



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

FACULTAD DE INGENIERÍA

**ANÁLISIS DE SEÑALES OPTO-ACÚSTICAS
POR MEDIO DE FRACTALES PARA
LA CARACTERIZACIÓN DE MATERIALES**

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO ELÉCTRICO – ELECTRÓNICO

P R E S E N T A :

ROSALES ROMERO CARLOS ANGEL



**DIRECTORA DE TESIS
DRA. MARGARITA NAVARRETE MONTESINOS**

MÉXICO D.F.

FEBRERO 2006



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



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.

AGRADECIMIENTOS

En primer lugar agradezco a mis padres por el apoyo que me han otorgado todos estos años, sus consejos, así como sus observaciones e historia, ya que para entender nuestro presente y futuro, es necesario saber el pasado, de donde venimos y que hemos logrado.

A mis hermanos, con los cuales siempre he contado, pues gracias a ellos sé que se pueden lograr las metas disfrutando el camino para llegar a ellas; también agradezco mis sobrinos y a su mamá, que son parte esencial en la familia.

Para mis amigos de la facultad que siempre conté con su apoyo tanto a nivel académico como personal.

También agradezco a todos mis amigos ya que influyeron en mí, así como yo influí en ustedes.

A toda mi familia le agradezco ya que de manera directa, ha influenciado en mi vida originando en mí una personalidad, así como disfrutando momentos, y apoyándonos siempre que lo necesitamos.

A mis maestros, que fueron una parte fundamental para mi desarrollo profesional y personal.

Agradezco al proyecto IN-112402 por la beca que se me otorgó durante la realización de este trabajo.

Un agradecimiento muy importante a mi tutora Dra. Margarita Navarrete Montesinos por haberme ayudado a la realización y supervisión de este trabajo, ya que sin su apoyo no habría sido posible.

Agradecimientos	
Resumen	
Introducción y Objetivos	1
Capítulo I. CONCEPTOS BÁSICOS	
I.1. Antecedentes	3
I.2. Auto-similitud	3
I.3. Clasificación de fractales	5
I.4. La dimensión fractal	5
I.5. El coeficiente de Hurst y las series de tiempo	7
Capítulo II. ANÁLISIS DE FRACTALES ALEATORIOS	
II.1. Series de tiempo	8
II.2. Métodos de varianza aplicados al ventaneo escalado	10
II.3. Comparación de los métodos	12
Capítulo III. MÉTODO DFA (Análisis de la de-tendencia a las fluctuaciones)	
III.1. Análisis de un proceso auto-similar	14
III.2. Descripción del método DFA	15
III.3. Los coeficientes de Hurst y de Autocorrelación	16
III.4. Algoritmo del método DFA	17
III.5. Diagrama de flujo del método DFA	20
III.6. Programación del algoritmo del método DFA	26
Capítulo IV. PROCEDIMIENTO EXPERIMENTAL	
IV.1. Procedimiento experimental	28
IV.2. Procesamiento de señales OA	30
IV.3. Grado de conversión de dobles enlaces vs. coeficiente de Hurst	38
CONCLUSIONES Y RECOMENDACIONES	40
Glosario de abreviaturas	42
Apéndice A. MÉTODO DE CONTEO DE CAJA	43
Apéndice B. LEY DE POTENCIAS	45
Apéndice C. CÓDIGO FUENTE DEL PROGRAMA	46
Referencias	58

ÍNDICE DE FIGURAS

Figura 1. Curva de Peano.	Pág. 3
Figura 2. Curva de Koch.	Pág. 4
Figura 3. Conjunto de Cantor.	Pág. 4
Figura 4.a) Espiral logarítmica.	Pág. 4
Figura 4.b) Triángulo de Sierpinsky.	Pág. 4
Figura 5. Dimensión topológica y euclidiana.	Pág. 6
Figura 6. Ruido fGn y fBm.	Pág. 7
Figura 7.a) Dos ventanas de observación con escala de tiempo η_1 y η_2 , y una ampliación de la ventana más pequeña.	Pág. 9
Figura 7.b) Diagrama de Log-Log de las escalas características de fluctuaciones.	Pág. 9
Figura 8.a) Estimación de H, método estándar.	Pág. 12
Figura 8.b) Estimación de H, método LD.	Pág. 12
Figura 8.c) Estimación de H, método BD.	Pág. 12
Figura 9.a) Ejemplo de una serie de tiempo.	Pág. 14
Figura 9.b) Ruido blanco.	Pág. 14
Figura 10. Método DFA.	Pág. 15
Figura 11. Algoritmo dividido por módulos.	Pág. 20
Figura 12. Primer módulo.	Pág. 21
Figura 13. Segundo módulo.	Pág. 21
Figura 14. Tercer módulo, corte.	Pág. 22
Figura 15. Tercer módulo, normalización y cambio de referencia.	Pág. 23
Figura 16.a) Cuarto módulo, cálculo del coeficiente H.	Pág. 24
Figura 16.b) Cuarto módulo, cálculo del coeficiente H.	Pág. 25

Figura 17. Quinto módulo, despliegue de resultados.	Pág. 25
Figura 18. Interfaz del programa.	Pág. 26
Figura 19. Diagrama experimental para la obtención de señales OA de una resina dental durante su foto-polimerización.	Pág. 29
Figura 20.a) Señal opto-acústica, amplitud vs. tiempo.	Pág. 31
Figura 20.b) Señal opto-acústica, amplitud vs. número de puntos	Pág. 31
Figura 21. Señal opto-acústica cortada.	Pág. 32
Figura 22.a) Señal opto-acústica normalizada y referenciada a cero, amplitud vs. tiempo.	Pág. 32
Figura 22.b) Señal opto-acústica normalizada y referenciada a cero, amplitud vs. número de puntos.	Pág. 33
Figura 23. Interfaz gráfica del programa.	Pág. 33
Figura 24.a) Conjunto de señales del intervalo I.	Pág. 34
Figura 24.b) Su comportamiento (gráfica logarítmica).	Pág. 34
Figura 25. Gráfica logarítmica y su derivada donde se delimitan los intervalos para analizar.	Pág. 35
Figura 26. Coeficientes H para los tres intervalos.	Pág. 36
Figura 27. Comportamiento doble logarítmico del “intervalo extendido II”.	Pág. 37
Figura 28. Dependencia del coeficiente H de microgeles sobre el tiempo de reacción t para el sistema BisGMA/TEDGMA.	Pág. 37
Figura 29. Dependencia del coeficiente H sobre el tiempo de reacción, t, para el sistema BisGMA/TEDGMA.	Pág. 38
Figura 30. Variación del porcentaje de conversión de los dobles enlaces y el parámetro de auto-similitud, H calculada por el método DFA durante la foto-polimerización de (bis-GMA/TEGDMA).	Pág. 39
Figura. A.1. Gráfica logarítmica.	Pág. 44

ÍNDICE DE TABLAS

Tabla 1. Intervalo I.	Pág. 29
Tabla 2. Intervalo II.	Pág. 29
Tabla 3. Intervalo III.	Pág. 30
Tabla 4. Intervalo IV.	Pág. 30
Tabla 5. Intervalo V.	Pág. 30
Tabla 6. Intervalo VI.	Pág. 30
Tabla 7. Intervalo VII.	Pág. 30
Tabla 8. División por intervalos.	Pág. 35

El método de fractales conocido con el nombre de Análisis de la Dependencia a las Fluctuaciones (DFA, por sus siglas en inglés), es programado en el lenguaje Matlab y se aplica para analizar series de tiempo que se generan consecutivamente durante el proceso de entrecruzamiento del copolímero BisGMA/TEGDMA (resina dental) durante la reacción de foto-polimerización *in situ* mediante opto-acústica. La cinética de la resina es monitoreada en función del tiempo de reacción adquiriendo señales opto-acústicas (OA), durante un intervalo de tiempo fijo. El método opto-acústico consiste en aplicar un pulso láser sobre la superficie de la muestra que genera inmediatamente debajo de ella, una onda acústica que avanza a través de su espesor sondeándola. Dicha señal optoacústica es detectada con un sensor piezoeléctrico.

Los parámetros de auto-similitud, H , durante la reacción de foto-polimerización son calculados a partir de las señales OA y se comparan con el grado de conversión obtenido por la Transformada de Fourier Infrarroja (FTIR, por sus siglas en inglés) bajo condiciones similares.

Introducción

Trescientos años antes de cristo, Euclides trata de explicar el total del universo fundándose en la geometría clásica, sin embargo, el mundo real es indefinible utilizando este único concepto; ya que como sabemos la naturaleza con sus distribuciones no uniformes es indescriptible en términos de la geometría euclidiana. Fue hasta 1975 cuando Benoît Mandelbrot presentó los primeros conceptos que describían un mundo fractal, y con éstos se logra describir geoméricamente figuras de la naturaleza con sus detalles a cualquier escala. Fractal viene de la raíz latina *fractus* la cual se refiere a fragmentado, roto o discontinuo.

Esta es la geometría que nosotros encontramos en la naturaleza, la de las figuras irregulares, en general los fractales se caracterizan por tener un detalle infinito, longitud infinita y la ausencia de bordes suaves. La geometría fractal es una extensión de la geometría clásica, pero no la reemplaza, sino que la enriquece y la profundiza. Actualmente, con el uso de las computadoras se pueden desarrollar modelos precisos de estructuras físicas con geometría fractal que van desde las conchas de mar hasta galaxias.

Desde 1975, en que Mandelbrot da a conocer el mundo fractal, hasta 1990 aparecen más de 5000 títulos publicados referentes a ello [2].

Este trabajo se enfoca al análisis de fractales aleatorios, ya que las series de tiempo en tener estas mismas características.

Los métodos para analizar fractales aleatorios se fundamentan en procedimientos de conteo de cajas (ver, apéndice A) y son llamados “métodos de varianza aplicados al ventaneo escalado que son: el estándar, el de-tendencia lineal y el de de-tendencia puente, estos métodos son para el cálculo del parámetro de auto-similitud o coeficiente de Hurst.

La presentación de este trabajo es como sigue:

En el primer capítulo se dan los conceptos básicos en que se fundamentan los fractales en general, luego se especifican los conceptos para desarrollar el análisis de fractales aleatorios aplicados a las series de tiempo (Coeficiente Hurst).

En el segundo capítulo se describe el análisis de fractales aleatorios y los métodos de varianza aplicados al ventaneo escalado así como se explica la elección del método DFA.

El tercer capítulo contiene la explicación del método DFA, y se desarrolla el algoritmo para Matlab.

En el cuarto capítulo se describe la metodología para la adquisición de señales opto-acústicas, que son el conjunto de series de tiempo que se generan consecutivamente durante

el proceso de entrecruzamiento del co-polímero Bis-GMA/TEGDMA (resina dental) durante la reacción de foto-polimerización *in situ* .

En conclusiones se hace una revisión de los beneficios y limitaciones del método propuesto para la caracterización por medio de fractales.

De igual forma se incluye un glosario de las abreviaturas usadas en este trabajo.

Objetivos

El objetivo principal de esta tesis es programar el algoritmo DFA para utilizarlo en series de tiempo con características de fractales aleatorios, usando MatLab como lenguaje de programación. El método DFA da una medida de la auto-similitud (el coeficiente H), esto es, encontrar un factor de escalamiento en la señal. En particular se aplica a señales OA de la resina Bis-GMA/TEGDMA en proceso de foto-polimerización, con ello se va a dilucidar el comportamiento fractal durante su proceso de transformación de resina líquida a sólida. Para ello se compara el coeficiente H con el porcentaje de polimerización η para observar si existe relación entre ambos.

I.1. Antecedentes

La geometría fractal se fue desarrollando por partes y por necesidad. Entre 1800 y principios de 1900, Kart Weierstrass, Georg Cantor y Henri Poincaré ven la necesidad crear una nueva clase de geometría que representara los aspectos geométricos del mundo real con un lenguaje euclidiano básico. Fue así como Kart Wierstrass desarrolla el primer fractal en 1861, una curva formada completa y únicamente de esquinas, la cual la hace no-diferenciable, es decir, no se puede definir su razón de cambio en ningún punto [1, 2].

Cuando los matemáticos de la época estaban en la búsqueda para definir el concepto de dimensión, Giuseppe Peano (1858–1932) presentó la llamada “curva que llena todo el espacio”, es continua pero su relación no es “uno a uno” (ver Figura 1). No existe punto sobre el plano el cual la curva de Peano no incluyera. En este caso, el concepto de dimensión no tendría un significado topológico en lo absoluto. Ya que como sabemos, la topología es el estudio de las propiedades geométricas y relaciones espaciales que permanecen inafectadas si las alargamos, doblamos o giramos [1].

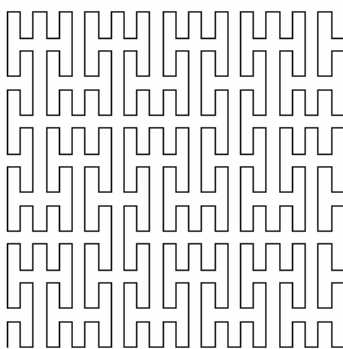


Figura 1. Curva de Peano.

I.2. Auto-Similitud

La auto-similitud significa que un objeto está compuesto de sub-unidades y de sub-sub-unidades semejantes a la estructura total del objeto en múltiples niveles (estadísticamente). Matemáticamente, esta característica se debe cumplir en todas las escalas, pero en la práctica hay límites inferiores y superiores que se aplican en el comportamiento auto-similar. Por ejemplo, el conjunto de Cantor (Figura 3) consiste simplemente de dos pequeñas copias de sí mismo, esta propiedad es conocida como auto-similitud (ver Figura 2); cualquier parte de una línea recta es así misma, idéntica a su totalidad excepto por un factor de escala. Muchas de las formas Euclidianas no comparten esta propiedad, por ejemplo el arco de un círculo no es un círculo, así mismo el lado de un triángulo no es triangular, sin embargo, en las figuras que presenta la naturaleza abunda la auto-similitud, como en árboles, nubes, y montañas están reensambladas por pequeñas partes de si mismos. Estas formas son increíblemente complicadas para ser descritas en términos euclidianos

aún, si éstas son comparadas con las formas llamadas “patológicas” que desarrollan los matemáticos actuales, en donde se despliegan series interminables de motivos dentro de motivos repetidos a todas las escalas [1, 2, 9].

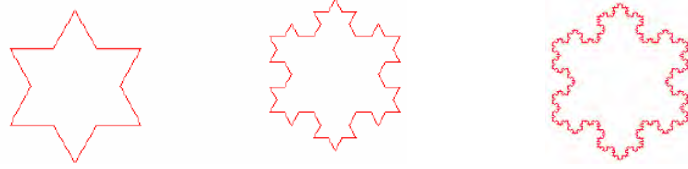


Figura 2. Curva de Koch.

Por ejemplo, los anillos de Júpiter despliegan una estructura fractal semejante al conjunto de Cantor (ver Figura 3) con aberturas que corresponden a orbitas inestables.

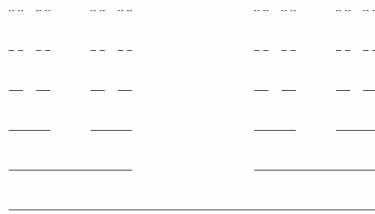


Figura 3. Conjunto de Cantor.

Es importante hacer notar la diferencia entre un objeto fractal que presenta la propiedad de auto-similitud en todas sus partes y aquellos que lo presentan en un lugar en específico. Por ejemplo, en una espiral logarítmica observamos auto-similitud solo si consideramos un acercamiento en el punto de convergencia (Figura 4.a), mientras el triángulo de Sierpinsky (Figura 4.b) presenta la propiedad de auto-similitud en todas partes, no importa en qué lugar hacemos el acercamiento, siempre obtendremos una réplica del triángulo, esta propiedad se conoce como auto-similitud exacta [1, 6, 7].



Figura 4.a) Espiral logarítmica.

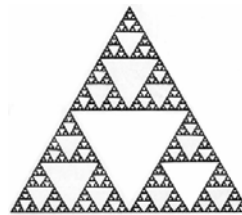


Figura 4.b) Triángulo de Sierpinsky.

En la naturaleza se puede observar auto-similitud exacta en las grietas de una pared; por lo general, la naturaleza no posee la propiedad de auto-similitud en todas partes, pero si consideramos ciertas partes de los objetos podemos hallar la auto-similitud requerida como para aplicar los métodos de la geometría fractal [7].

Los objetos que poseen auto-similitud exacta son conocidos como fractales regulares. Benoit Mandelbrot, considerado el padre de la geometría fractal, postuló la siguiente definición de fractal: “Un fractal es por definición un conjunto para el cual su dimensión de Hausdorff excede estrictamente su dimensión topológica”, pero más tarde se retractó y la reemplazó por: “Un fractal es una forma la cual está compuesta de partes similares al objeto total”, la definición de fractal que se usa en el presente trabajo es un trazo que conserva la propiedad de auto-similitud en varias escalas, cuya dimensión fractal no es un número entero [7].

I.3. Clasificación de fractales

A los fractales se les puede dividir en dos grupos, los fractales regulares y los aleatorios. Los primeros se componen de copias escaladas y posiblemente rotadas de ellos mismos, mientras que los fractales aleatorios contienen un componente aleatorio o estadístico, el cual se le puede agregar a un proceso de construcción de un fractal regular. Por lo que los fractales aleatorios son de gran utilidad para la descripción de propiedades de muchos procesos y objetos naturales [4, 7].

Para la estimación de la dimensión de fractales regulares tenemos D_S (dimensión de similitud), con la cual tenemos que identificar primero las partes auto-similares de la estructura que estamos analizando.

La caracterización de los fractales aleatorios se puede hacer por medio de su dimensión fractal o bien por el coeficiente de auto-similitud (coeficiente H), esto partiendo de que el subconjunto contiene las mismas propiedades estadísticas que la total, con este antecedente, nos enfocaremos a los fractales aleatorios [7].

I.4. La dimensión fractal

Se deben considerar las definiciones de la dimensión topológica y la dimensión euclidiana, para conceptualizar la dimensión fractal de un objeto. La dimensión euclidiana D_E , es simplemente el número de coordenadas requeridas para especificar un objeto en el universo que lo contiene (Ver Figura 5). Para definir la dimensión topológica, se debe dar el concepto de topología y se tiene lo que A. W. Tucker y H. S. Bailey dijeron en 1950 [5]:

“La topología es la rama de las matemáticas que trata de las propiedades de posición que son invariantes por cambios en tamaño o forma. Sus objetos están constituidos por superficies, redes y muchas otras figuras. Tal vez el modo más fácil de definir propiedades topológicas consiste en decir que son propiedades geométricas que permanecen inmutables a pesar de estiramientos o encorvamientos. La topología está llena de paradojas aparentes e imposibilidades aparentes y es, probablemente, más divertida que cualquier otra rama de las matemáticas”.

Por lo que un objeto tiene dimensión topológica, D_T , cuando cualquier recubrimiento de ese objeto tiene como mínimo una dimensión topológica igual a D_T+1 (ver Figura 5) [2, 8, 9]. La dimensión topológica D_T es una propiedad invariante bajo una transformación de tamaño o forma, resulta más intuitiva y pragmática para comprender, ésta establece la dimensión de un punto igual a cero, la de una curva igual a uno, la de una superficie igual a dos, etc.

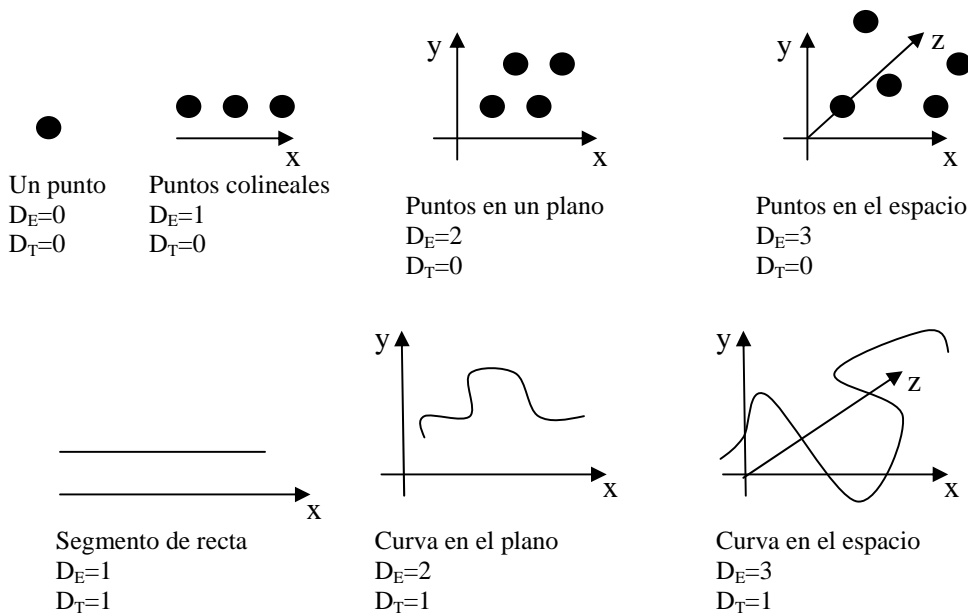


Figura 5. Dimensión topológica y euclidiana.

En 1911 Luitzen Brouwer (1881–1966) demostró que la dimensión es una invariante topológica, y no puede ser alterada por una deformación continua. Así, se definió la denominada “dimensión topológica”. En cambio, Félix Hausdorff (1869-1972) concibe otro concepto de dimensión (la dimensión fractal), enfocándose en la manera de como las formas llenan el espacio alrededor de ellas. Hausdorff dedujo una medida en la cual se extienden ideas intuitivas de dimensionalidad. Su aproximación proporciona una dimensión fraccionaria para las formas complicadas y otras entidades que están muy lejos de ser objetos euclidianos normales, permitiendo la posibilidad intrigante de la existencia de objetos que poseen una y media dimensión [1, 2, 7].

¿Cómo puede una figura ubicarse en el medio de las dimensiones? Solo siendo fractal.

La dimensión fractal permite caracterizar a los fractales, llegando a la conclusión de que son objetos cuya dimensión no es entera, de aquí en adelante se supondrá que la dimensión de tales estructuras fractales es un número no entero D que será llamado dimensión fractal. La palabra dimensión para el caso de un objeto fractal se debe interpretar como el espacio que ocupa en el universo en el que se encuentra inmerso [1, 2, 3].

A los fractales se les puede caracterizar según su coeficiente de auto-similitud (fractales aleatorios) o su dimensión (dimensión fractal); podemos encontrar muchas definiciones de la dimensión fractal, algunas definiciones que podemos hallar son: la dimensión de similitud D_S ; la divisora, D_D ; la de Hausdorff, D_H ; la de correlación D_C ; la puntual D_P , la de conteo de caja D_B ; y la de Lyapunov D_L [6].

I.5. El coeficiente de Hurst y las series de tiempo

El coeficiente de Hurst (H) es una medida de la auto-similitud entre elementos en una serie de tiempo y puede ser usado para comparar diferentes series de tiempo [15]. El rango de H varía de $[0,1]$, la definición equivalente de H se basa en dos diversos procesos estocásticos, el primero es el Ruido Gaussiano Fraccionario (fGn), que es un proceso estacionario¹. El otro proceso estocástico es el Movimiento Browniano Fraccionario (fBm), que es un proceso no estacionario, auto-similar. Estos dos procesos se relacionan entre sí, ambos son ruidos que pueden encontrarse inmersos en las señales, el Movimiento Browniano Fraccionario (fBm) es un proceso estocástico Gaussiano de media cero y no estacionario, esto es, que sus principales propiedades estadísticas dependen del tiempo. Mientras que el Ruido Gaussiano Fraccionario se forma por las diferencias del fBm, en sentido estricto el fBm no tiene derivada, pero al tomar un intervalo muy pequeño de la señal se puede ver que se suaviza, por lo que obtenemos de las diferencias del fBm el fGn, entonces las sumas acumulativas o integración del fGn da como resultado un fBm, como se ve en la Figura 6, por lo que es necesario especificar no solamente el valor de H para una señal, sino también si la señal es fGn o fBm [15, 16].

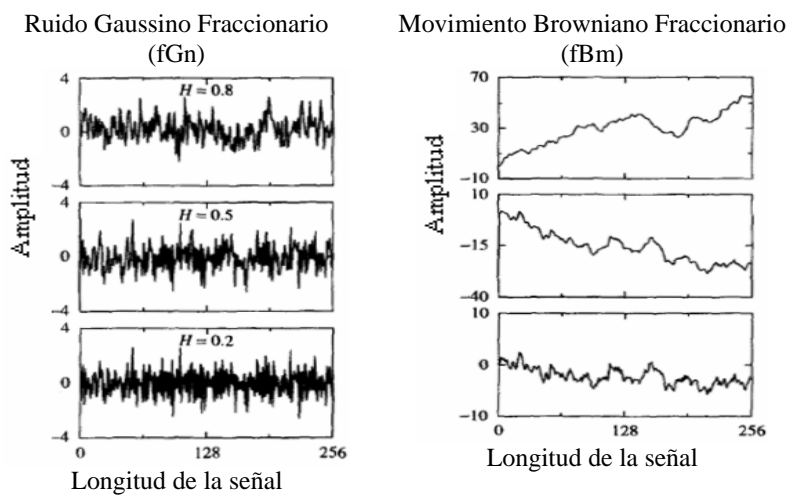


Figura 6. Ruido fGn y fBm.

Un proceso gaussiano fraccionario (fGn) es estacionario, tiene momentos finitos y se definen mejor con señales más largas [15, 21].

¹ Un proceso estocástico del parámetro discreto $\{X_t\}$ se dice que va a ser estacionario de segundo orden si $E\{X_t\} = \mu$ y $\text{cov}\{X_t, X_{t+\tau}\} = s_\tau$ para $\tau = 0, \pm 1, \pm 2, \dots$, donde μ y el s_τ son números finitos independiente de t .

II.1. Series de tiempo

Aunque las series de tiempo se trazan generalmente en un espacio de 2 dimensiones, una serie de tiempo implica realmente dos variables físicas, el eje horizontal representa el tiempo, mientras que el eje vertical representa el valor de la variable que cambia con respecto al tiempo. Estos dos ejes tienen unidades físicas independientes, tiempo y amplitud (diferencia de potencial, en este caso), esto es diferente de las curvas geométricas (tales como siluetas) las cuales están inmersas en un espacio de 2 dimensiones, donde ambos ejes representan la misma variable física.

Para determinar si una curva geométrica de 2 dimensiones es auto-similar, podemos hacer lo siguiente:

- (i) Llevar un subconjunto del objeto y escalarlo al mismo tamaño del objeto original, usando el mismo factor de la ampliación para el eje horizontal y vertical y entonces,
- (ii) comparar las características estadísticas del escalamiento del objeto con el objeto original.

En contraste para comparar correctamente un subconjunto de una serie de tiempo con la original, necesitamos dos factores de ampliación o de escalamiento (uno para el eje horizontal y otro para el vertical), ya que estos dos ejes representan diferentes variables físicas.

Un proceso dependiente del tiempo tiene propiedades de auto-similitud si

$$y(t) \stackrel{d}{=} H^H y\left(\frac{t}{H}\right) \quad (1)$$

Donde $\stackrel{d}{=}$ quiere decir que las medias de las características estadísticas de ambos lados de la ecuación son idénticas. Es decir un proceso auto-similar, $y(t)$, con un coeficiente H tiene la distribución idéntica de la probabilidad como una característica del proceso de escalamiento $H^H y\left(\frac{t}{H}\right)$ una serie de tiempo que ha sido reescalada:

En el eje X por un factor H : $t \rightarrow \frac{t}{H}$

En el eje Y por un factor de H^H : $y \rightarrow H^H y$

En la práctica es imposible determinar si dos procesos son idénticos estadísticamente. En sentido estricto se requiere tener sus funciones de distribución idénticas, incluyendo no sólo la media y la varianza, sino todos los momentos de orden más altos. Por lo tanto, uno se aproxima generalmente a esta igualdad examinando solamente las medias y las varianzas

(primero y segundo momentos) de las funciones de distribución para ambos lados de la ecuación (1).

La Figura 7.a muestra un ejemplo de una serie de tiempo auto-similar. En ella se observa que con la elección apropiada de los factores de escalamiento en el eje X y el Y, la serie reescalada se asemeja a la serie de tiempo original.

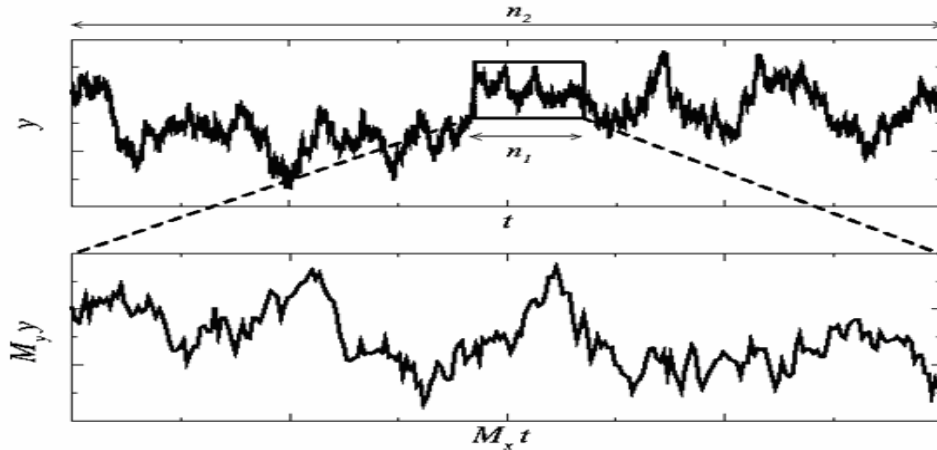


Figura 7.a) Dos ventanas de observación con escala de tiempo η_1 y η_2 , y una ampliación de la ventana más pequeña.

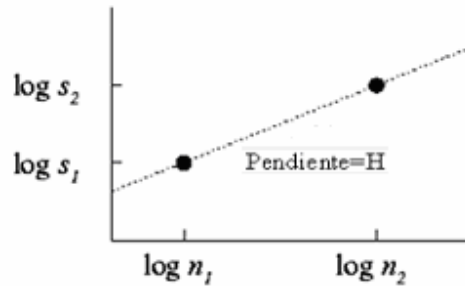


Figura 7.b) Diagrama de Log-Log de las escalas características de fluctuaciones.

El coeficiente H que se encuentra en la ecuación (1) se puede calcular por:

$$H = \frac{\ln(M_y)}{\ln(M_x)} \quad (2)$$

En la ecuación (2), M_x y M_y son los factores de ampliación apropiados a lo largo del eje horizontal y vertical, esto es la pendiente de la línea de la gráfica logarítmica va a ser igual al coeficiente H.

En la práctica no sabemos el valor del exponente, por lo que se debe de extraer este exponente de escalamiento (si existe uno) de una serie de tiempo dada. Por lo que es necesario estudiar la serie de tiempo en ventanas panorámicas con diversos tamaños y adoptar el criterio de la auto- semejanza definido anteriormente para calcular el exponente [14, 17, 19, 20].

II.2. Métodos de varianza aplicados al ventaneo escalado

Existen diferentes métodos de varianza aplicados al ventaneo escalado [15] para encontrar el coeficiente H de una serie de tiempo, estos métodos se basan en el conteo de caja (apéndice A), entre los cuales encontramos:

- a) Estándar
- b) Por de-tendencia lineal (quitando la tendencia, LD)
- c) Por de-tendencia puente (por saltos, BD)

a) Método estándar

La idea básica de este método se ilustra en la figura 7.a, donde vemos dos ventanas, la ventana 1 con tamaño horizontal η_1 y la ventana 2 con tamaño horizontal η_2 . La meta es encontrar los factores correctos de ampliación tales que la ventana 1 del escalamiento logre asemejarse a la ventana 2. El factor de ampliación a lo largo de la dirección horizontal se puede determinar como:

$$M_x = \frac{\eta_2}{\eta_1} \quad (3)$$

Para calcular el factor de ampliación a lo largo de la variable dependiente, se necesitan determinar las escalas características verticales de las ventanas 1 y 2. Una estimación razonable de las escalas características para las alturas verticales, es decir, las fluctuaciones típicas, puede ser definida usando las desviaciones estándar, denotadas como S_1 y S_2 , respectivamente [13, 14]. Así, tenemos

$$M_y = \frac{S_2}{S_1} \quad (4)$$

Sustituyendo M_x y M_y en la ecuación 2, obtenemos:

$$H = \frac{\ln(M_y)}{\ln(M_x)} = \frac{\ln(S_2) - \ln(S_1)}{\ln(\eta_2) - \ln(\eta_1)} \quad (5)$$

El coeficiente H es simplemente la pendiente de la línea que se encuentra entre estos dos puntos (η_1, S_1) y (η_2, S_2) en la gráfica de Log-Log (Figura 7.b).

Para calcular el coeficiente H de las series de tiempo, $y(t)$ con N valores, en el ventaneo escalado estándar, se realiza el procedimiento anterior pero con diversos tamaños de ventanas para obtener varios puntos y realizar una mejor aproximación del coeficiente H en la gráfica logarítmica.

b) Método por de-tendencia lineal (quitando la tendencia, LD)

Los métodos por “de-tendencia” son modificaciones del método estándar, por ejemplo, en el método por de-tendencia lineal, “LD”, al igual que el estándar, se divide en ventanas

pero no se obtiene la desviación estándar de ellas, en lugar de esto, se calcula la tendencia de las ventanas por medio de una regresión lineal usando todos los puntos contenidos en la ventana, y se resta de ella, con lo cual “quitamos la tendencia”. Se suma el error de cada punto de la ventana, obteniendo R_1 y R_2 si hacemos una analogía con el método estándar, ahora tendremos puntos (η_1, R_1) y (η_2, R_2) en la gráfica de Log-Log y H será la pendiente entre los puntos. Se realiza este procedimiento pero con diversos tamaños de ventanas para obtener varios puntos y realizar una mejor aproximación del coeficiente H en la gráfica logarítmica.

c) Método por de-tendencia puente (por saltos, BD)

El método por de-tendencia puente “BD”, es similar al LD, sólo que en éste al calcular la regresión lineal de la ventana, sólo se toman en cuenta el primero y el último punto de la ventana y se resta esta recta a todos los puntos de la ventana y se sigue el mismo procedimiento que con el método LD.

Para reducir el error de la estimación y la variación según [15], ciertos tamaños de ventana deben ser excluidos, como son las ventanas pequeñas que no pueden capturar cualidades de la auto-similitud de una señal correlacionada. También, en ventanas pequeñas, las desviaciones estándar se calculan usando solamente pocos puntos, y son por lo tanto parámetros estadísticos menos confiables. Debido a esto la pendiente en la gráfica logarítmica comienza a decrecer para los tamaños pequeños de ventana, tendiendo H a 0.5, cuando los puntos para los tamaños pequeños de la ventana se excluyen, el error de cálculo se reduce.

Los tamaños grandes de ventana también deben ser excluidos porque el valor de SD es más variable para los tamaños grandes de ventana. Así, al quitar los tamaños grandes de ventana en el cálculo de la regresión lineal del log-log se reduce el error de la estimación de H , se recomienda hasta $\frac{1}{4}$ del tamaño total de la serie y quitar las oscilaciones (si es que existen) del final [15]. Para el BD y el LD, las ventanas que consisten en solamente dos puntos deben ser excluidos siempre, porque cada uno de ellos tendrá solamente valores cero [15, 22].

Las series con $H > 0.5$ no son afectadas por la adición de ruido. Un problema es que al usar el método de varianza aplicados al ventaneo escalado para estimar H , la desviación estándar en ventanas pequeñas es específicamente sensible al ruido agregado porque hay pocos puntos por ventana. Así, una estrategia para evitar diagonal hacia $H = 0.5$ (H para el contaminante) es excluir los tamaños pequeños de ventana al calcular la regresión lineal para el $\log SD(n)$ vs. el $\log (n)$ [15].

II.3. Comparación de los métodos

En la Figura 8 se evalúa la precisión de los tres métodos: estándar, LD y BD [22-26]. Los parámetros que se varían son: longitud de la serie y valor de H, para una señal conocida y como se observa, las estimaciones son medias de 100 cálculos de H. En estas gráficas se pretende observar el comportamiento según estos dos parámetros para realizar la elección de un método.

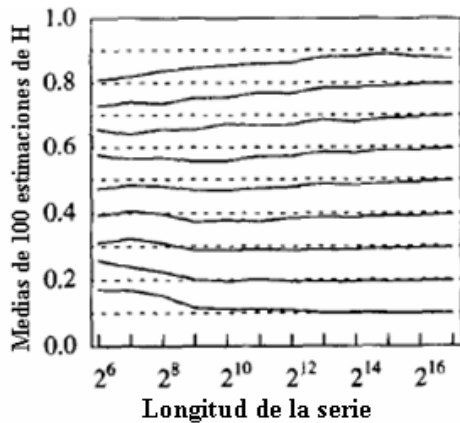


Figura 8.a) Estimación de H, método estándar.

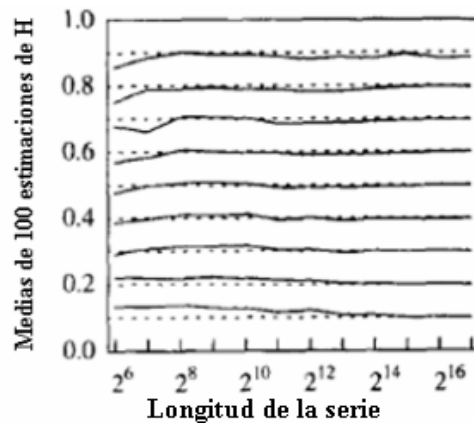


Figura 8.b) Estimación de H, método LD.

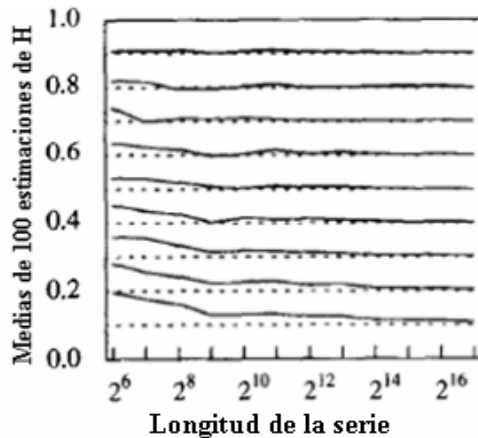


Figura 8.c) Estimación de H, método BD.

En la figura 8 se observa en cada una de las gráficas, se observa el valor real de H con una línea punteada y las aproximaciones de los métodos correspondientes son las líneas continuas.

En la figura 8.a se observa el comportamiento del método estándar, el cual tiene un gran error en series pequeñas (menores de 2^8), sin embargo para menores a 0.4 y de media longitud es buen estimador.

En la figura 8.b se ve que el método LD tiene un gran error en series pequeñas (menores de 2^8) pero tiene una buena estimación para series mayores a 2^8 y con H menor a 1.

En la figura 8.c tenemos que el método BD también tiene un error al principio para series pequeñas y de bajo valor de H, pero es un buen estimador para series largas de coeficiente pequeño o bien de series cortas y coeficiente H grande.

El método que se considera con una buena aproximación es el LD, que no tiene errores tan grandes en la estimación del coeficiente H como los métodos BD (en valores pequeños de H) y el estándar (en valores grandes de H). El método que se usa en este trabajo es una variante del método LD, el cual es el “Análisis de la de-tendencia a las fluctuaciones”, llamado DFA, por sus siglas en inglés, “Detrended Fluctuation Analysis”[23-26], este método tiene ventajas sobre los métodos de varianza aplicados al ventaneo escalado como son la detección falsa de auto-similitud y su insensibilidad al ruido agregado.

III. MÉTODO DFA (Análisis de la de-tendencia a las fluctuaciones)

III.1. Análisis de un proceso auto-similar

Para un proceso auto-similar con coeficiente $H > 0$, las fluctuaciones aumentan con el tamaño de la ventana conforme a la ley de potencias, estas fluctuaciones en ventanas grandes son exponencialmente más grandes que las de ventanas más pequeñas. La serie de tiempo es limitada, y si se considera el caso de la serie de tiempo mostrada en la figura 9.a, tomando un subconjunto de la serie de tiempo, notamos al parecer un patrón de auto-similitud.

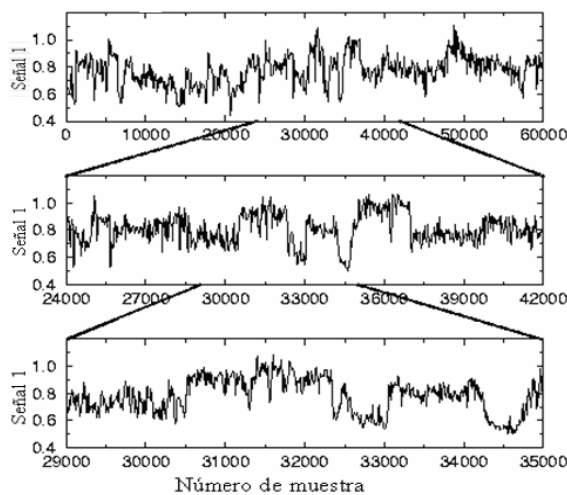


Figura 9.a) Ejemplo de una serie de tiempo.

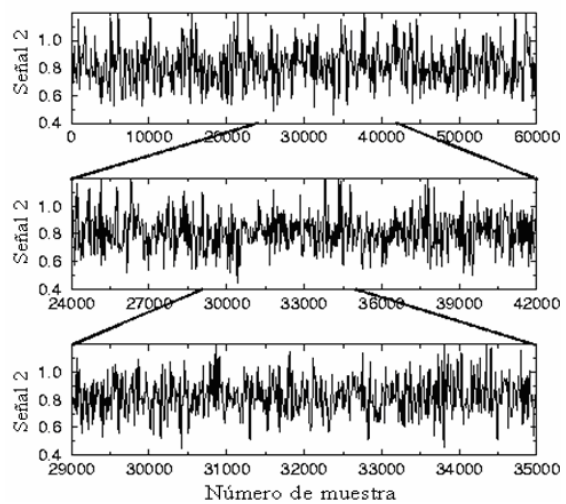


Figura 9.b) Ruido blanco.

Para visualizar la auto-similitud no se necesita reescalar el eje Y, esto es $M_Y = 0$, solamente es necesario reescalar el eje X, por lo que en la ecuación (2), el coeficiente H es 0, lo que no representa ninguna información. Si se considera otro ejemplo donde se selecciona “una serie de tiempo totalmente sin correlación” (Figura 9.b, ruido blanco), ésta también tiene un coeficiente $H = 0$. Sin embargo, los patrones en las figuras anteriores son absolutamente diferentes, un problema inmediato es cómo distinguir el coeficiente trivial 0 del último caso de ruido del coeficiente no trivial 0 calculado para los primeros datos.

Para ello se desarrolla una solución innovadora para este problema central en el análisis de series de tiempo [23, 25, 26], se resuelve al estudiar las características fractales de la acumulación de las series de tiempo (integración), en lugar de las características de la señal original. Por lo que la serie de tiempo limitada original pasa a ser una señal integrada, este es un paso crucial en análisis fractal de series de tiempo, y es la razón de que se apliquen técnicas de análisis fractal después de la integración de la serie de tiempo original [14].

III.2. Descripción del método DFA

El análisis DFA (Detrended Fluctuation Analysis, esto es Análisis de la de-tendencia a las fluctuaciones) de una serie de tiempo limitada se puede pasar a un proceso auto-similar por medio de la integración, pero el problema que surge con ello, es que las series resultantes son a menudo altamente no estacionarias, por lo que el procedimiento de integración aumenta la no estacionariedad de la serie.

Para superar este problema (series no estacionarias), se introduce el análisis DFA, que sobre otros métodos convencionales (métodos de varianza aplicados al ventaneo escalado, estándar o BD [15]) evita la detección falsa de la auto-similitud [24, 26]:

El algoritmo de DFA trabaja mejor para ciertos tipos de series de tiempo no estacionarias, como se observó en la comparación de métodos (tiene las mismas características que el método LD). Para ilustrar el algoritmo DFA, se utiliza la serie de tiempo de la Figura 10 como ejemplo.

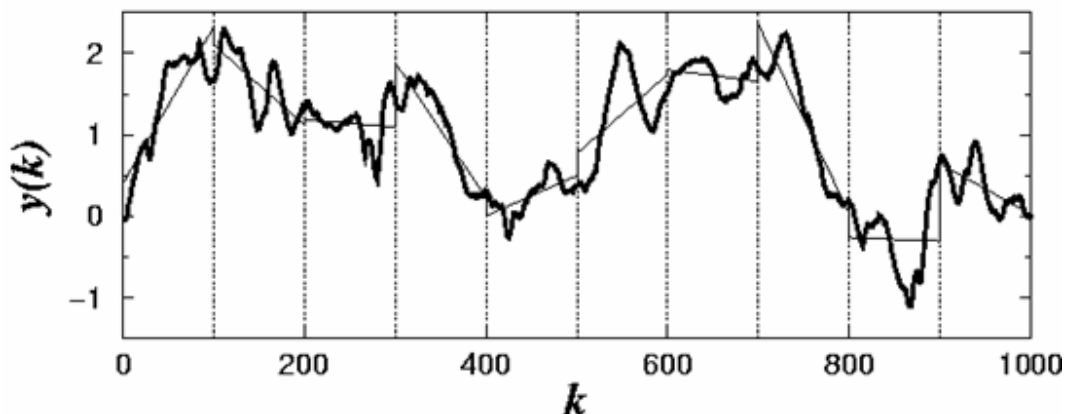


Figura 10. Método DFA.

Como primer paso, la serie de tiempo (de longitud total N) es integrada de la siguiente forma:

$$y(k) = \sum_{i=1}^k [X(i)] \quad (6)$$

donde X(i) es el i-ésimo valor (al momento de obtener el valor de F(n) se resta implícitamente el valor de la media [26]), según lo anterior, esta integración es el paso de la serie de tiempo original a un proceso auto-similar.

Entonces se mide la escala característica vertical de la serie de tiempo integrada, para ello, la serie de tiempo integrada se divide en ventanas de longitud igual a “n”. En cada ventana de longitud n, se ajusta una recta por mínimos cuadrados a los datos (que representan la tendencia en esa ventana, ver figura 10).

La variable dependiente de los segmentos de recta se denota como $y_n(k)$. Después se quita la tendencia a la serie integrada $y(k)$, esto se hace restando la tendencia local en cada ventana $y_n(k)$ para un tamaño dado n de ventana, el tamaño característico de la fluctuación $F(n)$ para la serie integrada y eliminada su tendencia se calcula por:

$$F(n) = \sqrt{\frac{1}{N} \sum_{k=1}^N [y(k) - y_n(k)]^2} \quad (7)$$

$F(n)$ es similar pero no idéntica a la varianza S , éste cálculo se repite sobre todas las escalas de tiempo (tamaños de ventana) para proporcionar una relación entre $F(n)$ y el tamaño n de ventana. Normalmente, $F(n)$ aumentará con el tamaño n de ventana, ya que tienen una relación lineal en una gráfica logarítmica, donde este tipo de relación indica la presencia de auto-similitud; las fluctuaciones en ventanas pequeñas se relacionan con las fluctuaciones en ventanas más grandes conforme a la ley de potencias. Por lo que el siguiente paso será graficar en una escala logarítmica $F(n)$ vs. n , y la pendiente de la línea que relaciona del $\text{Log}(F(n))$ vs. el $\text{Log}(n)$ determina el coeficiente H (coeficiente auto-similitud) [14, 26]. En la gráfica logarítmica se observa el comportamiento del coeficiente H , por lo que si se tiene una inflexión en ella, denota un cambio de coeficiente, esto es, que la señal presenta diferentes tipos de escalamiento, y estamos en presencia de una señal multifractal [25].

III.3. Los coeficientes de Hurst y de Autocorrelación

La relación del coeficiente H de una serie de tiempo integrada y su coeficiente de autocorrelación $C(\tau)$, de la señal original (no integrada) es la siguiente:

- $H=0.5$ del valor integrado, $y(k)$, para el ruido blanco que está totalmente sin correlación, corresponde la función de autocorrelación $C(\tau)$, es 0 para cualesquier τ (intervalo) diferente de cero [2, 4].
- $0.5 < H \leq 1$ indica correlaciones de largo alcance *persistentes* como la ley de potencias.
- $0 < H < 0.5$, indica *contra-correlaciones*, tales que los valores grandes son más probables ser seguidos por valores pequeños y viceversa [3].
- $H > 1$, las correlaciones existen pero dejan de estar de una forma de la ley de potencias.
- $H=1.5$ indica el ruido fBm, que es la integración del ruido blanco.

El coeficiente H puede también verse como indicador de las “asperezas” de la serie de tiempo original: cuanto más grande es el valor de H , la serie de tiempo es menos “áspera” [14, 17, 19]. Es importante saber esta relación, ya que se trabaja con la serie integrada en el método DFA.

III.4. Algoritmo del método DFA

El análisis debe contar con un conjunto de señales que tengan principalmente dos características, ser obtenidas a la misma frecuencia de muestreo y que el intervalo de tiempo sea igual, la razón es que si se varía alguno de estos dos parámetros se tiene como resultado una mayor o menor cantidad de puntos por señal, lo cual afecta a nuestro análisis, ya que la precisión de este proceso depende de la longitud de las señales. Estas señales se deben tener en código ASCII (American Standard Code for Information Interchange), en formato texto de dos columnas, la primera con el tiempo y la segunda con su amplitud. Una vez observadas las características del conjunto de señales, podemos seguir los pasos del algoritmo:

1. **Cargar el conjunto de señales.** Se deben de escoger y cargar las señales en el orden que sea conveniente, ya que se tiene como resultado de todo el proceso una gráfica del coeficiente H vs. el número de señal, por lo que el orden en que carguemos las señales va a ser muy importante para observar el comportamiento del proceso. Esto también ayuda a tener a las señales en memoria para manejarlas durante todo el proceso.
2. **Graficar las señales.** Se grafican las señales antes de realizar cualquier operación sobre ellas, ya que se puede observar la forma de onda de cada una, la cantidad de ruido, si alguna señal está contaminada, o bien, si tiene alguna anomalía que afectaría a nuestro análisis. Se realizan dos gráficas, una de amplitud vs. tiempo y otra de amplitud vs. número de puntos de cada señal, es conveniente ver estas dos gráficas, ya que se puede ver el rango de interés, tanto en tiempo como en número de puntos.
3. **Acondicionar las señales.** Este proceso se tiene que hacer por dos razones, una de las cuales es la forma en que entrega la señal el osciloscopio digital, ya que por la forma de obtener las señales (realizando promedios para tener menor cantidad de ruido blanco), tienen una componente de directa, la cual se debe eliminar, también se tiene una parte de la señal con tiempos negativos, ya que el osciloscopio guarda la señal momentos antes del disparo del canal A, por lo que se tiene una pequeña parte negativa de la señal, la cual también se debe quitar. La segunda razón es que se debe escoger el rango de interés de la señal, ya que al final de la señal se encuentra ruido de igual magnitud que la señal, y también con una señal de longitud mayor, el número de operaciones se incrementa y por lo tanto tarda más el proceso. También se debe normalizarla respecto a su desviación estándar, ya que este análisis depende de la desviación estándar de la señal, así que al normalizar con respecto a ésta, se convierte sólo en una constante, por lo cual se ahorra tiempo en el cálculo del coeficiente H .
4. **Cálculo del coeficiente H .** Este coeficiente se tiene que obtener para cada señal del conjunto, por lo que se calcula en el orden en que las señales fueron cargadas; para este cálculo se deben tener los siguientes datos:

- Número de señales

- Longitud de las señales N (número de puntos de cada una)
- Orden de regresión (por defecto 1)
- Mínimo y máximo tamaño de ventana

Dependiendo del orden de regresión elegido (en este caso siempre es 1, para una regresión lineal), se puede calcular el tamaño mínimo de ventana, y con la longitud de la señal, se calcula el tamaño máximo de ventana permitido, que son necesarios para el cálculo del coeficiente H. Este cálculo se realiza en el orden siguiente:

- a) **Calcular el tamaño máximo y tamaño mínimo de ventanas.** Se divide la señal en ventanas o bien en subconjuntos de series de puntos, las cuales van a comenzar desde un tamaño mínimo que se calcula con el orden de regresión dado, esto es $2(n_{fit} + 1)$, siendo n_{fit} el orden de regresión. El tamaño máximo de ventana es $\frac{1}{4} N$ (longitud total de nuestra señal). Como ambos datos son conocidos se calculan en primera instancia estos dos parámetros.
- b) **Calcular los tamaños de ventana a utilizar.** Se deben calcular los tamaños de ventanas a utilizar en el análisis, ya que se grafican en una escala logarítmica; para los valores más grandes los puntos contenidos en ella convergen entre sí, por ello, no es necesario calcular tamaños subsecuentes de ventanas grandes, por lo que debe haber una razón entre cada una, esto es, para tener una menor cantidad de operaciones y por ende un proceso más rápido.
- c) **Seleccionar la señal a analizar.** Al tener un conjunto de señales, se comienza por la primera señal cargada en la memoria, hasta llegar a la última, en el orden en que son cargadas.
- d) **Integrar la señal.** Se debe integrar la señal seleccionada antes de comenzar, esto ayudará a que no se detecte una falsa auto- semejanza como se explica en el método DFA. En este paso sólo se toma en cuenta el valor de la magnitud de la señal, los incrementos de tiempo correspondientes se tomarán como valor unitario, para realizar sólo una suma acumulativa de los puntos de la señal.
- e) **Escoger el tamaño de ventana.** Se selecciona el tamaño de ventana, ya que se hará un barrido por todos los tamaños calculados en el paso b), comenzando desde el mínimo (la menor cantidad de puntos que podemos tomar para formar una ventana) hasta el mayor para cada una de las señales del conjunto.
- f) **Dividir la señal integrada en ventanas.** Se divide la señal en un subconjunto de tamaño n, según el tamaño de ventana elegido en el paso anterior, estas ventanas no tienen traslape, con esto se quiere decir que terminando una ventana (un subconjunto de puntos de la señal), comienza la siguiente, sin que algún punto pertenezca a dos ventanas simultáneamente.

- g) **Escoger la ventana.** Se va a tener N/n número de ventanas (del mismo tamaño) a lo largo de la señal, se va a escoger primero la ventana que se encuentra al comienzo de la señal, hasta llegar a la del final para continuar con proceso.
- h) **Calcular la regresión lineal.** Una vez que se escoge la ventana, se obtiene la regresión lineal del conjunto de puntos que contiene, esto se realiza por mínimos cuadrados, con ello, se obtiene una aproximación lineal.
- i) **Cálculo del error.** Con la aproximación lineal de la ventana, se obtiene un error cuadrático restando la tendencia lineal calculada, a la serie integrada. Este resultado se guarda para realizar la suma de los errores de todos los puntos de las ventanas del mismo tamaño de la señal, y después se obtiene el promedio del error cuadrático, y se calcula la raíz de este.
- j) **Cambiar tamaño de ventana.** Siguiendo los pasos del e) al i) se obtiene $F(n)$, que es la suma de los errores, para una ventana de tamaño “ n ”, ahora hay que cambiar el tamaño de ventana al siguiente tamaño calculado en el paso b) y realizar nuevamente los pasos del e) al i), continuando hasta alcanzar aproximadamente el máximo tamaño de ventana.
- k) **Cambiar de señal.** En el paso c) se selecciona la señal con la cual se hizo todo el análisis hasta el paso j), ahora se tiene que cambiar la señal para realizar nuevamente los pasos anteriores, esto es del c) al j), hasta terminar con el conjunto de señales cargado; guardando los valores calculados $F(n)$ y n , de cada una de las señales.
- l) **Graficar.** Con los pasos anteriores se hace el barrido de todos los tamaños de ventana para el conjunto de señales. Con ello se tienen los valores de $F(n)$ (para cada señal) correspondientes a valores de “ n ” (tamaños de ventana). Ahora se obtiene una gráfica logarítmica, usando “ n ” como variable independiente y $F(n)$ como dependiente.
- m) **Observar comportamiento.** Se tiene que observar el comportamiento de los puntos obtenidos, ya que para ciertos valores (los valores más grandes de ventana) como se menciona, comienzan fluctuaciones grandes en períodos cortos, por lo que no se deben tomar en cuenta para el análisis, ya que el cálculo sería susceptible a tener un error en la estimación del coeficiente H .
- n) **Calcular el coeficiente H .** Una vez que se eliminan las fluctuaciones, se realiza el cálculo del coeficiente H (el cual se obtiene para cada una de las señales) por medio de su gráfica logarítmica, en ella se tiene que calcular la pendiente que describe la curva de cada señal, lo cual se hace por mínimos cuadrados.
- o) **Graficar el coeficiente H .** Se debe graficar el coeficiente H vs. el número de señal para observar el comportamiento, el resultado es una curva para el comportamiento del material en cuestión.

5. **Despliegue de resultados.** Observamos los resultados en la gráfica de H vs. el número de señal, en ella se tiene el comportamiento del coeficiente H, lo que se pretende es analizar esta curva y observar si tiene una tendencia.

El siguiente paso de realizar el algoritmo del método DFA es un diagrama de flujo, ambos (algoritmo y diagrama de flujo) son para una mejor y rápida programación del método DFA.

III.5. Diagrama de flujo del método DFA

Un diagrama de flujo representa la esquematización del algoritmo, el cual muestra los pasos o procesos a seguir para alcanzar la solución del problema, también muestra el sistema como una red de procesos funcionales conectados entre sí por líneas de datos que permite describir el movimiento de ellos a través del sistema, describe lugares de origen y destino de los datos, así como transformaciones a las que son sometidos estos.

En nuestro caso se presenta el algoritmo únicamente por 5 módulos principales:

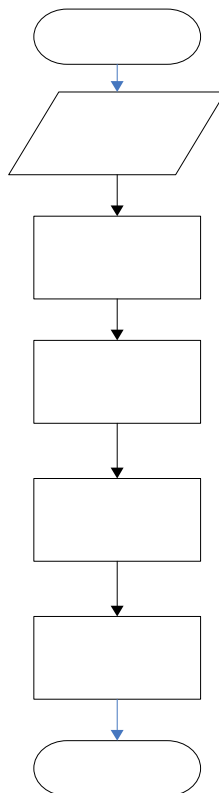


Figura 11. Algoritmo dividido por módulos.

Ahora se presenta cada módulo con sus características principales:

1. Primer módulo (figura 12) donde se carga el conjunto de señales.

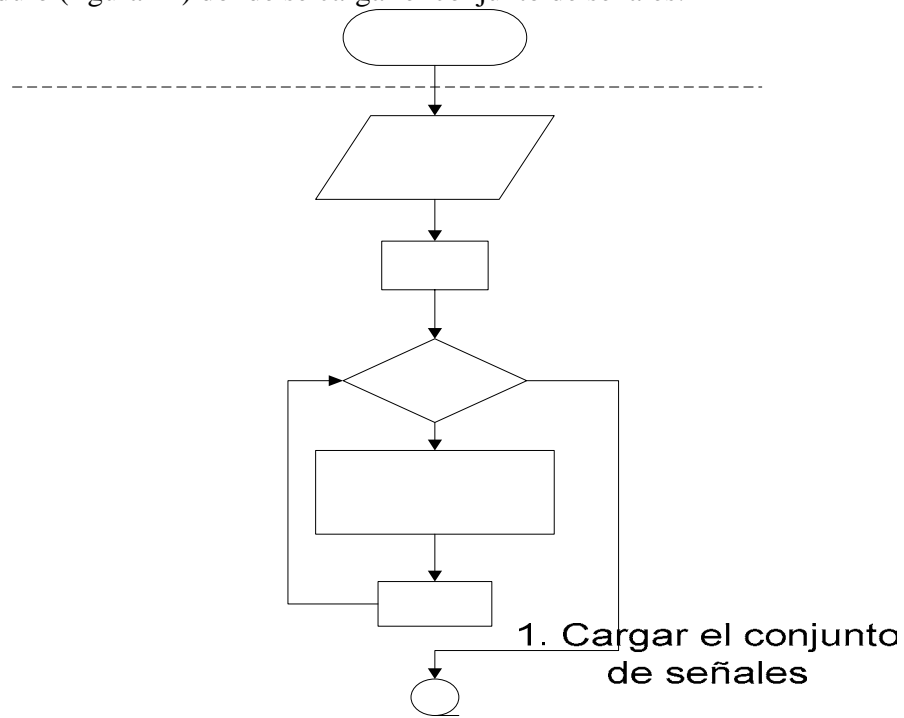


Figura 12. Primer módulo.

2. Segundo módulo (figura 13) en donde se graficar las señales.

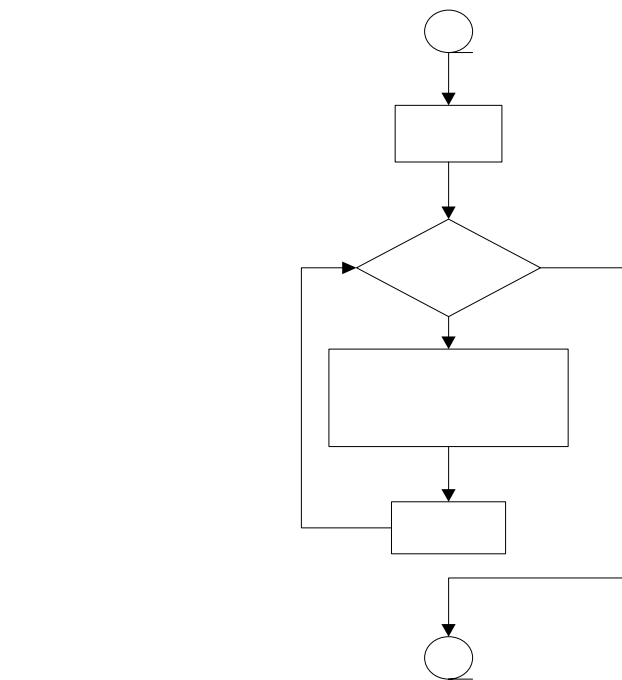


Figura 13. Segundo módulo.

3. Tercer módulo, en donde se realiza el acondicionamiento de las señales; se compone de dos partes (figuras 14 y 15), la primera de corte y la segunda de normalización y cambio de referencia a cero.

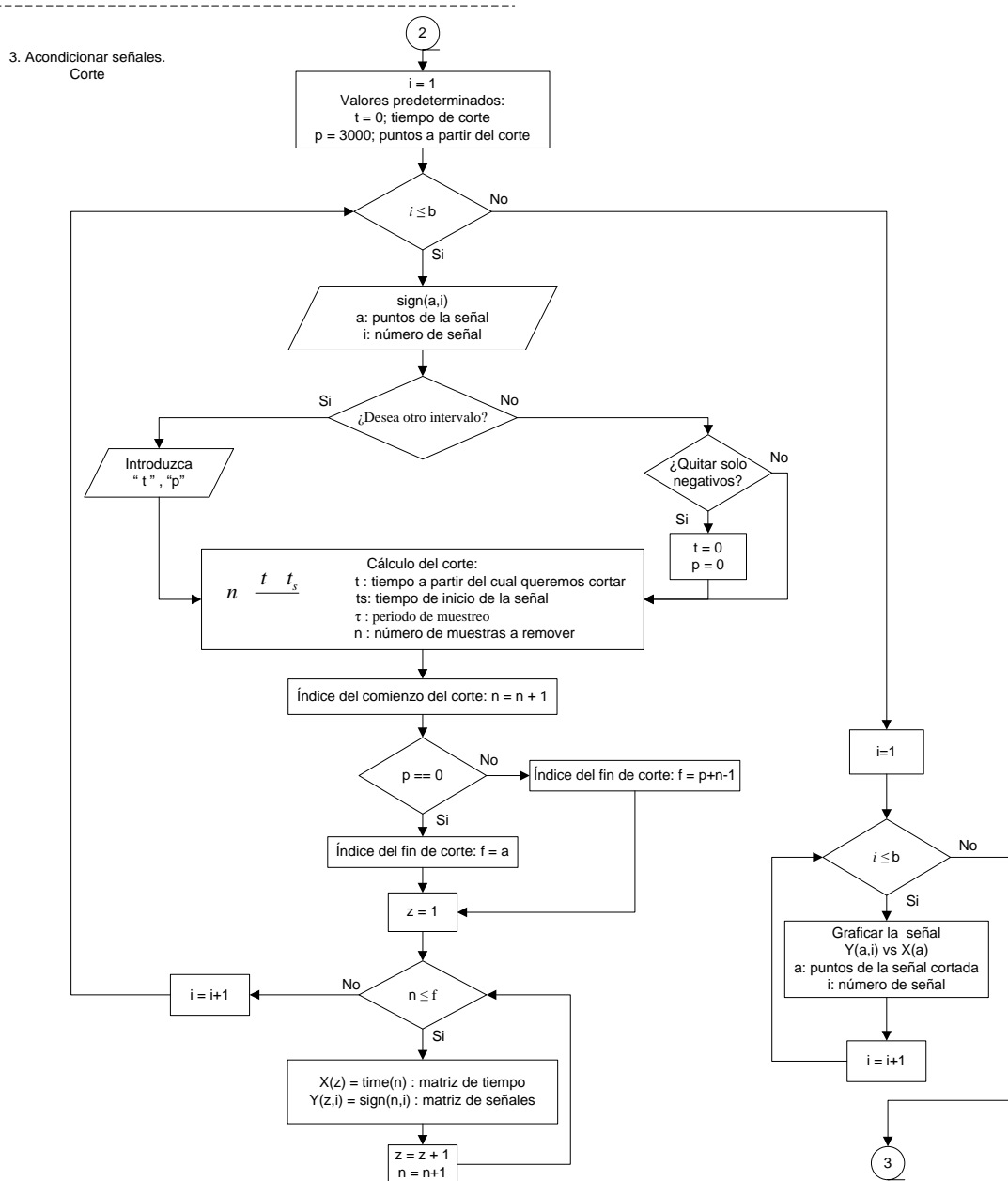


Figura 14. Tercer módulo, corte.

3. Acondicionar señales.
Normaliza y referencia a cero

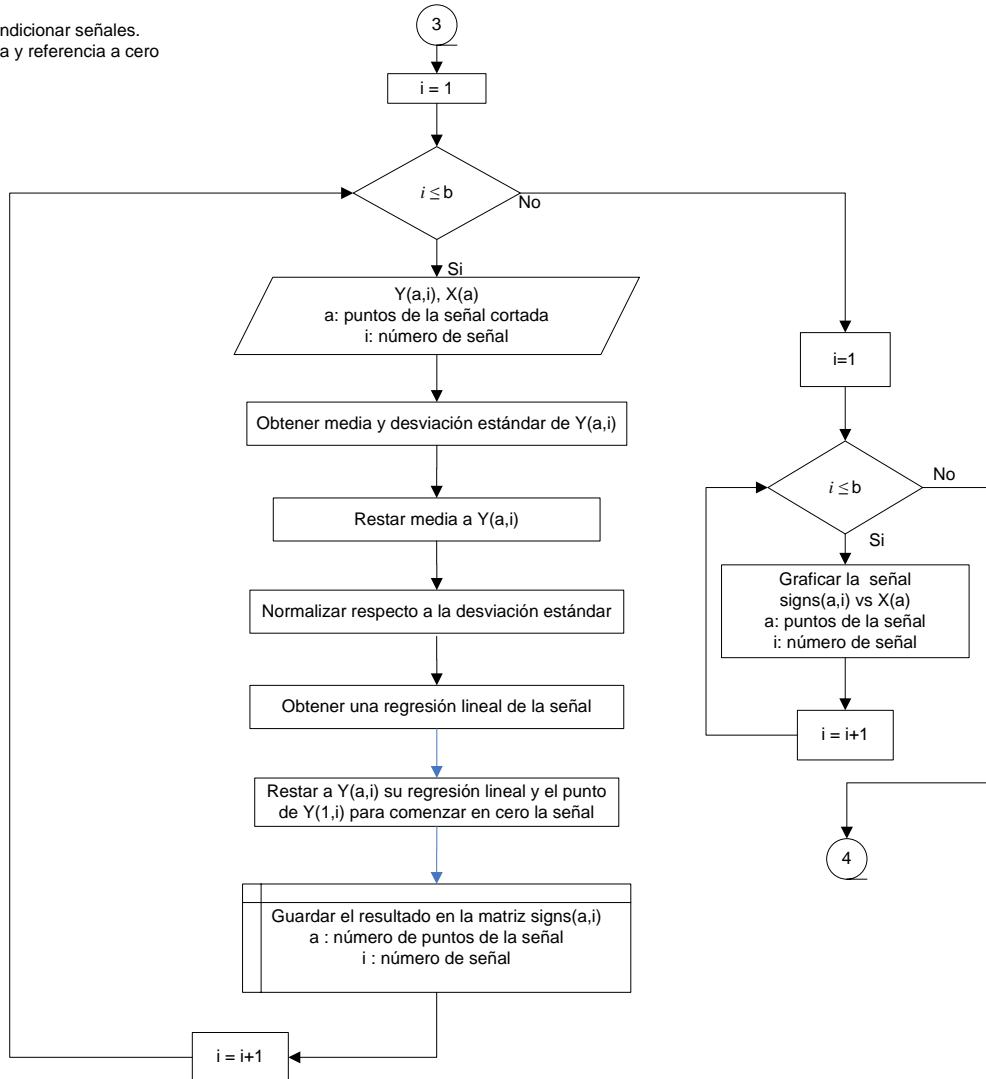


Figura 15. Tercer módulo, normalización y cambio de referencia.

4. Cuarto módulo (figura 16.a y 16.b) donde se hace el cálculo del coeficiente H.

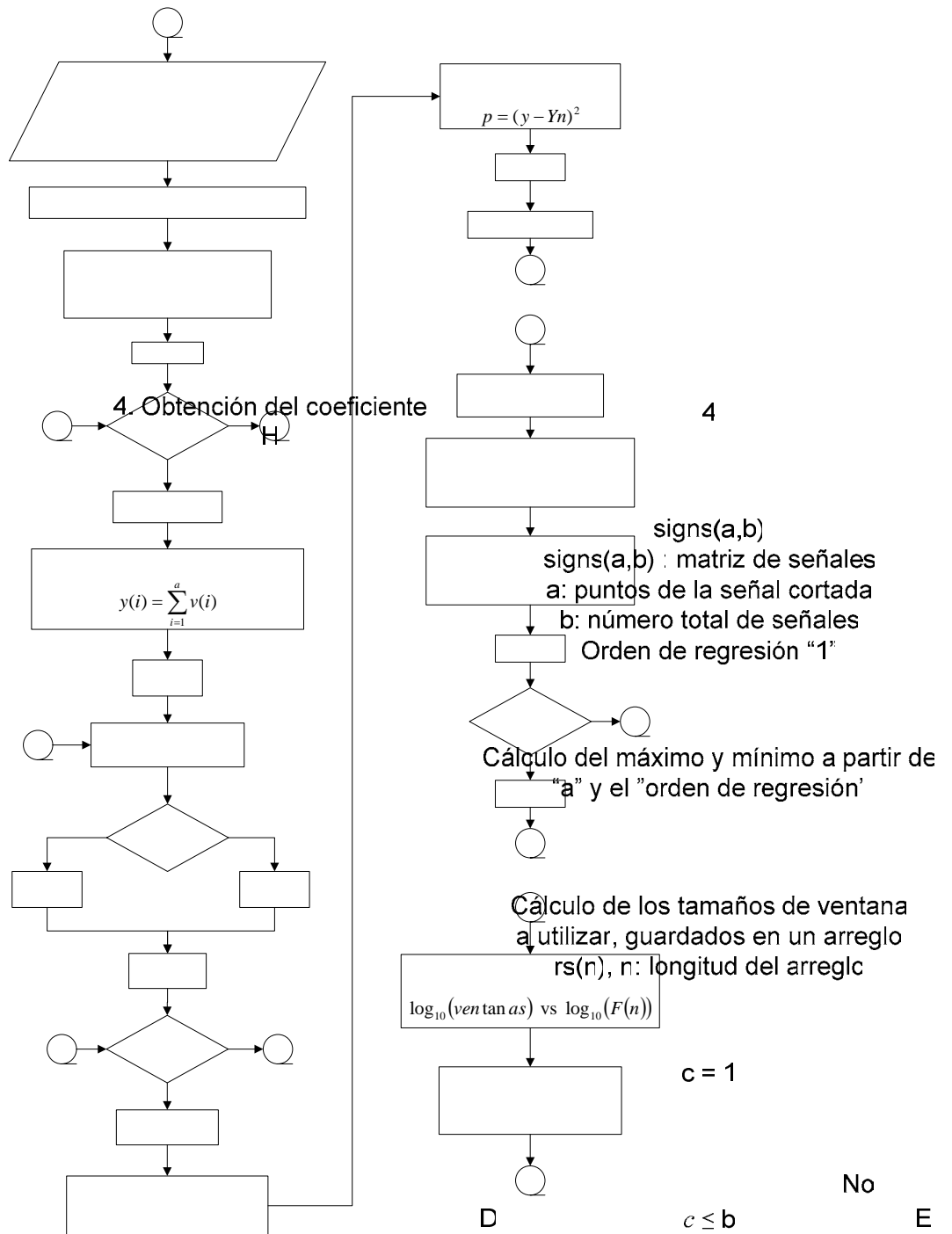


Figura 16.a) Cuarto módulo, cálculo del coeficiente H.

4. Cálculo del coeficiente H

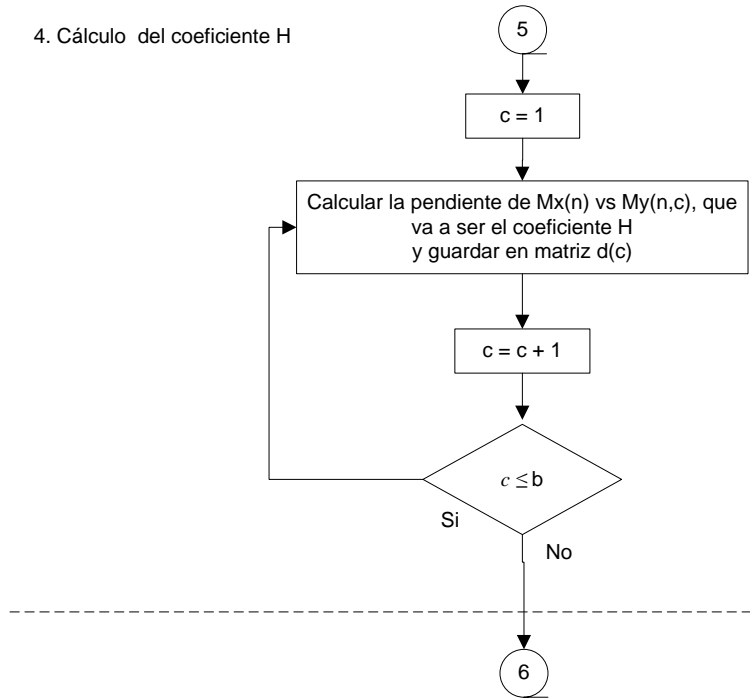


Figura 16.b) Cuarto módulo, cálculo del coeficiente H.

5. Quinto módulo (figura 17) donde se despliegan los resultados del método DFA.

5. Despliegue de resultados

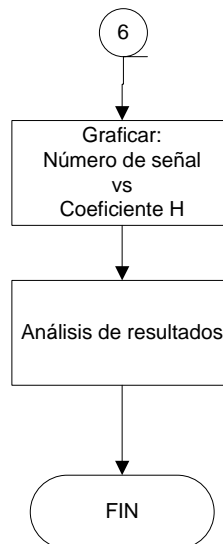


Figura 17. Quinto módulo, despliegue de resultados.

III.6. Programación del algoritmo del método DFA

Basándose en la problemática planteada, surge la necesidad de seleccionar una plataforma para el desarrollo de esta aplicación. En primera instancia, se piensa desarrollarlo en lenguaje C, ya que por sus características como lenguaje de programación, trabaja bien con los recursos de la máquina, pero el lenguaje C obtiene su eficiencia usando el manejo a bajo nivel de los recursos como la memoria (esta parte se maneja en el código) por lo que se opta por escoger otro lenguaje de programación más eficiente. Se ve en la posibilidad de desarrollarlo en otros lenguajes como son C++ o bien Visual Basic, pero otro inconveniente que tienen estos lenguajes es que se debe realizar en todos los casos (C, C++ y Visual Basic) un módulo para la graficación y manipulación de resultados, por lo cual, la aplicación se desarrolla en el lenguaje de programación MatLab, ya que por su flexibilidad permite una rápida implementación, y realizar una interfaz gráfica para la aplicación de nuestro programa, con esto se logra que exista una interactividad y un manejo de datos no definidos desde el código fuente, además de que contamos con distintos tipos de librerías para la graficación de las señales, esto ayuda para la visualización de los resultados de una manera sencilla. El código se muestra en el apéndice B. La interfaz gráfica del programa es la siguiente:

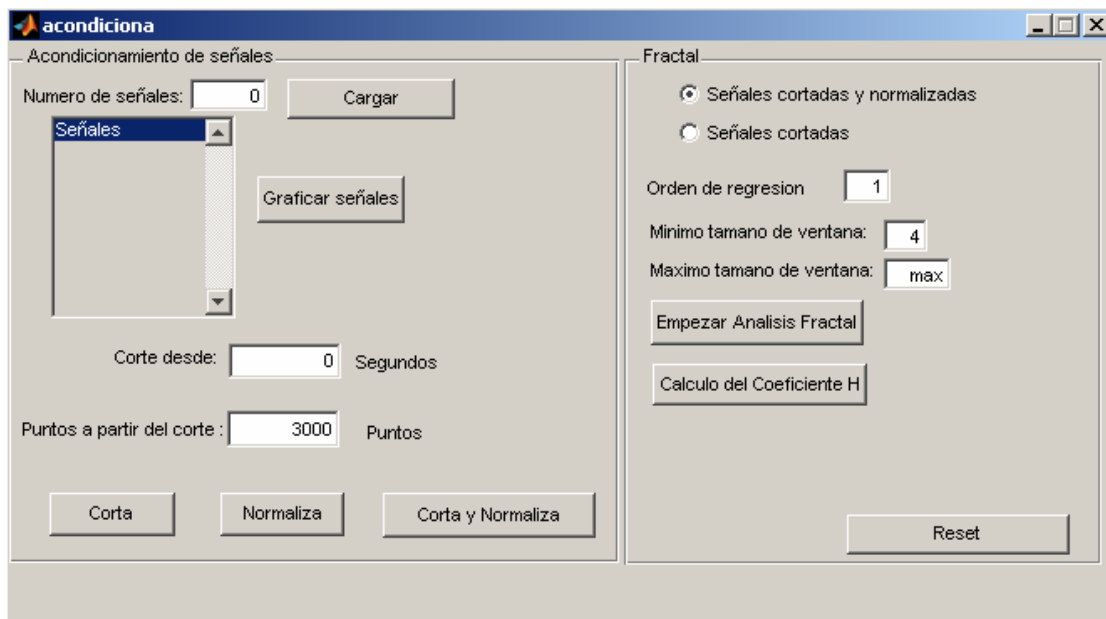


Figura 18. Interfaz del programa.

Como se observa en la figura 18, en la parte superior izquierda se indica que el nombre del programa es “acondiciona.m”. Como primer paso, se debe dar el número de señales que se quieren cargar (tamaño del conjunto), y presionar el botón “Carga”, para seleccionar los archivos que las contienen, esto se debe hacer en el orden que para nosotros sea conveniente, al realizar esto, se observan los archivos con extensión “.dat” que es la extensión de código ASCII que reconoce el programa.

Una vez cargadas, el cuadro gris de la izquierda con la palabra “Señales” despliega el nombre del archivo de cada señal, en el orden en que se cargaron. Con el botón que se encuentra a un lado “Graficar señales”, se grafican todas las señales, con esto se observa su comportamiento (como se mencionó anteriormente), despliega dos gráficas, una de amplitud versus tiempo y otra de amplitud versus número de puntos.

En la parte inferior se propone un corte, el cual podemos cambiar dependiendo del rango que nos interese de las señales, en los cuadros blancos se cambia el intervalo de corte dependiendo de la cantidad de puntos que se desean. Como se menciona, las señales tienen tiempos negativos por lo que el corte comienza en cero y a partir de éste ponemos los puntos que se desean. Si se quiere que solo corte el principio y nos deje la señal hasta el final solo se pone “0” y con ello solo cortará los tiempos negativos de la señal.

Los botones de la parte inferior son los de acondicionamiento de la señal, donde el primero “Corta” realiza el corte para nuestro rango dado y despliega dos gráficas para observar el corte de la señal una de amplitud vs. tiempo y otra de amplitud vs. número de puntos. El siguiente botón “Normaliza” va a normalizar el rango de la señal elegido con el botón anterior, con respecto a su desviación estándar y cambia la referencia a cero (quita la componente de directa de la señal), y despliega dos gráficas del rango de interés normalizado y respecto a cero, una de amplitud vs. tiempo y otra de amplitud vs. número de puntos. El tercer botón “Corta y Normaliza” realiza el corte y normalización en un solo paso y muestra las gráficas del rango de interés normalizado y respecto a cero. Los botones anteriores parecen redundantes, pero al realizar el proceso podemos hacer un seguimiento de nuestra señal para ver paso a paso el avance de ella en nuestro proceso, esto es muy útil para observar si existen errores en alguna parte de éste.

Una vez hecho el acondicionamiento, se continua con el cálculo del coeficiente H, es importante mencionar que al realizar el acondicionamiento, contamos con los datos necesarios para hacer este cálculo, esto es longitud de la serie (rango de interés) y por defecto el orden de regresión es igual a 1, con lo que los cuadros de mínimo y máximo tamaño de ventana se llenan al realizar el acondicionamiento por los datos calculados por el programa. El botón “Empezar análisis” va a realizar el barrido desde el tamaño mínimo de ventana hasta el máximo para cada una de las señales del conjunto, desplegando una gráfica logarítmica de $F(n)$ vs. n de todas las señales. Con ello podemos observar el comportamiento de la gráfica logarítmica y ver si existen desviaciones u oscilaciones en esta gráfica. Una vez que hicimos esto podemos hacer el cálculo del coeficiente H con el botón “Cálculo del coeficiente H”, el cual despliega una gráfica del número de señal vs. el coeficiente H de cada una.

En el siguiente capítulo se ve la aplicación de este programa a señales OA, también se muestra la interfaz del programa (figura 23) con un conjunto de señales cargado, y se describe el procesamiento de estas señales.

IV.1. Procedimiento experimental

En este trabajo se aplica el método DFA para el cálculo del coeficiente H de un conjunto de señales opto-acústicas obtenidas durante el proceso de foto-polimerización de una resina dental, con el objetivo de dilucidar su comportamiento fractal durante su proceso de transformación de resina líquida a sólida.

Materiales y muestras. Los monómeros empleados en este estudio son BisGMA y TEGDMA y fueron adquiridos en Polysciencis, Inc. A. Las muestras de la resina compuesta están constituidas por una mezcla equimolar de BisGMA/TEGDMA con una relación en peso de 70/30 % respectivamente, a la cual se le adiciona 0.2% de camforquinona (CQ) y 0.3% de n, n-dimetil-p-toluidina (DMPT) como activadores. Para seguir la evolución de la foto-polimerización, se prepararon cinco muestras dentro de contenedores vidrio con base rectangular de 2x22mm y una altura de 15mm, en donde se introduce la resina por gravedad, obteniéndose así un volumen controlado de la misma. La luz usada para la foto-polimerización es aplicada con una lámpara dental.

El sistema de medición. En la Fig. 19, se muestra un diagrama esquemático de la configuración experimental. La muestra es iluminada por su parte superior con una lámpara dental (Delux, 75 W, $\lambda=468-480\text{nm}$), la cual es manualmente controlada para producir intervalos de luz de un segundo. Las ondas ultrasónicas son generadas sobre la superficie anterior del contenedor protegida con papel aluminio aplicando pulsos láser (Nd:YAG de Continuum, 10Hz, $\lambda=532\text{nm}$, $\tau = 12\text{ns}$, $E = 2 \text{ mJ/mm}^2$). Un sensor piezoeléctrico es adherido con grasa de silicona en la parte posterior de la muestra. La salida de voltaje del sensor es adquirida y registrada con un osciloscopio de 500 MHz (Tektronik TDS 540). Las variaciones de intensidad que presenta el láser pulsado se registran con un sensor piroeléctrico (RjP-735 de Laser Precision Corp), denotado como “Medidor de energía”. Se emplea la señal de un fotodiodo rápido para disparar el osciloscopio y acoplar el tiempo de arriba con la señal de disparo. El haz pulsado se expande y se colima con un par de lentes y se enfoca sobre la muestra con un diámetro de 8 mm. Se promedian alrededor de 500 pulsos por cada señal para reducir el ruido aleatorio en la medición.

El procedimiento de medición es el siguiente: a) la mezcla previamente preparada del copolímero es vaciada por gravedad en el contenedor, el sensor se fija en la parte posterior y se coloca para que el pulso láser incida en la parte anterior en la misma dirección, b) se aplica la luz para la foto-polimerización durante *un segundo*, c) inmediatamente se aplican los pulsos láser y se registran las señales. El procedimiento anterior se repite hasta alcanzar la vitrificación del polímero. Todo el procedimiento es registrado en función del tiempo.

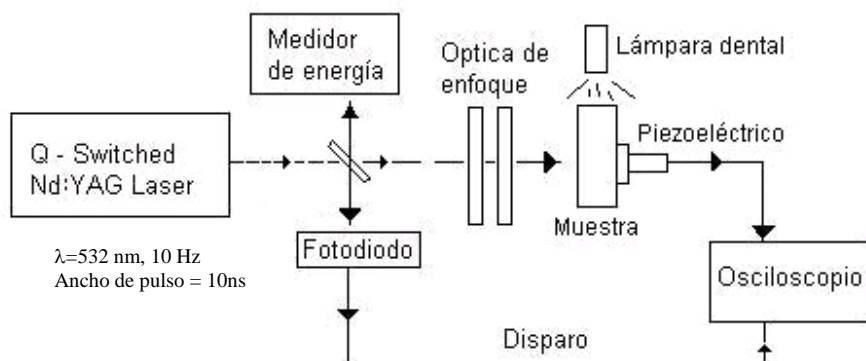


Figura 19. Diagrama experimental para la obtención de señales OA de una resina dental durante su foto-polimerización.

La adquisición de señales se realiza de la siguiente forma:

1. Se adquiere la señal OA de la resina líquida y se fija como tiempo cero.
2. Se irradia por un segundo la resina con luz UV e inmediatamente se inicia el proceso de foto-polimerización.
3. Se adquieren y almacenan señales OA a diferentes intervalos de tiempo a partir del pulso de luz UV aplicado hasta que en el osciloscopio ya no se observan cambios en la señal.
4. Los pasos anteriores se repiten hasta que la resina se solidifica totalmente.

Se registraron 55 señales OA durante todo el proceso de foto-polimerización. Cada señal es etiquetada con el nombre TEK0000#.WFM y poseen 15000 puntos cada una. En las siguientes tablas se dan los nombres de los archivos de señales OA adquiridas, en orden y en tiempo en función del número de pulsos de luz ultravioleta aplicados, con un ancho de un segundo sobre la resina y se dividen en siete intervalos:

Primer pulso, t=0 [s].

Tabla 1. Intervalo I.

Señal	Tiempo [s]	
	Inicio	Fin
TK 00006	30	57
TK 00007	105	130
TK 00008	170	202
TK 00009	242	275
TK 00010	313	350
TK 00011	387	415
TK 00012	453	479
TK 00013	519	557
TK 00014	596	624
TK 00015	665	704
TK 00016	749	793
TK 00017	839	878

Segundo pulso, t=909 [s].

Tabla 2. Intervalo II.

Señal	Tiempo [s]	
	Inicio	Fin
TK 00018	921	952
TK 00019	995	1024
TK 00020	1066	1110
TK 00021	1157	1207
TK 00022	1252	1280
TK 00023	1320	1360
TK 00024	1399	1437
TK 00025	1483	1512
TK 00026	1561	1600
TK 00027	1643	1694
TK 00028	1738	1790
TK 00029	1844	1879

Tercer pulso, $t=1916$ [s]

Tabla 3. Intervalo III.

Señal	Tiempo [s]	
	Inicio	Fin
TK 00030	1942	2004
TK 00031	2048	2086
TK 00032	2130	2175
TK 00033	2217	2258
TK 00034	2315	2365
TK 00035	2423	2484
TK 00036	2676	2720
TK 00037	2767	2797

Cuarto pulso, $t=2829$ [s]

Tabla 4. Intervalo IV.

Señal	Tiempo [s]	
	Inicio	Fin
TK 00038	2852	2891
TK 00039	2934	2969
TK 00040	3012	3052
TK 00041	3096	3133
TK 00042	3175	3220
TK 00043	3340	3378
TK 00044	3426	3479
TK 00045	3599	3673
TK 00046	3718	3762

Quinto pulso, $t=3785$ [s]

Tabla 5. Intervalo V.

Señal	Tiempo [s]	
	Inicio	Fin
TK 00047	3811	3841
TK 00048	3880	3918
TK 00049	4038	4071
TK 00050	4311	4351
TK 00051	4471	4510
TK 00052	4630	4673

Sexto pulso, $t=4699$ [s]

Tabla 6. Intervalo VI.

Señal	Tiempo [s]	
	Inicio	Fin
TK 00053	4719	4762
TK 00054	4882	4935
TK 00055	5063	5113
TK 00056	5233	5293
TK 00057	5413	5463

Séptimo pulso, $t=5481$ [s]

Tabla 7. Intervalo VII.

Señal	Tiempo [s]	
	Inicio	Fin
TK 00058	5507	5556
TK 00059	5676	5720
TK 00060	5840	5893

IV.2. Procesamiento de señales OA

B) Acondicionamiento de señales

Una vez que se tienen todos los archivos de las señales OA que se adquirieron durante el proceso de foto-polimerización, se acondicionan para su análisis. Primeramente se convierten a formato *.dat, ya que el osciloscopio las almacena en *.wfm, mediante un programa llamado cnvrfm.exe. Después se ejecuta el programa acondiciona.m desde el entorno de MatLab y se hace el cargado de las señales. Inmediatamente, se hace una visualización general de ellas graficándolas toda; ya sea en función del tiempo o por el número de puntos, ver figura 20.a y 20.b.

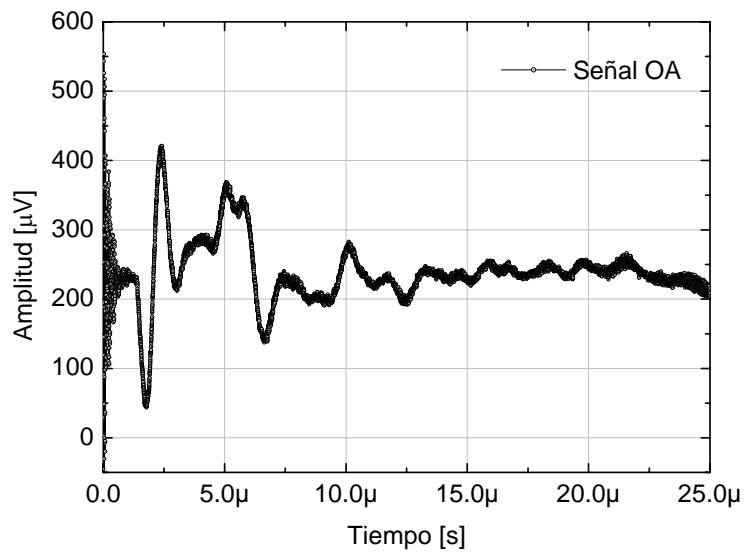


Figura 20.a) Señal opto-acústica, amplitud vs. tiempo.

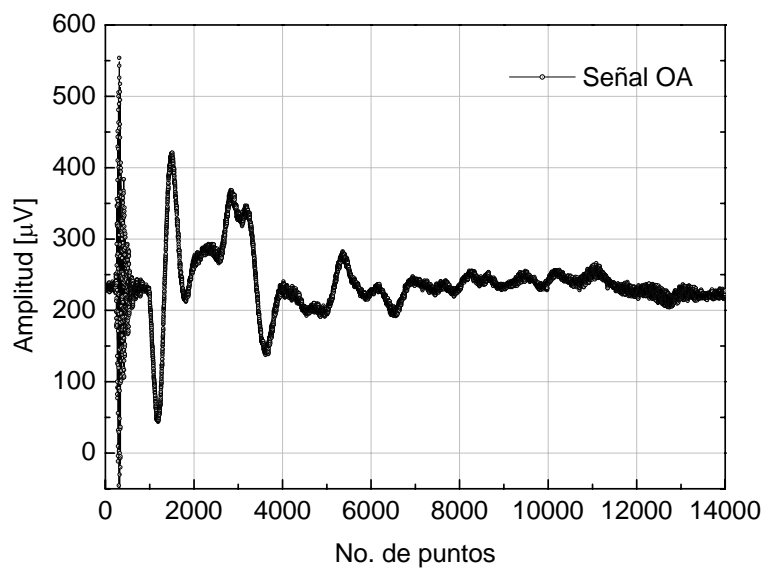


Figura 20.b) Señal opto-acústica, amplitud vs. número de puntos.

Como es de esperarse en muestras transparentes, el principio de la señal es la respuesta del sensor al pulso de luz láser que pasa directo y es casi constante durante todo el experimento por lo que se observa en todas las señales. Estos datos son eliminados ya que no aportan información de las características de la muestra. A estas señales se les quitan los primeros 800 puntos (aproximadamente un 1 [μs] en tiempo, ver la Figura 21).

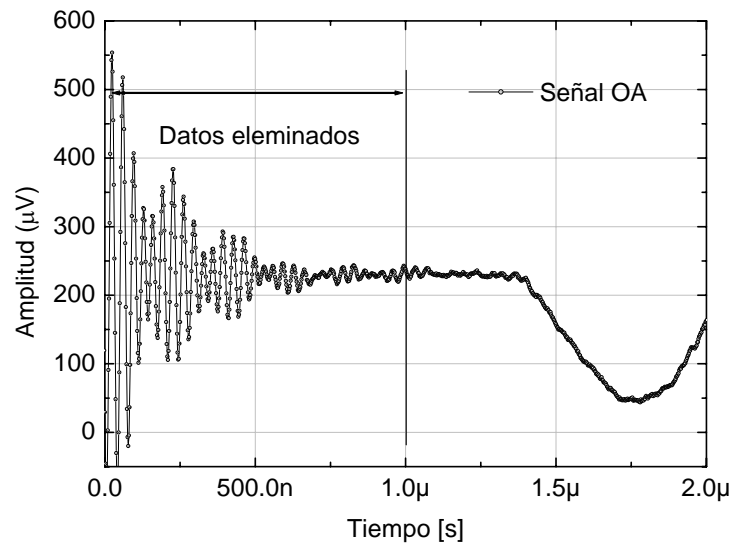


Figura 21. Señal opto-acústica cortada.

Una vez eliminados los primeros 800 datos de todas la señales, se normalizan respecto a su desviación estándar y se nivelan a la referencia de $V=0$ (volts) y $t = 0$ (seg), para quitar cualquier componente de directa y poder ser comparadas. Ver gráficas de la figura 22.a y 22.b.

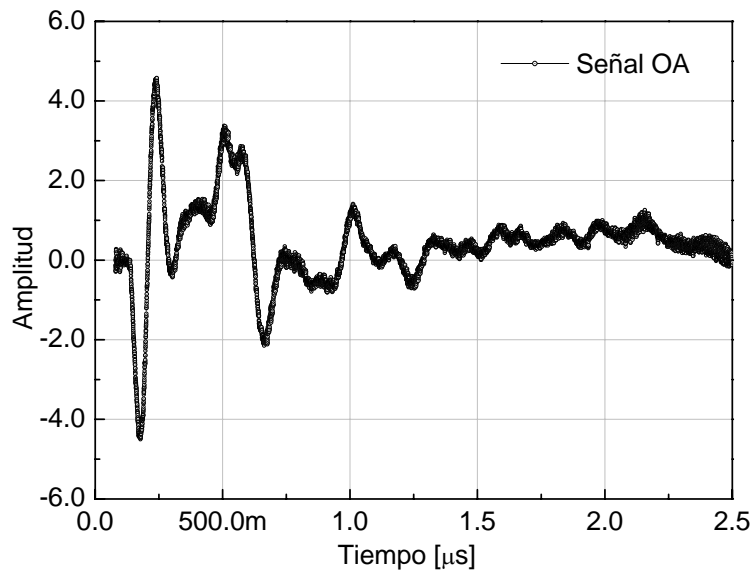


Figura 22.a) Señal opto-acústica normalizada y referenciada a cero, amplitud vs. tiempo.

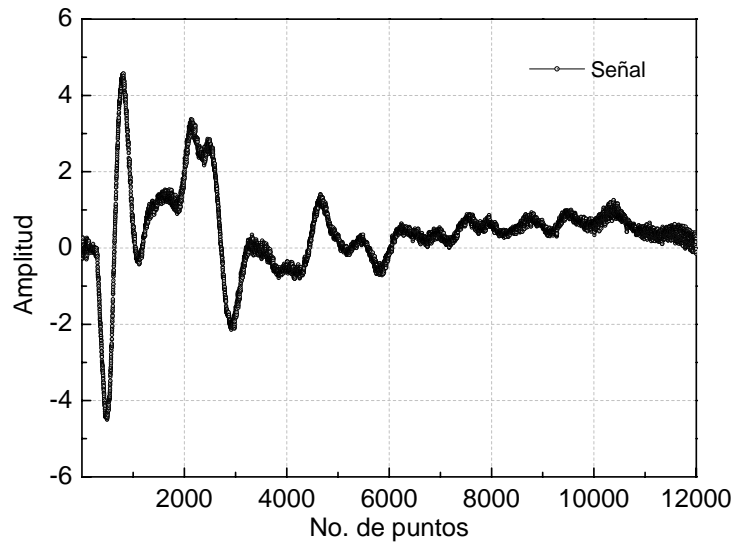


Figura 22.b) Señal opto-acústica normalizada y referenciada a cero, amplitud vs. número de puntos.

B) Aplicación del método DFA

Las señales ya acondicionadas pasan a la primera etapa de procesamiento. El programa entonces calcula el tamaño máximo y mínimo de ventana, dependiendo del orden de regresión que se elige. En la siguiente figura se reproduce la interfaz gráfica del programa, donde se realiza en la parte derecha el acondicionamiento y normalización de las señales del intervalo I y en la parte izquierda su procesamiento mediante fractales.

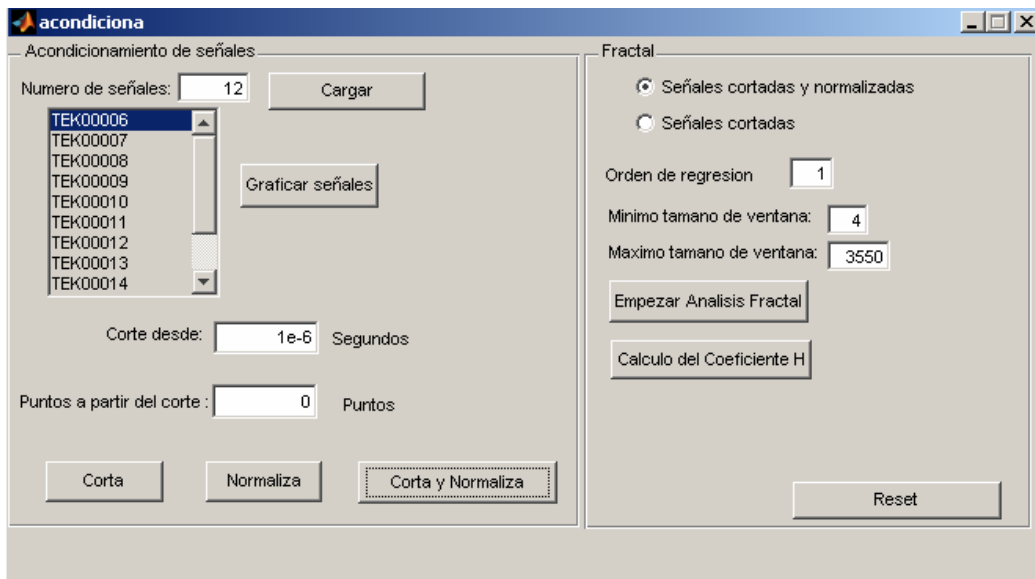


Figura 23. Interfaz gráfica del programa.

En la figura 24.a se muestra el conjunto de señales del intervalo I acondicionadas. En la figura 24.b su gráfica logarítmica que se obtiene al pulsar el botón que dice “Empezar Análisis Fractal”.

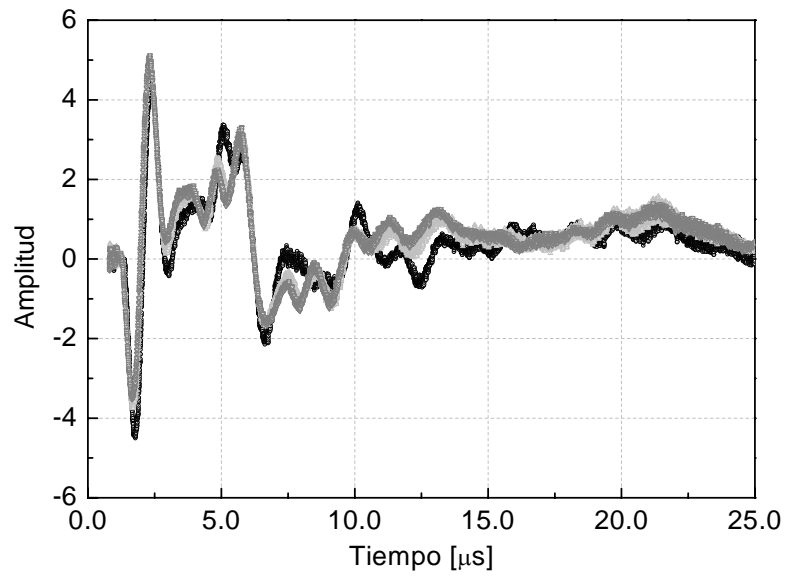


Figura 24.a) Conjunto de señales del intervalo I.

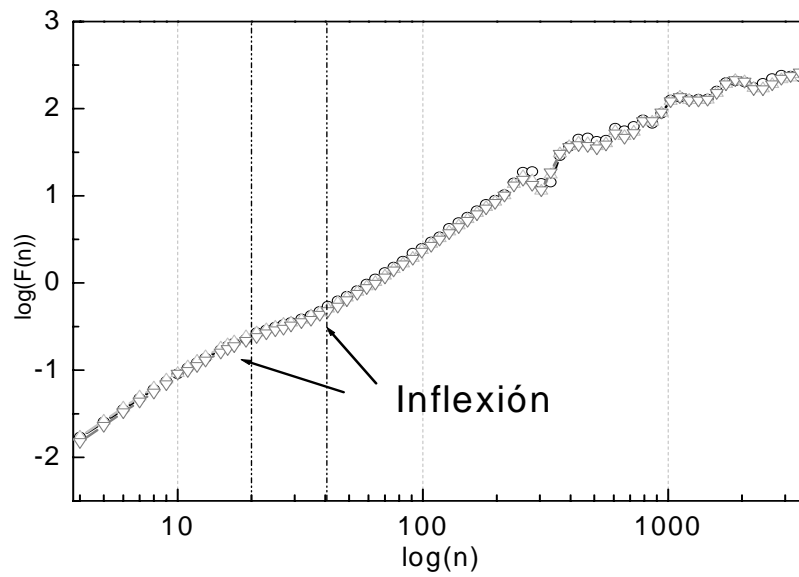


Figura 24.b) Su comportamiento (gráfica logarítmica).

Al estudiar visualmente esta gráfica se obtiene información esencial del proceso. Primeramente, se despliegan varios cambios de pendiente, luego oscilaciones, sabemos que las oscilaciones se deben al algoritmo y se presentan cuando se tienen ventanas grandes [15]. Los cambios de pendiente nos dicen que no hay un solo valor de H para todos los intervalos, esto significa que la señal exhibe de múltiples escalamientos, es decir, que puede ser de naturaleza multifractal [25]. Sin embargo, esto puede deberse también a ruido y debemos asegurarnos que los escalamientos no se deban a ruido del tipo fBm, o fGn. Para definir correctamente donde empiezan y terminan los intervalos que se observan en la gráfica doble logarítmica le aplicamos la derivada, ver figura 25.

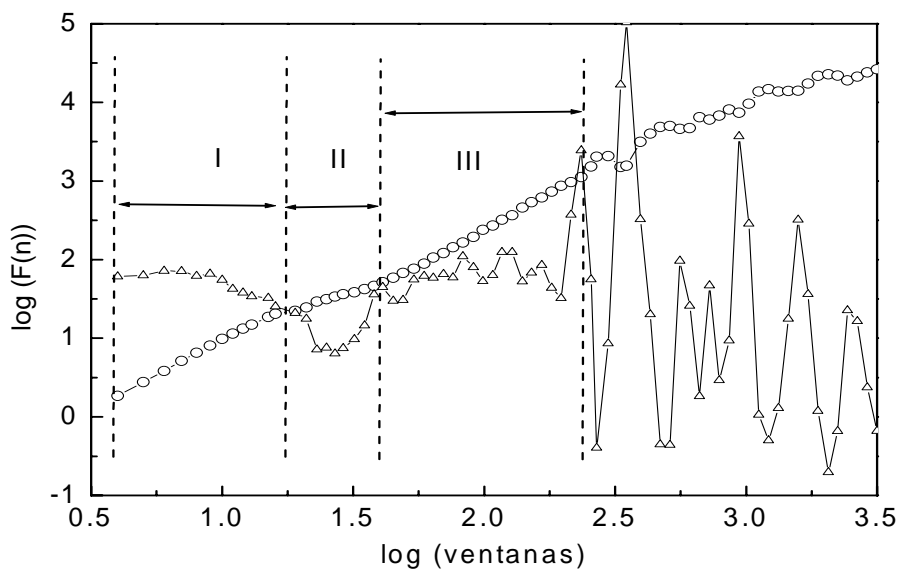


Figura 25. Gráfica logarítmica y su derivada donde se delimitan los intervalos para analizar.

Para seguir el comportamiento lineal de la gráfica doble logarítmica la aproximamos con tres rectas cuyos intervalos se extraen de su derivada. Los datos de los intervalos así obtenidos se proporcionan en la siguiente tabla.

Tabla 8. División por intervalos.

Intervalo	Inicio			Fin		
	Tamaño de ventana	En tiempo	Índice del arreglo	Tamaño de ventana	En tiempo	Índice del arreglo
I	4	6 [ns]	1	21	40 [ns]	15
II	23	44 [ns]	16	38	74 [ns]	22
III	41	80[ns]	23	197	392 [ns]	41

Una vez definidos los intervalos se procede al cálculo de H por intervalos.

El cálculo de H y su comportamiento se despliega en la figura 26.

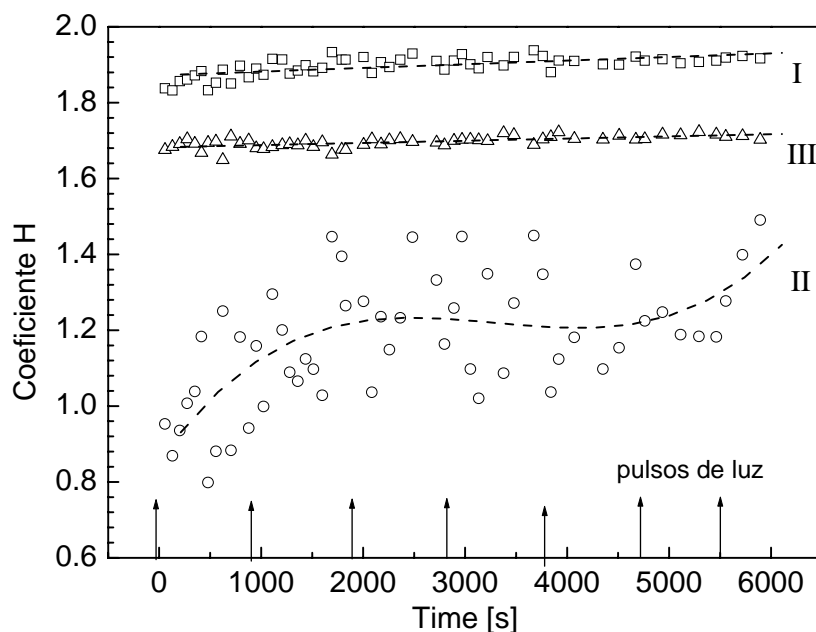


Figura 26. Coeficientes H para los tres intervalos.

Como se puede observar en la gráfica anterior ninguno de los valores de H se acercan al valor de $H = 1.5$ o $H = 0.5$ que significa ruido del tipo fBm, o fGn, por lo que estamos frente a un proceso multifractal, es decir que nuestras señales van cambiando su escalamiento conforme avanza el proceso de polimerización en dos formas casi lineales y una no lineal (intervalo II).

Para hacer más evidente este comportamiento no lineal se suaviza la señal por medio de promedios buscando que la pendiente del intervalo II se extienda hasta III. Esto se logra, para nuestro caso, haciendo una suavización por promedios de 30 puntos a las señales OA. Con estas señales suavizadas obtenemos la gráfica doble logarítmica y obtenemos el nuevo intervalo que denominaremos “intervalo extendido II”. Ver figura 27.

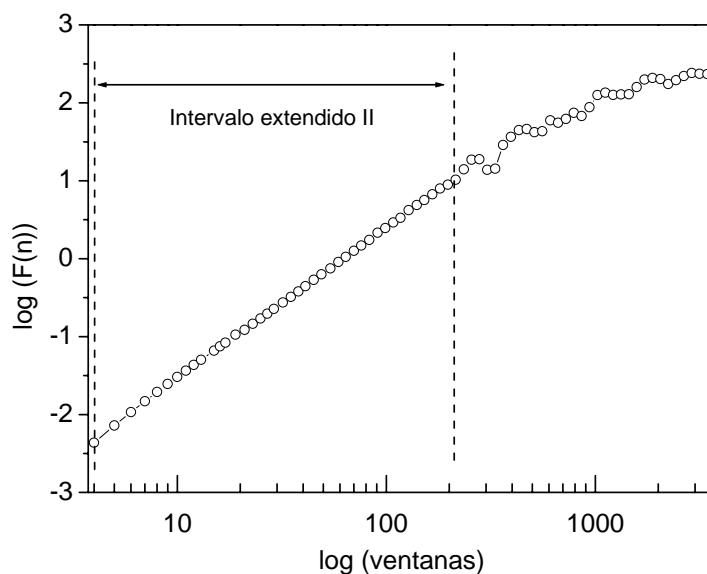


Figura 27. Comportamiento doble logarítmico del “intervalo extendido II”.

Aplicamos el mismo criterio a todas las señales OA y se calcula su coeficiente H teniéndose como resultado el comportamiento de la figura 28:

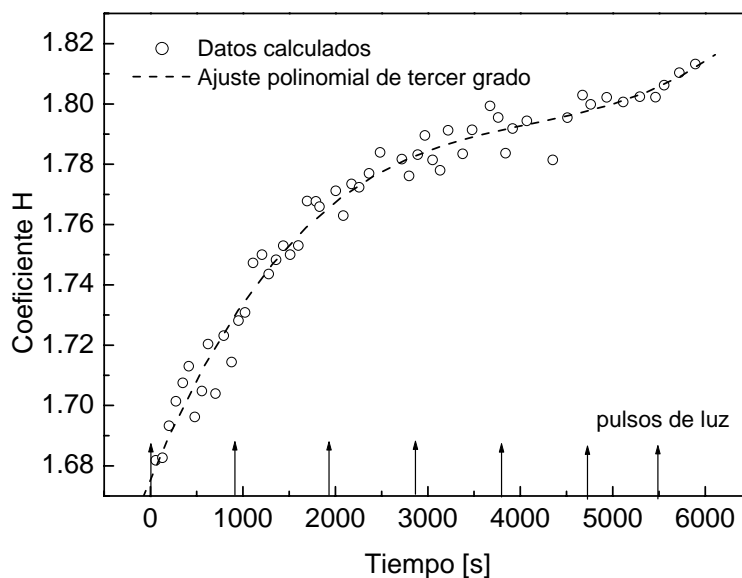


Figura 28. Dependencia del coeficiente H de microgeles sobre el tiempo de reacción t para el sistema BisGMA/TEDGMA.

IV.3. Grado de conversión de dobles enlaces vs. coeficiente de Hurst

Varios investigadores han mostrado que usando métodos de análisis fractal y modelos de agregación irreversibles se pueden clasificar las reacciones de entrecruzamiento en dos tipos: homogénea y no homogénea; la primera se caracteriza por un valor constante de coeficiente H, mientras que el proceso de reacción no homogénea presenta un crecimiento rápido de H, tal como se muestra en la figura 29. La línea curva muestra la reacción no homogénea de curado y la línea recta muestra la condición constante de $H=1.7$, la reacción homogénea

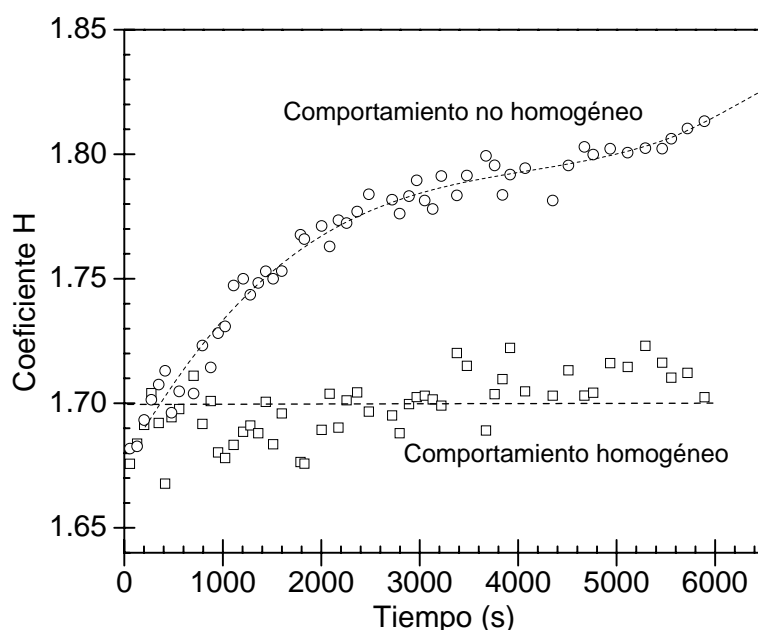


Figura 29. Dependencia del coeficiente H sobre el tiempo de reacción, t, para el sistema BisGMA/TEDGMA.

El valor constante de H nos da información del crecimiento de los microgeles que se forman cerca del punto de gelación, la formación de la red del sistema completo desde el sistema fluido pasando por el entrecruzamiento de microgeles hasta la formación del polímero elástico. El comportamiento no homogéneo, $H(t)$, nos ayuda a visualizar el efecto de auto-aceleración del proceso de curado del polímero entrecruzado.

El grado de conversión de los dobles enlaces, η , se monitorea por espectroscopía FTIR (Fourier Transform Infrared). Con este método se adquieren los espectros IR de la resina durante la foto-polimerización a iguales condiciones de experimentación que las empleadas por opto-acústica. Los valores de η se calculan encontrando la proporción entre áreas de la banda de transformación de dobles enlaces del primer espectro con el subsiguiente [41]. Siguiendo el comportamiento de los valores de η se conoce la morfogénesis que ha sufrido la resina durante su foto-polimerización.

En la figura 30 se muestra el comportamiento de η y el coeficiente H durante la reacción. La reacción de la foto-polimerización de la resina en función del tiempo se puede dividir en cuatro estados: En el estado I, η aumenta rápidamente debido al crecimiento y ramificación de las cadenas, las reacciones intermoleculares ocurren inicialmente en estado líquido hasta un grado crítico de ramificación, alrededor del 15% de conversión; En el estado II, η continua incrementándose debido al entrecruzamiento entre cadenas y el sistema se vuelve heterogéneo y viscoso debido a la formación de macrogeles; el estado III, es el estado total de gelación en donde hay un incremento ligero de η , los puntos de ramificación entre las cadenas se interconectan formándose la red tridimensional, esto sucede alrededor del 50% de conversión, formándose un material viscoelástico homogéneamente sólido; en el estado IV, el material se transforma en un polímero vitrificado, donde la viscosidad es tan alta que la movilidad molecular es muy difícil y el aumento en sus propiedades ya es muy lento, aquí pueden pasar varios meses antes de alcanzar su valor estacionario final.

La anterior morfogénesis que nos da el parámetro η que puede ser equiparado con el coeficiente H, que también está caracterizando el desarrollo de su estructura, es decir, el grado de entrecruzamiento y compactación, de hecho se pueden estudiar separadamente los fenómenos de entrecruzamiento y autoaceleración. La interpretación de estos resultados será tema de otro trabajo.

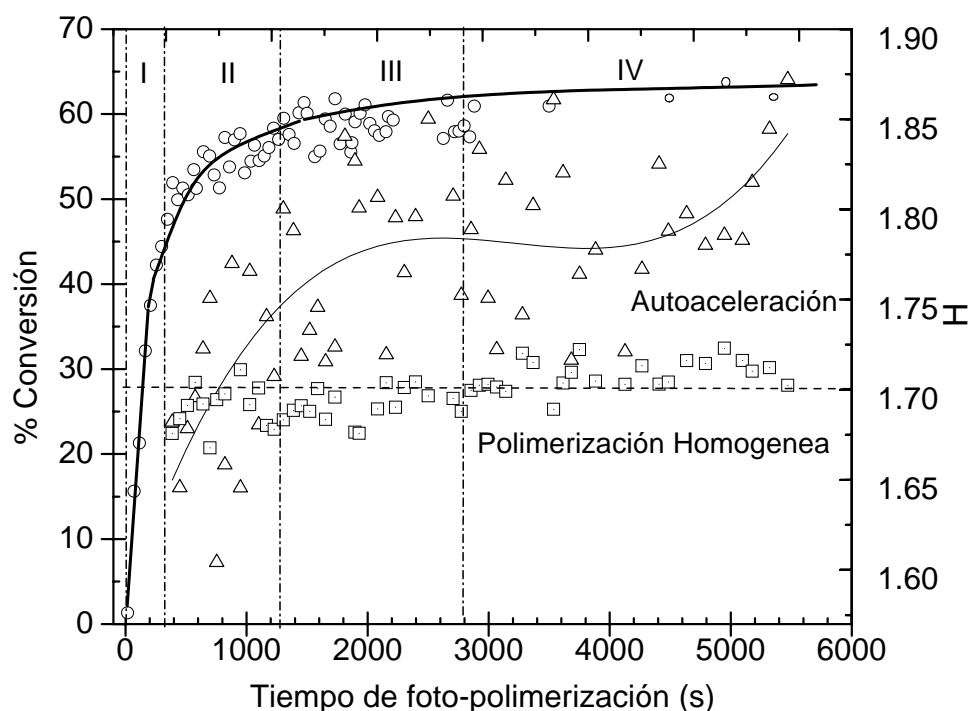


Figura 30. Variación del porcentaje de conversión de los dobles enlaces y el parámetro de auto-similitud, H calculada por el método DFA durante la foto-polimerización de (bis-GMA/TEGDMA).

CONCLUSIONES Y RECOMENDACIONES

En este trabajo se cumplen los objetivos principales propuestos que son programar y aplicar una herramienta para el análisis de señales digitales con base en fractales. En particular, se aplica a un conjunto de señales opto-acústicas que se adquieren en función del tiempo durante la reacción de foto-polimerización de la resina BisGMA/TEGDMA, usada comúnmente en las restauraciones dentales. Y en ésta se estudia la morfogénesis del proceso de entrecruzamiento y autoaceleración de la reacción bajo el punto de vista del coeficiente H.

El algoritmo se programa en Matlab, ya que por su flexibilidad permite una rápida implementación, realizar una interfaz gráfica, y un manejo de datos no definidos desde el código fuente. Además se cuenta con distintos tipos de librerías para la graficación de las señales, esto ayuda para la visualización de los resultados de una manera sencilla.

El programa desarrollado presenta una interfaz gráfica amigable para el usuario. Primeramente hace un acondicionamiento a las señales, las corta (quita puntos no deseados, a criterio del usuario), las normaliza con respecto a su desviación estándar y las despliega en gráficas. Luego, se aplica el método DFA y se despliega una gráfica doble logarítmica: F_n vs. n , en donde el usuario puede darse cuenta fácilmente si está frente a una señal que presenta similitud o multi-similitudes, distinguiendo simplemente los cambios de pendiente que se observen en ella. En el caso de presentarse multi-similitud el usuario debe decidir el intervalo adecuado de ventanas para el cálculo del coeficiente H.

Aplicación. Análisis fractal de la foto-polimerización del sistema BisGMA/TEGDMA a partir de sus señales OA.

Al introducir las señales OA al programa se observa inmediatamente, que se está frente a un proceso multifractal. Entonces, se secciona el análisis en tres intervalos. Para cada intervalo se obtienen los coeficientes H de las señales que lo conforman y se grafican. Como estos valores de H no están cerca de/o son iguales a $H=1.5$ o $H=0.5$, descartamos que tengan ruido del tipo fGn o fBm . También, debido a lo anterior concluimos que las señales presentaban dos comportamientos uno lineal y otro no y lo relacionamos con el proceso de foto-polimerización, el cual se sabe que presenta dos estados: el homogéneo y el no homogéneo.

Para observar mejor ambas tendencias lineal y no lineal, a las señales se les realiza una suavización, con ello se disminuye a dos intervalos tal como se ve en Figura 27 y se vuelven a calcular los coeficientes H (ver figura 28).

Una técnica común que nos dice como se desarrolla la morfología de la resina durante la foto-polimerización es FTIR, y nos da el porcentaje de conversión de los dobles enlaces, η . Los valores de η y los coeficientes de H durante el proceso de foto-polimerización de la resina Bis-GMA/TEGDMA son colocados en una sola figura para su comparación y análisis, esto último queda fuera del alcance de esta tesis.

Como se observa en el desarrollo de este trabajo una de las recomendaciones es realizar un seguimiento de señal en el procesamiento de señales, ya que se puede observar la salida de cada módulo y la entrada al siguiente. También son de gran ayuda el algoritmo y los diagramas de flujo, ya que por medio de estos podemos realizar una implementación del programa con mayor eficiencia.

En cuanto al método DFA la recomendación más importante es el análisis de la gráfica logarítmica, en donde se observa el comportamiento del coeficiente H , ya que aquí se podrá distinguir la señal y el ruido, o bien si es del tipo multifractal.

DFA. Detrended Fluctuation Análisis, Análisis de la de-tenencia a las fluctuaciones.

FTIR. Fourier Transform Infrared.

OA. Opto-acústicas.

H. Coeficiente de autosimilitud o de Hurst.

η . Porcentaje de polimerización.

BisGMA/TEGDMA. Resina dental, mezcla equimolar de BisGMA/TEGDMA con una relación en peso de 70/30 % respectivamente, a la cual se le adiciona 0.2% de camforquinona (CQ) y 0.3% de n, n-dimetil-p-toluidina (DMPT) como activadores.

fGn. Ruido Gaussiano Fraccionario, por sus siglas en inglés.

fBm. Movimiento Browniano Fraccionario, por sus siglas en inglés.

LD. Linear Detrended, Método de varianza aplicado al ventaneo escalado, por de-tendencia lineal.

BD. Bridge Detrended. Método de varianza aplicado al ventaneo escalado, por de-tendencia puente.

Método de Conteo de caja

La dimensión de conteo de caja se obtiene a partir de cubrir al objeto fractal con cajas de longitud δ . El número de cajas N , requeridas para cubrir el objeto que se considera fractal está relacionado con δ a través de su dimensión de conteo D_B .

Considerando un segmento de recta (objeto unidimensional) de longitud unitaria, se colocan segmentos de longitud δ para cubrir el segmento de línea recta, para lo cual se requieren de N segmentos.

El número de segmentos que requerimos para cubrir este segmento de línea unitaria es igual a $N \cdot \delta = 1$ por lo tanto:

$$N = 1 / \delta \quad (a.1)$$

Repetiendo el procedimiento anterior, pero ahora para un área unitaria es fácil ver que para cubrir tal área unitaria requeriremos de $N = 1 / \delta^2$ cuadrados de longitud δ y dimensión euclidiana mayor o igual a dos. De manera similar, si se cubre el objeto con cubos se tiene $N = 1 / \delta^3$ cubos de longitud δ , con dimensión euclidiana mayor o igual a tres para cubrir el objeto, se tiene en cada caso que el exponente δ es una medida de la dimensión del objeto. En general se requiere de:

$$N = 1 / \delta^{D_B} \quad (a.2)$$

N número de cajas para cubrir el objeto, donde el exponente D_B es la dimensión de conteo de caja del objeto, de esta forma se llega a la siguiente expresión para calcular D_B que esta definido por:

$$D_B = \frac{\log(N)}{\log(1/\delta)} \quad (a.3)$$

En general, para calcular la dimensión de conteo de caja para un objeto (es decir, longitud, área, etc.) no unitario dado por V es igual a:

$$D_B = \frac{\log(N) - \log(V)}{\log(1/\delta)} \quad (a.4)$$

Reagrupando términos:

$$\log(N) = D_B \log(1/\delta) + \log(V) \quad (a.5)$$

Se puede observar que ésta es la ecuación de una recta cuya pendiente es precisamente la dimensión de caja D_B y cuya ordenada al origen es $\log(V)$, como se muestra en la figura A1.

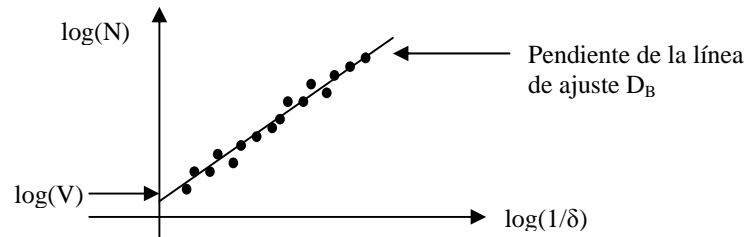


Figura. A.1. Gráfica logarítmica.

D_S y D_B son diferentes, ya que D_S requiere que las partes auto-semejantes del fractal sean identificadas plenamente, mientras que para D_B requerimos que el objeto sea cubierto con cajas auto-semejantes, por lo que D_B nos permite más flexibilidad en el tipo de objeto fractal que investiguemos [7].

Ley de potencias (Power law)

La ley de potencias es una relación entre dos cantidades escalares X y Y, que se describe como:

$$y = ax^k \quad (\text{a.7})$$

Donde a (constante de proporcionalidad) y k (el exponente de la ley de potencias) son constantes. Se puede considerar como línea recta en una gráfica log-log ya que tomando los logaritmos de ambos lados, la ecuación es igual a:

$$\log(y) = k \log(x) + \log(a) \quad (\text{a.8})$$

Que es igual a la ecuación de una línea

$$y' = mx' + c \quad (\text{a.9})$$

Están entre las leyes más frecuentes del escalamiento que describen la invariancia de escala encontrada en muchos fenómenos naturales.

Código fuente del programa.

El programa se elaboró utilizando el software de desarrollo MatLab, el cual se llama acondiciona.m, y tiene la interfaz gráfica que se muestra en la Figura 18, que de igual manera se realizó en MatLab, el código fuente es el siguiente:

```
function varargout = acondiciona(varargin)

gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
                  'gui_Singleton', gui_Singleton, ...
                  'gui_OpeningFcn', @acondiciona_OpeningFcn, ...
                  'gui_OutputFcn', @acondiciona_OutputFcn, ...
                  'gui_LayoutFcn', [], ...
                  'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% _____

function acondiciona_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;

% valores iniciales
handles.fin=3000;
handles.inic=0;
handles.escoje=0;
handles.regfrac=1;
handles.minfrac=2*(handles.regfrac+1);

% Salvando
guidata(hObject, handles);
```

```
%  
  
function varargout = acondiciona_OutputFcn(hObject, eventdata, handles)  
varargout{1} = handles.output;  
  
function number_CreateFcn(hObject, eventdata, handles)  
usewhitebg = 1;  
if usewhitebg  
    set(hObject,'BackgroundColor','white');  
else  
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));  
end  
  
%  
  
function number_Callback(hObject, eventdata, handles)  
num = str2double(get(hObject, 'String'));  
num=round(num);  
set(hObject, 'String', num);  
if isnan(num)  
    set(hObject, 'String', 0);  
    error('Debe ser un numero','Error');  
end  
% Salvando  
handles.num = num;  
guidata(hObject,handles)  
  
%  
function load_Callback(hObject, eventdata, handles)  
% % % % % %  
todos="";  
for i=1:handles.num  
    clear filename;  
    clear pathname;  
    [filename,pathname]=uigetfile({'*.DAT'; '*.dat'}, 'ARCHIVOS A CARGAR');  
    % para abrir ventana de carga y conseguir path y nombre del archivo  
  
    if isequal(filename,0)|isequal(pathname,0)  
        %si se pone cancelar o no se encuentra el archivo obtenemos mensaje de error  
        disp('No se encontro el archivo')%mensaje  
        break;  
    else  
        %solocortada lo cargamos  
        longfilename=strcat(pathname,filename); %encontramos toda la ruta al archivo
```

```

load(longfilename);%cargamos la variable en workspace con el nombre del archivo

for j=1:length(filename) %quitamos extension del archivo
    if filename(1,j)=='.' %encontramos el .
        break;%salimos del ciclo
    end
    fil(1,j)=filename(1,j); %fil es igual al nombre del archivo sin extension
end %termina quitado de extension
t=['sign0' num2str(i)];
eval(['handles.' fil '=eval(fil);'])
han='handles.';
nam=strcat(han,fil);
eval(['handles.' t '=nam;'])% guardamos nombre del archivo en un arreglo de letras
todos=strvcat(todos,fil);
set(handles.lista,'String',todos)
end
%disp('fil')%mensaje
end
% Salvando
guidata(hObject,handles)

%
function lista_CreateFcn(hObject, eventdata, handles)
gris=0;
if gris
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end
function lista_Callback(hObject, eventdata, handles)

%
function graficar_Callback(hObject, eventdata, handles)
for i=1:handles.num
    [y,x]=plotterall((eval(eval(['handles.sign0' num2str(i)]))));
    fly=['y0' num2str(i)];
    eval(['sig0' num2str(i) '= fly;']);
    eval(['y0' num2str(i) '=y;']);
    m1(:,i)=eval(eval(['sig0' num2str(i)]));
end
figure(2); plot(x,m1);
set(2,'name','Todas VS tiempo');xlabel('Timpo');ylabel('Amplitud');
figure(3); plot(m1);
set(3,'name','Todas VS No. de puntos');xlabel('No. de puntos');ylabel('Amplitud');
guidata(hObject,handles)

%

```

```
function empiezacorte_CreateFcn(hObject, eventdata, handles)
usewhitebg = 1;
if usewhitebg
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end
```

%

```
function empiezacorte_Callback(hObject, eventdata, handles)
inic = str2double(get(hObject, 'String'));
if isnan(inic)
    set(hObject, 'String', 0);
    inic=0;
    errordlg('Debe ser un numero','Error');
end
% Salvando
handles.inic = inic;
guidata(hObject,handles)
```

%

```
function fincorte_CreateFcn(hObject, eventdata, handles)
usewhitebg = 1;
if usewhitebg
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end
```

%

```
function fincorte_Callback(hObject, eventdata, handles)
fin = str2double(get(hObject, 'String'));
fin=round(fin);
set(hObject, 'String',fin);
if isnan(fin)
    set(hObject, 'String', 3000);
    fin=3000;
    errordlg('Debe ser un numero','Error');
end
% Salvando
handles.fin = fin;
guidata(hObject,handles)
```

```

%
function cortesisg_Callback(hObject, eventdata, handles)
if handles.fin==0
    handles.fin=length(eval(eval(['handles.sign01'])));
end
for i=1:handles.num
    [y,x]=ord((eval(eval(['handles.sign0' num2str(i)]))),handles.inic,handles.fin);
    fly=['y0' num2str(i)];
    eval(['sig0' num2str(i) '= fly;']);
    eval(['y0' num2str(i) '= y;']);
    m1(:,i)=eval(eval(['sig0' num2str(i)]));
end
figure(4); plot(x,m1);
set(4,'name','Cortadas VS Tiempo');xlabel('Tiempo');ylabel('Amplitud');
figure(5); plot(m1);
set(5,'name','Cortadas VS No. de Puntos');xlabel('No. de puntos');ylabel('Amplitud');
%b.-numero de señales
%a.-longitud de las señales
[a,b]=size(m1);
handles.maxfrac=floor(1/4*a);
set(handles.maxcaja,'String',handles.maxfrac);
%Salvando en el directorio
hgsave(4,'signcortadas.fig');
save -ascii -double mc.dat m1; %guarda matriz y's
save -ascii -double time.dat x;%timepo por separado
handles.m1=m1;
handles.time=x;
%salvando
guidata(hObject,handles)

%
function normalizar_Callback(hObject, eventdata, handles)
m2=normalizatodo(handles.time,handles.m1);
figure(6); plot(handles.time,m2);
set(6,'name','Normalizadas VS Tiempo');xlabel('Tiempo');ylabel('Amplitud');
figure(7); plot(m2);
set(7,'name','Normalizadas VS No. de Puntos');xlabel('No. de Puntos');
ylabel('Amplitud');
handles.m2=m2;
%Salvando
guidata(hObject,handles)
save -ascii -double mn.DAT m2; %guarda matriz y's

%

```

```

function cortanormaliza_Callback(hObject, eventdata, handles)
if handles.fin==0
    handles.fin=length(eval(eval(['handles.sign01'])));
end
for i=1:handles.num
    [y,x]=ord((eval(eval(['handles.sign0' num2str(i)]))),handles.inic,handles.fin);
    fly=['y0' num2str(i)];
    eval(['sig0' num2str(i) '= fly;']);
    eval(['y0' num2str(i) '= y;']);
    m1(:,i)=eval(eval(['sig0' num2str(i)]));
end

m2=normalizatodo(x,m1);

[a,b]=size(m1);
handles.maxfrac=floor(1/4*a);
set(handles.maxcaja,'String',handles.maxfrac);
figure(6); plot(x,m2);
set(6,'name','Normalizadas VS Tiempo');xlabel('Tiempo');ylabel('Amplitud');
figure(7); plot(m2);
set(7,'name','Normalizadas VS No. de Puntos');xlabel('No. de Puntos');
ylabel('Amplitud');
handles.m1=m1;
handles.m2=m2;
handles.time=x;
hgsave(6,'signalsnorm_1.fig');
save -ascii -double time.dat x;%timepo por separado
save -ascii -double mn.dat m2; %guarda matriz y's
% Salvando
guidata(hObject,handles)

```

% _____

```

function unitgroup_SelectionChangeFcn(hObject, eventdata, handles)

if (hObject == handles.matrizcortynorm)
    handles.escoje=0;
else
    handles.escoje=1;
end
guidata(hObject,handles);

```

% _____

```

function regresion_CreateFcn(hObject, eventdata, handles)
usewhitebg = 1;
if usewhitebg

```

```
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end
```

```
% _____
```

```
function regresion_Callback(hObject, eventdata, handles)
regfrac = str2double(get(hObject, 'String'));
regfrac=round(regfrac);
set(hObject,'String',regfrac);
if isnan(regfrac)
    regfrac=1;
    set(hObject, 'String', regfrac);
    errordlg('Debe ser un numero','Error');
end
% salvando
handles.regfrac = regfrac;
handles.minfrac=2*(handles.regfrac+1);
set(handles.mincaja,'String',handles.minfrac);
guidata(hObject,handles)
```

```
% _____
```

```
function mincaja_CreateFcn(hObject, eventdata, handles)
usewhitebg = 1;
if usewhitebg
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end
```

```
% _____
```

```
function mincaja_Callback(hObject, eventdata, handles)
minfrac = str2double(get(hObject, 'String'));
minfrac=round(minfrac);
set(hObject,'String',minfrac);
if isnan(minfrac)
    minfrac=2*(handles.regfrac+1);
    set(hObject, 'String', minfrac);
    errordlg('Debe ser un numero','Error');
end
if minfrac<(2*(handles.regfrac+1))
    minfrac=2*(handles.regfrac+1);
    set(hObject, 'String', minfrac);
    errordlg('El menor es 2*(orden de regresion +1)','Error');
```

```
end
% Salvando
handles.minfrac = minfrac;
guidata(hObject,handles)

%_____

function maxcaja_CreateFcn(hObject, eventdata, handles)
usewhitebg = 1;
if usewhitebg
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

%_____

function maxcaja_Callback(hObject, eventdata, handles)
maxfrac = str2double(get(hObject, 'String'));
maxfrac=round(maxfrac);
set(hObject,'String',maxfrac);
[a,b]=size(handles.m1);
if isnan(maxfrac)
    maxfrac=floor(1/4*a);
    set(hObject, 'String', maxfrac);
    errordlg('Debe ser un numero','Error');
end
if maxfrac>(floor(1/4*a))
    maxfrac=floor(1/4*a);
    set(hObject, 'String', maxfrac);
    errordlg('El mayor es 1/4 de la longitud de la serie','Error');
end
% Salvando
handles.maxfrac = maxfrac;
guidata(hObject,handles)

%_____

function empanfractal_Callback(hObject, eventdata, handles)
[a,b]=size(handles.m1);
if handles.escoje==0

[logcajas,logss]=dfa_matlab(handles.m2,a,b,handles.regfrac,handles.minfrac,handles.maxfrac,1);

else
```

```
[logcajas,logss]=dfa_matlab(handles.m2,a,b,handles.regfrac,handles.minfrac,handles.maxfrac,0);
```

```
end
```

```
cajas=10.^(logcajas);  
ss=10.^(logss);  
figure(8); loglog(cajas,ss,');  
set(8,'name','Comportamiento Fractal');xlabel('Log(numero de ventana)');  
ylabel('Log(F(n))');  
hgsave(8,'fractal_1.fig');  
save -ascii -double mat_fractal.dat ss;  
save -ascii -double cajas.dat cajas;  
handles.logcajas=logcajas;  
handles.logss1=logss;  
guidata(hObject,handles)
```

```
%
```

```
function sacardimension_Callback(hObject, eventdata, handles)  
d1=obtienedimension(handles.logcajas,handles.logss1);  
figure(9); plot(d1(:,1),d1(:,2),'o');  
save -ascii -double muestras_vs_dimensiones.dat d1;  
hgsave(9,'muestra_vs_dimensiones.fig');  
set(9,'name','Dimensiones');xlabel('No. de muestra');ylabel('Dimension');  
guidata(hObject,handles)
```

```
%
```

RESET

```
function reset_Callback(hObject, eventdata, handles)  
close (acondiciona);  
acondiciona;
```

```
%
```

CORTAR

```
function [ynueva,x]=ord(s,p,d);  
%corta desde p segundos hasta d puntos a partir del corte  
j=length(s); % numero de muestras (tamaño de la señal)  
num=round((p-s(1,1))./abs(s(2,1)-s(1,1)));  
num=num+1;  
f=d+num-1;  
if d==j  
    f=j;  
end  
x=s((num:f),1);  
ynueva=s((num:f),2);
```

```
% _____GRAFICAR_____

function[ynueva,x]=plotterall(s);
j=length(s); %numero de muestras (tamaño de la señal)
x=s((1:j),1);
ynueva=s((1:j),2);

% _____NORMALIZA_____

function[RR]=normalizatodo(time,signs);
x=time;
[a,b]=size(signs);
for i=1:b
    sdate=(signs(:,i)-mean(signs(:,i)))/std(signs(:,i));
    p=polyfit(x,sdate,1);
    reg1=polyval(p,x);
    nuev=sdate-reg1;
    n1=nuev(1,1);
    RR(:,i)=nuev-n1;
end

% _____COEF_____

function[dimensiones]=obtienedimension(xs,ys);
[a,b]=size(ys);
for i=1:b
    v=ys(:,i);
    d=polyfit(xs,v,1);
    dimensiones(i,2)=d(1,1);
    dimensiones(i,1)=i;
end

% _____DFA_____

function[Mx,My]=dfa_matlab(signs,a,b,nfit,minbox,maxbox,sw);

%por si esta al revés maxbox y minbox
if minbox>maxbox
    c=minbox;
    minbox=maxbox;
    maxbox=c;
    if minbox<2*(nfit+1)
        minbox=2*(nfit+1);
    end
end
```

```

end

boxratio=2^(1/8); %razon entre cada caja para que no se junten en la escala logaritmica
rslen = floor(log10(maxbox/minbox)/log10(boxratio)+1.5);% numero de cajas

rw=minbox;
rs(1)=rw;
ir=1;
n=2;
%calculo de los tamaños de las cajas
while and(n<=rslen,rw<=maxbox)==1,
    rw=floor(minbox*(boxratio^ir)+0.5);
    ir=ir+1;
    if (rw>rs(n-1))
        if (rw<=maxbox)
            rs(n)=rw;
            n=n+1;
        end
    end
end
end
n=n-1;

x=1:rs(n);%ordenada para cada tamaño de cajas
x=transpose(x);
%barra de espera
barfijo=b*n;
countbar=0;
winbar = waitbar(0,'Espere por favor...');
%% %
for i=1:b %desde la señal "1" hasta "b" el numero de señales

    v=signs(:,i); %v señal que se va a analizar
    y=cumsum(v); %integracion de la señal
    %DFA
    for j=1:n
        %desde la caja (minbox) "rs[1]", hasta la caja (aproximadamente maxbox)"rs[n]"
        boxsize=rs(j); %tamaño de caja a analizar
        %ventana deslizable (sw=0); no deslizable (sw=1) traslape o no traslape
        if sw==0
            inc=1; %incremento de los puntos en la señal para el barrido de cajas
            stat=(floor(a-boxsize+1))*boxsize;
        else
            inc=boxsize;%incremento de los puntos en la señal para el barrido de cajas
            stat=(floor(a/boxsize))*boxsize;
        end
        k=boxsize;
        sali(j)=0;
    end
end

```



```
p=0;
while k<=a%desde el tamaño de la caja "k" hasta el tamaño de la muestra "a"
    p=0;
    pol=polyfit(x(1:boxsize),y((k-boxsize+1):k),nfit);
    %obtenemos el polinomio de la caja
    yn=polyval(pol,x(1:boxsize));%valuamos el polinomio de la caja
    for r=1:boxsize
        p=p+(y(r+(k-boxsize))-yn(r)).^2;
        %calculo de los errores para obtener F(n)
    end
    k=k+inc;
    sali(j)=sali(j)+p;
end
sal(j)=sali(j)./stat;
f(j,i+1)=log10(sal(j))/2; %logaritmo base 10 de F(n)
f(j,1)=log10(boxsize); %logaritmo base 10 de las cajas
Mx(j,1)=f(j,1); %logaritmo de cajas
My(j,i)=f(j,i+1);%logaritmo de F(n)
rs(j);%numero de caja a analizar
%%%
countbar=countbar+1;
waitbar(countbar/barfijo,winbar)
%%%
end
i;%numero de señal analizada
end
close(winbar);%%%
%
```

- [1] Nigel Lesmoir-Gordon, Will Rood and Ralph Edney, *Introducing, Fractal Geometry*, Totem Books, 2001.
- [2] Mandelbrot, Benoit, *La Geometría fractal de la naturaleza*, t. de Josep Llosa, Barcelona, Tusquets, 1997.
- [3] Feder J. *Fractals*. New York: Plenum Press, 1988.
- [4] Peitgen, Heinz-Otto, *Introduction to fractals and chaos*, Springer, 1992.
- [5] A. W. Tucker y H. S. Bailey, Jr., Topología, en *Matemáticas en el Mundo Moderno, Selecciones de Scientific American* Editorial Blume, Madrid y Barcelona, 1974.
- [6] Jiménez López, José, *Dimensión fractal y límite de predictibilidad de tres variables climáticas; Tlaxcala capital*, México, 1998.
- [7] Matías Hernández, Sergio Alejandro, *Aproximación fractal de perturbaciones atmosféricas sobre un frente de onda plano*, México UNAM 1998.
- [8] Jiménez Figueroa, Sergio, *Re-distribución fractal en el Centro Histórico de la, México*, UNAM 1998.
- [9] Edgar, Gerald A., *Measure, topology and fractal geometry*, New york : Springer, 1990.
- [10] Falconer K., *Fractal Geometry: Mathematical Foundations and Applications*, Chichester : J. Wiley, 1990.
- [11] Falconer K., *Techniques in Fractal Geometry*, John Wiley Sons, Inc.; 1997.
- [12] Tricot C. In: Belair J, *Fractal Geometry and Analysis*, Kluwer; 1989.
- [13] Zhi Chen, Plamen Ch. Ivanov, Kun Hu, H. Eugene Stanley¹ *Effect of nonstationarities on detrended fluctuation analysis*, Physical Review E., 2002
- [14] Peng C-K, Hausdorff JM, Goldberger AL. *Fractal mechanisms in neural control: Human heartbeat and gait dynamics in health and disease*. In: Walleczek J, ed. *Nonlinear Dynamics, Self-Organization, and Biomedicine*. Cambridge: Cambridge University Press, 1999.

- [15] Michael J. Cannon, Donald B. Percival, David C. Caccia , *Evaluating scaled windowed variance methods for estimating the Hurst coefficient of time series*, Physica A, 1997.
- [16] Lindstrom, Tom, *Brownian motion on nested fractals*, Providence, Rhode Island, American mathematical society, 1990.
- [17] G. Carolan-Rees, A.C. Tweddel, K.K. Naka, T.M. Griffith, *Fractal dimensions of laser doppler flowmetry time series*, Medical Engineering & Physics, 2001
- [18] R.R. Nigmatullin, S.I. Osokin, *Signal processing and recognition of true kinetic equations containing non-integer derivatives from raw dielectric data*, Signal Processing, 2002.
- [19] Peng C-K, Buldyrev S.V., Havlin S., Simons N., Stanley H. E., and Golberger A.L., *Mosaic organization of DNA nucleotides*, Physical Review E, February 1994.
- [20] T.F. Xie, S.P. Zhou, *On a class of fractal functions with graph Box dimension 2*, Chaos, Solitons and Fractals, 2003.
- [21] Montroll EW, Shlesinger MF. *The wonderful world of random walks*. In: *Nonequilibrium Phenomena II. From Stochastics to Hydrodynamics*. Lebowitz L, Montroll EW, Amsterdam: North-Holland, 1984.
- [22] Pengjian Shang, Santi Kamae, *Fractal nature of time series in the sediment transport phenomenon*, Chaos, Solitons and Fractals, 2005
- [23] Kun Hu, Plamen Ch. Ivanov, Zhi Chen, Pedro Carpena, and H. Eugene Stanley, *Effect of trends on detrended fluctuation analysis*, Physical Review E, 2001.
- [24] Peng C-K, Hausdorff JM, Havlin S, et al. *Multiple-time scale analysis of physiological time series under neural control*. Physica A. 1998.
- [25] Jan W. Kantelhardt, Stephan A. Zschiegner, et. all., *Multifractal detrended fluctuation analysis of nonstationary time series*, Physica A, 2002
- [26] C.V. Chianca, A. Ticona, T.J.P. Penna, *Fourier-detrended fluctuation analysis*, Physica A, 2005
- [27] Devaney, Robert, *Chaos, fractals and dynamics : Computer experiments in mathematics*, New York, 1989
- [28] Peitgen, Heinz-Otto, *Chaos and fractals : new frontiers of science*, Springer, 1992.
- [29] Stevens, Roger T., *Fractal programming and ray tracing with c++*, California, M & T, 1990.

- [30] Stevens, Roger T., *Advanced fractal programming in c*, California, M & T, 1990.
- [31] Martin Kappes, Frank Nießner, *Succinct representations of languages by DFA with different levels of reliability*, Theoretical Computer Science, 2004
- [32] B. Ravikumar, G. Eximan, *Weakminimization of DFA—an algorithm and applications*, Theoretical Computer Science, 2004
- [33] Radhakrishnan Nagarajan, *Effect of coarse-graining on detrended fluctuation analysis*, Physica A, 2005.
- [34] Radhakrishnan Nagarajan, Rajesh G. Kavasseri, *Minimizing the effect of trends on detrended fluctuation analysis of long-range correlated noise*, Physica A, 2005.
- [35] J. Planinic, B. Vukovic, V. Radolic, *Radon time variations and deterministic chaos*, Journal of Environmental Radioactivity, 2003
- [36] José Álvarez-Ramírez, Eduardo Rodríguez, Ilse Cervantes, Juan Carlos Echeverria, *Scaling properties of image textures: A detrending fluctuation analysis approach*, Physica A, 2005.
- [37] P.Norouzzadeh, G.R. Jafari, *Application of multifractal measures to Tehran price index*, Physica A, 2005.
- [38] José Álvarez-Ramírez, Eduardo Rodríguez, Juan Carlos Echeverria, *Detrending fluctuation analysis based on moving average filtering*, Physica A, 2005.
- [39] Pilar Grau-Carles, *Bootstrap testing for detrended fluctuation analysis*, Physica A, 2005.
- [40] José Álvarez-Ramírez, Gilberto Espinosa-Paredes, Alejandro Vázquez, *Detrended fluctuation analysis of the neutronic power from a nuclear reactor*, Physica A, 2005.
- [41] M. Navarrete, F. Rivera Torres, R. Vera Graziano, M. Villagrán Muniz, *Evolución del módulo elástico en el proceso de entrecruzamiento de Bis-gma/tegdma durante la fotopolimerización mediante fotoacústica*, SOMI XVIII.

