



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

---

FACULTAD DE ESTUDIOS SUPERIORES ACATLÁN

UN MODELO PREDICTIVO PARA  
SERIES DE TIEMPO ECONÓMICAS  
IRREGULARES

TESINA

QUE PARA OBTENER EL TÍTULO DE:  
**Lic. en Matemáticas Aplicadas y  
Computación**

PRESENTA:

**Pedro Raúl Jiménez Macías**

DIRECTOR DEL TRABAJO:  
Alma López Blanco

Ethelvina Alemán Zepeda

México, 2016



Santa Cruz Acatlán, Naucalpan, Edo. de México



Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

*A mi madre,  
porque sin su apoyo nada de esto sería posible.*

*Dalia Jiménez,  
por su amor y paciencia.*

*Ethelvina Alemán,  
por su amistad e incondicional apoyo.*

*Y a todas aquéllas personas que hicieron  
esto posible.*

# Índice general

---

|  |           |
|--|-----------|
| Introducción . . . . .                                       | 7         |
| <b>1. Reconstrucción por tiempos de retraso</b>              | <b>9</b>  |
| 1.1. Sistemas dinámicos lineales y no lineales . . . . .     | 9         |
| 1.2. Reconstrucción del espacio de configuraciones . . . . . | 13        |
| 1.2.1. El método de los falsos vecinos . . . . .             | 14        |
| 1.2.2. Estimando el tiempo de retraso . . . . .              | 16        |
| 1.3. El algoritmo genético . . . . .                         | 17        |
| 1.4. La Transformación de wavelets . . . . .                 | 19        |
| <b>2. Series de tiempo económicas irregulares</b>            | <b>25</b> |
| 2.1. Evolución del Producto Interno Bruto . . . . .          | 26        |
| 2.1.1. Transformación de Wavelet . . . . .                   | 26        |
| 2.2. Obteniendo el modelo predictivo . . . . .               | 29        |
| <b>3. Análisis de Resultados</b>                             | <b>33</b> |
| 3.1. Análisis general . . . . .                              | 33        |
| 3.2. Predicciones . . . . .                                  | 35        |
| <b>4. Conclusiones</b>                                       | <b>37</b> |



# Índice de figuras

---

|      |   |    |
|------|---|----|
| 1.1. | Gráfica de la solución de un sistema dinámico lineal continuo . . . . .   | 11 |
| 1.2. | Gráfica de la solución $f(x)$ del sistema de Lorenz . . . . .   | 12 |
| 1.3. | Gráfica de la solución a un sistema dinámico discreto, el cual cuenta con una constante que sigue una distribución de probabilidad normal . . . . . | 12 |
| 1.4. | Comparación entre el atractor de Lorenz y el atractor obtenido mediante el método de reconstrucción por tiempos de retraso . . . . .                | 15 |
| 1.5. | Diagrama del proceso iterativo en la transformación de Wavelet . . . . .  | 21 |
| 1.6. | Ejemplo de transformación de Wavelet . . . . .  | 21 |
|      |   |    |
| 2.1. | Variación porcentual del PIB mexicano desde 1996 hasta el 2015 . . . . .  | 27 |
| 2.2. | Gráfica de los coeficientes de Wavelet del PIB mexicano . . . . .   | 27 |
| 2.3. | Transformación de Wavelet a la evolución de la variación porcentual del PIB mexicano a dos escalas diferentes . . . . .                             | 28 |
| 2.4. | Gráfica de la correlación lineal como función del tiempo de retraso . . . . .   | 29 |
| 2.5. | Gráfica de la Información mutua como función del tiempo de retraso . . . . .  | 30 |
| 2.6. | Gráfica que muestra el número de falsos vecinos como función de la dimensión de embedding . . . . .   | 31 |
| 2.7. | Atractor reconstruido a partir del teorema de Takens para la evolución del PIB mexicano . . . . .   | 31 |
|      |   |    |
| 3.1. | Ajuste obtenido mediante el algoritmo genético a la serie suavizada de la evolución del PIB mexicano . . . . .                                      | 34 |
| 3.2. | Ajuste obtenido mediante el algoritmo genético a la serie de tiempo de la evolución del PIB mexicano . . . . .                                      | 34 |
| 3.3. | Comparación entre los dos modelos generados, con y sin transformación de Wavelet . . . . .  | 35 |
| 3.4. | Comparación entre las predicciones obtenidas de los modelos con y sin transformación de Wavelet . . . . .   | 36 |



# Introducción

---

**E**L análisis de señales ha sido objeto de estudio por mucho tiempo, donde tal vez uno de los avances más significativos vino de la mano de Jean-Baptiste Joshep Furier quien en 1822 publicase “Théorie analytique de la chaleur” [1], donde inspirado en la resolución de un problema físico, Fourier desarrolla un formalismo que sustenta la expansión en serie de una función arbitraria. La repercusión de este tipo análisis es inmensurable, aún en las teorías más exóticas de la ciencia contemporánea el uso de conjuntos completos sigue siendo una herramienta fundamental. Posteriormente saldrían a la luz otros conjuntos con las mismas propiedades, unos más complejos que otros, pero la mayoría de ellos implementados con un mismo fin: observar el problema desde un punto de vista diferente, llevarlo a otro espacio, donde tal vez la resolución del problema se obtenga de una manera más fácil o quizá, surjan propiedades que a primera vista no son aparentes, por ejemplo, observar el comportamiento de una serie de tiempo en el espacio de Fourier permite indagar sobre la frecuencia con la que los datos se transforman. Una de las principales aplicaciones de este tipo de transformaciones es el suavizado de una señal que posee ruido. Esto sin lugar a dudas es de vital importancia en ciertas circunstancias, pero ¿qué tal si en aquel ruido viven las propiedades más importantes del sistema?, esto suele ocurrir en sistemas que son irregulares por naturaleza y donde la señal no esta siendo de ninguna forma contaminada por agentes externos. Si lo que se busca es hacer una predicción en base a la información que se tiene, suavizar los datos implica necesariamente una perdida de información sin la cual nuestra predicción no puede estar completa. Una de las áreas más importantes de aplicación de este y otro tipo de formalismos es la economía, donde la imposibilidad de involucrar todos los factores que intervienen en la evolución de esta clase de sistemas limita al tipo probabilístico o especulativo a las predicciones que pueden ser dadas, i.e., pese a que no se puede saber con certeza que pasará el día de mañana si podemos estimar comportamientos futuros en base a la información presente. Es por esto que un análisis basado únicamente en el suavizado de los datos no es suficiente para darle un carácter determinista a nuestras predicciones.

La irregularidad de los datos puede deberse a diversas razones, por ejemplo, es muy común que la presencia de un factor estocástico sea la responsable de tales fluctuaciones, aunque en sistemas deterministas puede presentarse este fenómeno. En la ausencia de una ley dinámica, es decir, un modelo teórico que describa la evolución de los datos, distinguir

las causas de este tipo de comportamientos se dificulta bastante. Sin embargo, en la literatura se han desarrollado un sin fin de técnicas que permitan estimar correlaciones y muchas otras propiedades de las series de tiempo. En el presente trabajo se propone el uso del algoritmo genético [2] como la principal herramienta para determinar dicha ley dinámica. Este algoritmo está basado en la evolución de un sistema biológico, donde una población inicial de ecuaciones se reproduce y transforma hasta llegar a un conjunto que describa, de la mejor manera posible, a la línea de tiempo. El computo requerido en ausencia total de información sobre el sistema en cuestión puede hacer que el algoritmo genético no sea la mejor opción e inclusive, no se logre un resultado. Es por eso que utilizaremos como herramienta auxiliar el método de reconstrucción por tiempos de retraso [3]. Tal método pretende encontrar la dimensión en que vive el sistema, dicho de otra forma, con cuántos valores de la serie de tiempo está relacionado de manera directa un sólo punto de la misma. Con ello el algoritmo genético sabrá de antemano con que clase de ecuaciones debe empezar a trabajar por obtener mejores resultados. Se ha escogido trabajar con la línea de tiempo generada por el valor porcentual del PIB, además de su carácter irregular, por la ventajas que conllevaría poder estimar un comportamiento futuro en tan importante serie económica. A modo de resumen presentamos una lista descriptiva de la metodología que será desarrollada a continuación:

En el primer capítulo del trabajo comenzamos discutiendo las propiedades de los sistemas dinámicos lineales y no lineales, con el objetivo de mostrar que un comportamiento irregular puede bien ser asociado con factores no lineales. Proseguimos describiendo el método de reconstrucción por tiempo de retraso, con principal énfasis en el ya mencionado teorema de Takens. Después se explican varias formas de estimar los parámetros necesarios para la reconstrucción del sistema, el tiempo de retraso y la dimensión de *embedding*. A continuación se describe a detalle los pasos que conforman el algoritmo genético que posteriormente será implementado. Finalizando con una profunda descripción sobre la transformación de wavelet, la cual permite controlar el nivel de suavizado en nuestra línea de tiempo.

En la siguiente sección hablamos sobre la evolución temporal del PIB y la importancia que tiene el modelado sobre esta línea de tiempo económica. Continuamos aplicando la teoría previamente descrita, comenzando por la transformación de wavelet para continuar con el uso del algoritmo genético que con ayuda del método de reconstrucción nos ofrecerá un modelo predictivo.

Finalizamos haciendo un análisis de los resultados obtenidos, tanto la comparación entre modelos con y sin suavizado, así como las respectivas predicciones. Para ello emplearemos la bien conocida prueba de Kolmogorov-Smirnov para distinguir la calidad de cada uno de nuestros ajustes.

## Objetivo del trabajo

Aplicar de manera conjunta el método de reconstrucción por tiempo de retraso y el algoritmo genético para poder obtener un modelo que permita hacer predicciones en series de tiempo económicas irregulares y verificar la validez de tal procedimiento. El trabajo pretende suponer correlaciones no lineales en la evolución temporal de un sistema económico para poder inferir propiedades globales en la evolución del mismo y de esta forma poder emplear el algoritmo genético como herramienta de predicción.

## Reconstrucción por tiempos de retraso

---

Es muy común que al recolectar información sobre algún sistema (entiéndase físico, biológico, económico,...,etc.) los datos recabados representen nuestro *único* conocimiento sobre éste. Generalmente, cuando se busca hacer una predicción con respecto a estos datos primero se realiza un análisis que supone una correlación lineal, pero los resultados no son siempre favorables. En la mayoría de los casos esto se debe a la presencia de *ruido* en la información. Por lo que suele realizarse una *identificación* de la señal con la cual se garantiza que los datos están libres de agentes externos. Sin embargo, hay ocasiones que la información es *caótica* por naturaleza. En esta sección explicaremos un método de análisis para ese tipo de información. Comenzaremos definiendo que es un *sistema dinámico lineal y no lineal*. Después nos centraremos en el *teorema de Takens*, parte medular del *método de tiempos de retraso* para la reconstrucción del espacio de configuraciones, el cual implementaremos usando la técnica de *los falsos vecinos*. Continuaremos explicando en que consiste el *algoritmo genético*, el cual nos permitirá realizar predicciones sobre el comportamiento de los datos.

### 1.1. Sistemas dinámicos lineales y no lineales

Para comenzar nuestro tratamiento es necesario hacer algunas definiciones.

**Definición 1.** Sea  $\mathcal{M}$  una variedad diferenciable y  $\kappa$  un semigrupo aditivo. Dado un mapeo lineal  $f : \mathcal{M} \times \kappa \rightarrow \mathcal{M}$  que escribimos  $f^t = f(x, t)$ . Si  $f^t$  satisface lo siguiente

1. *Identidad:*  $f^0 = \text{identidad}$ , i.e.,  $f(0, x) = x$  donde 0 es la identidad de  $\kappa$ .

2. *Aditividad*:  $f^t \circ f^s = f^{t+s} \forall t, s \in \kappa$ .

Entonces  $(\mathcal{M}, \kappa, f)$  o simplemente  $\mathcal{F} = \{f^t\}_{t \in \kappa}$ , es llamado **sistema dinámico** (SD) definido en el espacio de configuraciones<sup>1</sup>  $\mathcal{M}$ .

Dado un sistema dinámico  $\mathcal{F} = \{f^t\}_{t \in \kappa}$  definido en  $\mathcal{M}$ . El mapeo  $t \mapsto f^t(x)$  es llamado *movimiento* de  $x$ . La *trayectoria* de un punto  $x$  es el conjunto  $\mathcal{F}(x) = \{f^t(x)\}_{t \in \kappa}$ . Note que si  $\kappa = \mathbb{R}^+$  se habla de un *sistema dinámico continuo* y si  $\kappa \subseteq \mathbb{N}$  de un *sistema dinámico discreto*.

Generalmente  $f^t$  es el resultado de resolver un sistema de  $n$  ecuaciones diferenciales<sup>2</sup> de la forma:

$$\mathcal{H}(t, f^t, \frac{df^t}{dt}, \frac{d^2 f^t}{dt^2}, \dots, \frac{d^k f^t}{dt^k}) = 0.$$

Limitándonos al caso donde es posible resolver esta última expresión para la derivada de mayor orden

$$\frac{d^k f^t}{dt^k} = \mathcal{G}(t, f^t, \frac{df^t}{dt}, \frac{d^2 f^t}{dt^2}, \dots, \frac{d^{k-1} f^t}{dt^{k-1}}) \quad (1.1)$$

**Proposición 1.** *Todo sistema de ecuaciones diferenciales de la forma (1.1) puede reescribirse como un sistema de ecuaciones de primer orden a través del cambio de variable*

$$u^l = \frac{d^l f^t}{dt^l} \quad l = 0, 1, \dots, k$$

*generando un total de  $kn$  ecuaciones diferenciales.*

Cabe aclarar que la Proposición (1) no es el único medio de reducir el problema a un conjunto de ecuaciones de primer orden, pero es suficiente para enfocarnos en el caso más general:

$$\frac{df_i^t}{dt} = \mathcal{H}(t, f_1^t, f_2^t, \dots, f_n^t) \quad i = 1, 2, \dots, n \quad (1.2)$$

de las propiedades de la función  $\mathcal{H}$  podemos distinguir dos clases de SD:

**Definición 2.** *Un SD lineal es todo aquel sistema donde  $\mathcal{H}$  es una función lineal, lo que permite reescribir la ecuación (1.2) como*

$$\frac{d\mathbf{x}}{dt} = \mathbf{A} \cdot \mathbf{x} \quad (1.3)$$

donde  $\mathbf{x}$  es el vector columna con componentes  $f_1^t, f_2^t, \dots, f_n^t, t$  y  $\mathbf{A}$  es una matriz constante de  $(n+1) \times (n+1)$  elementos.

**Ejemplo 1.** Supongamos un SD lineal continuo cuya ecuación de movimiento está dada por

$$\frac{df^t}{dt} + \omega^2 f^t = 0$$

La solución se obtiene inmediatamente:

$$f^t(x) = x_0 \exp[-\omega^2 t].$$

Al tratarse de un sistema lineal se espera un comportamiento *regular* en la evolución de  $x$ , como el mostrado en la Figura (1.1).

<sup>1</sup>En la literatura también suele llamársele *espacio fase*.

<sup>2</sup> $n$  es el número de *grados de libertad* del sistema, por lo que  $f^t(x) \in \mathcal{M} \subseteq \mathbb{R}^n$ .

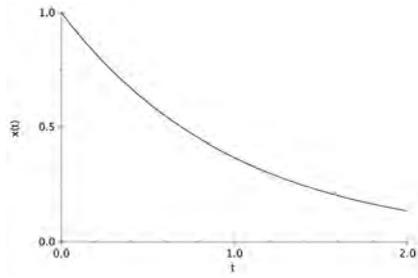


Figura 1.1: Solución del Ejemplo 1 usando  $x_0 = \omega = 1$ .

**Definición 3.** Un *SD no lineal* es aquel cuyo sistema de ecuaciones no pueda llevarse a la forma (1.3).

Es claro que estas definiciones de SD lineal y no lineal predisponen un conocimiento sobre el sistema de ecuaciones, pero nos permite identificar qué clase de comportamiento es el que esperamos observar en  $f^t(x)$ .

**Ejemplo 2.** Consideremos el sistema de ecuaciones de Lorenz (1963):

$$\begin{aligned}\frac{df^t(x)}{dt} &= \sigma\{f^t(y) - f^t(x)\} \\ \frac{df^t(y)}{dt} &= -f^t(x)f^t(z) + rf^t(x) - f^t(y) \\ \frac{df^t(z)}{dt} &= f^t(x)f^t(y) - bf^t(z)\end{aligned}$$

donde  $\sigma$ ,  $r$  y  $b$  son constantes. La solución de este sistema no puede obtenerse de manera analítica, sin embargo, utilizando el método de Runge-Kutta de 4to orden podemos apreciar el comportamiento de la solución  $f^t(x)$  en la Figura (1.2). El comportamiento *irregular* es característico de este tipo de sistemas y en muchas ocasiones, la primera razón por la que se supone no lineal.

Si bien el comportamiento irregular puede ser indicador de *correlaciones* no lineales, no es garantía de éste.

**Ejemplo 3.** Cuando a un sistema lineal se le incorpora un factor *estocástico* generalmente el comportamiento observado es irregular. Considere el siguiente sistema dinámico discreto

$$f^n(x) = 0.99f^{n-1}(x) + \eta_n,$$

donde  $\eta_n$  es una *variable aleatoria* que sigue una distribución *normal*. Aunque el modelo es claramente lineal, éste presenta un comportamiento altamente irregular (Figura 1.3).

Más adelante abordaremos el *método de los falsos vecinos* el cual puede ser usado de manera indirecta para detectar *atractores* los cuales pueden indicarnos propiedades importantes del sistema.

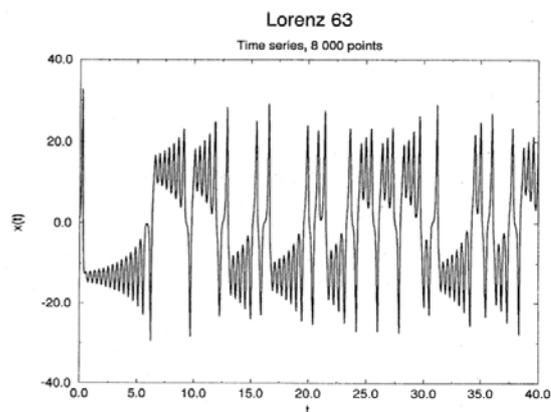


Figura 1.2: Gráfica de  $x(t)$  en el Ejemplo 2, usando  $r = 45.92$ ,  $b = 4.0$  y  $\sigma = 16.0$ . Figura tomada de [4]

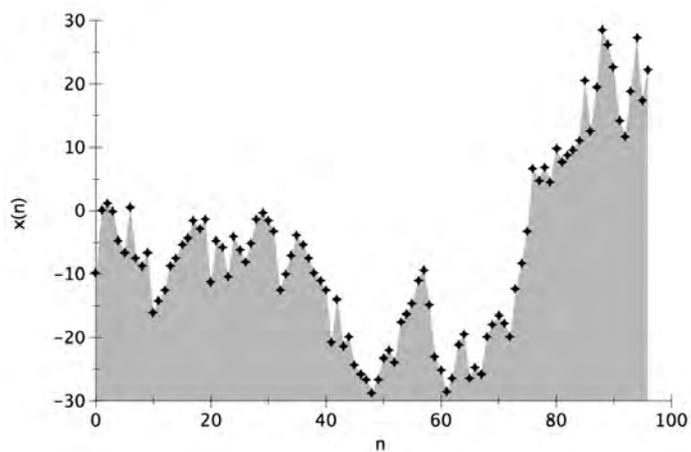


Figura 1.3: Gráfica de la función  $f^n$  en el Ejemplo 3. Se utilizó un  $x_n$  inicial de 0.5 y  $\eta_n$  con una media igual a 50 y una desviación estándar de 10000.

**Definición 4.** Un **atractor** es un conjunto  $\mathcal{A} \subset \mathcal{M}$  tal que

1.  $\mathcal{A}$  es un conjunto invariante,  $f^t(a) = a \forall a \in \mathcal{A}$ .
2.  $\mathcal{A}$  posee una vecindad abierta  $\mathcal{U}$  tal que  $\forall u \in \mathcal{U}$   $f^t(u)$  converge a  $\mathcal{A}$  cuando  $t \rightarrow \infty$ .

Todo SD que posea uno o varios atractores se dice *disipativo* o *no conservativo*.

## 1.2. Reconstrucción del espacio de configuraciones

**Definición 5.** Dado un SD  $\mathcal{F} = \{f^t\}_{t \in \mathcal{K}}$ , llamaremos **línea de tiempo** a un conjunto  $\mathfrak{Y} \subseteq \mathcal{F}(x) = \{f^t(x)\}_{t \in \mathcal{K}}$ .

Tomemos por conocida una serie de tiempo  $\mathfrak{Y} = \{f^{t_0}, f^{t_1}, \dots, f^{t_m}\}$  donde suponemos que  $f^t \in \mathcal{F}$  para algún SD. ¿Es posible inferir algunas propiedades de tal sistema dinámico a partir de la serie de datos  $\mathfrak{Y}$ ? Los primeros resultados en esta clase de reconstrucción fueron publicados por Takens en 1981. De tal trabajo tiene origen el celebre teorema que lleva su nombre:

**Teorema 1. (Takens 1981)** Sea  $\mathcal{M}$  una variedad compacta de dimension  $d$ . Con el par  $(\phi, \varphi)$  donde  $\phi : \mathcal{M} \rightarrow \mathcal{M}$  es un difeomorfismo y  $\varphi : \mathcal{M} \rightarrow \mathbb{R}$  una función suave, es una propiedad genérica (densa y suave) que el mapeo  $\mathcal{H}[\phi, \varphi] : \mathcal{M} \rightarrow \mathbb{R}^{2d+1}$  definido como

$$x \mapsto (\varphi(x), \varphi(\phi(x)), \dots, \varphi(\phi^{2d}(x))) \quad (1.4)$$

es una inmersión, i.e.,  $\mathcal{H}$  es un mapeo uno a uno entre  $\mathcal{M}$  y su imagen, donde  $\mathcal{H}$  y  $\mathcal{H}^{-1}$  son ambas diferenciables.

La demostración no es para nada trivial, Takens la brinda en su artículo [5]. Sin embargo ésta se sustenta en los invariantes topológicos de un SD. Una demostración diferente fue brindada por P. Huke[6] en la cual explica detalladamente la importancia de cada hipótesis en el teorema. Abordar tal demostración nos desviaría de nuestro objetivo por lo que nos limitaremos a explicar de manera intuitiva su significado. Supongamos que solo se puede *medir* una componente del sistema,  $f^t(x)$ , por supuesto esta información es insuficiente para conocer la dinámica real del sistema. Pero si se pudiera medir hasta la  $\rho$ -ésima derivada de  $f^t(x)$ , entonces, si la dimensión del sistema dinámico es menor que  $\rho$  se tendría suficiente información para describir completamente al sistema. Si consideramos un parámetro  $\tau \ll 1$ , podemos aproximar la primera derivada de  $f^t(x)$  como

$$\frac{df^t}{dt} \approx \frac{f^{t+\tau} - f^t}{\tau},$$

de igual manera podemos aproximar cada una de las derivadas posteriores:

$$\frac{d^k f^t}{dt^k} \approx \frac{1}{\tau^k} \left( f^{t+k\tau} + k \sum_{i=1}^{k-1} (-1)^{k+i} f^{t+i\tau} + (-1)^k f^t \right).$$

Por lo que la idea principal de este método recae en que realmente no necesitamos conocer las derivadas para identificar la estructura de las órbitas en el espacio de configuraciones,

sino que podemos utilizar las *coordenadas retardadas*  $f^{t+T}$ , donde  $T \in \mathbb{R}$  es una constante a ser determinada. De esta forma la evolución del sistema dinámico real será aproximada mediante la evolución de

$$\mathcal{Q} = (f^t, f^{t+T}, \dots, f^{t+(e-1)T}), \quad (1.5)$$

donde el Teorema (1) nos indica que  $\varrho > 2d + 1$ . La evolución temporal de este vector estará dado por una función  $\mathcal{H}$ , que en principio puede ser lineal o no lineal. Por lo que tenemos dos cabos sueltos, el valor de *la dimensión de embedding*  $\varrho$  y  $T$  el *tiempo de retraso*<sup>3</sup>.

**Ejemplo 4.** En la gráfica superior de la Figura (4) podemos ver la evolución del SD presentado en el Ejemplo (2). Podemos apreciar la presencia de los llamados *atractores de Lorenz* que indican que el sistema dinámico eventualmente tenderá a un estado del cual no podrá salir. Utilizando el vector  $\mathcal{Q} = (f^t(x), f^{2T}(x), f^{3T}(x))$  con  $T = 0.2$  obtenemos una evolución como la mostrada en la gráfica inferior de la Figura (4). Con esto podemos ver la esencia del teorema de Takens, aunque nuestro sistema reconstruido no es exactamente igual al original, cualitativamente poseen las mismas propiedades.

A continuación presentamos dos formas de estimar el valor de las cantidades  $\varrho$  y  $T$ .

### 1.2.1. El método de los falsos vecinos

En 1995 Nikolai et al.[7] propusieron un método con el nombre de esta sección<sup>4</sup>, fueron los primeros en notar que si la dimensión de embedding era muy pequeña comparada con la dimensión real o *si el sistema dinámico carece de un atractor*, la distancia entre dos puntos cercanos se ve gravemente afectada por el incremento de la dimensión. Por lo que el método consiste en construir un vector  $\varrho$ -dimensional (1.5) y en asociar a cada punto un conjunto de vecinos, i.e, los puntos más cercanos a éste. Después se incrementa la dimensión y se realiza el mismo procedimiento con el vector  $(\varrho + 1)$ -dimensional. Cuando el sistema dinámico está correctamente caracterizado en el espacio  $\varrho$  se espera que las propiedades sean las mismas en el espacio  $\varrho + 1$ . Sin embargo, si  $\varrho$  no es la dimensión correcta en el espacio  $\varrho + 1$  algunos pares de puntos, llamados *falsos vecinos*, que eran cercanos en el espacio  $\varrho$  dejarán de serlo en  $\varrho + 1$ .

Supongamos que el vector  $\mathcal{Q}(t)$  tiene el vecino  $\mathcal{Q}_v(t)$ , la distancia euclidiana entre estos vectores está dada por:

$$\mathcal{R}_\varrho(t)^2 = [f^t - f_v^t]^2 + \dots + [f^{t+(e-1)T} - f_v^{t+(e-1)T}]^2,$$

en la dimensión  $\varrho + 1$  esta distancia es igual a

$$\mathcal{R}_{\varrho+1}(t)^2 = \mathcal{R}_\varrho(t)^2 + [f^{t+eT} - f_v^{t+eT}]^2.$$

Por lo que si

$$\frac{\mathcal{R}_{\varrho+1}(t) - \mathcal{R}_\varrho(t)}{\mathcal{R}_\varrho(t)} > \mathcal{R} \quad (1.6)$$

<sup>3</sup>Formalmente el teorema indica que cualquier valor de  $T$  es válido, pero como veremos más adelante estimar su valor puede mejorar los resultados.

<sup>4</sup>El nombre original es “the false nearest neighbours method”. Aunque en sentido estricto todos los puntos del espacio de configuración son vecinos, aquí nos limitaremos a detonar con esta palabra a los vecinos más cercanos.

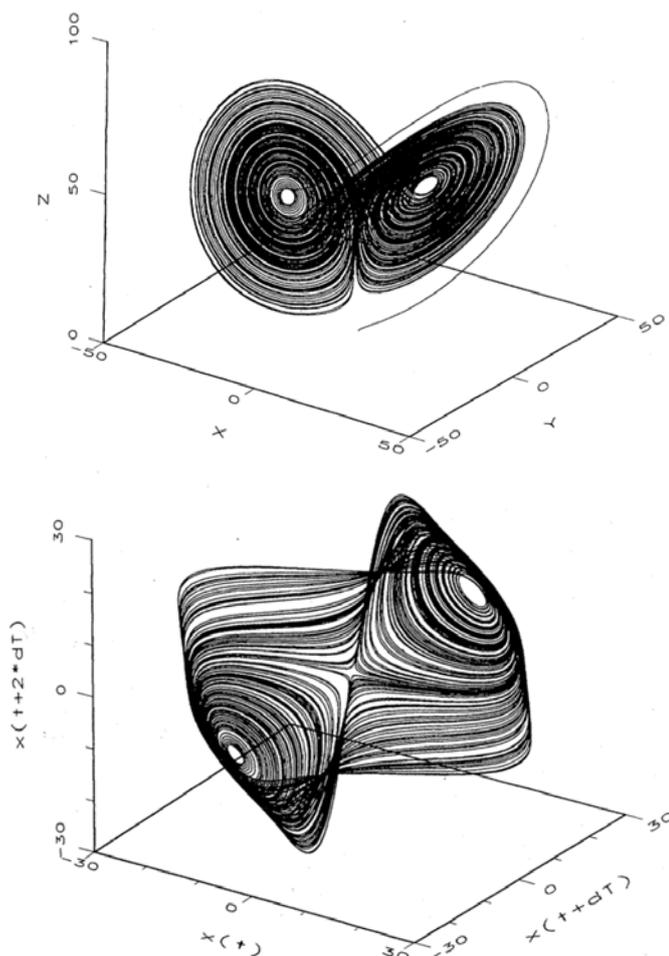


Figura 1.4: En la parte superior el espacio de configuraciones del sistema de Lorenz con los parámetros  $r = 45.92$ ,  $b = 4.0$  y  $\sigma = 16.0$ . La gráfica inferior muestra el sistema reconstruido a través del teorema de Takens.

$\mathcal{Q}_v(t)$  es declarado un falso vecino.  $\mathcal{R}$  denota un incremento relativo de la distancia que es *tolerable*. Realizando este procedimiento para cada  $t$  disponible en nuestra serie podemos ir *contando* el número de falsos vecinos en cada dimensión. Con eso podemos asociar a  $\rho$  la dimensión para la cual el número de falsos vecinos es despreciable (teóricamente debería ser cero).

### 1.2.2. Estimando el tiempo de retraso

Existen dos problemas asociados a la elección de  $T$ . Si escogemos un tiempo de retraso muy corto entonces las coordenadas  $f^{t+nT}$  y  $f^{t+(n+1)T}$  estarán tan próximas entre ellas mismas que nuestros cálculos numéricos *no distinguirán* entre estos dos datos; por otra parte, si  $T$  es demasiado grande puede ocurrir que sean completamente *independientes*. Por lo que necesitamos un criterio que nos ubique en un punto medio. Podemos considerar que los valores de  $f^t$  provienen de una *distribución* de probabilidad desconocida y con esto calcular la *función lineal de autocorrelación*

$$\mathcal{C}(T) = \frac{\frac{1}{N} \sum_{t=1}^N [f^{t+T} - \langle f \rangle][f^t - \langle f \rangle]}{\frac{1}{N} \sum_{t=1}^N (f^t - \langle f \rangle)^2} \quad (1.7)$$

donde

$$\langle f \rangle = \frac{1}{N} \sum_{t=1}^N f^t.$$

Escogiendo  $T$  como el primer cero de  $\mathcal{C}(T)$  garantizamos que en promedio  $f^t$  y  $f^{t+T}$  sean *linealmente independientes*, lo cual puede ser un buen criterio para la elección de este parámetro.

Si buscamos algo un poco más general, i.e., que denote independencia estadística (lineal, no lineal, *parabólica*,...,etc.) es conveniente recurrir a teoría de la información [8].

**Definición 6. (Entropía de Shannon)** Sean  $z$  y  $w$  variables aleatoria discretas,  $z, w : \Omega \rightarrow E$  donde  $\Omega$  es el espacio de probabilidad asociado y  $E$  un conjunto medible. La entropía de Shannon está dada por

$$H(z) = - \sum_{i=1} p_i(z) \log(p_i(z)),$$

donde  $p_i(z)$  denota la probabilidad  $p(z = z_i)$ . Así mismo la entropía conjunta de  $z$  y  $w$  es

$$H(z, w) = - \sum_{i,j} p_{i,j} \log(p_{i,j})$$

donde  $p_{i,j}$  denota la probabilidad  $p(z = z_i, w = w_j)$ .

**Definición 7. (Entropía condicional)** Sea  $z$  y  $w$  dos variables aleatorias, la entropía condicional de Shannon se define como:

$$H(z|w) = H(z, w) - H(w) = - \sum_{i,j} p_{i,j} \log(p_{i,j}),$$

donde  $p_{i|j} = p(z = z_i | w = w_j)$ .

**Definición 8. (Información mutua)** La información mutua entre  $z$  y  $w$  es igual a

$$I(z, w) = H(z) + H(w) - H(z, w)$$

**Proposición 2.** Si  $z$  y  $w$  son dos variables aleatorias estadísticamente independientes, entonces  $I(z, w) = 0$ .

*Demostración.* De la definición de entropía condicional tenemos

$$\begin{aligned} I(z, w) &= H(z) - H(z|w) \\ &= -\sum_i p_i(z) \log(p_i(z)) + \sum_{i,j} p_{i,j} \log(p_{i|j}) \\ &= -\sum_{i,j} p_{i,j} \log(p_i(z)) + \sum_{i,j} p_{i,j} \log(p_{i|j}) \\ &= \sum_{i,j} p_{i,j} \log\left(\frac{p_{i|j}}{p_i(z)}\right) \\ &= \sum_{i,j} p_{i,j} \log\left(\frac{p_{i,j}}{p_i(z)p_j(w)}\right) \end{aligned}$$

Donde se utilizó que  $\sum_j p_{i,j} = p_i(z)$ . De esta última expresión es claro que si  $p_{i,j} = p_i(z)p_j(w)$  entonces  $I(z, w) = 0$ . ■

Si implementamos  $z = f^t(x)$  y  $w = f^{t+T}(x)$  entonces

$$I(\mathcal{T}) = \sum_{t=1}^n p(f^t, f^{t+T}) \log\left(\frac{p(f^t, f^{t+T})}{p(f^t)p(f^{t+T})}\right) \quad (1.8)$$

Donde se pueden utilizar las *probabilidades a priori* para el cálculo. Fraser y Swinney[9] sugieren que debemos utilizar  $T = \mathcal{T}$  para el primer *mínimo* de la función  $I(\mathcal{T})$ . De esta forma  $f^t$  y  $f^{t+T}$  son independientes pero no estadísticamente, es decir,

$$p(f^t, f^{t+T}) \neq p(f^t)p(f^{t+T}).$$

Cuando  $I(\mathcal{T})$  no tiene un mínimo bien definido Fraser and Swinney sugieren emplear un  $T = \mathcal{T}$  tal que

$$\frac{I(\mathcal{T})}{I(0)} \approx \frac{1}{5}.$$

### 1.3. El algoritmo genético

En 1975 Holland publicó un libro llamado “*Adaptation in Natural and Artificial Systems*” [10] donde presentó el llamado *algoritmo genético* como una abstracción de un proceso evolutivo biológico. El método de Holland pasa de un conjunto de ecuaciones (*cromosomas* en el sistema biológico) a un nuevo conjunto utilizando una clase de *selección natural*, junto con unas operaciones llamadas *intercambio*, *mutación* y *ordenamiento*. La operación de selección decide que cromosomas tienen permitido reproducirse, la operación de intercambio combina partes de diferentes cromosomas, la mutación cambia de

manera aleatoria algunas propiedades de los nuevos cromosomas y el orden permite distinguir que cromosomas tienen más probabilidades de reproducirse. Todo con el fin de obtener un sistema biológicamente *estable*, tal como ocurre en la naturaleza. A continuación discutiremos a detalle este algoritmo para su implementación a series de tiempo.

Supongamos que tenemos la serie de tiempo  $f^1, f^2, \dots, f^m$  y deseamos determinar la dependencia del valor  $f^t$  con los valores previos  $f^\lambda \forall \lambda < t$ . El algoritmo comienza con población inicial de  $N$  ecuaciones de cadena (EC), las cuales consisten en dos argumentos relacionados por una operación aritmética. Dichos argumentos pueden ser números reales o elementos de nuestra serie de tiempo. Una ecuación de cadena puede ser también un argumento de otra ecuación de cadena, de tal suerte que estas expresiones serán los cromosomas con lo que operará el algoritmo. Los pasos que componen el algoritmo genético son los siguientes:

1. *Inicialización.* Las  $N$  EC son generadas de la forma:

$$S_j = (A \odot B) \odot (C \odot D),$$

donde  $A, B, C$  y  $D$  son números reales o elementos de la serie de tiempo,  $f^{t-\lambda}$ . El símbolo  $\odot$  representa alguna de las 4 operaciones aritméticas: suma, resta, multiplicación o división. Estas operaciones son asignadas de manera aleatoria y junto con el parámetro  $\lambda$  se distribuirán de manera *uniforme*. Para evitar la división entre cero implementaremos lo siguiente:

$$A \nabla \cdot B = \frac{A}{\text{Max}(0.001, |B|) \text{sgn}(B)},$$

donde  $\text{sgn}(B)$  devuelve el signo de  $B$ .

2. *Calculando la aptitud.* Cada EC  $j$  es utilizada para estimar cada valor  $f^t$  en función de  $f^\lambda$ :

$$f_j^t = S_j(f^{t-1}, f^{t-2}, \dots, f^1), \quad j \in [1, N] \quad (1.9)$$

del mismo modo calculamos el error asociado con esta estimación:

$$F_j^2 = \sum_{t=1}^m (f_j^t - f^t)^2. \quad (1.10)$$

Mientras menor sea  $F$  más apto será nuestra EC. Como medida general, tomamos

$$R_j^2 = 1 - \frac{F_j^2}{\sum_{t=1}^m (f^t - \langle f \rangle)^2},$$

como el porcentaje de la varianza de la EC  $j$ ,  $\langle f \rangle$  representa el promedio de la serie de tiempo original.

3. *Ordenando las EC.* Las EC son ordenadas de forma ascendente según su aptitud.
4. *Elección de compañero.* En este paso se generan pares de EC. La EC con mejor ajuste (menor  $F$  y  $R^2$ ) es la primera en elegir compañero, debe elegir entre las EC restantes. Cada EC tiene una probabilidad de ser elegida de acuerdo con su aptitud. El proceso se repite hasta formar  $N/4$  parejas, las EC restantes son eliminadas.

5. *Reproducción e intercambio.* Éste es quizá el paso más importante del algoritmo. Cada una de las  $N/4$  parejas tendrá 4 crías (así la *población* se mantiene constante). Ahora definimos el método con que la *reproducción* transmite *información genética* a la siguiente generación. Las dos primeras crías son idénticas a sus *padres*. Las otras dos crías se forman a través de una combinación de sus padres: de manera aleatoria, estas EC contendrán elementos de sus padres. Por ejemplo asumamos lo siguiente:

$$\begin{aligned}\text{Padre 1} & : (A \cdot B)\nabla \cdot C, \\ \text{Padre 2} & : (D - (E\nabla \cdot F)),\end{aligned}$$

Las crías 3 y 4 podrían ser iguales a

$$\begin{aligned}\text{Cría 3} & : (A \cdot (E\nabla \cdot F))\nabla \cdot C \\ \text{Cría 4} & : (D - B),\end{aligned}$$

donde los elementos  $(E\nabla \cdot F)$  y  $B$  fueron intercambiados.

6. *Mutación.* Se elige aleatoriamente un conjunto de EC y se altera alguna de sus componentes, un elemento o una operación aritmética de igual forma que en el paso 1. Las primeras EC de nuestra lista ordenada se ven exentas de esta mutación, la cual pretende mejorar la aptitud.

Los pasos 2-6 son repetidos cierto número de veces, llamadas *generaciones*. El algoritmo puede terminar por ejemplo, cuando  $R_j^2$  no disminuya más. Al final del proceso la EC en el primer puesto del paso 3 es elegida como la ecuación de ajuste a los datos y puede ser implementada para predicciones futuras.

## 1.4. La Transformación de wavelets

La *transformación de wavelet* fue propuesta por el matemático Alfred Haar en 1909 [11], aunque el término *wavelet* (onda pequeña) fue inventado por el geofísico Jean Morlet y el físico Alex Grossman en 1984 [12]. Las ondas pequeñas fueron introducidas para una *representación* de funciones más eficiente que la serie de Fourier.

Antes de abordar las definiciones formales acerca de los wavelets daremos una breve introducción a las ideas que sustentan esta clase de transformación. Supongamos la serie de tiempo  $\mathfrak{Y}$  de dimensión  $n = 2^J$  para algún<sup>5</sup>  $J \in \mathbb{N}$ . Si deseamos conocer el grado de *variación* entre los elementos de la serie, es decir, que tan lejos está uno del otro podemos emplear la siguiente ecuación:

$$d_k = y_{2k} - y_{2k-1}, \quad k = 1, \dots, n/2 \quad (1.11)$$

donde  $y_k \in \mathfrak{Y}$ . Si  $y_{2k}$  es muy *similar* a  $y_{2k-1}$  entonces  $d_k$  será muy pequeño; si son exactamente iguales entonces  $d_k$  será cero y si son muy diferentes  $d_k$  tendrá una magnitud muy

---

<sup>5</sup>Tomar  $n = 2^J$  no es un factor restrictivo, puesto que el análisis posterior puede hacerse para cualquier  $n$ . Comúnmente se refiere en la literatura como una *secuencia diádica* a aquellas que cumplen con dicha igualdad.

grande. Esta forma de análisis, que puede llegar a parecer trivial, cobrará una importancia vital más adelante.

La secuencia  $\{d_k\}_{k=1}^{n/2}$  contiene la diferencia entre pares sucesivos en la serie de tiempo  $\mathfrak{Y}$  y nos permite conocer que tan variados son nuestros datos, sin embargo, un análisis más profundo demandaría calcular diferencias entre elementos no consecutivos. Si ahora empleamos

$$c_k = y_{2k} + y_{2k-1}, \quad k = 1, \dots, n/2$$

en esta ocasión la secuencia  $\{c_k\}_{k=1}^{n/2}$  es un conjunto de semipromedios (*semi* por carecer del factor  $1/2$ ) y cada elemento  $c_k$  contiene información proveniente de ambos elementos  $y_{2k}$  y  $y_{2k-1}$ . Por lo que en  $c_k$  estamos teniendo un grado de *detalle* más *grueso* que en  $\{d_k\}$ . Es por esto que se suele referir a  $d_k$  como el parámetro *de detalle fino*. Es necesario cuantificar de alguna forma el grado de detalle que estamos teniendo en nuestro análisis, por lo que introduciremos el subíndice  $j$ . La serie de tiempo original  $\mathfrak{Y}$  consta de  $2^J$  *observaciones* y  $\{d_k\}$  de  $2^{J-1}$  elementos, por lo que para el primer grado de detalle  $j = J - 1$ . Ahora referiremos a  $d_k$  como  $d_{J-1,k}$  y a  $c_k$  como  $c_{J-1,k}$  para denotar la escala de detalle<sup>6</sup>. Para obtener una escala de detalle mayor repetimos el proceso de la siguiente manera:

$$d_{J-2,\ell} = c_{J-1,2\ell} - c_{J-1,2\ell-1}, \quad \ell = 1, \dots, n/4$$

Por ejemplo, si  $\ell = 1$  entonces  $d_{J-2,1} = (y_4 + y_3) - (y_2 + y_1)$ , por lo que  $d_{J-2,\ell}$  es un conjunto de diferencias entre el semipromedio de dos de nuestros datos. Con ello  $d_{J-1,\ell}$  y  $d_{J-2,\ell}$  representan diferencias a *dos escalas* diferentes. De manera análoga tenemos:

$$c_{J-2,\ell} = c_{J-1,2\ell} + c_{J-1,2\ell-1}, \quad \ell = 1, \dots, n/4.$$

Repitiendo este procedimiento producimos parámetros a escalas cada vez más gruesos, a su vez el número de elementos en las series  $\{d_{j,k}\}$  y  $\{c_{j,k}\}$  se ve reducido por un factor de  $1/2$  cada vez que aumentamos el nivel de detalle. Un diagrama representativo del proceso puede apreciarse en la Figura (1.5).

**Ejemplo 5.** Supongamos que tenemos la siguiente línea de tiempo  $\mathfrak{Y} = (1, 1, 7, 9, 2, 8, 8, 6)$ , donde  $n = 8$  y por lo tanto  $J = 3$ . Primero aplicamos la ecuación (1.11) para obtener, por ejemplo,  $d_{2,1} = y_2 - y_1 = 1 - 1 = 0$ . De tal suerte que obtendremos un diagrama del tipo *piramide* como el mostrado en la Figura (1.5).

Hasta el momento hemos evitado el uso de la palabra *wavelet* en la descripción del *algoritmo de multi-escala*, sin embargo, los coeficientes de detalle  $d_{j,k}$  son los coeficientes del *wavelet madre* y los  $c_{j,k}$  los coeficientes de la *función padre* que definiremos a continuación. Así que el algoritmo que explicamos anteriormente es una clase de *transformación de wavelet discreta* y el algoritmo piramidal de wavelets se debe a Mallat (1989b). Los wavelets detrás de esta transformación son llamados los wavelets de Haar (1910).

**Definición 9.** Una función  $\psi(t)$  que satisfice

$$\int_{\mathbb{R}} \psi(t) dt = 0$$

<sup>6</sup>De aquí en adelante siempre se conocerá como *escala* al factor  $2^j$  y *nivel* al índice  $j$

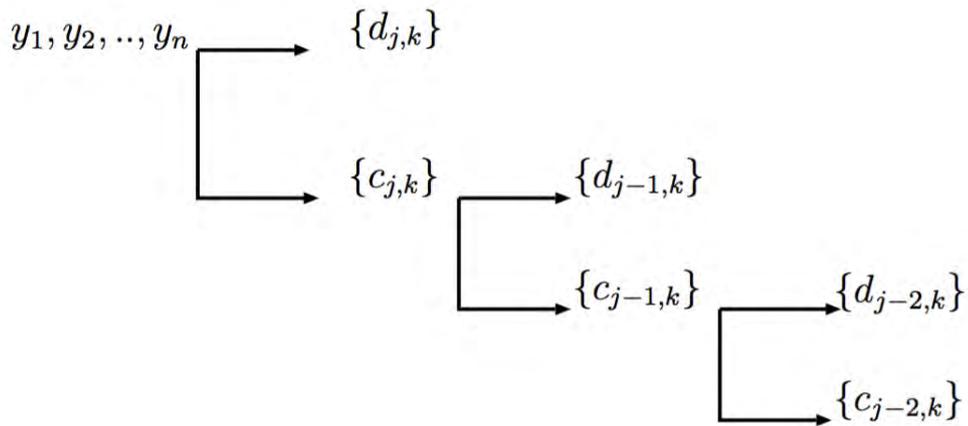


Figura 1.5: Diagrama representativo de la generación de parámetros de detalle. Como podemos ver el conjunto  $\{c_{j,k}\}$  es transformado en dos conjuntos  $\{d_{j-1,k}\}$  y  $\{c_{j-1,k}\}$  donde a su vez este último sirve como entrada para generar dos nuevos conjuntos.

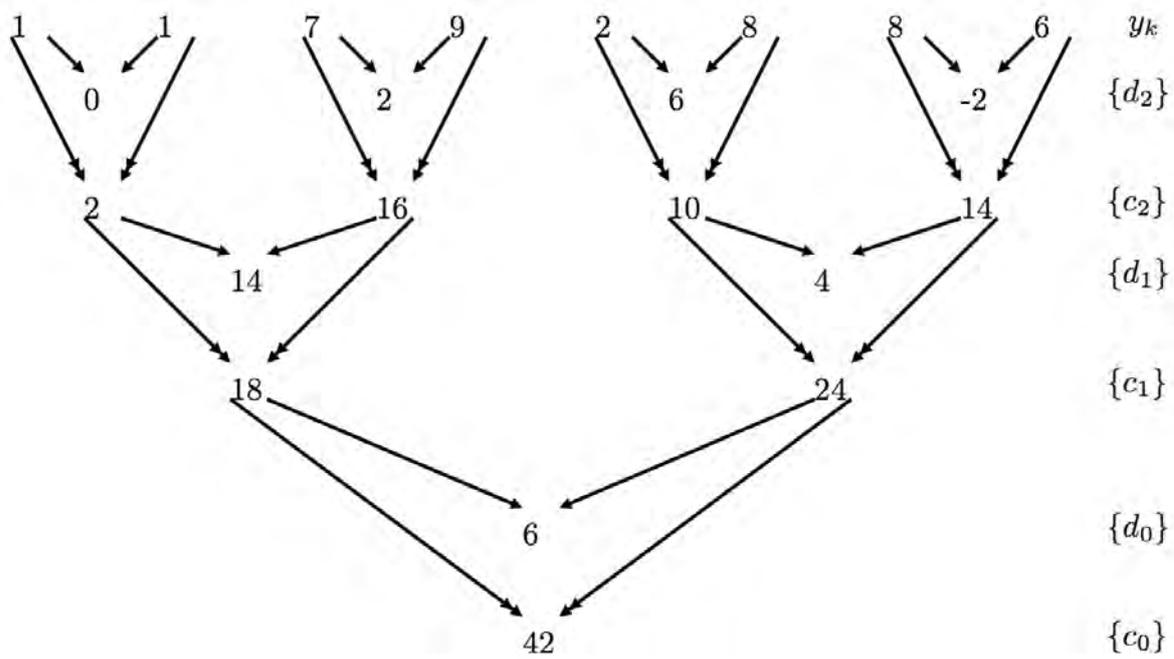


Figura 1.6: Diagrama del Ejemplo 5. Podemos apreciar que el número de elementos en cada conjunto  $\{d_{j,k}\}$  y  $\{c_{j,k}\}$  se ve reducido a la mitad en cada iteración. Esta serie de coeficientes contiene información sobre la secuencia inicial  $\mathfrak{Y}$ , por ejemplo, el elemento 42 nos indica que la suma de la serie inicial es 42, mientras que el 18 nos indica que la suma de los primeros cuatro elementos de la serie es 18. El valor de esta suma cobra significado cuando se conoce el origen de la serie de tiempo.

representa un **wavelet**, puesto que esta condición implica que  $\psi$  cambia de signo en  $\mathbb{R}$  y que su soporte es compacto. Por lo tanto, los wavelets son funciones matemáticas que oscilan, pero decaen muy rápido.

**Definición 10.** La función padre de Haar está dado por:

$$\phi(t) = \begin{cases} 1 & t \in [0, 1] \\ 0 & \text{c.o.c} \end{cases} \quad (1.12)$$

Claramente

$$\int_{\mathbb{R}} \phi(t) dt = 1$$

Por lo que  $\phi(t)$  no es wavelet, pero es una función que será de gran ayuda en nuestro análisis.

**Proposición 3.** Definimos ahora

$$\phi_{J,k}(t) = 2^{J/2} \phi(2^J t - k) = \begin{cases} 2^{J/2} & t \in [2^{-J}k, 2^{-J}(k+1)] \\ 0 & \text{c.o.c.} \end{cases}$$

donde el factor  $2^{J/2}$  es una constante de normalización. Con lo que  $\{\phi_{J,k}\}$  es un conjunto completo y es posible expandir una función  $f(t)$  como una combinación lineal<sup>7</sup>:

$$f(t) \approx f_J(t) = \sum_{k=0}^{2^J-1} c_{j,k} \phi_{J,k}(t), \quad (1.13)$$

a través de coeficientes  $c_{j,k}$  que están dados por:

$$c_{J,k} = \int_{\mathbb{R}} f(t) 2^{J/2} \phi(2^J t - k) dt = \langle f | \phi_{J,k} \rangle \quad (1.14)$$

donde  $\langle \cdot, \cdot \rangle$  denota producto interno. En la literatura a  $\phi_{J,k}$  se le llama el el wavelet padre de Haar, aunque en sentido estricto no sea un wavelet.

*Demostración.* Si consideramos la transformada de Fourier:

$$\hat{f}\{\phi_{J,k}\} = \frac{i2^{(J-1)/2} \exp[i\omega 2^{-J}] (\exp[i\omega 2^{-J}] - 1) (\mathfrak{L}(-2^{-J}) - 1)}{\sqrt{\pi\omega}},$$

donde  $\mathfrak{L}(t) = 1 \forall t \in \mathbb{R}^+ \cup \{0\}$  y  $\mathfrak{L}(t) = 0$  para cualquier otro caso. Por simplicidad considerando el caso  $J = 0$ , tenemos que

$$\begin{aligned} \langle \hat{f}\{\phi_{0,k}(t)\} | \hat{f}\{\phi_{0,m}(t)\} \rangle &= \int_{\mathbb{R}} \exp(\{k-m\}i\omega) \frac{1 - \cos \omega}{\pi\omega^2} d\omega \\ &= \delta_{k,m} \end{aligned}$$

Gracias al teorema de Parseval  $\langle \phi_{0,k} | \phi_{0,m} \rangle = \delta_{k,m}$ . ■

<sup>7</sup>Escribimos una aproximación a la función  $f(t)$  puesto que la igualdad se obtiene únicamente al tomar  $J$  tal que  $n = 2^J$ , para una  $J$  menor obtendremos una función diferente.

**Proposición 4.** *Los coeficientes  $c_{J-1,k}$  cumplen con la siguiente relación de recurrencia:*

$$c_{J-1,k} = 2^{-1/2}(c_{J,2k} + c_{J,2k+1}) \quad (1.15)$$

*Demostración.* Operando de manera directa:

$$\begin{aligned} c_{j-1,k} &= \int_{2^{-(j-1)k}}^{2^{-(j-1)(k+1)}} f(t)\phi_{j-1,k}(t)dt \\ &= 2^{-1/2} \left\{ \int_{2^{-j}2k}^{2^{-j}(2k+1)} f(t)2^{j/2}\phi(2^j t - 2k)dt + \int_{2^{-j}(2k+1)}^{2^{-j}(2k+2)} f(t)2^{j/2}\phi(2^j t - (2k+1))dt \right\} \\ &= 2^{-1/2} \left\{ \int_{2^{-j}2k}^{2^{-j}(2k+1)} f(t)\phi_{j,2k}(t)dt + \int_{2^{-j}(2k+1)}^{2^{-j}(2k+2)} f(t)\phi_{j,2k+1}(t)dt \right\} \\ &= 2^{-1/2}(c_{j,2k} + c_{j,2k+1}) \end{aligned}$$

donde se utilizó que  $\phi(t) = \phi(2t) + \phi(2t-1)$  que puede apreciarse de la definición misma. ■

Supongamos ahora que tenemos dos aproximaciones distintas sobre la función  $f(t)$ , por ejemplo,  $f_0(t)$  y  $f_1(t)$  donde

$$\begin{aligned} f_0(t) &= c_{0,0}\phi(t) \\ f_1(t) &= c_{1,0}\phi_{1,0}(t) + c_{1,1}\phi_{1,1}(t) \end{aligned}$$

¿Cuál es la diferencia entre  $f_0(t)$  y  $f_1(t)$ ? La diferencia es el “detalle” perdido al pasar de la representación  $f_1$  a  $f_0$ . Matemáticamente:

$$\begin{aligned} f_1(t) - f_0(t) &= c_{0,0}\phi(t) - 2^{1/2}\{c_{1,0}\phi(2t) + c_{1,1}\phi(2t-1)\} \\ &= c_{0,0}\{\phi(2t) + \phi(2t-1)\} - 2^{1/2}\{c_{1,0}\phi(2t) + c_{1,1}\phi(2t-1)\} \end{aligned}$$

Con lo que

$$f_1(t) - f_0(t) = (c_{0,0} - 2^{1/2}c_{1,0})\phi(2t) + (c_{0,0} - 2^{1/2}c_{1,1})\phi(2t-1),$$

empleando la ecuación (1.15),  $c_{0,0} = (c_{1,0} + c_{1,1})/\sqrt{2}$ , tenemos

$$f_1(t) - f_0(t) = \{(c_{1,1} - c_{1,0})\phi(2t) + (c_{1,0} - c_{1,1})\phi(2t-1)\}/\sqrt{2}.$$

Si

$$d_{0,0} \stackrel{\text{def}}{=} (c_{1,1} - c_{1,0})/\sqrt{2}$$

entonces

$$f_1(t) - f_0(t) = d_{0,0}\{\phi(2t) - \phi(2t-1)\} \quad (1.16)$$

**Definición 11.** *El wavelet madre de Haar está dado por*

$$\begin{aligned} \psi(t) &= \phi(2t) - \phi(2t-1) \\ &= \begin{cases} 1 & t \in [0, 1/2) \\ -1 & t \in [1/2, 1) \\ 0 & \text{c.o.c} \end{cases} \end{aligned}$$

Claramente

$$\int_{\mathbb{R}} \psi(t) dt = \int_0^{1/2} dt - \int_{1/2}^1 dt = 0$$

Note que el wavelet de Haar está localizado en el tiempo, i.e., es cero fuera de un intervalo definido, pero no es continuo.

En vista de la última definición podemos reescribir la ecuación (1.16) como

$$f_1(t) - f_0(t) = d_{0,0}\psi(t).$$

Con lo que obtenemos un resultado sumamente importante

$$f_1(t) = c_{0,0}\phi(t) + d_{0,0}\psi(t).$$

En otras palabras, la aproximación al nivel 1 puede ser obtenida por una aproximación al nivel cero más el detalle *encapsulado* en  $d_{0,0}$ . Esto puede generalizarse para cualquier nivel de detalle, por que la *transformación de wavelet discreta* está dada por:

$$f_{J+1}(t) = \sum_{k=0}^{2^J-1} c_{J,k}\phi_{J,k}(t) + \sum_{k=0}^{2^J-1} d_{J,k}\psi_{J,k}(t). \quad (1.17)$$

Donde  $\psi_{J,k} = 2^{J/2}\psi(2^J t - k)$  y  $d_{J,k} = 2^{-1/2}(c_{J+1,2k+1} - c_{J+1,2k})$ . Al comienzo de la sección introducimos los coeficientes de detalle fino como la diferencia entre elementos consecutivos de la serie, utilizando el wavelet de Haar definimos un concepto similar que considera diferencias entre integrales definidas en intervalos consecutivos.

En este capítulo comenzamos mostrando que un comportamiento irregular puede adjudicarse a correlaciones no lineales. Hemos visto que de existir estas correlaciones es posible inferir algunas propiedades *globales* del sistema. Se habló del teorema de Takens, que sustentado en los invariantes topológicos de un sistema dinámico, nos permite determinar propiedades cualitativas a través de la evolución de un *sistema equivalente*. La obtención de éste último demanda inferir algunos parámetros. Se discutió sobre como proponer un tiempo de retraso apropiado y como mediante el método de los falsos vecinos es posible determinar la dimensión en la que vive nuestro sistema. Posteriormente se describió los pasos que componen el algoritmo genético, el cual será utilizado como herramienta de ajuste para la obtención de un modelo predictivo. Finalmente discutimos a detalle la transformación de wavelet, la cuál permite realizar un *semi-suavizado* de la información a fin de ayudar al algoritmo genético a encontrar un buen ajuste sin perder información del sistema.

# Series de tiempo económicas irregulares

---

En esta sección del trabajo nos enfocaremos en la aplicación de la teoría del capítulo anterior en una serie de tiempo económica irregular, en particular, la evolución temporal del producto interno bruto mexicano. Todos los métodos que serán aplicados demandan su respectiva programación por lo que se eligió desarrollar cada una de estas rutinas en **Wolfram Mathematica 10**, aquí nos enfocaremos en presentar los resultados obtenidos para su posterior análisis por lo que los códigos pueden consultarse en la sección de anexos. A modo de resumen presentamos una lista descriptiva de la metodología que será desarrollada a continuación:

1. Realizamos la transformación de Wavelet para obtener una línea de tiempo con menos variaciones, pero *sin perder* las propiedades generales del sistema, i.e, un semi-suavizado de la información. Se realizará una comparación entre los resultados obtenidos con y sin transformación de wavelet.
2. Aunque el teorema de Takens nos indica que cualquier tiempo de retraso puede servir para nuestros propósitos, propondremos un valor basándonos en los criterios ya explicados.
3. Utilizando el método de los falsos vecinos calcularemos la dimensión de embedding.
4. Una vez que se conoce la forma del vector  $\mathcal{Q}$  podemos reconstruir el espacio de configuraciones del sistema donde vive  $\mathfrak{N}$  para identificar atractores o alguna otra propiedad global del sistema.

5. Empleando el algoritmo genético obtendremos la función  $\mathcal{H}$  que dicta la evolución temporal del vector  $\mathcal{Q}$  con el fin de poder hacer las predicciones concernientes.
6. Puesto que no se utilizarán todos los datos disponibles de las series económicas, podremos comparar nuestra predicción con los datos reales.

## 2.1. Evolución del Producto Interno Bruto

El *producto interno bruto* (PIB) es una magnitud *macroeconómica* que expresa el valor monetario de la producción de bienes y servicios de un país durante un periodo determinado de tiempo. El PIB es uno de los indicadores económicos más importantes de las cuentas nacionales debido a que representa una medida amplia de la actividad económica y proporciona señales de la *dirección* general de la económica. El PIB no se mantiene constante y su cambio está dictado por diversos factores, económicos y *no económicos*. Dentro de las razones económicas se encuentran, por ejemplo, los cambios en políticas públicas como la política fiscal y la política monetaria. Sin embargo, resulta verdaderamente imposible contemplar todos los factores que intervienen en su evolución por lo que no es posible obtener una ley dinámica que nos permita hacer predicciones sobre su comportamiento. En la literatura hay una gran cantidad de modelos y técnicas que buscan analizar estas series de tiempo, la mayoría con un mismo fin, extraer la mayor cantidad de *información* del sistema de tal suerte que podamos inferir comportamientos futuros o inclusive *comprender* las tendencias actuales. El principal problema radica en la conducta tan *irregular* de este tipo de series. Si analizamos la variación porcentual del PIB mexicano (Figura 2.1) podemos ver tal comportamiento. Es por esto que emplearemos esta serie de tiempo como el vector  $\mathfrak{Y}$  para hacer uso de la teoría previamente descrita y buscar un modelo que nos permita hacer predicciones.

### 2.1.1. Transformación de Wavelet

Al emplear 64 datos de la serie de tiempo (Anexos Tabla 4.2)  $\mathfrak{Y}$  tenemos que  $J = 6$  puesto que  $n = 2^6$ , con lo que tendremos 6 niveles de escalamiento. Cada escala representa una aproximación diferente de  $\mathfrak{Y}$ , en la Figura (2.3) mostramos la serie de tiempo resultante para  $J = 1$  y  $J = 6$ . El procedimiento para obtener tales curvas comienza calculando todos los coeficientes  $c_{J,k}$  y  $d_{J,k}$  para un grado de escala dado, después se toma  $d_{J,k} = 0 \forall k$  y se aplica una transformación inversa para obtener la línea de tiempo sin fluctuaciones. Como podemos ver conforme la escala aumenta el *suavizado* de la curva es mayor. Esto se debe a que los coeficientes  $d_{5,k}$  contienen los cambios con mayor *frecuencia* en la serie de tiempo. Una comparación más detallada de todos los coeficientes  $d_{J,k}$  y  $c_{J,k}$  puede apreciarse en la Figura (2.2), donde podemos notar que conforme  $J$  aumenta los elementos de  $\{d_{J,k}\}$  son en su mayoría positivos, mientras que para valores más grandes de  $J$ , los elementos de  $\{c_{J,k}\}$  se mantienen en un rango definido mostrando únicamente grandes *picos* para  $J = 5, 4$  y  $3$ . Un elemento positivo en el conjunto  $\{d_{J,k}\}$  habla de una tendencia de crecimiento por parte de la serie de tiempo (puesto que esto implica que  $c_{2k} < c_{2k+1}$ ), por el contrario un elemento negativo nos habla de un descenso. El conjunto  $\{d_{5,k}\}$  (Anexos Tabla 4.1)

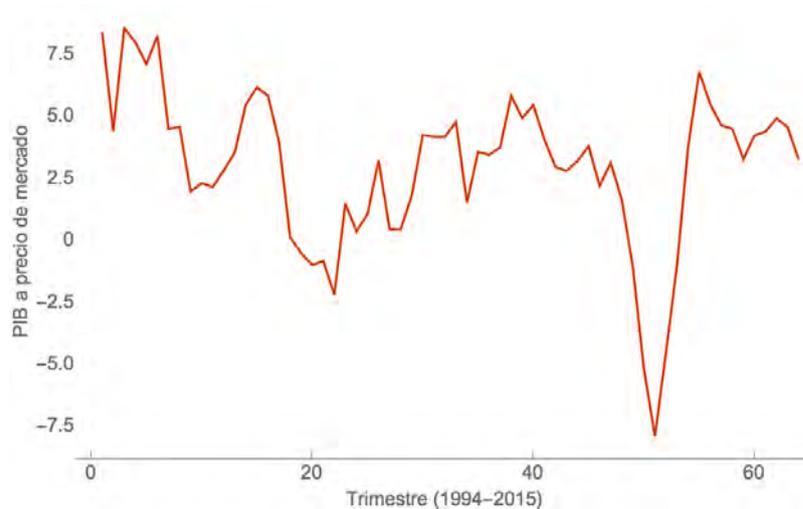


Figura 2.1: Variación porcentual del PIB mexicano por trimestre. La gráfica consta de 64 puntos que representan la variación porcentual trimestral desde el último trimestre de 1996 hasta el último del 2015. Datos obtenidos del Banco de Información Económica del INEGI a través de su sitio web <http://www.inegi.org.mx/sistemas/bie/>.

consta de 17 elementos negativos y 20 positivos, por lo que en un plazo de 64 trimestres (1996-2012) el PIB presentó un mayor número de crecimientos (comparando años consecutivos), más aún, que el único elemento de  $\{d_{0,k}\}$  sea positivo indica que de manera general el PIB presentó un crecimiento. Esto significa que la *magnitud* total de crecimiento fue mayor. Gracias a que los coeficientes de detalle fino  $\{d_{J,k}\}$  capturan la información relacionada a las fluctuaciones en la serie de tiempo, analizando al conjunto  $\{d_{5,k}\}$  podemos notar que la *diferencia* más grande entre elementos consecutivos tiene una magnitud aproximada de 4.94, después le sigue 2.49 y 1.91. Tomando el promedio de este conjunto de valores obtenemos el valor 1.24824, por lo que esta es nuestra primera propuesta para el parámetro  $\mathcal{R}$  en el método de los falsos vecinos. Es importante mencionar que la línea de tiempo obtenida mediante la transformación de wavelet será utilizada únicamente en la implementación del algoritmo genético, es decir, todo el análisis se realizará con base en la serie de tiempo original y este semi-suavizado de la información pretende únicamente *ayudar* al algoritmo genético a encontrar una función que describa los datos de una manera más sencilla.

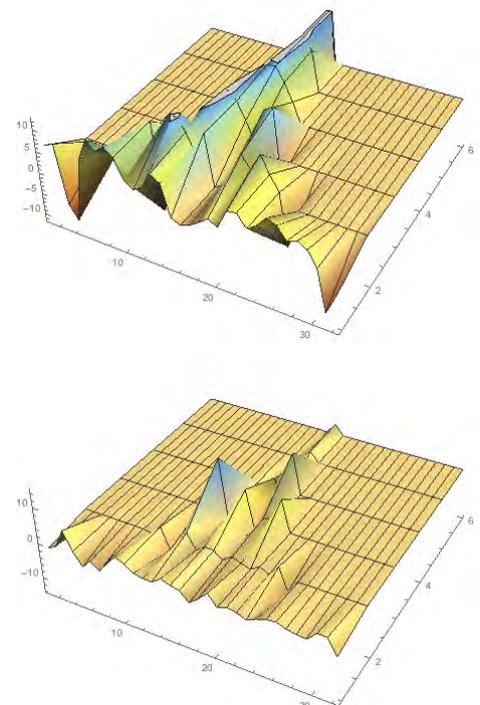


Figura 2.2: La gráfica superior muestra los conjuntos  $\{d_{J,k}\}$ , la gráfica inferior a los conjuntos  $\{c_{J,k}\}$  generados en la transformación discreta de wavelet.

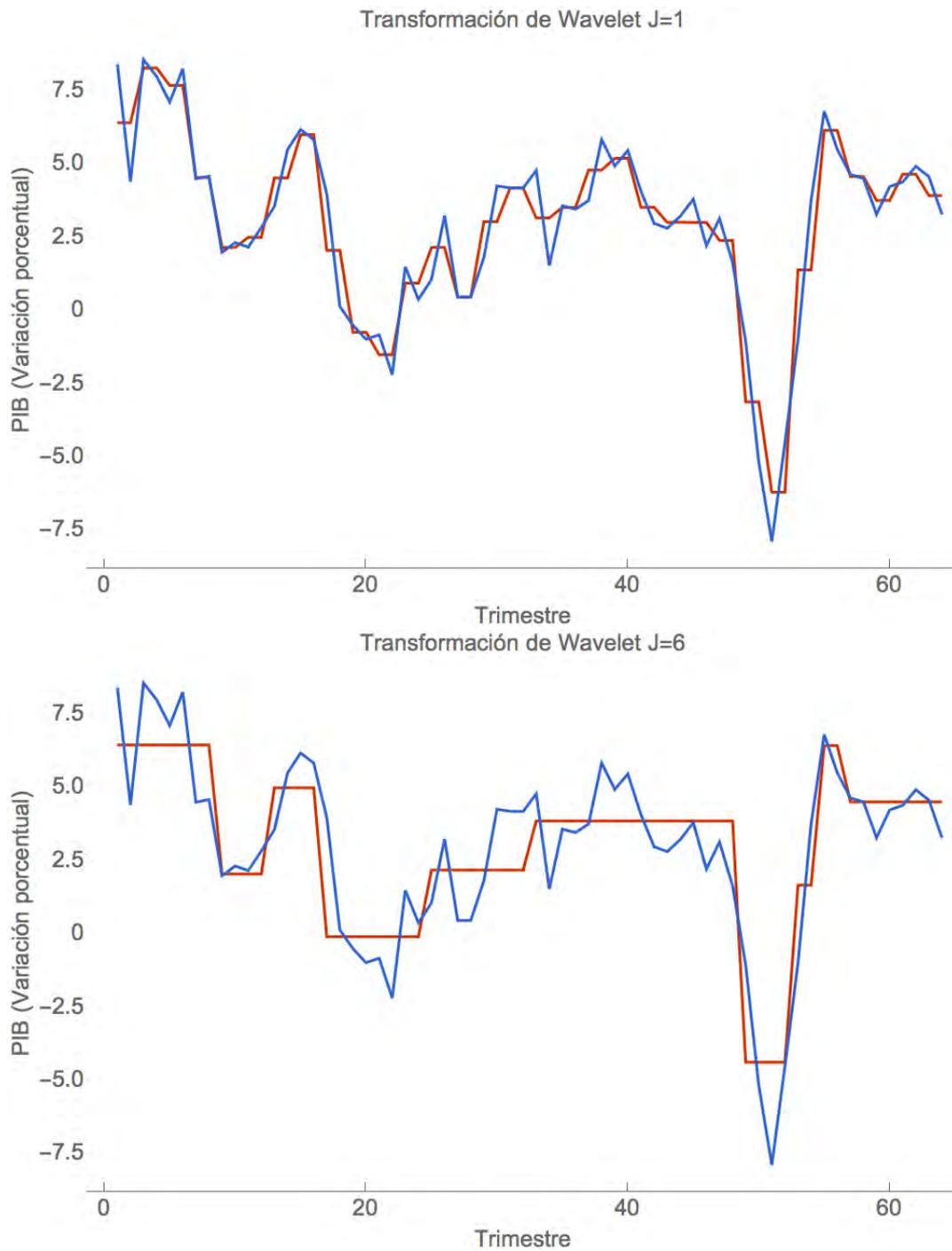


Figura 2.3: Gráficas que muestran la transformación de wavelet (línea roja) a escalas  $J = 1$  y  $J = 6$  a la serie de tiempo del PIB (línea azul). Como podemos observar para  $J = 6$  tenemos una menor cantidad de fluctuaciones en comparación con la curva correspondiente a  $J = 1$

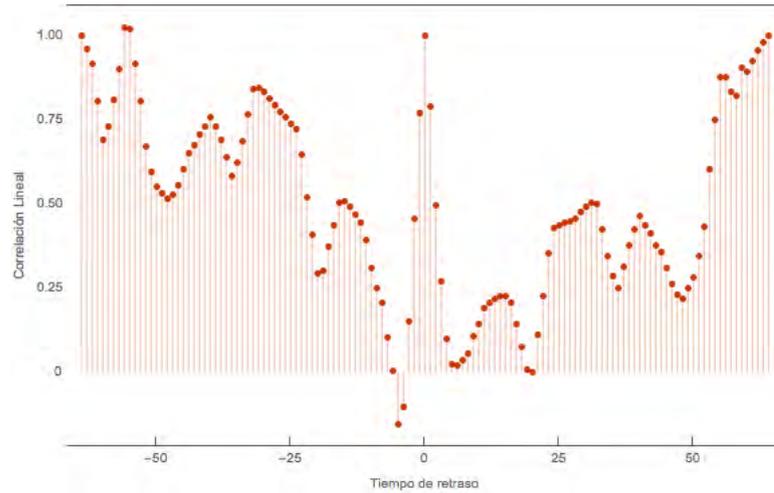


Figura 2.4: Correlación lineal de la serie de tiempo  $\mathfrak{Y}$  como función del tiempo de retraso  $\mathcal{T}$ .

## 2.2. Obteniendo el modelo predictivo

Primero analizaremos los criterios para delimitar el tiempo de retraso que mencionamos en el capítulo anterior. Al graficar la correlación lineal como función del tiempo de retraso  $\mathcal{T}$ , obtenemos el comportamiento mostrado en la Figura (2.4). Donde podemos apreciar que tenemos 4 ceros para tal función. Uno de ellos entre  $\mathcal{T} = -6$  y  $\mathcal{T} = -5$ , otro entre  $\mathcal{T} = -4$  y  $\mathcal{T} = -3$ , el siguiente entre  $\mathcal{T} = 19$  y  $\mathcal{T} = 20$  y el último de ellos entre  $\mathcal{T} = 20$  y  $\mathcal{T} = 21$ . Sin embargo, ninguno de ellos ocurre para algún valor entero de  $\mathcal{T}$  por lo que tendremos que recurrir al siguiente criterio para poder discernir entre alguno estos valores. Vale la pena notar que para casi todos los valores de  $\mathcal{T}$ ,  $C(\mathcal{T})$  tiene un valor positivo implicando que si las dos series están linealmente correlacionadas entonces  $\mathfrak{Y}(t + \mathcal{T})$  dirige a  $\mathfrak{Y}(t)$ .

Cada elemento de  $\mathfrak{Y}$  es diferente, por lo que de manera estricta cada elemento tiene la misma probabilidad a priori. Puesto que el uso de una probabilidad constante no nos proporcionaría información útil haremos las siguientes propuestas. Haciendo un histograma de frecuencias podemos calcular la probabilidad asociada a cada elemento de la serie de tiempo  $\mathfrak{Y}$ , dado que el valor más grande de  $\mathfrak{Y}$  es 8.5401 y el más pequeño es  $-9.1814$  empleamos 4 clases de magnitud 5. Con ello definiremos la siguiente función de probabilidad para  $\mathfrak{Y}$

$$p(t) = \begin{cases} 3/32 & -10 < t < 5 \\ 5/32 & -5 \leq t < 0 \\ 13/32 & 0 \leq t < 5 \\ 11/32 & 5 \leq t < 10 \\ 0 & c.o.c \end{cases}$$

Para el cálculo de las probabilidades conjuntas precedemos de manera análoga, tomamos

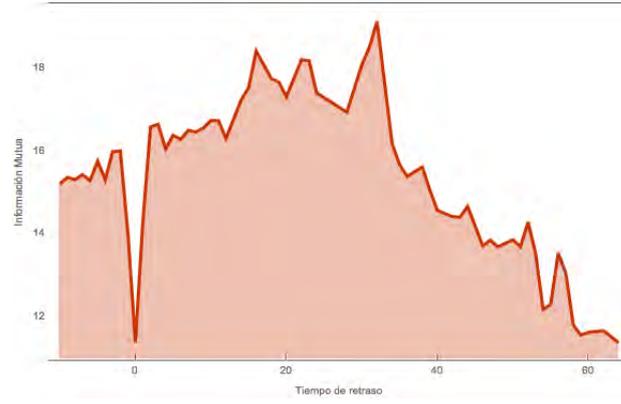


Figura 2.5: Información mutua en función de  $\mathcal{T}$  de la serie de tiempo  $\mathfrak{Y}$ .

el producto  $\mathfrak{Y} \times \mathfrak{Y}$ , definiendo 4 clases de magnitud 3.0206:

$$p_j(t_1, t_2) = \begin{cases} 31/256 & 0.9020 \leq \|(t_1, t_2)\| < 3.1281 \\ 3/8 & 3.1281 \leq \|(t_1, t_2)\| < 6.1487 \\ 13/32 & 6.1487 \leq \|(t_1, t_2)\| < 9.1693 \\ 99/1024 & 9.1693 \leq \|(t_1, t_2)\| < 12.9845 \\ 0 & c.o.c \end{cases}$$

Note que bajo esta propuesta de probabilidad conjunta  $p_j(t_1, t_2) = p_j(t_2, t_1)$ , esto sólo es cierto si  $p(t_1|t_2) = p(t_2|t_1) \iff p(t_1) = p(t_2)$ , por lo que nuestra propuesta se respalda en el hecho de que cada elemento de  $\mathfrak{Y}$  es diferente.

Haciendo uso de estas funciones de probabilidad tenemos que el comportamiento de la información mutua como función del parámetro de retraso  $\mathcal{T}$  es el mostrado en la Figura (2.5). Como podemos apreciar la función consta de varios mínimos locales, el primero de ellos es cero puesto que la información mutua presenta un comportamiento creciente en la vecindad de este punto. Como hemos mencionado la correlación lineal presenta un cero entre  $\mathcal{T} = -3$  y  $\mathcal{T} = -4$  y valores muy cercanos a cero para  $\mathcal{T} = 5$  y  $\mathcal{T} = 6$ ; sin embargo, la función de información mutua tiene un valor relativamente grande para estos tiempos de retraso por lo que emplearemos<sup>1</sup>  $\mathcal{T} = -1$ . Ciertamente le estamos dando un mayor *peso* al criterio de información mutua, puesto que esperamos que las correlaciones presentes en la evolución del PIB no sean lineales. Con el tiempo de retraso establecido, procedemos a calcular la dimensión de embedding. Usando la condición (1.6) con  $\mathcal{R} = 1.2482$  en el método de los falsos vecinos, obtenemos la curva mostrada en la Figura (2.6). Como podemos ver a partir de  $d = 4$  el número de falsos vecinos es igual a cero, para  $d = 3$  tenemos 1 falso vecino que en comparación con el número total de datos (87) es pequeño. Por lo que  $d = 4$  será la dimensión de embedding que emplearemos en el teorema de Takens para la reconstrucción por tiempos de retraso y por consiguiente en el algoritmo genético. La gráfica del espacio de configuraciones utilizando  $d = 3$  se

<sup>1</sup>El parámetro  $\mathcal{T}$  bien puede ser positivo o negativo, puesto que el teorema de Takens no impone restricción, en este tratamiento aunque se analizaron ambos casos hemos de proceder con un valor negativo puesto que esperamos que la serie de tiempo respete un principio *causal*.

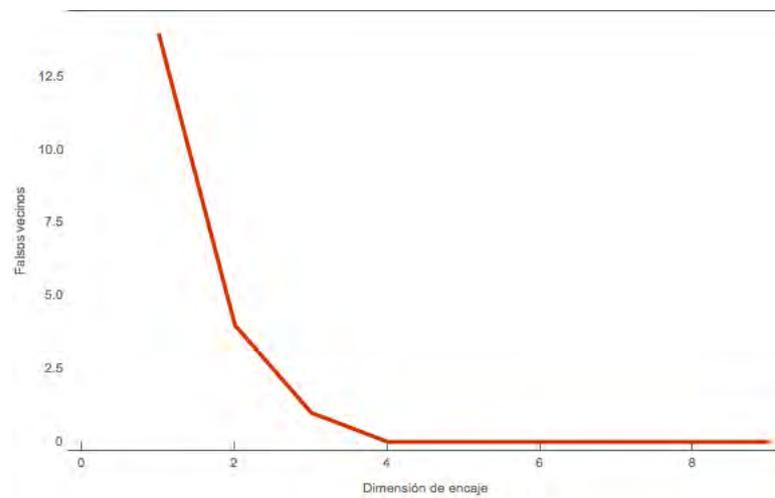


Figura 2.6: Número de falsos vecinos en función de la dimensión de embedding  $d$ .

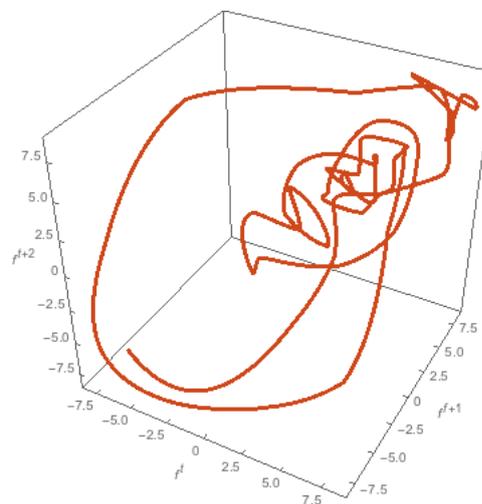


Figura 2.7: Espacio fase reconstruido con una dimensión de embedding  $d = 3$  donde vive la serie de tiempo del PIB.

muestra en la Figura (2.7); aunque el número de valores disponibles en la serie de tiempo no permite identificar algún atractor de una manera clara, gracias al teorema de Takens y a que el número de falsos vecinos tiende a cero a partir de  $d = 4$ , podemos presumir la existencia de tal atractor. Lo cual implicaría que la variación porcentual del PIB mexicano evoluciona hacia un conjunto de valores bien definido; sin embargo, la trayectoria que sigue el sistema dinámico no tiene que satisfacer ninguna propiedad especial excepto permanecer en el atractor; ésta puede ser periódica, caótica o de cualquier otro tipo.

Finalmente, al implementar el algoritmo genético además de las operaciones aritméticas utilizamos las funciones trigonométricas seno y coseno con el fin de aumentar el rango de posibilidades. Primero, se aplicó dicho algoritmo en la serie de datos semi-suavizada, con lo que después de bastante tiempo de cómputo (aproximadamente 5 días de tiempo efectivo utilizando un procesador Intel Core i5 de 2.7 GHz) a la cabeza de nuestra lista de ecuaciones de cadena obtuvimos con una aptitud de 101.903 la siguiente función no lineal para  $f^{t+1}$ :

$$\mathcal{H}_{cw}(f^t, f^{t-1}, f^{t-2}, f^{t-3}) = \sin(0.567764f^{t-3}f^{t-3}) + f^t. \quad (2.1)$$

Cuando aplicamos el algoritmo genético a la serie sin transformación de Wavelet el tiempo de cómputo requerido para obtener una aptitud aproximadamente igual a la del caso anterior aumentó considerablemente (aproximadamente 7 días de tiempo efectivo), sin embargo, obtuvimos una función con una aptitud ligeramente superior. Al dar por finalizado el proceso obtuvimos la función con una aptitud de 99.1814:

$$\mathcal{H}_{sw}(f^t, f^{t-1}, f^{t-2}, f^{t-3}) = \sin[f^{t-3}f^{t-3}] - 0.0318094f^{t-3} + f^t. \quad (2.2)$$

En este capítulo comenzamos describiendo la serie de tiempo generada por las variaciones porcentuales del PIB mexicano. Posteriormente pudo apreciarse como la transformación de wavelet permite controlar el nivel de suavizado en nuestra línea temporal y la información contenida en las fluctuaciones de su evolución. Propusimos un tiempo de retraso empleando las técnicas previamente descritas y mediante el método de falsos vecinos encontramos que la evolución del PIB vive en un espacio 4-dimensional. Lo que a su vez sustenta la existencia de atractores en dicho sistema y el carácter no lineal de las correlaciones que se supuestó inicialmente. Finalmente mostramos las ecuaciones de ajuste brindadas por el algoritmo genético, implementado con y sin suavizado de información, a fin de comparar las predicciones correspondientes.

A continuación procederemos a analizar estos resultados y brindar una conclusión concierne.

# Análisis de Resultados

---

En esta última sección discutiremos las ecuaciones obtenidas a través del algoritmo genético en las dos variantes ya mencionadas, con y sin transformación de Wavelet. Hablaremos sobre que tan buenos son nuestros ajustes tanto cualitativamente como analíticamente mediante la prueba de Kolmogorov-Smirnov. Finalmente, contrastaremos la predicción obtenida mediante nuestros modelos con los datos reales, para brindar una conclusión final sobre la factibilidad de la metodología aquí presentada.

### 3.1. Análisis general

Aunque se trata de una función de amplitud constante, la presencia del término  $f^{t-3}$  en el argumento de la función seno impide que  $\mathcal{H}_{cw}$  tenga un periodo constante. En la Figura (3.1) podemos contrastar el comportamiento de  $\mathcal{H}_{cw}$  con la serie de tiempo semi-suavizada. La principal dificultad con la que se encontró el algoritmo genético fue la caída tan abrupta que sufre la serie entre los años 2008 y 2009 (trimestres 48 y 53), ciertamente un periodo de crisis para la economía mexicana, puesto que durante esos años el PIB sufrió las consecuencias de la llamada *Gran Recesión* sufrida por Estados Unidos, país del cual depende directamente nuestra economía. Nuestro ajuste pareciera ignorar este comportamiento, razón por la cual obtenemos una aptitud de tal magnitud. Al aplicar la prueba de Kolmogorov-Smirnov para saber si nuestro ajuste sigue la misma distribución que los datos originales (entiéndase sin transformación de Wavelet) obtenemos un estadístico de prueba de 0.36166, con lo que existe el 36% de probabilidad de que nuestro ajuste represente a la serie de tiempo en cuestión.

En la Figura (3.2) podemos observar el ajuste generado por  $\mathcal{H}_{sw}$ . Nótese que la caída ya mencionada sigue siendo un problema, puesto que la transformación de Wavelet res-

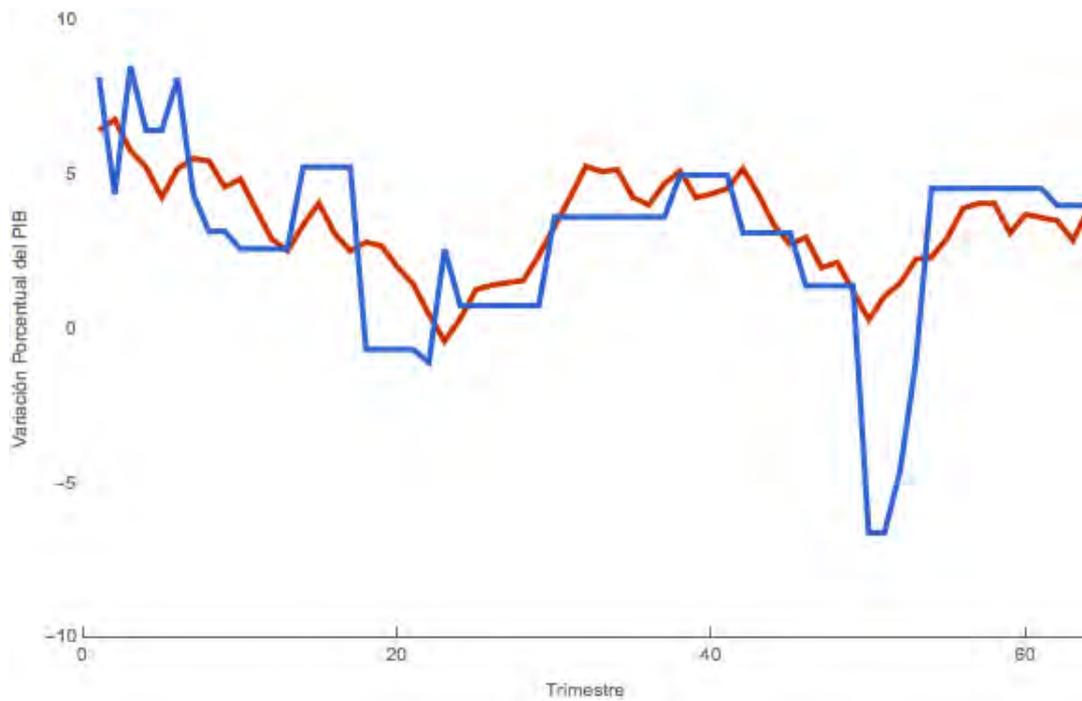


Figura 3.1: En color rojo el ajuste obtenido mediante la implementación del algoritmo genético a la serie de tiempo semi-suavizada en color azul.

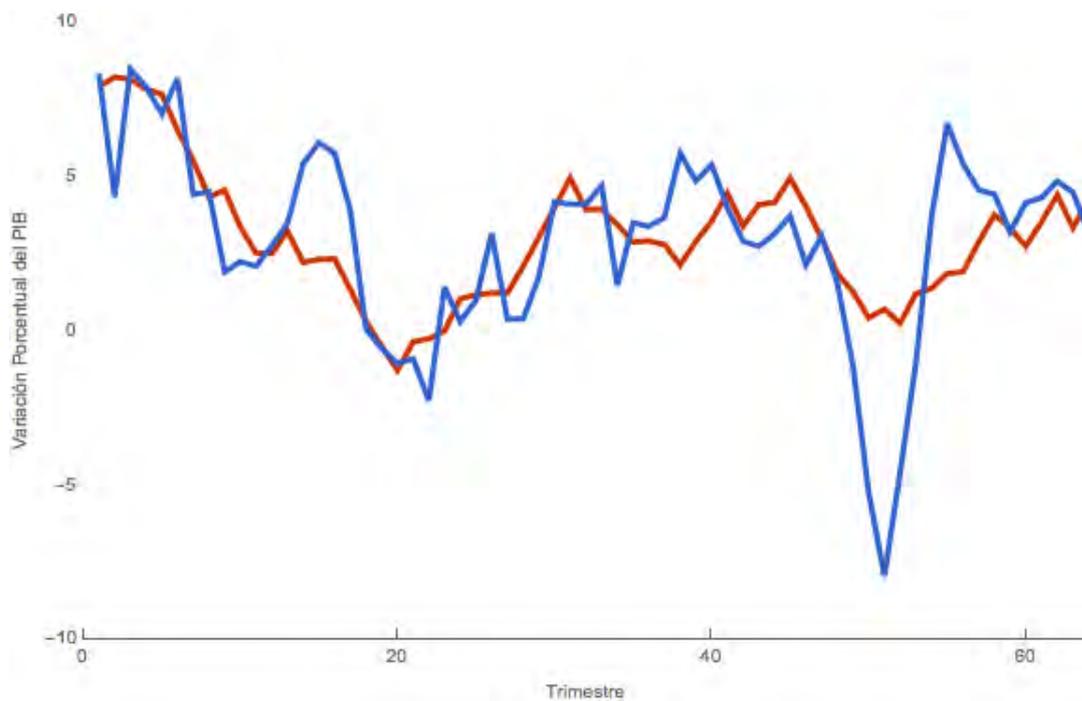


Figura 3.2: En color rojo el ajuste obtenido mediante la implementación del algoritmo genético a la serie de tiempo original (sin transformación de Wavelet) en color azul.

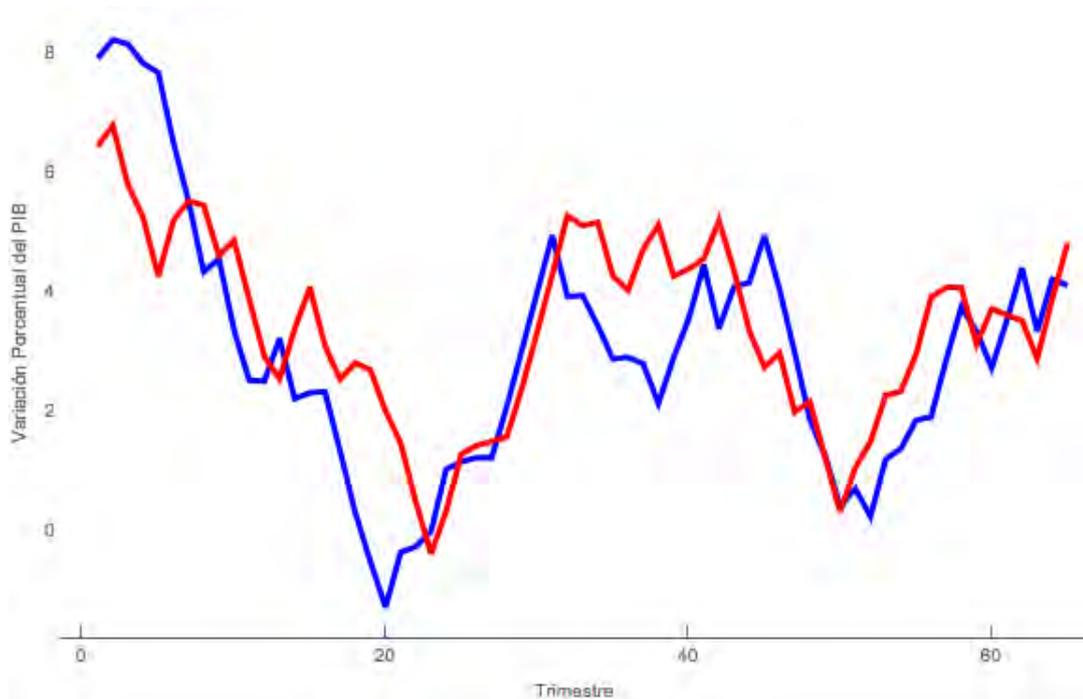


Figura 3.3: Comparación gráfica entre las series de tiempo generadas por  $\mathcal{H}_{cw}$  (en rojo) y  $\mathcal{H}_{sw}$  (en azul).

peto tal fluctuación es normal observar la misma dificultad en ambas implementaciones. El estadístico de prueba en el test de Kolmogorov-Smirnov esta vez resultó de 0.339886, es decir, una probabilidad de casi el 34% de que nuestro ajuste represente a  $\mathcal{Q}$ . Funcionalmente  $\mathcal{H}_{cw}$  y  $\mathcal{H}_{sw}$  son muy semejantes; al comparar gráficamente sus comportamientos (Figura 3.3) podemos notar que una se encuentra desplazada con respecto a la otra, pero dentro del intervalo [35, 45] el comportamiento es esencialmente diferente. Al aplicar la prueba de Kolmogorov-Smirnov entre estos dos ajustes obtenemos que la probabilidad de que ambas series pertenezcan a la misma distribución es de tan solo el 4% (un estadístico de prueba de 0.0385808). Con esto último queremos resaltar que al realizar un suavizado de los datos (aunque sea de manera *mínima* como en el caso de la transformación de Wavelet) nuestros resultados, en este caso el modelo obtenido para ajustar los datos, se ven directamente afectados. A continuación procederemos a analizar las predicciones que emanan de  $\mathcal{H}_{cw}$  y  $\mathcal{H}_{sw}$  respectivamente.

## 3.2. Predicciones

Al no emplear todos los datos disponibles de la variación porcentual del PIB podemos comparar la predicción de cada uno de nuestros modelos con lo ocurrido realmente. En la Figura (3.4) podemos ver que  $\mathcal{H}_{sw}$  genera una predicción que ciertamente es opuesta al comportamiento real. Mientras que nuestro modelo nos indica que a través de una serie de *subidas* y *bajadas* la variación del PIB regresará aproximadamente a su valor inicial después de transcurridos los 11 trimestres, el comportamiento real tuvo un comporta-

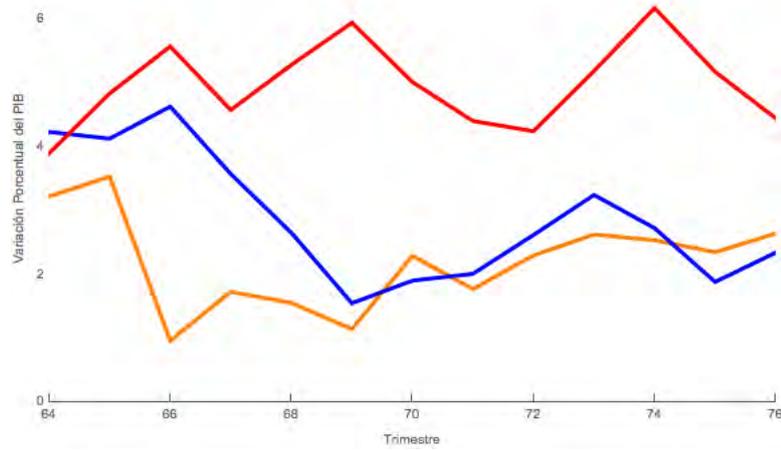


Figura 3.4: Comparación entre las predicciones generadas por  $\mathcal{H}_{cw}$  (en rojo) y  $\mathcal{H}_{sw}$  (en naranja) con respecto a la evolución real de la variación del PIB (en azul).

miento donde la variación final resultó ser menor. Aplicando una vez más la prueba de Kolmogorov-Smirnov obtenemos un estadístico de  $7.91637 \times 10^{-5}$  entre nuestra predicción y la serie real. Por otro lado, en la Figura (3.4) podemos notar que la predicción de  $\mathcal{H}_{sw}$  es cualitativamente mejor que la generada por  $\mathcal{H}_{cw}$  puesto que ésta acierta en la disminución final de la variación del PIB, lo que a su vez se ve reflejado en un mejor estadístico de Kolmogorov-Smirnov, el cual resultó ser de  $1.1664 \times 10^{-2}$ .

# Conclusiones

---

El algoritmo genético representa ciertamente una técnica del tipo *ensayo y error*, y aunque no es claro se presume que de evolucionar un *tiempo suficientemente largo* el sistema de ecuaciones de cadena obtendría la función que representa cierto conjunto de datos. El problema radica en ese “tiempo suficientemente largo”, de implementar el algoritmo sin ningún tipo de información sobre el sistema éste hubiese tenido que *buscar* entre todas las posibles combinaciones, no sólo de operaciones aritméticas, sino de elementos  $f^{t-\mathcal{T}^d}$  en las ecuaciones recursivas. Ésto no solo dificultaría la dicha programación del algoritmo sino que también hubiese aumentado considerablemente el tiempo de cómputo requerido. En el presente trabajo se buscó utilizar el método de reconstrucción por tiempos de retraso para *ayudar* en cierto sentido al algoritmo y que éste encontrará una solución de una manera más rápida. Como ya se mencionó, aún con esta ayuda los tiempos de cómputo son enormes y no se llega a una solución óptima. Pese a ésto, de las predicciones que de nuestros modelos emanan podemos concluir tres cosas:

1. El suavizado de la información afecta considerablemente los resultados, a tal grado que las predicciones de  $\mathcal{H}_{sw}$  y  $\mathcal{H}_{cw}$  parecen ser totalmente diferentes aunque ambas funciones representan ajustes muy semejantes a la línea de tiempo original.
2. El hecho de que  $\mathcal{H}_{sw}$  tuviese una predicción considerablemente aceptable nos hace pensar que la información obtenida mediante el teorema de Takens y la demás teoría utilizada ciertamente ponen por buen camino la búsqueda de un modelo óptimo, de lo cual tenemos la certeza que de aumentar la velocidad (o el tiempo) de cómputo, ya sea por optimización en la programación o un hardware más capaz, eventualmente se obtendrían modelos con predicciones cada vez más certeras.
3. Desde un punto de vista económico, esta metodología es sensible a periodos de re-

cesión o de gran crisis puesto que el algoritmo genético presenta una gran dificultad para ajustar tal comportamiento. Lo que sugiere que dichos periodos deben ser considerados como grandes perturbaciones ajenas al sistema. Si se tuviese el conjunto de ecuaciones no lineales que gobiernan la evolución temporal del PIB, éste no presentaría caídas tan abruptas como las generadas por estos tiempos de crisis; es decir, la evolución *natural* del PIB no tiende a periodos de recesión económica. Con lo que nuestra metodología es viable siempre y cuando no se espere un comportamiento de esta especie.

A su vez, cabe mencionar que si se dispone de un tiempo considerablemente grande la metodología aquí descrita representa una buena técnica para buscar modelos con carácter predictivo; por el contrario, si debe obtenerse una predicción de manera rápida esta técnica aún no es la vía correcta y el análisis cualitativo puede representar una mejor opción. Por lo que en el futuro puede buscarse una mejor implementación computacional, tal vez haciendo uso de la programación paralela, para agilizar la obtención de resultados. También debe buscarse alguna forma de incorporar a esta metodología los periodos de recesión, ya sea alterando el algoritmo genético para que éste pueda lidiar con esta clase de comportamientos o al igual que con el teorema de Takens, buscar alguna forma de *ayudar* al algoritmo para que pueda realizar un mejor ajuste.

En esta sección del trabajo se podrán consultar las rutinas implementadas en **Wolfram Mathematica 10** para el desarrollo del presente trabajo así como los datos empleados.

## Transformación de Wavelet

```
data = Import["Data.txt", "List"];
dwd = DiscreteWaveletTransform[data, HaarWavelet[], 6];
WaveletListPlot[dwd, Ticks -> Full]
Dalia = {};
Dalia = InverseWaveletTransform[WaveletThreshold[dwd]];
```

## Función de Autocorrelación lineal

```
Promedio = Mean[Data];
Correlacion[T_] :=
Sum[If[t + T > n ||
t + T < 1, (Data[[t]] - Promedio)*(Data[[t]] -
Promedio), (Data[[t + T]] - Promedio)*(Data[[t]] -
Promedio)], {t, 1, n}]/Sum[(Data[[t]] - Promedio)^2, {t, 1, n}];
A = {};
For[i = -64, i <= 64, i++, AppendTo[A, {i, Correlacion[i]}]];
```

## Información mutua

```
k1 = (Max[Data] - Min[Data])/12
HD1 = BinCounts[
Data, {Min[Data], Max[Data], (Max[Data] - Min[Data])/12}]
P[t_] := Piecewise[{{HD1[[1]]/(n),
Min[Data] <= t < Min[Data] + k1}, {HD1[[2]]/(n),
Min[Data] + k1 <= t < Min[Data] + 2*k1}, {HD1[[3]]/(n),
Min[Data] + 2*k1 <= t < Min[Data] + 3*k1}, {HD1[[4]]/(n),
Min[Data] + 3*k1 <= t <= Min[Data] + 4*k1}, {HD1[[5]]/(n),
Min[Data] + 4*k1 <= t <= Min[Data] + 5*k1},
{HD1[[6]]/(n), Min[Data] + 5*k1 <= t <= Min[Data] + 6*k1},
{HD1[[7]]/(n), Min[Data] + 6*k1 <= t <= Min[Data] + 7*k1},
```

```

{HD1[[8]]/(n), Min[Data] + 7*k1 <= t <= Min[Data] + 8*k1},
{HD1[[9]]/(n), Min[Data] + 8*k1 <= t <= Min[Data] + 9*k1},
{HD1[[10]]/(n), Min[Data] + 9*k1 <= t <= Min[Data] + 10*k1},
{HD1[[11]]/(n), Min[Data] + 10*k1 <= t <= Min[Data] + 11*k1},
{HD1[[12]]/(n), Min[Data] + 11*k1 <= t <= Min[Data] + 12*k1}}];
cartesianlist = Table[{Data[[i]], Data[[j]]}, {i, 1, n}, {j, 1, n}];
Normas = {};
For[i = 1, i <= n, i++,
  For[j = 1, j <= n, j++,
    AppendTo[Normas, Norm[cartesianlist[[i, j]]]]];
HD = BinCounts[
  Normas, {Min[Normas], Max[Normas], (Max[Normas] - Min[Normas])/12}]
k = (Max[Normas] - Min[Normas])/12;
ProCon[t_] :=
  Piecewise[{{HD[[1]]/(n*n),
    Min[Normas] <= t < Min[Normas] + k}, {HD[[2]]/(n*n),
    Min[Normas] + k <= t < Min[Normas] + 2*k}, {HD[[3]]/(n*n),
    Min[Normas] + 2*k <= t < Min[Normas] + 3*k}, {HD[[4]]/(n*n),
    Min[Normas] + 3*k <= t <= Min[Normas] + 4*k},
  {HD[[5]]/(n*n), Min[Normas] + 4*k <= t <= Min[Normas] + 5*k},
  {HD[[6]]/(n*n),
    Min[Normas] + 5*k <= t <= Min[Normas] + 6*k}, {HD[[7]]/(n*n),
    Min[Normas] + 6*k <= t <= Min[Normas] + 7*k}, {HD[[8]]/(n*n),
    Min[Normas] + 7*k <= t <= Min[Normas] + 8*k}, {HD[[9]]/(n*n),
    Min[Normas] + 8*k <= t <= Min[Normas] + 9*k}, {HD[[10]]/(n*n),
    Min[Normas] + 9*k <= t <= Min[Normas] + 10*k},
  {HD[[11]]/(n*n),
    Min[Normas] + 10*k <= t <= Min[Normas] + 11*k}, {HD[[12]]/(n*n),
    Min[Normas] + 11*k <= t <= Min[Normas] + 12*k}}];
MutualInformation[T_] :=
  Sum[If[t + T > n || t + T < 1,
    ProCon[Norm[{Data[[t]], Data[[t]]}]]*
    Log[ProCon[Norm[{Data[[t]], Data[[t]]}]]]/(
      P[Data[[t]]]*P[Data[[t]])],
    ProCon[Norm[{Data[[t]], Data[[t + T]]}]]*
    Log[ProCon[Norm[{Data[[t]], Data[[t + T]]}]]]/(
      P[Data[[t]]]*P[Data[[t + T]])]], {t, 1, n}];
MutualData = {};
For[i = -10, i <= n, i++,
  AppendTo[MutualData, {i, MutualInformation[i]}]];

```

## Método de los Falsos Vecinos

```

Data = Import["/Users/pierreandroid/Dropbox/Mathematica/data2.txt",
  "List"];
Vari = Variance[Data]
n = 30;
ListLinePlot[Data]
m = 10;
R = 1.2482;
Creamos el vector d-dimensional

Datai;
T = -5;

```

```

Datai = Table[
  Table[If[i + j*T > n || i + j*T < 1, Data[[i]], Data[[i + j*T]]], {i, 1,
    n}, {j, 0, d}], {d, 1, m}];

Calculamos la distancia entre los puntos

Vecinos = Table[
  Table[EuclideanDistance[Datai[[d, i]], Datai[[d, i + j]]], {i, 1,
    n - 3}, {j, 1, 3}], {d, 1, m}];

Decidimos si los vecinos son un falso vecino

FalsosVecinos = Table[Table[If[Abs[Vecinos[[d + 1, i, j]]
  - Vecinos[[d, i, j]]]/Max[0.00001,
    Vecinos[[d, i, j]]] > 1.24*R, 1, 0],
  {i, 1, n - 3}, {j, 1, 3}], {d, 1, m - 1}];
(* 1:Falso vecino, 0: Vecino verdadadro*)

Contador = {};
For[i = 1, i <= m - 1, i++,
  AppendTo[Contador, Total[Total[FalsosVecinos[[i]]]]]];

```

## Reconstrucción del atractor

```

Data = Import["/Users/pierreandroid/Dropbox/Mathematica/data.txt",
  "List"];
tmax = Length[Data] - 3;

x[t_] := Data[[t]];
y[t_] := Data[[t + 1]];
z[t_] := Data[[t + 2]];

intF = Interpolation[Table[{{t}, {x[t], y[t], z[t]}}, {t, 0, tmax}]];
options = {PlotStyle -> {Orange,
  Specularity[White, 10], (Tube @@ {##}) &}, Background -> Black,
  Boxed -> False, Axes -> False, PlotRange -> All, BoxRatios -> 1};
Grafica =
  ParametricPlot3D[intF[t], {t, 0, tmax}, PlotTheme -> "Web",
  AxesLabel -> {"\!\(\(*SuperscriptBox[\(f\), -\(\(t\)\)\]\)",
  "\!\(\(*SuperscriptBox[\(f\), -\(\(f_{-+1}\)\)\]\)",
  "\!\(\(*SuperscriptBox[\(f\), -\(\(t_{-+2}\)\)\]\)"}];

```

## Algoritmo genético

```

(*Inicializacion*)
Data2 = Import["/Users/pierreandroid/Dropbox/Mathematica/data2.txt",
  "List"];
Test = Import["/Users/pierreandroid/Dropbox/Mathematica/Test.txt",
  "List"];
d = 3 ;(*La dimension de encaje*)
T = -1; (*Tiempo de retraso*)

Numero = 64;

```

```

If [EvenQ[Numero/2], Mitad = Numero/2, Mitad = Numero/2 + 1];

Media = Mean[Data2];
(*Dalia=Sum[(Data2[[i]]-Media)^2,{i,1,Numero}];*)
Dalia = {};
For[i = 1, i <= Numero + 1, i++, AppendTo[Dalia, Data2[[i]]]];

(*Funciones aritmeticas*)
S[x_, y_] = x + y;
R[x_, y_] = x - y;
M[x_, y_] = x*y;
Dv[x_, y_] = (Sign[y]*x)/Max[Abs[y], 0.1];
Potencia[x_, y_] = Sin[y*x];
Lo[x_, y_] = Cos[x + y];
Varianza = 1000;

Al = Table[RandomReal[{1, 1000}], {i, 1, Numero}];
\[Lambda] =
  Table[x[n - RandomInteger[{0, d}]], {i, 1, Numero}, {j, 1, d + 1}];
Aleatorio =
  Table[If[i == 1, Al[[j]], \[Lambda][[j]]], {j, 1, Numero}, {i, 1,
    2}];
Operaciones =
  Table[RandomChoice[{m, l, p, q, r, t}], {i, 1, Numero}];
(*Generamos las ecuaciones de cadena*)

Element1 =
  Table[RandomChoice[{S[Aleatorio[[j, 2, 1]], Aleatorio[[j, 2, 2]]],
    R[Aleatorio[[j, 2, 1]], Aleatorio[[j, 2, 2]]],
    M[Aleatorio[[j, 2, 1]], Aleatorio[[j, 2, 2]]],
    Dv[Aleatorio[[j, 2, 1]], Aleatorio[[j, 2, 2]]],
    Potencia[Aleatorio[[j, 2, 1]], Aleatorio[[j, 2, 2]]],
    Lo[Aleatorio[[j, 2, 1]], Aleatorio[[j, 2, 2]]]}], {j, 1,
    Numero}, {i, 1, 1}];

For[i = 1, i <= Numero, i++,
  If[Element1[[i, 1]] == 0, Element1[[i, 1]] = x[n]];

Element2 =
  Table[RandomChoice[{S[Aleatorio[[j, 1]]*Aleatorio[[j, 2, 4]],
    Aleatorio[[j, 2, 3]]],
    R[Aleatorio[[j, 1]]*Aleatorio[[j, 2, 4]], Aleatorio[[j, 2, 3]]],
    M[Aleatorio[[j, 1]]*Aleatorio[[j, 2, 4]], Aleatorio[[j, 2, 3]]],
    Dv[Aleatorio[[j, 1]]*Aleatorio[[j, 2, 4]], Aleatorio[[j, 2, 3]]],
    Potencia[Aleatorio[[j, 1]]*Aleatorio[[j, 2, 4]],
    Aleatorio[[j, 2, 3]]],
    Lo[Aleatorio[[j, 1]]*Aleatorio[[j, 2, 4]],
    Aleatorio[[j, 2, 3]]]}], {j, 1, Numero}, {i, 1, 1}];

Auxiliar =
  Table[Switch[Operaciones[[i]], m, S[Element1[[i]], Element2[[i]]],
    l, R[Element1[[i]], Element2[[i]]], p,
    M[Element1[[i]], Element2[[i]]], q,
    Dv[Element1[[i]], Element2[[i]]], r,
    Potencia[Element1[[i]], Element2[[i]]], t,

```

```

    Lo[Element1[[i]], Element2[[i]]], {i, 1, Numero}];
EcuacionDeCadena = {};
For[i = 1, i <= Numero, i++,
    AppendTo[EcuacionDeCadena, Auxiliar[[i, 1]]];
For[i = 1, i <= Numero, i++,
    If[EcuacionDeCadena[[i]] == 0,
        EcuacionDeCadena[[i]] =
            EcuacionDeCadena[[RandomInteger[{1, Numero}]]]];

Ciclo iterativo
(*Procedemos a calcular la aptitud y a ordenar con respecto a esta*)

Do[
    For[i = 1, i <= Mitad - 1, i = i + 2,
        If[EcuacionDeCadena[[i]] == EcuacionDeCadena[[i + 1]],
            EcuacionDeCadena[[i]] =
                EcuacionDeCadena[[RandomInteger[{Mitad, Numero}]]]];

    Clear[x];
    Clear[Estimaciones];
    Estimaciones = {};

    For[i = 1, i <= Numero, i++,
        AppendTo[Estimaciones,
            If[StringContainsQ[ToString[EcuacionDeCadena[[i]]], "x[-3_+_n]"],
                RecurrenceTable[{x[n + 1] ==
                    Min[1.7*10^(3), EcuacionDeCadena[[i]], x[0] == Data2[[1]],
                    x[1] == Data2[[2]], x[2] == Data2[[3]], x[3] == Data2[[4]]},
                    x, {n, 3, Numero + 3}], 1] If[
                    StringContainsQ[ToString[EcuacionDeCadena[[i]]],
                        "x[-2_+_n]"] && !
                        StringContainsQ[ToString[EcuacionDeCadena[[i]]],
                            "x[-3_+_n]"],
                        RecurrenceTable[{x[n + 1] ==
                            Min[1.7*10^(3), EcuacionDeCadena[[i]], x[0] == Data2[[1]],
                            x[1] == Data2[[2]], x[2] == Data2[[3]]},
                            x, {n, 2, Numero + 2}], 1] If[
                                StringContainsQ[ToString[EcuacionDeCadena[[i]]],
                                    "x[-1_+_n]"] && !
                                    StringContainsQ[ToString[EcuacionDeCadena[[i]]],
                                        "x[-3_+_n]"] && !
                                        StringContainsQ[ToString[EcuacionDeCadena[[i]]],
                                            "x[-2_+_n]"],
                                    RecurrenceTable[{x[n + 1] ==
                                        Min[1.7*10^(3), EcuacionDeCadena[[i]], x[0] == Data2[[1]],
                                        x[1] == Data2[[2]]},
                                        x, {n, 1, Numero + 1}], 1] If[
                                            StringContainsQ[ToString[EcuacionDeCadena[[i]]], "x[n]"] && !
                                            StringContainsQ[ToString[EcuacionDeCadena[[i]]],
                                                "x[-3_+_n]"] && !
                                                StringContainsQ[ToString[EcuacionDeCadena[[i]]],
                                                    "x[-2_+_n]"] && !
                                                    StringContainsQ[ToString[EcuacionDeCadena[[i]]],
                                                        "x[-1_+_n]"],
                                                RecurrenceTable[{x[n + 1] ==
                                                    Min[1.7*10^(3), EcuacionDeCadena[[i]], x[1] == Data2[[1]]},

```

```

    x, {n, 1, Numero}], 1]
  ]]; ==

Clear[F];
F = {};
For[j = 1, j <= Numero, j++,
  AppendTo[F,
    Sum[Abs[Estimaciones[[j, i]] - Data2[[i]]], {i, 1, Numero}]]];
AlConAptitud =
  Table[If[i == 1, Al[[j]], F[[j]]], {j, 1, Numero}, {i, 1, 2}];
\[Lambda]ConAptitud =
  Table[If[i == 1, \[Lambda][[j]], F[[j]]], {j, 1, Numero}, {i, 1,
    2}];
OperacionesConAptitud =
  Table[If[i == 1, Operaciones[[j]], F[[j]]], {j, 1, Numero}, {i, 1,
    2}];
Element1ConAptitud =
  Table[If[i == 1, Element1[[j]], F[[j]]], {j, 1, Numero}, {i, 1, 2}];
Element2ConAptitud =
  Table[If[i == 1, Element2[[j]], F[[j]]], {j, 1, Numero}, {i, 1, 2}];
EcuacionDeCadenaConAptitud =
  Table[If[i == 1, EcuacionDeCadena[[j]], F[[j]]], {j, 1,
    Numero}, {i, 1, 2}];
AlConAptitud = Sort[AlConAptitud, #1[[2]] < #2[[2]] &];
\[Lambda]ConAptitud = Sort[\[Lambda]ConAptitud, #1[[2]] < #2[[2]] &];
OperacionesConAptitud =
  Sort[OperacionesConAptitud, #1[[2]] < #2[[2]] &];
Element1ConAptitud = Sort[Element1ConAptitud, #1[[2]] < #2[[2]] &];
Element2ConAptitud = Sort[Element2ConAptitud, #1[[2]] < #2[[2]] &];
EcuacionDeCadenaConAptitud =
  Sort[EcuacionDeCadenaConAptitud, #1[[2]] < #2[[2]] &];
Clear[x];

(*Proceso de Reproduccion*)
Al = AlConAptitud[[All, 1]];
For[i = 1, i <= Mitad - 1, i = i + 2,
  For[j = 0, j <= 1, j++,
    Al[[i + Mitad + j]] =
      RandomChoice[{AlConAptitud[[i, 1]], AlConAptitud[[i + 1, 1]]}
  ]];

\[Lambda] = \[Lambda]ConAptitud[[All, 1]];
For[i = 1, i <= Mitad - 1, i = i + 2,
  For[j = 0, j <= 1, j++,
    \[Lambda][[i + Mitad + j]] =
      RandomChoice[{\[Lambda]ConAptitud[[i, 1]], \[Lambda]ConAptitud[[
        i, 1]]}
  ]];

Operaciones = OperacionesConAptitud[[All, 1]];
For[i = 1, i <= Mitad - 1, i = i + 2,
  For[j = 0, j <= 1, j++,
    Operaciones[[i + Mitad + j]] =

```

```

RandomChoice[{ OperacionesConAptitud [[i, 1]],
OperacionesConAptitud [[i, 1]]}
]];

(*Proceso de Mutacion*)
For[i = 1, i <= 10, i++, {
Al[[RandomInteger[{Mitad, Numero}]]] = RandomReal[];
\[Lambda][[
RandomInteger[{Mitad, Numero}]]] = {x[
n - RandomInteger[{0, d}]], x[n - RandomInteger[{0, d}]],
x[n - RandomInteger[{0, d}]], x[n - RandomInteger[{0, d}]]}];
Operaciones [[RandomInteger[{Mitad, Numero}]]] =
RandomChoice[{m, 1, p, q, r, t}];
}];

(*Construimos ecuaciones de cadena*)

Aleatorio =
Table[If[i == 1, Al[[j]], \[Lambda][[j]]], {j, 1, Numero}, {i, 1,
2}];
Element1 =
Table[If[j <= Mitad, Element1[[j]] = Element1ConAptitud[[j, 1, 1]],
RandomChoice[{S[Aleatorio[[j, 2, 1]], Aleatorio[[j, 2, 2]]],
R[Aleatorio[[j, 2, 1]], Aleatorio[[j, 2, 2]]],
M[Aleatorio[[j, 2, 1]], Aleatorio[[j, 2, 2]]],
Dv[Aleatorio[[j, 2, 1]], Aleatorio[[j, 2, 2]]],
Potencia[Aleatorio[[j, 2, 1]], Aleatorio[[j, 2, 2]]],
Lo[Aleatorio[[j, 2, 1]], Aleatorio[[j, 2, 2]]}]], {j, 1,
Numero}, {i, 1, 1}];
For[i = 1, i <= Numero, i++,
If[Element1[[i, 1]] == 0, Element1[[i, 1]] = x[n]];

Element2 =
Table[If[j <= Mitad, Element2[[j]] = Element2ConAptitud[[j, 1, 1]],
RandomChoice[{S[Aleatorio[[j, 1]]*Aleatorio[[j, 2, 4]],
Aleatorio[[j, 2, 3]]],
R[Aleatorio[[j, 1]]*Aleatorio[[j, 2, 4]],
Aleatorio[[j, 2, 3]]],
M[Aleatorio[[j, 1]]*Aleatorio[[j, 2, 4]],
Aleatorio[[j, 2, 3]]],
Dv[Aleatorio[[j, 1]]*Aleatorio[[j, 2, 4]],
Aleatorio[[j, 2, 3]]],
Potencia[Aleatorio[[j, 1]]*Aleatorio[[j, 2, 4]],
Aleatorio[[j, 2, 3]]],
Lo[Aleatorio[[j, 1]]*Aleatorio[[j, 2, 4]],
Aleatorio[[j, 2, 3]]}]], {j, 1, Numero}, {i, 1, 1}];

Clear[Auxiliar];
Auxiliar =
Table[Switch[Operaciones[[i]], m, S[Element1[[i]], Element2[[i]],
1, R[Element1[[i]], Element2[[i]], p,
M[Element1[[i]], Element2[[i]], q,
Dv[Element1[[i]], Element2[[i]], r,
Potencia[Element1[[i]], Element2[[i]], t,

```

```

    Lo[Element1[[i]], Element2[[i]]], {i, 1, Numero}];
Clear[EcuacionDeCadena];
EcuacionDeCadena = {};
For[i = 1, i <= Numero, i++,
  AppendTo[EcuacionDeCadena, Auxiliar[[i, 1]]];
For[i = 1, i <= Numero, i++,
  If[Auxiliar[[i, 1]] == 0,
    Auxiliar[[i, 1]] =
      EcuacionDeCadena[[RandomInteger[{1, Numero}]]]];
, {\[Theta], 1, 1}];
i = 1
ListLinePlot[{Data2,
  If[StringContainsQ[ToString[EcuacionDeCadenaConAptitud[[i, 1]]],
    "x[-3_+_n]"],
    RecurrenceTable[{x[n + 1] ==
      Min[1.7*10^(3), EcuacionDeCadenaConAptitud[[i, 1]]],
      x[0] == Data2[[1]], x[1] == Data2[[2]], x[2] == Data2[[3]],
      x[3] == Data2[[4]]}, x, {n, 3, Numero + 3}], 1] If[
StringContainsQ[ToString[EcuacionDeCadenaConAptitud[[i, 1]]],
  "x[-2_+_n]"] && !
  StringContainsQ[ToString[EcuacionDeCadenaConAptitud[[i, 1]]],
    "x[-3_+_n]"],
    RecurrenceTable[{x[n + 1] ==
      Min[1.7*10^(3), EcuacionDeCadenaConAptitud[[i, 1]]],
      x[0] == Data2[[1]], x[1] == Data2[[2]], x[2] == Data2[[3]]},
x, {n, 2, Numero + 2}], 1] If[
StringContainsQ[ToString[EcuacionDeCadenaConAptitud[[i, 1]]],
  "x[-1_+_n]"] && !
  StringContainsQ[ToString[EcuacionDeCadenaConAptitud[[i, 1]]],
    "x[-3_+_n]"] && !
  StringContainsQ[ToString[EcuacionDeCadenaConAptitud[[i, 1]]],
    "x[-2_+_n]"],
    RecurrenceTable[{x[n + 1] ==
      Min[1.7*10^(3), EcuacionDeCadenaConAptitud[[i, 1]]],
      x[0] == Data2[[1]], x[1] == Data2[[2]]}, x, {n, 1, Numero + 1}],
1] If[StringContainsQ[
  ToString[EcuacionDeCadenaConAptitud[[i, 1]]], "x[n]"] && !
  StringContainsQ[ToString[EcuacionDeCadenaConAptitud[[i, 1]]],
    "x[-3_+_n]"] && !
  StringContainsQ[ToString[EcuacionDeCadenaConAptitud[[i, 1]]],
    "x[-2_+_n]"] && !
  StringContainsQ[ToString[EcuacionDeCadenaConAptitud[[i, 1]]],
    "x[-1_+_n]"],
    RecurrenceTable[{x[n + 1] ==
      Min[81.7*10^(3), EcuacionDeCadenaConAptitud[[i, 1]]],
      x[1] == Data2[[1]]}, x, {n, 1, Numero}], 1],
If[StringContainsQ[ToString[EcuacionDeCadenaConAptitud[[i + 1, 1]]],
  "x[-3_+_n]"],
  RecurrenceTable[{x[n + 1] ==
    Min[1.7*10^(3), EcuacionDeCadenaConAptitud[[i + 1]]],
    x[0] == Data2[[1]], x[1] == Data2[[2]], x[2] == Data2[[3]],
    x[3] == Data2[[4]]}, x, {n, 3, Numero + 3}], 1] If[
StringContainsQ[ToString[EcuacionDeCadenaConAptitud[[i + 1, 1]]],
  "x[-2_+_n]"] && !
  StringContainsQ[

```

```

ToString[EcuacionDeCadenaConAptitud[[i + 1, 1]], "x[-3_+_n]"],
RecurrenceTable[{x[n + 1] ==
  Min[1.7*10^(3), EcuacionDeCadenaConAptitud[[i + 1]],
  x[0] == Data2[[1]], x[1] == Data2[[2]], x[2] == Data2[[3]]},
x, {n, 2, Numero + 2}], 1] If[
StringContainsQ[ToString[EcuacionDeCadenaConAptitud[[i + 1, 1]],
"x[-1_+_n]"] && !
StringContainsQ[
  ToString[EcuacionDeCadenaConAptitud[[i + 1, 1]],
  "x[-3_+_n]"] && !
StringContainsQ[
  ToString[EcuacionDeCadenaConAptitud[[i + 1, 1]], "x[-2_+_n]"],
RecurrenceTable[{x[n + 1] ==
  Min[1.7*10^(3), EcuacionDeCadenaConAptitud[[i, 1]],
  x[0] == Data2[[1]], x[1] == Data2[[2]]}, x, {n, 1, Numero + 1}],
1] If[StringContainsQ[
  ToString[EcuacionDeCadenaConAptitud[[i + 1, 1]], "x[n]"] && !
StringContainsQ[
  ToString[EcuacionDeCadenaConAptitud[[i + 1, 1]],
  "x[-3_+_n]"] && !
StringContainsQ[
  ToString[EcuacionDeCadenaConAptitud[[i + 1, 1]],
  "x[-2_+_n]"] && !
StringContainsQ[
  ToString[EcuacionDeCadenaConAptitud[[i + 1, 1]], "x[-1_+_n]"],
RecurrenceTable[{x[n + 1] ==
  Min[81.7*10^(3), EcuacionDeCadenaConAptitud[[i + 1, 1]],
  x[1] == Data2[[1]]}, x, {n, 1, Numero}], 1]],
PlotStyle -> {Blue, Red, Orange}, ImageSize -> Large]

```

EcuacionDeCadenaConAptitud // *MatrixForm*

```

c_{5} → {8.98003, 11.6221, 10.783, 6.33868, 2.9593, 3.45146, 6.31938,
8.40819, 2.82102, -1.13012, -2.21482, 1.23715, 2.96264, 0.56944,
4.20712, 5.83488, 4.38552, 4.89334, 6.69721, 7.26715, 4.90287,
4.17875, 4.1711, 3.3003, -4.48955, -8.84811, 1.87857, 8.61425,
6.38696, 5.23165, 6.50059, 5.46625, 3.17954, 2.32014, 2.42804,
2.8771, 3.65112, 3.52917}},
d_{5} → {2.82937,
0.403519, -0.807755, -0.0629496, -0.235992, -0.473858, -1.36525,
0.24235, 2.71346, 0.340368, 0.962184,
0.786277, -1.54041, -0.00188312, -1.72504, 0.00100528, 2.29352,
0.0832683, -1.4695, -0.372438, 0.773416, -0.290118, 1.12314,
1.05671, 2.90781, -2.37982, -3.35472, 0.921279,
0.0980957, -0.669645, -0.376285, 0.909925, 1.8202,
0.122564, -0.812799, -0.372703, 0.0629741, -0.208675}},
c_{4} → {14.5679, 12.1068, 4.53309, 10.414, 1.19565, -0.691316,
2.49756, 7.10077, 6.56115, 9.87429, 6.42167, 5.28308, -9.43115,
7.41954, 8.2156, 8.46184, 3.88886, 3.7513, 5.07723}},
d_{4} → {-1.86825, 3.1426, -0.348008, -1.47701, 2.79388, -2.44091,
1.69225, -1.151, -0.359082, -0.403007, 0.512032, 0.615747,
3.08197, -4.76285, 0.816924, 0.731387, 0.607681, -0.31753,
0.0862344}},
c_{3} → {18.8619, 10.5692, 0.35662, 6.78704, 11.6216,
8.27651, -1.42242, 11.7927, 5.40241, 13.8912}},
d_{3} → {1.74025, -4.15841, 1.33429, -3.25496, -2.34274,
0.805106, -11.9152, -0.174118, 0.097269, -6.71094}},
c_{2} → {20.8109, 5.05133, 14.0701, 7.33291, 13.6427}},
d_{2} → {5.86385, -4.547, 2.36534, -9.34452, -6.0025}},
c_{1} → {18.2874, 15.1342, 24.3624}},
d_{1} → {11.1437, 4.76391, -5.06872}},
c_{0} → {23.6326, 30.1579}},
d_{0} → {2.22962, 4.29566}}

```

Tabla 4.1: Coeficientes de la transformación de wavelet para  $J = 6$ .

| Trimestre | Valor porcentual | Trimestre | Valor porcentual |
|-----------|------------------|-----------|------------------|
| 1         | 3.183103315      | 50        | 4.875294556      |
| 2         | 5.592065561      | 51        | 5.402001633      |
| 3         | 4.860610695      | 52        | 4.01373919       |
| 4         | 5.251323733      | 53        | 2.919963934      |
| 5         | -1.099012393     | 54        | 2.749674872      |
| 6         | -8.08643143      | 55        | 3.159963655      |
| 7         | -6.934448868     | 56        | 3.743595648      |
| 8         | -6.774344533     | 57        | 2.155230148      |
| 9         | 1.640505248      | 58        | 3.080872481      |
| 10        | 6.702746208      | 59        | 1.586459222      |
| 11        | 6.886755665      | 60        | -1.118456874     |
| 12        | 8.350506894      | 61        | -5.230724037     |
| 13        | 4.349167094      | 62        | -7.939343478     |
| 14        | 8.50342152       | 63        | -4.573772908     |
| 15        | 7.932759975      | 64        | -1.043799015     |
| 16        | 7.053552671      | 65        | 3.700492468      |
| 17        | 8.195890703      | 66        | 6.742638578      |
| 18        | 4.43760891       | 67        | 5.439752941      |
| 19        | 4.526633028      | 68        | 4.585624136      |
| 20        | 1.92567026       | 69        | 4.446895814      |
| 21        | 2.259413791      | 70        | 3.225825447      |
| 22        | 2.10548141       | 71        | 4.172846832      |
| 23        | 2.775618494      | 72        | 4.330537298      |
| 24        | 3.503101001      | 73        | 4.862685099      |
| 25        | 5.433857194      | 74        | 4.50863888       |
| 26        | 6.116857219      | 75        | 3.221810144      |
| 27        | 5.774121957      | 76        | 3.53534504       |
| 28        | 3.913469001      | 77        | 0.961197084      |
| 29        | 0.076062551      | 78        | 1.72725602       |
| 30        | -0.558436709     | 79        | 1.553924326      |
| 31        | -1.039789871     | 80        | 1.142150397      |
| 32        | -0.885745335     | 81        | 2.291622052      |
| 33        | -2.246479025     | 82        | 1.7708748        |
| 34        | 1.430778063      | 83        | 2.297956929      |
| 35        | 0.318814163      | 84        | 2.626262248      |
| 36        | 1.005667881      | 85        | 2.537203451      |
| 37        | 3.184137529      | 86        | 2.347943102      |
| 38        | 0.401323304      | 87        | 2.64305376       |
| 39        | 0.403986433      |           |                  |
| 40        | 1.755096646      |           |                  |
| 41        | 4.194674943      |           |                  |
| 42        | 4.126596765      |           |                  |
| 43        | 4.125175088      |           |                  |
| 44        | 4.722800874      |           |                  |
| 45        | 1.479267377      |           |                  |
| 46        | 3.518995239      |           |                  |
| 47        | 3.40123605       |           |                  |
| 48        | 3.696549325      |           |                  |
| 49        | 5.774733361      |           |                  |

Tabla 4.2: Variación porcentual del PIB mexicano desde 1996 hasta el 2015



# Bibliografía

---

- [1] P. M. Fourier, *Théorie analytique de la Chaleur*. A Paris, 1822.
- [2] A. E. J. S. A.E. Eiben, J.E. Smith, *Introduction to Evolutionary Computing*. Springer-Verlag, 2003.
- [3] A. R. B. Kliková, “Reconstruction of phase space of dynamical systems using method of time delay,” 2011.
- [4] H. D. I. Abarbanel, R. Brown, J. J. Sidorowich, and L. S. Tsimring, “The analysis of observed chaotic data in physical systems,” *Rev. Mod. Phys.*, vol. 65, pp. 1331–1392, Oct 1993. <http://link.aps.org/doi/10.1103/RevModPhys.65.1331>.
- [5] F. Takens, *Dynamical Systems and Turbulence, Warwick 1980: Proceedings of a Symposium Held at the University of Warwick 1979/80*, ch. Detecting strange attractors in turbulence, pp. 366–381. Berlin, Heidelberg: Springer Berlin Heidelberg, 1981. <http://dx.doi.org/10.1007/BFb0091924>.
- [6] J. P. Huke, *Embedding Nonlinear Dynamical Systems: A Guide to Taken’s Theorem*. Technical report, 2006. <https://core.ac.uk/download/files/72/286451.pdf>.
- [7] N. F. Rulkov, M. M. Sushchik, L. S. Tsimring, and H. D. I. Abarbanel, “Generalized synchronization of chaos in directionally coupled chaotic systems,” *Phys. Rev. E*, vol. 51, pp. 980–994, Feb 1995. <http://link.aps.org/doi/10.1103/PhysRevE.51.980>.
- [8] F. Emmert-Streib and M. Dehmer, *Information Theory and Statistical Learning*. Springer.
- [9] M. F. Andrew and L. S. Harry, “Independent coordinates for strange attractors from mutual information,” *Phys. Rev. A*, vol. 33, February 1986.
- [10] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. Cambridge, MA, USA: MIT Press, 1992.

- [11] A. Haar, “Zur theorie der orthogonalen funktionensysteme. (erste mitteilung).,” *Mathematische Annalen*, vol. 69, pp. 331–371, 1910. <http://eudml.org/doc/158469>.
- [12] A. Grossmann and J. Morlet, “Decomposition of hardy functions into square integrable wavelets of constant shape,” *SIAM Journal on Mathematical Analysis*, vol. 15, no. 4, pp. 723–736, 1984. <http://dx.doi.org/10.1137/0515056>.
- [13] V. S. Anishchenko, V. Astahov, A. Neiman, T. Vadivasova, and Schimansky-Geier, *Nonlinear Dynamics of Chaotic and Stochastic Systems*. Springer Series in Synergetics, Springer Complexity.
- [14] L. Debnath and F. Shah, *Wavelet Transforms and Their Applications*. Birkhäuser, 2012.