

## SINCRONIZACIÓN CAÓTICA DEL MAPA DE HÉNON EN CONFIGURACIÓN MAESTRO-ESCLAVO IMPLEMENTADA EN LA PLACA ARDUINO

Acosta-Del Campo O. R.<sup>1</sup>, Cruz-Hernández C.<sup>2</sup>, López-Gutiérrez R.M.<sup>3</sup>, Arellano-Delgado A.<sup>2</sup>, Ortega-Corral C.<sup>1</sup>, López-Montoya J.E.<sup>1</sup>

<sup>1</sup>Universidad Tecnológica de Tijuana (UTT)  
Tecnologías de la Información y Comunicación  
Km. 10 Carretera Libre Tijuana-Tecate, El Refugio. Quintas Campestre. Tijuana, B.C., C.P. 22253  
Tel +52 (664) 969 4700  
e-mail [oscar.acosta@uttijuana.edu.mx](mailto:oscar.acosta@uttijuana.edu.mx), [cesar.ortegac@uttijuana.edu.mx](mailto:cesar.ortegac@uttijuana.edu.mx), [jenrique.lopez@uttijuana.edu.mx](mailto:jenrique.lopez@uttijuana.edu.mx)

<sup>2</sup>Centro de Investigación Científica y Educación Superior de Ensenada (CICESE)  
Departamento de Electrónica y Telecomunicaciones  
Carretera Ensenada-Tijuana, No. 3918, Zona Playitas, Ensenada, B.C., CP.22860  
Tel +52 (646) 175-05-00  
e-mail [ccruz@cicese.mx](mailto:ccruz@cicese.mx), [aarellan@cicese.mx](mailto:aarellan@cicese.mx)

<sup>3</sup>Universidad Autónoma de Baja California (UABC)  
Facultad de Ingeniería, Arquitectura y Diseño  
Km. 103, Carretera Tijuana-Ensenada, Ensenada, B.C., CP 22860  
Tel. +52 (646) 175-07-10  
e-mail [roslopez@uabc.edu.mx](mailto:roslopez@uabc.edu.mx)

### RESUMEN.

En este trabajo se reporta la sincronización de dos sistemas caóticos acoplados unidireccionalmente. En particular, se realiza la sincronización maestro-esclavo del mapa de Hénon, dicha sincronización se logra haciendo uso de la teoría compleja utilizada para la sincronización de redes complejas, esta metodología de sincronización es superior a otras reportadas, además se ha aplicado exitosamente a la sincronización muchos sistemas caóticos e hipercaóticos, su principal ventaja es que solo se utiliza un estado como retroalimentación para lograr la sincronía.

Las pruebas de generación de dinámicas caóticas y sincronización se realizaron utilizando la placa Arduino Uno, los resultados fueron graficados utilizando el software Matlab, además dicha sincronización se comparó con la realizada utilizando el software de simulación antes mencionado.

**Palabras Clave:** Sincronización caótica, Mapa de Hénon, Arduino.

### ABSTRACT.

In this paper the synchronization of two unidirectionally coupled chaotic systems is reported. In particular, we show the synchronization master-slave of Hénon map, such synchronization is achieved using complex theory, used for synchronization of complex networks, this synchronization method is superior to other reported, also been successfully

applied to the synchronization of any chaotic and hyperchaotic systems, its main advantage is that only one state is used as feedback to achieve synchrony.

The results of chaotic dynamic generation and, synchronization is performed using the Arduino Uno board, the results were plotted using the Matlab software, such synchronization also was compared with the realized using the above mentioned simulation software.

**Keywords:** Chaotic Synchronization, Hénon map, guidelines, Arduino.

### 1. INTRODUCCIÓN

La sincronización caótica ha recibido mucha atención en las últimas décadas, [1-4]. Muchos diferentes métodos han sido propuestos para conseguir la sincronización de sistemas caóticos, control de retroalimentación no lineal o lineal [5], control adaptivo [6], control pasivo [7], control impulsivo [8] o enfoque desde el enfoque de observador [9].

La sincronización es una propiedad importante de los sistemas dinámicos y más aún cuando tales sistemas tienen un comportamiento caótico, lograr la sincronización de sistemas caóticos ofrece alternativas superiores a explorar, una de las que ha tenido mayor auge en la actualidad es la aplicación a las comunicaciones seguras [10-12].

La mayoría del trabajo reportado de sincronización se ha basado en la sincronización de sistemas caóticos e hipercaóticos en tiempo continuo [13,14], donde en muchos casos se prefiere tener sistemas en tiempo real que tiempo discreto. Una desventaja de los sistemas caóticos clásicos es el uso de sistemas electrónicos complejos hechos de componentes electrónicos análogos como OP-AMPs, resistencias, capacitores, inductores y diodos. Por lo regular esta complejidad nos lleva a una tediosa implementación. [15]. La nueva tendencia, por lo tanto, es realizarlos con modelos discretos que representan fielmente la dinámica de tales sistemas. Esto es principalmente debido a dos razones. La primera es que, en la práctica, las mediciones se llevan a cabo por lo general en intervalos de tiempo específicos. En segundo lugar, las simulaciones digitales se pueden realizar fácil y rápidamente, ya sea en un microcontrolador [16,17] o en una tarjeta FPGA [18].

En este trabajo se realiza la sincronización de dos sistemas caóticos discretos, particularmente la sincronización del mapa de Hénon [19] en configuración maestro-esclavo. Dicha sincronización se logra por medio de la teoría compleja [20] utilizada para la sincronización de redes complejas, sin embargo, para lograr la sincronización de sistemas discretos como es el caso del mapa de Hénon, se realizó una modificación a la teoría de sincronización utilizada lo cual será reportado próximamente. La implementación de los mapas de Hénon utilizados, así como su sincronización se realiza en una placa de desarrollo Arduino Uno, lo cual no ha sido reportado en la literatura. La elección de la placa Arduino Uno fue motivada por las ventajas que ofrece. Exhibe buenas actuaciones de rendimiento y costo en términos de recursos utilizados. Además, el uso de un Arduino simplifica la cantidad de hardware y desarrollo de software necesario para conseguir un sistema en funcionamiento. En cuanto al software, Arduino proporciona una serie de bibliotecas para hacer la programación del microcontrolador más fácil.

En este trabajo se muestran los resultados obtenidos de la sincronización y generación de dinámicas caóticas provenientes de la placa Arduino Uno, tales resultados se muestran como graficas generadas con el lenguaje de programación de alto nivel Matlab, además los resultados se comparan con la sincronización realizada en dicho lenguaje de programación.

El resto del documento está organizado de la siguiente manera: En la sección 2 se dan las preliminares, es decir, la teoría necesaria para realizar la sincronización y los detalles del sistema a sincronizar. En la sección 3 se muestran los resultados experimentales, tanto de la implementación del mapa de Hénon en la placa Arduino como los obtenidos en el lenguaje de programación Matlab, también se muestran los resultados de la sincronización maestro-esclavo de dichos

sistemas. En la sección 4 se dan las conclusiones finales de este trabajo.

## 2. PRELIMINARES

### 2.1. Teoría de redes complejas.

Consideremos una red compleja compuesta de  $N$  nodos idénticos, acoplados linealmente a través de la primera variable de estado de cada nodo [20]. En esta red dinámica, cada nodo constituye un sistema dinámico de dimensión  $n$ , descrito como sigue

$$\dot{X}_i = f(X_i) + u_i, \quad i = 1, 2, \dots, N. \quad (1)$$

donde  $X_i = (x_{i1}, x_{i2}, \dots, x_{in})^T \in \mathcal{R}^n$  son las variables de estado del nodo  $i$ , mientras que  $u_i = u_{i1} \in \mathcal{R}^n$  es la señal de entrada del nodo  $i$ , definida por

$$u_{i1} = c \sum_{j=1}^N a_{ij} \Gamma X_j, \quad i = 1, 2, \dots, N. \quad (2)$$

$c > 0$  representa la fuerza de acoplamiento y  $A = (a_{ij})_{(N \times N)}$  es la matriz de acoplamiento de la red dinámica compleja (1). Si hay conexión entre el nodo  $i$  y el nodo  $j$ , entonces  $a_{ij} = 1$ ; por otro lado,  $a_{ij} = 0$  ( $i \neq j$ ).

Si se considera un acoplamiento simétrico y difusivo. En particular, se supone que:

- (i)  $A$  es una matriz simétrica e irreducible.
- (ii) Los elementos fuera de la diagonal,  $a_{ij}$  ( $i \neq j$ ) de  $A$ , son o bien 1 (cuando hay conexión entre el nodo  $i$  y el nodo  $j$ ) o 0 (cuando no hay conexión entre el nodo  $i$  y el nodo  $j$ ).
- (iii) Los elementos de la diagonal principal de  $A$  satisfacen

$$a_{ii} = - \sum_{j \neq i}^N a_{ij}, \quad i = 1, 2, \dots, N. \quad (3)$$

Las condiciones anteriores implican que un valor propio de  $A$  es cero con multiplicidad 1 y todos los otros valores propios son estrictamente negativos.

La red dinámica compleja (1) se puede caracterizar por los valores propios distintos de cero de la matriz de acoplamiento  $A$ . Un resultado típico establece que la red dinámica compleja (1) sincronizará si estos valores propios son lo suficientemente negativo [20].

**Lema 1** [20]. Considere la dinámica compleja de la red (1). Sea  $\lambda_1$  el mayor valor propio diferente de 0 de la matriz de acoplamiento  $A$  de la red. El estado de sincronización de red compleja (1) definida por  $x_1 = x_2 = \dots = x_n$  es asintóticamente estable, si

$$\lambda_1 \leq -\frac{T}{c} \quad (4)$$

donde  $c > 0$  representa la fuerza de acoplamiento de la red y  $T$  es una constante positiva tal que cero es un punto exponencialmente estable. La condición (4) garantiza la sincronización si  $\lambda_1$  es lo suficientemente negativo.

## 2.2. Mapa de Hénon.

El mapa de Hénon fue presentado por el astrónomo Francés Michel Hénon (1976). Dicho sistema está descrito por las siguientes ecuaciones no lineales en diferencias [19]:

$$\begin{aligned} x(n+1) &= \alpha + \beta y(n) - x^2(n) \\ y(n+1) &= x(n) \end{aligned} \quad (5)$$

Con el propósito de generar la dinámica caótica y un atractor caótico para el mapa de Hénon, se utilizan los siguientes valores para los parámetros en simulaciones numéricas generadas en Matlab,  $\beta = 0.3$ ,  $\alpha = 1.4$  y condiciones iniciales  $x(0) = 0.11$  y  $y(0) = 0$ . El atractor caótico generado por el mapa de Hénon con los parámetros anteriores se ilustra en la figura 1.

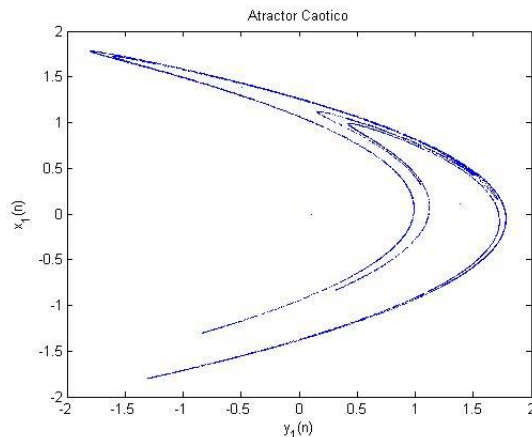


Figura 1. Atractor caótico del mapa de Hénon.

## 2.3. Plataforma Arduino.

La realización de este trabajo fue desarrollada en la plataforma de código abierto Arduino Uno, compuesto por un procesador AVR de Atmel (microcontrolador). Esta implementación se realizó debido a la flexibilidad y lo fácil de utilizar el hardware y software de la plataforma Arduino. Tiene 14 pines digitales de entrada/salida, seis entradas analógicas, un oscilador de cristal de 16MHz, una conexión USB, un conector de alimentación, un header ICSP y un botón de reinicio como se muestra en la figura 2. Hay que destacar que la placa Arduino puede ser programada de varias maneras [21]. En este trabajo se eligió el ambiente de desarrollo IDE (Integrated

Development Environment). En este ambiente de desarrollo se creó el código a partir de (5), que nos ofrece las dinámicas caóticas del mapa de Hénon tanto del maestro como del esclavo, además en dicho código se realiza la sincronización maestro-esclavo. A partir de (4) se obtienen las condiciones que garantizan dicha sincronización. Los resultados obtenidos fueron datos numéricos generados por la placa Arduino, posteriormente estos datos fueron graficados mediante el software de simulación Matlab, donde se aprecian las dinámicas caóticas y el error de sincronía.



Figura 2. Placa de desarrollo Arduino Uno.

## 3. RESULTADOS EXPERIMENTALES

### 3.1 Dinámicas caóticas.

A continuación se muestran los resultados obtenidos al implementar el mapa discreto de Hénon en la plataforma Arduino, estos resultados son comparados con los obtenidos al realizar la simulación con el lenguaje de programación Matlab. Los parámetros utilizados para la implementación son los siguientes:  $\beta = 0.3$ ,  $\alpha = 1.4$  y condiciones iniciales  $x(0) = 0.11$  y  $y(0) = 0$ . En la figura 3 se aprecian las dinámicas de los estados  $x(n)$  y  $y(n)$  del mapa de Hénon generado en Arduino.

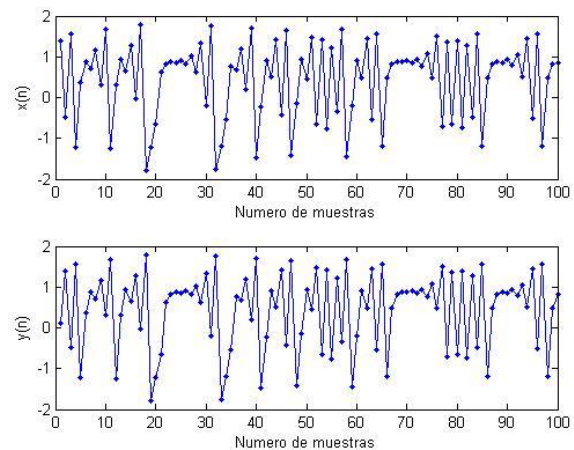


Figura 3. Dinámicas caóticas generadas en Arduino.

En la figura 4 se muestran las dinámicas caóticas generadas en el software de simulación Matlab, los parámetros y condiciones iniciales fueron las mismas que las utilizadas en la placa Arduino.

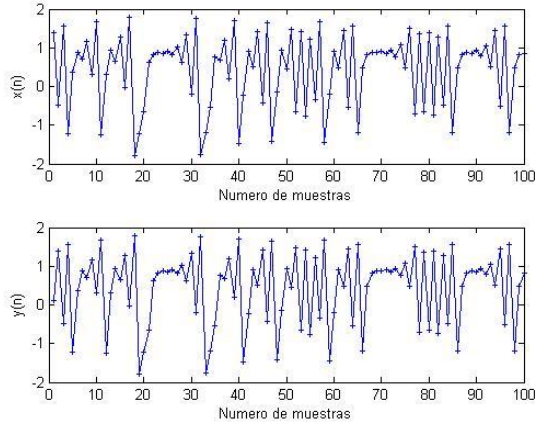


Figura 4. Dinámicas caóticas generadas en Matlab.

La figura 5 muestra el atractor caótico generado en la placa Arduino, el atractor se forma graficando el estado  $x(n)$  vs  $y(n)$ . En la figura 6 se aprecia el atractor generado en Matlab, como se puede observar son exactamente iguales.

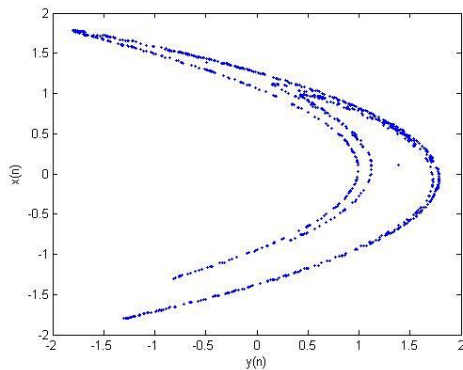


Figura 5. Atractor caótico generado en Arduino.

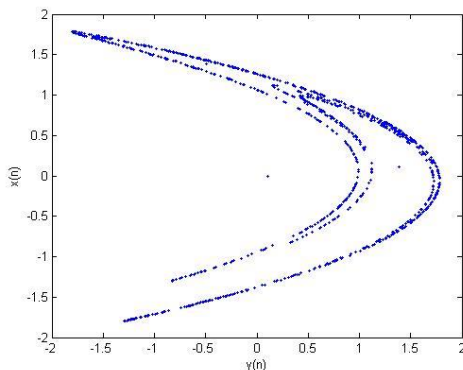


Figura 6. Atractor caótico generado en Matlab.

### 3.1. Sincronización caótica.

Una vez comprobado que el mapa de Hénon puede ser implementado en la placa Arduino, se prosiguió a realizar la sincronización utilizando la teoría compleja, sin embargo, la teoría compleja surge a partir de querer la sincronización de redes complejas y específicamente para sistemas dinámicos continuos, al ser el mapa de Hénon discreto no se puede implementar dicha teoría directamente, sin embargo, en investigaciones previas se logró adaptar dicha teoría a la sincronización de sistemas discretos, específicamente el mapa de Hénon, dichos resultados serán próximamente reportados en la literatura.

Para realizar la sincronización maestro-esclavo es necesario agregar en el primer estado del esclavo la retroalimentación del error entre el primer estado del maestro y el primer estado del esclavo, las ecuaciones del maestro y esclavo se muestran a continuación:

$$\begin{aligned} x_m(n+1) &= \alpha + \beta y_m(n) - x_m^2(n) \\ y_m(n+1) &= x_m(n) \end{aligned} \quad (6)$$

$$\begin{aligned} x_e(n+1) &= \alpha + \beta y_e(n) - x_e^2(n) + c(x_e^2 - x_m^2) \\ y_e(n+1) &= x_e(n) \end{aligned} \quad (7)$$

Los parámetros utilizados para el mapa de Hénon maestro y el mapa de Hénon esclavo son los siguientes:  $\beta = 0.3$ ,  $\alpha = 1.4$ , las condiciones iniciales del maestro son:  $x_m(0) = 0.11$  y  $y_m(0) = 0$ , las condiciones iniciales de esclavo son:  $x_e(0) = 0$  y  $y_e(0) = 0$ . Con una fuerza de acoplamiento  $c = 0$  los dos mapas están aislados, por lo tanto no habrá sincronización, esto se aprecia en las siguientes figuras.

La figura 7 muestra el atractor caótico generado por el maestro, el cual fue implementado en Arduino y los resultados obtenidos graficados en Matlab.

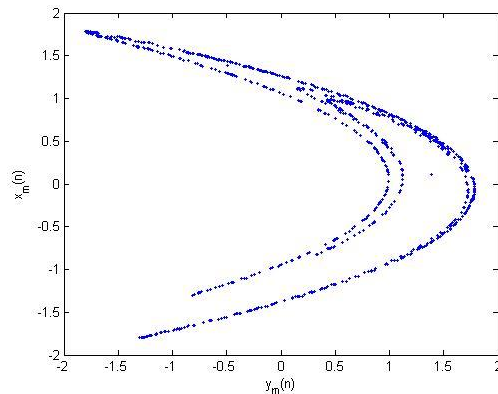
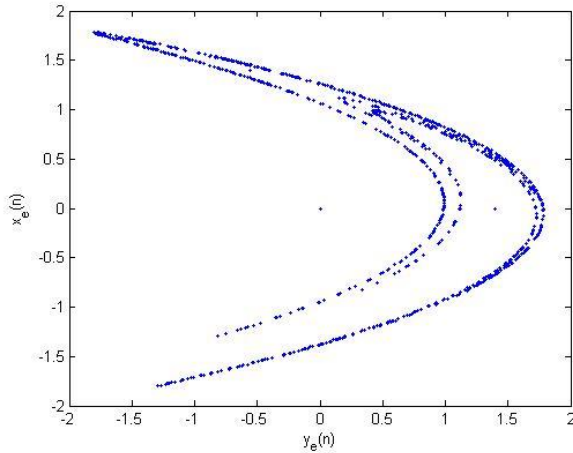


Figura 7. Atractor caótico del maestro cuando  $c = 0$ .

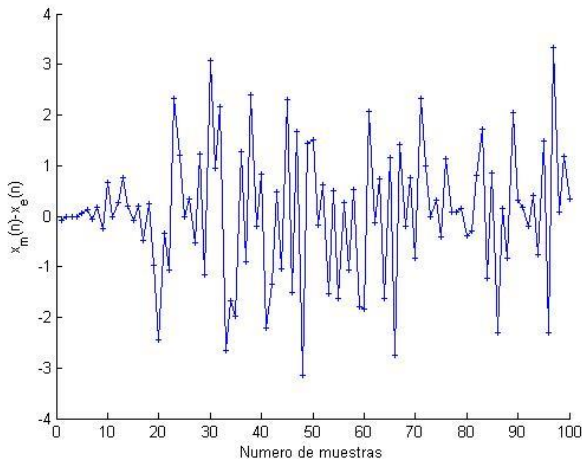


La figura 8 muestra el atractor caótico generado por el esclavo, el cual fue implementado en Arduino y los resultados obtenidos graficados en Matlab.



**Figura 8. Atractor caótico del esclavo cuando  $c = 0$ .**

A simple vista los dos atractores son iguales, sin embargo, al graficar el error de sincronía entre ellos podemos ver que las dinámicas generadas por los mapas no son iguales, por lo tanto, no hay sincronía entre el maestro y el esclavo. Esto se aprecia en la figura 9.

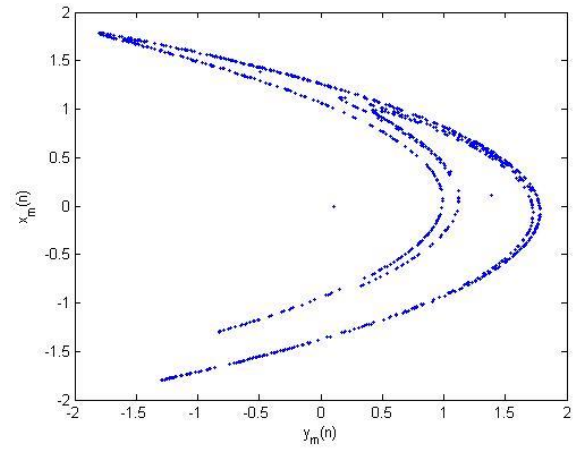


**Figura 9. Error de sincronía entre el maestro y esclavo cuando  $c = 0$ .**

Para garantizar la sincronización entre el maestro y el esclavo se eligió una fuerza de acoplamiento  $c = 0.8$ , este valor es suficiente para garantizar la sincronización, lo cual se aprecia en las siguientes figuras.

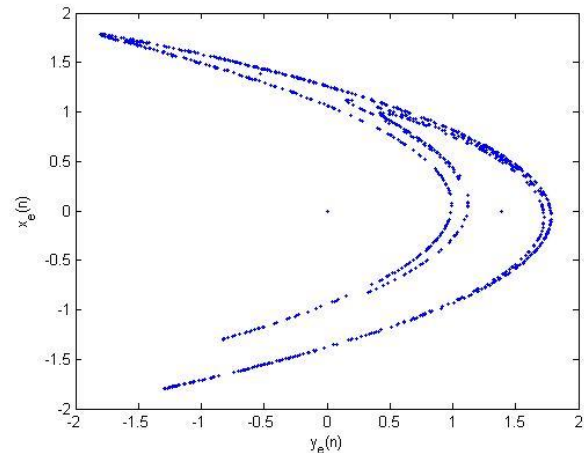
La figura 10 muestra el atractor caótico generado por el maestro, el cual fue implementado en Arduino y los resultados

obtenidos graficados en Matlab, para este atractor hay acoplamiento entre el maestro y esclavo ( $c = 0.8$ ).



**Figura 10. Atractor caótico del maestro cuando  $c = 0.8$ .**

La figura 11 muestra el atractor caótico generado por el esclavo, el cual fue implementado en Arduino y los resultados obtenidos graficados en Matlab, para este atractor hay acoplamiento entre el maestro y esclavo ( $c = 0.8$ ).



**Figura 11. Atractor caótico del esclavo cuando  $c = 0.8$ .**

En la figura 12 se ilustra el error de sincronía entre el maestro y el esclavo ( $x_m(n) - x_e(n)$ ) utilizando una fuerza de acoplamiento entre ellos de  $c = 0.8$

En la figura 13 se aprecia como el estado  $x_e(n)$  del esclavo marcado en rojo se sincroniza con el estado  $x_m(n)$  del maestro marcado en azul, se observa que en la octava iteración los dos estados ya se encuentran sincronizados. Cabe señalar que se puede elegir otra fuerza de acoplamiento, esto para lograr una sincronización más rápida o bien retrasar dicha sincronización.

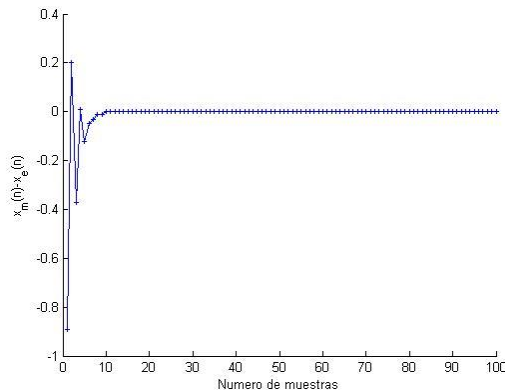


Figura 12. Error de sincronía entre el maestro y esclavo cuando  $c = 0.8$ .

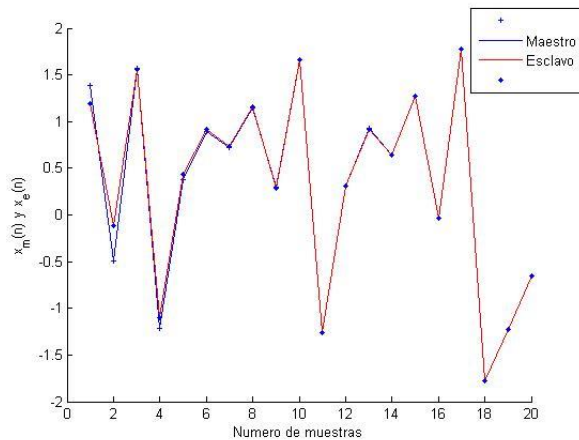


Figura 13. Sincronía entre el maestro y el esclavo con  $c = 0.8$ .

#### 4. CONCLUSIONES

Con la realización de esta investigación se comprueba que se puede lograr la implementación de sistemas caóticos discretos en la placa de desarrollo Arduino y más aún se puede obtener la sincronización entre dos o más sistemas caóticos, en este caso se implementó la sincronización de dos mapas de Hénon dispuestos en configuración maestro-esclavo, los resultados obtenidos fueron graficados en Matlab, donde se observó que dichos resultados son iguales a los obtenidos realizando la simulación de la sincronización en Matlab.

Para tal sincronización se utilizó la teoría compleja, la cual se modificó para poder adaptarla a los sistemas discretos, además se comprobó que con una pequeña fuerza de acoplamiento, en este caso  $c = 0.8$  se logró obtener la sincronización deseada. Con esta investigación se abre el paso a la potencial aplicación de esta placa de desarrollo para el encriptado de información, meta que se piensa cumplir en posteriores investigaciones.

#### REFERENCIAS

- [1] Pecora L.M. and Carroll T.L. (1990). Synchronization in chaotic systems. *Phys. Rev. Lett.* 64:821-824.
- [2] Nijmeijer H. and Mareels I.M.Y. (1997). An observer looks at synchronization. *IEEE Trans. Circuits Syst.* 44(10):882-890.
- [3] Acosta-Del Campo O.R., Cruz-Hernández C., López-Gutiérrez R.M., Arellano-Delgado A., Cardoza-Avendaño L. (2011): Complex network synchronization of coupled time-delay chua oscillators in different topologies. *Nonlinear Dynamics and Systems Theory*, Vol.11, Pag.341-372.
- [4] Lopez-Mancilla D. and Cruz-Hernandez C. (2008) Out-put synchronization of chaotic systems under non-vanishing perturbations. *Chaos, Solitons & Fractals* 37(4):1172-86.
- [5] Wang, H., Han, Z.Z., Mo Commun, Z.: Synchronization of hyperchaotic systems via linear control. *Commun. Nonlinear. Sci. Numer. Simul.* 15(7), 19101920 (2010).
- [6] Han, X., Lu, J.A., Wu, X.: Adaptive feedback synchronization of L? system. *Chaos Soliton Fractals* 22(1), 221–227 (2004)
- [7] Hamiche, H., Kemih, K., Ghanes, M., Zhang, G., Djennoune, S.: Passive and impulsive synchronization of a new four-dimensional chaotic system. *Nonlinear Anal. Theory Methods Appl.* 74(4), 1146–1154 (2011)
- [8] Yang, T., Chua, L.O.: Impulsive stabilization for control and synchronization of chaotic systems: theory and application to secure communication. *IEEE. Trans. Circuits Syst. I Fundam. Theory. Appl.* 44(10), 976–978 (1997)
- [9] Acosta-Del Campo, O.R., Cruz-Hernández, C., López-Gutiérrez R.M., Arellano-Delgado, A.: Synchronization of chaotic chua oscillators of fractional order. 16 Convención Científica de Ingeniería y Arquitectura (CUJAE), Cuba, 2012.
- [10] Acosta-Del Campo, O.R., Cruz-Hernández, C., López-Gutiérrez R.M., Arellano-Delgado, A. (2013). Communication in star coupled network with fractional hyperchaotic nodes. *IEEE 4th Latin American symposium on circuits and systems (LASCAS)*, Peru.
- [11] Murillo-Escobar, M.A., Cruz-Hernández C., López-Gutiérrez R.M., Abundiz-Pérez F., Acosta-Del Campo O.R. (2014): A RGB image Encryption Algorithm base don total plain image characteristic and chaos. *Signal Processing*, vol 109, 119-131.
- [12] Cruz-Hernandez C. (2004). Synchronization of time-delay Chua's oscillator with application to secure communication. *Nonlinear Dyn. Syst. Theor.* 4(1):1{13.
- [13] Blakely, J.N., Eskridge, M.B., Corron, N.J.: A simple Lorenz circuit and its radio frequency implementation. *Chaos* 17(2), 023112 (2007)
- [14] Qi, G., Chen, G.: Analysis and circuit implementation of a new 4D chaotic system. *Phys. Lett. A* 352(4–5), 386–397 (2006).
- [15] Huang, C.K., Tsay, S.C., Wu, Y.R.: Implementation of chaotic secure communication systems based on OPA circuits. *Chaos Solitons Fractals* 23(2), 589–600 (2005)
- [16] Volos, C.K.: Chaotic random bit generator realized with a microcontroller. *J. Comput. Model.* 3(4), 115–136 (2013)
- [17] Zuppa, L.A.: Chaotic logistic map implementation in the PIC12F629 microcontroller unit. 10th IFAC Workshop on Programmable Devices and Embedded Systems, 10(1), Poland, (2010)
- [18] Koyuncu, I., Ozcerit, A.T., Pehlivan, I.: Implementation of FPGA-based real time novel chaotic oscillator. *Nonlinear Dyn.* 77(1–2), 49–59 (2014).
- [19] Dmitriev A. S., G. A. Kassian, Khilinsky A. D. "Chaotic synchronization of Hénon mappings: the information approach", *Technical Physics Letters* 28 (2002) 5.
- [20] Wang X.F. and Chen G. (2002) Synchronization in small-world dynamical networks," *Int. J. Bifurcation and Chaos* 12, 187-192.
- [21] Arduino. [www.arduino.cc](http://www.arduino.cc)