el diagrama esquemático del sistema embebido propuesto.

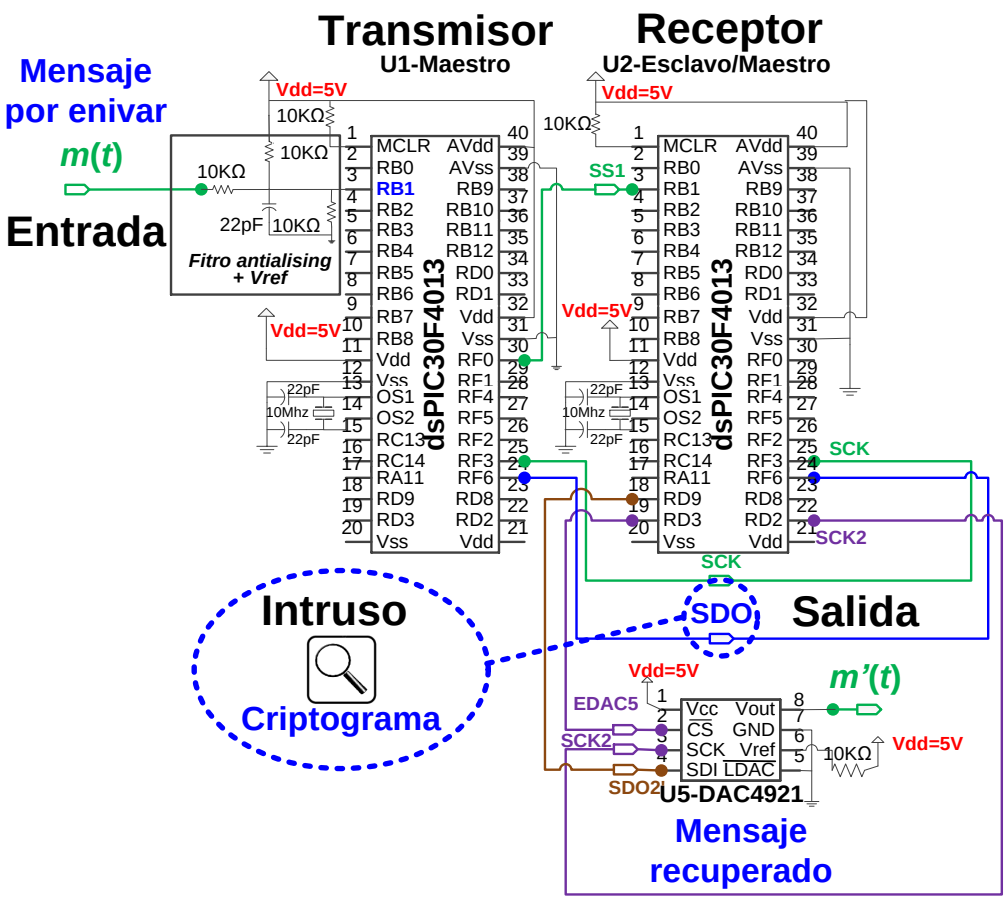


Figura 25. Diagrama esquemático del sistema embebido completo.

4.7.1. Complejidad temporal de los algoritmos

El tiempo de complejidad (TC) se considera como la estimación contando el número de operaciones elementales realizadas para los algoritmos de U1, donde una operación elemental toma un tiempo fijo para realizar el tiempo de ejecución de una iteración n (Sipser, 2006). Para obtener el rendimiento del sistema embebido propuesto, se calcula el tiempo de complejidad que genera el algoritmo U1 para determinar la cantidad de iteraciones n generadas en 1 segundo. El tiempo de complejidad de U1 se calcula con el período de tiempo $T_{Q(U1)}$, que es recíproco de la frecuencia definida como $f_{Q(U1)}$,

$$f_{Q(U1)} = \frac{1}{T_{Q(U1)}}. \tag{41}$$

El algoritmo de cifrado es implementado en U1. Utilizando la expresión (41) se calcula el tiempo de complejidad de U1. La tabla 8 muestra los resultados experimentales para el algoritmo de U1.

Tabla 8. Cálculos de tiempo y frecuencia de operación del algoritmo de transmisión sobre el dsPIC U1.

Proceso	$T_{Q(U1)}$**(µs)**	$f_{Q(U1)}$**(Hz)**	**Representación matemática**
Proceso de adquisición	10	100000	$\boldsymbol{A}$
Etapa de cifrado	335	3205	$\boldsymbol{B \rightarrow C \rightarrow D}$
Transmisión de mensajes cifrados	5	200000	$\boldsymbol{E}$
Total	**350**	**2857**	**-**

De acuerdo a la tabla 8, el algoritmo de U1 tiene un tiempo de complejidad de 350 µs y $f_{Q(U1)}$ = 2857 Hz. Para nuestro estudio, consideramos $f_{Q(U1)}$ como n = 2857 iteraciones por segundo. A partir de la capacidad de procesamiento de U1, se utiliza la expresión (24) para determinar f_s = 2857 Hz y f_{max} = 1429 Hz. Esto significa que U1 puede muestrear la estructura del mensaje $m(t)$ hasta f_{max} = 1429 Hz iteración por iteración n.

La tabla 9 muestra los resultados experimentales para el algoritmo de U2.

Tabla 9. Cálculos de tiempo y frecuencia de operación del algoritmo de recepción sobre el dsPIC U2.

Proceso	$T_{Q(U2)}$**(µs)**	**Representación matemática**
Recepción del mensaje encriptado	5	$\boldsymbol{F}$
Etapa de Desencriptación	160	$\boldsymbol{G \rightarrow H \rightarrow P}$
Transmisión del mensaje recuperado	125	$\boldsymbol{Q}$
Total	**290**	**-**

Para reconstrucción del mensaje $m'(t)$ en el receptor, es deseable que el tiempo de complejidad del algoritmo de U2 sea menor que el algoritmo de U1; para ello, se configura el PPL interno de U2 y se acelera la capacidad de procesamiento de U2, esto permite un procesamiento más rápido en el algoritmo de desencriptación de U2. Por lo tanto, el TC de U2 debe ser menor que U1, expresado como sigue,

$$T_{Q(U1)} > T_{Q(U2)}. \tag{42}$$

De acuerdo con los resultados obtenidos en las tablas 8 y 9, el tiempo de complejidad de $T_{Q(U1)}$ es mayor que $T_{Q(U2)}$, entonces se cumple la condición (43) para que U1 y U2 sean sincronizados. De acuerdo

con la capacidad de procesamiento de U1, el sistema embebido propuesto soporta mensajes de frecuencia máxima hasta 1429 Hz.

4.7.2. Resultados experimentales sobre el sistema embebido

El rendimiento de los algoritmos de encriptación y desencriptación son obtenidos a partir del desempeño del sistema embebido. La capacidad del procesamiento del mensaje $m(t)$, $f_{Q(U1)}$, f_s y f_{ma} son determinados a partir del tiempo de complejidad de U1. El hardware para la adquisición del mensaje $m(t)$ se mostró en la figura 25. La terminal RB1 es configurada como entrada del canal ADC en U1 y las referencias de tensión del canal ADC interno de U1 se fijan por los voltajes AVss = 0 V y AVdd = 5 V. El divisor de voltaje externo se fija con el voltaje de referencia Vref = Vdd/2 = 2,5 V. Adicionalmente, se fija un filtro de anti-alias con f_c = 2K Hz de acuerdo con la condición (25).

Se proponen dos tipos de mensajes para llevar a cabo tres pruebas experimentales sobre el sistema embebido. El primer mensaje se denomina $m_1(t)$ y se utiliza un generador de funciones para reproducir una señal sinusoidal mediante la siguiente expresión,

$$m_1(t) = 2\,\mathrm{sen}(2\pi ft + \theta) + V_{ref}, \tag{43}$$

en la cual, f es la frecuencia delimitada por la condición $f \leq f_s$, y θ es el ángulo de fase.

El segundo mensaje utilizado es una señal de voz representada por $m_2(t)$ y se utilizan dispositivos de audio externos sobre el sistema embebido para realizar pruebas de sonido profesional. La recepción de los mensajes $m_1(t)$ y $m_2(t)$ se definen como $m_1'(t)$ y $m_2'(t)$, respectivamente.

4.7.2.1. Primera prueba: sensibilidad de llaves secretas

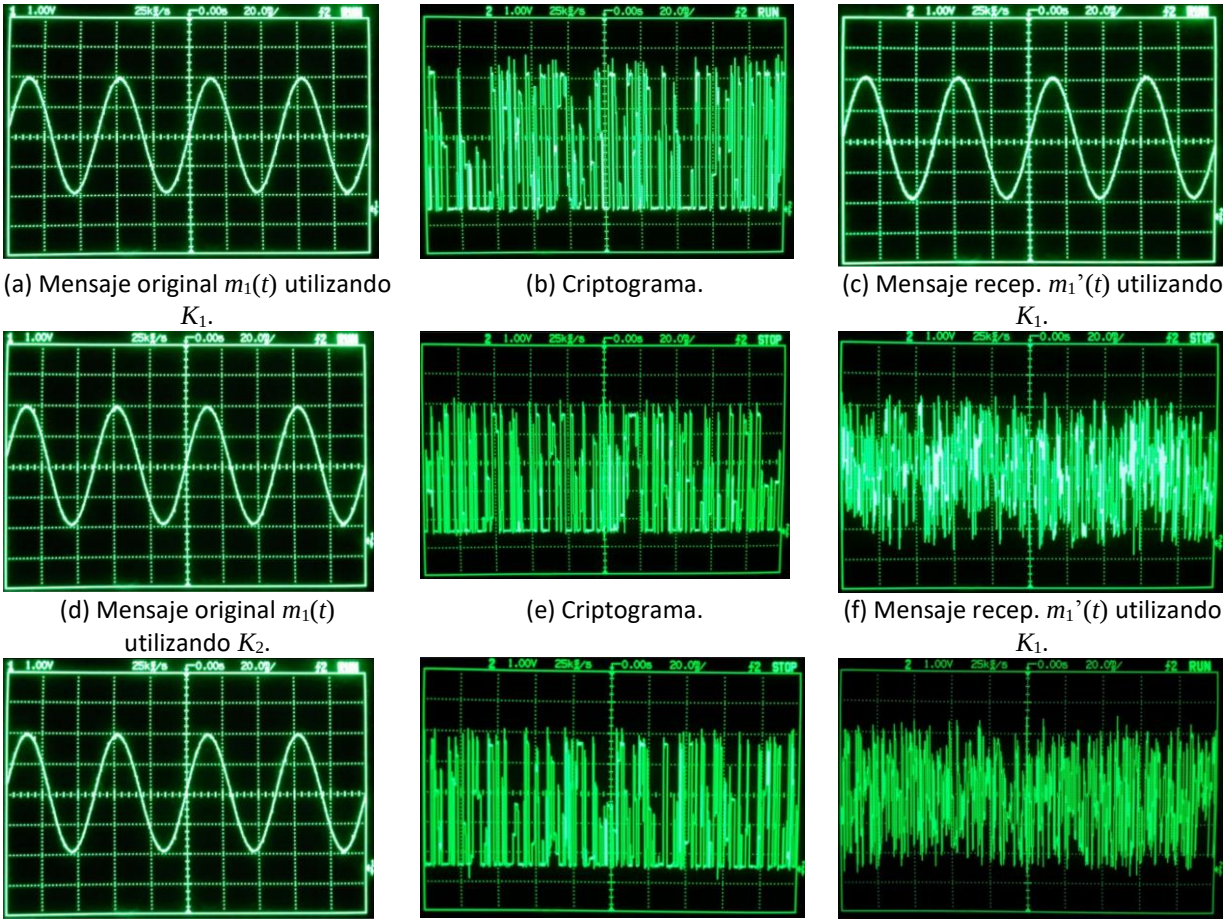
La prueba de sensibilidad de llaves secretas se llevó a cabo utilizando las llaves secretas K_1, K_2 y K_3 indicadas en la tabla 5. Las llaves secretas se ingresan programando directamente los algoritmos de U1 y U2, pero las llaves secretas se pueden registrar usando un teclado externo conectado a los dsPICs U1 y U2. En la tabla 10 se muestra la propuesta de tres combinaciones de las llaves secretas K_1, K_2 y K_3 para llevar

a cabo la prueba de sensibilidad. Para esta prueba, se utiliza la expresión (43) para reproducir el mensaje $m_1(t)$ considerando una f = 20 Hz sobre el sistema embebido.

Tabla 10: Prueba de sensibilidad de llaves secretas sobre el sistema embebido.

Caso	Llave secreta en U1	Llave secreta en U2
1	K_1	K_1
2	K_2	K_1
3	K_3	K_1

La figura 26 muestra el resultado de la prueba de sensibilidad entre las tres claves secretas. Concluimos que la definición de clave secreta apoya la hipótesis de que su construcción de 124 bits es segura porque para ligeras variaciones de las condiciones iniciales y parámetros de claves secretas, el resultado del mensaje recuperado $m_1{}'(t)$ fue diferente al mensaje original $m_1(t)$ para diferentes claves secretas, resultados similares están reportados en Liu *et al.*, (2016).



(a) Mensaje original $m_1(t)$ utilizando K_1. (b) Criptograma. (c) Mensaje recep. $m_1{}'(t)$ utilizando K_1.

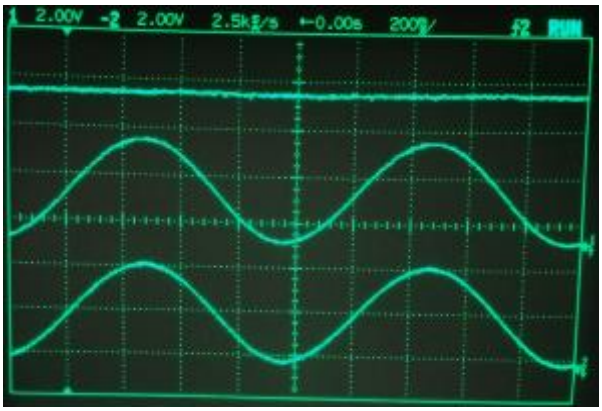
(d) Mensaje original $m_1(t)$ utilizando K_2. (e) Criptograma. (f) Mensaje recep. $m_1{}'(t)$ utilizando K_1.

(g) Mensaje original $m_1(t)$ utilizando K_3. (h) Criptograma. (i) Mensaje recep. $m_1{}'(t)$ utilizando K_1.

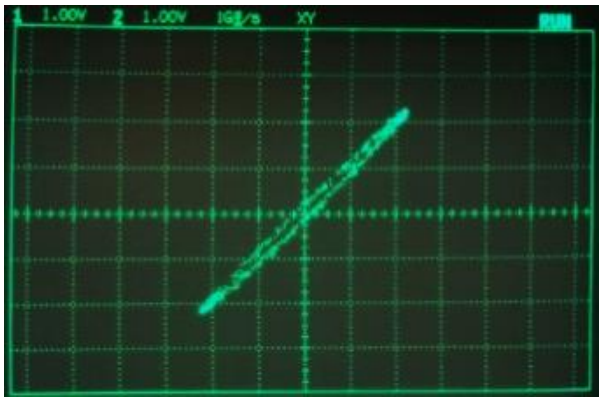
Figura 26: Resultado de la prueba de sensibilidad de las llaves sobre el sistema embebido.

4.7.2.2. Segunda prueba: señal $m_1(t)$ como mensaje

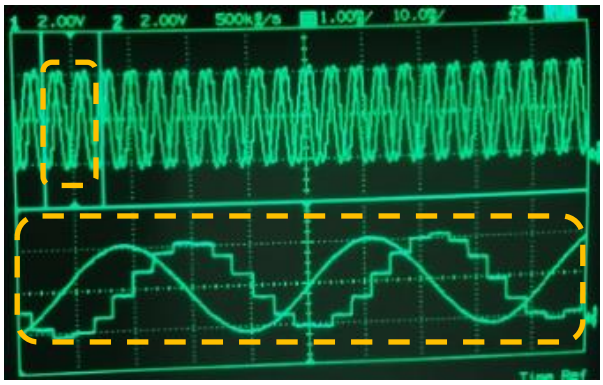
Utilizamos la misma señal sinusoidal descrita por (43) para reproducir $m_1(t)$ para diferentes valores de f. Para esta prueba, se consideró la misma clave secreta K_1 para U1 y U2 como se muestra en la tabla 5. La figura 16 muestra los resultados y la descripción de los mensajes enviados $m_1(t)$ y mensajes recuperados $m_1'(t)$ en el sistema embebido. La figura 27a muestra 3 señales: señal de error entre los mensajes $m_1(t)$ y $m_1'(t)$, el error referido como $e_{m1}(t)$ que es equivalente $m_1(t)$ - $m_1'(t)$, para f = 10 Hz. La figura 27b muestra el plano de fase de $m_1(t)$ versus $m_1'(t)$ para f = 10 Hz. La figura 27c muestra la diferencia de fase de $m_1(t)$ con $m_1'(t)$, donde la línea naranja segmentada representa una ventana aumentada 10 veces de señales de $m_1(t)$ y $m_1'(t)$ utilizando f = 210 Hz. La figura 27d muestra el plano de fase de $m_1(t)$ versus $m_1'(t)$, donde θ se incrementó a $\pi/2$ para cada f = 210 Hz, con $\theta = \pi/2$ como diferencia de fase.



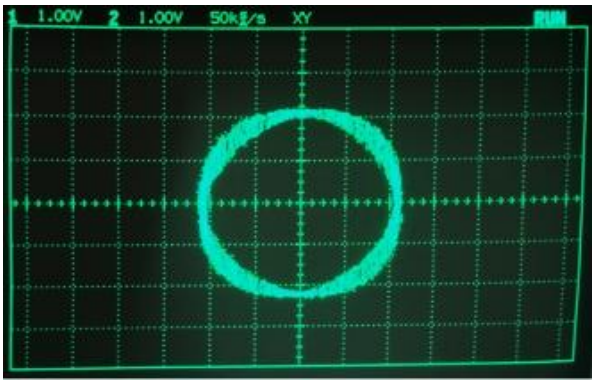
(a) Señal seno con error de mensaje $e_m(t)$, mensaje $m(t)$ y mensaje recuperado $m'(t)$.



(b) Plano de fase $m(t)$ versus $m'(t)$ sin desfase.



(c) Señal f = 210 Hz.

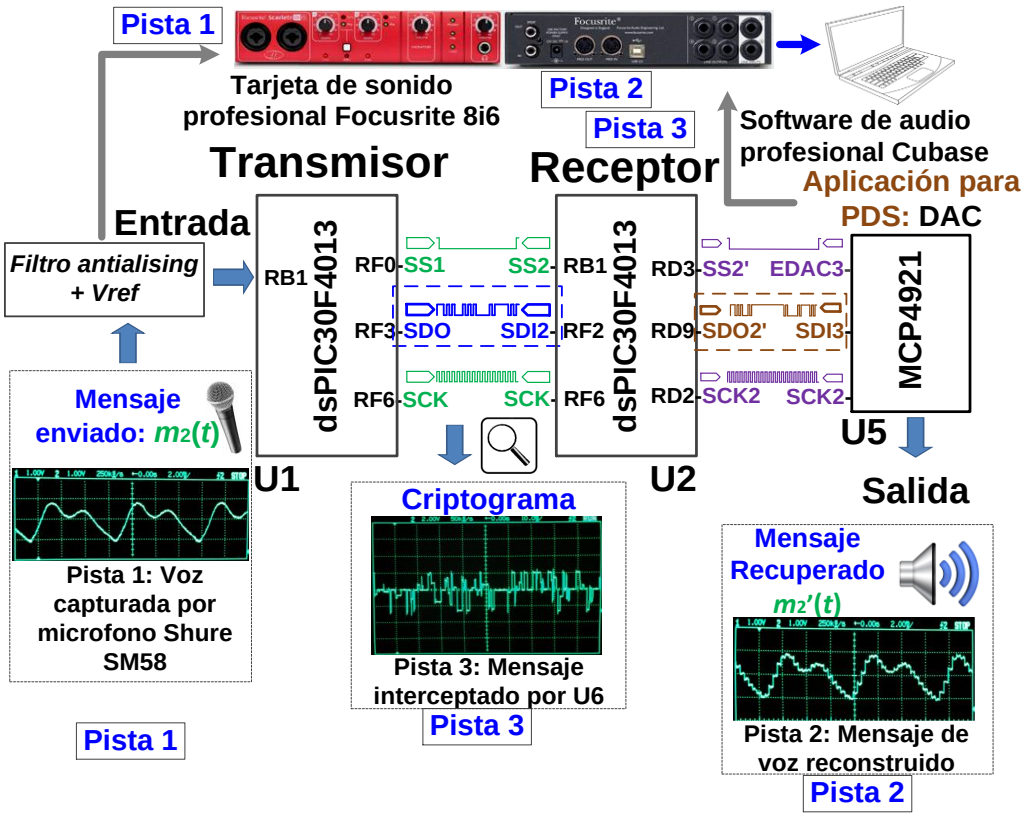


(d) Plano de fase $m(t)$ versus $m'(t)$ con desfase 90°.

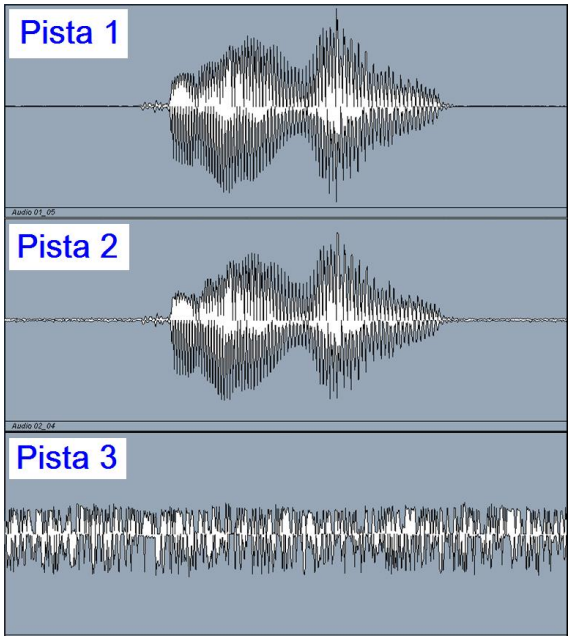
Figure 27: Pruebas de procesamiento digital con mensajes con señal de entrada seno $m(t)$ y salida $m'(t)$.

4.7.2.3. Tercera prueba: mensaje de voz $m_2(t)$

La figura 17 muestra el sistema embebido con dispositivos externos utilizados para grabar voz. Para esta prueba, se consideró la misma clave K_1 para U1 y U2. El hardware utilizado fue: el micrófono Shure SM58, una tarjeta de sonido externa Focusrite 8i6 y el procesador Notebook i5 con software profesional de audio Cubase, que se muestran en la Fig. 17a. El mensaje de voz transmitido fue "*hola mundo*". La figura 17b muestra las tres señales grabadas usando el software Cubase que se representan como pistas de audio: $m_2(t)$ mensaje de voz enviado (pista 1), $m_2'(t)$ mensaje de voz recuperado (pista 2) y el criptograma adquirido utilizando U6 (pista 3). Finalmente, se recuperó el mensaje de voz $m_2'(t)$; Este mensaje tiene menos fidelidad de audio porque la capacidad de procesamiento de voz de audio fue delimitada por ancho de banda y frecuencia de muestreo f_s = 1429 Hz.



(a) Sistema embebido con equipamiento de audio profesional.



(b) Grabación sobre software Cubase 5, mensaje original $m_2(t)$ en pista 1, mensaje recuperado $m_2\,'(t)$ en pista 2 y criptograma en pista 3.

Figura 28. Montaje de audio profesional para la grabación del mensaje de audio: "*hola mundo*".

4.8. Conclusiones del capítulo

En este capítulo hemos presentado el diseño e implementación de un algoritmo de encriptación caótico mediante el uso del protocolo de comunicación SPI de 16 bits en los microcontroladores dsPICs para adquirir, procesar, cifrar, transmitir, sincronizar, recibir, descifrar y volver a transmitir un mensaje utilizando un DAC junto con la teoría del procesamiento digital de señales. Se realizaron tres tipos de pruebas para verificar el rendimiento del mensaje en el sistema embebido: clave secreta de sensibilidad, señal analógica generada a partir de un generador de funciones y grabación de voz. El criptograma fue sometido a una prueba estadística del algoritmo propuesto. El criptograma tiene un buen rendimiento para generar secuencias de seudoaleatorias seguras. Este algoritmo se puede utilizar en un circuito integrado que disponga del protocolo SPI estándar.

Los resultados numéricos demostraron mediante el uso de Mikroc Pro para dsPIC compilador que incluye las propiedades de estándar IEEE-754, AN575 de 32 bits en el dsPICs. La calidad del mensaje depende de la capacidad del proceso que tienen los CIs. El ES es fácil de montar en protoboard y tiene una implementación de bajo costo porque el sistema cuesta menos de 30 USD.

El material de este capítulo se recoge del artículo publicado en (Méndez-Ramírez *et al.*, 2017).

Capítulo 5. Algoritmos numéricos para discretización de sistemas dinámicos

En este capítulo, se presentan los algoritmos numéricos de Euler, Heun y RK4 que se utilizarán para representar la versión discretazada de 5 sistemas caóticos en 3 dimensiones propuestos en esta tesis. Se presenta un estudio utilizando una variable de estado del sistema de Lorenz en su versión discretizada y se calcula el error cuadrático medio para determinar efectivamente la exactitud de los algoritmos numéricos considerados.

5.1. Algoritmos numéricos de discretización

Un algoritmo es un conjunto finito de instrucciones o pasos que sirven para ejecutar una tarea y/o resolver un problema. De un modo más formal, un *algoritmo* es una secuencia finita de operaciones realizables, no ambiguas, cuya ejecución proporciona una solución de un problema en tiempo finito. Un *algoritmo numérico* es el conjunto de instrucciones ordenadas para resolver un problema que involucra procesos matemáticos representados en seudocódigos o en diagramas de flujo (Pinales y Velázquez, 2014).

Sin pérdida de generalidad, en este trabajo de tesis, consideramos sistemas dinámicos en 3 dimensiones, denotando el vector de estados por $(x, y, z) \in \mathbb{R}^3$, de esta manera las ecuaciones de estado para estos sistemas, se describe como sigue:

$$\begin{cases} \dot{x} = f(x, y, z), \\ \dot{y} = g(x, y, z), \\ \dot{z} = h(x, y, z), \end{cases} \tag{44}$$

considerando el valor inicial $(x_{(0)}, y_{(0)}, z_{(0)}) = 0$.

5.1.1. Algoritmo de Euler

Euler es el primer algoritmo numérico empleado, es sabido que al discretizar el sistema continuo (44) el método de Euler se deriva de la expansión en serie de Taylor del sistema, en el cual, el término cuadrático y de orden superior no son considerados. Este algoritmo presenta sólo un paso, es fácil de implementar porque emplea menos operaciones matemáticas (Yang *et al*., 2005). Para una mejor comprensión del estudio de los sistemas caóticos en 3 dimensiones, el algoritmo numérico de Euler para el sistema (44) esta descrito por:

$$\begin{cases} x_{n+1} = x_n + \tau f(x_n, y_n, z_n), \\ y_{n+1} = y_n + \tau g(x_n, y_n, z_n), \\ z_{n+1} = z_n + \tau h(x_n, y_n, z_n). \end{cases} \tag{45}$$

5.1.2. Algoritmo de Heun

El segundo algoritmo implementado es el de *Heun* (Yang *et al*., 2005). Este método se conoce como trapezoidal de dos pasos, es un método de predicción y corrección, en el cual, el primer paso predice, y el segundo paso corrige, el algoritmo numérico de Heun es descrito para el sistema (44) esta descrito por,

$$\begin{cases} x^*_{n+1} = x_n + \tau f\left(x_n, y_n, z_n\right), \\ y^*_{n+1} = y_n + \tau g\left(x_n, y_n, z_n\right), \\ z^*_{n+1} = z_n + \tau h\left(x_n, y_n, z_n\right), \\ x_{n+1} = x_n + \dfrac{\tau}{2}\left(f\left(x_n, y_n, z_n\right) + x^*_{n+1}\right), \\ y_{n+1} = y_n + \dfrac{\tau}{2}\left(g\left(x_n, y_n, z_n\right) + y^*_{n+1}\right), \\ z_{n+1} = z_n + \dfrac{\tau}{2}\left(h\left(x_n, y_n, z_n\right) + z^*_{n+1}\right). \end{cases} \tag{46}$$

5.1.3. Algoritmo de RK4

El tercer método de discretización utilizado es el RK4, este método es uno de los más utilizados para resolver las ecuaciones diferenciales dada su precisión (Yang *et al*., 2005), y es descrito por el siguiente sistema:

$$\begin{cases} x_{n+1} = x_n + \dfrac{\tau}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right), \\ y_{n+1} = y_n + \dfrac{\tau}{6}\left(l_1 + 2l_2 + 2l_3 + l_4\right), \\ z_{n+1} = z_n + \dfrac{\tau}{6}\left(m_1 + 2m_2 + 2m_3 + m_4\right). \end{cases} \tag{47}$$

$$\begin{cases} k_1 = u(x_n, y_n, z_n), \\ l_1 = v(x_n, y_n, z_n), \\ m_1 = w(x_n, y_n, z_n), \\ k_2 = u\left(x_n + \dfrac{\tau}{2}k_1, y_n + \dfrac{\tau}{2}l_1, z_n + \dfrac{\tau}{2}m_1\right), \\ l_2 = v\left(x_n + \dfrac{\tau}{2}k_1, y_n + \dfrac{\tau}{2}l_1, z_n + \dfrac{\tau}{2}m_1\right), \\ m_2 = w\left(x_n + \dfrac{\tau}{2}k_1, y_n + \dfrac{\tau}{2}l_1, z_n + \dfrac{\tau}{2}m_1\right), \\ k_3 = u\left(x_n + \dfrac{\tau}{2}k_2, y_n + \dfrac{\tau}{2}l_2, z_n + \dfrac{\tau}{2}m_2\right), \\ l_3 = v\left(x_n + \dfrac{\tau}{2}k_2, y_n + \dfrac{\tau}{2}l_2, z_n + \dfrac{\tau}{2}m_2\right), \\ m_3 = w\left(x_n + \dfrac{\tau}{2}k_2, y_n + \dfrac{\tau}{2}l_2, z_n + \dfrac{\tau}{2}m_2\right), \\ k_4 = u\left(x_n + \tau k_3, y_n + \tau l_3, z_n + \tau m_3\right), \\ l_4 = v\left(x_n + \tau k_3, y_n + \tau l_3, z_n + \tau m_3\right), \\ m_4 = w\left(x_n + \tau k_3, y_n + \tau l_3, z_n + \tau m_3\right). \end{cases} \tag{48}$$

en el cual, los parámetros k_i, con $i = 1, \ldots, 4$, se refieren a los coeficientes de la primera ecuación. De manera similar, los parámetros l_i con $i = 1, \ldots, 4$, se refieren a coeficientes de la segunda ecuación y los parámetros m_i, con $i = 1, \ldots, 4$, se refieren a coeficientes de la tercera ecuación del sistema descrito en

(47). Posteriormente estos coeficientes son calculados en (48). Los cálculos de cada paso del algoritmo (47)-(48) se conforman hasta completar una iteración, y de esta forma se describe el sistema (44).

5.2. Análisis del comportamiento caótico usando RMSE

Para calcular y comparar el rendimiento y la exactitud de los algoritmos numéricos descritos (45), (46) y (47)-(48), se utiliza el error cuadrado medio de raíz (RMSE). Para esta prueba, se reproduce la variable de estado x del sistema continuo de Lorenz (8). El RMSE se define como,

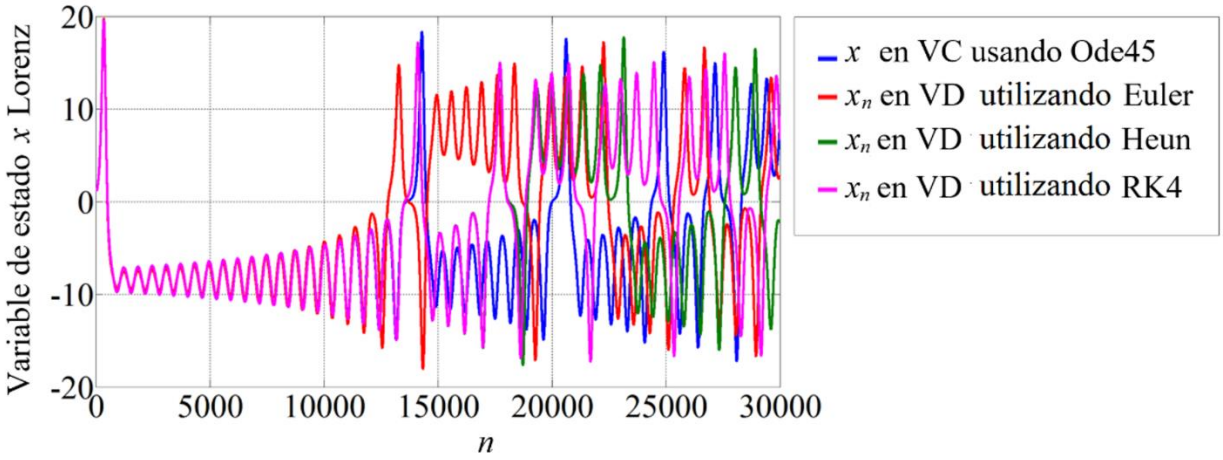
$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(x_{\mathrm{e}} - x_n\right)^2}\ , \tag{49}$$

en el cual, la variable de estado x_{e} es el valor estimado real, que en este caso corresponde a la variable caótica del estado x de la versión continua del sistema de Lorenz (8), el cual, se representa utilizando la función ode45. La variable de estado x_n se refiere al valor estimado calculado para la versión discretizada del sistema de Lorenz utilizando los algoritmos (45), (46) y (47)-(48) como el valor estimado. Ode45 es un algoritmo numérico que entrega una solución utilizando un solo paso, se basa en una fórmula explícita de Runge-Kutta 4 y 5 y el par Dormand-Prince y es uno de los métodos más utilizados en Matlab para caracterizar y solucionar EDO con el problema del valor inicial (44) (Dormand *et al.,*1980; Shampine y Reichelt, 1997).

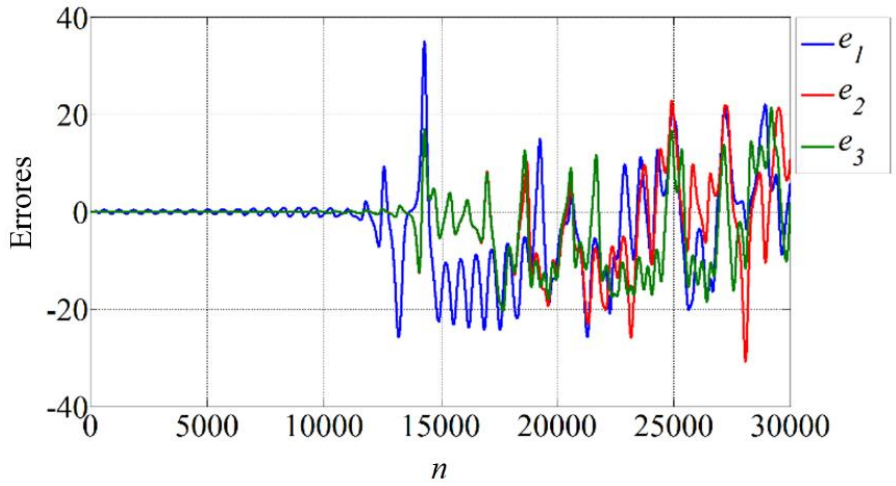
Para discretizar el sistema caótico de Lorenz (8), se consideraron $30{,}000$ iteraciones y un tamaño de paso pequeño $\tau = 0.001$ con el fin de obtener una degradación caótica baja en la versión discretizada del sistema caótico de Lorenz (8). Se utilizan los algoritmos numéricos de Euler (45), Heun (46), y RK4 (47)-(48) para obtener el sistema caótico de Lorenz en versión discretizada.

Se estudia la evolución de la trayectoria x del sistema caótico de Lorenz considerando x_{e} como referencia de la versión continua con respecto a x_n como la trayectoria del sistema caótico de Lorenz en versión discretizada. Para este estudio se definen los siguientes errores: e_1 representa la diferencia entre los algoritmos ode45 y Euler, e_2 representa la diferencia entre los algoritmos ode45 y Heun y e_3 representa la diferencia entre los algoritmos Ode45 y RK4.

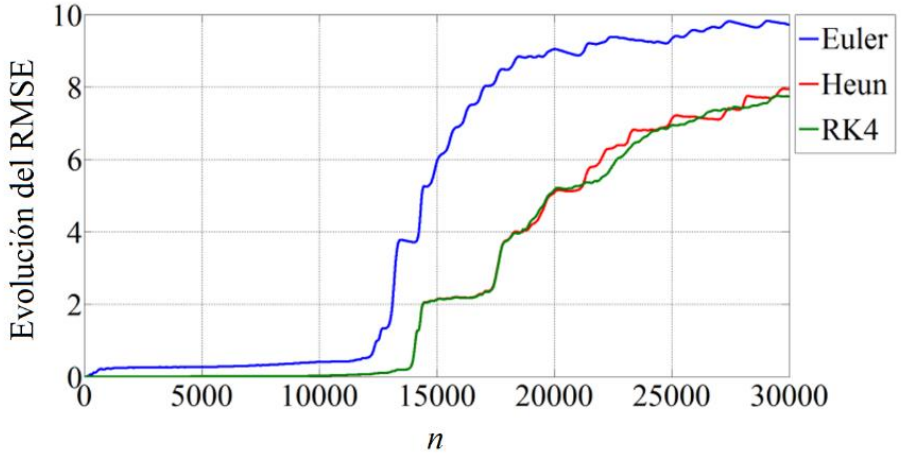
La figura 29 muestra el resumen de este estudio comparativo. En la figura 29a se muestra la evolución de la trayectoria de la variable de estado x del sistema caótico Lorenz (8) en versión continua con respecto a la versión discretizada utilizando los algoritmos numéricos (45)-(48). En la figura 29b se muestra la evolución de las trayectorias de los errores e_1, e_2 y e_3. Finalmente, en la figura 29c se muestra el cálculo acumulado del RMSE utilizando (49), donde el algoritmos de Euler (45) presenta un RMSE más elevado en comparación al RMSE obtenido utilizando Heun (46) y RK4 (47)-(48) que presentan comportamientos dinámicos similares.



(a) Comparación de las trayectorias de x del sistema caótico de Lorenz (8) utilizando ode45 para la versión continua y (45)-(48) para la versión discretizada.



(b) Comparación de errores de las trayectorias x del sistema caótico Lorenz en versión continua con respecto a x_n de la versión discretizada del sistema caótico Lorenz.



(c) RMSE de los algoritmos numéricos (45)-(48).

Figura 29. Comparación de las trayectorias x del sistema caótico de Lorenz (8) en versión continua con respecto a la versión discretizada, estimación de errores y evolución de RMSE.

5.3. Conclusiones del capítulo

En este capítulo se describieron los algoritmos numéricos de Euler, Heun y RK4 para la discretización de sistemas dinámicos. Se describió un estudio comparativo de las trayectorias de uno de los estados del sistema caótico de Lorenz en versión continua, considerando la función ode45 de Matlab con respecto a la versión discretizada utilizando los algoritmos numéricos Euler, Heun y RK4, y se estimaron los errores de trayectoria y evolución del RMSE. Los resultados concluyen que la discretización de Heun presenta una alternativa aceptable, para seguir la trayectoria del sistema de Lorenz considerando una secuencia de 30000 iteraciones, dado que su algoritmo numérico presenta simplicidad algebraica con respecto a RK4.

Capítulo 6. Sistemas embebidos

En este capítulo, se describe de manera general el diseño de hardware y software para implementar la familia de cinco sistemas caóticos en 3-D en versión discretizada reportados en esta tesis. Se describen los compiladores, tipos de microcontroladores presentados en 8, 16 y 32 bits y FPGA propuestos en la implementación de un sistema embebido presentado en 4 versiones, el cual, determina efectivamente la reproducción de los sistemas 5 sistemas caóticos en su versiones discretizadas.

6.1. Microcontroladores

Los microcontroladores han sido ampliamente empleados en la industria debido a su fácil programación, implementación y bajo costo (Angulo y Angulo, 1999). Actualmente existen diversos fabricantes de microcontroladores en este estudio en particular se trabaja con la familia de microcontroladores Microchip de gama baja de 8 bits, media de 16 bits y alta de 32 bits debido a su bajo costo y por la sencillez en el uso de sus herramientas de software y hardware. En la figura 30 se muestra la funcionalidad versus el desempeño de la familia de microcontroladores Microchip, en particular en este estudio se utilizan los microcontroladores gama baja PIC, gama media dsPIC y gama alta PIC32. Las diferencias técnicas entre estos microcontroladores son la memoria, funcionabilidad y desempeño; la selección que depende directamente de la aplicación que requiere el diseñador. El material y la familia de los microcontroladores Microchip está ampliamente reportada en la literatura y si el lector desea profundizar en el tema se recomiendan por ejemplo consultar las siguientes referencias (Angulo y Angulo, 1999; Di Jasio, 2007; Di Jasio, 2008).

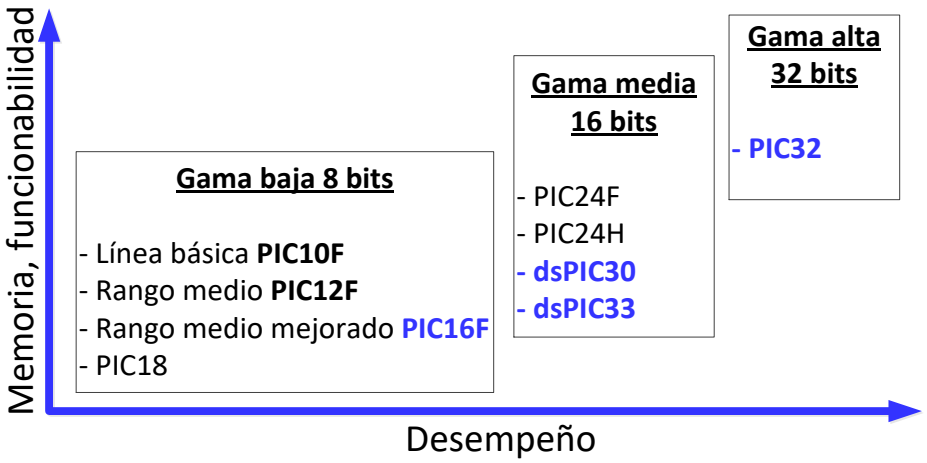


Figura 30. Familia de microcontroladores Microchip PIC, dsPIC y PIC32.

Para programar los microcontroladores PIC, dsPIC y PIC32, el fabricante Microchip propone utilizar el compilador MPLAB X IDE en sus variantes respectivas. En el mercado existen además, otros compiladores que ofrecen una alternativa más económica para programar los microcontroladores Microchip, en particular se utilizan los compiladores desarrollados por el fabricante Mikroelektronika para PIC, dsPIC y PIC32 (Mikroelektronika, 2017). En particular, los compiladores de Mikroelektronika son más económicos y ofrecen librerías que permiten desarrollar aplicaciones de manera amigable como librerías multimedia para pantallas Visual TFT, protocolo SPI, entre otras. En el capítulo 4 se mencionó las cualidades del compilador de Mikroelektronika para programar los microcontroladores dsPIC U1 y U2 utilizando el standard IEEE-754 con la extensión AN575, estas mismas características presentan los compiladores Mikroelektronika para PIC y PIC32; esto permite obtener resultados numéricamente equivalentes para la representación de los algoritmos numéricos simulados en Matlab, ver (Microchip Technology Inc., 2004) y (Yang *et al.*, 2005). Una vez, que el algoritmo del microcontrolador es compilado, se procede a grabar los microcontroladores Microchip, que en particular se utiliza el grabador ICD3 original de Microchip y como alternativa el grabador Mikroprog del fabricante Mikroelektronika.

Por otra parte, la simulación de los algoritmos numéricos utilizando los compiladores de los microcontroladores Microchip se reproduce numéricamente paso a paso, desafortunadamente esto dificulta la visibilidad de los registros internos de los algoritmos numéricos, y para grandes cantidad de iteraciones, no es posible determinar, y si se requiere utilizar periféricos externos con protocolos como el SPI, se dificulta aún más la visibilidad y comprensión del protocolo digital SPI, por ejemplo para escribir un DAC (Mikroelektronika, 2017). Como alternativa se utiliza el simulador Proteus de Labcenter ya que

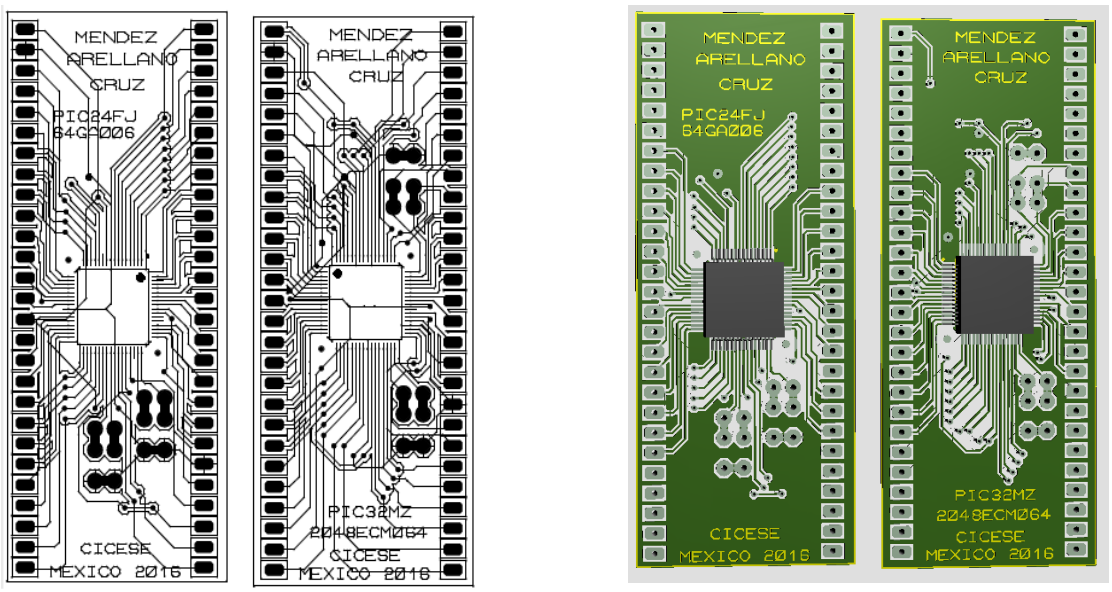
permite reproducir un algoritmo completo en ciertas familias de microcontroladores PIC de 8 bits (16F, 18F) y dsPIC de 16 bits (PIC24FJ, dsPIC33FJ). Además Proteus permite diseñar las placas electrónicas con encapsulados específicos para diseñar un sistema embebido completo.

Con el fin de resumir las características y las herramientas de software y hardware que utilizan los microcontroladores seleccionados para desarrollar las aplicaciones en este estudio, en la tabla 11 se muestran las especificaciones correspondientes.

Tabla 11. Descripción de hardware de los microcontroladores Microchip.

Unidad	**Descripción**	**Bits**	**Compilador basado en lenguaje C**	**Grabador**	**Simulador**	**Encapsulado**
U7	PIC16F874A	8	Mikro C for PIC	ICD3 / Mikroprog	Proteus	PDIP - 40 pines
U8	DSPIC33FJ32MC204	16	Mikro C for dsPIC	ICD3 / Mikroprog	Proteus	TQFP - 44 pines
U9	PIC32MZ2048ECM064	32	Mikro C for PIC32	ICD3 / Mikroprog	-	TQFP - 44 pines

Los microcontroladores U8 y U9 son encapsulado de montaje superficial TQFP (sigla en inglés *Thin Quad Flat Pack*) con conectores que se extienden en sus 4 lados, su estructura no permite montarse directamente en un protoboard, y necesariamente se utiliza Proteus para desarrollar placas electrónicas para obtener una versión de montaje sobre protoboard de U8 y U9 y realizar pruebas experimentales. La figura 31 se muestra el diseño de las placas considerando las especificaciones técnicas de U8 y U9.



(a) Diseño de placa utilizando Proteus. (b) Placas terminadas.

Figura 31. Desarrollo de placas para montar los microcontroladores U8 y U9.

6.2. FPGA

En palabras simples, un FPGA (por sus siglas en inglés *Field Programmable Gate Array*) es un dispositivo lógico programable cuyas funciones digitales básicas se programan a nivel físico (Muthuswamy y Banerjee, 2014). Los dispositivos FPGA se utilizan ampliamente para la reproducción de sistemas caóticos, debido a sus propicias herramientas para la simulación e implementación que disponen los fabricantes de FPGA, por ejemplo el fabricante Altera dispone como Quartus II, Modelsim, Nios II, Matlab/Simulink DSP Builder, entre otros (Altera Corporation, 2011; DSP Builder for Intel FPGAs, 2017; Tlelo-Cuautle *et al.*, 2015; 2016). Estas herramientas de software permiten trabajar con cualquier familia de FPGA Altera donde en particular, estamos utilizando la plataforma terasic DE2i-150 del fabricante Terasic (Terasic Inc., 2017). Esta plataforma contiene el FPGA Cyclone IV GX–EP4CGX150DF31C7 de uso comercial que se muestra en la figura 32.

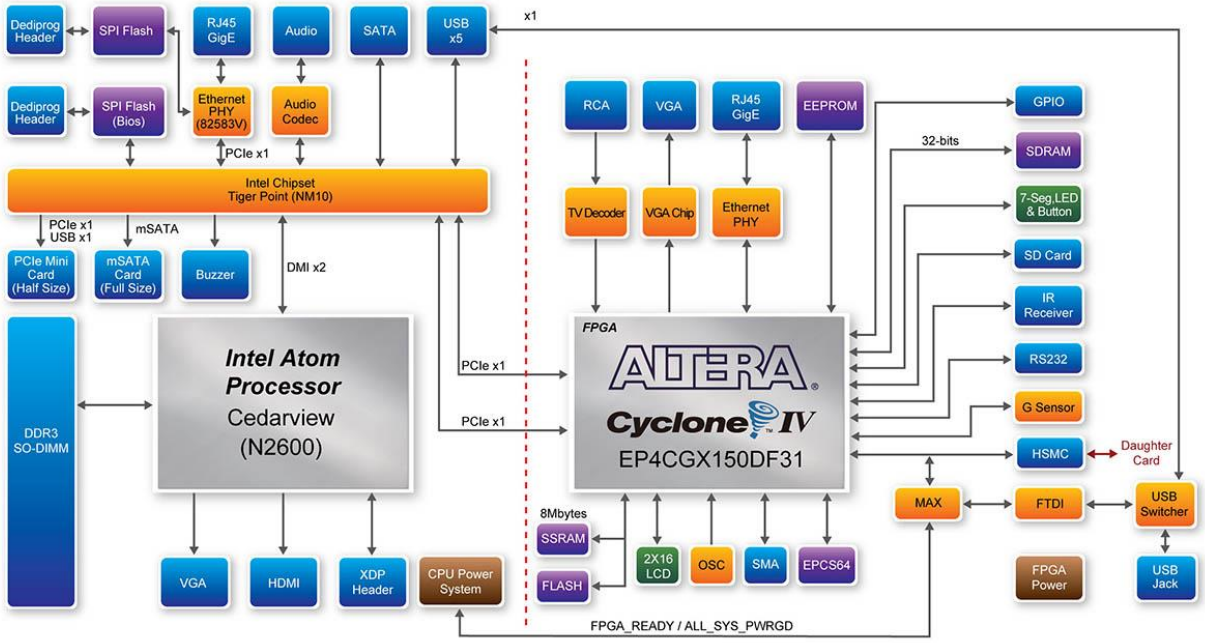


Figura 32. Diagrama de bloques de FPGA Terasic DE2i-150.

Las herramientas que poseen los dispositivos FPGA Altera son muy robustas, dado que es posible reproducir por ejemplo sistemas complejos de control utilizando la herramienta de Matlab Simulink DSP/Builder, pero desafortunadamente la obtención de las licencias del compilador Quartus II, DSP Builder y además de Matlab originales que en su conjunto tiene un costo de $10,000 USD (The Mathworks Inc., 2017). Con el uso de esta herramienta, se procede a reproducir la VD del sistema MACM (9), el método utilizado se denomina de integración que es similar a la discretización de Euler (46) pero la diferencia es que esta implementado en bloques, utilizando la interface de Simulink en el DSP Builder, este método fue propuesto por (Xiaohong y Zhiguang, 2013). El resultado de la versión discretizada del sistema MACM (9) se muestra en la figura 33.

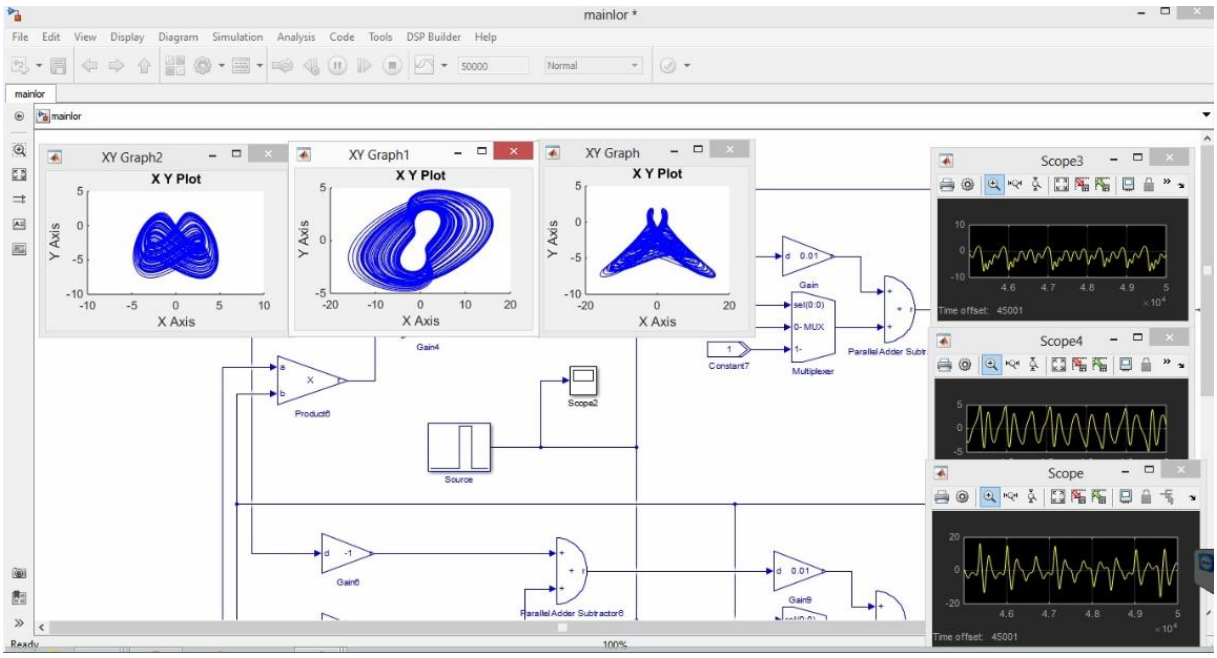


Figura 33. Simulación del sistema caótico MACM (9) utilizando la herramienta de Matlab Simulink/DSP Builder.

Otra alternativa es utilizar la novedosa herramienta Nios II-eclipse que posee el compilador Quartus II, esta herramienta permite el diseño de un microcontrolador embebido dentro del FPGA, y solo se requiere la licencia de Quartus II para utilizar la versión de Nios II/f (lo que significa que es la versión del microcontrolador de procesamiento Nios más rápida). Para personalizar el diseño del microcontrolador Nios II se requiere la herramienta Qsys (interconexión entre periféricos). La herramienta Qsys permite personalizar el diseño del microcontrolador y se pueden adicionar por ejemplo puertos de comunicación, almacenamiento de memoria, protocolos de comunicación serial SPI, I^2C, RS-232, entre otros (Altera Corporation, 2011). Nios II utiliza el estándar IEEE-754 2008 (IEEE standard for floating-point arithmetic-redline, 2008); esto significa que los resultados cálculos numéricos en simulación utilizando Matlab y en implementación utilizando un FPGA son equivalentes. En la tabla 12 se describe el hardware y software del dispositivo FPGA utilizado.

Tabla 12. Descripción de hardware del FPGA.

Unidad	Descripción	Bits	Compilador basado en lenguaje C	Grabador	Simulador SE	Encapsulado
U10	EP4CGX150DF31C7, Terasic DE2-i150.	32	Quartus II, Qsys y Nios II eclipse	Quartus II / programmer	Consola Nios II	FBGA – 148 pines

En este trabajo de tesis, se introduce el procesador Nios II como un microcontrolador embebido en el FPGA, considerando las mismas condiciones de software y hardware que poseen los microcontroladores mostrados en la tabla 11. Para la implementación de hardware interno del FPGA, se utilizan las siguientes librerías personalizadas de la herramienta Qsys del compilador Quartus II:

- Procesador Nios II/f (versión rápida),
- Reloj acelerado para una frecuencia 150 MHz, utilizando el multiplicador avalon altpll,
- Jtag, Uart,
- 200Kb en memoria de programa,
- Periféricos de ID del sistema,
- Control de bus SPI para el control de 3 dispositivos externos en modo esclavo.

6.3. Conclusiones del capítulo

En este capítulo se describieron las herramientas de software y hardware, que se utilizan para implementar un sistema embebido, además se mostraron las características relevantes de los microcontroladores Microchip PIC, dsPIC, PIC32 de 8, 16 y 32 bits; respectivamente, y la versión del microcontrolador Nios II en el FPGA Altera. El estudio concluye que estas 4 familias de microprocesadores son altamente atractivas para implementaciones de los algoritmos numéricos basados en lenguaje C utilizados en esta tesis, ya que los compiladores de los microcontroladores y del dispositivo FPGA están normalizados bajo el estándar IEEE-754, que es mismo standard que se utiliza en las simulaciones numéricas que utiliza Matlab. Las herramientas de dearrollo que tiene Proteus concluyen que es una herramienta visual que permite realizar simulaciones con microcontroladores Microchip de 8 y 16 bits

como parte central de un sistema embebido, y permite construir placas electrónicas para montar los microcontroladores de montaje superficial dsPIC33 y PIC32MZ.

Capítulo 7. Estudio de degradación de caos en sistema discretizados

En este capítulo se presenta un estudio de los límites de conservación de la dinámica caótica en la versión discretizada de los sistemas caóticos en 3 dimensiones de Lorenz, Rössler, Chen, Liu y Chen y MACM. Se muestran resultados numéricos, utilizando el método de series de tiempo para calcular los exponentes de Lyapunov y determinar efectivamente los límites de la dinámica caótica de los 5 sistemas caóticos propuestos utilizando los métodos numéricos de Euler, Heun y RK4.

7.1. Degradación de sistemas caóticos continuos en 3-D

La degradación de un sistema caótico mediante el método de series de tiempo permite obtener numéricamente los exponentes de Lypunov para determinar la conservación del caos dentro del sistema (Briggs, 1990 y Wolf *et al.*, 1985). Se estudia un conjunto de 5 sistemas caóticos, donde los sistemas de Lorenz (8) y MACM (9), fueron descritos en los capítulos anteriores y a continuación se describen los otros sistemas caóticos. El sistema de Rössler se reportó en 1976 (Rössler, 1976), y se describe a continuación,

$$\begin{cases} \dot{x} = -y - z, \\ \dot{y} = x + ay, \\ \dot{z} = b + z(x - c). \end{cases} \tag{50}$$

El sistema de Rössler presenta comportamientos caóticos con los siguientes parámetros: $a = 0.2$, $b = 0.2$ y $c = 5.7$.

El sistema de Chen es reportado en (Chen, 1999), como un sistema dual al sistema de Lorenz, el sistema es descrito como,

$$\begin{cases} \dot{x} = a(y - x), \\ \dot{y} = (c - a)x - xz + cy, \\ \dot{z} = xy - bz. \end{cases} \tag{51}$$

El sistema de Chen presenta comportamientos caóticos con los siguientes parámetros: $a = 35$, $b = 3$ y $c = 28$.

Por otra parte, el sistema de Liu y Chen fue reportado en 2002 (Liu y Chen, 2002), descrito por las siguientes ecuaciones diferenciales:

$$\begin{cases} \dot{x} = ax + d_1 yz, \\ \dot{y} = cy + d_2 xz, \\ \dot{z} = bz + d_3 xy. \end{cases} \tag{52}$$

Este sistema no lineal presenta un comportamiento caótico cuando se cumple la siguiente condición $ab + ac + bc \neq 0$. En este sistema se crea un atractor caótico con dos enrollamientos utilizando los siguientes parámetros: $d_1 = -1$, $d_2 = d_3 = 1$, $a = 5$, $c = -10$ y $b = -3.4$.

Estudios comparativos del conjunto de sistemas caóticos (8)-(9), (50)-(52) en su versión discretizada, utilizando el algoritmo numérico de Euler (45) están reportados en (Mendez-Ramírez *et al.*, 2017). En la tabla 13 se presenta el resumen de los 5 sistemas caóticos en 3-D, en el cual, todas las pruebas numéricas se realizaron considerando las mismas condiciones iniciales.

Tabla 13. Resumen parámetros y condiciones iniciales de los sistemas caóticos.

Sistema caótico	Parámetros	Parámetro crítico	No linealidades	Condiciones iniciales
Lorenz (8)	σ, r, b	σ	2	(1, 1, 1)
Rössler (50)	a, b, c	c	1	(1, 1, 1)
Chen (51)	a, b, c	a	2	(1, 1, 1)
Liu y Chen (52)	a, b, c, d_1, d_2, d_3	c	3	(1, 1, 1)
MACM (9)	a, b, c, d	b, d	2	(1, 1, 1)

Con el fin de verificar la presencia de caos, se calculan los exponentes de Lyapunov, utilizando el método de series de tiempo, en el cual la matriz jacobeana es calculada para la versión continua de los 5 sistemas caóticos en 3-D (8)-(9) y (50)-(52), determinada por

$$\boldsymbol{J}(f, g, h) = \begin{pmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} & \frac{\partial f}{\partial z} \\ \frac{\partial g}{\partial x} & \frac{\partial g}{\partial y} & \frac{\partial g}{\partial z} \\ \frac{\partial h}{\partial x} & \frac{\partial h}{\partial y} & \frac{\partial h}{\partial z} \end{pmatrix}. \tag{53}$$

La tabla 14 muestra los resultados del cálculo de los exponentes de Lyapunov y la dimensión fractal D_{KY} (conocida como la dimensión de Kaplan-Yorke) de los 5 sistemas caóticos en 3-D (8)-(9) y (50)-(52), calculados utilizando (53). En las siguientes subsecciones, se presentará la comparación de los sistemas caóticos en versión continua con respecto a la versión discretizada de los 5 sistemas caóticos.

Tabla 14. Exponentes Lyapunov para los 5 sistemas caóticos en 3-D en versión continua.

Sistemas caóticos	L_1	L_2	L_3	D_{KY}
Lorenz (8)	0.91	0	-14.47	2.062
Rössler (50)	0.07	0	-5.39	2.012
Chen (51)	2.02	0	-12.02	2.168
Liu y Chen (52)	0.87	0	-9.27	2.093
MACM (9)	0.24	0	-2.74	2.087

7.2. Estudio de degradación de los 5 sistemas caóticos en 3-D

Se obtienen las versiones discretizadas de los 5 sistemas caóticos en 3-D utilizando los algoritmos numéricos Euler, Heun y RK4 descritos en (45), (46) y (47)-(48). La versión discretizada de la matriz jacobeana se describe por

$$\boldsymbol{J}_D(f,g,h)=\begin{pmatrix} \frac{\partial f}{\partial x_n} & \frac{\partial f}{\partial y_n} & \frac{\partial f}{\partial z_n} \\ \frac{\partial g}{\partial x_n} & \frac{\partial g}{\partial y_n} & \frac{\partial g}{\partial z_n} \\ \frac{\partial h}{\partial x_n} & \frac{\partial h}{\partial y_n} & \frac{\partial h}{\partial z_n} \end{pmatrix}. \tag{54}$$

7.2.1. Discretización por Euler

En la tabla 15 se muestran los resultados de la degradación de los 5 sistemas caóticos en 3-D (8)-(9) y (50)-(52) utilizando el algoritmo de Euler (45); se encuentra un tamaño de paso máximo τ_{max}, donde se conservan las dinámicas caóticas de estos 5 sistemas, para los cuales, se calculan los exponentes de Lyapunov (el sistema caótico MACM presenta mejor desempeño en su versión discreta y este estudio esta reportado en Mendez-Ramírez *et al.*, 2017). El resultado de este estudio muestra interesantes comportamientos, por ejemplo del sistema de Rössler (50) discretizado por el algoritmo numérico de Euler

(46), presenta comportamientos de ciclo límite para valores $0.005 \leq \tau \leq 0.091$ y para valores más altos como $\tau = 0.091$ el sistema de Rössler mediante (54) utilizando la discretizacion (45) diverge, los exponentes de Lyapunov no pueden ser calculados; se recomienda utilizar tamaños de paso para valores bajos.

Tabla 15. Exponentes de Lyapunov para los 5 sistemas caóticos en 3-D (8)-(9) y (50)-(52) discretizados utilizando el algoritmo numérico de Euler (45).

Sistema caótico	τ_{max}	L_1	L_2	L_3	D_{KY}
Lorenz	≤ 0.024	0.039	-62.8 μ	-0.389	2.101
Rössler	≤ 0.005	49 μ	34.9 μ	-5.53	2.0142
Chen	≤ 0.002	3.8 m	0.11 m	-23.1 m	2.169
Liu y Chen	≤ 0.002	197 μ	-53.21 μ	-18.72 m	2.1029
MACM	≤ 0.085	0.05	-64.58 μ	-0.243	2.099

7.2.2. Discretización por Heun

En la tabla 16 se muestran los resultados del estudio de degradación de los 5 sistemas caóticos en 3-D (8)-(9) y (50)-(52) utilizando el algoritmo de Heun (46); se encuentra un tamaño de paso máximo τ_{max}, donde se conservan las dinámicas caóticas de estos 5 sistemas, para los cuales, se calculan los exponentes de Lyapunov.

Tabla 16. Exponentes de Lyapunov para los 5 sistemas caóticos en 3-D (8)-(9) y (50)-(52) discretizados utilizando el algoritmo numérico de Heun (46).

Sistema caótico	τ_{max}	L_1	L_2	L_3	D_{KY}
Lorenz	≤ 0.068	86.6 m	-2.016 m	-0.6231	2.1358
Rössler	≤ 0.191	0.0273	196 μ	-0.24122	2.1141
Chen	≤ 0.017	38.7 m	23.67 μ	-0.19558	2.1984
Liu y Chen	≤ 0.017	17.41 m	-17.88 μ	-0.15542	2.112
MACM	≤ 0.228	95.29 m	315.64 μ	-0.476	2.2006

7.2.3. Discretización por RK4

Por último, en la tabla 17 muestra la degradación caótica de los 5 sistemas caóticos en 3-D (8)-(9) y (50)-(52) utilizando el algoritmo de RK4 (47)-(48); se encuentra un tamaño de paso máximo τ_{max}, donde se conservan las dinámicas caóticas de estos 5 sistemas, para los cuales, se calculan los exponentes de Lyapunov.

Tabla 17. Exponentes de Lyapunov para los 5 sistemas caóticos en 3-D (8)-(9) y (50)-(52) discretizados utilizando el algoritmo numérico de RK4 (47)-(48).

Sistema caótico	τ_{max}	L_1	L_2	L_3	D_{KY}
Lorenz	≤ 0.106	64.17 m	-4.03 m	-1.1924	2.0504
Rössler	≤ 0.251	35.53 m	339.62 μ	-0.402	2.0891
Chen	≤ 0.057	97.48 m	-2.949 m	-0.6509	2.1452
Liu y Chen	≤ 0.057	55.52 m	-1.83 m	-0.389	2.0974
MACM	≤ 0.547	0.23081	-48.55 m	-1.0098	2.1805

Finalmente, en la figura 34 se obtiene la degradación caótica resumida en las tablas 15-17 del conjunto de los 5 sistemas caóticos en 3-D (8)-(9) y (50)-(52) utilizando los algoritmos numéricos (45)-(48).

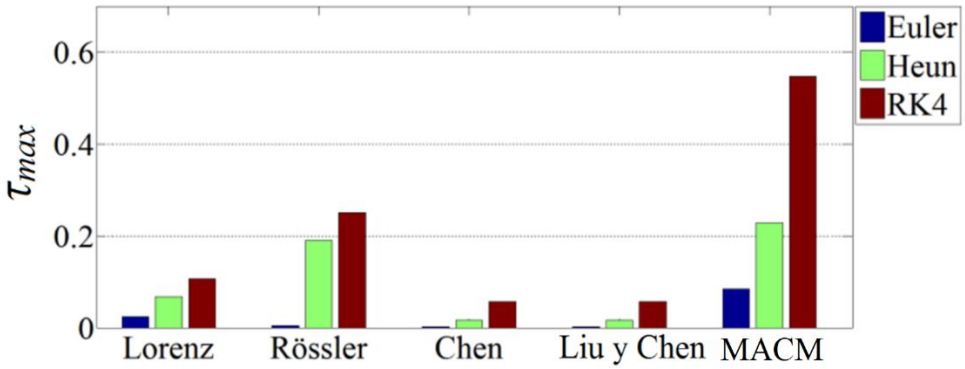


Figura 34. Resumen del tamaño de paso máximo donde se conserva el caos en los 5 sistemas caóticos (8)-(9) y (50)-(52) en versión discretizada utilizando los algoritmos numéricos (45), (46) y (47)-(48).

7.3. Conclusiones del capítulo

En este capítulo se realizaron pruebas numéricas para encontrar los límites de la dinámica caótica de los sistemas caóticos de Lorenz, Rössler, Chen, Liu y Chen y MACM en su versión discretizada utilizando los algoritmos numéricos de Euler, Heun y RK4. Los resultados concluyen que al obtener el tamaño de paso de los 5 sistemas caóticos con los 3 algoritmos numéricos estudiados en esta tesis, es atractivo para utilizarse, por ejemplo, en aplicaciones de comunicaciones seguras, ya que se puede utilizar como parámetro adicional si se desea implementar como parte de una clave secreta para aplicaciones de sistemas criptográficos digitales. El nuevo sistema caótico MACM presenta un mejor desempeñó en su versión discreta utilizando los algoritmos numéricos de Euler, Heun y RK4, ya que su tamaño de paso es más alto en relación a los otros sistemas caóticos, lo cual, concluye que es altamente atractivo para

también para implementarse en su versión discreta digitalmente en aplicaciones de sincronización y encriptación.

Capítulo 8. Implementación de sistemas caóticos en 3-D discretizados en sistema embebidos

En este capıtulo se presentan resultados experimentales de un sistema embebido presentado en 4 versiones utilizando microcontroladores Microchip de 8, 16 y 32 bits y un FPGA Altera, en los cuales, se implementan los algoritmos numéricos de Euler, Heun y RK4 para establecer efectivamente la reproducción de los 5 sistemas caóticos Lorenz, Rössler, Chen, Liu y Chen y MACM. Se establecen la generación de planos de fase y trayectorias caóticas de los 5 sistemas caóticos en sus versiones discretizadas obtenidas en este trabajo de tesis en simulación e implementación dependiendo de la versión del sistema embebido. Se habla en particular del rendimiento en la implementación de los algoritmos numéricos en el sistema embebido en sus 4 versiones propuestos en esta tesis, del cual se describe de manera general las herramientas de implementación de software y hardware de los simuladores y compiladores.

8.1. Diseño del sistema embebido

En capítulos anteriores se describieron las herramientas de software y hardware que poseen los compiladores microcontroladores Microchip PIC, dsPIC, PIC32 y los dispositivos FPGA Altera. Para el diseño del sistema embebido se utilizan los mismos DAC MCP4921 reportados en el capítulo 4 (Microchip Technology Inc., 2004), y las variables de estado son representadas por $x_{(t)}$, $y_{(t)}$ y $z_{(t)}$ que corresponde a la a las variables de estado para cada sistema caótico descrito en (8)-(9) y (50)-(52). Para la construcción de las versiones del sistema embebido, se utiliza el hardware que se describe en la tabla 18.

Tabla 18. Descripción de hardware del sistema embebido.

Unidad	Descripción de hardware y modo de configuración SPI
U7	Microcontrolador PIC 16F874A, modo maestro
U8	Microcontrolador dsPIC33FJ32MC204, modo maestro
U9	Microcontrolador PIC32MZ2048ECM064, modo maestro
U10	FPGA EP4CGX150DF31C7, modo maestro
U11	DAC MCP4921 muestra variable de estado $x_{(t)}$, modo esclavo 1
U12	DAC MCP4921 muestra variable de estado $y_{(t)}$, modo esclavo 2
U13	DAC MCP4921 muestra variable de estado $z_{(t)}$, modo esclavo 3

Los DACs U11-U13 están conectados externamente en modo esclavo y solo cambia la parte central del sistema embebido propuesto, que está en modo maestro donde la versión 1 (V1) se implementa con U7 y U11-U13, la versión 2 (V2) se implementa con U8 y U11-U13, la versión 3 (V3) se implementa con U9 y U11-U13. Por último, la versión 4 (V4) se implementa con U10 y U11-U13. Los microcontroladores U7- U9 y el FPGA U10 se configuran de acuerdo con el rendimiento recomendado por los fabricantes y en la figura 35 describe el sistema embebido representado en las 4 versiones propuestas.

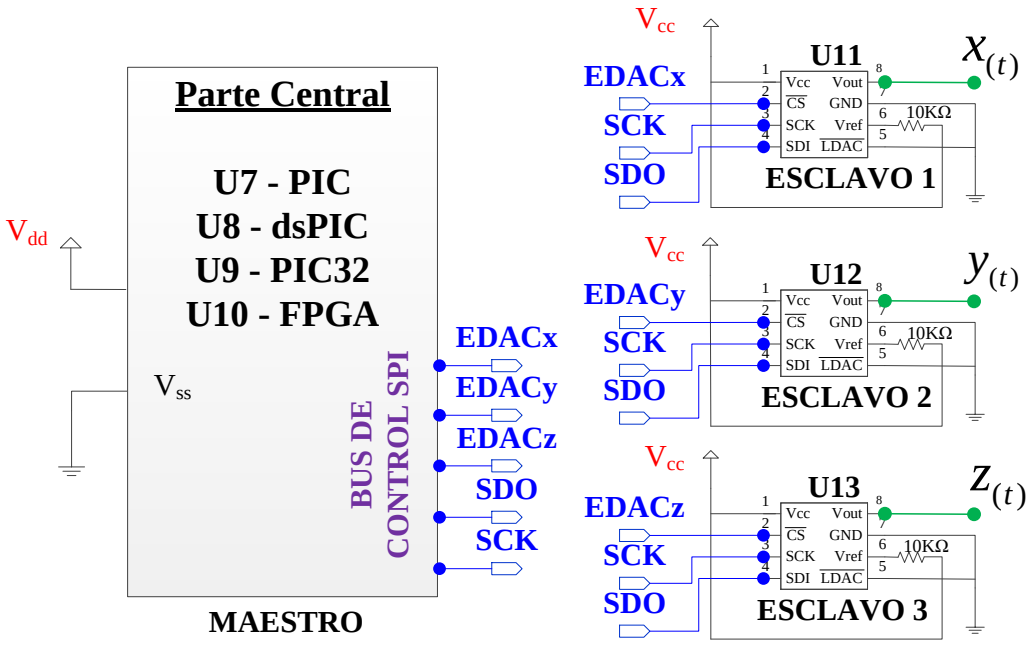


Figura 35. Diagrama esquemático del sistema embebido para V1-V4.

Se implementan los 5 sistemas caóticos en 3-D en versión discretizada utilizando los algoritmos numéricos (45), (46) y (47)-(48) sobre las 4 versiones del sistema embebido propuesto considerando las especificaciones de U7-U10 para obtener el mejor desempeño recomendado por el fabricante (Microchip Technology Inc.: 1997, 2004). Se utiliza el método reportado por (Méndez-Ramírez *et al.*, 2017) para estimar la semejanza en la simulación e implementación del sistema embebido implementado en las 4 versiones mostradas en la figura 35. Tomando en consideración la expresión (41) utilizando U7-U10, respectivamente, se define la cantidad total de iteraciones Q_T como el máximo número de n iteraciones generadas en 1 segundo,

$$Q_T = \frac{\tau}{T_Q} = \frac{\tau}{t_c + t_{Tg}} = f_Q \tau \,, \tag{55}$$

en el cual, el periodo de tiempo T_Q se considera como el tiempo total que el algoritmo numérico de U7-U10 necesita para reproducir una iteración n y τ es el tamaño de paso, el cual, es deseable que su valor numérico sea el máximo de acuerdo al estudio propuesto en el capítulo 7. El periodo de tiempo T_Q está conformado por la suma del tiempo de complejidad t_c del algoritmo numérico y el tiempo total t_{Tg} que necesitan los DACs U11-U13 para graficar las variables de estado de la versión discretizada de los sistemas caóticos en 3-D reportados. f_Q es el recíproco de T_Q, la frecuencia f_Q representa el número máximo de iteraciones n que el sistema embebido genera en 1 segundo (ips). En este estudio, consideramos la ecuación (55) como referencia para obtener el desempeño de los cinco sistemas caóticos en 3-D (8)-(9) y (50)-(52) reproducidos en sus versiones discretizadas utilizando los algoritmos numéricos (45), (46) y (47)-(48) sobre el sistema embebido implementado en las V1-V4.

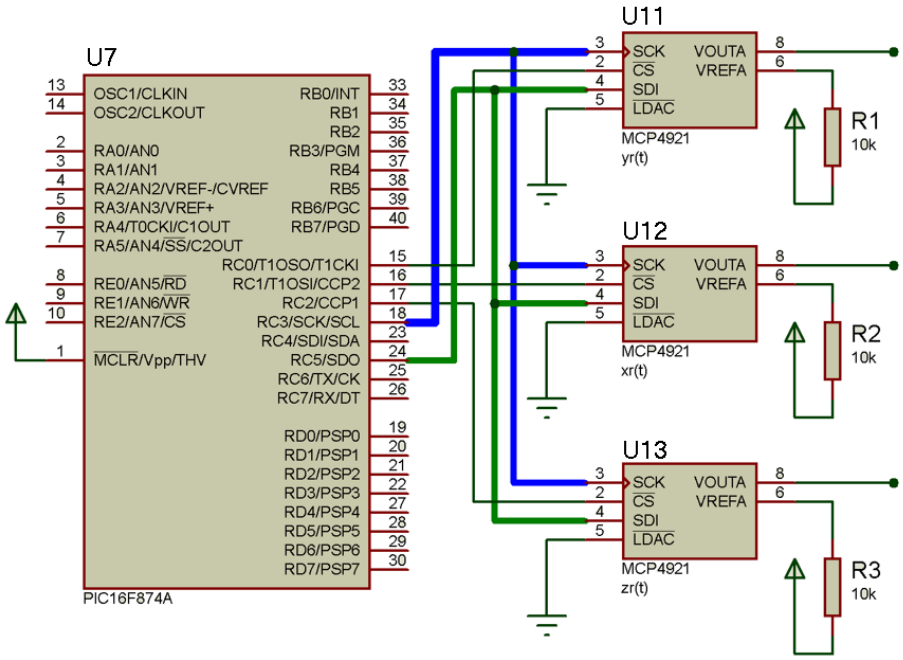
8.1.1. Sistema embebido implementado en V1: U7 – PIC

La primera versión del sistema embebido propuesto se reproduce inicialmente en el simulador de Proteus que soporta la librería de hardware para reproducir el microcontrolador U7 de 8 bits, utilizando un cristal externo de 10 MHz como oscilador, V_{cc} = V_{dd} = +5 V, V_{ss} = 0 V y los DACs externos U11-U13 conectados a U7. Como ejemplo, se utiliza el algoritmo de Euler (45) para obtener el sistema caótico de Lorenz (9) en versión discreta que está dado por

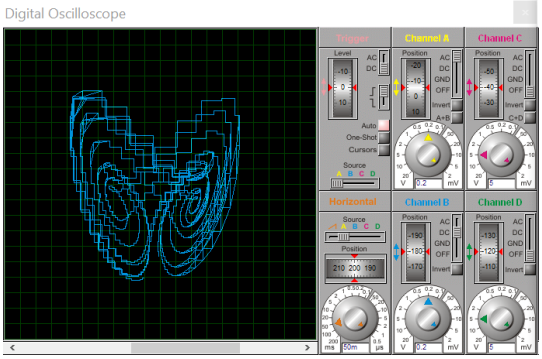
$$\begin{cases} x_{n+1} = x_n + \tau\left(\sigma\left(y_n - x_n\right)\right), \\ y_{n+1} = y_n + \tau\left(rx_n - x_n z_n - y_n\right), \\ z_{n+1} = z_n + \tau\left(x_n y_n - bz_n\right). \end{cases} \tag{56}$$

La figura 36 muestra la simulación en Proteus de la 1ra versión del sistema embebido propuesto utilizando el sistema caótico de Lorenz discretizado (56), considerando $\tau_{max} = 0.024$; en la figura 36a se muestra el diagrama esquemático, en la figura 36b se muestran los planos de fase de $x_{1(n)}$ versus $z_{1(n)}$ y en la figura 36c se muestran las trayectorias de x_n y z_n. Con fin de establecer una comparación, se implementa experimentalmente el sistema caótico de Lorenz discretizado (56) considerando el mismo $\tau_{max} = 0.024$ en la V1 del sistema embebido donde en la figura 37 se muestra el planos de fase x_n versus z_n y las trayectorias de las variables de estado x_n y z_n obtenidas. Se reproducen los cinco sistemas caóticos en 3-D (8)-(9) y (50)-(52) reproducidos en sus versiones discretas utilizando los algoritmos numéricos (45) y (46) sobre el sistema embebido implementado en la V1, el resultado del desempeño utilizando el microcontrolador U7

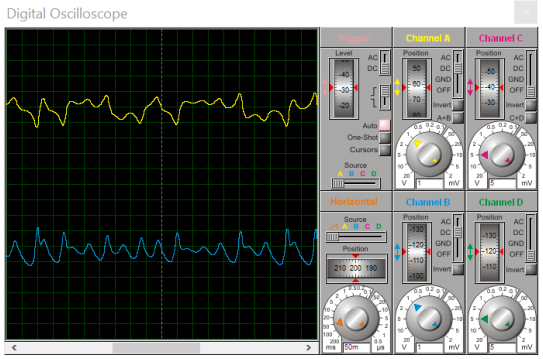
se muestra en la tabla 19. Dada la características de gama baja y de la capacidad de procesamiento de U7, la memoria de programa que posee este microcontrolador, no permite reproducir los 5 sistemas caóticos propuestos utilizando el algoritmo numérico RK4 (47)-(48).



(a) Diagrama esquemático del sistema embebido para la V1 utilizando Proteus.



(b) Plano de fase x_n versus z_n.

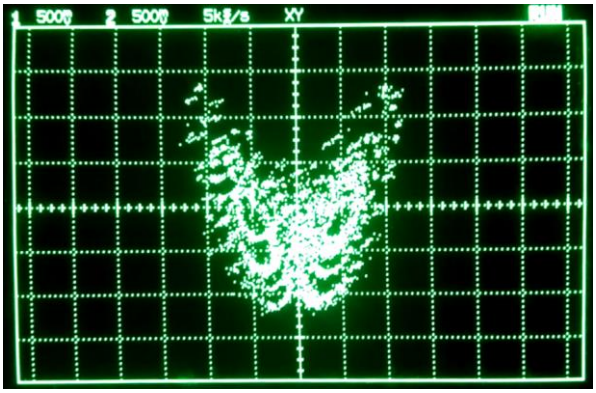


(c) Evolución de las variables de estado x_n y z_n utilizando $\tau_{max} = 0.024$.

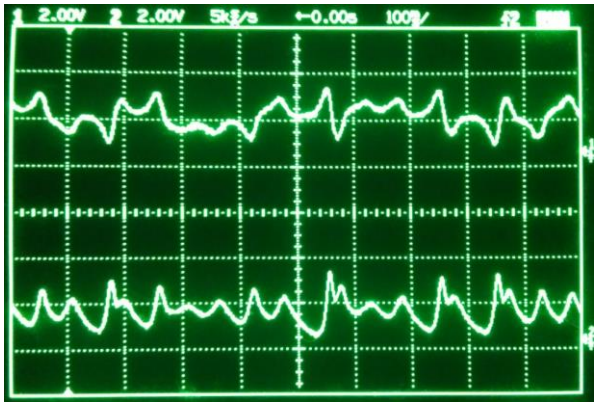
Figura 36. Simulación del sistema embebido en la V1 del sistema caótico Lorenz (56) en versión discreta utilizando el algoritmo numérico (45).

Tabla 19. Desempeño de los cinco sistemas caóticos (8)-(9) y (50)-(52) discretizados utilizando los algoritmos numéricos (45)-(46), en V1.

Sistema caótico	**Euler**		**Heun**	
	***T* (µs)**	***f* (ips)**	***T* (µs)**	***f* (ips)**
Lorenz	2046	488.7	3450	289.8
Rössler	1720	581.3	2760	362.3
Chen	2046	488.7	3450	289.8
Liu y Chen	2480	403.2	4420	226.2
MACM	2046	488.7	3450	289.8



(a) Plano de fase x_n versus z_n.



(b) Evolución de las variables de estado x_n y z_n utilizando $\tau_{max} = 0.024$.

Figura 37. Implementación del sistema embebido en la V1 del sistema caótico de Lorenz (56) discretizado utilizando el algoritmo numérico (45).

8.1.2. Sistema embebido implementado en V2: U8 – dsPIC

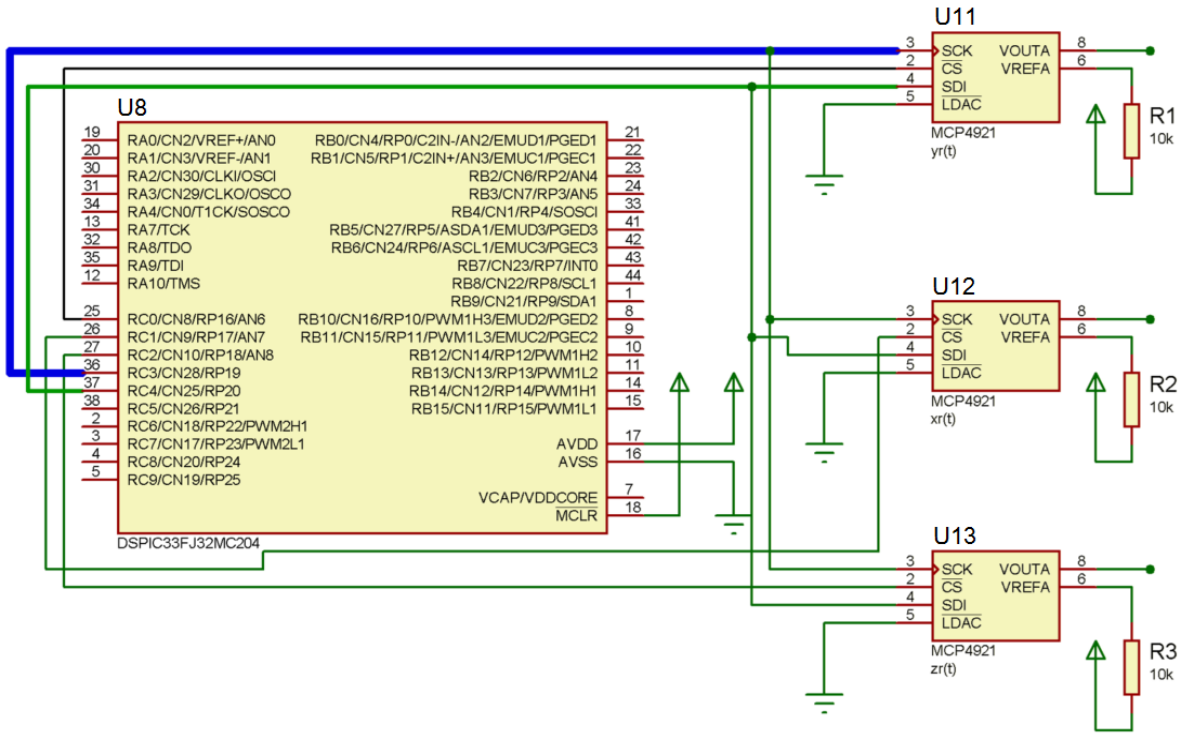
La segunda versión del sistema embebido, se reproduce utilizando el simulador Proteus que soporta la librería de hardware del microcontroladores de la familia dsPIC33, para realizar este estudio utilizando el microcontrolador U8 de 16 bits, en la implementación de hardware de V2 se utiliza el cristal interno de U8 como oscilador, V_{cc} = V_{dd} = +3.5 V, V_{ss} = 0 V y los DACs U11-U13 conectados externamente. Como ejemplo, se utiliza el algoritmo de Euler (45) para obtener el sistema caótico de Liu y Chen (52) en su versión discretizada está dado por:

$$\begin{cases} x_{n+1} = x_n + \tau(ax_n + d_1 y_n z_n), \\ y_{n+1} = y_n + \tau(cy_n + d_2 x_n z_n), \\ z_{n+1} = z_n + \tau(bz_n + d_3 x_n y_n). \end{cases} \tag{57}$$

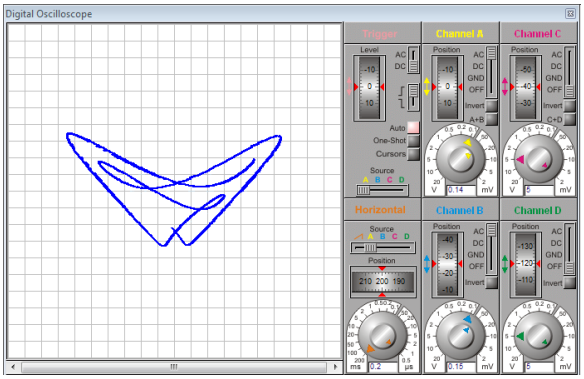
La figura 38 muestra la simulación en Proteus de la 1ra versión del sistema embebido propuesto, utilizando sistema caótico de Liu y Chen discretizado (57), considerando $\tau_{max} = 0.002$; en la figura 38a se muestra el diagrama esquemático, en la figura 38b se muestran los planos de fase de x_n versus z_n y en la figura 38c se muestran las trayectorias de x_n y z_n. Con fin de establecer una comparación, se implementa experimentalmente el sistema caótico de Liu y Chen en versión discreta (57), considerando el mismo $\tau_{max} = 0.002$ en la V2 del sistema embebido, donde en la figura 39 se muestra el planos de fase x_n versus z_n y las trayectorias de las variables de estado x_n y z_n obtenidas. Se reproducen los cinco sistemas caóticos en 3-D (8)-(9) y (50)-(52) reproducidos en sus versiones discretas utilizando los algoritmos numéricos (45), (46) y (47)-(48) sobre el sistema embebido implementado en V2, el resultado del desempeño utilizando el microcontrolador U8 se muestra en la tabla 20. Dada la características de gama media y de la capacidad de procesamiento de U8, la memoria de programa permite reproducir los 5 sistemas caóticos considerados, utilizando el algoritmo numéricos propuestos (45)-(48).

Tabla 20. Desempeño de los cinco sistemas caóticos (8)-(9) y (50)-(52) discretizados utilizando los algoritmos numéricos (45)-(48), en V2.

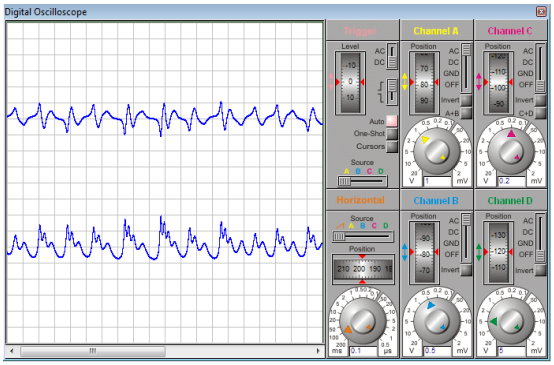
Sistema caótico	**Euler**		**Heun**		**RK4**	
	***T* (µs)**	***f*** (ips)	***T* (µs)**	***f*** (ips)	***T* (µs)**	***f*** (ips)
Lorenz	84	11905	138	7246	245	4082
Rössler	81	12345	132	7575	233	4291
Chen	84	11905	138	7246	245	4082
Liu y Chen	89	11236	143	6993	271	3690
MACM	84	11905	138	7246	245	4082



(a) Diagrama esquemático del sistema embebido para la V2 utilizando Proteus.

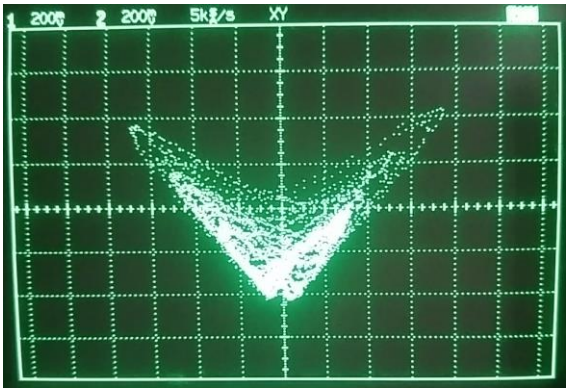


(b) Plano de fase x_n versus z_n.

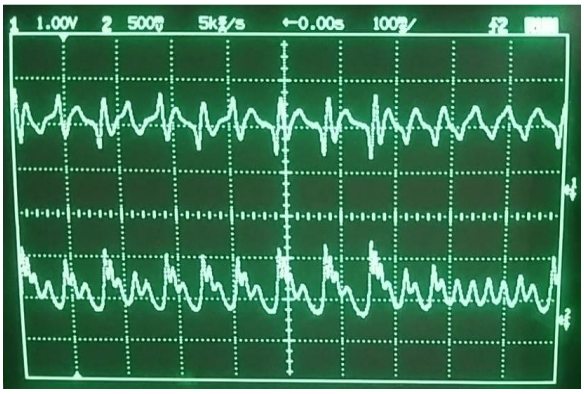


(c) Evolución de las variables de estado x_n y z_n utilizando $\tau_{max} = 0.002$.

Figura 38. Simulación del sistema embebido en V2 del sistema caótico de Liu y Chen (57) discretizado, utilizando el algoritmo numérico (45).



(a) Plano de fase x_n versus z_n.



(b) Evolución de las variables de estado x_n y z_n utilizando $\tau_{max} = 0.002$.

Figura 39. Implementación del sistema embebido en V2 del sistema caótico de Liu y Chen (57) discretizado, utilizando el algoritmo numérico (45).

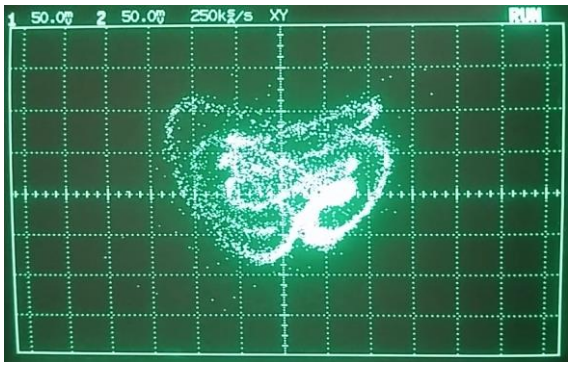
8.1.3. Sistema embebido implementado en V3: U9 – PIC32

La tercera versión del sistema embebido propuesto solo se puede simular numéricamente en su parte central, utilizando el compilador Mikro C for PIC32 ya que Proteus no soporta la familia de microcontroladores PIC32. En la implementación de hardware en V3, se utiliza como oscilador el cristal interno de U9, $V_{cc} = V_{dd} = +3.5$ V, $V_{ss} = 0$ V y los DACs U11-U13 conectados externamente. Como ejemplo, se utiliza el algoritmo de Heun (46) para obtener el sistema caótico de Chen (52) en versión discreta que está dado por

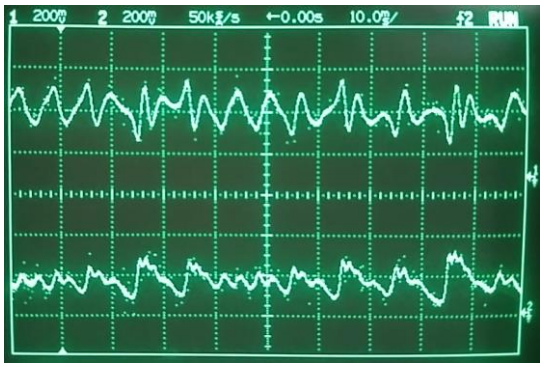
$$\begin{cases} x_{n+1}^{*} = x_n + \tau\left(a\left(y_n - x_n\right)\right), \\ y_{n+1}^{*} = y_n + \tau\left(\left(c-a\right)x_n - x_n z_n + c y_n\right), \\ z_{n+1}^{*} = z_n + \tau\left(x_n y_n - b z_n\right), \\ x_{n+1} = x_n + \dfrac{\tau}{2}\left(a\left(y_n - x_n\right) + x_{n+1}^{*}\right), \\ y_{n+1} = y_n + \dfrac{\tau}{2}\left(\left(c-a\right)x_n - x_n z_n + c y_n + y_{n+1}^{*}\right), \\ z_{n+1} = z_n + \dfrac{\tau}{2}\left(x_n y_n - b z_n + z_{n+1}^{*}\right). \end{cases} \tag{58}$$

La figura 40 muestra la implementación de la 3ra versión del sistema embebido propuesto, utilizando el sistema caótico de Chen discretizado (58); en la figura 40a muestra el plano de fase de x_n versus z_n, en la

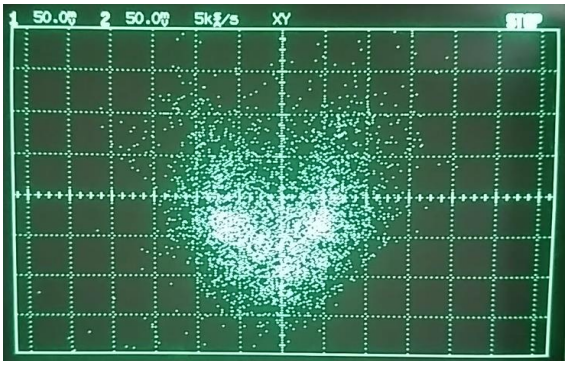
figura 40b muestra las trayectorias temporales de x_n y z_n considerando un $\tau = 0.001$ pequeño, en la figura 40c muestra el plano de fase de x_n versus z_n, en la figura 40d muestra las trayectorias temporales de x_n y z_n considerando un $\tau_{\max} = 0.017$ donde se conservan las dinámicas caóticas de acuerdo a la tabla 21. Se reproducen los cinco sistemas caóticos en 3-D (8)-(9) y (50)-(52) reproducidos en sus versiones discretas utilizando los algoritmos numéricos (45), (46), y (47)-(48) sobre el sistema embebido implementado en la V3, el resultado del desempeño utilizando el microcontrolador U9 se muestra en la tabla 21. Dada la características de gama alta y de la capacidad de procesamiento de U9, la memoria de programa permite reproducir sin inconvenientes los 5 sistemas caóticos propuestos utilizando el algoritmo numéricos propuestos (45)-(48).



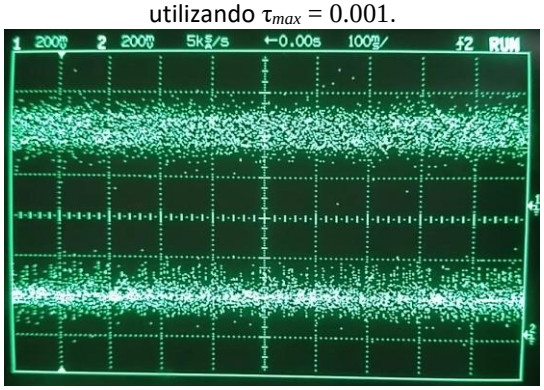
(a) Plano de fase x_n versus z_n.



(b) Evolución de las variables de estado x_n y z_n utilizando $\tau_{max} = 0.001$.



(c) Plano de fase x_n versus z_n.



(d) Evolución de las variables de estado x_n y z_n utilizando $\tau_{max} = 0.017$.

Figura 40. Implementación del sistema embebido en V3 del sistema caótico Chen (58) discretizado utilizando el algoritmo numérico (46).

Tabla 21. Desempeño de los cinco sistemas caóticos (8)-(9) y (50)-(52) discretizados utilizando los algoritmos numéricos (45)-(48), en V3.

Sistema caótico	Euler		Heun		RK4	
	T (µs)	f (ips)	T (µs)	f (ips)	T (µs)	f (ips)
Lorenz	30	33333	50	20000	87	11494
Rössler	27	37037	44	22727	78	12820
Chen	30	33333	50	20000	87	11494
Liu y Chen	31	32258	53	18867	94	10638
Lorenz	30	33333	50	20000	87	11494

8.1.4. Sistema embebido implementado en V4: U10 – FPGA

La cuarta versión del sistema embebido propuesto es implementada con el FPGA U10. Los algoritmos son simulados solo en la parte central del sistema embebido propuesta en V4, utilizando el compilador Nios II eclipse sobre U10 (Altera Corporation, 2011). En la implementación de hardware del sistema embebido en V4 se utiliza el reloj interno configurado a 150 MHz como oscilador de U10, $V_{cc} = V_{dd} = +3.5V$, $V_{ss} = 0$ V y los DACs U11-U13 conectados externamente. La figura 41 muestra el diagrama de bloques que soporta la síntesis que incluye por ejemplo el protocolo SPI y que es creada con la herramienta Qsys para crear el microcontrolador Nios II/f y utilizar la plantilla del proyecto "*hello_world.c*" del compilador Nios II Eclipse IDE C/C++ para implementar embebido en U10.

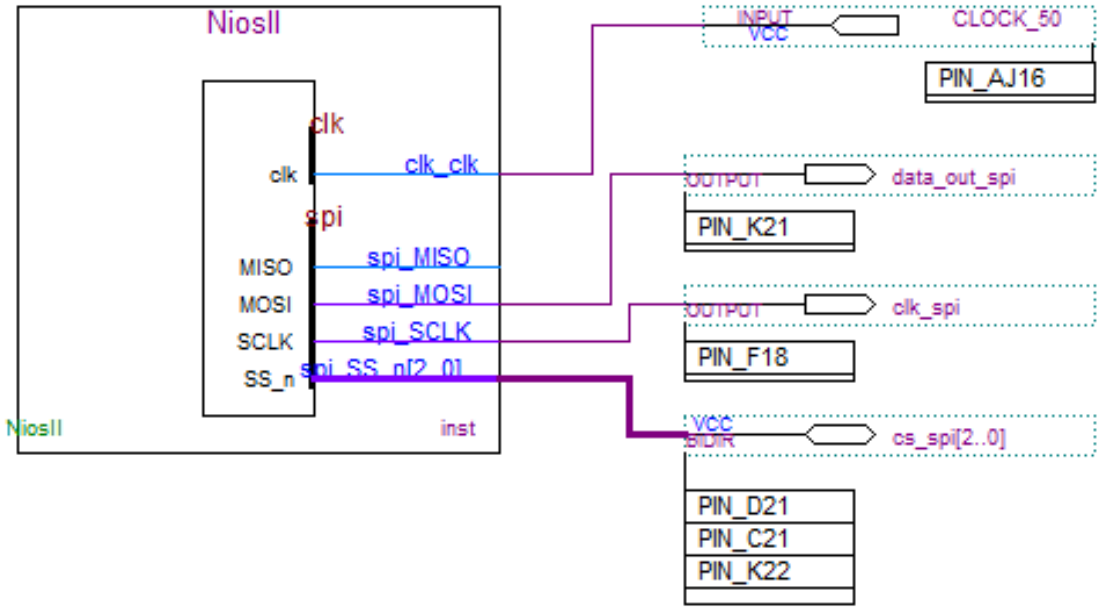


Figura 41. Diagrama de bloques y diseño esquemático del sistema embebido utilizando Quartus II para la V4.

Como ejemplo, se utiliza el algoritmo de RK4 (47)-(48) para obtener el MACM (9) en versión discreta que está dado por

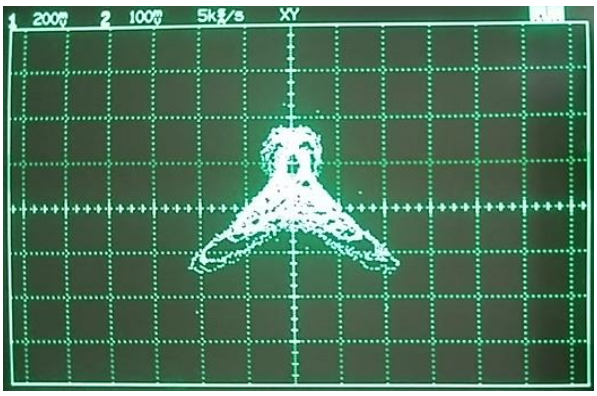
$$
\begin{cases}
k_1 = -ax_n - by_n z_n, \\
l_1 = -x_n + cy_n, \\
m_1 = d - y_n^{\,2} - z_n, \\
k_2 = -a\left(x_n + \dfrac{\tau}{2}k_1\right) - b\left(y_n + \dfrac{\tau}{2}l_1\right)\left(z_n + \dfrac{\tau}{2}m_1\right), \\
l_2 = -\left(x_n + \dfrac{\tau}{2}k_1\right) + c\left(y_n + \dfrac{\tau}{2}l_1\right), \\
m_2 = d - \left(y_n + \dfrac{\tau}{2}l_1\right)^2 - \left(z_n + \dfrac{\tau}{2}m_1\right), \\
k_3 = -a\left(x_n + \dfrac{\tau}{2}k_2\right) - b\left(y_n + \dfrac{\tau}{2}l_2\right)\left(z_n + \dfrac{\tau}{2}m_2\right), \\
l_3 = -\left(x_n + \dfrac{\tau}{2}k_2\right) + c\left(y_n + \dfrac{\tau}{2}l_2\right), \\
m_3 = d - \left(y_n + \dfrac{\tau}{2}l_2\right)^2 - \left(z_n + \dfrac{\tau}{2}m_2\right), \\
k_4 = -a\left(x_n + \tau k_3\right) - b\left(y_n + \tau l_3\right)\left(z_n + \tau m_3\right), \\
l_4 = -\left(x_n + \tau k_3\right) + c\left(y_n + \tau l_3\right), \\
m_4 = d - \left(y_n + \tau l_3\right)^2 - \left(z_n + \tau m_3\right).
\end{cases}
\tag{59}
$$

$$\begin{cases} x_{n+1} = x_n + \dfrac{\tau}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right), \\ y_{n+1} = y_n + \dfrac{\tau}{6}\left(l_1 + 2l_2 + 2l_3 + l_4\right), \\ z_{n+1} = z_n + \dfrac{\tau}{6}\left(m_1 + 2m_2 + 2m_3 + m_4\right). \end{cases} \tag{60}$$

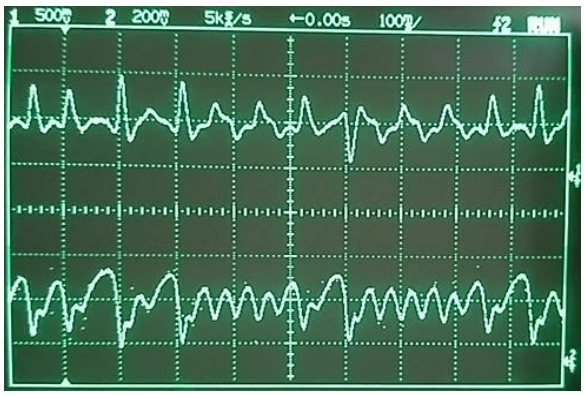
Se implementa la 4ta versión del sistema embebido propuesto del sistema MACM discretizado utilizando (59)-(60); en la figura 42a muestra el plano de fase de x_n versus z_n, en la figura 42b muestra las trayectorias de x_n y z_n considerando un $\tau = 0.001$ pequeño, en la figura 42c muestra el plano de fase de x_n versus z_n, en la figura 42d muestra las trayectorias de x_n y z_n considerando un $\tau_{\max} = 0.084$, donde se conservan las dinámicas caóticas de acuerdo a la tabla 17. Se reproducen los cinco sistemas caóticos en 3-D (8)-(9) y (50)-(52) discretizados utilizando los algoritmos numéricos (45), (46), y (47)-(48) sobre el sistema embebido implementado en V4, y en la tabla 22 se muestra el desempeño utilizando el FPGA U10. Dada la características de gama alta y de la capacidad de procesamiento de U10, la memoria de programa permite reproducir sin inconvenientes los 5 sistemas caóticos utilizando el algoritmo numéricos (45)-(48).

Tabla 22. Desempeño de los cinco sistemas caóticos (8)-(9) y (50)-(52) discretizados utilizando los algoritmos numéricos (45)-(48), en V4.

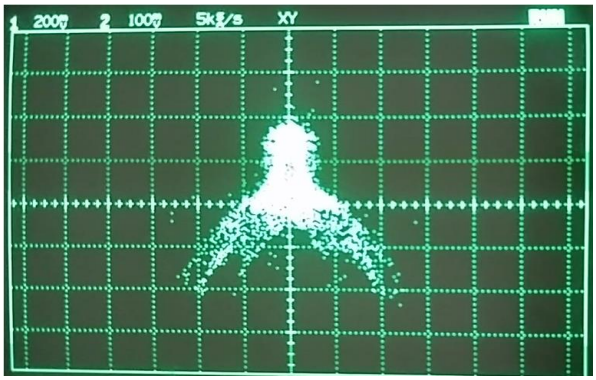
Sistema caótico	**Euler**		**Heun**		**RK4**	
	***T* (µs)**	***f*** (ips)	***T* (µs)**	***f*** (ips)	***T* (µs)**	***f*** (ips)
Lorenz	40	25000	70	14286	159	6289
Rössler	37	27027	53	18868	135	7407
Chen	40	25000	70	14286	159	6289
Liu y Chen	45	22222	83	12048	167	5988
Lorenz	40	25000	70	14286	159	6289



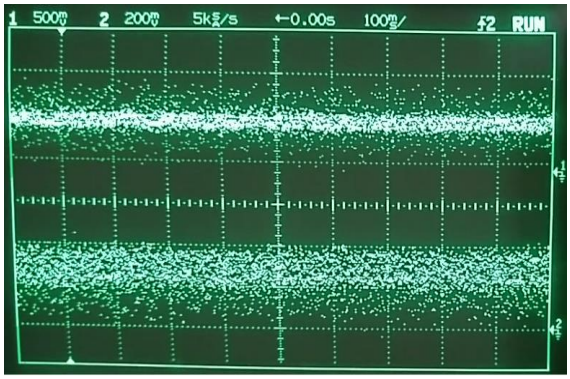
(a) Plano de fase x_n versus z_n.



(b) Evolución de las variables de estado x_n y z_n utilizando $\tau = 0.001$.



(c) Plano de fase x_n versus z_n.



(d) Evolución de las variables de estado x_n y z_n utilizando $\tau_{max} = 0.084$.

Figura 42. Implementación del sistema embebido en la V4 del MACM (59)-(60) discretizado utilizando el algoritmo numérico (47)-(48).

8.2. Desempeño del sistema embebido presentado en V1-V4

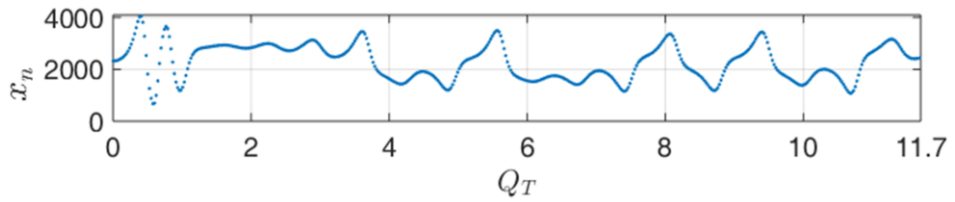
Con los resultados obtenidos de la implementación de los 5 sistemas caóticos en 3-D (8)-(9) y (50)-(52), con base en los algoritmos numéricos (45)-(48) sobre las V1-V4 del sistema embebido, se procede a reproducir las dinámicas caóticas utilizando la estimación propuesta en (55). Como ejemplo, se utilizan los resultados obtenidos en la tabla 19, la cual muestra el desempeño de los algoritmos de Euler (45) y Heun (46) reproducidos en V1 del sistema embebido, donde se obtuvo el tiempo de complejidad $T_{Q(U7)}$, el reciproco $f_{Q(U7)}$ estimado en ips, el tamaño de paso máximo τ_{max} y se determinan finalmente las unidades de tiempo total Q_T generadas en 1 segundo. En la tabla 23, se muestran los resultados de las unidades de tiempo máximas Q_T que se pueden estimar en simulación e implementación.

Tabla 23. Desempeño del sistema embebido en V1 expresado en Q_T, para la reproducción de los cinco sistemas caóticos (8)-(9) y (50)-(52) discretizados utilizando (45)-(46).

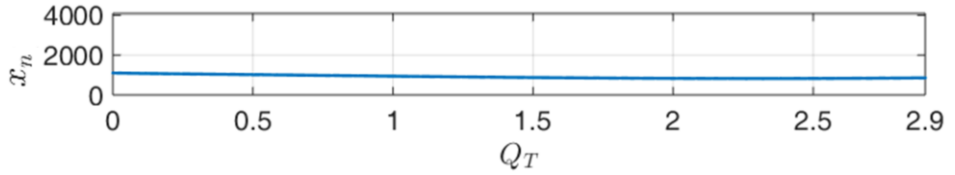
Sistema caótico	**Euler**			**Heun**		
	f (ips)	$\tau_{\max}$	Q_T	f (ips)	$\tau_{\max}$	Q_T
Lorenz	488.7	0.024	11.729	289.8	0.024	6.9552
Rössler	581.3	0.005	2.9065	362.3	0.005	1.8115
Chen	488.7	0.002	0.9774	289.8	0.002	0.5796
Liu y Chen	403.2	0.002	0.8064	226.2	0.002	0.4524
MACM	488.7	0.084	41.051	289.8	0.084	24.3432

Como ejemplo ilustrativo, en la figura 43 se muestran interesantes comportamientos dinámicos del estado x_n de los 5 sistemas caóticos (8)-(9) y (50)-(52) en versión discretizada utilizando (45), los cuales, se complementan con resultados obtenidos en las tablas 19 y 23 de la implementación del sistema embebido en V1. De acuerdo a los resultados obtenidos, os sistemas de Lorenz, Chen y el MACM utilizan el mismo t_c, pero la trayectoria x_n del sistema MACM es más compacta, en comparación por ejemplo al sistema de Rössler, el cual, presenta un mejor desempeño en su t_c, pero la dinámica de x_n es más lenta dado que su τ_{max} es pequeño. Por último, el sistema de Liu y Chen presenta un t_c más grande, su algebra es más lenta dado que presenta 3 no linealidades y la dinámica de su variable caótica x_n es compacta, en comparación a la variable caótica x_n del sistema MACM que muestra un comportamiento similar, este cuenta con solo dos no linealidades y su t_c muestra un mejor desempeño.

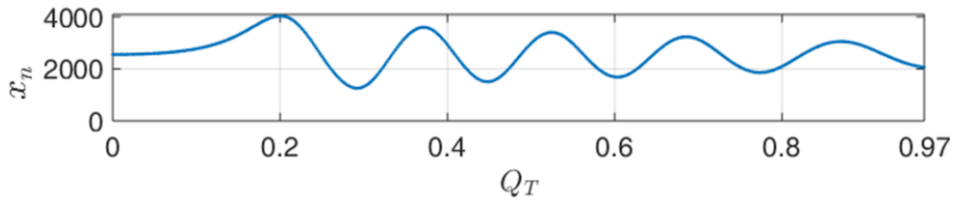
Generalizando, de la misma forma que los resultados obtenidos en la implementación de la primera versión del sistema embebido, se utiliza (55) para calcular el tamaño de paso máximo τ_{max} y las unidades de tiempo máximas Q_T que se obtienen al estimar la simulación e implementación de los 5 sistemas caóticos (8)-(9) y (50)-(52) y sus discretizaciones presentadas en las tablas 20-22 en versiones V2-V4 del sistema embebido, por último, el desempeño expresado en Q_T está resumido es las tablas 24, 25 y 26, respectivamente.



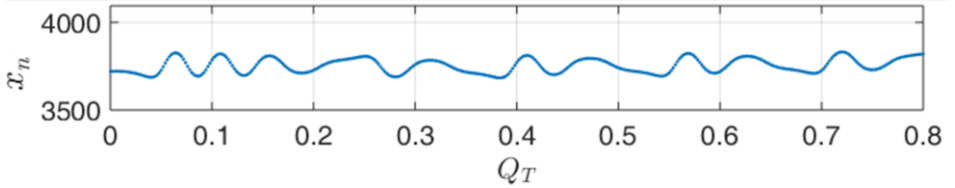
(a) Sistema caótico de Lorenz en su versión discretizada.



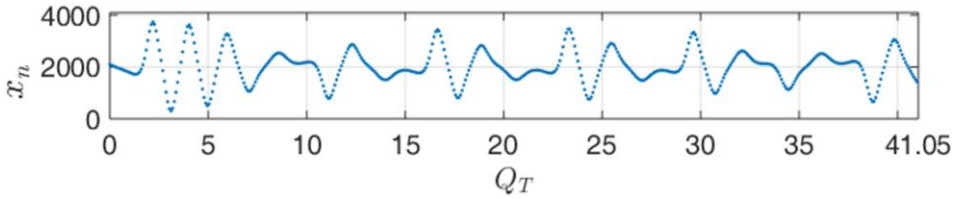
(b) Sistema caótico de Rössler en su versión discretizada.



(c) Sistema caótico de Chen en su versión discretizada.



(d) Sistema caótico de Liu Chen en su versión discretizada.



(e) Sistema caótico MACM en versión discretizada.

Figura 43. Gráficas temporales del estado caótico x_n expresadas en unidades de tiempo Q_T generadas en 1 segundo para comparar el desempeño de los 5 sistemas caóticos (8)-(9) y (50)-(52) en versión discretizada utilizando el método de Euler (45).

Tabla 24. Desempeño del sistema embebido en la V2 expresado en Q_T, para la reproducción de los cinco sistemas caóticos (8)-(9) y (50)-(52) discretizados utilizando (45)-(48).

Sistema caótico	**Euler**		**Heun**		**RK4**	
	τ_{max}	Q_T	τ_{max}	Q_T	τ_{max}	Q_T
Lorenz	0.024	285.7	0.068	492.7	0.106	432.7
Rössler	0.005	61.7	0.191	1446.8	0.251	1077
Chen	0.002	23.8	0.017	123.2	0.057	232.7
Liu y Chen	0.002	22.5	0.017	118.9	0.057	210.3
MACM	0.085	1011.9	0.228	1652.1	0.547	2232.9

Tabla 25. Desempeño del sistema embebido en la V3 expresado en QT, para la reproducción de los cinco sistemas caóticos (8)-(9) y (50)-(52) discretizados utilizando (45)-(48).

Sistema caótico	**Euler**		**Heun**		**RK4**	
	τ_{max}	Q_T	τ_{max}	Q_T	τ_{max}	Q_T
Lorenz	0.024	800	0.068	1360	0.106	1218.4
Rössler	0.005	185.2	0.191	4340.9	0.251	3217.8
Chen	0.002	66.7	0.017	340	0.057	655.2
Liu y Chen	0.002	64.5	0.017	320.7	0.057	606.4
MACM	0.085	2833.3	0.228	4560	0.547	6287.2

Tabla 26. Desempeño del sistema embebido en la V4 expresado en Q_T, para la reproducción de los cinco sistemas caóticos (8)-(9) y (50)-(52) discretizados utilizando (45)-(48).

Sistema caótico	**Euler**		**Heun**		**RK4**	
	τ_{max}	Q_T	τ_{max}	Q_T	τ_{max}	Q_T
Lorenz	0.024	600	0.068	971.4	0.106	666.6
Rössler	0.005	135	0.191	3603.8	0.251	1859.2
Chen	0.002	50	0.017	242.9	0.057	358.5
Liu y Chen	0.002	44	0.017	204.8	0.057	341.3
MACM	0.085	2125	0.228	3257.2	0.547	3440.1

8.3. Conclusiones del capítulo

En este capıtulo se reportaron los resultados experimentales obtenidos por la implementación del sistema embebido en las versiones PIC 8 bits, dsPIC 16 bits, PIC32 32 bits y FPGA Altera Nios II, utilizando los algoritmos numéricos de los 5 sistemas caóticos Lorenz, Rössler, Chen, Liu y Chen y MACM mediante

la discretización de Euler, Heun y RK4. Se presentaron las simulaciones e implementaciones de las trayectorias caóticas y planos de fase generados para los diferentes algoritmos numéricos de los sistemas caóticos propuestos este trabajo de tesis y se realizaron las pruebas descritas en los capítulos anteriores para determinar y reproducir la presencia de caos en cada uno de ellos. Los resultados obtenidos en las tablas 19-22 (representado en iteraciones por segundo) y en las tablas 23-26 (expresado en unidades de tiempo máximas) concluyen que los microcontroladores y FPGA utilizados permiten reproducir todos los algoritmos numéricos del conjunto de los 5 sistemas caóticos discretizados, la cuales, conservan la dinámica caótica, con la excepción del algoritmo RK4 que no es reproducible sobre el PIC de 8 bits ya que no soporta algoritmos numéricos grandes dado que su memoria de programa es pequeña. Se llegó a la conclusión que se cumplen los objetivos planteados utilizando los algoritmos numéricos y sistemas caóticos propuestos en este trabajo de tesis.

Capítulo 9. Conclusiones

En este capıtulo final, se presentan las conclusiones más importantes del presente trabajo de tesis realizado. Además, se hace mención a los trabajos futuros a partir de los resultados obtenidos en este estudio.

En este trabajo de tesis doctoral, se presentó la implementación de sistemas caóticos en sistemas embebidos. Se propuso el sistema caótico MACM en 3 dimensiones, el cual, mostró satisfactoriamente interesantes propiedades como dos parámetros de bifurcación, que es fácil de implementar electrónicamente en su versión continua mediante el uso de amplificadores operacionales y su versión discreta conserva su dinámica caótica utilizando un tamaño de paso más alto en comparación a otra familia de sistemas caóticos clásicos y actuales, los resultados concluyen que el sistema caótico MACM es muy atractivo para implementarse en microcontroladores y microprocesadores más lentos, por ejemplo, en microcontroladores de 8 bits, Arduino, PIC, Atmel, etc.

Se diseñó un sistema criptográfico digital basado en el mapa caótico de Hénon, el cual fue implementado en un sistema embebido conformado por microcontroladores dsPICs. El nuevo sistema criptográfico digital muestra satisfactoriamente un buen desempeño para aplicaciones de encriptado de audio utilizando el procesamiento de señales digitales donde la comunicación de hardware se realiza mediante el uso del protocolo de comunicación SPI en modo maestro-esclavo de 16 bits. El algoritmo criptograma presentó un buen desempeño en sus pruebas estadísticas, las cuales concluyen que el sistema criptográfico digital es seguro.

Como contribución de esta tesis, se realizaron pruebas numéricas con los algoritmos de Euler, Heun y RK4 para reproducir la versión discretizada de un conjunto de sistemas caóticos de Lorenz, Rössler, Chen, Liu y Chen y MACM para obtener la capacidad de procesamiento de un sistema embebido presentado en cuatro versiones, las primeras tres versiones abarcan tres familias de microcontroladores microchip de gama baja, media y alta implementadas con microcontroladores de 8 bits PIC, 16 bits dsPIC y 32 bits PIC32, respectivamente, y la cuarta versión se implementó utilizando un FPGA Altera, los resultados concluyen que el microcontrolador PIC32 de 32 bits muestra un mejor desempeño y sirve como base por ejemplo para obtener la mayor cantidad de iteraciones por segundo si se desea implementar como base de un sistema criptográfico digital.

Los resultados experimentales están en concordancia con los resultados numéricos, lo que implica que la metodología aplicada se encuentra bien encaminada.

Con los resultados numéricos y experimentales obtenidos y mostrados durante esta tesis, podemos concluir que los objetivos planteados se cumplieron en su totalidad y de manera exitosa.

9.1. Trabajo futuro

Este trabajo de tesis se enfocó en tres tópicos: la construcción de un nuevo sistema caótico en 3 dimensiones, un sistema criptográfico digital basado en caos, y un estudio de la conservación del caos en la versión discretizada de un conjunto de 5 sistemas caóticos utilizando los algoritmos numéricos de Euler, Heun y RK4, los cuales, fueron implementados en un sistema embebido compuesto por microcontroladores y FPGA. Sin embargo, existen muchas otras tareas por realizar y profundizar en los tópicos de este trabajo de tesis:

- Emplear el sistema caótico MACM para aplicaciones de sincronización, encriptación de información, entre otros.

- Emplear el sistema caótico MACM como base para encontrar su versión hipercaótica.

- Demostrar analíticamente una región acotada del sistema caótico MACM.

- Extender el uso del sistema criptográfico digital basado en caos para otros protocolos de comunicación digital como I^2C, I^2S, RS2-232, USB, SPI en modo multi-maestro y multi-esclavo en 32 bits, entre otros.

- Mejorar la implementación de los sistemas caóticos en versión discretazada estudiados en esta tesis en sistemas embebidos más robustos, los cuales, permitan un mejor desempeño para el procesamiento y encriptación de contenidos grandes de información digital, por ejemplo para encriptar video en formato standard 480p, alta definición 720p, full alta definición 1080p y ultra alta definición 4K.

- Emplear las pantallas inteligentes Mikromedia PIC32MX Plus y Mikromedia dsPIC33EP para la encriptación de información digital a tiempo real y ampliar su conectividad para uso de protocolos inalámbricos como Wi-Fi, Bluetooth, etc.

Literatura citada

Abundiz-Pérez, F., Cruz-Hernández, C., Murillo-Escobar, M. Á., López-Gutiérrez, R. y Arellano Delgado, A. (2016). A fingerprint image encryption scheme based on hyperchaotic Rössler map. *Mathematical Problems in Engineering*, **15**.

Aguilar-Bustos, A. Y., Cruz-Hernández, C., López Gutiérrez, R. M., Tlelo-Cuautle, E. y Posadas-Castillo, C. (2010). Hyperchaotic encryption for secure e-mail communication. *Emergent Web Intelligence: Advanced Information Retrieval*, 471–486.

Aihara, K. (2002). Chaos engineering and its application to parallel distributed processing with chaotic neural networks. *Proceedings of the IEEE*, **90**(5): 919–930.

Alligood, K. T., Sauer, T. D. y Yorke, J. A. (1996). *Chaos: An Introduction to Dynamical Systems*. Springer-Verlag, Berlin.

Altera Corporation (2011). Nios II Hardware Development. Tutorial. TU-N2HWDV-4.0.

Alvarez, G., y Li, S. (2006). Some basic cryptographic requirements for chaos-based cryptosystems. International Journal of Bifurcation and Chaos, **16**(8): 2129–2151.

Angulo, J. U. y Angulo I. (1999). *Microcontroladores PIC: diseño practico de aplicaciones*, 2ª Ed. S.A. Mcgraw-Hill / Interamericana de España.

Anishchenko, V. S., Kapitaniak, T., Safonova M. A. y Sosnovzeva, O. V. (1994). Birth of double-double scroll attractor in coupled Chua circuits, *Phys. Lett. A*, **192**: 207214.

Arellano-Delgado A., López-Gutiérrez R. M., Murillo-Escobar M., Cardoza-Avendaño L., y Cruz-Hernández C. (2017). The Emergence of Hyperchaos and Synchronization in Networks with Discrete Periodic Oscillators. *Entropy*, **19**: 413.

Arellano-Delgado, A., López-Gutiérrez, R., Cruz-Hernández, C., Posadas-Castillo, C., Cardoza-Avendaño, L. y Serrano-Guerrero, H. (2013). Experimental network synchronization via plastic optical fiber. Optical Fiber Technology, **19**(93): 93–98.

Azar, A. T., Volos, C., Gerodimos, N., Pham, V-T., Radwan, A. G., Vaidyanathan, S., Ouannas, A. y Munoz-Pacheco, J. M. (2017). A novel chaotic system without equilibrium: dynamics, synchronization, and circuit realization. *Complexity*, **2017**: 11.

Azzaz, M. S., Tanougast, C., Sadoudi, S., Fellah, R. y Dandache, A. (2013). A new auto-switched chaotic system and its FPGA implementation. *Communications in Nonlinear Science and Numerical Simulation*, **18**(7): 1792–1804.

Barnsley, M. F. (2014). *Fractals everywhere*. Academic Press.

BDTI Industry reports (2006). *Focus Report: FPGAs for DSP*. (2nd. ed.).

Bertuglia, C. S. y Vaio, F. (2005). *Nonlinearity, chaos, and complexity: the dynamics of natural and social systems.* Oxford University Press on Demand.

Boccaletti, S., Kurths J., Osipov, G., Valladares, D. L. y Zhou, C. S. (2002). The synchronization of chaotic systems. *Phys Rep*, **366**:1–101.

Briggs, K. (1990). An Improved Method for Estimating Lyapunov Exponents of Chaotic Time Series. *Phys. Let. A*., **151**: 27–32.

Brucoli, M., Carnimeo, L. y Grassi, G. (1996). A method for the synchronization of hyperchaotic circuits. *Int. J. Bifurc. Chaos*, **6**(9): 1673–1681.

Buscarino, A., Fortuna, L., Frasca, M. y Sciuto, G. (2014). *A Concise Guide to Chaotic Electronic Circuits.* Springer, Berlin.

Cardoza-Avendaño, L., López-Gutiérrez, R. M., Cruz-Hernández, C., Spirine, V., Chávez-Pérez, R. A. y Arellano-Delgado, A. (2012). Encrypted audio transmission via synchronized chaotic Nd:YAG lasers. *Revista Mexicana de Física*, **58**(6): 472–480.

Cardoza-Avendaño, L., Spirine, V., López-Gutiérrez, R. M., López-Mercado, C. A. y Cruz-Hernández, C. (2011). Experimental characterization of DFB and FP chaotic lasers with strong incoherent optical feedback. *Optics and Laser Technology*, **43**(5): 949–955.

Cates, M. E., Head, D. A. y Ajdari, A. (2002). Rheological chaos in a scalar shear-thickening model. *Physical review E*, **66**: 025202(R).

Cetina, (2017). Diseño de trayectorias caóticas en robots móviles. Tesis de Maestría en Ciencias en Electrónica y Telecomunicaciones con orientación en Instrumentación y Control. *CICESE*, 2017.

Chen, G, y Lai, D. (1996). Feedback control of Lyapunov exponents for discrete-time dynamical systems. *Int J Bifurc Chaos*, **6**: 1341–1349.

Chen, G. (1997). Control and anticontrol of chaos. *IEEE 1st International Conference, Control of Oscillations and Chaos Proceedings*, **2**: 181–186.

Chen, G. y Dong, X. (1998). *From Chaos to Order–Perspectives, Methodologies, and Applications*. World Scientific Pub. Co. Singapore.

Chen, G. y Lai, D. (1998). Anticontrol of chaos via feedback. *Int J Bifur Chaos*, **8**: 1585–1590.

Chen, G. y Ueta, T. (1999). Yet another chaotic attractor. *Int. J. Bifur Chaos*, **9**: 1465–1466.

Chen, H. K. y Lee, C. (2004). Anti-control of chaos in rigid body motion. *Chaos, Solutions and Fracta*ls, **21**: 957–965.

Chua, L. O., Komuro, M. y Matsumoto, T. (1986). The doubles scroll family. *IEEE Trans. Circuits Syst.*, **33**: 1073–1118.

Cong, L. y Xiaofu, W. (2001). Design and realization of an FPGA-based generator for chaotic frequency hopping sequences. IEEE Trans. Circuits Syst. I, 48(5): 521–532.

Cornish-Bowden, A., (1985). Nomenclature for incompletely specified bases in nucleic acid sequences: recommendations 1984. *Nucleic Acids Res*., **13**(9): 3021-3030.

Cramer, J. A. y Booksh, K. S. (2006), Chaos theory in chemistry and chemometrics: a review. *J. Chemometrics*, **20**: 447–454.

Cruz-Hernández, C., López-Gutiérrez, R. M., Aguilar Bustos, A. Y. y Posadas-Castillo, C. (2010). Communicating encrypted information based on synchronized hyperchaotic maps. *International Journal of Nonlinear Sciences and Numerical Simulation*, **11**(5): 337–349.

Cruz-Hernández, C. (2004). Synchronization of time-delay Chua's oscillator with application to secure communication. Nonlinear Dynamics and Systems Theory, **4**(1): 1–13.

Cruz-Hernández, C. y Nijmeijer, H. (2000). Synchronization through filtering. *International Journal of Bifurcation and Chaos*, **10**(4): 763–775.

Cuomo, K. M., Oppenheim, A. V. y Strogatz, S. H. (1993). Synchronization of Lorenz based Chaotic. Circuits with Applications to Communications. *IEEE Transactions On Circuits And Systems II. Analog And Digital Signal Processing*, **40** (10).

Devaney, R., Siegel, P. y Mallinckrodt, A. J. (1992). A first course in chaotic dynamical systems: Theory and experiment. *Computers in Physics*, **7**: 416.

Di Jasio, L. (2007). *Programming 16-Bit Pic Microcontrollers in C: Learning to Fly the Pic 24.* Newnes.

Di Jasio, L. (2008). *Programming 32-bit Microcontrollers in C. Exploring the PIC32*. Newnes.

Diacu F. y Holmes P. (1996). *Celestial Encounters: The Origins of Chaos and Stability*. Princeton University Press, Princeton, NJ.

Díaz-Brecia, S. (2009). Primeras reflexiones de los fractales, la teoría del caos y su aplicación en el aula. *Revista Red Visual*, 9–10.

Dormand, J. R. y Prince, P. J. (1980). A family of embedded Runge-Kutta formulae. *J. Comp. Appl. Math*., **6**: 19–26.

DSP Builder for Intel FPGAs (2017). Advanced Blockset Handbook. HB_DSPB_ADV.

Dyson F. (1988). *Infinite in All Directions*. Harper and Row, New York.

EI Naschie, M. S. (1996). Introduction to chaos, information and diffusion in quantum physics. *Chaos, Solitons & Fractals*, **7**: 7–10.

Enns, R. y McGuire, G. (1997). *Nonlinear Physics with Maple for Scientists and Engineers*. Birkhäuser, Boston.

Farmer, D. (1982). Chaotic Attractors Of An Infinite-Dimensional Dynamical System, *Physica 4D*, :366–393.

Feigenbaum, M. (1978). Quantitative Universality for a Class of Non-Linear Transformations. *J. Stat. Phys.*, **19**(25).

Feigenbaum, M. (1980). The Metric Universal Properties of Period Doubling Bifurcations and the Spectrum for a Route to Turbulence. *Ann. New York. Acad. Sci*., **357**: 330-336.

Fermilab. (2017). Fermi National Accelerator Laboratory Managed by Fermi Research Alliance, LLC for the U.S. Department of Energy Office of Science.

Figueroa-Torres, C. A., Medina-Monroy, J. L., Lobato-Morales, H., Chávez-Pérez, R. A. y Calvillo-Téllez, A. (2017). A novel fractal antenna based on the Sierpinski structure for super wide-band applications. *Microwave and Optical Technology Letters*, **59**(5): 1148–1153.

Figueroa-Torres, C. A., Medina-Monroy, J. L., Lobato-Morales, H., Chávez-Pérez, R. A. y Calvillo-Téllez, A. (2016). A Microstrip Antenna Based on a Standing-Wave Fractal Geometry for CubeSat Applications. *Microwave and Optical Technology Letters*, **58**(9): 2210–2214.

Fu, X., Chen, X., Hou, Q., Wang, Z. y Yin, Y. (2014). An Improved Chaos Genetic Algorithm for T-Shaped MIMO Radar Antenna Array Optimization, *International Journal of Antennas and Propagation*, **2014**: 1–6.

Fuster, A. (2004). *Técnicas criptográficas de protección de datos*. Ra-ma.

Gámez-Guzmán, L. M., Cruz-Hernández, C., López Gutiérrez, R. M. y García-Guerrero, E. E. (2009). Synchronization of Chuas circuits with multi-scroll atractors: application to communication. *Communications in Nonlinear Science and Numerical Simulation*, **14**(6): 2765–2775.

Glenn, E. (2002). *Fractal Dimension of Broccoli*. The Physics Factbook.

Gottwald, G. A. y Melbourne, I. (2004). A new test for chaos in deterministic systems. Proceedings of the Royal Society of London A. *Mathematical, Physical and Engineering Sciences. The Royal Society*, **460**: 603–611.

Guariglia, E. (2016). Entropy and Fractal Antennas. *Entropy* **18**(3).

Guerrero, G. (2000). Determinismo, modelos y modalidades. *Universidad Complutense de Madrid*, **24**.

Guglielmi, V., Pinel, P., Fournier-Prunaret, D. y Taha, A-K. (2009). Chaos-based cryptosystem on DSP. *Chaos, Solitons and Fract*als, **42**(4): 2135–2144.

Hausdorff, F. (1919). Dimension und ausseres mass. *Mathematische Annalen*, **79**(1-2): 157–179.

Hénon, M. (1976). A Two-Dimensional Mapping with a Strange Attractor. *Commun. Math. Phys*., **50**(1): 69–77.

Huang, Y., Zhang, P. y Zhao, W. (2015). Novel Grid Multiwing Butterfly Chaotic Attractors and Their Circuit Design. *IEEE Trans. Circuits Syst. II: Exp. Briefs*, **62**(5): 496–500.

Hübler, A. (1989). Adaptive control of chaotic systems, *Helv Phys Acta,* **62**: 343–347.

IEEE standard for floating-point arithmetic-redline (2008). IEEE Std 754-2008 (Revision of IEEE Std 754-1985)–Redline, :1–82.

Ikeda, K. (1979). Multiple valued Stationary State and its Instability of the Transmitted Light by a Ring Cavity System. *Opt. Commun*, **30**: 257–261.

Juang, Ch., Hwang, T. M., Juang, J. y Lin, W-W. (2000). A Synchronization Scheme Using Self-Pulsating Laser Diodes in Optical Chaotic Communication, IEEE *Journal Of Quantum Electronics*, **36**(3).

Kapitaniak, T. (1992). Controlling chaotic oscillations without feedback. *Chaos, Solitons & Fractals*, **2**: 519–530.

Kapitaniak, T. (1995). Continuous control and synchronization in chaotic systems. *Chaos, Solitons & Fractals*, **6**: 237–244.

Kennedy, M. P. (1995). On the Relationship between the Chaotic Colppits Oscillator and Chua's Oscillator. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, **42**(6): 376–379.

Koyuncu, I., Ozcerit, A.T. y Pehlivan, I. (2014). Implementation of FPGA-based real time novel chaotic oscillator. *Nonlinear Dyn. 77*, **1**(2): 49–59.

Labcenter Electronics Ltd. (2017). Proteus Design Suite. Getting Started Guide. Tutorial

Lakshmanan M. y Murali K. (1996). *Chaos in nonlinear oscillators: controlling and synchronization*. World Scientific Singapore.

Leung, H. y Lo, T. (1993). Chaotic radar signal processing over the sea, *IEEE Journal of Oceanic Engineering*, **18**(3): 287–295.

Li, T. Y. y Yorke, J. A. (1975). Period three implies chaos. *Am Math Mon*, **82**: 985–992.

Liao, X., Wong, K.-W., Leung, C.-S. y Wu, Z. (2001). Hopf bifurcation and chaos in a single delayed neuron equation with non-monotonic activation function. *Chaos, Solitons & Fractals*, **12**(8): 1535–1547.

Li-li, L., Ying, L. y Qi-guo, Y. (2014). Robust synchronization of chaotic systems using sliding mode and feed-back control. *Journal of Zhejiang University SCIENCE C*, **15**(3): 211–222.

Lin, F.-Y. y Liu, J.-M. (2004). Chaotic radar using nonlinear laser dynamics. *IEEE Journal of Quantum Electronics*, **40**(6): 815–820.

Liu, Ch., Liu, T., Liu, L., y Liu, K. (2005). A new chaotic attractor. *Chaos, Solitons & Fractals*, **22**(5): 1031–1038.

Liu, W. B. y Chen, G. (2002). A new chaotic system and its generation. *Int J Bifur Chaos*, **12**: 261–267.

Liu, W., Sun, K. y Zhu, C. (2016). A fast image encryption algorithm based on chaotic map. *Optics and Lasers in Engineering*, **84**: 26–36.

Liu, Z., Zhu, X., Hu, W. y Jiang, F. (2007). Principles of chaotic signal radar. *International Journal of Bifurcation and Chaos*, **17**(5): 1735–1739.

López-Gutiérrez, R. M., Posadas-Castillo, C., López-Mancilla, D. y Cruz-Hernández, C. (2009). Communicating via robust synchronization of chaotic lasers. *Chaos, Solitons & Fractals*, **42**(1): 277–285.

López-Mancilla, D. y Cruz-Hernández, C. (2004). An analysis of robustness on the synchronization of chaotic systems under nonvanishing perturbations using sliding modes. *WSEAS Transactions on Mathematics*, **3**(2): 364-369.

Lorenz, E. N. (1963).Deterministic non periodic flow. Journal of the Atmospheric Sciences, **20**(2): 130–141.

Lü, J. y Chen, G. (2002). A new chaotic attractor coined. *Int. J. Bifurcation Chaos*, **12**(3): 659–661.

Mandelbrot, B. (1983). *The Fractal Geometry of Nature*. Freeman, Paris.

Marinkovic, S. J. y Popovici, E. M. (2011). Nano-Power wire-less wake-up receiver with Serial Peripheral Interface. *IEEE Journal on Selected Areas in Communications*, **29**(8): 1641–1647.

Martins-Filho, L. S. y Macau, E. E. (2007). Patrol mobile robots and chaotic trajectories. *Mathematical Problems in Engineering*.

Martins-Filho, L. S., Macau, E. E., Rocha, R., Machado, R. F. y Hirano, L. A. (2005). Kinematic control of mobile robots to produce chaotic trajectories. *En: Proc. of the 18th Int. Congress of Mechanical Engineering, Ouro Preto*.

Martins-Filho, L. S., Machado, R. F., Rocha, R. y Vale, V. (2004). Commanding mobile robots with chaos. *En: ABCM Symposium Series in Mechatronics*, **1**: 40–46.

May, R. M. (1976). Simple mathematical models with very complicated dynamics. *Nature*, **261**(5560): 459–467.

Melnikov, V. K. (1963). On the stability of the center for time periodic perturbations. *Trans. Moscow Math Soc.*, **12**: 1–57.

Méndez-Ramírez, R., Arellano-Delgado, A., Cruz-Hernández, C., Abundiz-Pérez, F. y Martínez-Clark, R. (2017). Chaotic Digital Cryptosystem by using SPI Protocol and its dsPICs Implementation. *Frontiers of Information Technology & Electronic Engineering*, (ZUSC-D-16-01346). Artículo aceptado, en impresión.

Méndez-Ramírez, R., Cruz-Hernández, C., Arellano-Delgado, A. y López-Gutiérrez, R. M. (2016). Degradation Analysis of Generalized Chua's Circuit Generator of Multi-Scroll Chaotic Attractors and its Implementation on PIC32. *IEEE Future Technologies Conference 2016 (FTC 2016), San Fco, CA; USA. Diciembre 6-7, 2016.*

Méndez-Ramírez, R., Cruz-Hernández, C., Arellano-Delgado, A. y Martínez-Clark, R. (2017). A new simple chaotic Lorenz-type system and its digital realization using a TFT touch-screen display embedded system. *Complexity,* **2017**(6820492): 1–13.

Méndez-Ramírez, R., Cruz-Hernández, C., Are-llano-Delgado, A., Cardoza-Avendaño, L., López-Gutiérrez, R.M. y Aranda-Bricaire, E. (2015). Implementación del circuito hipercaótico de Chua en un sistema embebido de bajo costo. *Congreso Nacional de Control Automático, AMCA 2015, Cuernavaca, Morelos, del 14 al 16 de Octubre, México*, :171–176.

Menezes, A. J., Van-Oorschot, P. C. y Vanstone, S. A. (1996). *Handbook of Applied Cryptography*, CRC Press.

Microchip Technology Inc. (1997). AN575 IEEE 754 Compliant Floating Point Routines. DS00575B.

Microchip Technology Inc. (2004). dsPIC30F3014, dsPIC30F4013 Data Sheet. DS70138C.

Microchip Technology Inc. (2004). MCP4921/4922. 12-Bit DAC with SPI Interface. DS21897.

Mikroelektronika d.o.o. (2017). Software MikroC Pro for dsPIC.

Mikroelektronika d.o.o. (2017). Software MikroC Pro for PIC.

Mikroelektronika d.o.o. (2017). Software MikroC Pro for PIC32.

Mohamed, S. A., Camel, T., Sadoudi, S., Fellan, R. y Dandache, A. (2013). A new auto-switched chaotic system and its FPGA implementation. *Communications in Nonlinear Science and Numerical Simulation*, **18**(7): 1792–1804.

Moon, F. C. (1992). *Chaotic and Fractal Dynamics: An Introduction for Applied Scientists and Engineers*. John Wiley & Sons, Icn.

Motorola, Inc. (2003). SPI Block Guide V03.06. S12SPIV3/D.

Muhaya, F. B., Usama, M. y Khan, M. K. (2009). Modified AES using chaotic key generator for satellite imagery encryption. *Emerg. Intell. Comput. Technol. Appl*., **5754**: 1014–1024.

Murillo-Escobar, M. Á., Cruz-Hernández, C., Cradoza-Avendaño, L. y Méndez-Ramírez, R. D. (2017). A novel pseudorandom number generator based on pseudorandomly enhanced logistic map. *Nonlinear Dynamics*, **87**(1): 407–425.

Murillo-Escobar, M., Cruz-Hernández, C., Abundiz-Pérez, F. y López-Gutiérrez, R. M. (2015). A robust embedded biometric authentication system based on fingerprint and chaotic encryption. *Expert Systems with Applications*, **42**(21): 8198–8211.

Murillo-Escobar, M., Cruz-Hernández, C., Abundiz-Pérez, F., López-Gutiérrez, R. y Acosta del Campo, O. (2015). A RGB image encryption algorithm based on total plain image characteristics and chaos. *Signal Processing*, **109**: 109–131.

Muthuswamy, B., y Banerjee, S. (2014). *A Route To Chaos Using FPGAs, Volume I: Experimental Observations*. Springer.

Nik, H. S. y Golchaman, M. (2014). Chaos Control of a Bounded 4D Chaotic System, *Neural Comput Applic*, **25**(3): 683–692.

Nucleic Acids Res. (2013). *Database issue*, **41**: D36-D42.

Nyquist, H. (1928). Certain Topics in Telegraph Transmission Theory. *AIEE Trans*., **47**: 617–644.

Ott, E., Grebogi C. y Yorke J.A. (1990). Controlling chaos. *Phys Rev Lett*, **64**: 1196–1199.

Parker, T. y Chua, L. (1989). *Practical Numerical Algorithms for Chaotic Systems*, Springer-Verlag New York.

Pecora, L. M. y Carroll T. L. (1990). Synchronization in chaotic systems. *Phys Rev Lett*, **64**: 821–824.

Peitgen, H. -O., Jurgens, H. y Saupe, D. (2006). Chaos and fractals: new frontiers of science. *Springer Science & Business Media*.

Petras, I. (2011). *Fractional-Order Nonlinear Systems*, Springer.

Pham, V. T., Jafari, S., Volos, Ch., Giakoumis, A., Vaidyanathan, S. y Kapitaniak, T. (2016). A chaotic system with equilibria located on the rounded square loop and its circuit implementation. *IEEE Trans. Circuits Syst. II: Exp. Briefs*, **63**(9): 878–882.

Pinales, F. J. y Velázquez, C. E. (2014). *Problemario de Algoritmos Resueltos con Diagramas de Flujo y Pseudocódigo*. Universidad Autónoma de Aguas Calientes.

Poincaré, H. (1899). *Les méthodes nouvelles de la méchanique céleste*. Gauthier-Villars, Paris.

Poincaré, H. (1900). La théorie de Lorentz et le principe de réaction, Archives néerlandaises des sciences exactes et naturelles, **5**(2): 252–278.

Posadas-Castillo, C., Cruz-Hernández, C. y López-Gutiérrez, R. M. (2007). Experimental realization of synchronization in complex networks with Chua's circuits like nodes. *Chaos, Solitons & Fractals*, **40**(4): 1963-1975.

Rhouma, R. y Belghith, S. (2011). Cryptanalysis of a chaos-based cryptosystem on DSP. *Commun Nonlinear Sci Numer Simulat*, **16**(2): 876–884.

Robinson, R. C. (2004). *An Introduction to Dynamical Systems: Continuous and Discrete*. Prentice Hall, New York.

Rössler, O. E. (1976). An equation for continuous chaos. *Physics Letters A*, **57**: 397–398.

Ruelle, D. y Takens, F. (1971). On the nature of turbulence. *Commun Math Phys*, **20**: 167–192.

Russell, F. (1953). *On the Notion of Cause.* Feigl, H. and Brodbeck, M. Readings in the Philosophy of Science.

Schuster, H. G. y Just, W. (2005). *Deterministic Chaos.* Wiley-VCH, Weinheim.

Serrano-Guerrero, H., Cruz-Hernández, C., López-Gutiérrez, R. M., Posadas-Castillo, C. y Inzunza González, E. (2010). Chaotic synchronization in star coupled networks of three dimensional cellular neural networks and its application in communications. *International Journal of Nonlinear Sciences and Numerical Simulation*, **11**(8): 571–580.

Shampine, L. F. y Reichelt, M. W. (1997). The MATLAB ODE Suite. *SIAM Journal on Scientific Computing*, **18**: 1–22.

Shannon, C. (1949). Communication in the Presence of Noise. *Proceedings IRE*, **37**: 10–21.

Sharkovskii, A. N. (1964). Coexistence of cycles of a continuous map of a line into itself. *J Ukr Math*, **16**: 61–71.

Siddiqui, R. A., Grosvenor, R. I. y Prickett, P. W. (2015). DSPIC based advanced data acquisition system for Monitoring, Control and Security Applications. *12th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, : 293–298.

Sipser, M. (2006). *Introduction to the Theory of Computation*. Second Edition. Thomson Course Technology.

Sira-Ramírez, H., y Cruz-Hernández, C. (2001). Synchronization of chaotic systems: A generalized Hamiltonian Systems Approach. *International Journal of Bifurcation and Chaos*, **11**(5): 1381–1395.

Sprott, J. C. (1994). Some simple chaotic flows. *Physical Review E*, **50**(2): R647–R650.

Sprott, J. C. (1997). Some Simple Chaotic Jerk Functions. *American Journal of Physics*, **65**: 537–543.

Sprott, J. C. (2010). *Elegant Chaos: Algebraically Simple Chaotic Flows.* World Scientific, Singapore.

Stefanski, A. y Kapitaniak, T. (2003). Synchronization of two chaotic oscillators via a negative feedback mechanism. *Chaos, Solitons & Fractals*, **40**: 5175–5185.

Stockman, H. J. y Ferry. D. K. (2006). Chaos in Microwave resonators, *Séminaire Poincaré IX*, **1**.

Strogatz, S. H. (1994). *Nonlinear Dynamics and Chaos: With Applications To Physics, Biology, Chemistry, And Engineering. Massachusetts*. US: Perseus Books.

Terasic Inc. (2017). DE2i-150 FPGA Development Kit, FPGA System. User Manual.

The Mathworks, Inc. (2017). Matlab and Simulink Standard Suite R2017a. Software.

Tlelo-Cuautle, E., Rangel-Magdaleno, J. y de la Fraga, L. G. (2016). *Engineering Applications of FPGAs. Chaotic Systems, Artificial Neural Networks, Random Number Generators, and Secure Communication Systems*. Springer.

Tlelo-Cuautle, E., Rangel-Magdalenoa, J., Pano-Azucenaa, A., Obeso-Rodelob, P. y Nunez-Pérez, J. C. (2015). FPGA realization of multi-scroll chaotic oscillators. Communications in Nonlinear Science and Numerical Simulation, **27**(1–3): 66–80.

Torres, N. (2005). Caos en sistemas biológicos. *Matematicalia: Revista Digital de Divulgación Matemática de la Real Sociedad Matemática Española,* **1**(3).

Tucker, W. (2002). A Rigorous ODE Solver and Smale's. *14th Problem. Found. Comput. Math*, **2**: 53–117.

Uchida, A. (2012). *Optical Communication with Chaotic Lasers: Applications of Nonlinear Dynamics and Synchronization*. Wiley-VCH Verlag GmbH & Co. KGaA.

Uriz, A. J., Agüero, P. D., Moreira, J., Hidalgo, R., Gonzalez, E. y Tulli, J. (2016). Flexible pseudorandom number generator for tinnitus treatment implemented on a dsPIC. *IEEE Latin America Transactions*, **14**(1): 72–77.

Volos, C. K., Kyprianidis, I. M. y Stouboulos, I. N. (2012). A chaotic path planning generator for autonomous mobile robots. *Robotics and Autonomous Systems*, **60**(4): 651–656.

Wang, Q., Yu, S., Li Ch., Lü, J., Fang, X., Guyeux, Ch. y Bahi, J. M. (2016). Theoretical design and FPGA-based implementation of higher-dimensional digital chaotic systems. *IEEE Transactions on Circuits and Systems. I. Regular Papers*, **63**(3): 401–412.

Wang, X. y Luo, C. (2006). Researches on chaos phenomenon of EEG dynamics model. *Appl. Math. Compu*t., **183**(1): 30–41.

Wang, Z. L. y Zhang H. G. (2002). Synchronization of high dimensional chaotic systems based on nonlinear feedback control. *In: Proc 15th IFAC Conf, Barcelona, Spain*, :347–351.

Wolf, A., Swift, J. B., Swinney, H. L. y Vastano, J. A. (1985). Determining Lyapunov Exponents from a Time Series. *Physica D.*, **16**: 285–317.

Wu, S., Moore, B. E. y Shah, M. (2010). Chaotic invariants of Lagrangian particle trajectories for anomaly detection in crowded scenes. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, : 2054–2060.

Xiaohong, Z. y Zhiguang, Z. (2013). Hyper-chaotic LÜ System Simulation Design of Digital Circuit Based on DSP Builder. *TELKOMNIKA*, **11**(5): 2679–2691.

Yalcin, M. E., Suykens J. A. K. y Vandewalle J. (2004). True random bit generation from a double-scroll attractor. *IEEE Trans. Circ. Syst. I Regular Papers*, **51**(7): 1395–1404.

Yang, Q. y Chen, G. (2008). A chaotic system with one saddle and two stable node-foci, *International Journal of Bifurcation and Chaos in Applied Sciences and Engineering*, **18**(5): 1393–1414.

Yang, W., Cao, W., Chung, T-S. y Morris, J. (2005). *Applied numerical methods using Matlab*. John Wiley and Sons, Inc.

Zamorano, A. A. (2012). Theories Of Chaos And Linguistics: A Caological Approach To Verbal Human Communication UNED. *Revista Signa*, **21**: 679-705

Zhang, H., Liu D. y Wang Z. (2009). *Controlling Chaos Suppression, Synchronization and Chaotification*. Springer.

Zhou, N., Pan, S., Cheng, S. y Zhou, Z. (2016). Image compression–encryption scheme based on hyper-chaotic system and 2D compressive sensing. *Optics & Laser Technology*, **82**(2016): 121–133.

Apéndice A. Matriz de secuencias de ADN

La matriz de secuencia de ADN se utilizó en el proceso de confusión (Sección 4.2.2), y el proceso de confusión inversa (Sección 4.2.1). Las secuencias de ADN se pueden obtener de las funciones de Matlab getgenbank y randseq (N) o podemos solicitar directamente la información de acceso a genbank (Nucleic Acids Res., 2013). En este trabajo, la matriz de ADN se realizó mediante el uso de las funciones Matlab. La Tabla A1 muestra una secuencia de cuatro nucleótidos denominada Ad, Cy, Gu y Th con el fin de diferenciar los vectores y variables usados en las secciones anteriores para representar la base de una cadena de ADN (Cornish-Bowden, 1985).

Tabla A1. Representación binaria de nucleótidos.

Nucleotide	Description	Binary number
Ad	Adenine	$(00)_2$
Cy	Cytosine	$(01)_2$
Gu	Guanine	$(10)_2$
Th	Thymine	$(11)_2$

Para el diseño de los elementos de la matriz de ADN, se consideraron pares de nucleótidos para representar las posiciones determinadas por k. Por ejemplo al construir un par de nucleótidos considerando Ad (adenina) y Th (timina) se obtiene la posición binaria $(0011)_2$, esto significa que el par $AdTh$ representa la posición 3. Los elementos determinan posiciones de 1 al valor mayor de k que es 16, porque el protocolo SPI está configurado para 16 bits. Estos elementos están desordenados, y no pueden repetir.

Los arreglos numéricos de 16 pares de ADN, representan una secuencia para la matriz de ADN propuesta. Se formaron 24 secuencias de ADN definidas r = 24 filas y s = 32 columnas. Las secuencias se repiten dos veces de acuerdo con el diseño de los algoritmos para encriptar y desencriptar. Los resultados de las secuencias de ADN están contenidas en las matrices $\boldsymbol{D}$ y $\boldsymbol{G}$.

Inicialmente, los datos tienen 16 elementos contenidos en los vectores $\boldsymbol{C}$ y $\boldsymbol{F}$. Los DHM caóticos (2) y (16) definen las coordenadas iniciales i, j, y l, u. Evaluamos estas coordenadas en las matrices $\boldsymbol{D}$ y $\boldsymbol{G}$ para determinar una secuencia de ADN con 16 nuevas posiciones. Las nuevas posiciones de estos elementos están contenidas en los vectores $\boldsymbol{E}$ y $\boldsymbol{H}$. Finalmente, la información completa de estas secuencias ADN se definen en la matriz A1.

$$\boldsymbol{D} = \boldsymbol{G} =$$

$$\begin{bmatrix}
5 & 10 & 0 & 13 & 2 & 7 & 4 & 11 & 6 & 1 & 8 & 9 & 15 & 14 & 3 & 12 & 5 & 10 & 0 & 13 & 2 & 7 & 4 & 11 & 6 & 1 & 8 & 9 & 15 & 14 & 3 & 12 \\
14 & 5 & 12 & 9 & 4 & 7 & 8 & 10 & 11 & 3 & 1 & 15 & 13 & 2 & 0 & 6 & 14 & 5 & 12 & 9 & 4 & 7 & 8 & 10 & 11 & 3 & 1 & 15 & 13 & 2 & 0 & 6 \\
10 & 5 & 9 & 2 & 7 & 3 & 13 & 4 & 14 & 6 & 8 & 1 & 11 & 15 & 12 & 0 & 10 & 5 & 9 & 2 & 7 & 3 & 13 & 4 & 14 & 6 & 8 & 1 & 11 & 15 & 12 & 0 \\
13 & 7 & 5 & 10 & 0 & 9 & 8 & 11 & 2 & 4 & 1 & 15 & 14 & 6 & 3 & 12 & 13 & 7 & 5 & 10 & 0 & 9 & 8 & 11 & 2 & 4 & 1 & 15 & 14 & 6 & 3 & 12 \\
11 & 13 & 12 & 8 & 14 & 9 & 4 & 7 & 5 & 2 & 6 & 0 & 3 & 1 & 15 & 10 & 11 & 13 & 12 & 8 & 14 & 9 & 4 & 7 & 5 & 2 & 6 & 0 & 3 & 1 & 15 & 10 \\
5 & 4 & 2 & 15 & 11 & 12 & 7 & 8 & 1 & 0 & 3 & 13 & 9 & 10 & 14 & 6 & 5 & 4 & 2 & 15 & 11 & 12 & 7 & 8 & 1 & 0 & 3 & 13 & 9 & 10 & 14 & 6 \\
14 & 13 & 15 & 5 & 10 & 3 & 11 & 12 & 7 & 9 & 1 & 2 & 0 & 6 & 4 & 8 & 14 & 13 & 15 & 5 & 10 & 3 & 11 & 12 & 7 & 9 & 1 & 2 & 0 & 6 & 4 & 8 \\
7 & 9 & 11 & 14 & 15 & 6 & 5 & 0 & 10 & 4 & 2 & 8 & 13 & 1 & 12 & 3 & 7 & 9 & 11 & 14 & 15 & 6 & 5 & 0 & 10 & 4 & 2 & 8 & 13 & 1 & 12 & 3 \\
3 & 4 & 5 & 6 & 1 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 0 & 7 & 2 & 3 & 4 & 5 & 6 & 1 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 0 & 7 & 2 \\
8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 & 15 & 14 & 13 & 12 & 11 & 10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 & 15 & 14 & 13 & 12 & 11 & 10 & 9 \\
2 & 4 & 8 & 6 & 10 & 12 & 14 & 0 & 1 & 3 & 5 & 7 & 9 & 11 & 13 & 15 & 2 & 4 & 8 & 6 & 10 & 12 & 14 & 0 & 1 & 3 & 5 & 7 & 9 & 11 & 13 & 15 \\
3 & 15 & 1 & 13 & 11 & 9 & 7 & 5 & 2 & 14 & 0 & 12 & 10 & 8 & 6 & 4 & 3 & 15 & 1 & 13 & 11 & 9 & 7 & 5 & 2 & 14 & 0 & 12 & 10 & 8 & 6 & 4 \\
10 & 12 & 11 & 7 & 13 & 8 & 3 & 6 & 15 & 1 & 5 & 9 & 2 & 0 & 4 & 14 & 10 & 12 & 11 & 7 & 13 & 8 & 3 & 6 & 15 & 1 & 5 & 9 & 2 & 0 & 4 & 14 \\
6 & 5 & 3 & 0 & 12 & 13 & 8 & 9 & 2 & 11 & 4 & 10 & 14 & 1 & 15 & 7 & 6 & 5 & 3 & 0 & 12 & 13 & 8 & 9 & 2 & 11 & 4 & 10 & 14 & 1 & 15 & 7 \\
4 & 12 & 9 & 5 & 0 & 13 & 15 & 1 & 6 & 7 & 11 & 8 & 10 & 2 & 14 & 3 & 4 & 12 & 9 & 5 & 0 & 13 & 15 & 1 & 6 & 7 & 11 & 8 & 10 & 2 & 14 & 3 \\
1 & 0 & 6 & 15 & 10 & 5 & 4 & 2 & 8 & 3 & 14 & 13 & 7 & 12 & 9 & 11 & 1 & 0 & 6 & 15 & 10 & 5 & 4 & 2 & 8 & 3 & 14 & 13 & 7 & 12 & 9 & 11 \\
0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\
15 & 14 & 13 & 12 & 11 & 10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 & 15 & 14 & 13 & 12 & 11 & 10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\
12 & 6 & 14 & 8 & 0 & 2 & 4 & 10 & 11 & 13 & 15 & 9 & 1 & 7 & 3 & 5 & 12 & 6 & 14 & 8 & 0 & 2 & 4 & 10 & 11 & 13 & 15 & 9 & 1 & 7 & 3 & 5 \\
1 & 3 & 5 & 7 & 9 & 11 & 13 & 15 & 0 & 2 & 4 & 6 & 8 & 10 & 12 & 14 & 1 & 3 & 5 & 7 & 9 & 11 & 13 & 15 & 0 & 2 & 4 & 6 & 8 & 10 & 12 & 14 \\
2 & 0 & 6 & 4 & 8 & 14 & 12 & 3 & 10 & 1 & 7 & 5 & 9 & 11 & 15 & 13 & 2 & 0 & 6 & 4 & 8 & 14 & 12 & 3 & 10 & 1 & 7 & 5 & 9 & 11 & 15 & 13 \\
7 & 15 & 11 & 9 & 7 & 5 & 3 & 14 & 12 & 4 & 6 & 8 & 10 & 2 & 0 & 1 & 7 & 15 & 11 & 9 & 7 & 5 & 3 & 14 & 12 & 4 & 6 & 8 & 10 & 2 & 0 & 1 \\
13 & 5 & 10 & 3 & 8 & 9 & 6 & 2 & 0 & 15 & 14 & 13 & 1 & 11 & 12 & 4 & 13 & 5 & 10 & 3 & 8 & 9 & 6 & 2 & 0 & 15 & 14 & 13 & 1 & 11 & 12 & 4 \\
9 & 6 & 15 & 11 & 0 & 10 & 4 & 13 & 2 & 1 & 8 & 14 & 12 & 3 & 7 & 5 & 9 & 6 & 15 & 11 & 0 & 10 & 4 & 13 & 2 & 1 & 8 & 14 & 12 & 3 & 7 & 5
\end{bmatrix} \quad \text{(A1)}$$