

Tesis

Estudio del Tráfico Autosimilar Orientado a la Simulación Mediante la Utilización de Wavelets u Onditas

Tesista: Reinaldo Jose Ramon Scappini
Director de Tesis: Luis Armando Marrone – UNLP

Trabajo presentado para obtener el grado de
Magíster en Redes de Datos

Facultad de Informática
Universidad Nacional de la Plata

Agosto 2010



Agradecimientos

Dedico este trabajo

Con profundo agradecimiento:

A la memoria de mi padre y mi madre, a quienes debo todo lo que soy; a mi esposa Miryam, mis dos hijos Antonio y Adolfo, quienes me soportan día a día; a mis hermanas Carmen y Rositín, por su apoyo.

A mi director de tesis Luis, por dispensarme generosamente su tiempo y guía, con precisas y oportunas observaciones.

A mis entrañables compañeros del grupo “Grant’s”, Ricardo, Laura, Roberto, Ernesto, Juan, Osvaldo, Martín, Daniel, Miguel Marcelo y Adriana. Estimados Profesores Lía, Jorge, Paula y Javier; además de una larga lista del personal administrativo de la UNLP, quienes sin ninguna duda, enriquecieron la agradable experiencia de este postgrado.

Índice de Contenidos

AGRADECIMIENTOS	I
ÍNDICE DE CONTENIDOS	III
NOMENCLATURA Y SÍMBOLOS UTILIZADOS EN ESTE TRABAJO	V
1 MOTIVACIÓN DE ESTE TRABAJO	- 1 -
1.1 EL CONTEXTO	- 1 -
1.2 EL PLANTEO	- 2 -
1.3 ¿POR QUÉ UTILIZAR LAS WAVELETS?	- 4 -
1.4 ¿CUALES SON LOS OBJETIVOS DE ESTA TESIS?	- 4 -
1.5 ¿POR QUE MATLAB®?	- 4 -
2 ESTADO DEL ARTE EN EL ESTUDIO DEL TRÁFICO AUTOSIMILAR EN LAS REDES TELEMÁTICAS	- 7 -
2.1 PUNTO DE PARTIDA	- 7 -
2.2 LA UBICUIDAD DE LA AUTOSIMILITUD	- 9 -
2.3 LA AUTOSIMILITUD EN LA ACTUALIDAD	- 10 -
2.4 COROLARIO	- 10 -
3 LOS PROCESOS AUTOSIMILARES Y EL TRÁFICO – MARCO TEÓRICO	- 13 -
3.1 TRÁFICO DE UNA RED	- 13 -
3.2 EL MODELO GENERAL	- 13 -
3.3 LOS PROCESOS Y EL TRÁFICO - PROPIEDADES Y DEFINICIONES	- 14 -
3.3.1 <i>Procesos estacionarios</i>	- 14 -
3.3.2 <i>Procesos débilmente estacionarios</i>	- 14 -
3.3.3 <i>La Función de Autocorrelación</i>	- 14 -
3.3.4 <i>Autocovarianza</i>	- 15 -
3.4 EFECTO MEMORIA - DEPENDENCIAS DE RANGO CORTO Y RANGO LARGO	- 15 -
3.5 AUTOSIMILITUD	- 17 -
3.5.1 <i>Autosimilitud exacta o determinista</i>	- 18 -
3.5.2 <i>Autosimilitud estocástica o estadística</i>	- 18 -
3.5.2.1 Definición de Autosimilitud Estadística de Tiempo Continuo	- 18 -
3.5.2.2 Definición Autosimilitud Estadística de tiempo discreto	- 18 -
3.5.2.2.1 Autosimilitud exacta de segundo orden	- 19 -
3.5.2.2.2 Autosimilitud asintótica de segundo orden	- 20 -
3.6 COEFICIENTE DE CORRELACIÓN DE UN PROCESO ESTOCÁSTICO	- 20 -
3.6.1 <i>Relación entre Correlación, Autosimilitud y la Dependencia de Rango Largo</i>	- 21 -
3.7 DENSIDAD ESPECTRAL	- 21 -
3.7.1 <i>Relación entre la Densidad Espectral y Dependencia de Rango Largo</i>	- 23 -
3.8 DISTRIBUCIONES DE COLA HIPERBÓLICA	- 23 -
3.9 RELACIÓN ENTRE LA AUTOSIMILITUD, LA DEPENDENCIA DE RANGO LARGO Y LA DISTRIBUCIÓN DE COLAS HIPERBÓLICAS	- 24 -
3.10 MANIFESTACIÓN DE LA AUTOSIMILITUD EN LOS PROCESOS ESTOCÁSTICOS	- 25 -
3.11 PROCESOS AUTOSIMILARES UTILIZADOS PARA MODELAR TRÁFICO	- 25 -
3.11.1 <i>Procesos de Incrementos Independientes</i>	- 25 -
3.11.2 <i>Proceso de Movimiento Browniano</i>	- 26 -
3.11.2.1 Análisis de la Autosimilitud del Movimiento Browniano	- 26 -
3.11.3 <i>Proceso de Movimiento Browniano Fraccionario</i>	- 28 -
3.11.3.1 Análisis de la Autosimilitud del Movimiento Browniano Fraccionario	- 28 -
3.11.4 <i>Ruido Gaussiano Fraccional</i>	- 30 -
3.11.4.1 Análisis de la Autocorrelación del Ruido Gaussiano Fraccional	- 30 -
3.11.5 <i>Ventajas del FGN como modelo de tráfico</i>	- 31 -
4 WAVELETS U ONDITAS – ANÁLISIS MULTI-RESOLUCIÓN	- 33 -
4.1 ANÁLISIS CON TRANSFORMADAS	- 33 -
4.2 ANÁLISIS DE FOURIER O ANÁLISIS ARMÓNICO	- 33 -
4.2.1 <i>Forma Compleja de la Transformada de Fourier</i>	- 35 -
4.2.2 <i>Transformada Inversa de Fourier</i>	- 35 -
4.2.3 <i>Limitaciones del Análisis de Fourier</i>	- 37 -
4.3 TRANSFORMADA DE FOURIER DE TIEMPO CORTO O TRANSFORMADA DE GABOR	- 37 -
4.3.1 <i>Interpretación de la Transformada de Gabor</i>	- 38 -

4.3.2 Ejemplos de análisis de Gabor	- 40 -
4.3.2.1 Análisis Gabor Caso 1	- 40 -
4.3.2.2 Análisis Gabor Caso 2	- 43 -
4.4 EL ANÁLISIS CON WAVELETS U ONDITAS	- 46 -
4.4.1 ¿Que es una Wavelet?.....	- 47 -
4.4.2 La Transformada Wavelet Continua o CWT.....	- 48 -
4.5 ANÁLISIS MULTI-RESOLUCIÓN.....	- 49 -
4.5.1 Transformada Wavelet Discreta DWT.....	- 52 -
4.5.1.1 Calculo de la Transformada Wavelet Discreta	- 53 -
4.6 ANÁLISIS MULTI RESOLUCIÓN DE PROCESOS AUTOSIMILARES	- 54 -
4.7 RELACIÓN ENTRE LAS WAVELETS Y LOS PROCESOS HSS, HSSI Y LRD	- 56 -
5 ESTUDIANDO EL TRÁFICO CON MATLAB®	- 59 -
5.1 MÉTODOS CUALITATIVOS	- 60 -
5.1.1 Escalamiento Temporal	- 60 -
5.1.1.1 Script serieTescalada.m	- 61 -
5.1.1.2 Script comparaGrafica.m	- 64 -
5.1.2 Grafico Rango Reescalado R/S - El efecto Hurst	- 65 -
5.1.2.1 Script rangoRescalado.m	- 66 -
5.1.2.2 Script rangoRescaladocomparado.m	- 68 -
5.1.2.3 Script rangoRescaladopromedio.m	- 68 -
5.1.3 Gráfico Varianza Tiempo	- 69 -
5.1.3.1 Script varianzaTiempo.m	- 70 -
5.1.3.2 Script varianzaTiempocomparado.m	- 72 -
5.2 MÉTODOS CUANTITATIVOS	- 72 -
5.2.1 Diagrama Log-escala – Estimación de H.....	- 72 -
5.2.1.1 La script LDestimate.m	- 74 -
5.2.1.1.1 Acerca de newchoosej1.m	- 76 -
5.2.1.2 Utilizando LDestimate	- 76 -
5.2.2 Método de Varianza versus Octavas.....	- 82 -
5.2.2.1 Utilizando varianzaW.m	- 83 -
5.3 SÍNTESIS DE PROCESOS AUTOSIMILARES	- 84 -
5.3.1 Utilización de Wavelets para la Generación de un proceso FGN con Parámetro H Predeterminado	- 84 -
5.3.1.1 Utilizando FGNW.m.....	- 85 -
5.3.2 Utilización de la Transformada Inversa de Fourier para la Generación de un proceso FGN con Parámetro H Predeterminado	- 86 -
5.3.2.1 Script FGNIFourier.m	- 86 -
6 DISCUSIÓN DE LOS RESULTADOS	- 87 -
6.1 EL ANÁLISIS EN EL DOMINIO DE LA ESCALA	- 87 -
6.2 DISCUSIÓN DE LA ESTIMACIÓN MEDIANTE LA SCRIPT LDESTIMATE	- 87 -
6.2.1 Ejemplos de análisis en distintos contextos	- 89 -
6.2.2 Análisis de capturas de tráfico real de distintas redes	- 91 -
6.2.2.1 Descripción de las Trazas.....	- 92 -
6.2.2.2 Trazas de Redes Ethernet	- 92 -
6.2.2.2.1 Trazas Ethernet 10 Mb.....	- 92 -
6.2.2.2.2 Trazas Ethernet 100 Mb.....	- 92 -
6.2.2.2.3 Trazas Gigabit Ethernet	- 92 -
6.2.2.2.4 Trazas 10 Gigabit Ethernet Cluster TeraGrid SDSC.....	- 92 -
6.2.2.3 Trazas sobre Fibra Óptica	- 93 -
6.2.2.3.1 Trazas OC12 AMPHAT Miami Florida	- 93 -
6.2.2.3.2 Trazas OC 48 Link entre ISP's	- 94 -
6.2.2.3.3 Trazas OC192 Datacenter Equinix en Chicago.....	- 94 -
6.2.3 Procedimiento previo aplicado a las trazas.....	- 95 -
6.2.4 Tabla de Resultados Obtenidos con LDestimate	- 95 -
6.3 DISCUSIÓN DE LA ESTIMACIÓN MEDIANTE LA SCRIPT VARIANZA W.M	- 96 -
6.4 TABLA COMPARATIVA DE RESULTADOS OBTENIDOS CON DISTINTOS MÉTODOS DE ESTIMACIÓN DE “H”	- 97 -
6.4.1 Acerca de la Exactitud.....	- 97 -
6.4.2 Acerca de la Precisión.....	- 98 -
6.5 DISCUSIÓN SOBRE LA SÍNTESIS DE PROCESOS AUTOSIMILARES	- 99 -
6.5.1 Comparando tiempos para generación de trazas sintéticas	- 99 -
6.5.2 Acerca del uso de las script FGNW.m y FGNIFourier.m	- 100 -
6.5.2.1 La script correlacionFBM.m	- 100 -
6.5.2.2 Analizando FGNW.m	- 101 -
6.5.2.3 Analizando FGNIFourier.m	- 101 -
6.5.3 La script correlacionFBM.m en el contexto de tráfico real.....	- 102 -
6.6 CONCLUYENDO LA DISCUSIÓN.....	- 105 -
7 CONCLUSIONES.....	- 107 -

7.1 APORTE COMO MARCO O ESTRUCTURA DIDÁCTICA	- 107 -
7.2 APORTE DE LAS APLICACIONES	- 108 -
7.2.1 <i>Aplicaciones desarrolladas</i>	- 108 -
7.3 LA VISIÓN DEL ESTUDIO DEL TRÁFICO AUTOSIMILAR DESDE LA ÓPTICA DE ESTE TRABAJO	- 109 -
7.4 ALGUNAS LÍNEAS DE TRABAJOS FUTUROS.....	- 110 -
7.5 UNAS ÚLTIMAS PALABRAS.....	- 110 -
8 REFERENCIAS BIBLIOGRÁFICAS.....	- 111 -
9 APÉNDICE 1 - CÓDIGO FUENTE DE LAS APLICACIONES.....	- 115 -
9.1 CODIGO FUENTE DE LAS SCRIPTS PARA MATLAB®.....	- 115 -
9.1.1 <i>Script estudioGabor.m</i>	- 115 -
9.1.2 <i>Script estudioWav.m</i>	- 116 -
9.1.3 <i>Script cargaVectores.m</i>	- 117 -
9.1.4 <i>Script serieTescalada.m</i>	- 118 -
9.1.5 <i>Script comparaGrafica.m</i>	- 119 -
9.1.6 <i>Script rangoRescalado.m</i>	- 120 -
9.1.7 <i>Script rangoRescaladocomparado.m</i>	- 120 -
9.1.8 <i>Script rangoRescaladopromedio.m</i>	- 121 -
9.1.9 <i>Script varianzaTiempo.m</i>	- 122 -
9.1.10 <i>Script varianzaTiempocomparado.m</i>	- 123 -
9.1.11 <i>Script varianzaW.m</i>	- 124 -
9.1.12 <i>Script FGNI Fourier.m</i>	- 124 -
9.1.13 <i>Script correlaciónFBM.m</i>	- 124 -
9.1.14 <i>LDestimate.m</i>	- 125 -
9.1.15 <i>FGNDW.m</i>	- 125 -
9.1.16 <i>Uvi_Wave.300</i>	- 125 -
10 APÉNDICE 2 – PROCESOS ESTOCÁSTICOS.....	- 127 -
10.1 DEFINICIÓN DE PROCESO ESTOCÁSTICO	- 127 -
10.2 ERGODICIDAD.....	- 127 -
11 APÉNDICE 3 – INGENIERÍA DE TRÁFICO.....	- 129 -
11.1 DEFINICIÓN DE INGENIERÍA DE TRÁFICO	- 129 -
11.2 TEORÍA DE COLAS	- 129 -
11.3 FÓRMULA ERLANG B.....	- 130 -
11.4 PROCESO DE POISSON	- 131 -

Nomenclatura y símbolos utilizados en este trabajo

Acerca de los símbolos y operadores utilizados en este trabajo no hay nada en particular que difiera de la notación matemática estándar, de todas maneras los símbolos poco usuales, o que pudieran presentar alguna duda se encuentran debidamente aclarados en los párrafos donde se utilizan.

Los capítulos se numeran consecutivamente empezando por el uno. Los correspondientes títulos y subtítulos son numerados en forma anidada siempre comenzando por el número de capítulo.

La notación utilizada para numerar e identificar las ecuaciones es [x.y], donde “x” es el número de capítulo, siendo “y” el número de ecuación que empieza en uno, en cada capítulo respectivamente.

Las notas al pie de página, están referenciadas como un número superíndice, y se numeran en forma consecutiva desde el uno en adelante, manteniendo esa numeración para todo el trabajo.

Las referencias bibliográficas se señalan con un número entre corchetes [n], “n” es el número de orden que identifica la referencia en el listado de las mismas al final de este trabajo. El listado de referencias bibliográficas responde al formato ISO 690, y se encuentra ordenado cronológicamente, empezando por el uno, desde las más antiguas a las más recientes.

Para facilitar eventuales consultas en algunas citas, junto al identificador [n], también se señala un número de página por ejemplo: [26] Pág. 23.

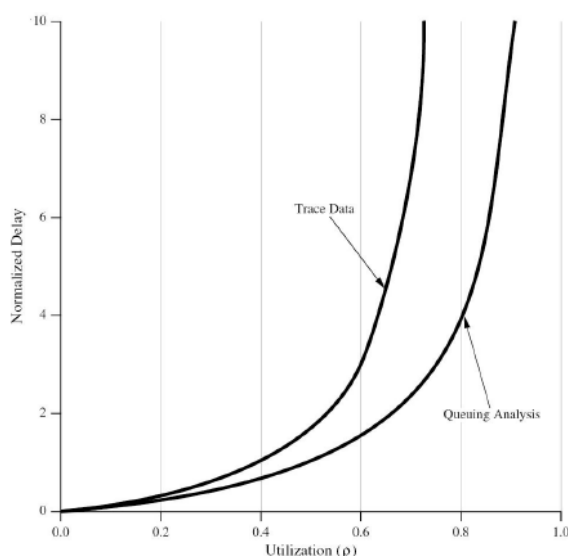
1 Motivación de este trabajo

1.1 El Contexto

Una actividad fundamental en el ámbito de estudio de las redes telemáticas¹, es el análisis del tráfico que transportan, pues el conocimiento de una red, requiere observación y medición de características tales como, flujo de datos, tiempos de arribo, tiempos de transmisión, comportamiento del tráfico en diferentes escalas de tiempo, etc. Este conocimiento permite optimizar los recursos de las redes, y también posibilita, que los servicios ofrecidos, cuenten con la calidad requerida, por otra parte las expectativas de los usuarios, acerca de las capacidades y prestaciones de las redes, aumentan día a día. Dar una respuesta apropiada a esta demanda, es lo que se conoce como ingeniería de tráfico², y un factor determinante en el desarrollo de esta actividad, es poder anticipar y analizar el comportamiento que tendría una red, o parte de ella bajo determinadas demandas. El logro este objetivo, es un área activa de estudio e investigación.

Cuando se habla de capacidades y prestaciones, en particular refiere a una amplia gama de tráficos posibles, esto es la utilización simultánea de los recursos de la red, por distintas fuentes u orígenes de tráfico, como ser correo electrónico, servicios Web, telefonía (voz), video, intercambio de archivos, mensajería, servicios multimedia y muchos otros, incluyendo la posibilidad de futuros servicios aún no implementados. Llamamos a esto la integración de los servicios y es la tendencia hacia la que converge el uso de las redes con independencia de su tipo o tecnología.

Promediando el año 1990, estudios realizados sobre muestras de tráfico tomadas de redes en funcionamiento³, han demostrado en forma inequívoca que el tráfico tiene propiedades autosimilares, esto es, la existencia de patrones estadísticos o comportamientos que se repiten a diferentes escalas de tiempo. Un tráfico con características autosimilares, afecta en forma negativa el desempeño de la red diseñada en función de un tráfico no autosimilar, o dicho de otra forma, si se diseña una red en forma clásica presumiendo un tráfico de características totalmente aleatorias e independientes (no autosimilar), se estaría subestimando la demanda en presencia de tráfico autosimilar. Para ilustrar esta circunstancia, en el siguiente gráfico extraído de [36], se puede observar que el retardo promedio de los paquetes se “dispara”, mucho antes de lo previsto por el análisis de colas tradicional.



¹ El término telemática alude a la fusión de las telecomunicaciones con la informática

² Ver Apéndice 3 – Ingeniería de Tráfico; punto 11.1

³ Se puede decir que el trabajo de W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, On the self-similar nature of ethernet traffic (extended version), IEEE/ACM Transactions on Networking, vol.2, pp.1–15, Feb. 1994, dio inicio a una nueva concepción en el estudio del tráfico.

El fenómeno de la autosimilitud en el tráfico evidenció que, los modelos tradicionales heredados de la telefonía, ampliamente conocidos y utilizados, no reproducen adecuadamente el comportamiento autosimilar del tráfico, el cual tiene un impacto negativo en el desempeño de la red, debido fundamentalmente a que este presenta complejas estructuras correlación y dependencia mientras que los modelos tradicionales están basados en procesos aleatorios independientes y por ende incorrelados. Por este motivo, resulta inapropiado aplicar los modelos tradicionales al tráfico autosimilar, pues éste no cumple con las condiciones de aleatoriedad e independencia que son necesarias para que sean válidos los análisis en que se basan los modelos tradicionales.

Esto es una circunstancia lamentable, en el sentido que los modelos tradicionales fundados en la teoría de colas, tienen resuelta su tratabilidad analítica, y tal vez su mejor ejemplo esté representado por la fórmula de Erlang "B"⁴ y las expresiones de Little⁵, que han sido ampliamente validadas en el ámbito de la telefonía y por ende muy utilizadas, de hecho todavía se consideran muy útiles y están vigentes en muchos contextos.

En oposición a lo anterior, los modelos que pueden capturar o reproducir el comportamiento autosimilar, son poco dóciles al tratamiento analítico, y además, teniendo en cuenta que la demanda impuesta en la actualidad por los usuarios de las redes generan múltiples tipos de tráfico, incluso tráfico agregado debido al uso de multiplexación, encontrar una relación funcional entre los parámetros que lo caracterizan, y desarrollar modelos matemáticos con solución analítica de manera compacta, es una tarea prácticamente imposible.

En este escenario, con tráfico originado en diversas fuentes, con sus respectivas características y particularidades, abordar un estudio para cuantificar o medir de manera apropiada la demanda que los usuarios imponen sobre los recursos de una red, requiere del uso de modelos que mediante abstracciones matemáticas, representen de una manera eficaz y eficiente un comportamiento compatible con las características observables en el tráfico real; surgen entonces, dos desafíos de importancia central:

1. El desarrollo de modelos generales que abarquen las principales características del tráfico a estudiar.
2. El desarrollo de aplicaciones, que utilizando esos modelos, permitan obtener conclusiones válidas.

1.2 El Planteo

Esta tesis se plantea en el contexto del segundo punto mencionado arriba. Aquí la simulación se erige como el paradigma de estudio más viable, dado que se la emplea cuando el proceso es muy complejo para ser estudiado de forma analítica. La simulación permite abordar el estudio de los complejos **modelos autosimilares** y producir estimaciones acertadas o lo suficientemente ajustadas a la realidad observable.

Esto también se puede conseguir mediante **generadores** de tráfico sintético, el cual imita al tráfico real en sus parámetros estadísticos, este tráfico sintético, se puede utilizar introduciéndolo

⁴ Agner Krarup Erlang, ingeniero danés que trabajó para la Copenhagen Telephone Exchange, publicó el primer artículo sobre la teoría de colas en 1909. David G. Kendall introdujo una notación de colas A/B/C en 1953. La notación de Kendall para describir las colas y sus características puede encontrarse en Tijms, H.C, "Algorithmic Analysis of Queues". El Dr. Leonard Kleinrock (nacido en 1934) Profesor de Ciencias de la Computación en la UCLA, Su obra más conocida y significativa es su trabajo en teoría de colas (1962), que tiene aplicaciones en multitud de campos, entre ellos como fundamento matemático de la conmutación de paquetes, tecnología básica detrás de Internet. Ver apéndice 3 puntos 11.2 y 11.3.

⁵ Trabajo titulado: A Proof of the Queuing Formula : $L=\lambda W$. Se puede acceder en: <http://www.doc.ic.ac.uk/~uh/forpaul/A%20PROOF%20FOR%20THE%20QUEUEING%20FORMULA%20L=IW.%207689045.pdf> Ver Apéndice 3 punto 11.2.

en un sistema, que puede ser real (una red o nodo de la misma), o virtual (software simulador), permitiendo de este modo, observar el comportamiento, analizar y obtener conclusiones válidas. En el ámbito de la simulación es importante conocer el valor de los parámetros que caracterizan un determinado tipo de tráfico, esto se consigue mediante el uso de **estimadores** de parámetros, los cuales tienen que ser fiables y eficaces en su misión de hallar el valor más aproximado al real con un mínimo sesgo.

Para evitar ambigüedad en el significado de los términos utilizados (resaltados en negrita), se los describe a continuación.

Se entiende como **modelo de tráfico**, a una abstracción matemática cuyo objetivo es representar cuantitativamente el comportamiento del tráfico real, un modelo puede considerarse como una entidad que captura la esencia de la realidad sin la presencia de la misma, esto se consigue mediante la obtención de parámetros que puedan caracterizar el tráfico, y al introducir estos parámetros en los modelos, se puede emular el comportamiento del tráfico con las características que sean necesarias.

Se entiende como **generador de tráfico**, la implementación ya sea por hardware o software de un determinado modelo de tráfico, el cual produce trazas o muestras de tráfico sintético, de forma tal que las muestras de tráfico resultante, tengan las propiedades estadísticas del modelo del que procede. La implementación de generadores de tráfico obedece a la necesidad de crear trazas sintéticas que emulen comportamientos de tráfico con determinadas características, su utilidad se manifiesta cuando no se dispone de muestras, o trazas reales capturadas de una red, o bien para el caso de redes que aún no existen, las que están en vía de desarrollo o ampliación, o también en aquellos casos donde no es posible obtener directamente las muestras por alguna cuestión operativa o de seguridad.

Se entiende como **estimador de parámetro**, a la aplicación de un procedimiento de análisis sobre el conjunto de datos que representan un determinado tráfico, para obtener un estimado del verdadero valor de un parámetro. La correcta elección de un estimador es fundamental, pues cuanto más se aproxime un parámetro a su valor verdadero, redundará en simulaciones con un alto grado de confiabilidad, de aquí la importancia que tienen los estimadores utilizados para ajustar el valor de un parámetro.

Otra ventaja de la simulación del tráfico, es que sin alterar ninguna infraestructura o instalación real, se puede repetir el estudio tantas veces como se quiera, cambiando los parámetros del tráfico, y así observar como inciden estos cambios en el comportamiento del sistema estudiado, de esta manera, se pueden obtener las estimaciones que sean necesarias. Además siempre se pueden contrastar los resultados y eventualmente ajustar los parámetros para que se aproximen lo máximo posible, al comportamiento real del sistema simulado.

Todo lo expuesto anteriormente muestra la necesidad de contar con modelos, generadores y estimadores que cumplan en lo posible con los criterios de:

- Reproducir lo más fielmente posible las características del tráfico que representa.
- Tener la suficiente tratabilidad analítica (modelos parsimoniosos).
- Tener la posibilidad de implementación mediante algoritmos.
- Tener la suficiente confiabilidad estadística.
- Tener un coste computacional razonable.

Evidentemente, poder cumplir con los criterios mencionados, requiere de un compromiso entre los mismos para obtener resultados óptimos, además la selección adecuada de un modelo, un generador, o un estimador exige un adecuado conocimiento de los fundamentos teóricos en los que se apoya.

1.3 ¿Por qué utilizar las Wavelets?

En la lectura de diversos estudios e investigaciones realizados en los últimos años sobre el tráfico autosimilar en las redes telemáticas, se evidencia que, existe consenso acerca de las ventajas que aportan los métodos basados en las wavelets u onditas en este campo, esto es, atendiendo a criterios de validez, confianza estadística y eficiencia computacional. En algunas investigaciones, se realizaron rigurosos análisis, contrastando los resultados obtenidos utilizando las wavelets, con otros métodos también muy difundidos; y en sus conclusiones, se puede apreciar la conveniencia del uso de las wavelets u onditas en este campo. En consecuencia, la utilización de las wavelets se ha convertido en una útil y eficaz herramienta para tareas de análisis, detección, estimación, modelado y simulación en el ámbito del tráfico autosimilar.

Como se menciona en las referencias bibliográficas [18] Pág. 84, y [26] Pág. 23; entre las ventajas del estudio del tráfico autosimilar por medio de wavelets u onditas podemos mencionar:

- La wavelets u onditas ofrecen un marco teórico que se puede aplicar tanto a procesos autosimilares, procesos LRD (Long Range Dependence, o dependencia de rango largo), trazas muestrales etc. Pudiendo hacer un análisis en el dominio de la escala, de forma que se adapta “naturalmente” a las necesidades de poder estudiar un comportamiento en este dominio.
- Permite la división controlada de un proceso madre de variabilidad extrema, en subprocesos a diferentes escalas tornando manejable su comportamiento, aprovechando la independencia de los subprocesos obtenidos, se pueden emplear herramientas de la estadística clásica sobre las secuencias de los coeficientes wavelets, y de esta forma poder diseñar estimadores simples y eficientes.
- Los bancos de filtros de análisis y síntesis, proporcionan una forma computacionalmente eficiente de llevar a cabo tareas de análisis y síntesis de procesos autosimilares.

1.4 ¿Cuales son los Objetivos de esta Tesis?

Este trabajo, esta orientado al estudio del trafico autosimilar en las redes telemáticas, fundamentado en la revisión crítica, de diversos trabajos científicos realizados en este campo hasta el presente, y al desarrollo de aplicaciones practicas basadas en las wavelets u onditas utilizando el software MATLAB®. El objetivo de este trabajo se formaliza a través de dos ejes:

1. Brindar un marco o estructura didáctica que permita el desarrollo del tema como una unidad de estudio y aprendizaje.
2. El desarrollo de scripts o aplicaciones, para abordar el estudio de distintas características y/o propiedades en el contexto de la autosimilitud, y de esta manera, permitir el análisis, detección de comportamiento, estimación de parámetros, y la producción de tráfico sintético, de manera que puedan utilizarse en el ámbito de la simulación.

Se considera este trabajo como un aporte curricular razonablemente autocontenido, con miras a facilitar el estudio de la temática considerada, y servir de base a otros estudios que contemplen la autosimilitud en el contexto del tráfico en las redes telemáticas.

1.5 ¿Por que MATLAB®?

Los productos de The MathWorks^{TMTM} ya están siendo ampliamente utilizados en la comunidad universitaria para hacer más real y tangible la educación científica y técnica. Más de 3.500 universidades de todo el mundo⁶ utilizan productos de The MathWorksTM para las tareas de enseñanza e investigación en un amplio espectro de disciplinas técnicas. MATLAB®

⁶ Dato extraído de <http://www.mathworks.es/academia/> .

(MatrixLaboratory) es un programa interactivo de uso general en Ciencias e Ingeniería. Resulta muy utilizado por sus características que permiten abordar estudios complejos en forma eficiente, con un mínimo esfuerzo de programación y además ofrece la posibilidad de una versión [Student](#) gratuita para prueba; la interacción se realiza mediante instrucciones (comandos), y también mediante funciones predefinidas, y programas (scripts). Los objetos básicos con los cuales opera MATLAB® son matrices. A continuación una cita extraída de la página [Web The MathWorks™](#) como una breve descripción.

“La potencia del cálculo técnico con MATLAB®

Cualquiera que sea el objetivo de su trabajo, un algoritmo, un análisis, un gráfico, un informe o una simulación de software - el cálculo técnico con MATLAB le permite trabajar con mayor eficacia y eficiencia. El entorno flexible de MATLAB le permite realizar análisis avanzados, visualizar datos y desarrollar algoritmos en una fracción del tiempo que se tarda en C, C++ o Fortran. El lenguaje abierto de MATLAB le permite compartir rápidamente ideas y desarrollar soluciones.

Con sus más de 600 funciones matemáticas, estadísticas y técnicas, MATLAB le proporciona acceso inmediato al cálculo numérico de alto rendimiento. Esta funcionalidad se amplía con funciones gráficas interactivas para crear gráficos 2D y 3D, imágenes, superficies y representaciones volumétricas.

Los algoritmos punteros de los toolboxes realzan la funcionalidad de MATLAB en dominios tales como procesamiento de señales e imágenes, análisis de datos y estadística, creación de modelos matemáticos y diseño de controles. Los toolboxes son colecciones de algoritmos, escritas por expertos en su campo, que proporcionan funciones numéricas para áreas de aplicación específicas, de análisis y gráficas. Confiando en el trabajo de estos expertos usted puede comparar y aplicar varias técnicas sin escribir códigos. Lo mismo que con los algoritmos de MATLAB, puede personalizar las funciones del toolbox para los requisitos de su proyecto. MATLAB y los toolboxes acompañantes proporcionan a los ingenieros, científicos, matemáticos y educadores un potente conjunto de herramientas. Estos productos sirven para una amplia gama de tareas en una gran variedad de sectores, desde el del automóvil y la electrónica al de equipamientos industriales y telecomunicaciones.

El cálculo técnico con MATLAB le permite acelerar su investigación, reducir el tiempo y los costes de desarrollo y dar mejores productos.”

2 Estado del Arte en el Estudio del Tráfico Autosimilar en las Redes Telemáticas

Para situarnos en el contexto del presente trabajo y mostrar un poco la evolución del estudio del tráfico autosimilar, a continuación se hace mención acerca de cual fue su génesis con los primeros trabajos realizados en los años 90, y como se encuentra la investigación en la actualidad, a la luz de las evidencias empíricas encontradas en diversas muestras tomadas de distintas redes, que responden a distintas tecnologías, y exhiben comportamientos compatibles con la autosimilitud, o a veces cierta aleatoriedad e independencia en algunos contextos.

2.1 Punto de Partida

Se define en forma genérica al tráfico como una sucesión de elementos que son transmitidos por enlaces o vínculos entre nodos de una red. Estos elementos pueden interpretarse, según el contexto, como paquetes, tramas, bits, llamadas, mensajes, flujos o cualquier otra unidad de información adecuada; los cuales transitan entre los nodos de la red.

El tráfico es una consecuencia de la demanda impuesta por los usuarios sobre los recursos de una red, evidentemente se puede asumir una alta variabilidad e indeterminación del comportamiento de los usuarios o en otra palabras la impredecibilidad de su comportamiento, entonces resulta de utilidad tratar al tráfico como una serie o secuencia temporal, y en este contexto se pueden modelar las series temporales, como realizaciones de procesos estocásticos.

Tradicionalmente se utilizaron modelos llamados “Poissonianos”, por derivar de distribuciones de probabilidad del tipo Poisson donde los tiempos entre arribos tienen una distribución exponencial; llamados también tráfico “Poisson”, estos modelos son conocidos como “sin memoria” pues debido a la distribución exponencial el tiempo que falta para la próxima llegada es totalmente independiente del tiempo transcurrido, y es precisamente esta independencia la que facilita el tratamiento analítico e hizo tan populares a estos modelos. En particular si graficamos una serie que representa un tráfico Poissoniano utilizando distintas escalas de tiempo, se observa que a medida que aumenta la escala de tiempo, las características de fluctuación, o variabilidad, se pierden y el tráfico modelado tiende a aplanarse, siendo ésta una manifestación gráfica de que el modelo no tiene memoria.

Ahora bien, si graficamos trazas o muestras de tráfico en distintas escalas de tiempo, y se observa que mantienen la fluctuación, aún al aumentar la escala del tiempo, entonces estamos ante una manifestación gráfica de que el tráfico tiene memoria o conducta persistente en distintas escalas.

Esto último es precisamente es lo que ocurre cuando se grafican trazas de muestras extraídas de ciertos tráficos de redes en funcionamiento; entonces, si un modelo estocástico puede capturar o representar las propiedades de variabilidad del tráfico (fluctuación), y la mantiene, aún cuando lo analizamos a mayores escalas en el tiempo, se dice que el modelo tiene memoria o características autosimilares.

La autosimilitud es un concepto que viene de la teoría de los Fractales, un fractal es un objeto geométrico cuya estructura básica se repite en diferentes escalas de una determinada magnitud. El término fue propuesto por Benoît Mandelbrot⁷ en 1975. En muchos casos los fractales pueden

⁷ Benoit B. Mandelbrot, principal creador de la Geometría Fractal, en 1982 publicó su libro *Fractal Geometry of Nature* en el que explicaba sus investigaciones en este campo. La geometría fractal se distingue por una aproximación más abstracta a la dimensión de la que caracteriza a la geometría convencional.

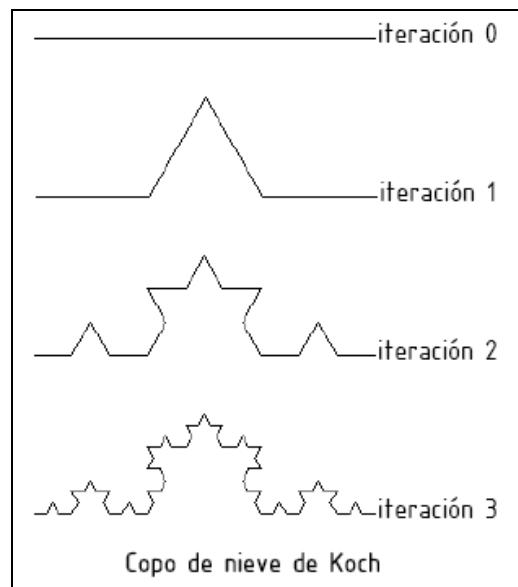
ser generados por un proceso recursivo o iterativo capaz de producir estructuras autosimilares independientemente de la escala específica.

Los fractales son estructuras geométricas que combinan irregularidad y estructura.

En la naturaleza también aparece la geometría fractal, aunque muchas estructuras naturales tienen estructuras de tipo fractal, un fractal matemático es un objeto que tiene por lo menos una de las siguientes características:

- Tiene detalle en escalas arbitrariamente pequeñas.
- Es demasiado irregular para ser descrito en términos geométricos tradicionales.
- Tiene autosimilitud exacta o estadística.
- Puede ser definido recursivamente.

Ejemplos de fractalidad:



Fractal Natural: *Dryopteris Erythrosora* ⁸ Fractal Matemático: La curva de Von Koch ⁹

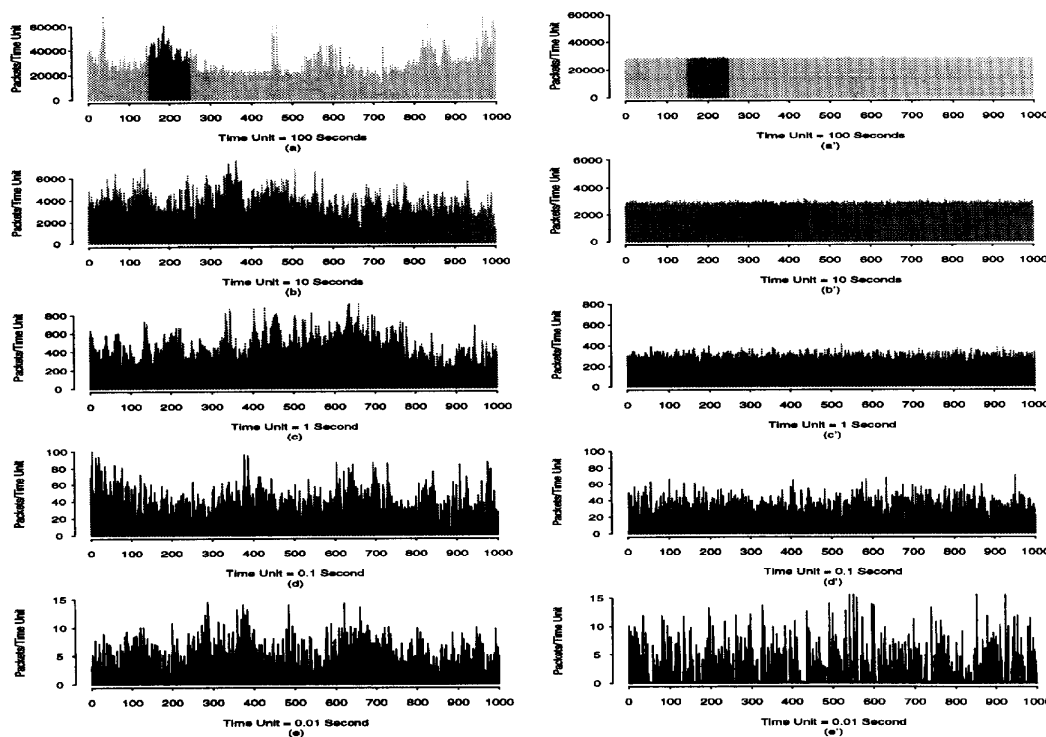
En resumen si observamos una estructura que se parece a si misma es decir con patrones que se repiten, sin importar la escala que usemos para observarla, entonces estamos en presencia de autosimilitud.

Para ilustrar un poco este concepto en el contexto del tráfico, veamos lo siguiente, si se observa una captura del tráfico de una red Ethernet, donde se grafica el tráfico capturado como una serie temporal, veremos que presenta la misma apariencia fluctuante, con independencia de la escala de tiempo que usemos para visualizarlo, sea este de milisegundos, segundos, minutos u horas. Este fenómeno fue estudiado y publicado en el año 1994, bajo el nombre “On the Self-Similar Nature of Ethernet Traffic (Extended Version)”, por Will E. Leland, Murad S. Taqqu, Walter Willinger, y Daniel V. Wilson [1], donde demostraron que el tráfico Ethernet de varias redes LAN de los laboratorios Bellcore Morristown Research and Engineering Center, presentaban características de autosimilitud estadística, dando inicio a una nueva concepción en el estudio del tráfico, el que hasta entonces generalmente se estudiaba con modelos tradicionales heredados de la telefonía, pero mediciones precisas como las efectuadas en el trabajo mencionado se evidenciaron importantes desviaciones en los comportamientos previstos por dichos modelos,

⁸ Extraído de [28] Pág. 12.

⁹ Extraído de <http://www.cprmarmenor.com/rfc06-07/sesion3/recursos3.pdf>.

fundamentalmente en las medidas de desempeño, las cuales resultaban afectadas en forma negativa cuando las características del tráfico eran autosimilares, proponiéndose en consecuencia, nuevos modelos que reflejen el comportamiento autosimilar. En la siguiente imagen extraída del trabajo mencionado, se puede apreciar la apariencia fluctuante del tráfico real de la red en comparación al tráfico sintético generado por un modelo tipo Poisson, la escala de tiempo va aumentando desde 0.01 seg. hasta 100 seg., como se ve a la derecha a medida que el tiempo aumenta el aspecto del tráfico tipo Poisson se va aplanando mientras que a la izquierda el tráfico real mantiene su aspecto fluctuante (burstiness).



2.2 La Ubicuidad de la Autosimilitud

Con posterioridad al estudio de Leland et al., el fenómeno de autosimilitud también fue detectado en muchos otros tipos de tráfico en las redes y descrito en numerosos trabajos que podemos considerar seminales en este campo de estudio, por citar algunos ejemplos:

- Vern Paxson and Sally Floyd, Wide-Area Traffic: The Failure of Poisson Modeling IEEE/ACM Transactions on Networking, Vol. 3 No. 3, pp. 226-244, June 1995.
- Mark E. Crovella and Azer Bestavros, Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes in IEEE/ACM Transactions on Networking, 5(6):835--846, December 1997. (An earlier version of this work appeared under the BU technical report TR-95-015 Boston University Computer Science Department, Revised, October 12, 1995).
- B. K. Ryu and A. Elwalid, The Importance of Long-range Dependence of VBR Video Traffic in ATM Traffic Engineering: Myths and realities ACM Computer Communication Review, vol. 26, pp. 3-14, Oct. 1996.
- Kihong Park, Gi Tae Kim, and Mark E. Crovella, On the Relationship Between File Sizes, Transport Protocols, and Self-Similar Network Traffic. In Proceedings of the International Conference on Network Protocols, pages 171-180, October, 1996. (Also available as Technical Report BU-CS-96-016, Boston University, Computer Science Dept., July, 1996).
- Feldmann, A. C. Gilbert, W. Willinger, and T. G. Kurtz, The Changing Nature of Network Traffic: Scaling Phenomena, ACM Computer Communication Review, vol. 28, pp. 5-29, Apr. 1998.

No obstante, aun resumiendo, la lista resultaría demasiado extensa, pero se puede obtener una idea de la abundancia de trabajos y el interés que despertó el tema en la comunidad de estudio en aquellos años, leyendo el siguiente trabajo que figura en las referencias bibliográficas como: [9] Walter Willinger, Murad Taqqu, and Ashok Erramilli, A Bibliographical Guide to Self-Similar Traffic and Performance Modeling for Modern High-Speed Networks Stochastic Networks: Theory and Applications, Royal Statistical Society Lecture Notes Series, Vol. 4, Oxford University Press, 1996.

Se trata de una guía bibliográfica muy extensa en el área de tráfico autosimilar. Aunque hoy se puede considerar como muy antigua, proporciona citas sobre una variedad de 420 trabajos hechos en este campo de estudio, desde sus inicios hasta el año 1996, y tiene como utilidad la de darnos a conocer esos trabajos y comparar sus conclusiones con trabajos actuales.

2.3 La autosimilitud en la actualidad

Se puede afirmar que debido a la fuerte evidencia empírica de la ubicuidad de la autosimilitud en muchos tipos de tráfico, sigue siendo un tema abierto a debate el impacto que tiene en desempeño de una red, tornando necesario el desarrollo de herramientas para la comprensión de los procesos autosimilares y la evaluación de los mismos. Como ejemplo de esto, cito algunos trabajos que podemos considerar actuales, como muestra del interés que sigue concitando esta temática entre los investigadores:

- Thomas Karagiannis, Mart Molle, and Michalis Faloutsos University of California Long-Range Dependence Ten Years of Internet Traffic Modeling, , IEEE INTERNET COMPUTING 1089-7801/04/\$20.00 © 2004 IEEE Published by the IEEE Computer Society SEPTEMBER • OCTOBER 2004.
- Alarcon-Aquino, V., Guerrero-Ojeda, L.G., Rodriguez-Asomoza, J. et al. Análisis de Tráfico Auto-similar en Redes de Comunicaciones Usando Onditas (Wavelets). Inf. tecnol., 2005, vol.16, no.2, p.61-66. ISSN 0718-0764.
- Xiaofeng Bai, A. Shami, “Modeling Self-Similar Traffic for Network Simulation,” Technical report, NetRep-2005-01, April 2005.
- Carla Di Cairano-Gilfedder, A decade of internet research: advances in models and practices, BT Technology Journal, Volume 23, Number 4, October 2005, pp. 115-128(14).
- Thomas M. Chen, Network Traffic Modeling Chapter in The Handbook of Computer Networks, Hossein Bidgoli (ed.), Wiley, to appear 2007.
- Oleg I. Sheluhin Moscow State Technical University of Service (MSTUS), Russia Sergey M. Smolskiy Moscow Power Engineering Institute (MPEI), Russia Andrey V. Osin Moscow State Technical University of Service (MSTUS), Russia SELF-SIMILAR PROCESSES IN TELECOMMUNICATIONS ISBN 978-0-470-01486-8 (HB) John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England Hardcover 334 pages April 2007.
- William Rea, Les Oxley, Marco Reale y Jennifer Brown, Estimators for Long Range Dependence: An Empirical Study Electronic Journal of Statistics. Vol. 0 (2009) ISSN: 1935-7524.
- Clegg Richard G.; Landa Raul; Rio Miguel, Criticisms of modelling packet traffic using long-range dependence. Presented at the PMECT Oct 2009 workshop (part of ICCCN 2009) in San Francisco.

2.4 Corolario

El éxito de los modelos autosimilares radica en su capacidad de capturar las complejas dependencias que muestra el tráfico a distintas escalas de tiempo mediante el uso de pocos

parámetros, en particular el parámetro de Hurst " H ",¹⁰ pero como contraparte, es un desafío su aplicación al análisis de las prestaciones y revelar su verdadero impacto en el desempeño de una red.

Dada la importancia central del parámetro de Hurst " H ", en la caracterización del tráfico, es necesaria su correcta detección y estimación. En pos de evaluar la calidad de los diferentes estimadores, es necesaria la generación de secuencias de números con autosimilitud exacta, con H , conocido y lo mejor ajustado posible. La estimación exacta para secuencias finitas, es todavía un problema abierto, y según estudios realizados, los estimadores Whittle MLE (Maximun Likelihood Estimator) y los basados en wavelets son los que presentan menor sesgo, siendo estos últimos más eficientes en varios aspectos.

Hay evidencias recientes, en mediciones efectuadas sobre tráfico real, que parecerían indicar un fuerte comportamiento autosimilar en redes LAN del tipo Ethernet, y un comportamiento que tiende a ser del tipo Poisson en importantes backbones de distintas tecnologías, y con elevados niveles multiplexación. Este escenario, plantea claramente la necesidad de contar con procedimientos para detectar fehacientemente fenómenos de escala, y eventualmente estimar el valor confiable de " H ", como recurso para caracterizar en forma correcta el tráfico.

Los trabajos citados arriba, son nada más que para mostrar de donde viene, cual es la evolución y cuales son las tendencias actuales dentro de lo que es el estudio del tráfico y en particular el tráfico autosimilar utilizando wavelets, pues en realidad sigue siendo existiendo mucha producción científica en este campo y también cierta controversia acerca de sus resultados. Esta claro que el modelado del tráfico es un proceso dinámico y cambiante conforme avanzan, tanto nuestra comprensión de los fenómenos involucrados, como la tecnología de las redes, entonces lo que sabemos hoy, con seguridad cambiará y será distinto en el futuro.

Esta tesis tiene entre sus objetivos, analizar y utilizar las conclusiones de una selección de trabajos, constituida con aquellos que se considera, son los que de la forma más clara, exponen y representan los avances hechos en el estudio del tráfico autosimilar en las redes hasta ahora, poniendo en perspectiva lo que se aprendió a lo largo de estos años. Si bien no es objetivo de este trabajo, el estudio del fenómeno de autosimilitud en si mismo, como entidad matemática, resulta necesario el conocimiento y manejo de los conceptos básicos relacionados al mismo, los cuales ya se encuentran estudiados y desarrollados para el ámbito de las redes, por este motivo se intenta brindar un marco didáctico, manteniendo un carácter tutorial, donde se describen, explican, y proveen referencias, de las principales características, definiciones y propiedades de los mencionados conceptos, y así poder aplicarlos a los análisis, estimaciones y síntesis, utilizadas en el estudio del tráfico autosimilar.

¹⁰ " H ", es conocido como parámetro de Hurst, y representa una medida de la persistencia del comportamiento de un proceso estocástico. Se origino a partir de las observaciones que llevo a cabo el ingeniero Harold Edwin Hurst al analizar las corrientes del río Nilo, para lo cual estudio el comportamiento de las series temporales del caudal producido por el mismo, este análisis lo llevo a desarrollar un método estadístico llamado análisis R/S (análisis de rango reescalado) que describió con detalle en su libro "Long-Term Storage: An Experimental Study", del año 1965.

3 Los Procesos Autosimilares y el Tráfico – Marco Teórico

3.1 Tráfico de una red

El “tráfico”, en una red es una consecuencia de la demanda que imponen los usuarios sobre los recursos de la misma; y una forma de representar la demanda, es mediante un conjunto o agrupación de “llegadas”, el termino llegada se usa de manera general, las “llegadas”, podrían ser llamadas, sesiones, paquetes, celdas, flujos, bits o cualquier otra unidad de medida que se considere apropiada para medir o cuantificar la magnitud que llamamos tráfico. La representación de éste conjunto, generalmente es una secuencia o serie, que identifica alguna característica temporal relacionada con la demanda, como podrían ser:

- Los instantes de tiempo en que se producen las llegadas $\{t_1, t_2, \dots, t_n\}$.
- Un proceso de conteo $N(t) = \{\max n : t_n > t_0\}$ que representa el número total de llegadas que han ocurrido desde el instante t_0 hasta el instante t .
- Las secuencias de tiempos entre llegadas $\{(t_1 - t_0), (t_2 - t_1), \dots, (t_{n-1} - t_{n-2}), (t_n - t_{n-1})\}$, representando los intervalos de tiempo transcurrido entre las llegadas.

Dado que la demanda está representada por el comportamiento de las aplicaciones y los usuarios al requerir los recursos de la red, y que este comportamiento es totalmente aleatorio, se intenta una aproximación de naturaleza estadística para el tráfico producido por las entidades demandantes que pertenecen a la población o conjunto estudiado. Con este comportamiento estadístico, se pueden modelar series de tiempo que representen al tráfico, como realización de un proceso estocástico¹¹, o conjunto de variables aleatorias $\{X_{(t)}\}$. El proceso utilizado como modelo, debe evidenciar o capturar las principales características estadísticas del tráfico que representa, en relación al impacto que puedan tener en el desempeño de la red.

3.2 El Modelo General

Sea un proceso estocástico de tiempo discreto $X_{(t)}, t \in \mathbb{Z}$, donde $X_{(t)}$ representa el número de llegadas que se producen en un intervalo de tiempo $(t-1, t)$; el proceso de acumulación subyacente es $Y_{(t)}, t \in \mathbb{Z}$, donde $Y_{(t)}$ es el número de llegadas hasta el instante t ; de esta manera se puede formular un modelo general donde el tráfico está representado por el incremento o evolución de la variable aleatoria $X_{(t)}$:

$$X_{(t)} = Y_{(t)} - Y_{(t-1)} \quad [3.1]$$

Es necesaria la suposición de que el proceso $X_{(t)}$, tenga las características de un proceso estacionario, esto significa que su comportamiento o propiedades estadísticas, no cambien ante desplazamientos temporales, es decir que sin importar que porción temporal del proceso se estudie, se tendrá el mismo resultado; en caso contrario, el modelo no serviría como una descripción del fenómeno estudiado.

Para caracterizar un proceso estocástico, se utilizan propiedades estadísticas, como ser su Valor Medio $\mu = E[X_{(t)}]$, su Función de Autocorrelación $R_{(\tau)} = E[X_{(t)}X_{(t+\tau)}]$, su Autocovarianza

¹¹ Ver apéndice 2 – Procesos Estocásticos

$C(t, t + \tau) = E[(X_t - \mu) - (X_{t+\tau} - \mu)]$, entre otras. Tanto la estacionariedad, como las características estadísticas mencionadas, son relevantes en todos los análisis por eso a continuación se hace un repaso de estos y otros conceptos del campo de la estadística que están directamente involucrados en el estudio del tráfico.

3.3 Los Procesos y el Tráfico - Propiedades y Definiciones

3.3.1 Procesos estacionarios

Se dice que un proceso estocástico $X_{(t)}$ es estacionario en el sentido estricto si la distribución conjunta de sus variables es igual a la distribución conjunta de las mismas para cualquier desplazamiento temporal “ a ” de las mismas, esto es, $F_X(x, t) = F_X(x, at); \forall a, t \in \mathbb{R}$:

$$\{X_{(t)}\} \stackrel{d}{=} \{X_{(at)}\}; \forall a > 0 \in \mathbb{R}$$

Esto significa que $\{X_{(t)}\}$ y $\{X_{(at)}\}$ son procesos estadísticamente indistinguibles, donde a puede ser cualquier número real mayor que cero. Esta definición implica que tanto el valor medio, como la varianza permanecen constantes $\mu_x(t) = \mu; Var[X_{(t)}] = \sigma^2$, esto es, que no dependen del tiempo.

Un proceso estrictamente estacionario resultaría demasiado restrictivo como modelo o representación del tráfico real, dado que, no se puede esperar que algo tan aleatorio como el tráfico real, se comporte exactamente igual en cualquier instante que se considere. Para representar el tráfico real, serían de utilidad procesos con una estacionariedad menos restrictiva, y estos son los procesos conocidos como procesos estacionarios en sentido amplio, o débilmente estacionarios.

3.3.2 Procesos débilmente estacionarios

Se dice que un proceso $X_{(t)}$ es estacionario en el sentido amplio o débilmente estacionario si cumple dos condiciones que son: su **Valor Esperado μ** es constante, y su **Función de Autocorrelación R** depende únicamente de la diferencia de tiempos, esto es:

- $\mu = E[X_{(t)}]$, permanece constante, es decir no depende del tiempo.
- $R(t, t + \tau) = E[X_{(t)}X_{(t+\tau)}] = R_{(\tau)} = R_{(-\tau)}; \forall t$, no depende del tiempo absoluto, depende de la separación de tiempo $\Delta t = t - t + \tau = \tau$.

3.3.3 La Función de Autocorrelación

Se define la Función de Autocorrelación $R(t, t + \tau)$, de un proceso X_t , como el momento conjunto de la variable X en un instante t y otro instante $t + \tau$. La Función de Autocorrelación o autocorrelación es una medida de la relación existente entre dos ejemplares temporales de un proceso estocástico:

$$R_{(t, t+\tau)} = E[X_{(t)}X_{(t+\tau)}] \quad [3.2]$$

La condición para que $X(t)$ sea débilmente estacionario, es que R no sea función del tiempo t , sino función de τ , o diferencia de tiempo $\Delta t = t - t + \tau$, esto en otras palabras es que no dependa del tiempo absoluto sino de la diferencia de tiempos que existe entre una muestra y la otra.

3.3.4 Autocovarianza

Se define la Autocovarianza $C(t, t + \tau)$, de un proceso X_t , como el momento conjunto:

$$C(t, t + \tau) = E[(X_t - \mu) - (X_{t+\tau} - \mu)]$$

La autocovarianza muestra, o representa, cuanto cambia el valor que toma la variable en relación a μ (que como vimos permanece constante en los procesos débilmente estacionarios), en el instante t , comparado con el valor que toma en otro instante que esta separado del primero en una cantidad de tiempo que llamamos τ . La autocovarianza C , representa la tendencia moverse en conjunto de la variable X en un instante t , y otro instante futuro $t + \tau$, esto es muy importante pues la covariación observada entre la variable en t y su futuro $t + \tau$, nos da información para interpretar, como esta tendencia a moverse conjuntamente, permite predecir su desenvolvimiento futuro.

Para interpretar mejor la importancia de estos conceptos se hace el siguiente análisis. Si calculamos la autocovarianza en un único instante de tiempo t , $C_{(t,t)} = E[(X_t - \mu) - (X_t - \mu)]$ desarrollamos el producto y llegamos a la siguiente expresión: $C_{(t,t)} = R_{(t,t)} - \mu^2$ Donde a $R_{(t,t)}$ lo llamamos $R_{(0)}$ y es la Función de Autocorrelación en el instante t cuando $\Delta t = \tau = 0$. Aquí entonces queda, $R_{(t,t)} = R_{(0)} = E[X_t X_t] = E[X_t^2]$ que es el valor cuadrático medio o segundo momento del proceso. Si a $R_{(t,t)}$ lo reemplazamos en la expresión de la autocovarianza para el instante de tiempo inicial, $\tau = 0$, vemos que la expresión $C_{(t,t)} = R_{(0)} - \mu^2 = E[X_t^2] - \mu^2$ se convierte en la varianza del proceso $X_{(t)} \Rightarrow C_{(t,t)} = C_{(0)} = Var[X_{(t)}]$, y como vemos es constante en relación al tiempo, lo cual es consistente con las condición de independencia con respecto al tiempo absoluto.

Como se dijo, si el valor medio o esperado se mantiene constante y la Función de Autocorrelación depende únicamente de la diferencia de tiempos, en este caso $\Delta t = \tau$, entonces la Autocovarianza del proceso estocástico también es función de τ , y su expresión es la siguiente:

$$C_{(t,t+\tau)} = E[(X_t - \mu) - (X_{t+\tau} - \mu)] = C_{(\tau)} = R_{(\tau)} - \mu^2 \quad [3.3]$$

Notar que si el valor medio del proceso es cero la Autocovarianza es igual a la Autocorrelación.

3.4 Efecto Memoria - Dependencias de Rango Corto y Rango Largo

Analizando que sucede con $C_{(\tau)}$ o bien $R_{(\tau)}$, cuando el valor de τ aumenta, surge una característica que resulta esencial para los procesos utilizados en el estudio del tráfico y es el tipo de dependencia o comportamiento de la variable con relación al intervalo de tiempo τ .

- Si $R_{(\tau)}$ tiende a cero rápidamente o en forma exponencial, cuando τ aumenta, decimos que el proceso no tiene memoria o tiene memoria breve llamada también dependencia de corto rango, o “**SRD**” por Short Range Dependence.
- Si $R_{(\tau)}$ tiende a cero en forma lenta o subexponencial cuando τ aumenta, decimos que el proceso tiene memoria larga, presentando dependencia aún para valores grandes de τ , llamamos a esto dependencia de rango largo, o “**LRD**” por Long Range Dependence.

Este efecto de “memoria” es un factor que incide en el desempeño de la red, por lo tanto es muy importante su detección y análisis. Para poner esto en contexto, me parece interesante comentar

el análisis que hace William Stallings [36] Pág.227, sobre el diseño de los buffers en los nodos de una red, supongamos entonces, dos situaciones distintas planteadas en el mismo escenario:

- a) Un nodo recibe un tráfico que responde a un modelo del tipo **Poisson** (análisis de cola tradicional), con una intensidad media de llegadas = λ , donde los tiempos entre llegadas, son independientes y exponencialmente distribuidos, de manera que el tiempo que toca esperar hasta ver la próxima llegada es independiente del instante en que se empiece a observar, esto se conoce como la falta de memoria de la distribución exponencial., y en este contexto, significa que los agrupamientos se producen a corto plazo, en una escala temporal característica (en el caso de Poisson el tiempo medio entre llegadas $1/\lambda$), supongamos del orden de los milisegundos, ahora si cambiamos dicha escala por otra mas grande por Ej. del orden de los segundos, y en forma sucesiva al orden de los minutos el agrupamiento tiende a desaparecer o dicho en otros términos los picos del tráfico se suavizan, (figura a) tal como lo predice un escalamiento con el modelo de Poisson, y en función de esto se dimensiona el buffer.

- b) Si en lugar de recibir un tráfico tipo Poisson, se recibe un tráfico con un comportamiento **LRD**, es decir tiene memoria y es **autosimilar**, aquí no hay escalas temporales características, dado que el comportamiento se repite a distintas escalas, esto significa que los agrupamientos tienden a persistir por el efecto de memoria, entonces a escalas mayores los agrupamientos persisten, tendiendo a crecer (los grupos se agrupan), los picos no se suavizan (figura b), y por ende los tamaños de las colas crecen, pudiendo incluso desbordar el buffer, este comportamiento evidentemente no se esperaría considerando un tráfico sin memoria, por ese motivo se afirma que los modelos sin memoria subestiman la demanda, en relación a los modelos que reproducen un comportamiento LRD.

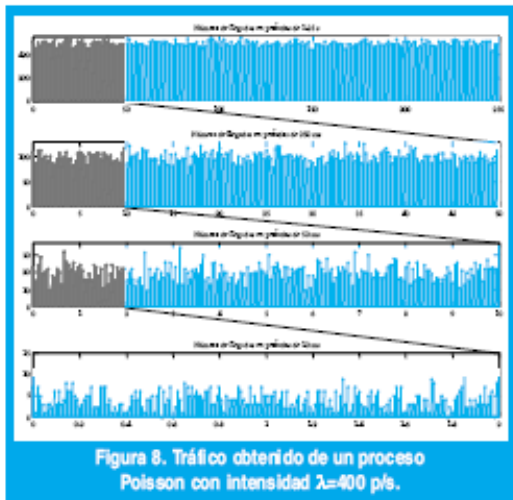


Figura a.

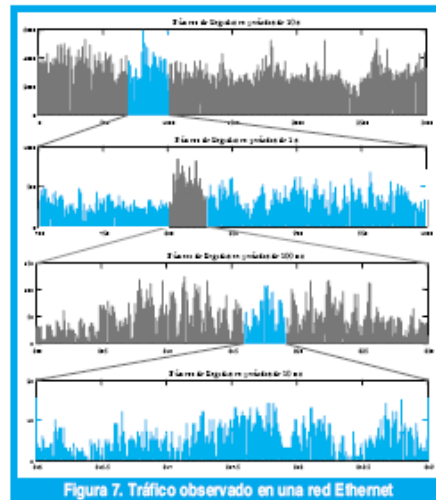


Figura b.

(Ver al pie)

En la situación planteada en b), la autosimilitud esta evidenciada por los agrupamientos que se repiten tanto en el orden de los milisegundos, como el de los segundos, o los minutos. Es importante tener presente que, si bien esto es consecuencia del efecto memoria, la autosimilitud es un concepto distinto del de LRD, el cual esta relacionado a la traslación temporal, (más adelante se aclara esta diferencia).

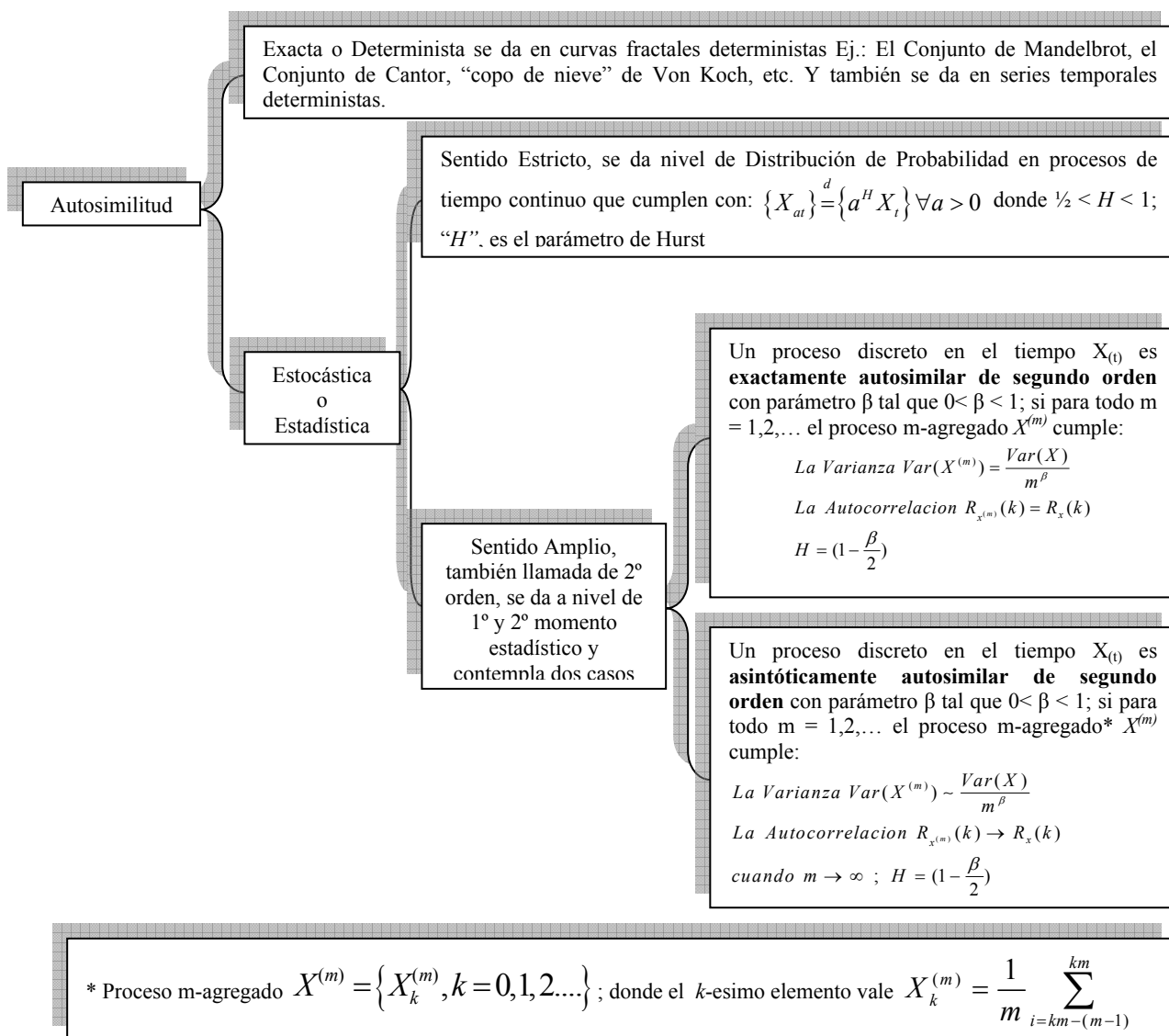
En general, para el estudio del desempeño de las redes, es importante analizar el comportamiento del tráfico en diferentes escalas de tiempo, pues acciones tales como el buffering, o la asignación de ancho de banda a diferentes flujos de paquetes, pueden considerarse como un proceso de

* Figuras extraídas de [24] - M. Alzate y A. Monroy “Introducción al Tráfico Autosimilar”. Revista Ingeniería Vol. 6 No. 2 2001 Páginas: 6 –17. Universidad Distrital Francisco José de Caldas.

cambio de escala temporal. En el ejemplo comentado, claramente se advierte que un proceso sin memoria (SRD), pierde rápidamente la dependencia al reescalarlo en el tiempo, y pasa a comportarse como un conjunto de variables aleatorias independientes. Como contrapartida, un proceso autosimilar es invariante ante la reescalización, su comportamiento es persistente, pues se preserva su estructura de autocorrelación, aún en sucesos con una separación en el tiempo relativamente grande. La preservación de una magnitud medible u observable ante cambios de escala, queda señalada por la autosimilitud. A continuación se define de manera formal la autosimilitud y sus distintos tipos.

3.5 Autosimilitud

Se dice que el fenómeno de la autosimilitud se manifiesta, cuando al observar la representación de un proceso, con distintos grados de ampliación o escalas de una cierta dimensión, se puede establecer la existencia de patrones de forma, o comportamiento, idénticos o similares, los cuales se repiten en las sucesivas ampliaciones o cambios de escala. En el caso del tráfico, la dimensión que se observa a distintas escalas, es el tiempo. Existen diversos tipos de autosimilitud en los procesos que presentan este fenómeno y se resumen en la siguiente figura:



A continuación se procede al desarrollo de las definiciones formales, para los distintos tipos de autosimilitud con su correspondiente análisis.

3.5.1 Autosimilitud exacta o determinista

Este tipo de autosimilitud se manifiesta en las curvas fractales deterministas, como el Conjunto de Cantor, la curva de Von Koch, etc. Estas curvas se generan a partir de procesos recursivos como por ejemplo el conjunto de Mandelbrot $\{c = c + c^2\}$. También, este tipo de autosimilitud, se manifiesta en series temporales deterministas.

3.5.2 Autosimilitud estocástica o estadística

Este tipo de autosimilitud se manifiesta, si el fenómeno se presenta a nivel de las propiedades estadísticas del proceso estudiado, es decir las estadísticas no cambian cuando cambia la escala temporal. La Autosimilitud Estadística se subdivide a su vez en dos tipos, que son la autosimilitud estadística en sentido estricto, y la autosimilitud estadística en sentido amplio. Los procesos que presentan este tipo de autosimilitud son los que resultan de interés para modelar el tráfico autosimilar.

3.5.2.1 Definición de Autosimilitud Estadística de Tiempo Continuo

La autosimilitud estadística en el sentido estricto para tiempo continuo, se define de la siguiente forma:

Sea un proceso $X_{(t)}, t \in \mathbb{R}$, un factor escalar $a > 0 \in \mathbb{R}$, y un parámetro H tal que $0.5 < H < 1$ y además se cumple para cualquier sucesión temporal de $X_{(t)}$ lo siguiente:

$$\{X_{at_1}, X_{at_2}, \dots, X_{at_n}\} \stackrel{d}{=} \{a^H X_{t_1}, a^H X_{t_2}, \dots, a^H X_{t_n}\} \quad [3.4]$$

Se dice que es autosimilar con parámetro H , o H_{SS} , en el sentido estricto, donde el operador $\stackrel{d}{=}$ significa que ambas secuencias tienen la misma distribución de probabilidad o lo que es lo mismo son indistinguibles desde el punto de vista estadístico. “ H ”, es conocido como parámetro de Hurst¹², y representa una medida de la persistencia del comportamiento del proceso, siendo mayor la manifestación de la misma, cuanto más se acerca al valor 1, y por el contrario, cuando H es igual a 0.5, no hay persistencia. Más adelante se analiza detenidamente este parámetro.

3.5.2.2 Definición Autosimilitud Estadística de tiempo discreto

Si relajamos la condición de equivalencia a nivel de Distribución de Probabilidad y establecemos la equivalencia a nivel de las estadísticas de 2º orden, como ser la Función de Autocorrelación R , y varianza $Var[X]$, de la secuencia temporal que representa el proceso estocástico, decimos que estamos en presencia de autosimilitud en el sentido amplio. Otra forma de llamarla es autosimilitud de segundo orden.

En el análisis de tráfico, se tratan procesos estocásticos definidos en puntos discretos del tiempo, entonces a partir de procesos estocásticos $X_{(t)}, t \in \mathbb{Z}$, podemos definir procesos agregados de la siguiente forma:

Sea un proceso estocástico $X_{(t)}, t \in \mathbb{Z}$; donde para una sucesión temporal estacionaria X se define la serie temporal m-agregada:

$$X^{(m)} = \{X_k^{(m)}, k = 0, 1, 2, \dots\}$$

¹² En el capítulo 5 punto 5.1.2, se hace una breve reseña del estudio que llevo a cabo el ingeniero Hurst para analizar las corrientes del río Nilo, el cual lo llevo a desarrollar un método estadístico llamado análisis R/S (análisis de rango reescalado) publicado en 1965, y desde entonces es frecuentemente utilizado en análisis de series temporales.

Cada elemento de la serie m-agregada es la suma de elementos de la serie temporal original en bloques adyacentes y no superpuestos de tamaño m tal que:

$$X_k^{(m)} = \frac{1}{m} \sum_{i=km-(m-1)}^{km} X_i$$

Para ejemplificar esta expresión, veamos como sería el desarrollo para una serie aleatoria ficticia con 12 elementos

Para m = 1

1 9 2 6 5 4 7 3 2 10 7 5

Para m = 2 tenemos la siguiente serie:

El primer elemento es $\frac{9+1}{2} = 5$; el segundo, $\frac{2+6}{2} = 4$ y así sucesivamente hasta el último elemento, quedando como resultado la serie:

5 4 4.5 5 6 6

Luego m toma el valor 3 y se repite el procedimiento

El primer elemento $\frac{1+9+2}{3} = 4$ el segundo $\frac{6+5+9}{3} = 5$, y así sucesivamente hasta obtener:

4 5 4 7.33

Como se puede ver, el máximo valor que puede tomar m, es la cantidad de elementos que tenga la serie original en cuyo caso el resultado es un solo número que es el promedio aritmético de la serie. Esta operación entonces, equivale a hacer un muestreo con un tamaño de muestra “m”, sobre la serie temporal original.

De esta forma se obtienen las series temporales agregadas como un procedimiento para comprimir la escala temporal, se ve entonces que para m = 1 tenemos la máxima resolución posible, para un valor de m = 2 tenemos una reducción de cantidad de elementos de la serie, por un factor 2 dado que se promedian dos puntos sucesivos, si el factor fuera 3 entonces el promedio es de tres puntos sucesivos, de esta forma la escala temporal se comprime, pero se pierden los detalles debido al promedio.

En este contexto, se pueden analizar las estadísticas de los procesos m-agregados $\{X^{(m)}\}$, y brindar las siguientes definiciones de autosimilitud.

3.5.2.2.1 Autosimilitud exacta de segundo orden

Un proceso estocástico X es **estrictamente autosimilar de segundo orden** con parámetro β , tal que $0 < \beta < 1$ si se cumple que $\forall m \in \mathbb{N} : m > 0$:

- La Varianza: $Var(X^{(m)}) = \frac{Var(X)}{m^\beta}$
- La Función de Autocorrelación: $R_{X^{(m)}}(\tau) = R_X(\tau)$

El parámetro β , está relacionado con el parámetro de Hurst de la siguiente forma:

$$H = 1 - \left(\frac{\beta}{2}\right); \text{ Por lo tanto, } \beta = (2 - 2H)$$

3.5.2.2.2 Autosimilitud asintótica de segundo orden

Si se considera una condición más débil, se dice que es un proceso **asintóticamente autosimilar de segundo orden** si se cumple que para todo τ lo suficientemente grande:

- La Varianza: $V a r (X^{(m)}) \sim \frac{V a r (X)}{m^\beta}$ Donde el símbolo \sim indica que son asintóticamente similares.
- La Función de Autocorrelación: $R_{x^{(m)}}(\tau) \rightarrow R_x(\tau)$, cuando $m \rightarrow \infty$.

3.6 Coeficiente de Correlación de un Proceso Estocástico

Una definición de autosimilitud de segundo orden, también puede ser dada en términos del **Coeficiente de Correlación**; mas adelante veremos que esta definición también puede ser dada en el dominio de las frecuencias, con la denominada **Densidad Espectral del Proceso**.

Sea un proceso débilmente estacionario X y sus correspondientes m-agregados $X^{(m)}$, recordar que si el proceso es débilmente estacionario, el valor medio es constante, y la Función de Autocorrelación depende solo de $\Delta t = \tau$, por lo tanto su coeficiente de correlación será:

$$\rho = \frac{E[(X_{(t)} - \mu)(X_{(t+\tau)} - \mu)]}{\sigma_{(t)}\sigma_{(t+\tau)}} = \frac{C_{(\tau)}}{\sigma_{(t)}\sigma_{(t+\tau)}} = \frac{\text{Autocovarianza}}{\sigma_{(t)}\sigma_{(t+\tau)}} \Rightarrow \boxed{\rho = \frac{R(\tau) - \mu^2}{\sigma_{(t)}\sigma_{(t+\tau)}}} \quad [3.5]$$

Se puede considerar al coeficiente de correlación como una medida de dependencia lineal normalizada para ser relativa a la cantidad de variabilidad del proceso en un intervalo de tiempo τ . Notar también, que si $\sigma_{(t)} = \sigma_{(t+\tau)}$; el denominador de la expresión, es la varianza $\sigma^2(t)$ del proceso.

Dos propiedades importantes de destacar es que el coeficiente de correlación:

1. ρ está acotado: $-1 \leq \rho \leq 1$.
 2. $\rho = 0$, cuando la variable aleatoria X es independiente.
- Se dice que un proceso es **estrictamente autosimilar de segundo orden**¹³, con parámetro H : $\frac{1}{2} < H < 1$, si para todo $m \in \mathbb{Z} > 1$, cumple que $\rho_\tau^m = \rho_\tau$, esto significa que el proceso X y los procesos m-agregados X^m , tienen idéntica estructura de correlación.
 - Se dice que un proceso es **asintóticamente autosimilar de segundo orden**, con parámetro H : $\frac{1}{2} < H < 1$, si para todo $m \in \mathbb{Z} > 1$, si se cumple la condición $\rho_\tau^m \rightarrow \rho_\tau$ Cuando $m \rightarrow \infty$ para todo τ lo suficientemente grande, donde la estructura de correlación es asintóticamente similar.

Aquí se debe tener en cuenta como lo aclara William Stallings [36] Pág. 234, que a veces las propiedades, tanto de los procesos autosimilares, como los procesos con dependencia de largo

¹³ También se la conoce como autosimilitud exacta de segundo orden.

alcance, tienen su expresión mas clara en términos de la Función de Autocorrelación $R_{(\tau)}$, mientras que en otras se expresan mejor en términos de la autocovarianza $C_{(\tau)} = R_{(\tau)} - \mu^2$, donde $\mu = E[X_{(t)}]$, o inclusive como lo estamos viendo aquí en términos de su coeficiente de correlación $\rho_{(\tau)}$ y hay que ser cuidadosos con las lecturas, pues en la literatura relativa al tráfico, a veces pueden aparecer con el mismo nombre la Autocorrelación y la Autocovarianza, o también llaman Autocorrelación al Coeficiente de Correlación o Autocovarianza Normalizada, pudiendo esto prestarse a confusión.

3.6.1 Relación entre Correlación, Autosimilitud y la Dependencia de Rango Largo

Como se vio en el punto anterior, el coeficiente de correlación de un proceso estocástico, aparte de poder definir el comportamiento autosimilar del proceso, relaciona también magnitudes como la varianza del proceso y su autocorrelación. Una conclusión muy importante que surge al analizar el comportamiento de la varianza de un proceso estocástico y sus “m-agregados”, al variar β entre cero y uno, o H entre $\frac{1}{2}$ y uno, es el tipo de proceso, dado que puede ser del tipo aleatorio sin correlación, o caso contrario presentar una cierta estructura de correlación resultando ser del tipo autosimilar. A continuación se realiza un análisis de estos comportamientos.

- Si el proceso es estacionario y ergódico¹⁴ (variable independiente) el valor $\beta = 1$, (o bien $H = \frac{1}{2}$), la varianza de la media temporal decrece hasta cero con una pendiente $\frac{1}{m}$ y no presenta LRD. Esto es consistente con la condición de ergodicidad donde la media del conjunto del proceso estocástico coincide con la media temporal de una realización del proceso, y una propiedad estadística de este tipo de procesos es precisamente que la varianza de la media de la muestra de tamaño m , es igual a la varianza de la variable subyacente dividido m . Aquí $X^{(m)}$ es estadísticamente considerada una media temporal con un tamaño de muestra m , por lo tanto es lógico afirmar que: $Var[X^{(m)}] = \frac{Var[X]}{m}$; y al ser la variable independiente, $\rho = 0$, se cumple $R_{(X)} = 0$, entonces $R_{(X^{(m)})} \rightarrow 0$, en forma rápida o exponencial, cuando $m \rightarrow \infty$, entonces aquí no hay efecto memoria, o estamos en presencia de SRD.
- Si el proceso es autosimilar con $\frac{1}{2} < H < 1 \Rightarrow 0 < \beta < 1$, la varianza decrece en forma más lenta que $\frac{1}{m}$, En realidad decrece en forma proporcional a $\frac{1}{m^\beta}$, o su forma equivalente $\frac{1}{m^{(2-2H)}}$, la variable no es independiente, $\rho \neq 0$, por ende, $R_{(X^{(m)})} \rightarrow R_{(X)} \neq 0$, cuando $m \rightarrow \infty$, el comportamiento de la Función de Autocorrelación es asintótico, entonces aquí se presenta el efecto memoria y estamos en presencia de LRD. Por este motivo también, se toma como equivalentes a la Autosimilitud y la LRD cuando “H” toma valores en el intervalo abierto $\frac{1}{2} < H < 1$.

3.7 Densidad Espectral

Las definiciones de LRD que se vieron hasta ahora, están dadas en el dominio del tiempo, otra definición que resulta de utilidad en los análisis es la que se da en el dominio de las frecuencias y permite observar el comportamiento del proceso mediante sus componentes frecuenciales con una expresión denominada **Densidad Espectral** o **Espectro de Potencia** del proceso.

¹⁴ Ver apéndice 2 – Procesos Estocásticos; punto 10.2.

Se define al Espectro de Potencia o Densidad Espectral de un proceso aleatorio **estacionario**¹⁵ $X(t)$, como la transformada de Fourier de su Función de Autocorrelación:

$$S(\omega) = \int_{-\infty}^{\infty} R(\tau) e^{-j\omega\tau} d\tau \quad [3.6]$$

En esta expresión ω es la frecuencia en radianes, y $j = \sqrt{-1}$. La densidad espectral representa la densidad media de potencia de las componentes frecuenciales en torno de ω .

La aplicabilidad de la transformada de Fourier, a una secuencia finita como es una serie temporal, plantea un verdadero reto, dado que la transformada esta definida para funciones que existen y están definidas entre $-\infty$ y ∞ , por ende hay que asumir ciertas condiciones como ser la ergodicidad y estacionariedad del proceso, de ahí la necesidad de introducir la Función de Autocorrelación, resultando el producto interno de la variable consigo misma, desplazada en un valor temporal que llamamos τ , y la suposición que el proceso es estacionario en ese intervalo, no siempre se ajusta exactamente a la realidad.

La Función de Autocorrelación $R(\tau)$, puede expresarse como la transformada inversa del espectro:

$$R(\tau) = \frac{1}{2\pi} \int_{-\infty}^{\infty} S(\omega) e^{j\omega\tau} d\omega \quad [3.7]$$

Si consideramos $\tau = 0 \Rightarrow R_{(0)} = \frac{1}{2\pi} \int_{-\infty}^{\infty} S(\omega) d\omega = E[|X_t|^2]$ esto esta justificado, pues si recordamos la Función de Autocorrelación $R_{(t,t)} = R_{(0)} = E[X_t X_t] = E[X_t^2]$. Por ende el área total bajo $\frac{S(\omega)}{2\pi}$ es igual a la potencia media del proceso $X_{(t)}$.

La componente de continua del espectro de potencia del proceso esta representada por $S(0) \Rightarrow \omega = 0$, entonces la expresión del espectro queda:

$$S(0) = \int_{-\infty}^{\infty} R(\tau) d\tau \quad [3.8]$$

Si el proceso $X(t)$ es real, su espectro $S(\omega)$, será real es decir no tendrá parte imaginaria, y es la razón por la que interesa abordar el estudio de la estructura frecuencial de los procesos reales $X(t)$ a través de su espectro y no de su transformada de Fourier, dado que esta última es siempre una función compleja más difícil de manejar e interpretar.

La integral $\int_{-\infty}^{\infty} R(\tau) d\tau$ será finita, si $R(\tau) \rightarrow 0$ con suficiente rapidez para que la suma sea finita cuando $\tau \rightarrow \infty$; lo cual es consistente con un proceso independiente o sin memoria.

Efectuado este análisis (que como se vera mas adelante es muy importante), se procede a la definición de LRD en términos de la frecuencia.

¹⁵ Resalto aquí la palabra estacionario pues es una condición esencial para que sea válida la transformada de Fourier, y precisamente es uno de los puntos flojos en el análisis con datos reales, pues no podemos afirmar categóricamente que la serie se ajuste a un proceso estocástico estacionario a lo largo del eje $-\infty, \infty$ y se asume la ergodicidad del proceso como condición necesaria.

3.7.1 Relación entre la Densidad Espectral y Dependencia de Rango Largo

Un proceso estocástico presenta **LRD**, si su **Densidad Espectral** obedece a una ley de potencia en cercanías del origen, esto es:

$$S(\omega) \sim \frac{1}{|\omega|^\gamma}, \text{ cuando } \omega \rightarrow 0; \text{ siendo } \gamma = 1 - \beta = 2H - 1$$

Aquí, de manera análoga al análisis del punto [3.6.1](#), donde se muestra la relación entre la correlación autosimilitud y LRD, hacemos el análisis para β y H , tomado valores en los intervalos $0 < \beta < 1$, o bien $0.5 < H < 1$.

- Si $\beta = 1$ o lo que es lo mismo $H = 0,5$; el valor de γ es cero y por lo tanto el proceso no tiene memoria y la densidad espectral sigue siendo finita cuando $\omega \rightarrow 0$.
- Si cuando $\omega \rightarrow 0$, el valor de $\gamma > 0$ esto implica que H toma valores > 0.5 ; o β toma valores entre 0 y 1; entonces el espectro de potencia del proceso presenta una divergencia en cercanías del origen indicando que el proceso presenta LRD. Una interpretación física de esto, es que hay una importante contribución de las componentes de baja frecuencia del espectro.

Una interesante observación es que si el proceso es autosimilar con $0.5 < H < 1$, entonces también presenta un comportamiento LRD, por eso hay una doble implicancia entre estos conceptos en el intervalo abierto $0.5 < H < 1$.

3.8 Distribuciones de Cola Hiperbólica

Para completar el panorama de los procesos autosimilares, se expone otro concepto relacionado con la autosimilitud y es el de las distribuciones de cola pesada (heavy-tailed) o cola hiperbólica. Estas distribuciones como su nombre lo indica, son de lento decaimiento y cuando se utilizan para caracterizar densidades de probabilidad como los tiempos de llegada de paquetes o longitudes de ráfagas, dan lugar a procesos que presentan LRD y por ende autosimilitud. Este comportamiento no se da en los modelos de tráfico tipo Poisson, porque los tiempos de llegada, se modelan con distribuciones exponenciales que dan lugar a un comportamiento SRD, o procesos sin memoria.

En general una variable aleatoria distribución de cola hiperbólica, tiene una varianza infinita y posiblemente una media infinita, dado que estas distribuciones contienen valores muy grandes con una probabilidad no despreciable, o dicho en otras palabras la posibilidad de que ocurran grandes apartamientos de un comportamiento, no es despreciable, esto esta en oposición a los presupuestos de una distribución exponencial donde la probabilidad de un gran apartamiento del valor medio, es despreciable. Uno de los atractivos de estas distribuciones es que dan lugar a modelos de simulación manejables y su estudio en detalle excede el alcance de este trabajo, pero para quienes estén interesados en la profundización del estudio de modelos con estas distribuciones, sugiero la lectura de las siguientes referencias bibliográficas: [2], [8], [12], [20], [21] y [40].

Se dice que la distribución de una variable aleatoria X tiene cola hiperbólica si:

$$1 - F(x) = \Pr[X > x] \sim \frac{1}{x^\alpha}; \text{ cuando } x \rightarrow \infty, 0 < \alpha$$

Una distribución que cumple con esta condición y es muy utilizada en el modelado y estudio del tráfico es la de **Distribución de Pareto**, con parámetros k y α con $\alpha < 0$, cuyas funciones de densidad y distribución respectivamente son:

$$f(x) = F(x) = 0 \quad (x \leq 0)$$

$$f(x) = \frac{\alpha}{k} \left(\frac{k}{x}\right)^{\alpha+1} ; F(x) = 1 - \left(\frac{k}{x}\right)^{\alpha} \quad (x > k; \alpha > 0)$$

- El valor medio de la distribución de Pareto está dado por: $E[X] = \frac{\alpha}{\alpha-1}k$ ($\alpha > 1$)
- La Varianza esta dada por: $Var[X] = \frac{\alpha}{(\alpha-1)^2(\alpha-2)}k^2$

El parámetro k especifica el valor mínimo que puede tomar la variable aleatoria, el parámetro α determina la media y la varianza de la variable aleatoria X .

- Si $\alpha \leq 2$ entonces la distribución tiene varianza infinita.
- Si $\alpha \leq 1$ entonces tiene media y varianza infinitas.

La distribución de Pareto es consistente con una amplia gama de fenómenos de la naturaleza y las ciencias sociales como así también en el ámbito de las comunicaciones, algunos estudios¹⁶, señalan que la cola hiperbólica de ciertas variables como por ejemplo, el tamaño de los archivos y la duración de las conexiones, son la causa subyacente de la LRD y la autosimilitud.

Otras distribuciones de cola hiperbólica también utilizadas para caracterizar los procesos, son la distribución Log-normal, y la distribución de Weibull.

3.9 Relación entre la Autosimilitud, la Dependencia de Rango Largo y la Distribución de Colas Hiperbólicas

La dependencia a largo plazo “LRD”, es una característica relacionada con el comportamiento de la correlación de un proceso estocástico, en particular con su invariabilidad frente a las traslaciones temporales.

La Autosimilitud es un concepto relacionado a la invariabilidad o persistencia de las características estadísticas de un proceso estocástico, frente a los cambios de escala temporales.

Como la LRD tiene que ver con los desplazamientos temporales, y la Autosimilitud tiene que ver con la escala temporal, entonces son conceptos diferentes, pero como se mostró en [3.6.1](#) y [3.7.1](#) la restricción impuesta por los valores del parámetro de Hurst $\frac{1}{2} < H < 1$ o su equivalente

$0 < \beta < 1$, torna equivalentes la LRD y la Autosimilitud, pues cuando se aplica esta restricción, existe una doble implicancia entre estos dos conceptos y se los utiliza con el mismo significado en la literatura relativa al tráfico.

Además vimos que si una variable aleatoria que representa un proceso estocástico, tiene una distribución de cola hiperbólica, esto conduce a un comportamiento LRD, y por ende también presentará características autosimilares, entonces podemos establecer una cadena de implicancias y decir que, con determinadas restricciones en su formulación, se cumple lo siguiente:

Distribución de Cola Hiperbólica \Leftrightarrow Dependencia de Largo Plazo \Leftrightarrow Autosimilitud

¹⁶ Vern Paxson and Sally Floyd, [Wide-Area Traffic: The Failure of Poisson Modeling](#). IEEE/ACM Transactions on Networking, Vol. 3 No. 3, pp. 226-244, June 1995.

3.10 Manifestación de la Autosimilitud en los Procesos Estocásticos

Como conclusión de las características y propiedades de los procesos autosimilares que hemos visto hasta aquí, se puede afirmar como lo dicen W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson en su trabajo [1] Pág. 5, el fenómeno de la autosimilitud se manifiesta en distintas formas, todas ellas equivalentes y son las siguientes:

- La varianza de las series m-agregadas decae más lentamente que la inversa del tamaño de la muestra m .
- Decaimiento lento de la varianza (Función de Autocorrelación asintótica).
- El espectro de potencia del proceso presenta una variación subexponencial u obedece a una ley de potencias en cercanías del origen.

Entonces dependiendo del contexto, se aprovechan estas características para efectuar el análisis correspondiente ya sea en el dominio del tiempo o en el de las frecuencias.

3.11 Procesos Autosimilares Utilizados para Modelar Tráfico

3.11.1 Procesos de Incrementos Independientes

Un proceso estocástico $\{X(t), 0 \leq t < \infty\}$, tendrá incrementos independientes si $X(0) = 0$, y además para cualquier selección de índices $t_0 < t_1 < t_2 < \dots < t_n$, se cumple que las n variables aleatorias definidas como $\{X(t_1) - X(t_0), X(t_2) - X(t_1), \dots, X(t_n) - X(t_{n-1})\}$, son independientes. El significado de esto es que la cantidad de movimiento que tenga el proceso en un intervalo de tiempo es independiente del movimiento que tenga en cualquier otro intervalo que no se solape con el primero. Además, estos incrementos independientes serán estacionarios si se cumple que para cualquier constante $a > 0$, y para cualquier selección de índices donde $t_2 > t_1$, la variable definida como $X(t_2 + a) - X(t_1 + a)$, tiene la misma distribución que $X(t_2) - X(t_1)$.

Existen dos propiedades de los procesos estocásticos con incrementos independientes estacionarios que son importantes y expresan:

1. Si $E[X(t)] = \mu(t)$, es una función continua del tiempo, entonces $\mu(t) = a + bt$ donde a y b son constantes.
2. Si $Var[X(t) - X(0)]$, es una función continua del tiempo, entonces se cumple que para todo s la $Var[X(s+t) - X(s)] = \sigma^2 t$, donde σ^2 es una constante.

También resulta útil tener presente las siguientes propiedades que se cumplen para cualquier constante $a \in \mathbb{R}$:

$$E[aX] = aE[X] \quad [3.9]$$

$$Var[aX] = a^2 Var[X] \quad [3.10]$$

Dos procesos autosimilares extensamente utilizados en el estudio de series temporales y también en el tráfico autosimilar, y de cuyo análisis se pueden extraer importantes conclusiones; son el Movimiento Browniano y el Movimiento Browniano Fraccional, siendo este último particularmente interesante pues al incrementar un proceso de Movimiento Browniano Fraccional (FBM), surge el proceso conocido como Ruido Gaussiano Fraccional (FGN), que tiene amplia utilización en el modelado de tráfico autosimilar, pues el FGN es un proceso que presenta LRD, cuando H toma valores en el intervalo abierto $\frac{1}{2} < H < 1$.

3.11.2 Proceso de Movimiento Browniano

“El movimiento browniano, es un movimiento aleatorio que poseen las partículas microscópicas suspendidas en un líquido o gas, y esta causado por la colisión con las moléculas del medio que las rodea. Este fenómeno físico constituye la base de la definición del proceso estocástico de movimiento browniano, que también recibe el nombre de proceso de Wiener, y de proceso de Wiener-Levy.”¹⁷. El movimiento browniano representa un papel fundamental en el estudio de los procesos estocásticos.

Se define un proceso BM (movimiento browniano), que llamamos $B(t)$ como aquel que satisface lo siguiente:

- $\{B(t), 0 \leq t < \infty\}$ Tiene incrementos independientes estacionarios.
- $\forall t > 0$, la variable aleatoria $B(t)$ tiene distribución normal.
- $\forall t > 0, E[B(t)] = 0$.
- $B(0) = 0$

Viendo la definición de autosimilitud estricta, o exacta, para tiempo continuo, dada en el punto [3.5.2.1](#), ecuación [3.4]. Tenemos que $\{X(at)\} \stackrel{d}{=} \{a^H X(t)\}$ lo cual significa que los dos procesos son indistinguibles estadísticamente por lo tanto si a^H es una constante, se cumple $\{X(t)\} \stackrel{d}{=} \left\{ \frac{X(at)}{a^H} \right\}$. Entonces la autosimilitud se puede verificar mediante las siguientes condiciones:

$1. E[X(t)] = \frac{E[X(at)]}{a^H} \quad \text{Media}$	El parámetro de Hurst H , es una medida de la persistencia de un fenómeno estadístico, y también es una medida de la longitud de dependencia en un proceso estocástico. Cuando $H = 0,5$, indica la ausencia de LRD, y para valores de $0.5 < H < 1$, mayor será el grado de persistencia de la misma, cuanto mas se aproxime H a 1.
$2. Var[X(t)] = \frac{Var[X(at)]}{a^{2H}} \quad \text{Varianza}$	
$3. R_x(t, s) = \frac{R_x(at, as)}{a^{2H}} \quad \text{Autocorrelación}$	

[3.11]

3.11.2.1 Análisis de la Autosimilitud del Movimiento Browniano

Por definición sabemos que la densidad de probabilidad de un proceso de movimiento browniano tiene la forma:

$$f_B(x, t) = \frac{1}{\sigma\sqrt{2\pi t}} e^{-x^2/2\sigma^2 t}$$

A partir de esto se tiene:

- $Var[B(t)] = t$
- $Var[B(t) - B(s)] = |t - s|$ Se asume que tiene incrementos estacionarios.

La autocorrelación de un proceso de movimiento browniano, la podemos desarrollar de la siguiente manera:

¹⁷ Cita del libro de William Stallings [36] Pág. 176.

Supongamos una sucesión de tiempos $t_0 < t_1 < t_2$, sabemos que la autocorrelación es el momento $E[(B(t_2) - B(t_1))(B(t_1) - B(t_0))]$, los intervalos $(t_2, t_1), (t_1, t_0)$ no están superpuestos, por lo tanto de acuerdo con la definición del proceso, $(B(t_2) - B(t_1))$ y $(B(t_1) - B(t_0))$ son independientes y $\forall t > 0, E[B(t)] = 0; B(0) = 0$, entonces teniendo en cuenta la propiedad estadística para dos variables aleatorias independientes X e $Y : E[XY] = E[X]E[Y]$, desarrollamos y tenemos:

$$\begin{aligned} 0 &= E[(Bt_2 - Bt_1)(Bt_1 - B(0))] \\ 0 &= E[(Bt_2 - Bt_1)B(t_1)] \\ 0 &= E[(Bt_2Bt_1)] - E[B^2(t_1)] \\ 0 &= E[(Bt_2Bt_1)] - Var[B(t_1)] \text{ (recordar que } \mu = 0) \\ 0 &= E[(Bt_2Bt_1)] - t_1 \Rightarrow E[(Bt_2Bt_1)] = t_1 \end{aligned}$$

Por lo tanto tenemos que la autocorrelación $R_B(t_1, t_2) = E[(Bt_2Bt_1)] = t_1$; donde $t_1 < t_2$, esto lo podemos generalizar como $R_B(t, s) = \min[t, s]$, y como sabemos que $B(t)$ tiene media nula, entonces la autocorrelación es igual a la autocovarianza (recordar la ecuación [3.3]).

Ahora que tenemos la expresión de la autocorrelación para $B(t)$, podemos verificar si es autosimilar para $H = 0.5$ examinando las tres condiciones:

1. Por definición $E[B(t)] = 0$, por lo tanto $E[B(at)] = 0$, donde se va a cumplir que $\frac{E[B(at)]}{a^{0.5}} = 0$ para todo t , y por lo tanto se cumple la 1ª condición.
2. En la ecuación [3.10] del punto 8 vimos $Var[aX] = a^2Var[X]$, además sabemos que $Var[B(t)] = t \therefore Var[B(at)] = at \Rightarrow Var[B(t)] = \frac{Var[B(at)]}{a^{2H}}$, con $H = 0.5$ Vemos que se cumple la 2ª condición.
3. En la ecuación [3.9] del punto 8 vimos que $E[aX] = aE[X]$, por lo tanto si tenemos que, $R_B(t, s) = \min[t, s] \Rightarrow R_B(at, as) = \min[at, as] = a \min[t, s] = aR_B(t, s)$, por lo tanto vemos que se cumple la 3ª condición.

Entonces estamos en condiciones de afirmar que el proceso de movimiento browniano es estrictamente autosimilar H_{SS} , con parámetro $H = 0.5$.

Para cualquier $t \geq 0$ y $\delta > 0$, el incremento de un proceso de movimiento browniano $B(t + \delta) - B(t)$, tiene una distribución normal, con media cero, y varianza δ :

$$\Pr[(B(t + \delta) - B(t)) \leq x] = \frac{1}{\sqrt{2\pi\delta}} \int_{-\infty}^x e^{-y^2/2\delta} dy \quad [3.12]$$

Esta distribución es independiente de t , y depende solamente de δ , y esto es consistente con que $B(t)$, tenga incrementos independientes y estacionarios.

3.11.3 Proceso de Movimiento Browniano Fraccionario

Es conocido que las muestras de tráfico tomadas en ciertas redes, presentan algún grado de dependencia, si queremos modelar adecuadamente esto, los incrementos del proceso que representa el tráfico, deben reflejar esa dependencia, o dicho de otra forma tener alguna correlación distinta de cero, y además el proceso debe ser estacionario, o por lo menos débilmente estacionario. El proceso de Movimiento Browniano, a pesar de tener una importancia teórica central, para muchos propósitos es demasiado restrictivo y además tiene incrementos independientes, por lo cual no cumple con el primer requisito. El proceso de Movimiento Browniano Fraccionario (FBM), cumple con ambos requisitos, y deriva del movimiento browniano, relajando el requisito de incrementos independientes.

Se define un proceso FBM (movimiento browniano fraccional), que llamamos $B_H(t)$, con parámetro H tal que $0 < H < 1$, y que cumple con lo siguiente:

- $B_H(t)$ es continuo.
- $B_H(0) = 0$.
- $\forall t \geq 0$ y $\delta > 0$ el incremento $B_H(t + \delta) - B_H(t)$ tiene una distribución normal con media igual a cero, y una varianza igual a δ^{2H} .

Entonces:

$$\Pr[(B_H(t + \delta) - B_H(t)) \leq x] = \frac{1}{\sqrt{2\pi\delta^{2H}}} \int_{-\infty}^x e^{-y^2/2\delta^{2H}} dy \quad [3.13]$$

Como se puede ver, la ecuación [3.13], se convierte en la [3.12], si consideramos $H = 0.5$ el proceso se convierte en un proceso de movimiento browniano.

3.11.3.1 Análisis de la Autosimilitud del Movimiento Browniano Fraccionario

Del mismo modo que se procedió con el BM (movimiento browniano), se analizará si el FBM, satisface las tres condiciones de un proceso autosimilar enunciadas en [3.11].

1. En la ecuación [3.13], si utilizamos un intervalo que comience con $B_H(0) = 0$, veremos que: $(B_H(0+t) - B_H(0)) = B_H(t)$ tiene una distribución normal con media igual a cero y varianza t^{2H} . Por ende tenemos $E[B_H(t)] = 0$ y también $E[B_H(at)] = 0$ entonces de la forma trivial se cumple que $E[B_H(t)] = E[B_H(at)] / a^H$ satisfaciendo la condición 1.
2. $Var[B_H(t)] = t^{2H} \Leftrightarrow Var[B_H(at)] = (at)^{2H} = a^{2H} Var[B_H(t)]$ Recordar la ecuación [3.10], vemos entonces que se cumple la condición 2.
3. En el análisis de la tercera condición, referida a la Función de Autocorrelación, es donde nos encontramos con importantes conclusiones, como se muestra a continuación:

Partimos del módulo del incremento $(B_H(t) - B_H(s))^2$ y desarrollamos el cuadrado del binomio:

$$(B_H(t) - B_H(s))^2 = B_H^2(t) + B_H^2(s) - 2B_H(t)B_H(s)$$

Donde agrupamos de la siguiente forma:

$$B_H(t)B_H(s) = \frac{1}{2} \left(B_H^2(t) + B_H^2(s) - (B_H(t) - B_H(s))^2 \right)$$

Si aplicamos el momento conjunto

$$E[B_H(t)B_H(s)] = \frac{1}{2} \left(E[B_H^2(t)] + E[B_H^2(s)] - E[(B_H(t) - B_H(s))^2] \right)$$

Vemos que el lado izquierdo de la igualdad es la Función de Autocorrelación, y el lado derecho lo podemos escribir del siguiente modo teniendo en cuenta que por definición $\forall t \geq 0, E[B(t)] = 0$

$$E[B_H(t)B_H(s)] = \frac{1}{2} \left(\text{Var}[B_H(t)] + \text{Var}[B_H(s)] - \text{Var}[B_H(t) - B_H(s)] \right)$$

Sabemos que $\text{Var}[B_H(t)] = t^{2H}$; $\text{Var}[B_H(s)] = s^{2H}$ y además por tener incrementos estacionarios la $\text{Var}[B_H(t) - B_H(s)] = |t - s|^{2H}$ entonces la expresión de la Autocorrelación $R_{B_H}(t, s)$ queda:

$$R_{B_H}(t, s) = E[B_H(t)B_H(s)] = \frac{1}{2} \left(t^{2H} + s^{2H} - |t - s|^{2H} \right) \quad [3.14]$$

Ahora, con la expresión de la Función de Autocorrelación desarrollada, podemos verificar la tercera condición de autosimilitud enunciada en [3.11].

$$\begin{aligned} R_{B_H}(at, as) &= \frac{1}{2} \left((at)^{2H} + (as)^{2H} - |at - as|^{2H} \right) \\ &= \frac{a^{2H}}{2} \left(t^{2H} + s^{2H} - |t - s|^{2H} \right) \\ &= a^{2H} R_{B_H}(t, s) \end{aligned}$$

$$R_{B_H}(at, as) = a^{2H} R_{B_H}(t, s) \Rightarrow R_{B_H}(t, s) = \frac{R_{B_H}(at, as)}{a^{2H}} \text{ Por lo tanto cumple la condición 3.}$$

Hasta aquí se ha demostrado que el proceso FBM es estrictamente autosimilar, pero si analizamos la expresión:

$$\{B_H(at)\} \stackrel{d}{=} \{a^H B_H(t)\}$$

Claramente se advierte que el proceso no puede ser estacionario, pues para eso debe cumplir la condición:

$$\{B_H(t)\} \stackrel{d}{=} \{B_H(at)\}$$

Por otra parte la media del proceso μ debe ser independiente del tiempo, lo que a su vez implica que $a\mu = a^H \mu$, y además la varianza debería cumplir:

$$a^2 \text{Var}[B_H(t)] = \text{Var}[a^H B_H(t)] \Rightarrow \sigma^2 = a^{-2H} \sigma^2$$

Ambas cosas para cumplirse requieren que $\mu = \sigma^2 = 0$. La única manera de que se cumplan estas condiciones, sería en la forma trivial $B_H(t) = 0, \forall t$, sin embargo al hacer el análisis en el siguiente punto, vemos que el proceso de incrementos de un FBM es estacionario en sentido amplio.

3.11.4 Ruido Gaussiano Fraccional

Al incremento de un proceso de Movimiento Browniano Fraccional se lo conoce como Ruido Gaussiano Fraccional, y como veremos a continuación es un proceso estacionario en sentido amplio, o débilmente estacionario.

Supongamos la correlación entre dos incrementos de un proceso FBM:

1. $X(t) = (B_H(t) - B_H(t-1))$
2. $X(s) = (B_H(s) - B_H(s-1))$

Utilizando la formula de autocorrelación [3.14] en el desarrollo del producto $[X(t)X(s)]$ y agrupando convenientemente los resultados:

$$E[X(t)X(s)] = \{E[B_H(t)B_H(s)] - E[B_H(t+1)B_H(s)] - E[B_H(t)B_H(s+1)] + E[B_H(t+1)B_H(s+1)]\}$$

Aplicando la ecuación [3.14] en cada momento del segundo miembro de la ecuación:

$$\frac{1}{2} \left\{ (t^{2H} + s^{2H} - |t-s|^{2H}) - ((t-1)^{2H} + s^{2H} - |(t-1)-s|^{2H}) - (t^{2H} + (s-1)^{2H} - |t-s+1|^{2H}) + ((t-1)^{2H} + (s-1)^{2H} - |t-1-s+1|^{2H}) \right\}$$

Agrupando los términos:

$$= \frac{1}{2} \left(((t-s)+1)^{2H} + ((t-s)-1)^{2H} - 2(t-s)^{2H} \right)$$

Como se puede ver la expresión depende del valor relativo $(t-s)$, y no depende del tiempo absoluto, lo que es consistente con la estacionariedad en sentido amplio (recordar la definición en el punto [3.3.2](#)), entonces si llamamos k al espacio u holgura temporal $(t-s)$, la autocorrelación del incremento queda:

$$R_{X(k)} = E[X(t)X(s)] = \frac{1}{2} \left((k+1)^{2H} + (k-1)^{2H} - 2k^{2H} \right) \quad [3.15]$$

3.11.4.1 Análisis de la Autocorrelación del Ruido Gaussiano Fraccional

Analizando el comportamiento de la correlación [3.15] con relación a los valores que puede tomar H , se pueden sacar las siguientes conclusiones.

- Si $H = 0.5$, vemos que la correlación se hace cero por lo tanto hay independencia entre los incrementos y el proceso es igual a un proceso de movimiento browniano Hss .
- Si $H = 1$, la correlación es igual a uno para cualquier valor de k y la aleatoriedad del proceso desaparece.
- Si $H > 1$, se pierde la estacionariedad del proceso $X(t)$, y por ende pierde interés como modelo de tráfico.
- Si $H < 0.5$, presenta una correlación negativa, y da lugar a lo que se llama proceso antipersistente. La antipersistencia es lo opuesto a la persistencia, y consiste en la tendencia, o la mayor probabilidad de la variable, de pasar de un valor alto, a un valor bajo, o viceversa, también se lo puede ver como la tendencia a regresar constantemente al lugar de procedencia (viendo el proceso como un desplazamiento significaría que tomaría el mismo valor inicial, o sea pasaría por cero infinitas veces), entonces es una situación irreal desde el punto de vista del tráfico y no tiene significado en este contexto.

- Si $\frac{1}{2} < H < 1$, presenta persistencia en el comportamiento, esto da lugar a la dependencia de largo plazo, y esta característica de persistencia se acentúa cuando H se acerca a uno. Este es precisamente el comportamiento observado en las muestras de tráfico real que presentan autosimilitud.

Por lo expuesto, cuando se analiza el tráfico en el contexto de la autosimilitud, el parámetro de Hurst tiene relevancia en el intervalo abierto $(\frac{1}{2}, 1)$.

Este análisis, también nos sirve para diferenciar claramente los conceptos de dependencia de rango largo y autosimilitud, pues el proceso de movimiento browniano es autosimilar con $H = 0.5$ (H_{ss}), pero no tiene dependencia de rango largo, de hecho en este proceso, los incrementos son independientes e idénticamente distribuidos.

3.11.5 Ventajas del FGN como modelo de tráfico

El proceso de movimiento browniano fraccional, es una generalización del proceso de movimiento browniano, donde la característica sobresaliente es la dependencia de sus incrementos en el intervalo abierto $\frac{1}{2} < H < 1$. El Ruido Gaussiano Fraccional es H_{sssi} (autosimilar con incrementos estacionarios), tiene un particular interés para el modelado del tráfico pues nos permite manejar la dependencia mediante un solo parámetro “ H ”, cumpliendo con la premisa de parsimonia. Además resulta muy atractivo como modelo de tráfico por dos razones fundamentales:

1. Al ser un proceso gaussiano, todavía es posible tratarlo analíticamente.
2. Se adapta naturalmente, porque es el proceso que surge de multiplexar un gran número de procesos on/off en los que la distribución de los tiempos de actividad y/o inactividad tiene "cola hiperbólica" y esto es de mucho interés puesto que las distribuciones de probabilidad de variables tales como el tamaño de los archivos transferidos mediante FTP, los tiempos de conexión TCP, el tamaño de los objetos multimediales en las páginas Web, la longitud de las escenas en video MPEG, etc., tienen este tipo de distribución, por ejemplo, Pareto o Weibul.

Como forma de corroborar esto y acceder a con mayor detalle a un ejemplo de uso del FGN, se sugiere la lectura, de la referencia bibliográfica: [5] Ilka Norros, “On the use of fractional brownian motion in the theory of connectionless networks” [J], IEEE J. Select. Areas Commun, vol. 13, no. 6, Aug. 1995: 953–962. Se trata de un estudio que muestra las ventajas de la utilización del FGN como modelo de tráfico, y establece una comparación con tráfico Ethernet real.

4 Wavelets u Onditas – Análisis Multi-Resolución

En el Capítulo 1, se mencionó que existen ventajas en los métodos basados en wavelets para el estudio del tráfico autosimilar, corroborando esa afirmación, en este capítulo se desarrolla el concepto de wavelet u ondita, su aplicación como método de análisis y los fundamentos del análisis multi-resolución para su utilización en el estudio del tráfico.

Es interesante comentar que las wavelets u onditas, como objeto de investigación formal y organizada, datan de la década de 1980, el concepto del análisis con wavelets, nace de una cantidad de conceptos relacionados al campo del análisis de señales, que se fueron desarrollando durante un período de mas de dos siglos a partir del trabajo de Jean Baptiste Joseph Fourier¹⁸, y luego abordados repetidamente por científicos de diversas disciplinas que los estudiaron y adaptaron para resolver problemas técnicos en sus respectivas áreas, contribuyendo de esta manera, al desarrollo de lo que podríamos llamar esta nueva y eficiente herramienta de análisis.

4.1 Análisis con Transformadas

La idea o concepto del análisis por transformadas es como su nombre lo indica, transformar o modificar un objeto de estudio, de manera tal que, adquiere una representación diferente, que resalta o evidencia, propiedades que estaban ocultas en la representación original. En particular nuestro objeto de estudio es lo que llamamos una señal. Una señal es cualquier magnitud que se pueda medir de alguna forma, y que contenga información sobre el comportamiento o la naturaleza de algún fenómeno. En el caso específico que refiere este trabajo la señal a estudiar es la representación del tráfico como una serie de valores discretos relativos a cualquiera de las medidas apropiadas como ser paquetes, tiempos entre arribos, celdas, etc.

Desde el punto de vista matemático una señal es una función de una o varias variables cuyo dominio de definición puede ser un continuo o una cantidad finita de puntos o serie discreta. Entonces, mediante el procedimiento matemático que llamamos transformada, modificamos o transformamos una señal, de forma tal que exhiba o muestre determinadas características que no son evidentes en su estado original. Al procedimiento de transformación y posterior estudio de los resultados, se lo denomina análisis.

Para poner en contexto el análisis mediante la transformada wavelet, y a la vez comprender mejor sus ventajas, es útil hacer una breve reseña de sus orígenes y antecedentes, partiendo de las herramientas clásicas utilizadas en el análisis de señales.

4.2 Análisis de Fourier o Análisis Armónico

El análisis de Fourier consiste en la descomposición de una señal en componentes senoidales de distintas frecuencias, múltiplos de una frecuencia, denominada fundamental, mediante el producto interno de la señal por un conjunto o base de funciones senoidales, de esta manera se consigue representar la señal estudiada, como una serie o suma de componentes en el dominio de las frecuencias, esto resulta de gran utilidad pues hace evidente aspectos de la señal estudiada, que son imposibles de visualizar en el dominio temporal.

En general, tanto la amplitud de una señal $s(t)$, como su frecuencia pueden variar con el tiempo. En 1812 Fourier publicó un tratado titulado Teoría analítica del calor, en el que asegura que toda

¹⁸ Jean-Baptiste-Joseph Fourier (21 de marzo 1768 en Auxerre - 16 de mayo 1830 en París), matemático y físico francés conocido por sus trabajos sobre la descomposición de funciones periódicas en series trigonométricas convergentes llamadas Series de Fourier, método con el cual consiguió resolver la ecuación del calor. La transformada de Fourier recibió el nombre en su honor.

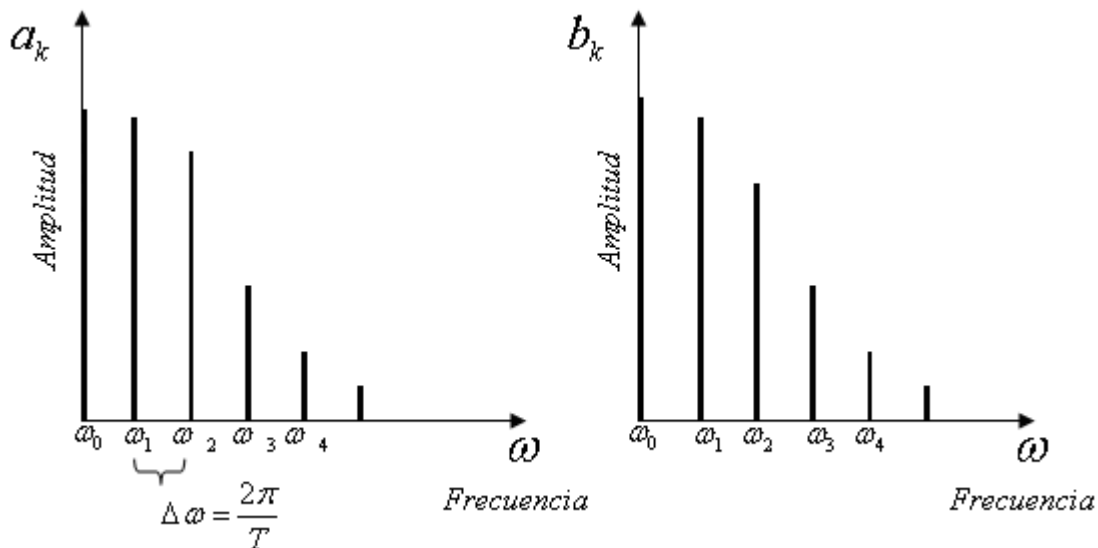
señal periódica de periodo T , es decir que satisface $s(t) = s(t+T)$, puede representarse como una suma infinita de señales estacionarias elementales, esto es:

$$s(t) = a_0 + 2 \sum_{k=1}^{\infty} a_k \cos\left(\frac{2\pi k}{T} t\right) + b_k \operatorname{sen}\left(\frac{2\pi k}{T} t\right)$$

Esta expresión recibe el nombre de Serie de Fourier, donde T , es el período de la señal, $\frac{2\pi}{T} = \omega$, es la frecuencia fundamental, la que a veces se expresa como $2\pi f_0$, y a_0, a_k y b_k , son los llamados coeficientes de Fourier y se calculan como sigue:

$$a_k = \frac{2}{T} \int_0^T s(t) \cos\left(\frac{2\pi k}{T} t\right) dt ; \text{ con } k \geq 1 ; b_k = \frac{2}{T} \int_0^T s(t) \operatorname{sen}\left(\frac{2\pi k}{T} t\right) dt$$

El coeficiente a_0 es el valor medio de la señal en el período T , pues para $k = 0$; $a_0 = \frac{1}{T} \int_0^T s(t) dt$; este coeficiente, también recibe el nombre de componente de continua o de frecuencia cero, en el ámbito de las señales eléctricas.



En la Figura se muestra la representación gráfica de cada uno de los coeficientes de Fourier para una hipotética señal $s(t)$. Los conjuntos $\{a_k\}$ y $\{b_k\}$ se denominan espectro de la señal; y las ordenadas representan la amplitud de cada componente frecuencial o armónica. El eje de abscisas es el mismo para ambos coeficientes, y queda definido por la frecuencia ω_k de cada una de las ondas armónicas simples. Es interesante notar que, el eje de frecuencias es discreto, y que su unidad de escala viene dada por $\Delta\omega = \frac{2\pi}{T}$ y por lo tanto, cuanto mayor sea el período T , menor será el espacio entre las frecuencias y por consiguiente será mayor la resolución frecuencial que podamos obtener.

Recordar que estas expresiones son válidas para señales periódicas, pero si se considera que el período $T \rightarrow \infty$ esto implica que la función no tiene período y en consecuencia $\Delta\omega \rightarrow 0$; en el caso límite los coeficientes se solapanán; su representación deja de ser discreta; en esta condiciones los coeficientes a_ω y b_ω , pasan a ser una función continua en ω . Ambos coeficientes a_ω y b_ω , se definen mediante las siguientes integrales:

$$a_{\omega} = \int_{-\infty}^{\infty} s(t)\cos(\omega t) dt \text{ y } b_{\omega} = \int_{-\infty}^{\infty} s(t)\text{sen}(\omega t) dt$$

En estas condiciones, cuando la señal no es periódica, la serie de Fourier pasa a ser la integral de Fourier conocida también como transformada inversa de Fourier:

$$s(t) = 2 \left[\int_0^{\infty} a_{\omega} \cos(\omega t) d\omega + \int_0^{\infty} b_{\omega} \text{sen}(\omega t) d\omega \right]$$

4.2.1 Forma Compleja de la Transformada de Fourier

La igualdad de Euler expresa que: $e^{i\theta} = \cos\theta + i\text{sen}\theta$, donde $\theta = \omega t$, $i = \sqrt{-1}$; partiendo de esta igualdad, se puede construir la función compleja $S(\omega) = a(\omega) - ib(\omega)$; reemplazando en $S(\omega)$, a_{ω} y b_{ω} por sus expresiones integrales, se obtiene lo que se conoce como la transformada de Fourier de una señal no periódica $s(t)$, y esta dada por la siguiente expresión:

$$S_F(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} s(t) e^{-i\omega t} dt, \text{ donde } \omega = 2\pi f, \text{ es la frecuencia en radianes, y } i = \sqrt{-1}$$

El subíndice F , es por el nombre Fourier, pues se denota a la señal en el dominio del tiempo con la letra “s”, minúscula, y su correspondiente transformada en el dominio de las frecuencias con “S”, mayúscula, entonces aquí se denomina a la transformada de Fourier mediante $S_F(\omega)$, se hace esta pequeña aclaración pues mas adelante se utiliza la “S” mayúscula para denominar otras transformadas, pero con su correspondiente subíndice identificando el tipo de transformada.

En resumen, una integral de Fourier, puede ser considerada como el límite formal de una serie de Fourier cuando el período tiende a infinito, lo cual permite el tratamiento de funciones no-periódicas o aleatorias.

De forma equivalente la transformada discreta de Fourier, también conocida por la abreviatura DFT (del inglés Discrete Fourier Transform), es una transformada de Fourier ampliamente empleada en tratamiento de señales y en campos afines para analizar las frecuencias presentes en una señal muestreada, o sea discreta, y esta expresada por:

$$S(j) = \sum_{k=0}^{n-1} s_k e^{-i\frac{2\pi}{n}jk} \text{ donde } j = 0, \dots, n-1; i = \sqrt{-1}$$

A $S(\omega)$ se lo llama espectro de la señal, y es la representación de $s(t)$ en el dominio de las frecuencias, como se mencionó resulta de gran utilidad pues nos permite conocer la composición frecuencial de la señal, o sea que permite conocer las frecuencias que intervienen en su constitución, y en que medida lo hace cada una.

4.2.2 Transformada Inversa de Fourier

La Transformada de Fourier es reversible, esto significa que partiendo del espectro de una señal, se puede obtener la señal original, a través de lo que reconoce como transformada inversa de Fourier y esta dada por las siguientes expresiones.

$$s(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} S(\omega) e^{i\omega t} d\omega, \text{ para el caso continuo.}$$

$$s(k) = \frac{1}{n} \sum_{j=0}^{n-1} S_j e^{i\frac{2\pi}{n}jk} \text{ donde } k = 0, \dots, n-1, \text{ para el caso discreto.}$$

Si bien las formulas y la Serie de Fourier son correctas, su aplicación requiere que la señal a analizar posea determinadas condiciones, ha requerido un gran esfuerzo de destacados científicos y matemáticos, poder distinguir en qué sentido la representación dada por la fórmula es acertada y aplicable a señales provenientes de procesos reales. Esta representación es correcta en el sentido de la convergencia en el espacio de las señales de energía finita. En particular esta demostrado que las series de Fourier convergen cuando se aplican a señales de energía finita también conocidas como de cuadrado integrable L^2 . El conjunto de todas las funciones medibles de cuadrado integrable sobre un dominio dado, forman lo que se denomina un espacio de Hilbert sumable. En cuanto a la representación punto a punto o en casi todo punto, la formula ha inspirado brillantes desarrollos matemáticos que culminaron con el trabajo de Lennart Carleson¹⁹ sobre la convergencia puntual de la serie de Fourier.

La rama de la matemática que estudia la transformada de Fourier y sus generalizaciones es denominada análisis armónico, este nombre proviene del hecho que a cada componente frecuencial de una señal se la denomina armónica.



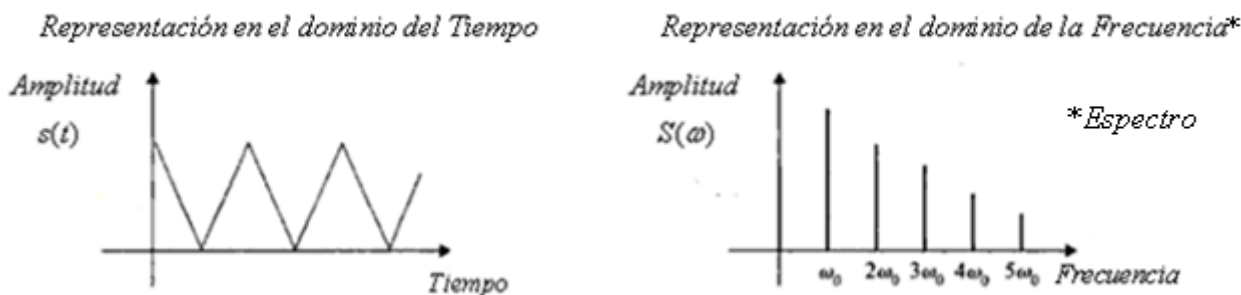
Esta figura, muestra esquemáticamente, como aplicando la transformada de Fourier, a una señal definida en el dominio temporal, se convierte en una señal en el dominio de las frecuencias representada mediante su espectro. Esto significa que podemos evaluar la señal en el tiempo, o podemos evaluar la señal en el dominio de las frecuencias, pero no en ambos al mismo tiempo.

Algo importante de destacar, es que el espectro de señales periódicas es discreto en el dominio de las frecuencias, mientras que el espectro de señales no periódicas es una función continua en el dominio de las frecuencias.

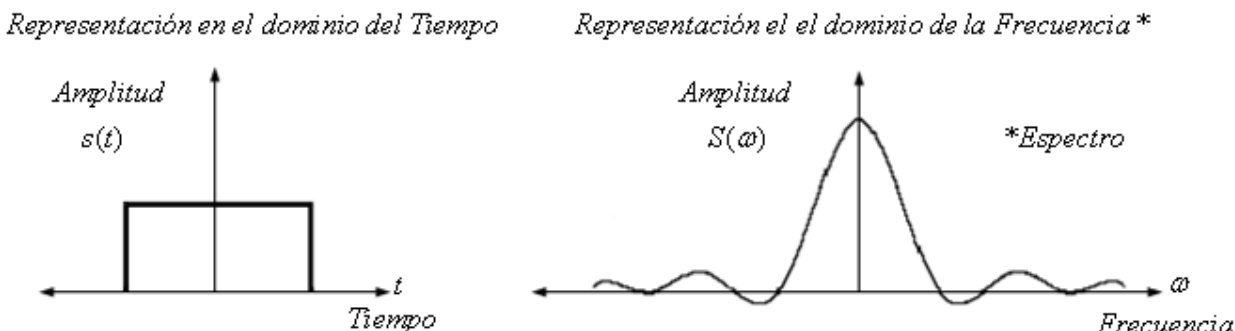
Como ejemplo, se muestran las siguientes figuras con las transformadas de Fourier de una señal periódica triangular, y una señal de pulso cuadrado que no es periódica.

¹⁹ Lennart Axel Edvard Carleson (n. 18 de marzo de 1928) es un matemático sueco, conocido como un líder en el ámbito de análisis armónico. Su trabajo ha incluido la solución de algunos problemas pendientes, por medio de técnicas de Combinatoria. Estos incluyen el teorema corona (1962) en la teoría de espacios de Hardy, en casi todas partes la convergencia de series de Fourier para las funciones integrables cuadradas, y el trabajo en dinámica compleja. También es conocido por la teoría de medidas de Carleson. Fue galardonado con el Premio Wolf en Matemáticas en 1992, la Medalla de Oro Lomonosov en 2002, la Medalla Sylvester en 2003, y el Premio Abel en 2006 por su profunda y seminal contribución al análisis armónico y la teoría de sistemas dinámicos.

Análisis de Fourier para una señal periódica



Análisis de Fourier para una señal no periódica



4.2.3 Limitaciones del Análisis de Fourier

El análisis de Fourier presenta dos inconvenientes o limitaciones:

- 1) La señal debe conocerse en todo instante, además debe ser estacionaria, es decir su comportamiento no varía en todo el intervalo estudiado.
- 2) Si bien permite conocer con precisión la composición frecuencial, no da ninguna información acerca de donde están ubicadas o donde ocurren estas frecuencias dentro del intervalo estudiado.

Con relación a la primera dificultad, casi ninguna señal de existencia real se ajusta estrictamente a esto, si bien es una dificultad salvable definiendo adecuadamente la señal, las señales tienen o pueden tener variaciones bruscas, interrupciones y efectos transitorios que son de mucho interés para el estudio siendo a menudo la parte más importante a estudiar. En cuanto a la segunda dificultad, no hay posibilidad alguna de conocer la ubicación en tiempo de una determinada frecuencia que está evidenciando algún fenómeno transitorio que deseamos estudiar y no podemos precisar donde ocurre. Resumiendo cuando se realiza la transformación al dominio de frecuencias, se pierde la información temporal. Cuando se observa una transformación de Fourier de una señal, es imposible saber cuándo sucede un evento particular. Estas limitaciones hacen que el análisis de Fourier no sea la mejor opción en muchos casos.

4.3 Transformada de Fourier de Tiempo Corto o Transformada de Gabor

En el año 1946 Denis Gabor²⁰, propuso una innovación para superar las limitaciones del análisis de Fourier, y adaptó la transformación de Fourier para analizar solamente una sección pequeña de la señal, mediante la utilización de ventanas.

²⁰ Dennis Gabor (Gábor Dénes) (5 de junio, 1900, Budapest - 9 de febrero, 1979, Londres) fue un físico húngaro que destaca por ser el inventor de la holografía. También es conocido por haber introducido el concepto de Filtro de Gabor, una variante local de la Transformada de Fourier, muy utilizada hoy en día en el mundo de las telecomunicaciones y procesamiento de imágenes.

La adaptación de Gabor, denominada Transformada de Fourier de corto tiempo (Short-Time Fourier Transform (STFT)) mapea una señal en una función bidimensional de tiempo y frecuencia. Esto se consigue mediante el cálculo consecutivo de la transformada de Fourier, de pequeños segmentos de la señal $s(t)$, cada segmento se desplaza por el eje temporal, en un valor correspondiente a la ventana temporal que llamamos $g(t)$, el ancho de la ventana $g(t)$, es lo suficientemente pequeño como para considerar que la señal es estacionaria en ese segmento temporal. La ventana $g(t)$, enmarca una porción de la señal y permite aplicar localmente la Transformada de Fourier; de este modo, se releva la información en frecuencia localizada temporalmente en el dominio efectivo de la ventana. Desplazando temporalmente la ventana se cubre el dominio de la señal obteniéndose la completa información en el plano tiempo – frecuencia. La expresión matemática que describe esto es la siguiente integral:

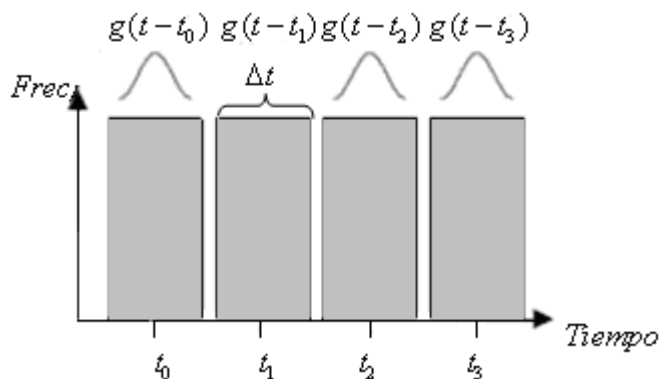
$$S_G(\omega, \tau) = \int_{-\infty}^{\infty} s(t)g(t - \tau)e^{-i\omega t} dt \text{ donde } \omega = 2\pi f ; i = \sqrt{-1}$$

Esta expresión recibe el nombre de transformada de Gabor o también transformada de Fourier de tiempo corto, y puede ser interpretada como la componente en la frecuencia “ ω ”, que la señal $s(t)$ tiene en el instante “ τ ”.

4.3.1 Interpretación de la Transformada de Gabor

La transformada de Gabor tiene dos interpretaciones:

1. Se fija $\tau = t_0$ de forma que se calcula la transformada para $s(t)g(t - t_0)$, esto es un segmento específico de la señal $s(t)$ seleccionado por la ventana $g(t - t_0)$, y de esta manera seleccionando distintos valores centrados de t_0 , se pueden calcular franjas con un ancho Δt constante, para $t_0, t_1, t_2, \dots, t_n$; en el plano frecuencia tiempo como se muestra en la figura:

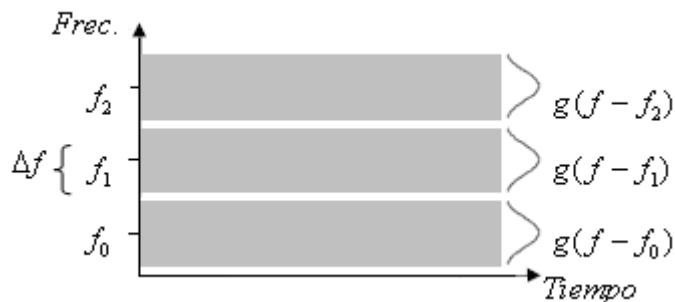


2. Si se escoge $g(t)$, de forma que su contenido espectral sea tan parecido como sea posible a un impulso en $f = 0$ (generalmente $g(t)$ es una gaussiana, pues la transformada de Fourier de una gaussiana es otra gaussiana), entonces se comporta como un filtro pasabajos con un ancho de banda Δf , que está determinado por el ancho y la forma de la ventana $g(t)$. Aquí la transformada de Gabor del producto $s(t)g(t - t_0)$, es:

$$S(f, t_0) = \int_{-\infty}^{\infty} s(t) * g(t - t_0) e^{-i2\pi f t} dt$$

Esto es igual al producto de convolución de $s(t)e^{-i2\pi f_0 t} g(t)$, y se puede interpretar como un filtro pasabanda centrado en f_0 .

Desplazando la ventana, se puede calcular para distintos valores de $f_0, f_1, f_2, \dots, f_n$, y se obtienen franjas con un ancho Δf constante, en el plano frecuencia tiempo, como lo muestra la siguiente figura:



Este análisis forma una cuadrícula en el plano tiempo – frecuencia, donde los lados del cuadro son Δt y Δf respectivamente. Asumiendo que la ventana real $g(t)$, esta bien localizada en un intervalo centrado en $t = t_0$, de longitud Δt , y que su transformada $g(f)$, esta también localizada en una banda centrada en $f = f_0$, de ancho Δf , las ventanas desplazadas y moduladas $g(t - t_0)e^{-if_0t}$ son funciones elementales bien localizadas en el dominio conjunto, tiempo - frecuencia. Cada función elemental se localiza en el rectángulo centrado en el punto (t_0, f_0) de dimensión $\Delta t \times \Delta f$. De esta forma eligiendo convenientemente la forma y tamaño de la ventana $g(t)$, se pueden ubicar tanto en frecuencia como en tiempo los fenómenos que se desean estudiar. Por tanto el conjunto de puntos $\{S(t, f)\}$, mapean la señal $s(t)$ en el dominio tiempo – frecuencia.

Al igual que con la Transformada de Fourier, la Transformada de Gabor se puede invertir recuperando la información del dominio temporal mediante la siguiente formula de inversión:

$$s(t) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} S(t, f)e^{-i f t} df dt$$

En la literatura sobre este tema, se encuentra indistintamente, expresiones en función de la frecuencia angular ω en radianes, que se relaciona con la frecuencia f en Hertz mediante $\omega = 2\pi f$, siendo $f = \frac{1}{T}$, como se ve las magnitudes, frecuencia angular ω ; frecuencia f y período T , se encuentran relacionadas, siendo cómodo utilizarlas en cualquiera de sus equivalencias dependiendo del contexto.

La cuestión que surge aquí es, ¿que tan exactamente se pueden ubicar los detalles a estudiar en una señal?; para contestar a esta pregunta, imaginemos que una señal $s(t)$, tiene una resolución infinita en el tiempo, esto es que se conoce su valor en cualquier instante, y se lo puede ubicar en el tiempo con infinita precisión, por otra parte, si mediante la transformada de Fourier se calcula su espectro, este tendrá una resolución infinita en el dominio de las frecuencias, pudiendo conocer con infinita precisión cualquiera de sus frecuencias, pero sin poder ubicarlas en el tiempo; la transformada de Gabor resuelve esta dificultad fijando un evento en tiempo y frecuencia, pero con la limitación de que la resolución no puede ser arbitrariamente pequeña, como consecuencia del principio de incertidumbre de Heisenberg, pues cuando mas estrecha sea la ventana mejorando la ubicación en el tiempo, mas ancho será el espectro de frecuencias resultante provocando una dispersión en las frecuencias. De hecho en la mecánica cuántica es conocida la indeterminación energía-tiempo, expresada por el producto $\Delta E \Delta t > \frac{h}{4\pi}$, donde h es la constante de Plank, y como la energía de una señal es proporcional a su frecuencia, entonces

existe un límite inferior para los parámetros Δt y Δf , de forma tal que siempre se verifica la relación $\Delta t \Delta f > \frac{\pi}{4}$. Entonces si se escoge un Δt pequeño, esto es la mínima distancia en tiempo que pueden estar dos impulsos para tener una buena resolución en tiempo, forzosamente tendré un Δf mas grande que disminuirá mi resolución en frecuencias, y viceversa, teniendo a Δf como la mínima distancia medida en Hz. a la que pueden estar dos componentes frecuenciales. Se concluye entonces, que existe un compromiso entre tiempo y frecuencia y una vez elegido el tamaño de la ventana, éste es el mismo para todas las frecuencias, esto significa que el análisis se hace con un **ancho de banda constante**.



Esta figura, muestra esquemáticamente como actúa la transformada de Gabor, dividiendo el plano tiempo - frecuencia en cuadros iguales de dimensión $\Delta t \times \Delta f$. Una vez fijado el ancho de la ventana, éste queda constante para todos los desplazamientos. El hecho que se mantenga constante es precisamente la debilidad que presenta el análisis mediante la transformada de Gabor, pues puede que tengamos mala resolución temporal en bajas frecuencias o viceversa mala resolución frecuencial a altas frecuencias dependiendo de cómo se elija Δt y Δf .

4.3.2 Ejemplos de análisis de Gabor

A modo de ejemplo, para comprobar los conceptos, y comprender mejor los resultados, se muestran dos análisis de Gabor, llamados Caso1 y Caso 2. Los análisis se hacen mediante dos aplicaciones desarrolladas con el software MATLAB®, las scripts se llaman “estudioGabor.m” y “estudioWav.m” respectivamente, y el código fuente de las mismas está disponible en el apéndice 1, en los puntos [9.1.1](#) y [9.1.2](#), el código está comentado y listo para llevar a cabo los dos ejemplos aquí mencionados.

El parámetro " α ", al que se hace alusión en ambas aplicaciones, esta relacionado con la apertura temporal de la ventana de Gabor, de forma tal, que disminuyendo su valor mejora la localización en el tiempo, pero con menos resolución en frecuencias (mayor dispersión), y en caso contrario, si se aumenta el valor de " α ", mejora la resolución en frecuencias pero empeora la localización temporal, como se muestra a continuación.

4.3.2.1 Análisis Gabor Caso 1

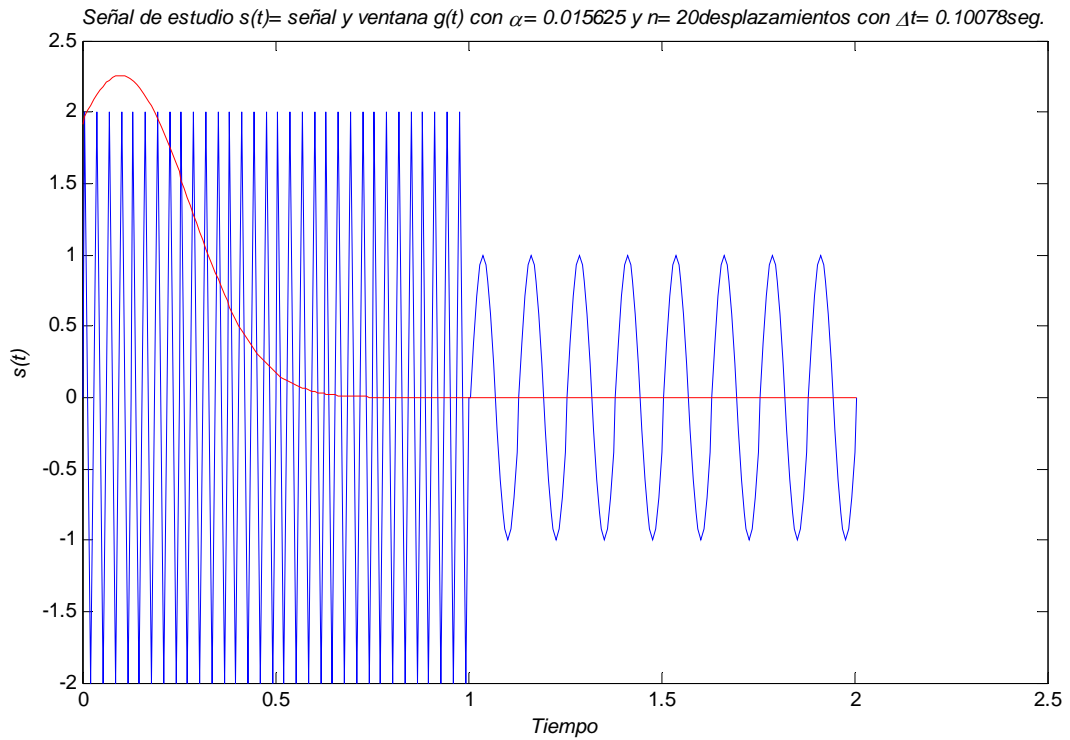
En este caso la señal consiste en senoidal compuesta de dos frecuencias, las cuales cambian luego de un cierto tiempo, empezando con una frecuencia durante un tiempo t_1 , y luego con una frecuencia mas baja, durante un tiempo t_2 .

Señal a estudiar $s_{(t)} = 2\sin(2\pi 32t_1); 0 \leq t_1 \leq 1; \sin(2\pi 8t_2); 1 \leq t_2 \leq 2$

Señal ventana: $g_{(t)} = \frac{1}{2\sqrt{4\alpha}} e^{-\frac{2t}{4\alpha}}$

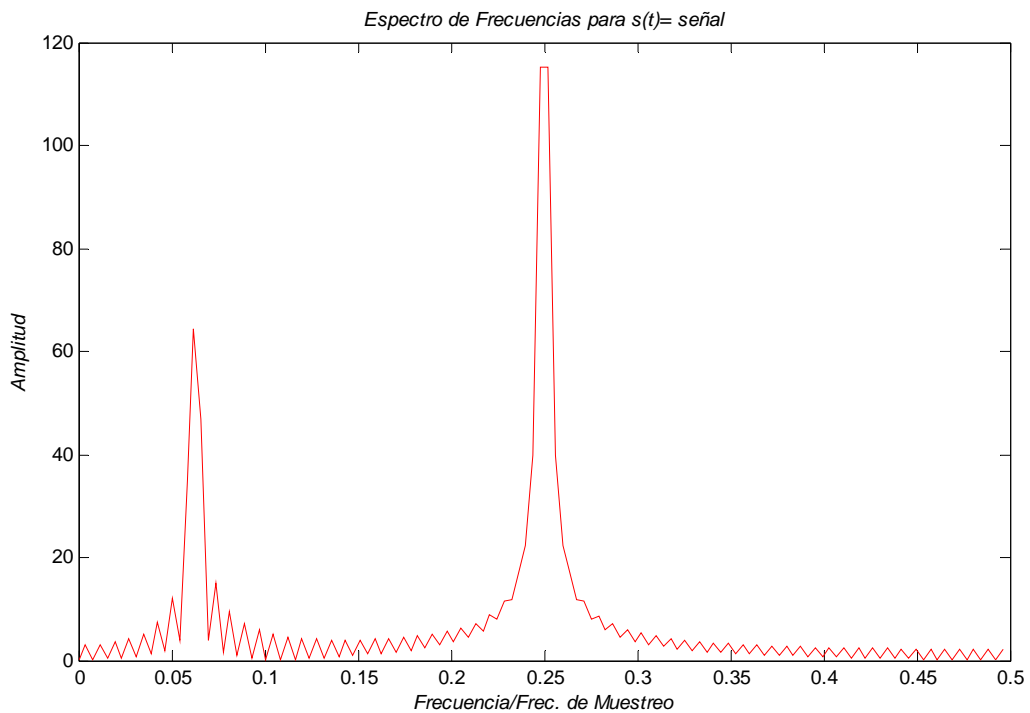
Frecuencia de muestreo = 128 Hz.

Gráfica de la señal $s_{(t)}$ y ventana $g_{(t)}$, con $\alpha = 1/64$, en el dominio del tiempo.



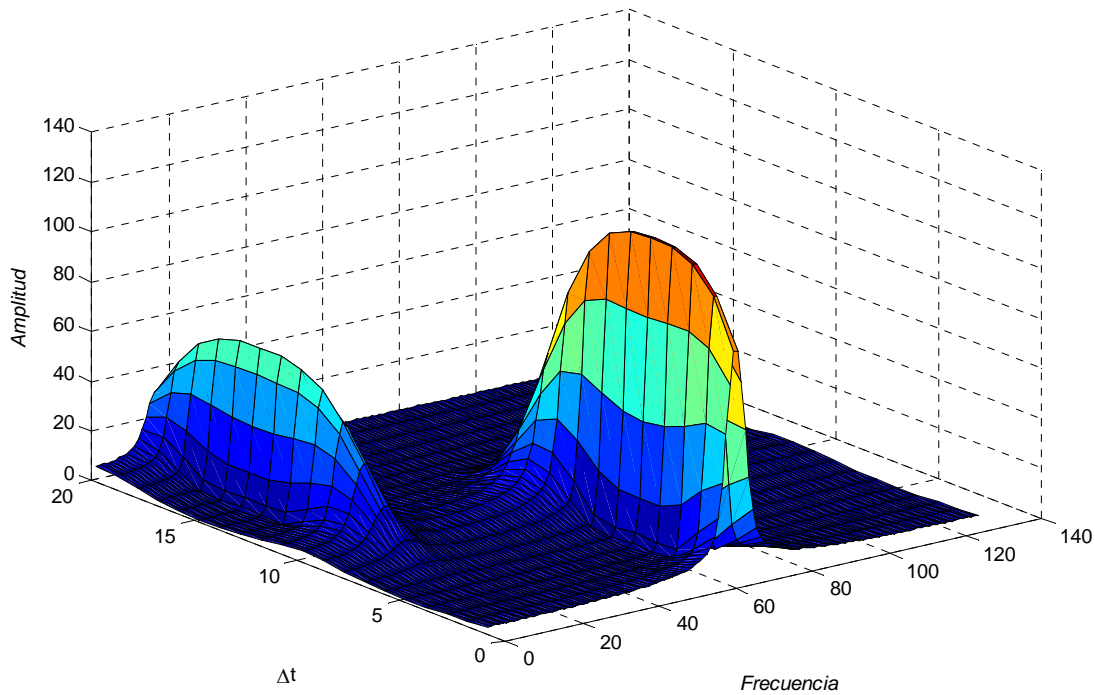
Espectro de Fourier de la señal $s_{(t)} = 2\sin(2\pi 32t_1); 0 \leq t_1 \leq 1; \sin(2\pi 8t_2); 1 \leq t_2 \leq 2$

El eje x tiene como escala las frecuencias normalizadas a la frecuencia de muestreo.



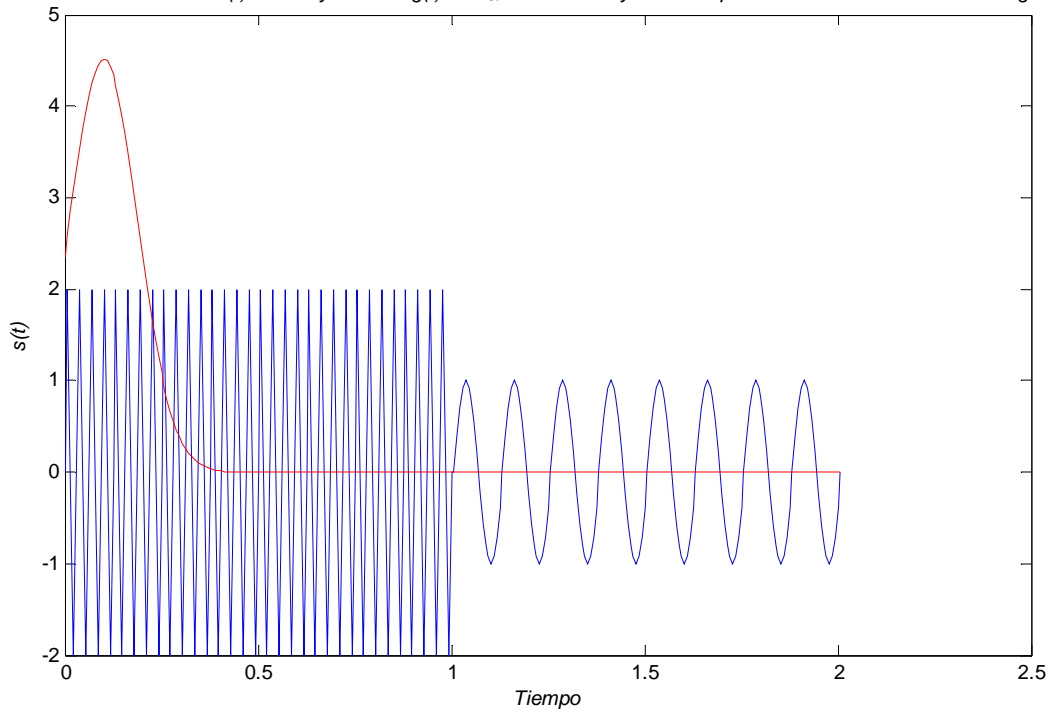
Gráfica de Gabor de la señal $s_{(t)}$ y ventana $g_{(t)}$ con $\alpha = 1/64$:

Análisis de Gabor para $s(t)$ = señal y ventana $g(t)$ con $\alpha= 0.015625$ y 20 desplazamientos con $\Delta t= 0.10078$ seg.



Gráfica de la señal $s_{(t)}$ y ventana $g_{(t)}$, con $\alpha = 1/256$, en el dominio del tiempo.

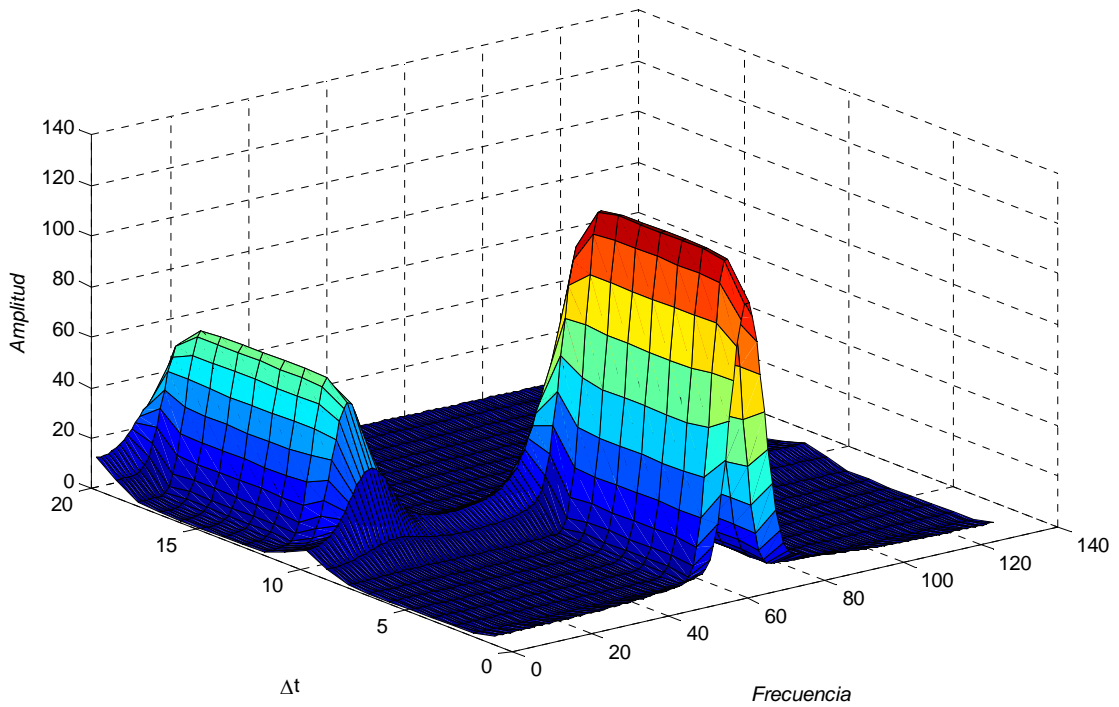
Señal de estudio $s(t)$ = señal y ventana $g(t)$ con $\alpha= 0.0039063$ y $n= 20$ desplazamientos con $\Delta t= 0.10078$ seg.



Aquí se observa que la ventana se achicó por un factor de 4.

Gráfica de Gabor de la señal $s_{(t)}$ y ventana $g_{(t)}$ con $\alpha = 1/256$:

Análisis de Gabor para $s(t)$ = señal y ventana $g(t)$ con $\alpha = 0.0039063$ y 20 desplazamientos con $\Delta t = 0.10078$ seg.



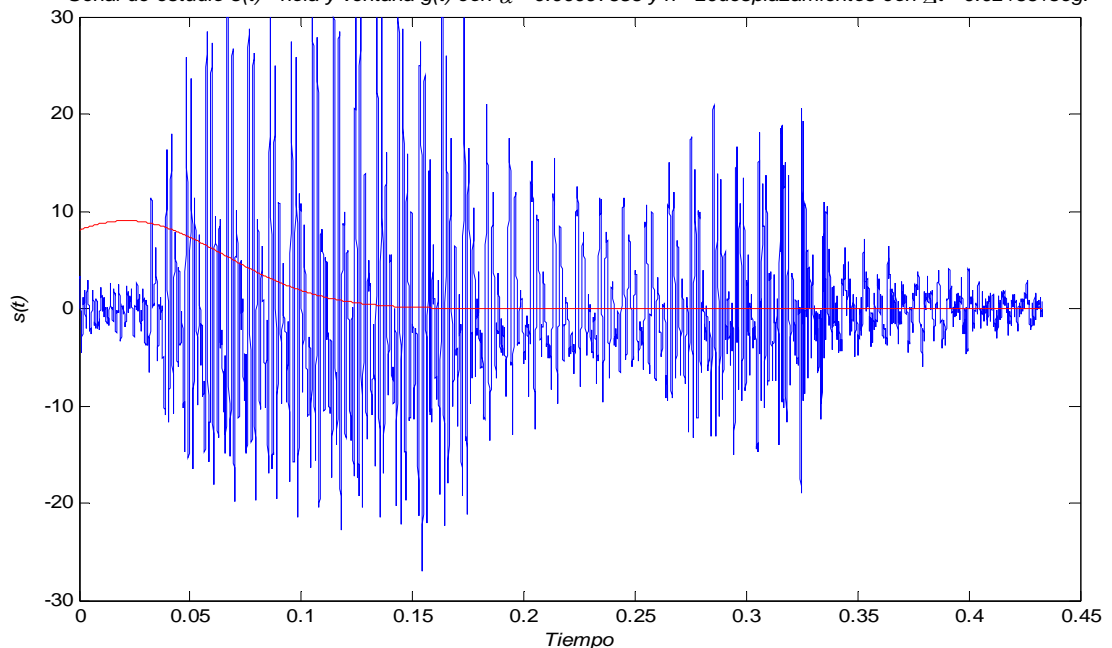
Comparando esta figura con la anterior (para $\alpha = 1/64$), se puede apreciar como se aplanó la parte superior del espectro, mejorando la localización en el tiempo pero con mayor dispersión de las frecuencias.

4.3.2.2 Análisis Gabor Caso 2

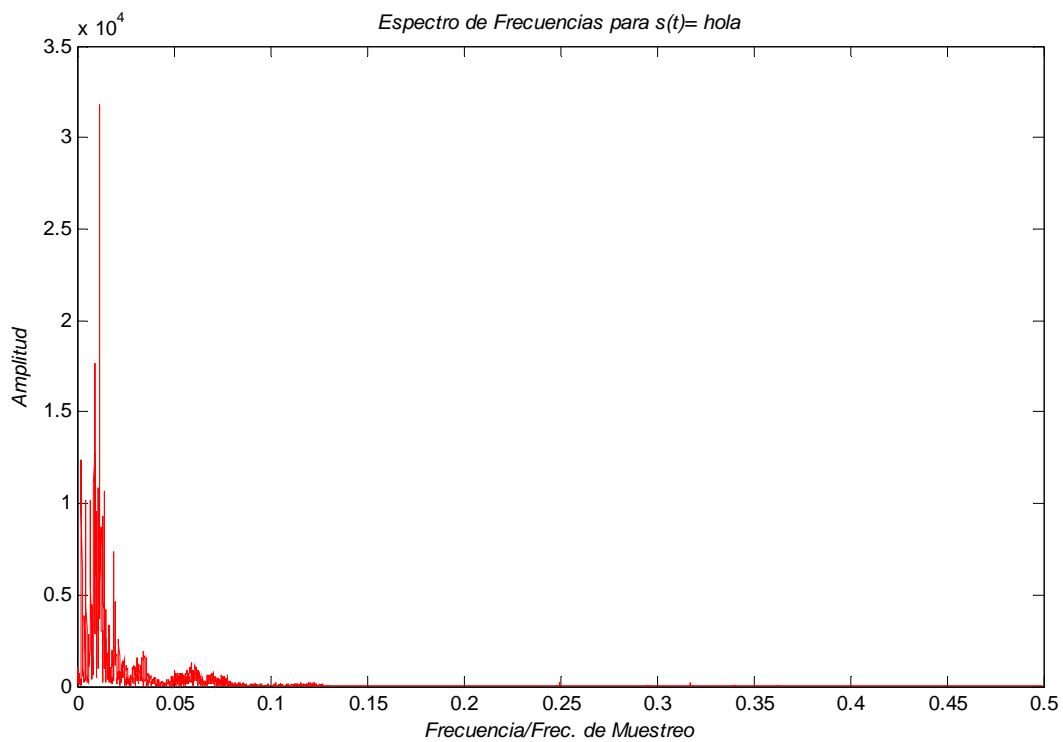
En este caso la señal se trata de un archivo de audio en formato .wav muestreado a una frecuencia de 44.1 KHz.. Se trata de la palabra “hola”, grabada con la grabadora de sonidos de Windows®, y el archivo obtenido, luego es procesado con la script “estudioWav.m”, esta script, utiliza la misma señal de ventana $g_{(t)}$, que la script del análisis anterior.

Gráfica de la señal “hola”, y ventana $g_{(t)}$, con $\alpha = 1/2^{10}$, en el dominio del tiempo.

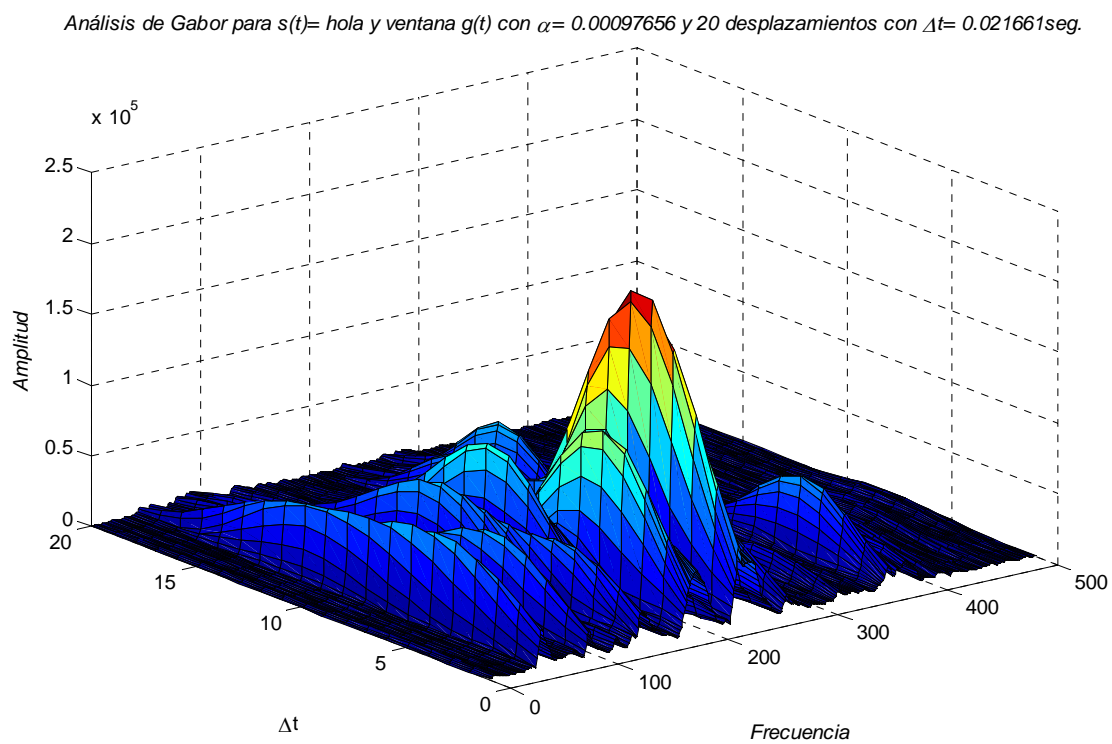
Señal de estudio $s(t)$ = hola y ventana $g(t)$ con $\alpha = 0.00097656$ y $n= 20$ desplazamientos con $\Delta t = 0.021661$ seg.



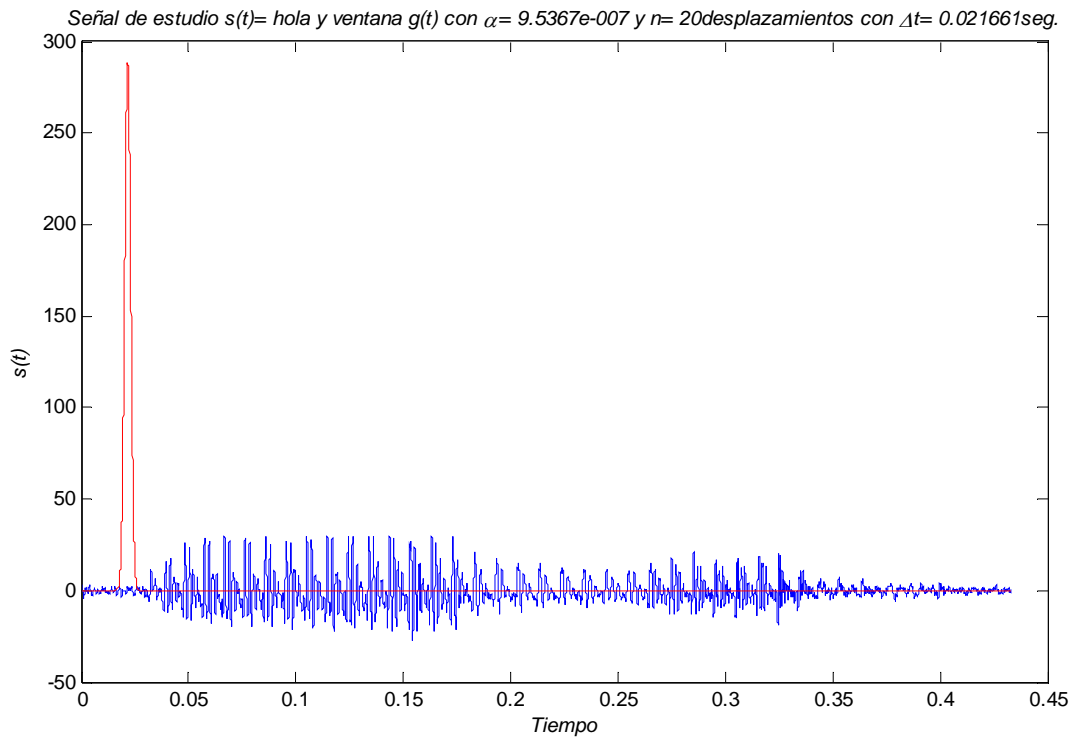
Espectro de Fourier de la señal de audio “hola”:



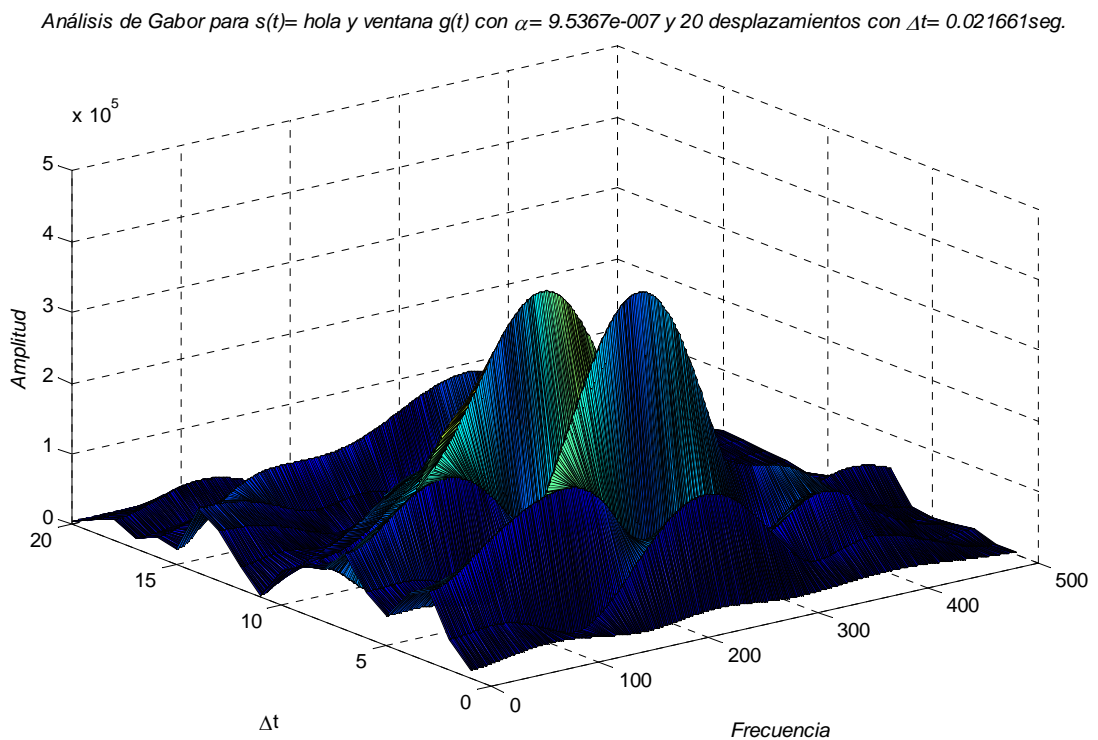
Gráfica de Gabor de la señal “hola”, y ventana $g(t)$ con $\alpha = 1 / 2^{10}$:



Gráfica de la señal “hola”, y ventana $g_{(t)}$ con $\alpha = 1/2^{20}$, en el dominio del tiempo.



Gráfica de Gabor de la señal “hola”, y ventana $g_{(t)}$ con $\alpha = 1/2^{20}$:



Comparando ambos gráficos de Gabor, se puede observar como se invierten las resoluciones temporales y frecuenciales, con el cambio del parámetro $\alpha = 1/2^{10}$, por $\alpha = 1/2^{20}$.

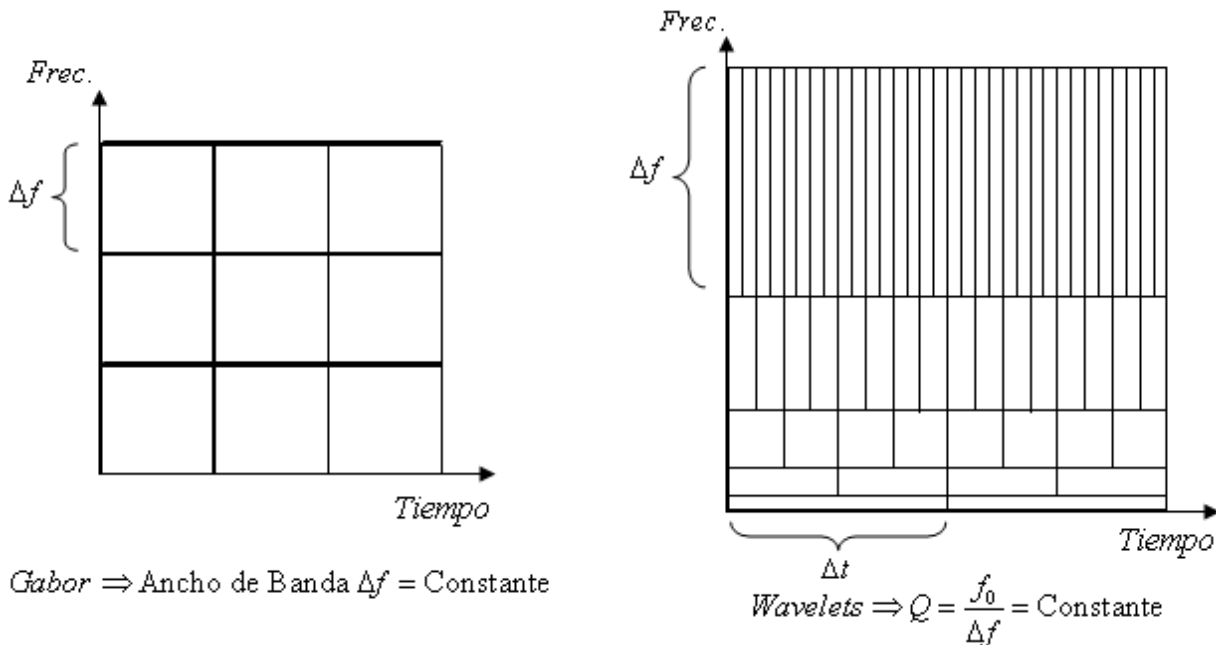
4.4 El análisis con Wavelets u Onditas

El análisis Wavelet consiste en lo que sería el siguiente paso lógico siguiendo la secuencia de los análisis que se comentaron, esto es, una técnica donde se utilicen ventanas con tamaño variable en vez de ventanas de tamaño constante como en el caso del análisis de Gabor. El análisis Wavelet permite utilizar intervalos temporales largos donde se requiere información de baja frecuencia, e intervalos más cortos donde se requiere información de alta frecuencia. Esto significa que con el mismo procedimiento de análisis, se puede trabajar simultáneamente:

- Zonas de frecuencias altas Δf grande con un Δt pequeño, con esto se tiene una buena resolución temporal, por ejemplo para ubicar un fenómeno transitorio de muy corta duración.
- Zonas de frecuencias bajas con un Δt grande se tiene un Δf pequeño con lo que se tiene una buena resolución en frecuencias bajas pudiendo distinguir o individualizar frecuencias muy cercanas.

En términos frecuenciales, esto equivale a decir que se esta trabajando la señal con "**factor de calidad Q constante**", el factor de calidad " Q ", es una magnitud muy utilizada en electrónica para el diseño de filtros de señales, Q se define como el cociente entre la frecuencia central y el ancho de banda. $Q = \frac{f_0}{\Delta f}$. Entonces, la posibilidad de efectuar el análisis a " Q " constante, resulta muy ventajosa, pues el ancho de banda Δf pasa a ser proporcional a la frecuencia elegida para el análisis, en lugar de permanecer fijo como es el caso del análisis de Gabor.

Para comparar como queda dividido el plano frecuencia-tiempo, tanto para el análisis de Gabor, como para el análisis Wavelet; se muestran las siguientes figuras:



Mientras que en la figura de la izquierda correspondiente a un análisis de Gabor, el plano frecuencia-tiempo queda dividido en áreas iguales, tanto para frecuencias altas como bajas, en la figura de la derecha, correspondiente a un análisis wavelet, se observa que el plano frecuencia tiempo, a medida que aumenta Δf , disminuye en forma proporcional Δt mostrando una alta resolución temporal en la zona de frecuencias altas, mientras que muestra una alta resolución frecuencial en la zona de bajas frecuencias, donde Δt es mas grande.

4.4.1 ¿Que es una Wavelet?

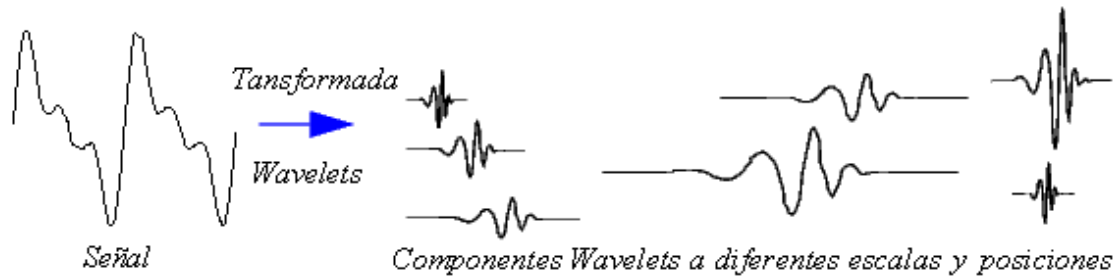
Hasta aquí, se tiene una descripción del concepto del análisis con wavelets y sus ventajas, expuesta en forma intuitiva; pero ¿que es una Wavelet? , ¿En que consiste la transformada Wavelet?; a continuación se contestan estas preguntas.

Tanto en el análisis de Fourier, como en el de Gabor, el procedimiento consiste en el producto interno o proyección de la señal sobre una determinada base de funciones, y algo similar ocurre con la denominada base Wavelet. Un breve examen del producto interno para los tres casos (Fourier, Gabor y Wavelets), permite simultáneamente resumir el concepto de cada uno, y arribar a una definición formal de una Wavelet y su transformada.

1. El análisis de Fourier descompone la señal $s(t)$ mediante el producto interno por las funciones de la base de Fourier $e^{-i\omega t}$ (sinusoides), estas son de duración infinita en el tiempo, pero locales en frecuencia. La Transformada de Fourier detecta la presencia de una determinada frecuencia pero no brinda información acerca de la evolución en el tiempo de las características espectrales de la señal.
2. El análisis de Gabor o STFT mejora el análisis de Fourier al obtener conjuntamente los marcos temporales y de frecuencias de una señal, aportando alguna información sobre cuándo y en qué frecuencias ocurre un determinado detalle de la señal. Lo hace mediante el producto interno de la señal $s(t)$ y la función ventana $g(t)$ de duración Δt . No obstante, la información obtenida mediante esta técnica presenta una precisión limitada, ya que la precisión es función del tamaño de la ventana (que tampoco puede ser arbitrariamente pequeño por el principio de incertidumbre). Mientras que la información de tiempo y de frecuencia (obtenida por STFT) puede ser útil, la desventaja es, una vez que se elige un tamaño particular para la ventana temporal, la ventana es la misma para todas las frecuencias, pudiendo la resolución resultar insuficiente, ya sea para bajas o altas frecuencias dependiendo de cómo se eligió la ventana.
3. En el análisis Wavelet, de manera análoga a los anteriores, se efectúa el producto interno entre la señal $s(t)$ y la base wavelet $\psi_{a,\tau}(t)$. A continuación se define formalmente el concepto de wavelet u ondita, y como se construye la base de funciones wavelet.

Una wavelet u ondita, es una señal bien ubicada tanto en frecuencia como en tiempo, esto es, una oscilación de una frecuencia determinada y duración determinada, de allí proviene el nombre de wavelet u ondita. La base de wavelets u onditas se construye a partir de la denominada wavelet madre $\psi_0(t)$, y se obtiene mediante la dilatación o contracción de la misma, de acuerdo a un parámetro de escala "a". Este parámetro de escala representa la relación entre la frecuencia central de la wavelet madre y la wavelet derivada, por este motivo, en un análisis wavelet, se habla de dominio de las escalas en lugar de dominio de las frecuencias. Por otra parte, de manera similar a la ventana de Gabor, la Wavelet se puede desplazar en una cantidad τ , $\psi_a(\tau - t)$ permitiendo lograr de manera simultánea (escala-tiempo), la adecuada ubicación de cualquier detalle que se quiera analizar en una señal.

La transformada Wavelet, queda definida por el producto interno entre la señal $s(t)$, y la base wavelet $\psi_{a,\tau}(t)$, y representa a la señal en términos de versiones trasladadas y dilatadas de una onda finita denominada wavelet madre $\psi_0(t)$. De manera muy general, la Transformada Wavelet de una función $s(t)$ puede ser vista como la descomposición de $s(t)$ en un conjunto de funciones $\psi_{a,\tau}(t)$, que forman una base y son llamadas "wavelets". En la siguiente figura se expone de manera gráfica la idea.



4.4.2 La Transformada Wavelet Continua o CWT

La Transformada Wavelet no es solamente local en tiempo, sino también en frecuencia, y además del análisis local de señales no estacionarias se utiliza en campos tales como de señales electrocardiográficas, sísmicas, de sonido, de radar, compresión y procesamiento de imágenes y detección o reconocimiento de patrones, siendo esto último de particular interés en el estudio de trazas de tráfico.

La Transformada Wavelet se define formalmente como:

$$S_W(a, \tau) = \int_{-\infty}^{\infty} s(t) \psi_a(t - \tau) dt$$

$S_W(a, \tau)$; es el conjunto de coeficientes $\{S(a, \tau) = \langle s, \psi_{a,\tau} \rangle; a \in \mathbb{R}^+, \tau \in \mathbb{R}\}$, que comparan mediante productos internos, la señal, con un conjunto llamado base de funciones de análisis o base wavelet:

$$\psi_{a,\tau}(t) = \frac{1}{\sqrt{a}} \psi_0\left(\frac{\tau - t}{a}\right); a \in \mathbb{R}^+, t \in \mathbb{R}$$

$\psi_{a,\tau}(t)$, es la función ondita, donde el parámetro "a", es el parámetro de dilatación, que se relaciona con la frecuencia de oscilación, y "τ", es el parámetro de corrimiento, relacionado con la posición en el tiempo de la wavelet u ondita madre ψ_0 .

A los efectos de posibilitar la transformación $S_W(a, \tau)$, las wavelets u onditas tienen que cumplir con los siguientes requisitos o condiciones:

- Deben estar limitadas a un intervalo de tiempo (soporte compacto).
- Su energía debe estar concentrada en una pequeña banda de frecuencias.
- Deben satisfacer la denominada condición de admisibilidad $\int \psi_0(t) dt = 0$, o sea que la media de la Wavelet madre ψ_0 es nula.

Otra manera de expresar la condición de admisibilidad es la siguiente:

- Sea $\psi(t) \in L^2(\mathbb{R})$ (Funciones de cuadrado integrable o energía finita).
- $C_\psi = \int_{-\infty}^{\infty} \frac{|\psi(\omega)|^2}{|\omega|} d\omega < \infty$ donde C_ψ es una constante que depende de ψ .

Las funciones que cumplen con la condición de admisibilidad, representan una oscilación de corta duración, con rápido decaimiento.

Al cumplir con las condiciones mencionadas, se puede reconstruir la señal invirtiendo la transformada:

$$s(t) = C_\psi \iint S_W(a, \tau) \psi_{a,\tau}(t) \frac{dad\tau}{a^2}$$

La fórmula de reconstrucción indica que la transformada $S_W(a, \tau)$ tiene toda la información contenida en $s(t)$, pero mapeada en un espacio bidimensional (escala-tiempo), esto significa que es una transformación redundante, en la que los coeficientes vecinos en el plano escala-tiempo, contienen mucha información en común, y esta circunstancia plantea la posibilidad de hacer un muestreo de los coeficientes en el plano escala-tiempo, de forma tal que retengan la información total, necesaria para reconstruir la señal, esto esta planteado en una teoría matemática conocida como **Análisis Multi-Resolución** o MRA, y el procedimiento de muestreo mencionado, conduce a lo que es conocido como la Transformada Wavelet Discreta o DWT, que es la que sirve a los propósitos del estudio de tráfico, dado que el mismo se modela con procesos estocásticos definidos en puntos discretos del tiempo, o análisis de trazas de tráfico, real o sintético, que también son conjuntos de valores discretos.

4.5 Análisis Multi-Resolución

El análisis multi-resolución es una técnica que permite analizar señales en múltiples bandas de frecuencia. Consiste en un conjunto de sub-espacios vectoriales anidados $\{V_j\}$ con $j \in \mathbb{Z}$; que satisfacen las siguientes cuatro condiciones:

1. $\bigcap_{j \in \mathbb{Z}} V_j = \{0\}$. Y además $\bigcup_{j \in \mathbb{Z}} V_j$, es un conjunto denso en $L^2(\mathbb{R})$
2. $V_j \subset V_{j-1}$
3. $s(t) \in V_j \Leftrightarrow s(2^j t) \in V_0$
4. \exists una función de escala $\phi_0(t) \in V_0$ tal que $\phi_0(t - k), k \in \mathbb{Z}$ es una base de Riesz para V

Una base de Riesz procede de una base ortonormal²¹ mediante algún operador invertible. Toda base de Riesz no es necesariamente ortonormal, pero tiene asociadas dos constantes positivas tales que sus productos por el cuadrado de la norma de un elemento del espacio, acotan superior e inferiormente la suma de los cuadrados de los productos escalares del elemento por los vectores de la base.

Básicamente el análisis consiste en la proyección de la señal sobre cada uno de los sub-espacios V_j , los que se describen a continuación.

Los V_j , son una aproximación del espacio de funciones de energía finita $L^2(\mathbb{R})$, las funciones de escala desplazadas, son linealmente independientes y en conjunto generan el espacio V_0 , pero no son necesariamente ortonormales, por eso se dice son una base de Riesz

²¹ Un espacio vectorial se llama ortogonal si el producto interno entre cualquier par de sus vectores es cero, y si además la norma estos vectores es 1, se lo denomina espacio ortonormal, por ejemplo, en un espacio Euclídeo (dos dimensiones), la ortogonalidad equivale a la perpendicularidad, entonces se dice que dos vectores son ortonormales si son perpendiculares, y su módulo es uno.

Teniendo en cuenta la condición 3, la función de escala básica ϕ_0 , dilatada por un factor de escala 2^j , es desplazada con un factor de escala discreto de traslación $k \in \mathbb{Z}$, y de esta manera forman el conjunto:

$$\left\{ \phi_{j,k}(t) = 2^{-\frac{j}{2}} \phi_0(2^{-j}t - k) \right\}$$

Se dice que este conjunto es una base de Riesz, para el espacio V_j .

Las funciones de escala $\phi_{j,k}(t)$ juegan el papel de funciones promedio. La correlación entre la función de escala y una función continua arbitraria $s(t)$, produce la aproximación promediada de $s(t)$:

$$a_s = \int_{-\infty}^{\infty} s(t) \phi_{j,k}(t) dt$$

El resultado de la proyección de $s(t)$ sobre $\phi_{j,k}(t)$, es el producto interno $\langle s(t), \phi_{j,k}(t) \rangle$, y representa la correlación entre las mismas muestreada a intervalos enteros, y da por resultado una aproximación menos detallada de $s(t)$ para cada valor de j . El conjunto de todas las aproximaciones de $s(t)$, forman el espacio $V_0 \in L^2(\mathbb{R})$. El espacio vectorial V_0 , puede ser interpretado como el conjunto de todas las posibles aproximaciones de la función $s(t)$ en $L^2(\mathbb{R})$, generado por el conjunto de las funciones de escala $\phi_{j,k}(t)$.

Suponiendo que las funciones de escala $\phi(t)$, cumplen con los requisitos enunciados para todas las escalas $a = 2^j$ con $j \in \mathbb{Z}$, generadas a partir de la misma $\phi(t)$, todas son de forma similar.

Debido a que la función de escala básica ϕ_0 , genera la base ortonormal $\phi(t-k)$ de V_0 , con un paso de traslación k entero, entonces:

- a) La función de escala dilatada $\phi(t/2)$ generará la base ortonormal $\phi(2^{-1}t - k)$ de V_1 , con un paso de traslación igual a 2.
- b) De igual forma, la función $\phi(t/4)$ generará la base ortonormal $\phi(2^{-2}t - k)$ de V_2 , con un paso de traslación igual a 4.
- c) La función $\phi(t/8)$ generará la base ortonormal $\phi(2^{-3}t - k)$ de V_3 , con un paso de traslación igual a 8, y así en forma sucesiva.

Existe entonces un conjunto de bases ortogonales de las funciones de escala. Cada base de la función de escala es ortonormal en el espacio de la misma escala.

Las proyecciones en $L^2(\mathbb{R})$, sobre el conjunto de bases ortonormales de la función de escala, forman un subconjunto de sub-espacios V_j . Cada sub-espacio V_j es a su vez el conjunto de todas las aproximaciones de $s(t) \in L^2$, generado por las traslaciones de la función de escala $\phi(2^{-j}t - k)$, este sub-espacio V_j , es abarcado por la base ortonormal de la función escala al nivel de resolución "j", por este motivo la función de escala genera los sub-espacios del análisis multi-resolución.

Las aproximaciones de una $s(t) \in L^2$, en diferentes resoluciones deben ser similares, ya que son todas generadas partiendo de la misma función de escala, pero con escalas diferentes. Los

espacios de aproximación V_j , pueden ser entonces, deducidos unos de otros por simple dilatación, como esta expresado en la tercera condición como $s(t) \in V_j \Leftrightarrow s(2^j t) \in V_0$.

Esto significa que toda la información útil para calcular la aproximación en un nivel de resolución "j", esta contenida en la función de aproximación del nivel superior "j-1", por lo tanto $V_j \subset V_{j-1}$, como lo expresa la segunda condición de los sub-espacios del análisis multi-resolución. El análisis multi-resolución es una técnica que permite analizar señales en múltiples bandas de frecuencia relacionadas a la escala de V_j , y son una aproximación del espacio de funciones de energía finita $L^2(\mathbb{R})$, conforme la secuencia:

$$V_j \subset V_{j-1} \dots \subset V_1 \subset V_0 \subset V_{-1} \subset V_{-2} \dots \subset L^2(\mathbb{R})$$

Cuando la resolución se incrementa con $j \rightarrow -\infty$, la función aproximada, debería converger a la función original. Esto esta expresado en la primera condición de los espacios como $\bigcup_{j \in \mathbb{Z}} V_j$ es denso en $L^2(\mathbb{R})$, y por el contrario cuando $j \rightarrow \infty$, las aproximaciones contienen cada

vez menos información y convergen a cero de manera que $\bigcap_{j \in \mathbb{Z}} V_j = \{0\}$.

En el análisis multi-resolución entonces, se proyecta $s(t)$ sobre cada uno de los sub-espacios V_j , de manera que la aproximación j , es la proyección de $s(t)$ sobre el sub-espacio V_j , generado por las traslaciones k , de $\phi_{(j,k)}(t)$:

$$\text{aproximación}_j(t) = \text{proyección}_{V_j} s(t) = \sum_k a_s(j, k) \phi_{(j,k)}(t)$$

La idea central del análisis multi-resolución es analizar una señal, a partir de aproximaciones cada vez más burdas, donde los detalles de la señal, como por ejemplo transientes de altas frecuencias, se cancelan sucesivamente. Debido a que la proyección de una función sobre la base de la función de escala ortonormal es una aproximación menos detallada de la función en un nivel, con respecto del nivel superior, se pierde algo de información en el proceso, esto significa que la función de escala $\phi(t)$, no es completa a cualquier nivel, conforme se hace la siguiente aproximación, y estos detalles consistentes en la información que se pierde se pueden expresar como:

$$\text{detalle}_j(t) = \text{aproximación}_{j-1}(t) - \text{aproximación}_j(t)$$

Los detalles se pueden obtener directamente por la proyección de $s(t)$ sobre un conjunto de sub-espacios $W_j = V_j - V_{j-1}$, llamados sub-espacios wavelets.

La teoría del análisis multi-resolución muestra la existencia de una función $\psi_0(t)$, llamada wavelet madre, que se obtiene a partir de la función escala $\phi_0(t)$, como el conjunto:

$$\left\{ \psi_{(j,k)}(t) = 2^{-j/2} \psi_0(2^{-j} t - k), \text{ con } k \in \mathbb{Z} \right\}$$

Este conjunto forma una base Riesz para W_j , se tiene entonces que el detalle j , es la proyección $s(t)$ en el sub-espacio W_j , formado por las traslaciones k , de $\psi_{(j,k)}(t)$:

$$detalle_j(t) = \text{proyección}_{W_j} s(t) = \sum_k d_s(j, k) \psi_{(j,k)}(t)$$

Una propiedad de las wavelets expresa que, si la transformada de Fourier de la wavelet madre $\widehat{\psi}(\omega)$, satisface las condiciones de ortogonalidad, el conjunto de las traslaciones discretas de la wavelet madre $\{\psi(2^{-j}t - k)\}$, forman una base ortonormal para cada escala 2^j , y en el mismo nivel de resolución j , el conjunto de traslaciones de la función escala $\{\phi(2^{-j}t - k)\}$, es ortogonal al conjunto de traslaciones wavelet, entonces el conjunto representado por el producto interno es igual a cero.

$$\langle \phi_{j,k}, \psi_{j,n} \rangle = 2^{-j} \int \phi_j(t - k) \psi_j(t - n) dt = 0 ; \forall k, n \in \mathbb{Z}$$

La proyección de $s(t)$ sobre las bases wavelet es una correlación entre $s(t)$ y $\psi(t)$, muestreada a intervalos discretos y contiene los detalles de $s(t)$, por otra parte la base wavelet $\{\psi(2^{-j}t - k)\}$ es ortogonal a la base formada por la función de escala $\{\phi(2^{-j}t - k)\}$, entonces dentro de la misma escala 2^j , el sub-espacio W_j , es el complemento ortogonal del sub-espacio V_j . esto es $V_j \perp W_j$. Como se mostró mas arriba, tanto V_j , como W_j , son sub-espacios de V_{j-1} : $V_j, W_j \in V_{j-1}$, y en razón de la complementariedad de V_j y W_j , V_{j-1} es la suma directa de ambos:

$$V_{j-1} = V_j \oplus W_j$$

Teniendo en cuenta lo expresado, si tengo una señal $s(t)$ la puedo expresar en términos de su aproximación y sus detalles como:

$$s(t) = \text{Proyección}_{V_j} + \text{Proyección}_{W_j}$$

En teoría, la variación de j , puede ser $-\infty \leq j \leq \infty$, pero en la práctica se limita al conjunto $\mathbb{Z}^+, \{0, 1, 2, \dots, J\}$, si se expresa la señal como una aproximación final de muy baja resolución perteneciente al espacio V_J , y un conjunto de detalles a diferentes resoluciones se tiene la siguiente expresión:

$$s(t) = \text{aproximación}_0(t) = \text{aproximación}_J + \sum_{j=1}^J \text{detalle}_j(t) \quad [4.1]$$

$$s(t) = \sum_k a_s(J, k) \phi_{J,k}(t) + \sum_{j=1}^J \sum_k d_s(j, k) \psi_{j,k}(t) \quad [4.2]$$

4.5.1 Transformada Wavelet Discreta DWT

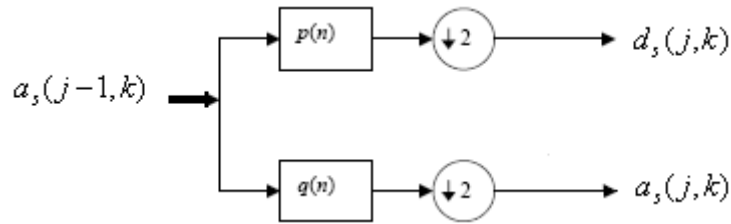
Teniendo en cuenta la ecuación [4.2], y dados los conjuntos formados por una función de escala $\phi(t)$, y su correspondiente función ondita $\psi(t)$, se puede definir la Transformada Wavelet Discreta o DWT, de una señal $s(t)$, como el conjunto de coeficientes definidos por el producto interno de la señal $s(t)$ con los respectivos conjuntos $\phi(t)$ y $\psi(t)$:

$$\text{DWT } s(t) = \left\{ \left\{ a_s(J, k), k \in \mathbb{Z} \right\}; \left\{ d_s(j, k), j = 1, 2, \dots, J, k \in \mathbb{Z} \right\} \right\} \quad [4.3]$$

Algo importante de resaltar, es que el producto interno entre la señal $s(t)$, y la base wavelet $\{\psi_{(j,k)}(t)\}$, representado por el conjunto $d_s(j,k)$, es un muestreo de la transformada wavelet continua CWT de $s(t)$, donde las muestras se toman en la denominada “grilla diádica”, $d_s(j,k) = Ts(2^j, 2^j k)$. El parámetro j , es el logaritmo en base 2 de la escala, $a = 2^j$ y se lo denomina octava, empleándose este término para referenciar una determinada escala.

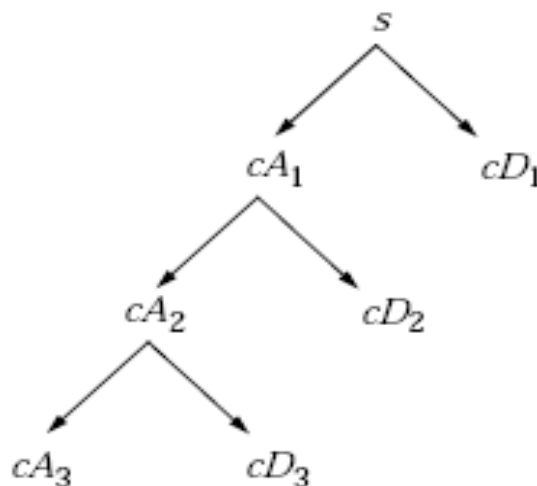
4.5.1.1 Calculo de la Transformada Wavelet Discreta

Gracias a la estructura anidada de los sub-espacios de aproximación Los coeficientes $a_s(j,k)$ y $d_s(j,k)$ se pueden calcular mediante una convolución en tiempo discreto entre la aproximación $a_s(j-1,k)$, y dos filtros digitales, uno pasa alto $p(n)$, y uno pasa bajo $q(n)$, que producen como salida los coeficientes de detalles $d_s(j,k)$, y los coeficientes de aproximación $a_s(j,k)$ respectivamente.



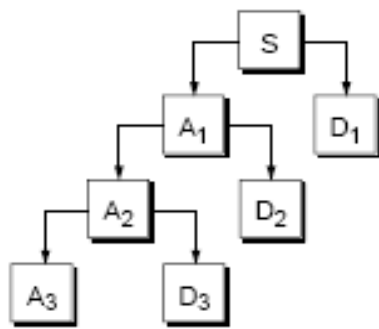
La respuesta para cada octava j , se obtiene mediante los submuestreadores ($\downarrow 2$), que eliminan una de cada dos muestras de la entrada, esto es una decimación diádica. La decimación (o submuestreo) es una técnica digital multi-rate muy eficiente para cambiar la frecuencia de muestreo de una señal en el dominio digital y comprimir eficientemente los datos. Como se indicó en la Figura, la compresión de la tasa de muestreo y reducción de datos en los coeficientes detalle $d_s(j,k)$, se logran descartando cada segunda muestra resultante del proceso de convolución, manteniendo la cantidad total de datos del vector original $a_s(j-1,k)$.

La aplicación recursiva de este método es conocida como algoritmo piramidal o de Mallat.²² Por ejemplo para una señal s , con tres iteraciones se tendría el siguiente esquema:



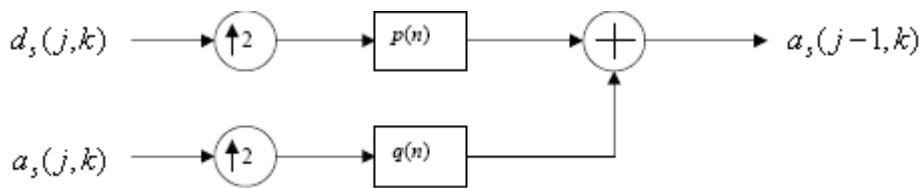
Donde cA_j , y cD_j , son los coeficientes de aproximación y detalle respectivamente, y relacionando esto con la ecuación [4.1] tenemos:

²² Stéphane Mallat autor del libro “A Wavelet Tour of Signal Processing”, ISBN : 0-12-466606-X Academic Press, 1999 - <http://www.cmap.polytechnique.fr/~mallat/book.html>



$$\begin{aligned}
 S &= A_1 + D_1 \\
 &= A_2 + D_2 + D_1 \\
 &= A_3 + D_3 + D_2 + D_1
 \end{aligned}$$

También es posible la reconstrucción de la señal usando filtros espejos y supermuestreadores que insertan un cero entre cada par de muestras, esto equivale a invertir el procedimiento:



4.6 Análisis Multi Resolución de procesos Autosimilares

Cuando se aplica MRA a un proceso estocástico, tanto las secuencias de aproximación como la secuencia de detalles son nuevos procesos estocásticos, si el proceso de entrada es estacionario de segundo orden, la transformada wavelet lo transforma en un campo aleatorio de segundo orden, siempre que las funciones de escala satisfagan ciertas condiciones respecto de la estructura de correlación del procesos de entrada. En general es suficiente con asumir que tanto la función de escala como su correspondiente wavelet, decaen suficientemente rápido en el dominio del tiempo, con esto se asegura que las estadísticas de segundo orden de la transformada existen para los procesos estocásticos de entrada que se utilizan como modelo de tráfico. A continuación se revisan algunos de los conceptos ya descritos en los capítulos anteriores a los efectos de analizar su relación con el análisis wavelet.

Un proceso $X_{(t)}, t \in \mathbb{R}$, con un factor escalar $a > 0 \in \mathbb{R}$, y un parámetro $H > 0$, que cumple para cualquier sucesión temporal de $X_{(t)}$ lo siguiente:

$$\left\{ x_{at_1}, x_{at_2}, \dots, x_{at_n} \right\} \stackrel{d}{=} \left\{ a^H x_{t_1}, a^H x_{t_2}, \dots, a^H x_{t_n} \right\} \Rightarrow X(at) \stackrel{d}{=} a^H X(t) \quad [4.4]$$

Donde el operador $\stackrel{d}{=}$, significa que tienen idéntica distribución de dimensión finita, se dice que el proceso $X(t)$, es *Hss*, o autosimilar de parámetro H . Esta autosimilitud es estricta, pero un proceso que se ajuste a esta definición no puede ser estacionario, puesto que si la media μ , y la varianza σ^2 , fueran independientes del tiempo como lo requiere la estacionariedad, tendrían que darse las igualdades $\mu = a^{-H} \mu$, $\sigma^2 = a^{-2H} \sigma^2$, y esto solo podría darse con $\mu = \sigma^2 = 0$, en cuyo caso estaríamos en presencia de un proceso que solo sería estacionario para el caso trivial $X(t) = 0, \forall t$. Un proceso puede no ser estacionario pero sí tener incrementos estacionarios, si las distribuciones de sus incrementos son finitas y no dependen del tiempo, entonces, se dice que un proceso es autosimilar con incrementos estacionarios *Hsssi*.

Como se mostró en el punto [3.11.3.1](#), el proceso FBM es el único proceso Gaussiano que cumple con la condición de ser H ssi, con media igual a cero y varianza δ^{2H} y con una Función de Autocorrelación $R_{X(k)}$, de la forma:

$$R_{X(k)} = E[X(t)X(s)] = \frac{1}{2} \left((k+1)^{2H} + (k-1)^{2H} - 2k^{2H} \right) \quad [4.5]$$

Donde k , es igual a la separación u holgura temporal $(t-s)$, es importante notar que la autocorrelación no depende del tiempo sino de k .

Como se menciona en el punto [3.9](#), del capítulo 3, la dependencia de rango largo y la autosimilitud son conceptos distintos pero íntimamente ligados y son equivalentes en el rango abierto $0.5 < H < 1$, y en el caso del FBM cuando $H = 0.5$, se convierte en un movimiento browniano que es un proceso H ss, pero su autocorrelación es cero, o sea la variable es independiente, en cambio para $H > 0.5$, empieza a tener dependencia, o sea un comportamiento LRD y este comportamiento se acentúa cuanto mas se acerque H , al valor 1.

La dependencia de rango largo es un fenómeno vinculado a procesos estacionarios y como sabemos, una manifestación de la misma en el dominio de las frecuencias, esta asociada a la densidad espectral del proceso estocástico, pues un proceso estocástico presenta LRD, si su Densidad Espectral obedece a una ley de potencia en cercanías del origen, esto es:

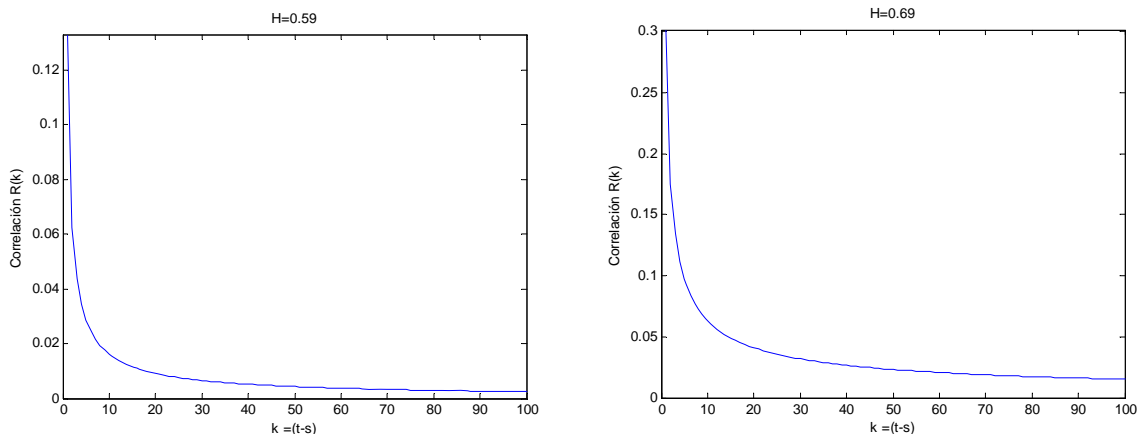
$$S(\omega) = c|\omega|^{-\gamma}, \text{ cuando } \omega \rightarrow 0 \quad [4.6]$$

El parámetro γ , esta relacionado con H , mediante la expresión $\gamma = 2H - 1$ (recordar el análisis en el punto [3.7.1](#)), c es una constante > 0 , y como se sabe la densidad espectral es por definición la transformada de Fourier de la Función de Autocorrelación, entonces la Función de Autocorrelación, satisface una relación del tipo:

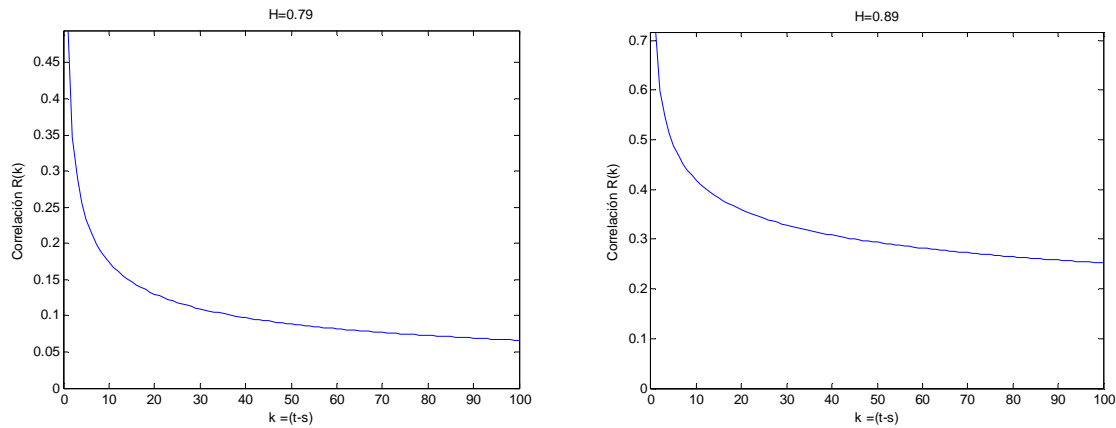
$$R(k) \approx dk^{\gamma-1}, \text{ cuando } k \rightarrow \infty \quad [4.7]$$

Las expresiones [4.6] y [4.7] son equivalentes²³, y demuestran un lento decaimiento de la correlación evidenciando LRD.

Si se hace un gráfico con la expresión de la autocorrelación del FBM en función de k , para distintos valores de H , se ve que el decaimiento de la correlación en función de la holgura temporal de la variable, es menos pronunciado conforme H , se aproxima a 1.



²³ Se recomienda consultar la referencia bibliográfica [18] Pág. 48, para examinar las dos expresiones con mayor detalle.



En la primer figura, donde $H = 0.59$, se observa un rápido decaimiento, mientras que en la última figura, donde $H = 0.89$, es evidente el apartamiento en el origen con un lento decaimiento.

4.7 Relación entre las Wavelets y los procesos Hss, Hsssi y LRD

Hay dos características de la transformada wavelet, que resultan fundamentales en el estudio de los fenómenos autosimilares y son:

1. Las bases wavelets utilizadas en el análisis multiresolución, se obtienen a partir de la dilatación de la wavelet madre, esta dilatación es en esencia una operación de cambio de escala, pues como se sabe, la escala a , esta relacionada con la frecuencia central f_c , de la wavelet madre, de manera que la base de funciones de análisis de la transformada wavelet, tiene una intrínseca invarianza a la escala.
2. La wavelet madre tiene un $N^o > 1$ de momentos desvanecientes, esto se puede expresar

como $\int_{-\infty}^{\infty} t^M \psi_0(t) dt = 0$; para $M = 0, 1, 2, \dots, N-1$, donde N , se puede escoger

seleccionando adecuadamente una determinada wavelet ψ_0 . La descomposición wavelet de una señal polinomial produce detalles nulos siempre que el número de momentos desvanecientes de la wavelet, exceda al grado del polinomio; otra forma de expresar esto es decir que el conjunto de coeficientes de detalles para un proceso $X(t)$, serán los mismos que para el proceso $X(t) + P(t)$, donde el grado de $P(t) \leq N-1$. De esta forma, sólo el ruido aparece en los detalles, en tanto que el polinomio aparece sólo en la aproximación. Esto permite definir correctamente los detalles wavelets y eliminar o disminuir la influencia de las tendencias determinísticas como pueden ser las tendencias lineales o las variaciones respecto de la media.

Una generalización de estas dos propiedades fundamentales de la wavelet para procesos *Hss*, *Hsssi* y *LRD*, surge como consecuencia de análisis matemáticos que exceden el alcance de este trabajo, pues son resultados no triviales de los mismos²⁴. Entonces se exponen las siguientes conclusiones de esos resultados.

Si $X(t)$ es proceso un proceso estocástico que presenta un fenómeno de escala, los coeficientes correspondientes a su transformada wavelet, tendrán las siguientes características:

El conjunto $\{d_x(j, k), j = 1, 2, \dots, J, k \in \mathbb{Z}\}$, es un proceso estacionario para cada octava j , si el N^o de momentos desvanecientes de la wavelet madre ψ_0 , es $N \geq \frac{\gamma-1}{2}$. La varianza de $d_x(j, k)$

²⁴ Para mayor detalle consultar la referencia bibliográfica [18] Pág. 50.

reproduce el comportamiento de escala subyacente, dentro de un rango de octavas $j_1 \leq j \leq j_2$. Dado que el valor medio de la wavelet es cero por la condición de admisibilidad, el segundo momento de $d_X(j, k)$ es proporcional a $2^{j\gamma}$, donde j_1, j_2 y γ , dependen del tipo de fenómeno de escala que exhiba el proceso original $X(t)$. Se cumplen entonces, las siguientes relaciones entre estos tres parámetros:

$$E[d_X(j, k)^2] \propto 2^{j\gamma} \quad [4.8] \quad (\text{ver al pie}^{25})$$

- a) Si $X(t)$ es *Hssi* $\rightarrow \gamma = 2H + 1$, y $-\infty < j < \infty$
- b) Si $X(t)$, presenta *LRD*, $\gamma = 2H - 1$; $j_2 = \infty$, y j_1 debe identificarse en función de los datos obtenidos en el análisis.
- c) En caso de que el proceso obedezca una ley de potencias, pero en un determinado rango de frecuencias, $f_1 \leq f \leq f_2$, (a este tipo de procesos se los denomina genéricamente procesos $1/f$), γ corresponde a la ley de potencias expresada y el rango de escalas (j_1, j_2) , debe obtenerse partiendo de las frecuencias (f_1, f_2) .

Si bien los coeficientes wavelet, reproducen el comportamiento de escala del proceso subyacente, ellos mismos no exhiben ningún tipo de dependencia estadística, si el número de momentos desvanecientes de la wavelet madre ψ_0 cumple con $N \geq \gamma / 2$, además cuando mayor sea N , menor es la correlación de $d_X(j, k)$, se puede expresar esto como:

$$E[d_X(j, k)d_X(j, k')] \approx |k - k'|^{\gamma - 2N}; \text{ a medida que } |k - k'| \rightarrow \infty \quad [4.9]$$

Esto significa que cada coeficiente wavelet, es una variable aleatoria no correlacionada con ninguna otra variable, ni dentro de su misma octava, ni en octavas superiores o inferiores.

Debido a que las proyecciones son sobre la misma wavelet madre, escalada y desplazada, se tiene que cumplir a nivel de distribuciones que:

$$\{d_X(j, 0), d_X(j, 1), \dots, d_X(j, n_j - 1)\} \stackrel{d}{=} 2^{j(H + \frac{1}{2})} \{d_X(0, 0), d_X(0, 1), \dots, d_X(0, n_j - 1)\} \quad [4.10]$$

Donde n_j , es el número de coeficientes en la octava j , teniendo en cuenta la invarianza a la escala de la distribución de los coeficientes wavelets y la propiedad estadística de la varianza descrita por la ecuación [3.10] del punto [3.11.1](#), la ecuación [4.8], se puede escribir:

$$E[d_X(j, k)^2] = 2^{j(2H+1)} E[d_X(0, k)^2] \quad [4.11]$$

Como consecuencia de estas expresiones, es factible la verificación del comportamiento de un proceso que exhiba un fenómeno de escala con un determinado parámetro H , comparando los histogramas de sus respectivos conjuntos de detalles para distintas octavas.

Concluyendo; se pueden aprovechar las características enunciadas, acomodándolas en función del contexto, tanto para una eficiente detección e identificación de fenómenos de escala, la estimación de parámetro H , o la síntesis de movimiento browniano fraccional.

²⁵ Se trata de una simplificación pues, en realidad se cumple la igualdad $E[d_X(j, k)^2] = C2^{j\gamma} : C = E[d_X(0, 0)^2]$, para profundizar en el tema, se recomienda consultar las referencias bibliográficas [11] Pág. 4, [15] Pág. 880, [18] Pág. 48, y [28] Pág. 45.

5 Estudiando el Tráfico con MATLAB®

Como se explico en los fundamentos de este trabajo, uno de los desafíos que presenta la tarea de cuantificar o medir la demanda que imponen los usuarios sobre las redes, es el desarrollo de aplicaciones con modelos de tráfico. En orden a esto, en los objetivos de esta tesis se propone el desarrollo de scripts, o aplicaciones con el software MATLAB®, producto de la firma The MathWorks™; y recordando lo manifestado en el punto [1.5](#), el software MATLAB®, cuenta con una versión [Student](#), lo que resulta conveniente para quien no lo posee y desea probarlo, esta versión puede conseguirse del sitio:

http://www.mathworks.es/academia/student_version/

Las aplicaciones desarrolladas apuntan a: Permitir una mejor comprensión de los conceptos involucrados en el tráfico autosimilar, servir de apoyo a su estudio, y posibilitar la experimentación y el desarrollo de simulaciones. El capítulo tres expone el marco teórico y los fundamentos del tráfico autosimilar; el presente capítulo, se inicia con la aplicación de esos fundamentos mediante el software MATLAB®, esto resulta de utilidad para ir desarrollando el tema, y al mismo tiempo mejorar la comprensión del mismo. Siguiendo con esa metodología, se finaliza el capítulo con la implementación de lo expuesto en el capítulo cuatro, sobre wavelets y MRA.

Para estudiar el comportamiento de tráfico en este contexto, partimos de las siguientes premisas debidamente fundamentadas en los capítulos anteriores.

- El tráfico esta representado por procesos estocásticos definidos en puntos discretos del tiempo, de tal manera que los procesos estocásticos $X_{(t)}, t \in \mathbb{Z}$, son series temporales que llamamos vectores, y cuyos componentes son los valores (agrupaciones de bytes, tiempos etc.), tomados en puntos discretos de tiempo a partir de trazas o muestras de tráfico real de una red, o también, construir el vector partiendo de trazas generadas sintéticamente que representen estadísticamente el comportamiento dichas muestras.
- El comportamiento autosimilar del proceso que representa al tráfico, esta cuantificado por el valor del parámetro H , siempre y cuando este tome valores comprendidos entre 0.5 y 1, siendo mas pronunciado el comportamiento autosimilar, cuanto mas se acerque al valor 1. Al parámetro H , se lo llama parámetro o exponente de Hurst, o también parámetro de autosimilitud.

Se emplean en forma general dos enfoques o métodos para el estudio del comportamiento o características de un proceso que represente al tráfico:

1. Cualitativos: Se parte de un proceso representado por una serie temporal y mediante el cambio de escala temporal se observa su comportamiento, y como consecuencia de estas observaciones, se hace una estimación. Esto brinda una visión general, o a priori, acerca de, si un conjunto de datos pertenecientes a una muestra de tráfico, presenta o no, características autosimilares. Por esto, si a una estimación la llamamos cualitativa, es porque no pretende establecer un valor puntual o preciso de H , sino sencillamente establecer por decirlo así, “a simple vista”; el comportamiento del proceso.
2. Cuantitativos: Métodos que presentan mayor rigor estadístico, y que precisan que los datos satisfagan las condiciones impuestas por la aplicación del procedimiento de análisis que se aplica sobre los mismos, ya sea para la determinación precisa de un valor o la síntesis de procesos con determinadas características.

5.1 Métodos Cualitativos

Conforme el primero de los enfoques mencionados, a continuación se exponen tres grupos de aplicaciones²⁶; Escalamiento Temporal, Rango Reescalado o Efecto Hurst, y Varianza/Tiempo, respectivamente. También algunas aplicaciones complementarias que permiten la manipulación de los datos y archivos de trabajo para adaptarlos al software MATLAB®.

5.1.1 Escalamiento Temporal

En el punto 3.1 se decía que al observar la representación de proceso la autosimilitud se manifiesta por la persistencia de un determinado comportamiento aún cuando escalamos una cierta dimensión, específicamente hablamos de la dimensión temporal. Una primera aproximación a este concepto sería entonces observar patrones de comportamiento, escalando el tiempo, y observando el comportamiento del proceso.

En particular, una captura de tráfico, típicamente representa agrupaciones en puntos discretos de tiempo; como ser cantidad de bytes de un paquete, o bien tiempos entre arribos de paquetes. Si observamos con una escala temporal del orden de los milisegundos, observamos el patrón de comportamiento o gráfico de las agrupaciones en puntos temporales, y vemos que presenta picos, con un comportamiento irregular o fluctuante, luego escalamos en un factor de 10 el tiempo, estando de esa forma en el orden de las centésimas de segundo y observamos nuevamente el gráfico si presenta el mismo aspecto o no, y así sucesivamente aumentando cada vez la escala, entonces, si el comportamiento observado persiste, estamos, al menos en forma cualitativa, en presencia de un comportamiento autosimilar.

Las primeras scripts de MATLAB® que se describen a continuación; permiten visualizar las trazas de una captura de tráfico en formato ASCII. Se dispone para tal efecto, de trazas de de los laboratorios Bellcore Morristown Research and Engineering, las mismas utilizadas en el trabajo de W. E. Leland and D. V. Wilson²⁷. Estas trazas ampliamente conocidas y mencionadas en muchas de las publicaciones hechas sobre este tema, están disponibles en el siguiente sitio llamado The Internet Traffic Archive:

<http://ita.ee.lbl.gov/html/contrib/BC.html>,

En particular se trata de cuatro capturas de tráfico, llamadas BC-pAug89; BC-pOct89; BC-Oct89Ext y BC-Oct89Ext4. La descripción detallada de las trazas, se pueden ver, tanto en la página del link, como en el trabajo citado.

La demostración de los métodos cualitativos que se hace a continuación, está hecha en base a la traza pAug89, procediendo de la siguiente manera:

- Se copian las scripts (todas tienen la extensión “.m”), y las trazas que vamos a utilizar en una carpeta, es importante que nos aseguremos de fijar el directorio de trabajo del MATLAB® en la misma.
- Se ejecuta la script llamada “cargaVectores”.

La funcionalidad de la script “cargaVectores”, es la siguiente: Carga la traza elegida, que en este caso es pAug89.TL, y la acomoda en dos vectores uno conteniendo la longitud de los paquetes en bytes llamado “vec”, y otro conteniendo los tiempo entre arribos de los paquetes, llamado “tiea” (por las siglas de tiempo entre arribos). Esta script, también genera o sintetiza, conforme las características del vector estudiado; otro par de vectores equivalentes, uno conteniendo la longitud de los paquetes en bytes con una distribución teórica de Poisson llamado “xpois” (pois, por Poisson), y cuya distribución presenta una intensidad característica λ calculada como

²⁶ El código fuente de todas las aplicaciones mencionadas está disponible en el apéndice 1.

²⁷ [1] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, “On the self-similar nature of ethernet traffic (extended version)”, IEEE/ACM Transactions on Networking, vol.2, pp.1–15, Feb. 1994.

regresión MLE (Maximum Likelihood Estimation), basado en los datos de los paquetes de la captura, y el otro vector, conteniendo los tiempos entre arribos, con una distribución Exponencial también calculada en base al parámetro λ mencionado, llamado “tiap” (tiempo entre arribos del proceso sintético Poisson).

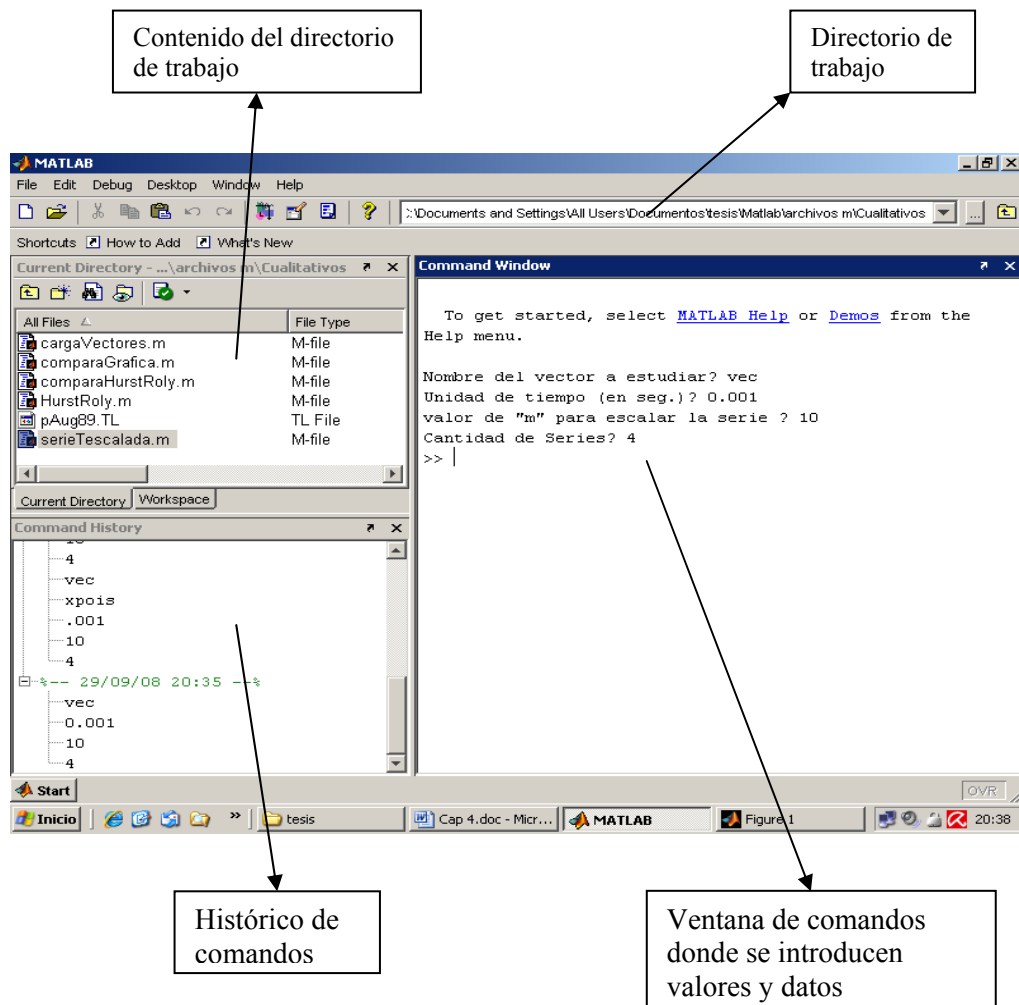
El proceso de carga de los vectores tomara un cierto tiempo, dependiendo de la potencia de proceso y memoria disponibles en la PC donde se ejecuta el software. Cabe aquí la mención de que, para una para una mejor interpretación y comprensión, el código fuente de esta script, como así también el código fuente de todas las aplicaciones que se mencionan en este trabajo, están comentados paso a paso, de manera que, si se editan las scripts con el editor incluido en el MATLAB®, los comentarios o explicaciones aparecen en color verde.

5.1.1.1 Script serieTescalada.m

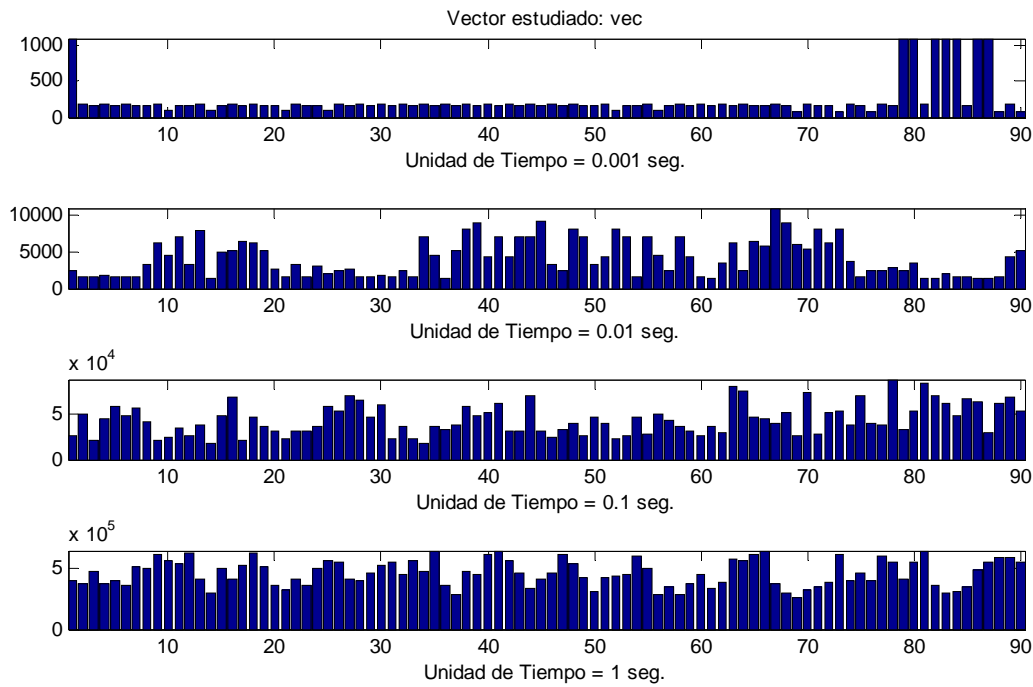
Una vez cargada la traza se procede a visualizarla, o graficarla con la script “serieTescalada”, que nos permite ver las primeras 90 agrupaciones en puntos temporales empezando con una escala de tiempo del orden del milisegundo, para eso contestamos las preguntas que nos hace la script, con “vec”, “0.001”, “10” y “4” respectivamente (ver figura). Una vez ejecutada la script, muestra o grafica el vector “vec”, cuyo contenido es la longitud de los paquetes en bytes, con una escala inicial de milisegundos, y luego, tres series mas, cada una de ellas, escalada por un factor de 10, y de esta forma se obtendrá un gráfico en pantalla donde se muestra al proceso en escalas de tiempo del orden de los milisegundos, centésimas de segundo, décimas de segundo y segundos respectivamente. Sugiero que en forma simultánea a su utilización se edite la script para analizar su contenido y leer los comentarios lo cual facilita comprender su funcionamiento.

A continuación se pueden observar capturas de pantalla ejecutando la script:

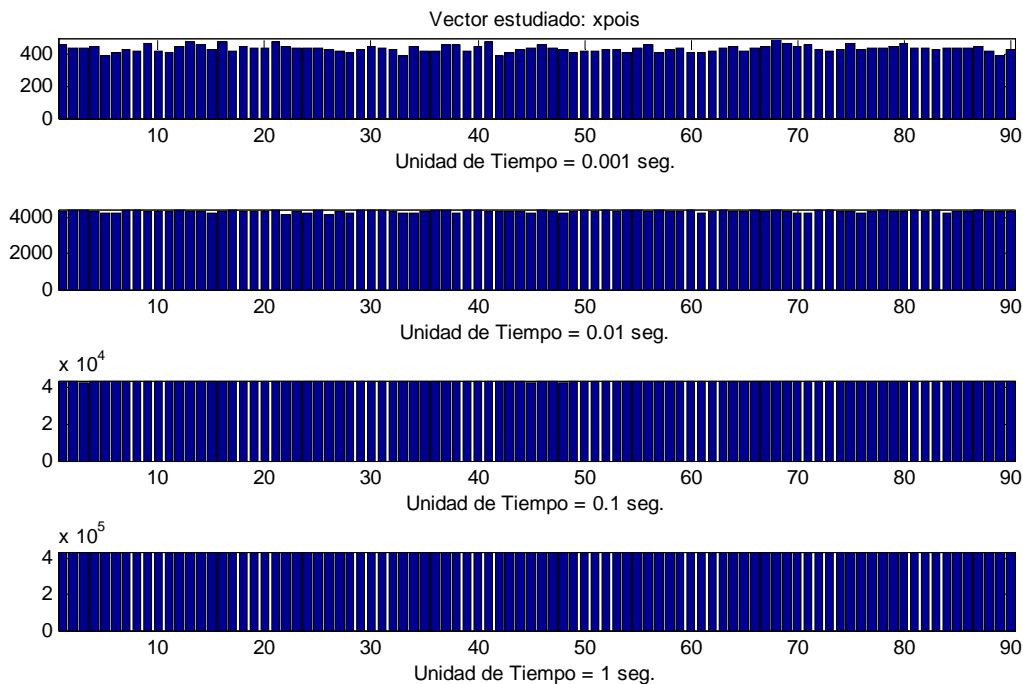
Aspecto de la pantalla de MATLAB®



Luego de ingresar el último valor, se obtendrá el gráfico que se observa en la siguiente figura, que corresponde a las agrupaciones de paquetes en bytes en el eje Y; y los valores de tiempo en el eje X



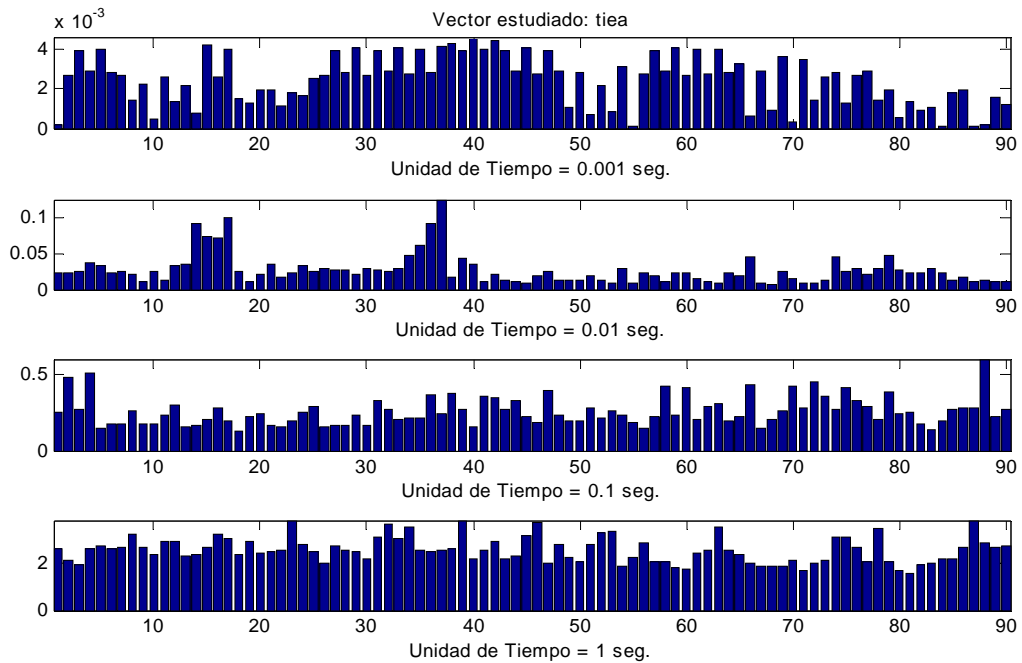
Como se puede observar, el aspecto del proceso en las cuatro escalas de tiempo es similar y fluctuante, en contraste; si ejecutamos nuevamente la script pero ahora con el vector “xpois”; escala de tiempo “0.001”, factor m = 10, y 4 series, respectivamente, se obtendrá la siguiente gráfica:



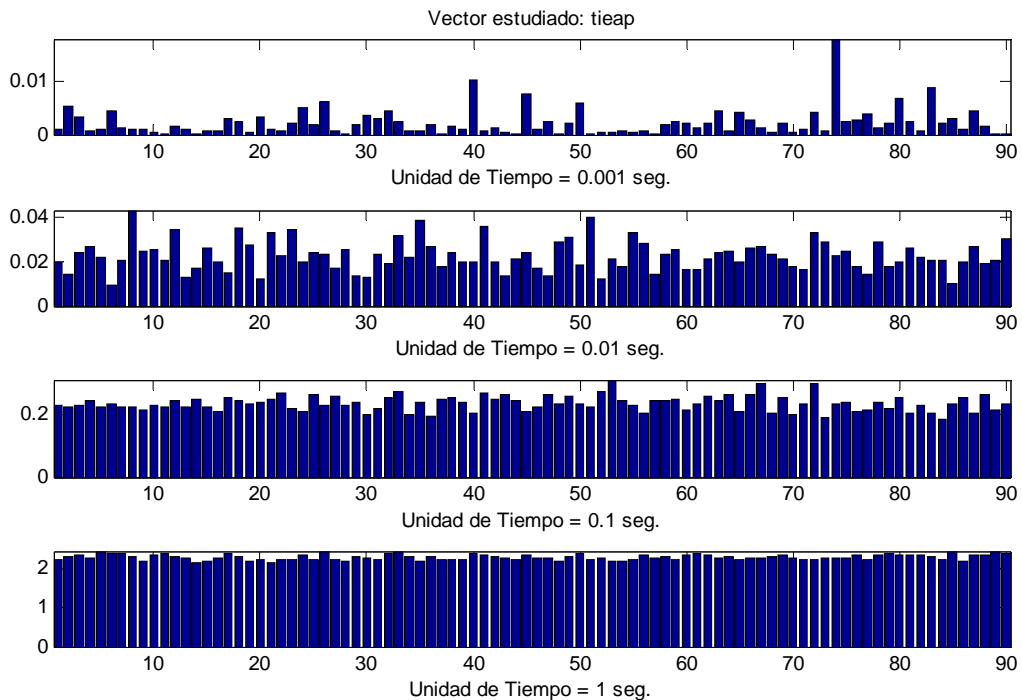
Como se observa en este caso, la fluctuación tiende a desaparecer y enseguida de aplana, apenas notándose en las centésimas y ya casi imperceptible en las décimas y los segundos, esto

evidencia el típico comportamiento del tipo Poisson o “sin memoria”, este comportamiento contrasta con el observado en la anterior pantalla.

Siguiendo la misma metodología, se ejecuta nuevamente la script, pero ahora con los tiempos entre arribos, llamados “tiea” y “tieap”, que corresponden al vector con los datos de la captura y al calculado teóricamente con distribución exponencial respectivamente. En el vector con los datos reales de la captura “tiea”, se observa un comportamiento fluctuante en todas las escalas:



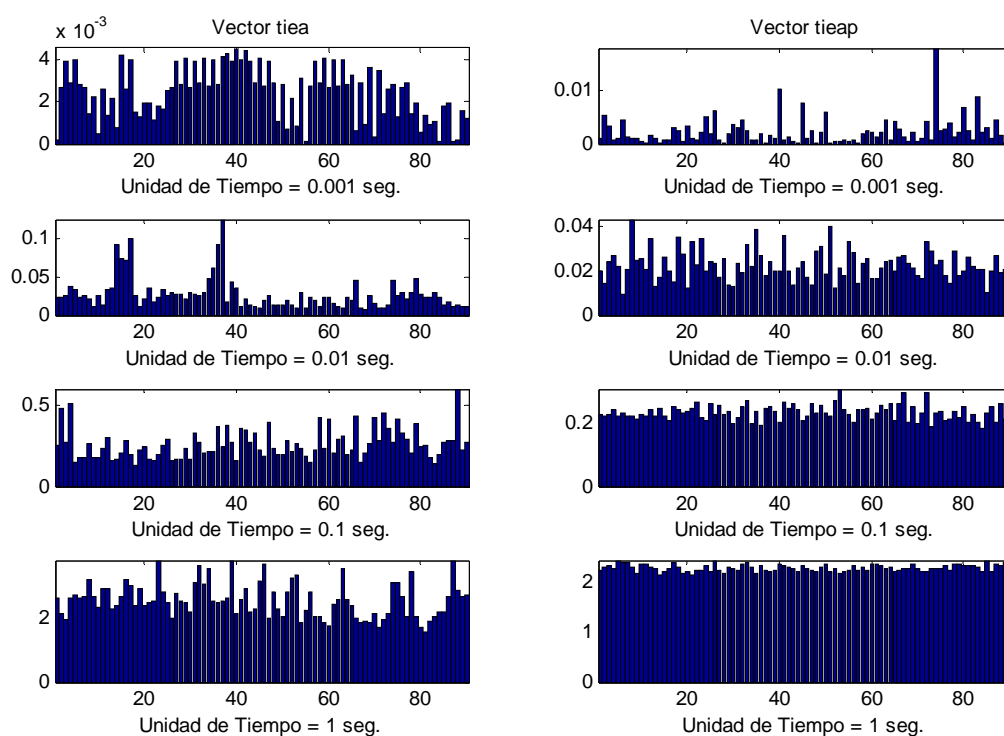
Ahora con el vector teórico “tieap” que sabemos esta exponencialmente distribuido, se observa como se suaviza el comportamiento a medida que aumenta la escala de tiempo, tal como es de esperar para valores con esta distribución.



Se pueden cambiar a voluntad los parámetros introducidos y ver los resultados en pantalla, por ejemplo si se disminuye el factor “m”, se puede aumentar la cantidad de series a graficar, y observar de esta forma si es gradual, o no, la disminución de la fluctuación, recordar que cuando mas grande es el factor de escala “m”, resultan menos puntos en el vector escalado, pues se parte de una longitud fija inicial para ir promediando los valores en cada punto, de todas formas la script tiene una rutina de protección por si se introduce una cantidad de series que sea incompatible con el factor de escala “m”, que se haya escogido y nos sugerirá el tamaño máximo apropiado para la cantidad de series, en función del “m”, que se haya escogido.

5.1.1.2 Script comparaGrafica.m

Resulta interesante también ejecutar la script llamada “comparaGrafica”, que grafica dos vectores que se introducen como datos, y permite también, introducir el valor de “m” o factor de escala para poder determinar el escalado o agregación de los mismos para visualizar el efecto causado, en sendos vectores al escalarlos simultáneamente por el mismo factor. Esto resulta útil si se quiere comparar vectores con valores que obedecen a distintas distribuciones, o sencillamente ver que pasa con distintos procesos cuando se los escala por un cierto factor “m”. En la siguiente figura, se muestra la pantalla con la salida de esta script, para los vectores “tiea” y “tieap”.



En forma evidente, se aprecia a la izquierda el comportamiento fluctuante en todas las escalas de los datos del tráfico real, en contraste con lo que sucede a la derecha con el proceso generado en forma sintética, que “suaviza” su comportamiento a medida que aumenta la escala de tiempo.

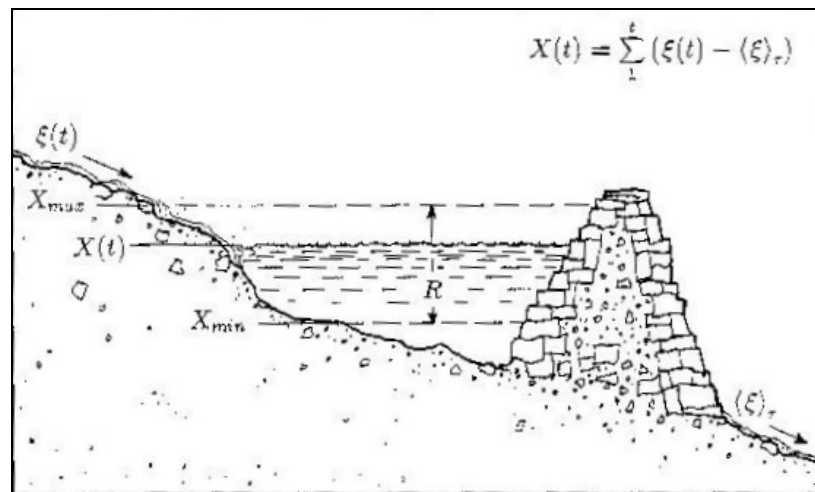
También puede resultar útil, el uso de “comparaGrafica”, para ver simultáneamente el comportamiento de la agrupación de los paquetes y los tiempos entre arribos de una determinada captura, a medida que se escala el tiempo.

Resumiendo, la utilidad de las scripts “serieTagregada” y “comparaGrafica”, es poder apreciar en forma cualitativa o visual el comportamiento de un determinado proceso, y en base a la observación, determinar si la evolución o escalado del proceso, es compatible con un comportamiento autosimilar. A continuación se considera la estimación del parámetro Hurst como un indicador de autosimilitud, con el desarrollo de nuevas scripts, basadas en las que se acaban de comentar.

5.1.2 Grafico Rango Reescalado R/S - El efecto Hurst

Hasta aquí se vino mencionando al parámetro de Hurst “ H ”; como una medida de la persistencia o intensidad del fenómeno de la autosimilitud en un proceso estocástico; a continuación se comenta brevemente su origen e historia. Este parámetro toma su nombre en honor al ingeniero Harold Edwin Hurst (1 Enero 1880 -7 Diciembre 1978), quien dedico su vida a explorar el río Nilo junto a otros ríos, y realizar estudios en relación a problemas de almacenamiento de agua en las represas. Publico trabajos como, "Long Term Storage Capacity of Reservoirs," Transactions of the American Society of Civil Engineers, 116, 770-799, año 1951; o “Long-Term Storage: An Experimental Study”, del año 1965. Hurst descubrió que, en contraste con otros ríos, los niveles de agua del Nilo en período de 800 años seguían un patrón de variabilidad autosimilar, y consecuentemente desarrollo una medida normalizada sin dimensión para caracterizar esa variabilidad, llamándola rango reescalado. El método desarrollado por Hurst permite hacer una estimación de “ H ”, y consiste en lo siguiente.

Se aborda el problema del diseño de una presa para regular el flujo del agua, basado en los registros disponibles (serie temporal), con un modelo de presa ideal, en la que el flujo de salida es constante e igual a la entrada promedio, pero considerando que la presa nunca se desborda y tampoco nunca se vacía. En este diseño se necesita conocer la medida de la variabilidad del flujo de agua, y como se dispone de los registros anuales se plantea el problema como se muestra en la siguiente figura²⁸.



- La variable $\xi(t)$, representa al ingreso de agua al reservorio.
- $[\xi](\tau)$, representa la salida de agua promedio en τ años.
- $X(t) = \sum_{i=1}^t \xi(t) - [\xi(\tau)]$, representa el nivel del agua, al año t .
- X_{max} , representa el máximo nivel acumulado hasta el año t .
- X_{min} , representa el mínimo nivel acumulado hasta el año t .
- R , representa el rango de variación entre el nivel máximo y el mínimo del reservorio hasta el año t .

Como se puede ver la serie temporal $X(t)$, representa en un determinado momento t , la suma acumulada de las diferencias entre el valor puntual y el valor promedio de salida de agua, se puede considerar que es la cantidad de agua acumulada por el valor de la cual se desvía la serie temporal respecto de su media hasta el momento t , entonces si se toma el valor máximo, y el valor mínimo, se obtiene por diferencia el valor del rango R , representado por:

²⁸ Figura perteneciente al libro Feder Jens. "Fractals". New York, NY: Plenum Press, 1988.

$$R(t) = \left[\max \sum_1^t X(t) - [X(t)] - \min \sum_1^t X(t) - [X(t)] \right]$$

En cierta forma esto representa la variabilidad de X , respecto de su media.

Siendo $S(t)$, la desviación estándar de $X(t)$, $S(t) = \sqrt{\frac{1}{t} \sum_{\tau=1}^t (\xi(t) - [\xi(\tau)])^2}$, se normaliza la medida

del rango dividiendo R , por el valor de S . y se obtiene la medida $\frac{R}{S}$, a la que Hurst dio el nombre de rango reescalado, pues en sus investigaciones, encontró que para muchos fenómenos naturales como ser, la descarga de los ríos la acumulación de sedimentos, los anillos de los árboles, y algunas series estadísticas, la relación empírica $\frac{R}{S} \sim \left(\frac{N}{2}\right)^H$ con $H > 0,5$, para grandes valores de N , describe bastante bien el fenómeno observado.

Aplicado logaritmos y sus propiedades, esta relación puede ser escrita de la siguiente manera:

$$\log \left[\frac{R}{S} \right] \sim H \log(N) - H \log(2)$$

Como se puede ver, si se representan los puntos $\log \left[\frac{R}{S} \right]$, frente $\log(N)$, los mismos, se ajustarán a una línea recta de pendiente H , y una ordenada al origen igual a $H \log(2)$. Este análisis recibe el nombre de estadístico R/S , o también grafica R/S .

Basado en esto se puede desarrollar un estimador del valor de H , calculando distintos puntos del rango reescalado y luego mediante una regresión lineal, calcular el valor de la pendiente que representa al valor de H .

5.1.2.1 Script rangoRescalado.m

Para implementar este método con MATLAB®, se desarrolla una script llamada “rangoRescalado”, la funcionalidad de la misma es la siguiente: A partir de un vector con los datos, se muestrea el proceso estudiado, para varios tamaños de muestra “ N ”. Calcula para cada muestra el valor de la razón de rango reescalado R/S . Calcula el $\log(R/S)$ y $\log(N)$, graficando los puntos. Estima el valor de H por regresión lineal aplicado el método de mínimos cuadrados. Finalmente entrega como salida la grafica R/S con los puntos calculados, la recta de la regresión y la cantidad de muestras tomadas para los cálculos.

El algoritmo para calcular los puntos para graficar es el siguiente:

- Fijar un tamaño de muestra que llamamos (m).
- Calcular el valor medio de la muestra $\bar{X} = [X(m)]$.
- Calcular la desviación estándar de la muestra $S(m)$.
- Calcular las desviaciones puntuales $[x_i - \bar{X}]$.
- Calcular la suma acumulada $\sum_{i=1}^m x_i - \bar{X}$.
- Hallar el máximo y el mínimo valor de la suma acumulada.

- Calcular $R(m) = \left[\max \sum_{i=1}^m x_i - [X(m)] - \min \sum_{i=1}^m x_i - [X(m)] \right]$
- Hallar la razón $\frac{R(m)}{S(m)}$
- Registrar los valores de $\log \left[\frac{R(m)}{S(m)} \right]$ y $\log(m)$
- Tomar un nuevo tamaño de muestra $(m) = \frac{m}{2}$
- Comenzar de nuevo.

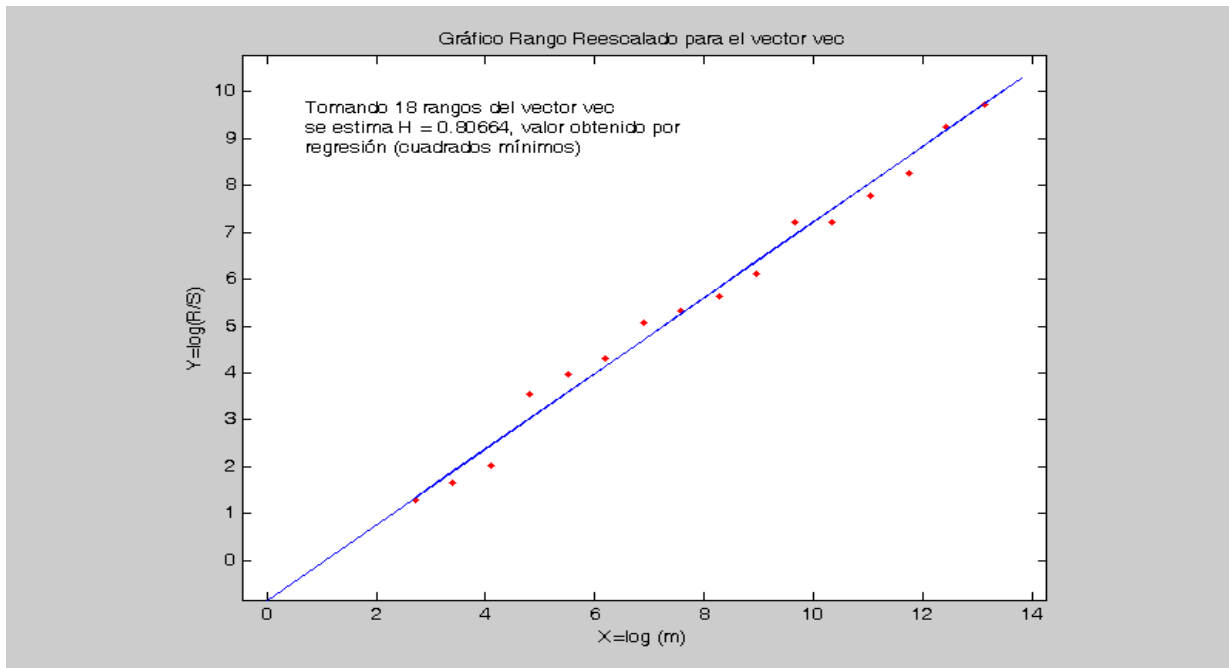
La siguiente figura, ilustra como se toman los tamaños de las muestras:

m_0										$m_0 =$ Cantidad total de datos Muestra inicial		
m_1										$m_1 = \frac{m_0}{2}$		
m_2											$m_2 = \frac{m_1}{2}$	
m_3												$m_3 = \frac{m_2}{2}$
m_4												$m_4 = \frac{m_3}{2}$

En general si m_0 es la cantidad total de datos, y en cada iteración se divide por 2, la n ésima muestra m_n tendrá $\frac{m_0}{2^n}$, cantidad de datos (se considera este cociente un número entero, pues si el cociente no da un número entero se aproxima al entero inmediato inferior), esto equivale a decir que disminuyo su tamaño en un factor igual a 2^n , donde n indica la cantidad de iteraciones o muestras que tomamos para iterar el procedimiento.

En principio, se puede iterar el procedimiento hasta que el tamaño de la muestra lo permita es decir no tome la forma indeterminada de cero sobre cero (recordar que se esta calculando un cociente donde el numerador R es una diferencia entre sumas acumuladas, que puede dar cero si la cantidad de elementos de la muestra es pequeña, lo mismo para el denominador S que es la desviación estándar de dicha muestra).

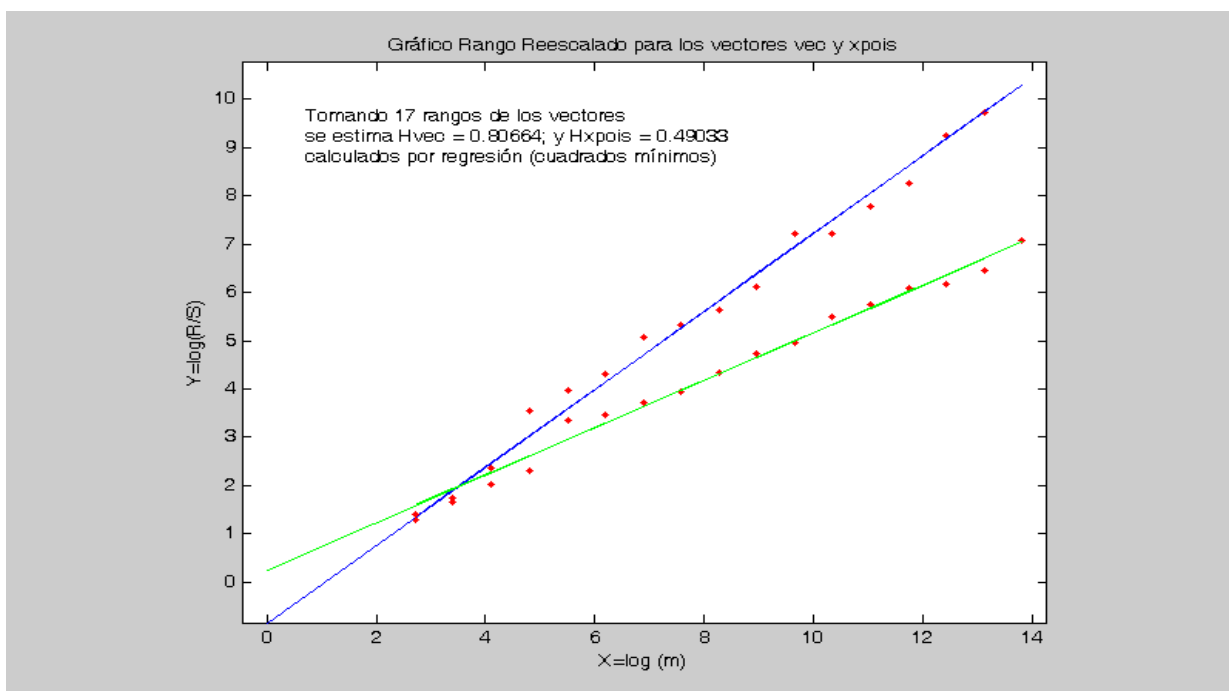
Para fijar un valor de corte, vemos que el mismo se puede calcular partiendo de $\frac{m_0}{m_n} = 2^n \Rightarrow \log_2 \left(\frac{m_0}{m_n} \right) = n$. Siendo n el número de iteraciones o veces que tomamos un nuevo tamaño de muestra. Específicamente en la script desarrollada que tiene el nombre de “rangoRescalado”, el valor de corte se fija de tal manera que el tamaño de muestra mínimo tenga por lo menos 8 valores, esto significa que para un vector con m_0 cantidad de datos; tendremos $\log_2 \left(\frac{m_0}{8} \right)$ cantidad de iteraciones o tamaños de muestra (este valor debe ser un número entero), empezando con a totalidad de los datos y disminuyendo a la mitad el tamaño de la muestra para cada iteración. La script esta comentada de forma que pueda ser estudiada y analizada.



La figura muestra la salida de la script “rango Rescalado”, para el vector “vec”, donde se puede ver que se tomaron 18 muestras, partiendo de la totalidad de los datos y luego disminuyendo a la mitad el valor del tamaño de la muestra en cada iteración.

5.1.2.2 Script rangoRescaladocomparado.m

También se desarrollo una script llamada rangoRescaladocomparado.m, que hace el mismo trabajo que la anterior pero comparando dos vectores y cuya salida se muestra a continuación.

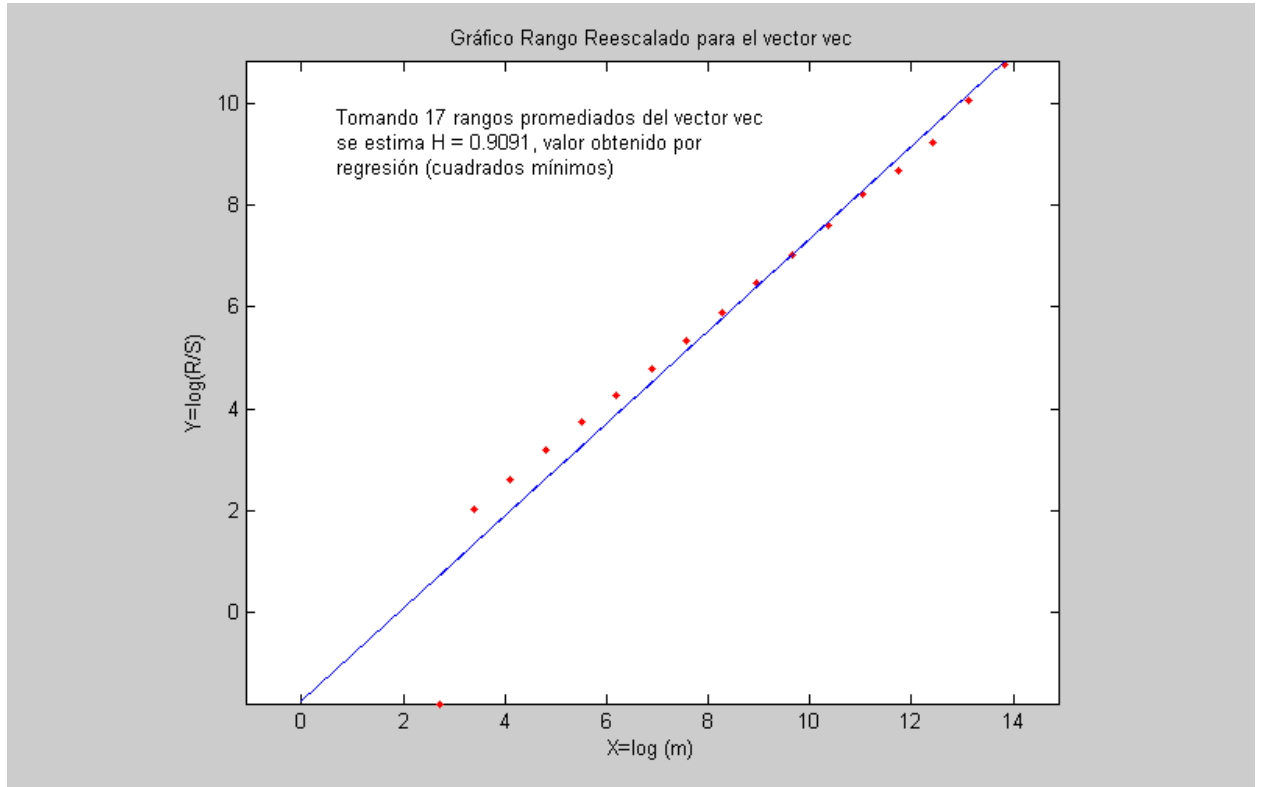


5.1.2.3 Script rangoRescaladopromedio.m

Otro desarrollo, consiste en una script llamada rangoRescaladopromedio.m, la cual es idéntica rangoReescalado.m, pero promedia el valor de todas las muestras para hacer la estimación. En el caso de la script rangoRescalado.m, solamente se toma el valor de una muestra de tamaño “m”, y para rangoRescaladopromedio.m, se toma el valor promedio de todas las muestras de tamaño “m” (se toman todos los datos existentes en el vector traza en bloques de tamaño “m”, y se los

promedia). El motivo para desarrollar esta script es poder comparar los resultados de la estimación tomando solamente una muestra o el promedio de todas las muestras para un determinado tamaño de bloque “m”; esto se hace en orden a que en algunos trabajos publicados sobre éste método, se sugiere que el método mejora promediando todas las muestras.

La siguiente figura, es la salida de rangoRescaladopromedio.m, para el vector “vec”, que representa a la traza pAug89.TL, y como se puede apreciar el valor del estimado de H , es 0.9091, resultando superior al estimado con rangoRescalado.m, que para la misma traza, arrojó un valor de 0.80664.



Como conclusión, se observa en general en las trazas estudiadas, que la estimación del valor de H , cuando se utiliza rangoRescaladopromedio.m, arroja un valor levemente superior que cuando se utiliza rangoRescalado.m, de todas formas se debe tener en cuenta que éste método es considerado cualitativo y no cuantitativo es decir el valor estimado no debe tomarse como el verdadero valor de H , sino como indicación o tendencia del proceso estudiado a un comportamiento compatible con ese valor.

Se sugiere al lector, probar estas scripts para distintas trazas y así poder contrastar los resultados y verificar lo comentado en estos párrafos y sacar sus propias conclusiones.

5.1.3 Gráfico Varianza Tiempo

Como se explico en el capítulo tres, un indicador de comportamiento autosimilar es el Parámetro de Hurst “ H ”, y en el punto 3.2.6 se dice que la autosimilitud asintótica de segundo orden se manifiesta en un proceso $\{X\}$, y sus procesos “m” agregados definidos como:

$$\{X_k^{(m)}\} = \frac{1}{m} \sum_{i=km-(m-1)}^{km} X_i$$

Si se cumple la condición que:

$$Var(X^{(m)}) \sim \frac{Var(X)}{m^\beta}$$

En esta expresión el operador \sim significa que son asintóticamente similares, por otra parte se sabe que el parámetro β está relacionado con el parámetro de Hurst de la siguiente forma:

$$H = 1 - \left(\frac{\beta}{2}\right).$$

Entonces si aplicamos logaritmo a la expresión nos queda:

$$\log Var(X^{(m)}) = \log Var(X) - \beta \log(m)$$

En estas condiciones, si se representan los distintos puntos que surgen de ir escalando por “m”, la variable X en un par ejes, donde las abscisas están representadas por $\log(m)$, y las ordenadas por $\log Var(X^{(m)})$, veremos que se agrupan en torno a una línea recta de pendiente $-\beta$ y cuya ordenada al origen es la constante $\log Var(X)$. A este tipo de gráfico se lo denomina Varianza Tiempo.

Entonces, si se hace la regresión correspondiente y se calcula el valor de β , se puede estimar valor de H , de la siguiente forma:

$$H = 1 - \left(\frac{\beta}{2}\right).$$

Sabiendo que estas son aproximaciones asintóticas, y por tanto el valor hallado puede tener un importante sesgo, sobre todo si tenemos en cuenta que la serie es finita; podemos tomar este procedimiento, como un estimador cualitativo, pues lo que se hace en realidad, es muestrear la serie o proceso original X con un tamaño de muestra “m”, idealmente para procesos ergódicos donde la variable es independiente e idénticamente distribuida, los promedios temporales son iguales a los promedios grupales esto implica que $Var(X)$ muestreada con valores de “m”, es igual a $Var(X^{(m)})$ y $\beta = 1$; indicando que el proceso es independiente, entonces los puntos se agruparán en torno a una recta de pendiente -1 lo que implica un $H = 0,5$, pero en procesos que presentan dependencia de rango largo, los puntos se agrupan en pendientes con $0 < \beta < 1$, por lo tanto H toma valores $0,5 < H < 1$, evidenciando un comportamiento autosimilar mas pronunciado, cuanto mas se aproxime β a cero, o lo que es lo mismo H se aproxime al valor 1.

Como lo comenta Stallings [36] Pág. 232, se puede ver a las series “m” agregadas como una forma de comprimir el tiempo, dándose el mayor nivel de detalle cuando $m = 1$, pero puedo comprimir el tiempo en un factor “m”, perdiendo detalles debido al promedio, y si en estas condiciones vemos que las estadísticas del proceso se mantienen, se pone en evidencia el comportamiento de un proceso autosimilar.

5.1.3.1 Script varianzaTiempo.m

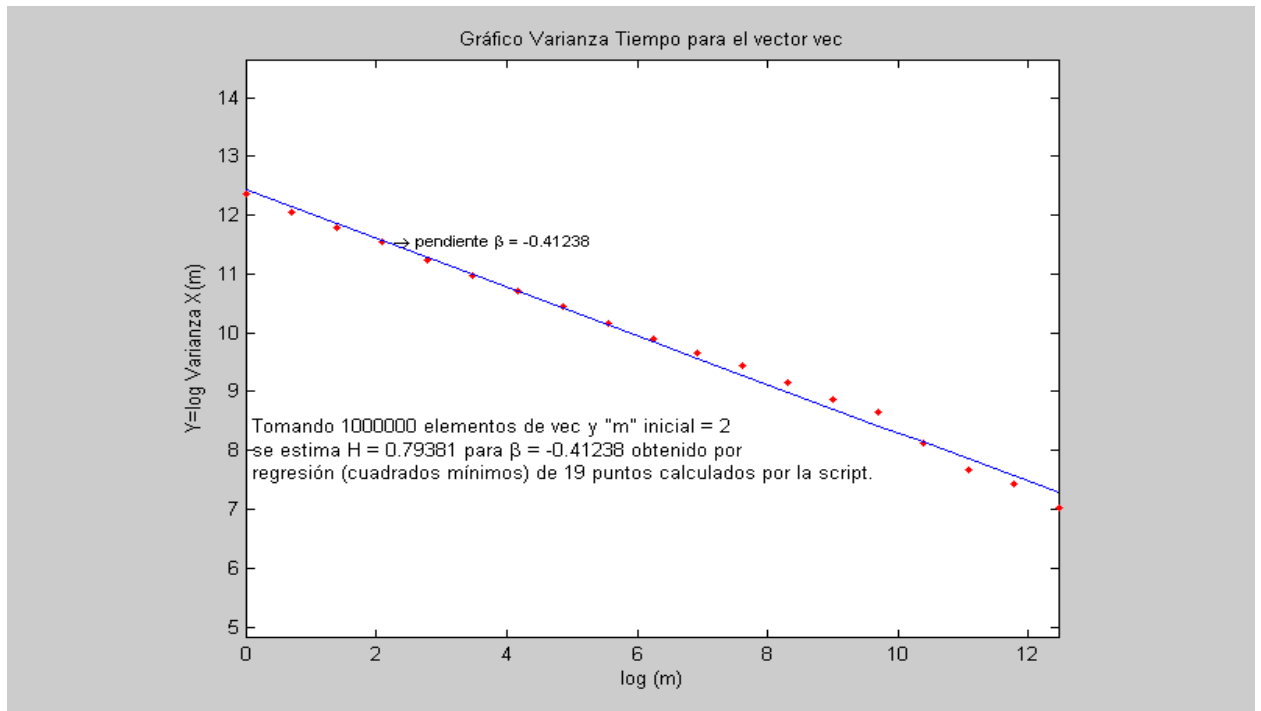
Para implementar lo comentado, se creo un script llamada “varianzaTiempo.m”, se la desarrolló partiendo de las scripts comentadas anteriormente en el punto [5.1.1](#), pero con la modificación de promediar las agrupaciones que se producen al escalar el tiempo, por un factor “m”, que se va incrementado para construir las nuevas series “m” agregadas, a las cuales se les va aplicando la variación y de esta forma se obtienen los puntos para hacer la gráfica y simultáneamente, aprovechando la potencia de calculo del MATLAB®, se hace la correspondiente regresión y el cálculo del la estimación de H .

La funcionalidad de la script es la siguiente:

En primer lugar se fijan los parámetros de entrada, esto es:

1. Se escoge un vector con los datos.
2. Se fija un valor de “m” para producir el escalado.

Considerando la cantidad de elementos que contenga el vector con los datos, y el valor “m”, que se haya escogido para escalar; la script calcula la cantidad máxima de puntos que es posible graficar para esa combinación de entrada, esto es un punto por cada serie “m” agregada que se genera al escalar “m” (esto se comenta en detalle en el siguiente párrafo), finalmente grafica los puntos calculados, y la recta de la regresión efectuada por el método de mínimos cuadrados, mostrando un valor estimado para H , y el estado de los datos de entrada que produjeron la salida o gráfica.



En la figura se muestra la salida de la script “varianzaTiempo”, para el vector “vec”, con datos de la captura “pAug89”.

Al analizar los detalles de la script, se observa que el calculo de los puntos del grafico, se hace calculando logaritmo de la varianza de la serie “m” agregada, esto significa que si la varianza es cero, al calcular el logaritmo aparecerá un error, pues no esta definido el logaritmo de cero, entonces si se analiza cual es el valor máximo que puede tomar el factor de escala o agregación (recordar que es como hacer un muestreo del vector con tamaño de la muestra igual a m), para calcular las series, se infiere que es aquel cuyo resultado da un solo elemento es decir que “m” e igual a la cantidad de elementos que tiene $X^{(m)}$, en la script, el calculo se realiza incrementando el factor de escala para cada nueva serie como una potencia, esto es: $m_{cs} = m^{cs}$, donde cs variara desde 1, que es el valor que toma para calcular el primer punto o primer serie, hasta aquel valor de cs, que haga igual a uno el cociente entre la cantidad de elementos, y m^{cs} , esto es:

$$\frac{\text{cantidad de elementos}}{m^{cs}} = 1$$

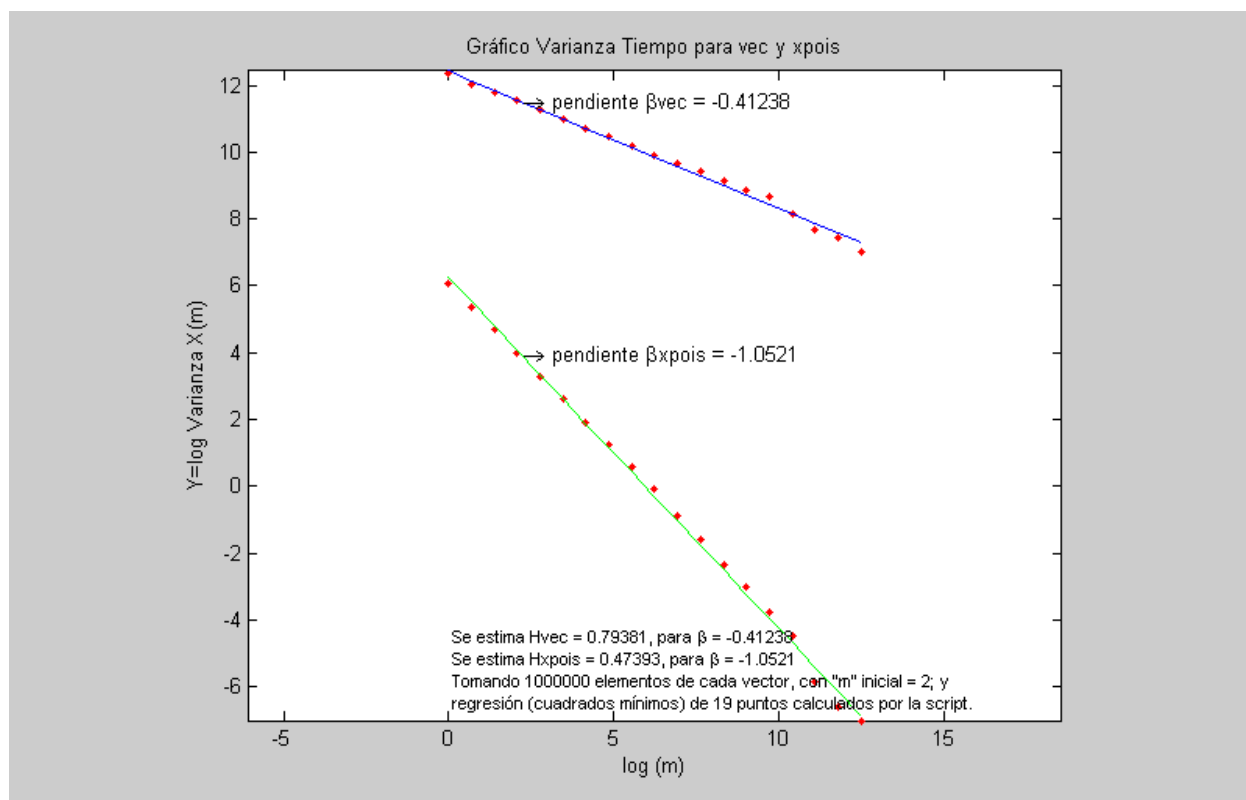
Entonces, aplicando logaritmo y sus propiedades a ambos miembros y despejando el valor de cs se obtiene:

$$\log\left(\frac{\text{cantidad de elementos}}{m^{cs}}\right) = 0 \rightarrow \log(\text{cantidad de elementos}) = cs \log(m) \Rightarrow cs = \frac{\log(\text{cantidad de elementos})}{\log(m)}$$

Entonces “cs”, es igual a la cantidad máxima de series m agregadas que se puede calcular con esta script, partiendo de un vector de longitud d, donde d es la cantidad de elementos que contiene, y un factor de escalamiento que se llama “m”. Analizando la script, en el bucle que calcula cada serie “m” agregada, se observa que esta gobernado por una variable llamada “cs”, y su valor está calculado de la forma aquí descrita en función de la longitud del vector de datos y el valor inicial que se da a “m”, siendo éstos los parámetros de entrada para la script.

5.1.3.2 Script varianzaTiempocomparado.m

Siguiendo la práctica de establecer comparaciones, para este método también se desarrollo una script llamada “varianzaTiempocomparado.m”, que permite estimar el parámetro de Hurst para dos vectores (procesos), o bien dos porciones diferentes de un mismo proceso, como podría ser una captura de tráfico de gran magnitud o extensión (varios millones de paquetes); y luego disponer de los resultados en un mismo gráfico, a los efectos de establecer comparaciones entre ambos; esto es válido también para “rangoRescaladocomparado.m”, o bien “comparaGrafica.m”.



En la figura se muestra la salida de “varianzaTiempocomparado”, para los vectores “vec” (representando aquí a pAug89) y “xpois”, utilizados anteriormente para los otros ejemplos.

5.2 Métodos Cuantitativos

A continuación, se desarrollan cinco aplicaciones, dos de estimación, dos de síntesis y una de análisis, orientadas al segundo enfoque mencionado al iniciar este capítulo, con el objetivo de obtener valores con aceptable nivel confiabilidad.

5.2.1 Diagrama Log-escala – Estimación de H

Como se mostró en el Capítulo 4, la gran ventaja estadística del análisis Wavelet en el dominio de las escalas se evidencia en la expresión:

$$E[d_X(j,k)d_X(j,k')] \approx |k - k'|^{\gamma-1-2N}; \text{ a medida que } |k - k'| \rightarrow \infty$$

Esto nos permite medir los promedios temporales y utilizarlos como estimaciones de los promedios estadísticos, y la falta de correlación entre los distintos coeficientes wavelets asegura que los estimadores temporales tengan una varianza pequeña.

Por otra parte la expresión $E[d_x(j,k)^2] = 2^{j(2H+1)} E[d_x(0,k)^2]$; puede tomarse como un estimador del espectro de potencia del proceso en las inmediaciones de la frecuencia correspondiente a la octava j , por analogía a lo visto en los puntos [3.3.4](#) y [3.7](#), referidos a la autocovarianza y la densidad espectral respectivamente, esto conduce a que se pueda estimar la varianza del proceso $d_x(j,k)$, como muestra la siguiente expresión:

$$\mu_j = \frac{1}{n_j} \sum_{k=1}^{n_j} |d_x(j,k)|^2$$

Donde n_j es el número de coeficientes en la octava j , y se observa que la varianza decrece conforme n_j aumenta, entonces la variable μ_j es una forma eficiente de representar en forma compacta el comportamiento de segundo orden del proceso estudiado $X_{(t)}$ en la octava j , y si se tiene en cuenta la expresión, los μ_j son prácticamente independientes entre sí generando un desacoplamiento del comportamiento de $X_{(t)}$ en las distintas octavas j , además teniendo en cuenta lo expresado en el punto [4.7](#), y dado que en el estimador la varianza decae hiperbólicamente en función de la octava j , se puede asumir que el exponente de escala del proceso γ , lo podemos estimar como una regresión lineal de $y_{(j)} \equiv \log_2(\mu_j)$, como una función de la octava j .

En general se puede afirmar que un proceso que presenta LRD tiene una densidad espectral que obedece a:

$$S(\omega) \sim \frac{c}{|\omega|^\gamma}, \text{ cuando } \omega \rightarrow 0$$

Donde ω es la frecuencia; c es una constante positiva, y $\gamma = 2H - 1$.

Como se expreso en la ecuación [4.8] (y su correspondiente nota al pie número 25), vemos que se pueden establecer equivalencias en ambas expresiones y expresar lo siguiente:

$$\log_2(E[d_x(j,k)^2]) = j\gamma + \log_2(c)$$

Lo que nos lleva a la siguiente aproximación:

$$\log_2(\mu_j) = j\gamma + \log_2(c)$$

A la grafica de esta recta de regresión acompañada de los correspondientes intervalos de confianza en cada punto calculado, se la conoce como Diagrama Log-escala y de la pendiente se puede extraer el valor de γ y despejar el valor de H .

En realidad lo expuesto hasta aquí es el fundamento básico del método, donde la pendiente γ , se puede estimar en la región del diagrama donde los puntos se alinean entre dos octavas j_1, j_2 , dado que es posible que no exista alineamiento en otras regiones, pero en realidad se acompaña de otros refinamientos pues como se mostró en el capítulo 4 en las dos propiedades fundamentales de la wavelet para procesos Hss , $Hssi$ y LRD , los factores j_1, j_2 y γ , dependen del tipo de fenómeno de escala que exhiba el proceso original $X(t)$.

Hay un par de trabajos mencionados en las referencias bibliográficas como: [11] Patrice Abry and Darryl Veitch, Wavelet Analysis of Long-Range-Dependent Traffic, IEEE Transactions On Information Theory, Vol. 44, No. 1, January 1998; y [30] P. Abry, P. Flandrin, M. S. Taqqu, and D. Veitch. Self-similarity and longrange dependence through the wavelet lens. In P. Doukhan, G.

Oppenheim, and M. S. Taqqu, editors, Theory and Applications of Long-Range Dependence, pages 526–556. Birkhäuser, 2003. Se recomienda su consulta, a quienes deseen profundizar mas en los fundamentos de este procedimiento, pues tratan con gran detalle lo aquí expuesto.

El código para MATLAB® descrito y utilizado aquí para la estimación de H , pertenece a Darryl Veitch, y permite la elección arbitraria de los parámetros j_1, j_2 como así también el N° de momentos desvanecientes de la wavelet madre utilizada para el análisis lo que confiere gran flexibilidad y al mismo tiempo permite experimentar de manera sencilla logrando con esto una buena comprensión de cómo se relacionan los parámetros, y que efectos causan en la estimación, la variación de los mismos.

Estas circunstancias, se ajustan a la perfección a los objetivos de esta tesis por ello escogí el conjunto de scripts desarrollados por Veitch, quien generosamente los pone a disposición para su uso en forma libre como lo establece en las condiciones expuestas en el sitio:

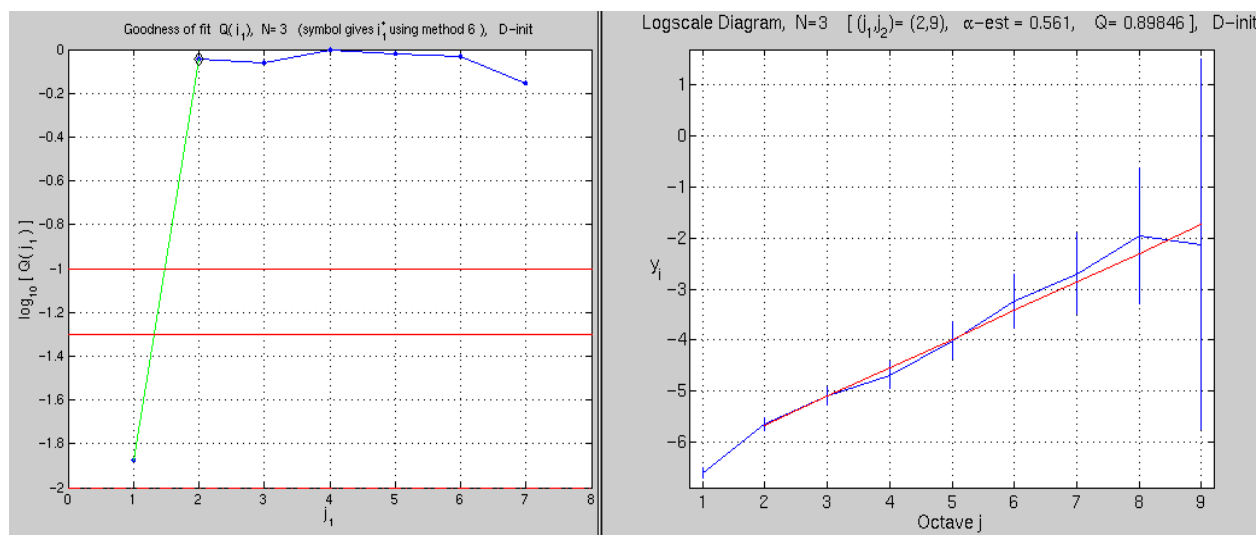
http://www.cubinlab.ee.unimelb.edu.au/~darryl/secondorder_code.html

El código está disponible gratuitamente para uso no comercial, específicamente para la investigación y la enseñanza, a condición de que las cabeceras de los derechos de autor y el encabezado de cada uno de los archivos no se quiten, y además una adecuada cita en los trabajos publicados que utilicen dichos archivos.

5.2.1.1 La script LDestimate.m

Según las notas que acompañan esta script, el autor parte de la definición de LRD dada en términos del espectro de potencia $f(v) \sim cf(v)^{-\alpha}$ cuando $v \rightarrow 0$, donde v , es la frecuencia, α es el exponente de escala, que es adimensional, y cf es un coeficiente con dimensión de varianza y describe aspectos cuantitativos de la longitud o extensión del comportamiento LRD, y como ejemplo de la importancia de cf expresa que los intervalos de confianza de la estimación de la media de LRD son proporcionales a \sqrt{cf} . La script entrega una estimación de sendos parámetros α y cf junto a otros que toman importancia según el contexto como se muestra a continuación.

Las siguientes figuras, son gráficos de salida utilizando la script LDestimate.m, partiendo con los datos de ejemplo llamado fgn8 que es un vector que representa un ruido gaussiano fraccional con $H=0.8$, y que forman parte del conjunto de scripts disponibles en la Web de Darryl Veitch.



Como se observa aparte del diagrama Log-escala (figura derecha), nos proporciona una grafica con la bondad de la estimación en cada punto como función de j_1 (figura izquierda), lo que permite una mejor elección de las octavas j_1, j_2 .

En la página citada mas arriba, tenemos los siguientes comentarios que hace Darryl Veytch acerca de esta script:

- Diferentes tipos de escala son posibles, sin embargo, el procedimiento de análisis es el mismo en cada caso. En primer lugar, el diagrama de Logscale se genera, y examina los datos para encontrar un punto de corte inferior de la escala j_1 , y otro punto de corte superior j_2 , en los que la alineación (línea recta) se observa. Estos puntos de corte deben ser experimentados para encontrar un rango que se ajuste a la regresión de los intervalos de confianza sobre el Diagrama Logscale (Los valores iniciales se deben especificar en la lista de argumentos de "LDestimate", pero estos se pueden cambiar interactivamente.).
- Para cada rango de alineación elegido, la función de los resultados de la estimación de la pendiente "alfa", toma valores reales. El valor de alfa, y el rango j_1, j_2 , ayudará a determinar qué tipo de escala se presenta, por ejemplo, un valor en (0,1) con alineamiento en escalas grandes sugiere LRD. (Por supuesto para determinar LRD, j_2 debe ser siempre el más grande disponible, dada la longitud de los datos, aunque esto puede no ser visto en datos reales.) Por otra parte la alineación en todas (o casi todas) las escalas de alfa > 1 sugieren autosimilitud. Cabe aclarar que aquí se refiere con el símbolo alfa (α), a lo que en los desarrollos de esta tesis está como gamma (γ).
- Por conveniencia, alfa se transforma en valores de los parámetros relacionados, como el parámetro de Hurst H , o la dimensión fractal de la muestra (válida sólo si es gaussiana). El usuario debe determinar qué tipo de escala está presente, y por ende, si el parámetro es adecuado. Por ejemplo la estimación no tiene ningún sentido a menos alfa se encuentre en el intervalo (1,3), y el parámetro de auto-similitud H no tiene sentido si alfa < 1 .
- NOTA: en el caso de LRD, alfa es el parámetro pertinente directamente, sin embargo, a veces es reescrita como una «H», pero esta no es la H de auto-similitud estricta, es simplemente una convención para reescribir alfa de esta forma para procesos LRD.
- La experimentación con el número momentos desvanecientes N de la wavelet es necesario para: (a) asegurarse de que la onda detalles están bien definidas (con valores de H altos, serán necesarios valores más altos de N , $N = 1$ es suficiente para estudiar LRD), y (b) eliminar o disminuir la influencia de las tendencias deterministas, como ser tendencias lineales o variaciones de nivel medio, que pueden estar presentes. En ambos casos es conveniente un aumento de N , hasta lograr un diagrama Logscale estable.
- Una advertencia: la ventaja de la utilización de onditas es que las estimaciones en distintas escalas son casi no correlacionadas entre sí. Por lo tanto, en el Diagrama Logscale que se verá considerablemente apartados en diferentes puntos sobre la línea de regresión (en particular para valores pequeños de alfa, debido al reajuste automático del grafico). Esta variación es natural y deseable y no debe interpretarse como una falta de alineación.
- Una estadística de bondad de ajuste $Q(j_1)$, basado en Chi-Cuadrado se emite para ayudar con la elección del rango de escala, y se representa en el título del gráfico correspondiente. es la probabilidad de observar que los datos, con las estimaciones de la varianza en cada escala realmente siga la forma de una recta para asegurar la efectividad de la regresión lineal. Un valor superior a 0,05 es aceptable y es aconsejable la elección de j_1 , a partir de donde se estabiliza el valor de $Q(j_1)$, en el caso del ejemplo mostrado arriba sería $j_1 = 2$. La estimación visual de la región de alineación es difícil, cuando los intervalos de confianza se vuelven muy pequeños, en escalas pequeñas, en este caso la estadística $Q(j_1)$, es una mejor guía.

- Si los datos no son gaussianos, los intervalos de confianza tanto para los puntos en el diagrama de Logscale y estimaciones para el final pueden ser un poco menor de lo que deberían ser. El valor Q también puede resultar ridículamente pequeño, como 0.0000001 o peor. En este caso no se deben tomar literalmente los intervalos de confianza, y Q sólo se puede utilizar en una forma relativa-para comparar las distintas opciones de escala. Esto también puede hacerse de manera efectiva mediante la script `newchoosej1`, que se describe a continuación, la cual es robusta a las desviaciones de gaussianidad.

5.2.1.1.1 Acerca de `newchoosej1.m`

- La función o script "`newchoosej1.m`" toma la información de `LDestimate` y calcula la bondad de ajuste Q de los valores y, a continuación, pasa a "`method6.m`" que es otra script, que aplica la heurística para devolver el valor óptimo $*j_1$.
- Un valor de j_2 se puede introducir, y para cada uno se devuelve j_1 .
- Es conveniente y natural para llamar a "`newchoosej1`" desde dentro de la script "`LDestimate`", por lo que este último tiene un parámetro de entrada para permitir su ejecución si se desea activarlo, pero de cualquier manera la $*j_1$, no se utiliza realmente en el calculado por el "`LDestimate`". El valor $*j_1$ para un valor seleccionado j_2 es convenientemente devuelto por "`LDestimate`".
- El valor j_2 se pueden cambiar mediante la edición de la llamada a la rutina "`newchoosej1`" en "`LDestimate`", la función es "enviada" con j_2 ajustado en la medida de lo posible, correspondiente a LRD.

5.2.1.2 Utilizando `LDestimate`

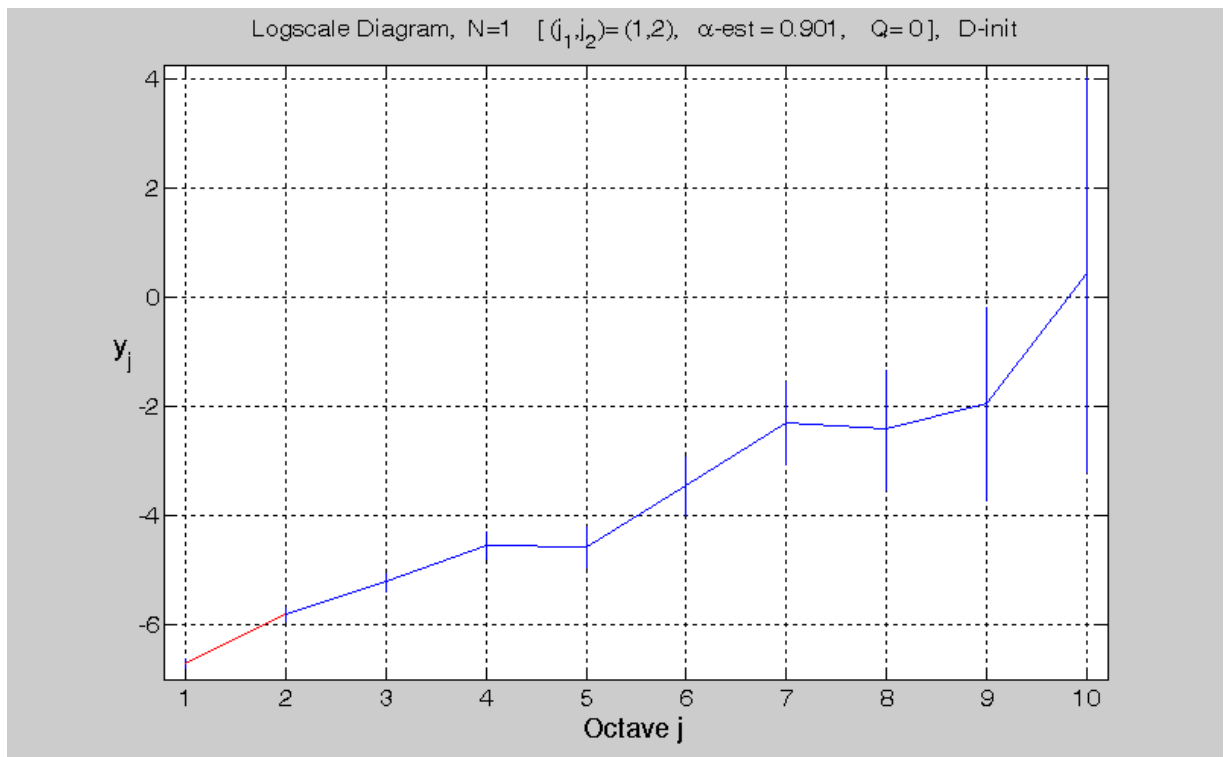
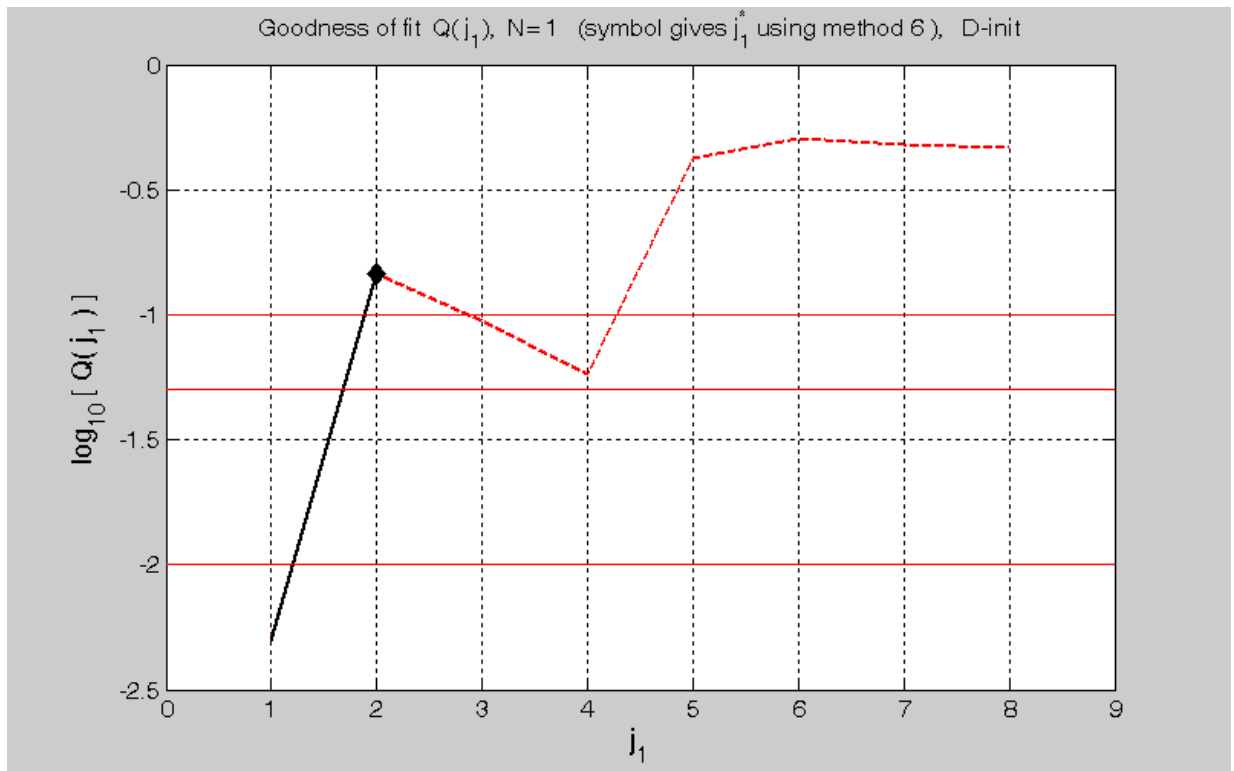
Antes de empezar a trabajar con esta script, sugiero bajar del link suministrado y descomprimir el archivo LDestimate_code.tar.gz en una carpeta y luego seleccionar esta carpeta como directorio de trabajo de MATLAB®. La script puede utilizarse como una función con 7 argumentos de entrada con la siguiente sintaxis:

`LDestimate(data,regu,j1,j2,discrete_init,calcj1,printout)`

1. `data` = Vector con los datos que se desean analizar (debe ser un vector fila).
2. `regu` = N o número de momentos desvanecientes de la wavelet, este parámetro esta relacionado con la regularidad de la estimación, a mayor valor mejoramos la bondad del ajuste Q, la variable `regu` esta disponible desde 1 a 10. (sugiero empezar con 1 e ir aumentando para observar la variación de Q y el grafico de regresión respectivamente).
3. `j1` = octava de corte inferior debe ser ≥ 1 .
4. `j2` = octava de corte superior, su valor máximo esta relacionada con la longitud de los datos, pero se puede cambiar interactivamente durante la ejecución de la script, sugiero arrancar con el valor 2 y luego ir probando otros valores.
5. `discrete_init` = con el valor 1 realiza la inicialización MRA para datos intrínsecamente discretos si el valor es cero, asume la de entrada de datos ya inicializado (es decir, ya está calculada la aproximación de secuencia o se utiliza un vector correspondiente a una aproximación).
6. `calcj1` = con el valor 1 realiza el calculo de j_1 optimo utilizando la script `newchooseJ1` mencionada mas arriba, si el valor es cero omite el cálculo. (sugiero dejarlo en 1).
7. `printout` = con el valor 1 realiza los dos gráficos correspondientes a "Q" y la regresión del diagrama Log-escala. Con el valor cero no realiza los gráficos y solamente entrega los valores calculados.

A modo de ejemplo, utilizando como vector de datos a "fgn8", se muestra la salida que produce `LDestimate` con los siguientes parámetros:

`alfa=LDestimate(fgn8,1,1,1,1,1,1)`



En la primera figura se observa que podemos tomar como j_1 a partir de la 5ª octava y hasta la 8ª (note la línea 7 y 8 en la pantalla de salida que se muestra a continuación).

En la segunda figura se observa el diagrama Log-escala con la regresión hecha entre la primera y segunda octava, pero atención, observe el valor que toma la bondad del ajuste (línea 12 en la pantalla de salida).

Lo siguiente, es lo que muestra la ventana de comandos del MATLAB® (las líneas están numeradas para la explicación, pero la numeración no es parte de la salida):

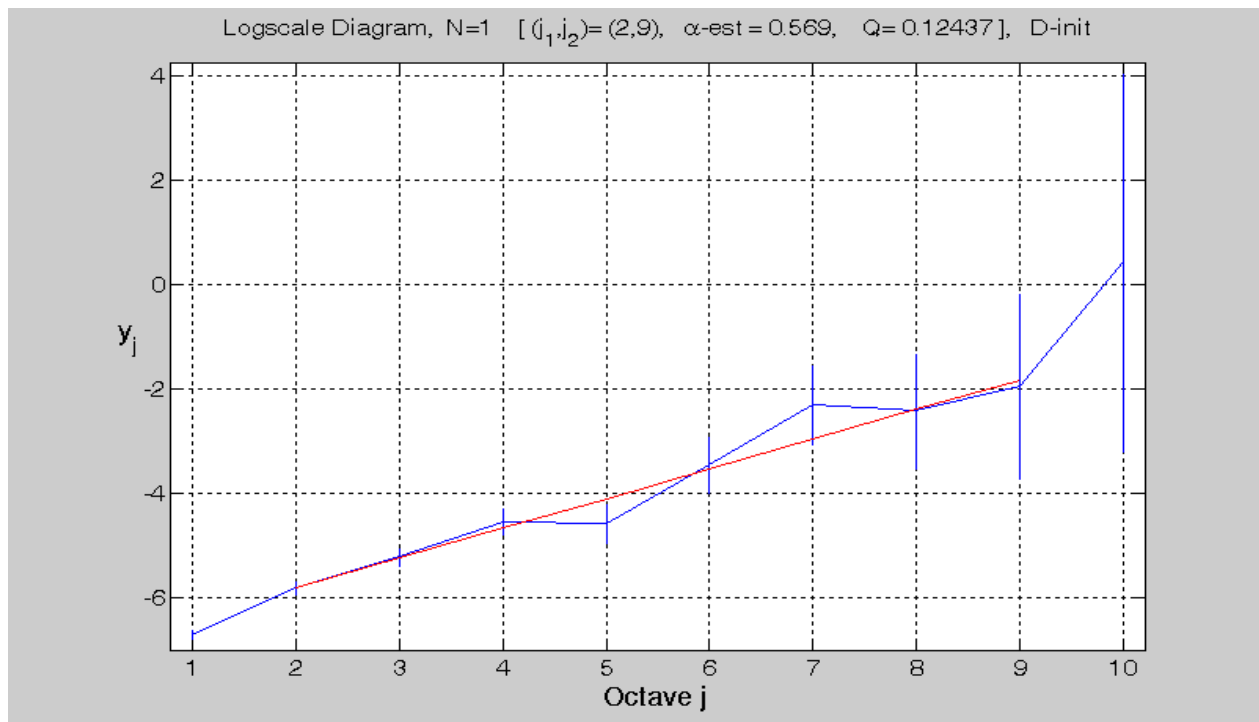
```

1. ** Using initialization for discrete series, filterlength = 405
2. No of points n_j at octave j: 1846 923 461 230 115 57 28 14 7 3
3. Number predicted by nj=n*2^j: 1846 923 461 230 115 57 28 14 7 3
4. *** Regrescomp: range choice (j1,j2)=(7,10) not safe for cfC and cf, regression will have to be redone!).
5. *** Regrescomp: range choice (j1,j2)=(8,10) not safe for cfC and cf, regression will have to be redone!).
6. *****In"newchoosej1"*****
7. j2 j1* j1= 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
8. 10: 2 0.00 0.15 0.09 0.06 0.42 0.51 0.48 0.47
9. *****
10. ***** Regrescomp: Cannot calculate Q, need at least 3 points.
11. Current plot held
12. Octaves: in_data available selected Goodness of fit (Prob of data assuming lin-regression)
           1--11.9 1--10 1--2 0.00000
13. Scaling parameters are: alpha (LRD) H (LRD rewrite) H=h (ss,Holder) D (frac dim, if alpha in (1,3))
    Estimates: 0.901 (0.950) -0.050 2.050
           CIs: [0.740, 1.062] [0.870, 1.031] [-0.130, 0.031] [1.969, 2.130]
14. Second parameters are: cf N/A sigma^2 N/A
    Estimates: 0.0039 Work in progress
           CIs: [0.00314, 0.00475] Work in progress
15. New initial octave j1? (hit return to exit loop) 2
16. New final octave j2? 9

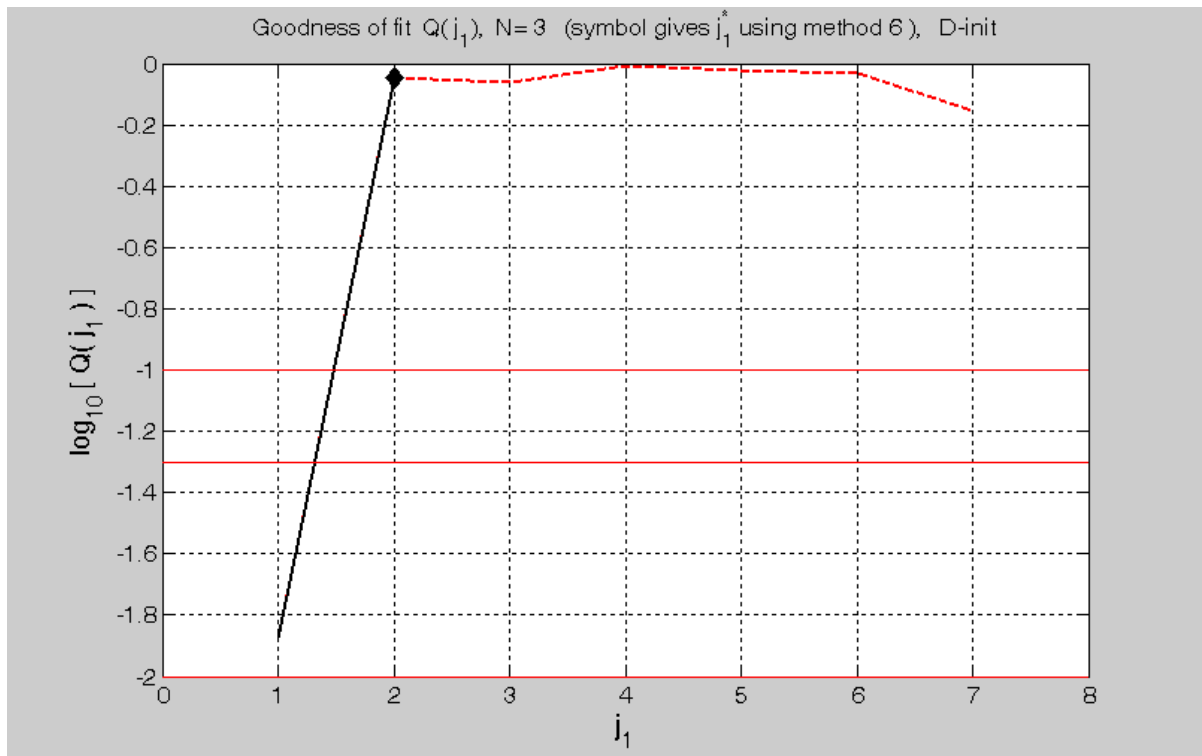
```

Como se puede observar la salida de la función proporciona información acerca de los valores utilizados y calculados por las rutinas como ser la cantidad de puntos por octava (línea 2), la bondad de la regresión que en este caso es cero (circulo rojo en la línea 12), lo cual nos advierte que no es segura la estimación (note las líneas 4 y 5). En la línea 13 el valor de α y H , con sus correspondientes intervalos de confianza “CI’s”. La línea 14 muestra la estimación del parámetro cf , cuya importancia se destaca en [5.2.1.1](#), junto a su respectivo “CI”

En la línea 16, donde dice: New final octave j2?. La script aguarda por la entrada de un nuevo valor para j_1 , y también nos pregunta por un nuevo valor de j_2 , que en este caso se asigna 2 y 9 respectivamente. Entonces recalcula los parámetros entregando la siguiente regresión:



Si se repite el análisis pero cambiando el N° de momentos desvanecientes (colocamos 3 en lugar de 1), la expresión queda de la misma manera que la utilizada en el ejemplo inicial $\alpha=LDestimate(fgn8,3,1,1,1,1,1)$ y se observa en el grafico Q de salida:



Claramente se advierte que mejoro la bondad del ajuste (en este caso es igual a 0.89846 como se muestra abajo), mejoro también el intervalo de octavas disponibles al cambiar el número de momentos desvanecientes N. Se obtienen entonces, los siguientes valores :

New initial octave j1? (hit return to exit loop) 2
 New final octave j2? 9

```
*****
Current plot held
Octaves:  in_data  available  selected  Goodness of fit (Prob of data assuming lin-regression)
          1--12.0  1--9      2--9      0.89846
Scaling parameters are:  alpha (LRD)  H (LRD rewrite)  H=h (ss,Holder)  D (frac dim, if alpha in (1,3))
Estimates:             0.561          (0.780)          -0.220           2.220
CIs: [0.487, 0.635]  [0.743, 0.817]  [-0.257, -0.183]  [2.183, 2.257]
```

De esta manera podemos cambiar los distintos argumentos de la función y ver como se reflejan estos cambios en la salida de la misma.

Para finalizar con la descripción de esta script se muestra el resultado del análisis para la traza de la captura “pAug89” que fue una de las empleadas en el trabajo original [1] de W. E. Leland, M. S. Taqqu, W. Willinger, y D. V. Wilson.

Utilizando la longitud en bytes de los paquetes como datos y fijando N igual a 4 se obtiene que se puede realizar la regresión entre la octava 10 y la 15 que arrojan los siguientes valores:

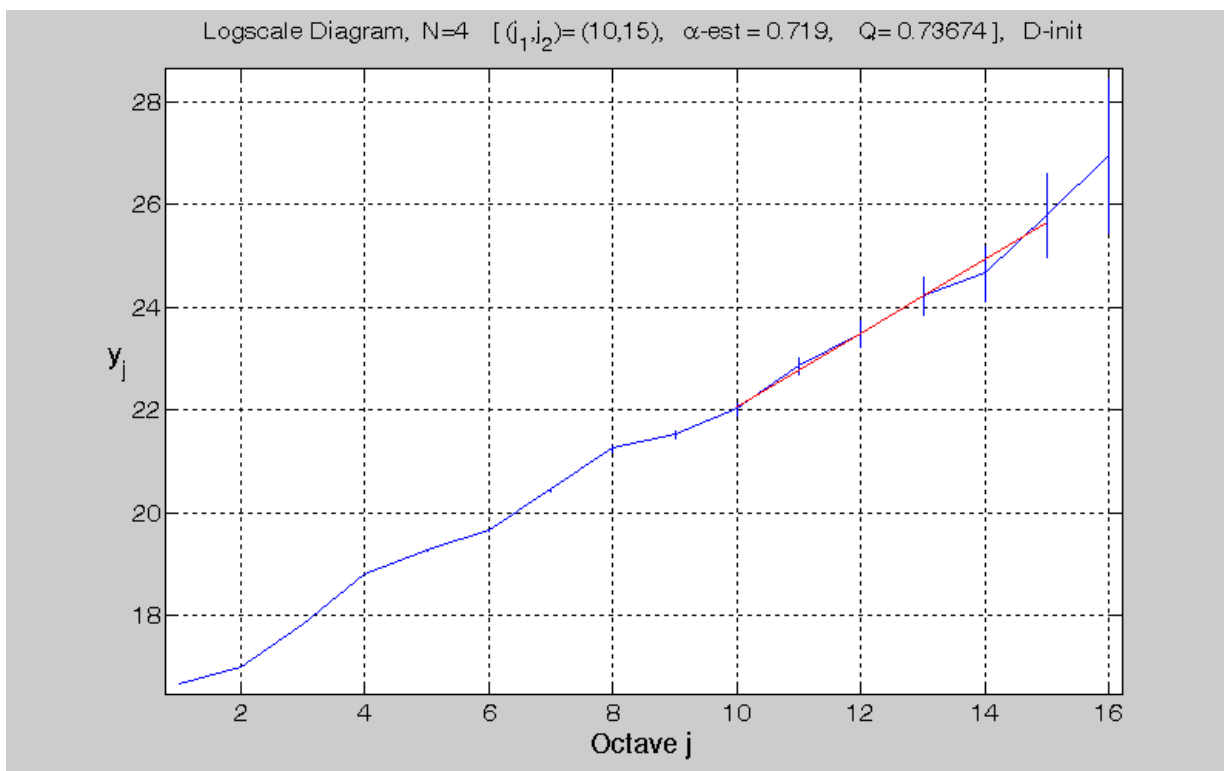
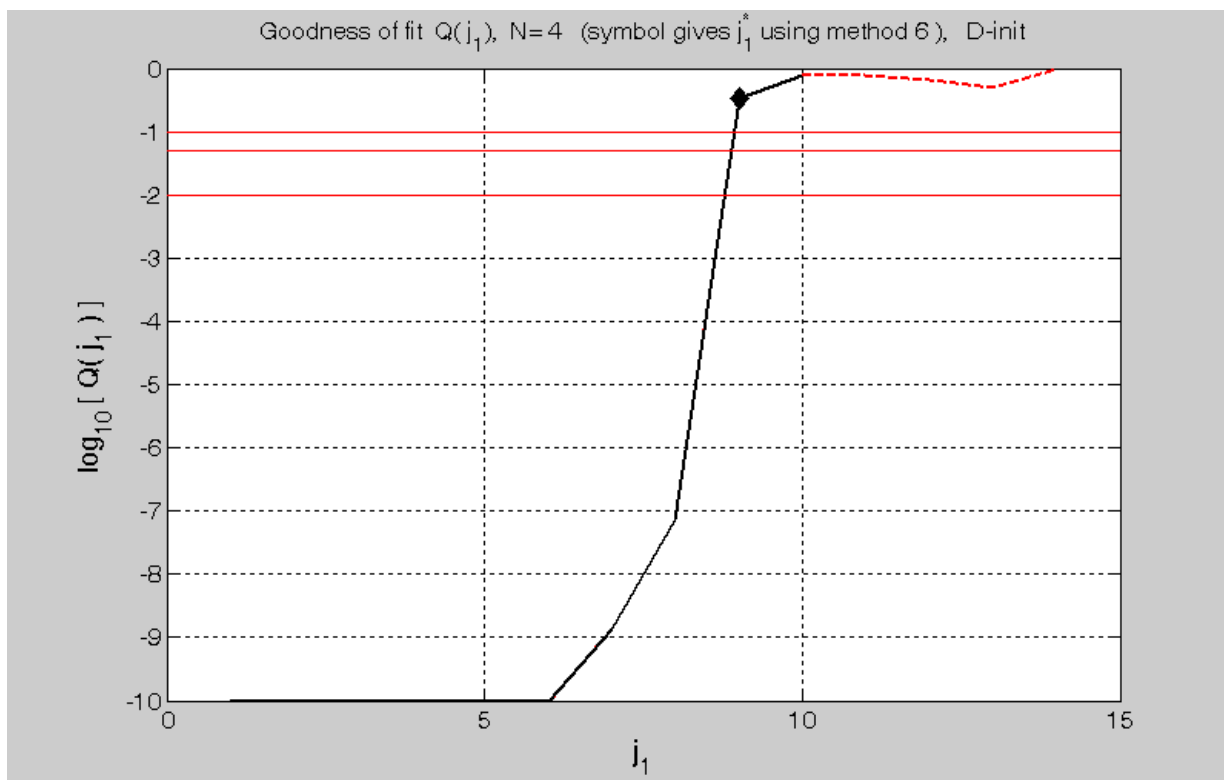
New initial octave j1? (hit return to exit loop) 10
 New final octave j2? 15

```
*****
Current plot held
Octaves:  in_data  available  selected  Goodness of fit (Prob of data assuming lin-regression)
          1--19.9  1--16     10--15     0.73674
Scaling parameters are:  alpha (LRD)  H (LRD rewrite)  H=h (ss,Holder)  D (frac dim, if alpha in (1,3))
Estimates:             0.719          (0.860)          -0.140           2.140
CIs: [0.639, 0.800]  [0.819, 0.900]  [-0.181, -0.100]  [2.100, 2.181]

Second parameters are:      cf      N/A      sigma^2      N/A
```

Estimates: 21832.9269
 CI's: [11023.02974, 38979.33665]

Work in progress



Con estos valores se aprecia que la región de alineación segura empieza a partir de la octava 10 y se obtiene un valor de $\alpha = 0.719$, lo que a su vez arroja un valor de $H = 0.860$.

Repitiendo el análisis pero ahora con el vector de tiempos entre arribos de la misma traza obtenemos que se puede hacer la regresión entre las octavas 10 y 15 arrojando los siguientes resultados:

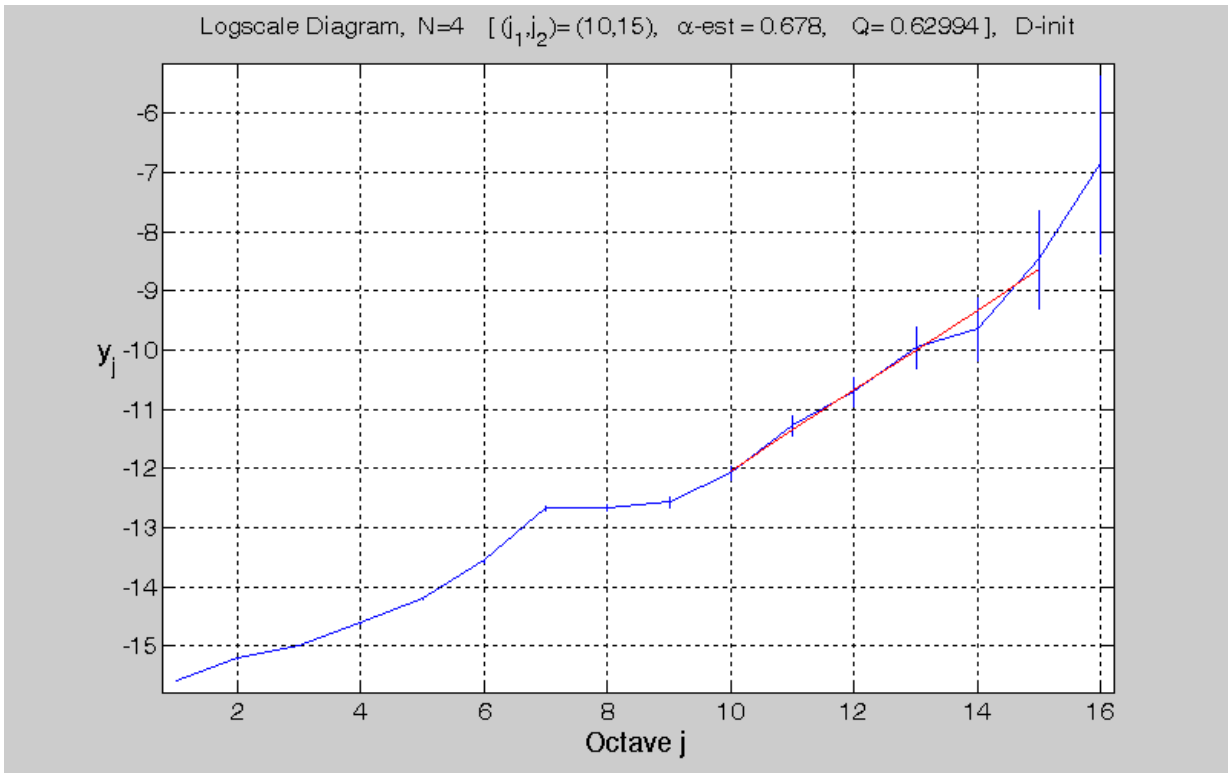
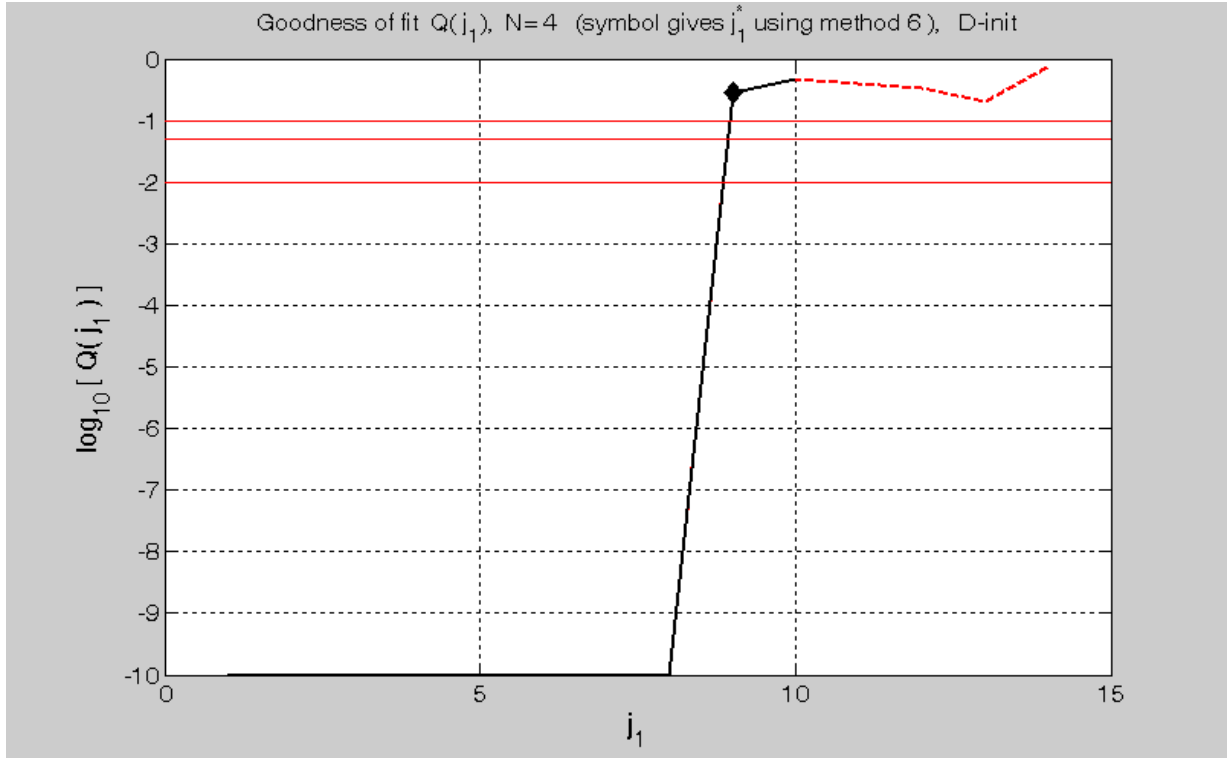
New initial octave j1? (hit return to exit loop) 10
 New final octave j2? 15

Current plot held

Octaves: in_data available selected Goodness of fit (Prob of data assuming lin-regression)
 1--19.9 1--16 10--15 0.62994

Scaling parameters are: alpha (LRD) H (LRD rewrite) H=h (ss,Holder) D (frac dim, if alpha in (1,3))
 Estimates: 0.678 (0.839) -0.161 2.161
 CI's: [0.598, 0.758] [0.799, 0.879] [-0.201, -0.121] [2.121, 2.201]

Second parameters are: cf N/A sigma^2 N/A
 Estimates: 0.0000 Work in progress
 CI's: [0.00000, 0.00000] Work in progress



De igual manera a como están descritas y comentadas las scripts del principio de este capítulo, la totalidad del código fuente de las scripts de Darryl Veytch, se encuentra comentado (en idioma inglés), y además cuentan con referencias de los trabajos donde fueron utilizadas.

5.2.2 Método de Varianza versus Octavas

Como se mostró en el capítulo 4, el conjunto de los detalles $\{d_X(j, k), j = 1, 2, \dots, J, k \in \mathbb{Z}\}$, es un proceso estacionario para cada octava j , si el número "N", de momentos desvanecientes de la wavelet madre ψ_0 , es $N \geq \frac{\gamma-1}{2}$. La varianza de $d_X(j, k)$ reproduce el comportamiento de escala subyacente, dentro de un rango de octavas $j_1 \leq j \leq j_2$. Dado que el valor medio de la wavelet es cero por la condición de admisibilidad, el segundo momento de $d_X(j, k)$ es proporcional a $2^{j\gamma}$, donde j_1, j_2 y γ , dependen del tipo de fenómeno de escala que exhiba el proceso original $X(t)$. Esto se puede expresar mediante la relación:

$$E[d_X(j, k)^2] \propto 2^{j\gamma}$$

Como el valor medio es igual a cero, equivale a decir que la varianza de $d_X(j, k)$, es proporcional a $2^{j\gamma}$, y por ende si aplicamos logaritmo en base dos, tenemos:

$$\log_2 \text{Var}[d_X(j, k)] \propto j\gamma$$

Como se observa estamos en presencia de una recta de pendiente γ , en una grafica log – log, donde en las ordenadas tenemos la varianza del conjunto de detalles $[d_X(j, k)]$, y en las abscisas la correspondiente octava j .

Debido al submuestreo ($\downarrow 2$), del filtrado de la transformada discreta, cada conjunto de detalles para una escala determinada, tiene la mitad de coeficientes que la escala anterior. Para fijar un valor de corte para las octavas o escalas a desarrollar, vemos que el mismo se puede calcular de la siguiente manera:

Si el valor m_0 es la longitud del vector datos y el valor m_n es la cantidad de coeficientes que queremos en la escala más pequeña o sea la longitud del vector $[d_{X_n}(j_n, k)]$, tenemos:

$$\frac{m_0}{m_n} = 2^n \Rightarrow \log_2 \left(\frac{m_0}{m_n} \right) = n$$

Siendo n el número de escalas o veces que se aplica la transformada wavelet.

Una vez conocida la cantidad de niveles que se pueden desarrollar en función de la longitud de los datos, se procede a aplicar en forma iterativa la transformada wavelet discreta al vector de datos, de manera que en cada iteración se calcula la varianza correspondiente a ese nivel o escala y se lo almacena en un vector. Terminado el cálculo de todos los niveles, se procede a graficar y realizar una regresión lineal para estimar la pendiente de la recta resultante.

Con la estimación de la pendiente " γ " de la recta, se puede conocer el valor de H , aplicando las propiedades de las wavelet vistas en el punto [4.7](#) del capítulo 4.

- Si $\gamma < 1$ se asume que el proceso subyacente es LRD $H = \frac{\gamma+1}{2}$
- Si $\gamma > 1$ se asume el proceso como fractal y se calcula $H = \frac{\gamma-1}{2}$

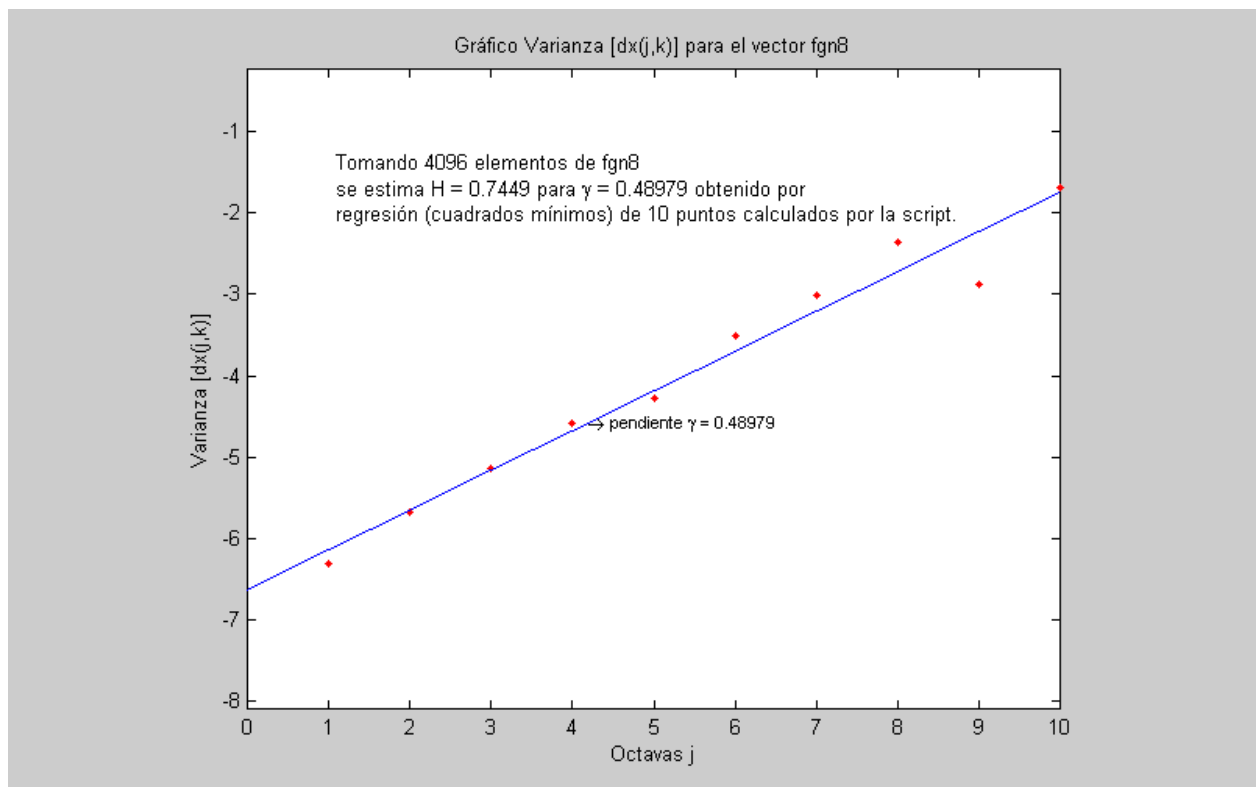
Cuando se analizan trazas de tráfico, siempre es bueno conocer el contexto en el que se toman las muestras pues este conocimiento ayuda a interpretar mejor los valores obtenidos y la naturaleza del proceso que representan, por ejemplo en el caso de suponer el proceso estacionario o débilmente estacionario como correspondería a muestras de tiempos entre arribos de paquetes, y obtenemos un exponente de escala menor a uno estaríamos validando la suposición de estacionariedad, por el contrario si el exponente es mayor a uno, habría que sospechar de falta de estacionariedad o estar en presencia de un comportamiento fractal, y aquí es donde el conocimiento del contexto del proceso (tipo de traza, circunstancias de la captura, etc.), nos ayudan a completar el panorama y dar un significado mas preciso a los valores estimados.

Para hacer esta estimación se desarrolló una script llamada “varianzaW.m”, que produce un gráfico $\log_2 Var[d_x(j,k)]$ versus j , y realiza la regresión para estimar el valor de H .

5.2.2.1 Utilizando varianzaW.m

Cabe señalar que este método es menos sofisticado que “LDestimate” para la estimación del valor de H , pues no se hace ningún tratamiento previo de los datos para “inicialar” o preparar la serie, y tampoco se hace ninguna selección de las octavas para conseguir la región de alineación más segura. De todas maneras cuenta con la ventaja estadística que los coeficientes de las correspondientes octavas son independientes, y por ende están desacoplados de la fuerte correlación existente en los promedios del dominio temporal. Puede utilizarse como una prueba de presencia de LRD para una secuencia dada, con la ventaja que requiere nada más que la introducción del vector datos, o sea es una función de un solo argumento como muestra la siguiente salida de pantalla:

```
>> load fgn8.dat
Nombre del vector que contiene los datos? fgn8
>>
```



La figura muestra la salida de la script “varianzaW.m”, para el vector fgn8.dat que se utilizo como ejemplo de LDestimate.m; como se aprecia, el valor de la estimación de $H = 0.7449$, resulta un poco menor al valor hallado por LDestimate.m ($H = 0.780$), el cual es mucho mas riguroso y por ende, tiene mayor exactitud.

5.3 Síntesis de procesos autosimilares

Hasta aquí se mostraron métodos de análisis de procesos LRD/autosimilares, a continuación se muestran métodos para sintetizar procesos autosimilares mediante la generación de secuencias de números que presenten tales características.

5.3.1 Utilización de Wavelets para la Generación de un proceso FGN con Parámetro H Predeterminado

Como se mostró en el capítulo 3. El proceso de movimiento browniano fraccional, es una generalización del proceso de movimiento browniano, donde la característica sobresaliente es la dependencia de sus incrementos en el intervalo abierto $\frac{1}{2} < H < 1$. El Ruido Gaussiano Fraccional es H_{sssi} (autosimilar con incrementos estacionarios), tiene un particular interés para el modelado del tráfico pues nos permite manejar la dependencia mediante un solo parámetro H cumpliendo a con la premisa de parsimonia. También tiene la ventaja de ser gaussiano, lo que permite su manipulación analítica, y representa en forma natural a fenómenos que se observan en el tráfico.

Existe una íntima relación entre los procesos autosimilares y los procesos con dependencia de rango largo, pues se demuestra que los incrementos de un proceso autosimilar con incrementos estacionarios (H_{sssi}), presentan dependencia de rango largo en el intervalo abierto $0.5 < H < 1$, además la dependencia de rango largo se manifiesta cuando la densidad espectral del proceso sigue una ley de potencias para frecuencias cercanas a cero, o en cercanías del origen, esto es:

$$S(\omega) = c|\omega|^{-\gamma}; \text{ cuando } \omega \rightarrow 0, \text{ y aquí se cumple la relación } \gamma = 2H - 1.$$

La densidad espectral de un proceso FGN para un determinado H , esta representada por la expresión:

$$f(\lambda, H) = cf |\lambda|^{1-2H} + O(|\lambda|^{\min(3-2H, 2)})$$

$$cf = \sigma^2 (2\pi)^{-1} \sin(\pi H) \Gamma(2H + 1)$$

Donde, σ^2 es la varianza de λ , $\Gamma(\cdot)$ es la función gamma, λ son las frecuencias, H el parámetro de Hurst, y $O(\cdot)$, representa el error residual

El método que se muestra a continuación fue desarrollado por Hae-Duck J. Jeongy, DonMcNicklez and Krzysztof Pawlikowski²⁹, y aprovecha la circunstancia de que la estructura de las wavelets coincide en forma natural con la estructura de dependencia de rango largo de los procesos autosimilares, cuyo caso típico son precisamente los procesos FGN.

El algoritmo empleado para la generación de series FGN sintéticas, parte de la formula de la densidad espectral para los procesos FGN. Y se lo describe a continuación.

Paso 1: Empieza un lazo con $i = 1$ hasta n donde se calcula un vector λ con la secuencia de frecuencias $\{f_1, f_2, f_3, \dots, f_n\}$, donde $\frac{\pi}{n} \leq f_i \leq \pi$. Fijado el valor de H , se procede al cálculo del espectro con la expresión, obteniendo la secuencia del espectro $\{f_i\}$.

²⁹ [16] Jeong, H.-D.J., McNickle, D., and Pawlikowski, K. (1999), Fast Self-Similar Teletraffic Generation Based on FGN and Wavelets, In Proceedings of IEEE International Conference on Networks (ICON'99), Brisbane, Australia

Paso 2: Se multiplica el espectro $\{f_i\}$, por la realización de una variable aleatoria independiente con distribución exponencial y media uno, y obtenemos $\{\widehat{f}_i\}$, esto se hace para ajustar el espectro, porque la densidad espectral estimada para una determinada frecuencia, se distribuye asintóticamente como una variable aleatoria exponencial de media $f(\lambda, H)$.

Paso 3: Se genera una secuencia de números complejos $\{Y_1, Y_2, Y_3, \dots, Y_n\}$, de manera que el módulo de $|Y_i| = \sqrt{\widehat{f}_i}$, y la fase de Y_i esté uniformemente distribuida entre 0 y 2π . Esta técnica de la fase aleatoria y uniformemente distribuida, preserva la densidad espectral y además asegura que la distribución marginal de la secuencia final sea normal (gaussiana), que son los requisitos del FGN.

Paso 4: Se calculan los coeficientes de síntesis de la Wavelets de Daubechies para ser utilizados en la transformación discreta inversa $IDWT\{Y_1, Y_2, Y_3, \dots, Y_n\}$, que como resultado entrega la secuencia $\{X_1, X_2, X_3, \dots, X_n\}$, representado una aproximación autosimilar del FGN en el dominio del tiempo.

En el Apéndice 1 se encuentra la script con el algoritmo descrito, desarrollada por [17], para MATLAB®, denominada FGNW.m que esta acompañada por otras dos scripts llamadas FGNDWSpectrum.m, y Daub.m.

Para poder utilizar la script FGNW.m, los autores de la misma se apoyaron en un conjunto de scripts desarrolladas por la Universidad de Vigo bajo las condiciones GNU. La copia completa del conjunto se puede obtener de http://www.tsc.uvigo.es/~wavelets/uvi_wave.html que contiene una completa colección de scripts para trabajar con wavelets.

Sobre este particular hago la aclaración que conviene bajar el conjunto Uvi_Wave300.zip desde el sitio http://www.control.auc.dk/~alc/Uvi_Wave300.zip, se lo copia a un directorio y se fija el mismo como directorio de trabajo del MATLAB®. Es importante señalar que, antes de utilizar las scripts, se debe ejecutar la script llamada starup.m, cuyo objetivo es iniciar todas las variables a ser utilizadas, señalo esto como importante, pues si se omite este paso, MATLAB® emitirá mensajes de error al ejecutar FGNW.m (que previamente también debemos ubicar en dicho directorio junto con sus scripts asociadas FGNDWSpectrum.m, y Daub.m).

5.3.1.1 Utilizando FGNW.m

Antes de empezar a utilizar FGNW.M, se estima conveniente hacer una aclaración sobre las scripts desarrolladas y publicadas en [17] y [28], dado que luego de analizarlas y probarlas debidamente, se introdujeron algunas modificaciones para poder utilizarlas en este trabajo y evitar posibles mensajes de error de MATLAB®. Las modificaciones son las siguientes:

La script llamada Daub.m es exactamente igual a la script daub.m (uno es con mayúsculas y el otro con minúsculas), del paquete Uvi_Wave300, difieren nada mas que en el nombre del argumento de la función; en el caso de Daub.m es VanishingMoment y daub.m lo llama num_coefs. Entonces sugiero utilizar la script daub.m del paquete Uvi_Wave300 introduciendo la debida corrección en la línea 3 del código de FGNW donde dice:

```
function SS = FGNDW(n,H,Scale, VanishingMoment)
```

lo reemplazamos por :

```
function SS = FGNDW(n,H,Scale, num_coefs).
```

Luego en la línea 20 donde dice:

$$[h,g,rh,rg] = \text{Daub}(\text{VanishingMoment});$$

lo reemplazamos por:

$$[h,g,rh,rg] = \text{daub}(\text{num_coefs}).$$

Con esto se evita el mensaje de error aludiendo al nombre de la función Daub como duplicada y nombre distinto, además las variables VanishingMoment y num_coefs son exactamente las mismas en cuanto a su tratamiento dentro de las scripts.

Otra aclaración importante: En la línea 18 del código fuente de FGNW, se hace referencia al vector complejo $z = \text{real_part} + \text{imag_part}*i$; para evitar un mensaje de error se debe agregar el comando $z=(z')$; esto se hace para transponer el vector de lo contrario se produce un error en un llamado interno al juego de scripts, que utiliza como entrada este vector, pero lo utiliza como vector columna en lugar de vector fila, que es el formato que tiene z en la script original descrita en [28].

Hechas estas aclaraciones sugiero probar las scripts comentadas y experimentar cambiando las variables y ver los resultados para hacer las comparaciones pertinentes como ser la verificación del valor de H haciendo uso de los estimadores disponibles o también cambiar el N° de momentos desvanecientes (último argumento de FGNW) y ver como incide en la estimación, o probar con distintas cantidades de niveles que figuran en el argumento Scale.

Ejemplo de sintaxis para generar 2^{20} números, con parámetro $H = 0.8$, 20 niveles, y utilizando wavelet Daubechies con 10 momentos desvanecientes:

$$a = \text{FGNDW}(2^{20},0.8,20,10)$$

5.3.2 Utilización de la Transformada Inversa de Fourier para la Generación de un proceso FGN con Parámetro H Predeterminado

Como se mostró mas arriba, aprovechando la primera parte del código de FGNW.m, es posible la generación de la serie compleja “ z ”. Partiendo de este punto, agregando el cálculo de la transformada inversa de Fourier a esta secuencia, y seleccionando la parte real de la misma estaríamos invirtiendo espectro de frecuencias y tendríamos la secuencia de números en el dominio temporal, que sería la representación temporal del proceso FGN, si bien esta estimación es mucho menos precisa que la anterior tiene la ventaja de ocupar mucho menos computo, y tampoco es necesaria la utilización del paquete Uvi_Wave300.

En esencia, es muy similar al trabajo que realizó V. Paxson, con el título [Fast, Approximate Synthesis of Fractional Gaussian Noise for Generating Self-Similar Network Traffic](#). ACM SIGCOMM, Computer Communication Review, 27(5):5–18, 1997; pero con un estimador de espectro más sencillo.

5.3.2.1 Script FGNIFourier.m

Para implementar esto, se desarrolló la script FGNIFourier.m. Para utilizar esta script, la colocamos como es habitual en el directorio de trabajo y se puede aplicar como función de dos argumentos a cualquier variable, como ejemplo a continuación se muestra la sintaxis para un vector o variable llamada “ a ”, conteniendo 2^{20} números, los cuales sintetizan un FGN con un parámetro $H = 0.8$.

$$a = \text{FGNIFourier.m}(2^{20},0.8)$$

6 Discusión de los Resultados

6.1 El análisis en el dominio de la escala

Como se mostró en el capítulo 4, hay variadas herramientas de análisis mediante transformaciones, cada una con sus ventajas y limitaciones, pero en particular la estructura del análisis multiresolución, se adapta naturalmente al estudio de los fenómenos de escala y por ende al de los procesos autosimilares. Para corroborar esta afirmación, recordar que:

- El análisis multiresolución es una técnica que permite analizar señales en múltiples bandas de frecuencias obtenidas por un factor de escala " a ", y siendo la frecuencia la inversa del período, esto equivale a múltiples escalas temporales. El estudio entonces, se hace directamente en el dominio de la escala.
- Las bases wavelets utilizadas en el análisis multiresolución, se obtienen a partir de la dilatación de la wavelet madre, esta dilatación es en esencia una operación de cambio de escala, pues como se sabe, la escala a , esta relacionada con la frecuencia central f_c , de la wavelet madre, de manera que la base de funciones de análisis de la transformada wavelet, tiene una intrínseca invarianza a la escala.
- Eligiendo adecuadamente la wavelet madre con un $N^\circ > 1$ de momentos desvanecientes, permite definir correctamente los detalles wavelets y eliminar o disminuir la influencia de las tendencias determinísticas como pueden ser las tendencias lineales o las variaciones respecto de la media.

En primer lugar el fenómeno de escala no se presupone, sino que se detecta al observar regiones de alineación en una gráfica, dado que es imposible conocer a priori en cual región de escalas se produce el fenómeno de la invarianza de las propiedades estudiadas, entonces para la detección de un fenómeno invariante a la escala, es útil el conocimiento del tipo de proceso que estamos estudiando, ese conocimiento nos ayuda a contextualizar los valores obtenidos del análisis, si se pone atención a lo expuesto en el punto [4.7](#) del capítulo 4, vemos que el proceso o secuencia estudiada, en términos generales puede presentar un comportamiento puramente autosimilar o fractal, puede presentar LRD dentro de un rango determinado, o bien puede ser resultado de los llamados procesos o ruidos $1/f$, en los cuales es posible establecer un rango definido donde la LRD y la autosimilitud son equivalentes. Entonces es importante que el parámetro de escala " γ " sea interpretado en el contexto adecuado. Es importante aclarar que la notación utilizada para el exponente de escala en este trabajo es " γ ", y la notación utilizada por Darryl Veitch en la script LDestimate.m es " α ", entonces para evitar confusión, en adelante seguiremos con $\alpha = \gamma$.

6.2 Discusión de la estimación mediante la script LDestimate

- a) Si $0 < \alpha < 1$, y se observa que la región de alineación va desde algún j hasta la máxima octava disponible en la descomposición wavelet, estamos ante las condiciones de un proceso LRD con el exponente α indicado por el estudio. También es posible que el proceso sea estacionario o débilmente estacionario, en cuyo caso podemos suponer ante estos valores que la secuencia represente el incremento de un proceso *sssi* con el H , indicado en la salida (la script utiliza por defecto $H = \frac{\alpha + 1}{2}$); por ejemplo si las trazas provienen de los tiempos entre arribos de paquetes, esto nos brinda información extra, y perfectamente podemos inferir que el proceso es LRD y presenta un comportamiento autosimilar dentro de los límites establecidos en el estudio. El ejemplo de aplicación mostrado en [5.2.1.2](#), correspondiente a la longitud en bytes de los paquetes de la traza pAug89, arroja un valor de $\alpha = 0.719$, por ende aquí el valor de $H = 0.860$

- b) Si $\alpha > 1$, y la región de alineación abarca todo el gráfico, en particular la región de las mayores escalas, se puede inferir que $-\infty < j < \infty$, y aquí el proceso no es estacionario y se puede asumir la fractalidad del mismo siendo el valor del parámetro $H = \frac{\alpha - 1}{2}$; este valor lo podemos visualizar en la salida de la script como el parámetro bajo el título H=h (ss,Holder), que está situado a la derecha de H (LRD rewrite), esto contrasta con el análisis realizado arriba donde $\alpha < 1$; (se debe tener en cuenta la aclaración hecha sobre el cálculo de H , pues como la script utiliza la expresión por defecto, bajo estas circunstancias indicara erróneamente un valor de $H > 1$). Un claro ejemplo de esto sería un proceso de conteo de la cantidad de bytes llegados hasta el instante t , al puerto de un enrutador (proceso del que estamos seguros es no estacionario) y al analizarlo obtenemos $\alpha = 2,59$, entonces aquí el valor correspondiente al parámetro $H = \frac{2,59 - 1}{2} = 0,795$.
- Normalmente valores de $\alpha > 1$, sugieren fractalidad, pero se deben evaluar los demás parámetros para poner todo en el debido contexto.
- c) También es posible que el proceso estudiado muestre alineación en una región limitada entre dos octavas j_1, j_2 ; aquí es muy posible estar en presencia de un proceso débilmente estacionario que responde a una ley de potencia del tipo $1/f$ y dependiendo de las circunstancias el parámetro H , podrá ser determinado en función del α obtenido y del conocimiento previo que tengamos de la traza estudiada, si estamos seguros que proviene de un proceso estacionario y $\alpha < 1$, directamente asumimos que el proceso subyacente es *sssi*, y el valor de H , es el indicado por la script, pero también puede darse la circunstancia de que exista un valor de corte j_2 , que esta mas allá de la máxima octava en la que se descompuso el proceso bajo estudio y si bajo esta circunstancia nos encontramos con $\alpha > 1$, podemos sospechar que el proceso no es estacionario o bien habría que revisar la posibilidad de descomponer al máximo número de octavas posibles.
- d) La salida de LDestimate, vemos que también entrega un valor llamado D , que es la dimensión fractal de la muestra asumiendo que ésta sea gaussiana, y solo tiene sentido si los valores que toma α , están comprendidos en el intervalo (1,3), y entonces se toma como válida la relación: $\alpha = 2h + 1 = 5 - 2D$. Lógicamente esto no tiene ningún sentido si $\alpha < 1$, pues de hecho se asume el proceso subyacente como fractal. El valor que tome este parámetro, bajo las condiciones descritas, ayuda a interpretar la naturaleza del proceso cuando se estudian datos empíricos.
- e) Otro parámetro de importancia que nos entrega la salida de la script, es cf , y según comenta Darryl Veitch, cuando toma valores reales positivos, implica que la suma de todas las correlaciones tiende a infinito, (de hecho es infinito), pero sus tamaños individuales pueden ser arbitrariamente pequeños para grandes “lag’s” u holguras temporales, dependen del valor que tomen los lag’s (recordar que la correlación en un proceso débilmente estacionario no depende del tiempo si no de la diferencia de tiempos $k = t - s$, como lo muestra la ecuación [4.5] del punto 4.6 en el capítulo 4). Esto también ayuda a poner en el debido contexto los valores hallados por la estimación.
- f) Existe la posibilidad que se verifique más de una región de alineación, por lo que el proceso estudiado puede ser bi-escala. Un proceso puede tener naturaleza puramente fractal a pequeñas escalas con un determinado exponente; y además debido a un comportamiento LRD, puede tener una región de alineación en la zona de escalas altas, donde tendrá otro exponente de escala.
- g) Por último existen los llamados procesos multifractales³⁰; con relación a esto hago una aclaración muy importante; en este trabajo se estudian procesos con exponente

³⁰ A quienes deseen conocer mas sobre esto, se recomienda la lectura del trabajo Multifractal Processes del autor Rudolf H. Riedi, disponible en: <http://www.spin.rice.edu/PDF/MP.pdf>

“ H ” constante, también conocidos como procesos monofractales; pero existen procesos donde “ H ”, no es constante y varía en función del tiempo, y si esta variación es a su vez fractal, corresponde a procesos que se conocen como multifractales. El concepto de multifractalidad es una generalización, que no es estudiada en este trabajo por razones de extensión; además es objeto de múltiples investigaciones actualmente, pero lógicamente el punto de partida para esos estudios se basa en los procesos monofractales.

6.2.1 Ejemplos de análisis en distintos contextos

Para la traza pAug89, estudiada por Leland y otros en [1]

Si tomamos como vector o proceso de estudio los tiempos entre arribos de pAug89:

Aplicando LDestimate, primero vemos que nos sugiere tomar los valores $j_1 = 9$, y $j_2 = 15$:

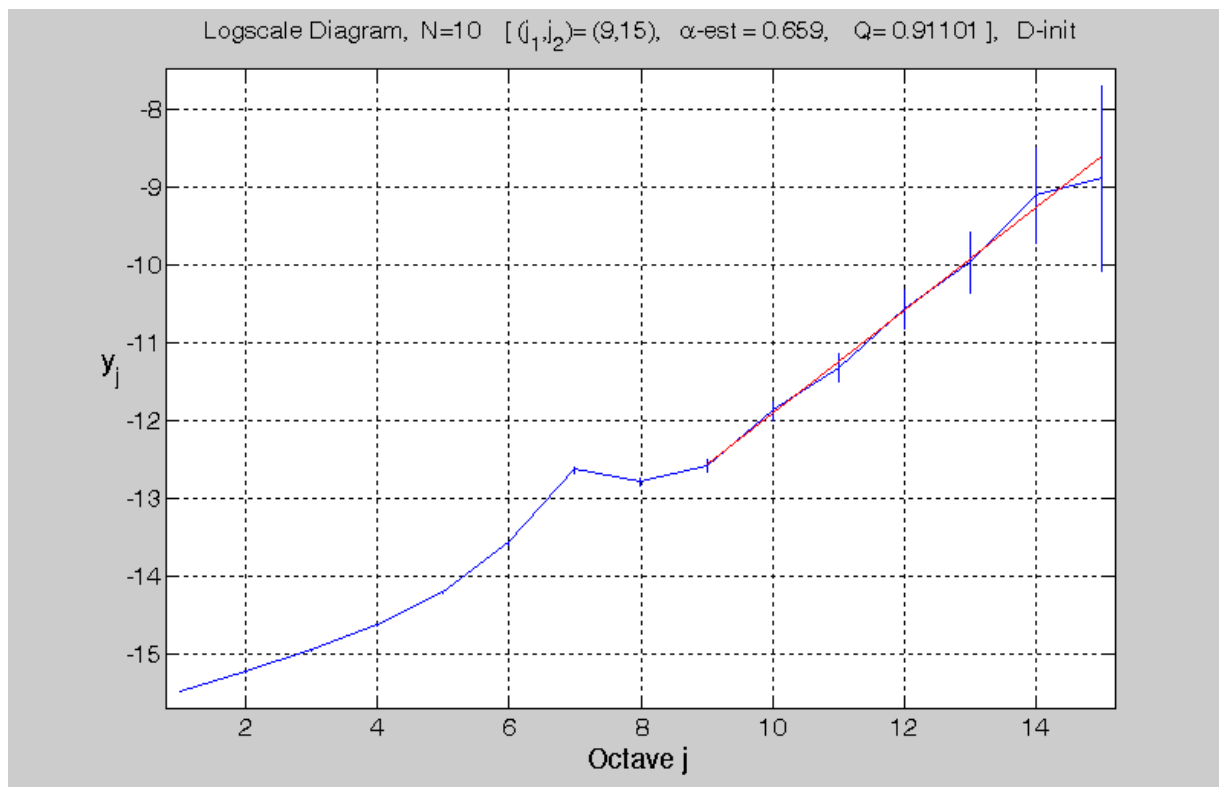
```
***** In "newchoosej1"*****
j2  j1*  j1=1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
15: 9      0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.91 0.86 0.89 0.78 0.49
*****
```

Con lo que obtenemos los siguientes parámetros:

```
*****
Current plot held
Octaves:  in_data  available  selected  Goodness of fit (Prob of data assuming lin-regression)
          1--19.9  1--15      9--15      0.91101

Scaling parameters are:  alpha (LRD)  H (LRD rewrite)  H=h (ss,Holder)  D (frac dim, if alpha in (1,3))
Estimates:              0.659          (0.829)         -0.171          2.171
CI's:                   [0.603, 0.715]  [0.801, 0.857]  [-0.199, -0.143]  [2.143, 2.199]
```

Gráfico Log-escape:

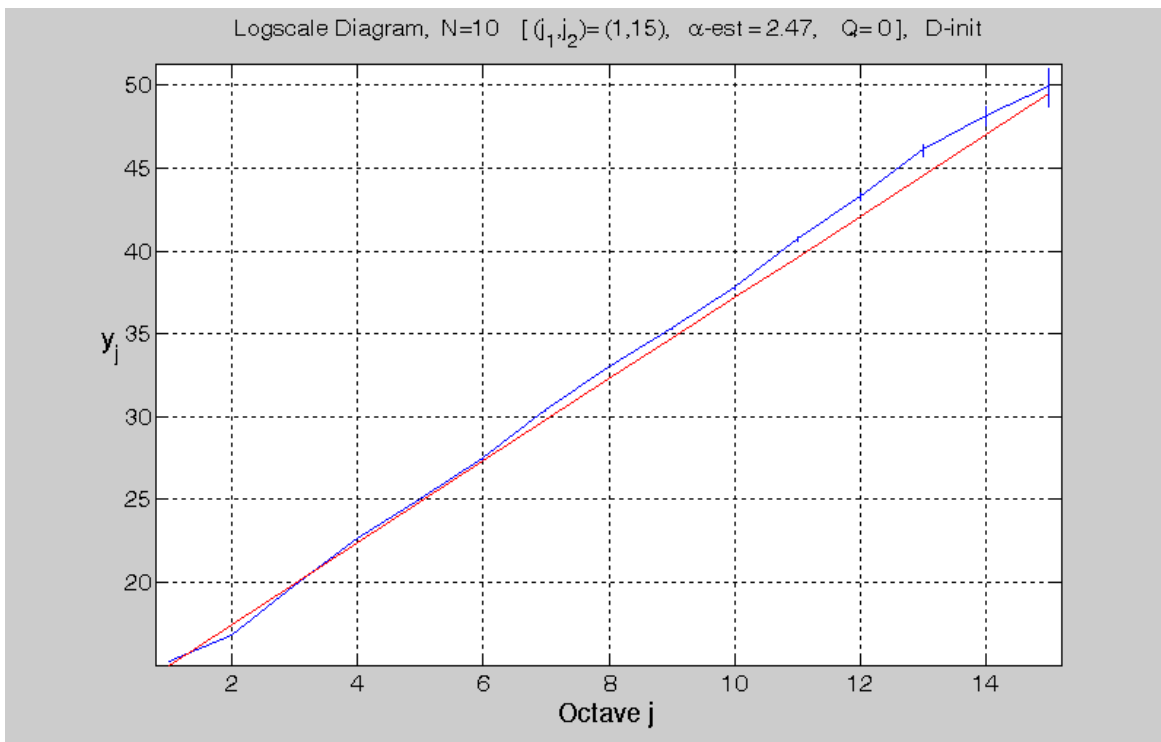


Al tratarse de un proceso que podemos asumir como estacionario, o al menos estacionario en el sentido amplio, estos resultados verifican esa hipótesis y como se observa, arroja valores que muestran un comportamiento LRD y autosimilar del proceso subyacente, entonces aquí estamos frente a un análisis como el descrito en el punto 6.2 a).

Ahora, si con la misma traza formamos el proceso de acumulación de bytes, tomando la longitud de las tramas y acumulando la suma a lo largo de todo el vector (la sintaxis de la expresión de Matlab para el proceso sería: `acumulado=cumsum(pAug89(:,2));`). Obtenemos la siguiente salida:

```
*****
Current plot held
Octaves: in_data available selected Goodness of fit (Prob of data assuming lin-regression)
         1--19.9  1--15  1--15  0.00000
Scaling parameters are: alpha (LRD) H (LRD rewrite) H=h (ss,Holder) D (frac dim, if alpha in (1,3))
Estimates: 2.466 (1.733) 0.733 1.267
CI's: [2.463, 2.468] [1.731, 1.734] [0.731, 0.734] [1.266, 1.269]
Second parameters are: cf N/A sigma^2 N/A
Estimates: 4249.3322 Work in progress
CI's: [4235.15064, 4263.54883]
```

Claramente el proceso así planteado, no es estacionario, lo cual se refleja en el valor de $\alpha = 2,466$, que al ser mayor que uno, tenemos que leer el valor de H , en la columna de $H=h$ (ss,Holder) donde vemos que $H = 0,733$, y en el grafico también se puede comprobar que la región de alineación abarca todas las octavas:



Este análisis se encuadra en el descrito en el punto 6.2 b). También es posible verificar que lo descrito en el punto 6.2 d), es válido.

Si analizamos una traza tomada de un link de 10 Gigabit Ethernet³¹ llamada 20040213-160000-1.gz, que representa una muestra de 5 minutos del tráfico sobre el link tomado el 21 de marzo de 2004 a las 16 horas, tenemos los siguientes valores:

Condiciones iniciales: $N=4$, $j_1=2$, $j_2=20$ y todos los demás parámetros de entrada igual a uno.

Se aclara que dado el enorme tamaño de las muestras (del orden de los gigabytes), el vector esta conformado con un millón de valores, correspondientes al primer millón de la muestra, y con los tiempos entre arribos calculados con la utilidad de línea de comandos Tshark perteneciente al software de análisis de protocolos Wireshark, más abajo, en el punto 6.2.3 junto con el análisis de las capturas de tráfico real, se describen en detalle el procedimiento y las sintaxis correspondientes a los distintos casos.

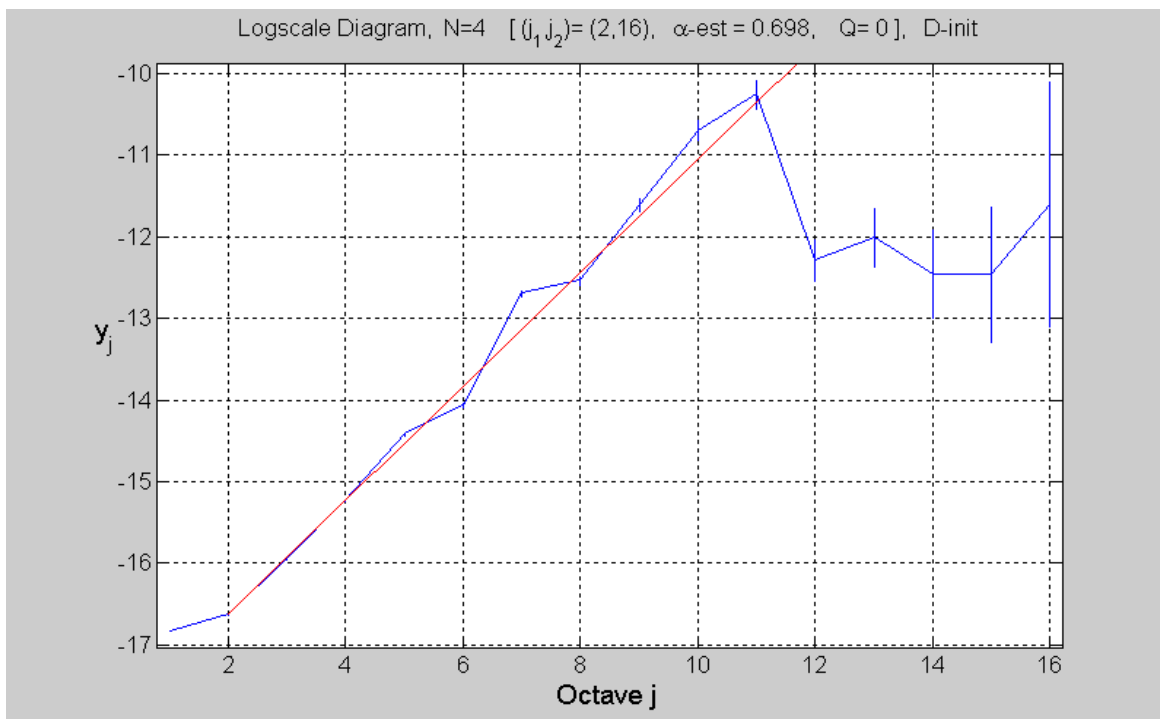
³¹ Tomado de <http://pma.nlanr.net/Special/> correspondiente a <http://pma.nlanr.net/Special/sdsc1.html>

Entonces para este ejemplo, obtenemos un vector llamado **Tera10giga16**, con la siguiente sintaxis:

```
Tshark -r 20040213-160000-1 -e frame.time_delta -T fields -c 100000 > Tera10giga16.txt
```

Luego aplicamos LDestimate al vector obtenido: **LDestimate(Tera10giga16,4,2,20,1,1,1)**

```
*****
Current plot held
Octaves:  in_data  available  selected  Goodness of fit (Prob of data assuming lin-regression)
          1--19.9  1--16      2--16      0.00000
Scaling parameters are:  alpha (LRD)  H (LRD rewrite)  H=h (ss,Holder)  D (frac dim, if alpha in (1,3))
Estimates:             0.698         (0.849)         -0.151          2.151
CI's: [0.694, 0.702] [0.847, 0.851] [-0.153, -0.149] [2.149, 2.153]
```



Como se puede observar estamos ante un caso de comportamiento bi-escala, como el descrito en el punto 6.2 f). En el gráfico se identifican dos regiones de alineación, una con $j_1 = 2, j_2 = 11$, y la otra con $j_1 = 12$, en adelante.

En líneas generales, con esto podemos apreciar la versatilidad de LDestimate, y verificar al mismo tiempo la validez de los postulados que se estudiaron en los capítulos anteriores. A continuación, para dar mayor detalle a esta discusión, se muestran los resultados obtenidos con diversas muestras de tráfico real obtenidas en Internet, de los sitios y direcciones especificados en su correspondiente descripción.

6.2.2 Análisis de capturas de tráfico real de distintas redes

A continuación y a manera de muestra de las posibilidades de estudio que brinda LDestimate, se analizan una serie de trazas que están disponibles para su uso en la investigación conforme las políticas de cada fuente (mas detalle a continuación en las correspondientes descripciones).

El análisis se realiza sobre muestras tomadas de siete tipos de enlaces:

1. Ethernet 10 Mbits.
2. Ethernet 100 Mbits.
3. Ethernet 1000 Mbits.
4. Ethernet 10000 Mbits.
5. OC12
6. OC48
7. OC192

6.2.2.1 Descripción de las Trazas

A los efectos de conocer el contexto del tráfico, se hace una breve descripción de las muestras y se facilitan todos los datos de las fuentes a los efectos de poder acceder a las mismas y disponer de toda aquella información que por razones de espacio aquí no se detalla.

6.2.2.2 Trazas de Redes Ethernet

El primer grupo de trazas utilizadas en esta discusión, fueron tomadas de redes tipo Ethernet de distintas velocidades, las cuales van desde 10 Mbps. a 10 Gbps.

6.2.2.2.1 Trazas Ethernet 10 Mb.

Son las clásicas trazas utilizadas en muchísimos trabajos y que fueron utilizadas en el trabajo seminal de Leland et.al [1]; como así también se utilizaron en ejemplos de capítulos anteriores de este trabajo.

Estas trazas se encuentran disponibles en: <http://ita.ee.lbl.gov/html/traces.html>, donde se explica con todo detalle el formato y técnicas de captura, en el link correspondiente a <http://ita.ee.lbl.gov/html/contrib/BC.html>, se puede acceder a las trazas BC-pAug89 y BC-pOct89 como así también a otras trazas ampliamente utilizadas y difundidas en la mayoría de la literatura que existe sobre estos temas, resultan de utilidad pues se pueden recrear los resultados obtenidos en aquellos estudios como así también probar nuevas técnicas de medición con las mismas.

6.2.2.2.2 Trazas Ethernet 100 Mb.

Las trazas pertenecen a la colección: “WIDE-TRANSIT 100 Megabit Ethernet Trace 2007-01-09 (Anonymized). 100 Megabit Ethernet anonymized packet trace without payload: WIDE-TRANSIT link @ Tokyo, Japan”. Se encuentran disponibles en:

[http://imdc.datcat.org/collection/1-055M-0=WIDE-TRANSIT+100+Megabit+Ethernet+Trace+2007-01-09+\(Anonymized\)](http://imdc.datcat.org/collection/1-055M-0=WIDE-TRANSIT+100+Megabit+Ethernet+Trace+2007-01-09+(Anonymized))

En el link mencionado esta la descripción de las mismas y acceso a los archivos.

Estas trazas se tomaron en el marco del proyecto “A Day in the Life of the Internet project” en el año 2007. Consisten en 50 horas de captura, separadas en segmento de 15 minutos. Se trata de un enlace externo de 100 Megabit Ethernet que conecta al backbone brindando conectividad corriente arriba de una red de area extendida situada en la ciudad de Tokio - Japón. Las muestras se tomaron a partir del 09-01-2007, hasta el 11-01-2007.

6.2.2.2.3 Trazas Gigabit Ethernet

Estas trazas corresponden a capturas realizadas durante 20 horas por NLANR PMA, en enero del 2004, mediante una tarjeta Endace DAG4.2GE dual Gigabit Ethernet network. Las trazas encuentran disponibles en la página del proyecto <http://pma.nlanr.net/Special/> en el link <http://pma.nlanr.net/Special/sdsc1.html> donde se puede acceder a un archivo de muestras tomadas cada 5 minutos en el siguiente link:

<http://pma.nlanr.net/Traces/Traces/long/sdag/1/>

6.2.2.2.4 Trazas 10 Gigabit Ethernet Cluster TeraGrid SDSC

Estas trazas se recogieron mediante un monitor del proyecto [PMA](http://pma.nlanr.net/Special/)³² (Passive Measurement and Analysis), llamado NLANR PMA OC192MON, instalado en el [Cluster TeraGrid](http://pma.nlanr.net/Special/tera1.html)³³ localizado

³² <http://pma.nlanr.net/Special/>

³³ <http://pma.nlanr.net/Special/tera1.html>

en: San Diego Supercomputer Center ([SDSC](#)), University of California, San Diego ([UCSD](#)), fueron tomadas durante una semana, partiendo del domingo 8 de febrero de 2004.

El monitor es un computador Dell PowerEdge 2650 de doble procesador corriendo a 3.0GHz, con 2GB de RAM, y dos grupos SCSI RAID 0 2x146GB SCSI RAID 0. La captura de datos se efectuó mediante un par de tarjetas de medición Endace DAG6.1 OC192c (<http://www.endace.com/dag6.1.htm>), operando en el modo 10Gigabit-LAN. Algunos errores son residuales y poco visibles dentro de la traza.

El formato de archivo de rastreo es extensible Endace del Registro Formato (FER), fija el tamaño de registro en 88 bytes cada uno. Existe una considerable actividad de VLAN, y una buena parte del tráfico es local del SDSC, seguido de la comunicación entre Teragrid SDSC y CPS. Actualmente, los archivos se dividen en intervalos de 5 minutos, con una carga equilibrada de dos enlaces redundantes objeto de seguimiento.

Granularidad del Timestamp: el oscilador principal funciona a 100 MHz, y el motor genera una marca para el timestamp con una frecuencia de 66 MHz, por lo tanto cada marca de reloj representa 16 nanosegundos, contando con 26 bits de resolución.

Las trazas están disponibles en el servidor FTP:

<ftp://pma.nlanr.net/traces/long/tera/1/>

6.2.2.3 Trazas sobre Fibra Óptica

Tres grupos de trazas sobre fibra óptica, OC12 – OC48, y OC192; fueron facilitadas por el Proyecto [CAIDA](#)³⁴ (CAIDA: The Cooperative Association for Internet Data Analysis).

Los datos de los links OC12 y OC48 pertenecen a un grupo denominado [The CAIDA Anonymized 2007 Internet Traces Dataset](#)³⁵.

Los datos del link OC192 pertenecen a un grupo denominado [The CAIDA Anonymized 2009 Internet Traces Dataset](#)³⁶.

6.2.2.3.1 Trazas OC12 AMPHAT Miami Florida

Este conjunto de datos contiene las trazas anonimizadas correspondientes al tráfico del monitor CAIDA AMPATH en un enlace en la OC12 de AMPATH Internet, capturados durante el 2007. AMPATH son las siglas de High Performance International Exchange Point in Miami, Florida, estas instalaciones facilitan las tareas de investigación e intercambio de datos entre investigadores de redes, como así también el desarrollo de las mismas.

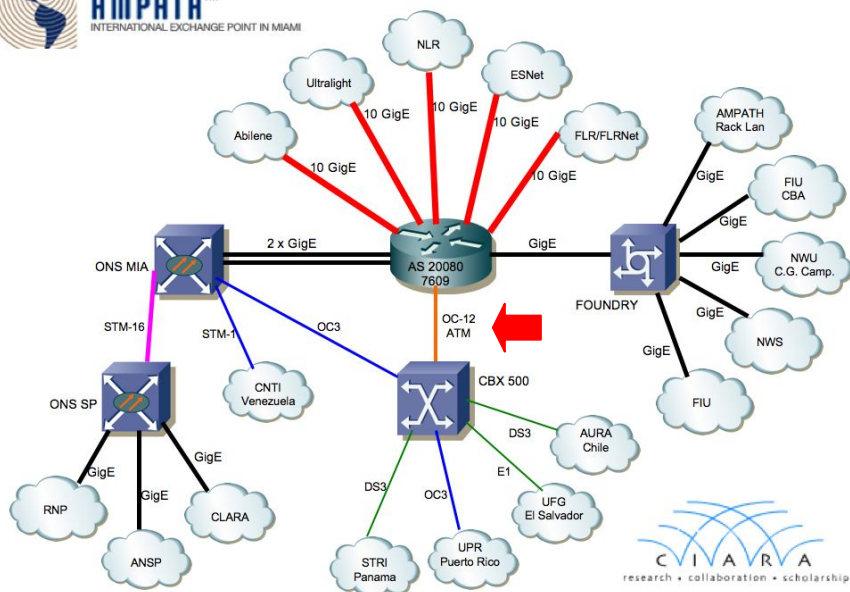
El monitor pasivo AMPATH OC12 (ampath-oc12) fue originalmente instalado como parte de la infraestructura del Proyecto NLANR PMA que finalizó en 2006, y a partir de allí el Proyecto CAIDA se responsabilizó por las tareas de desarrollo del mismo. El monitor controla un link OC12 que une América Central y América del Sur con el Centro en Miami y el tráfico pertenece a las siguientes redes cuya topología se muestra en la figura de abajo.

- CNTI (Venezuela).
- UPR (Puerto Rico), includes traffic from the Arecibo Observatory.
- STRI (Panama).
- UFG (El Salvador).
- AURA/gemini (Chile)

³⁴ <http://www.caida.org/home/>

³⁵ http://www.caida.org/data/passive/passive_2007_dataset.xml

³⁶ http://www.caida.org/data/passive/passive_2009_dataset.xml



El estudio con este monitor se inició el 21 de febrero de 2007 y terminó el 18 de marzo de 2008. Los informes están disponibles en:

<http://www.caida.org/data/realtime/passive/?monitor=miami>

6.2.2.3.2 Trazas OC 48 Link entre ISP's

Estas trazas se tomaron en el marco del proyecto CAIDA y su denominación es: "The CAIDA Anonymized OC48 Traces Dataset (2002-2003)".³⁷ Los datos se capturaron en un link OC48 que vincula dos grandes ISP de la costa oeste de los Estados Unidos, durante los años 2007 y 2008. Los datos se encuentran anonimizados con CryptoPAN³⁸. Las trazas se encuentran disponibles en dos tamaños de muestra, archivos grandes de una hora de captura y archivos más pequeños de 5 minutos de captura a los efectos de poder procesarlos debido a su gran tamaño. Particularmente las trazas contienen encabezados IP (48 Bytes) y la precisión del timestamp es del orden de los microsegundos. Para mas detalles se puede recurrir a <http://imdc.datcat.org/collection/1-0016-F=CAIDA+OC48+Traces+2002-08-14>.

6.2.2.3.3 Trazas OC192 Datacenter Equinix en Chicago

Estas muestras empezaron a ser tomadas durante el año 2008, en el ámbito del proyecto CAIDA Day in the Life of the Internet (DITL); y forman parte de la colección: CAIDA Anonymized 2008 Internet Traces Dataset³⁹. Las trazas se encuentran disponibles en el link de la referencia y pertenecen a un Datacenter llamado Equinix.

Monitor Pasivo: equinix-chicago: Este monitor se encuentra localizado en el Datacenter Equinix de la ciudad de Chicago (Illinois), y esta conectado a un link de backbone OC192 (9953 Mbps); el cual vincula dos ISP's Tier1, entre Chicago IL, y Seattle, WA.

La infraestructura consiste en dos máquinas Intel Dual-Core Xeon 3.00GHz CPUs, con 8 GB de memoria y 1.3 TB de disco RAID5, corren Linux 2.6.15 y software DAG version dag-2.5.7.1. Están numeradas como 1 y 2 equipadas con 2 tarjetas Endace 6.2 DAG, cada tarjeta monitorea una dirección llamando dirección A al vinculo entre Seattle y Chicago; y dirección B al vinculo entre Chicago y Seattle.

³⁷ http://www.caida.org/data/passive/passive_oc48_dataset.xml

³⁸ <http://www.cc.gatech.edu/computing/Telecomm/projects/cryptopan/>

³⁹ http://www.caida.org/data/passive/passive_2008_dataset.xml y <http://www.caida.org/data/monitors/passive-equinix-chicago.xml>

6.2.3 Procedimiento previo aplicado a las trazas

Debido al importante tamaño de los archivos de las trazas (típicamente del orden de los gigabytes), para poder procesarlos con PC's convencionales, a todas las trazas se las sometió al siguiente tratamiento.

Se utiliza el software Wireshark⁴⁰, que es un conocido analizador de protocolos basado en tcpdump, que permite manipular trazas que utilicen formato compatible con el tcpdump y además cuenta con una utilidad de línea de comandos llamada Tshark⁴¹, que resulta particularmente apropiada para esta tarea como se muestra a continuación:

- Se toma el primer millón de tramas de la traza y se lo convierte en un archivo con la extensión .pcap. La sintaxis del comando es: `Tshark -r [nombre de la traza] -c 1000000 -w [nombre.pcap]`.
- Si se quiere trabajar con los tiempos entre arribos, creamos el archivo correspondiente, de la siguiente manera: `Tshark -r [nombre.pcap] -e frame.time_delta -T fields > nombre.txt`. Esto lo que hace es, leer el archivo .pcap que se creó con el millón de trazas, y lo filtra mediante el contenido del campo timestamp, del que a su vez establece la diferencia con la lectura anterior creando el valor tiempo entre arribos para cada trama y luego guarda el archivo en formato ASCII.
- Del mismo modo si se desea trabajar con la longitud en bytes de la trama, se utiliza el campo frame.len para el filtrado de la siguiente forma: `Tshark -r [nombre.pcap] -e frame.len -T fields > nombre.txt`. Obteniendo de esta forma la salida en formato ASCII que es la más cómoda para poder utilizarla con el Matlab.
- Se aclara que es posible crear archivos con un número arbitrario de campos, como podría ser tiempos entre arribos y longitud de la trama, sencillamente repitiendo en la misma línea `-e [nombre de campo1], -e[nombre de campo2]`, se recomienda la consulta de la documentación de Tshark como así también la del Wireshark para ver la importante capacidad de filtrado y análisis de protocolos que tiene este software y que puede ser de mucha utilidad en trabajos donde se requiere la utilización de archivos con formato .pcap.

Aclarados todos los detalles que permitirán repetir los análisis aquí expuestos, a continuación se muestra el resumen de los resultados del análisis efectuado a estas trazas, con el primer millón de datos para cada una de ellas, los datos corresponden a tiempos entre arribos. Por economía de espacio no se incluyen las pantallas de salida con sus correspondientes gráficas⁴² pues aumentaría inútilmente la cantidad de páginas, dado que ya se discutió la utilización de LDestimate en el punto 6.2, junto a sus correspondientes alternativas en los ejemplos de 6.2.1, donde se muestra el significado de los valores en función del contexto. Sin embargo se facilitan todas las referencias en las descripciones de cada traza, pues eso brinda información del contexto en que fueron tomadas, y permite una correcta interpretación de los resultados que figuran en la tabla del siguiente punto.

6.2.4 Tabla de Resultados Obtenidos con LDestimate

La tabla exhibe los resultados de la estimación del parámetro H realizada con la script de Darryl Veitch, tomando una muestra de cada traza, conforme se describe en 6.2.3. Se aclara que las condiciones iniciales para todas las estimaciones son las mismas, es decir inicialmente se fijan los parámetros como: $N=4$ momentos desvanecientes, $j_1=2$; $j_2=20$ y los tres parámetros restantes igual a uno, esto significa que la script calculará en función de los resultados iniciales para cada caso lo siguiente:

⁴⁰ <http://www.wireshark.org/>

⁴¹ <http://www.wireshark.org/docs/man-pages/tshark.html>

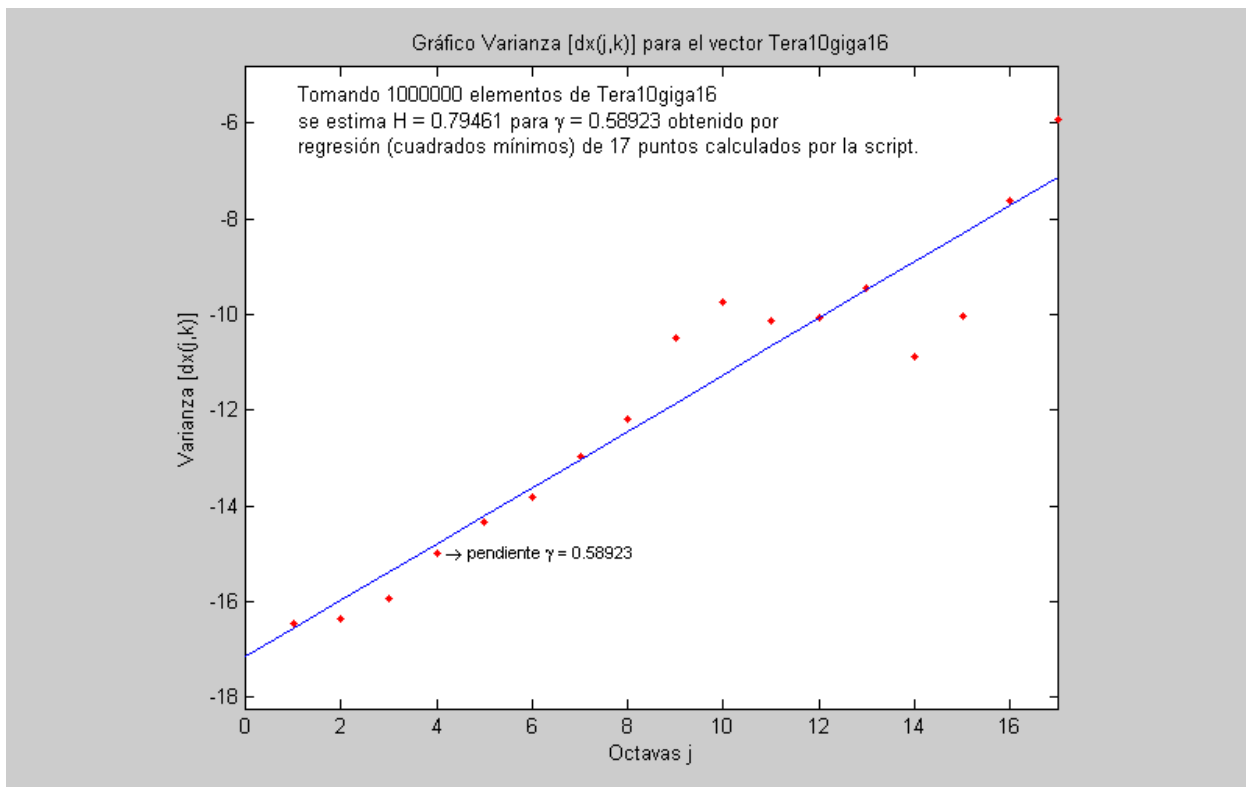
⁴² Se recomienda como ejercicio, repetir el análisis con las trazas y verificar los resultados aquí expuestos

1. El mejor valor para j_1 como función del j_2 que se encuentre como óptimo.
2. Tomando los valores sugeridos por la script para j_1 y j_2 , se repite la estimación obteniendo los datos que se muestran en la tabla.

Traza	Parámetro	j_1	j_2	α	α [95%]	H	H [95%]
10 Mbit. pAug89.TL		9	16	0,642	[0.590, 0.693]	0,821	[0.795, 0.846]
10 Mbit. pOct89.TL		6	16	0,51	[0.494, 0.527]	0,755	[0.747, 0.763]
100 Mbit. Tokio (200701090800)		8	16	0,315	[0.280, 0.349]	0,657	[0.640, 0.675]
100 Mbit. Tokio (200701091200)		6	16	0,304	[0.288, 0.321]	0,652	[0.644, 0.661]
Gigabit-Ethernet (20040130-132000-0)		10	16	0,557	[0.479, 0.634]	0,778	[0.740, 0.817]
10 Gigabit Ethernet (20040212-130000-0)		10	14	0,508	[0.286, 0.731]	0,754	[0.643, 0.865]
ampath-oc12.20070109.dag0.20070109-0000.anon		10	16	0,693	[0.616, 0.771]	0,847	[0.808, 0.885]
ampath-oc12.20070109.dag0.20070109-1500.anon		12	16	0,303	[0.106, 0.499]	0,651	[0.553, 0.750]
OC48-20020814-103000-0-anon.pcap		5	16	0,195	[0.184, 0.207]	0,598	[0.592, 0.603]
OC48-20020814-115000-0-anon.pcap		4	16	0,175	[0.167, 0.183]	0,588	[0.584, 0.592]
equinix-chicago.dirA.20090115-060100.UTC.anon		9	16	0,461	[0.410, 0.512]	0,73	[0.705, 0.756]
equinix-chicago.dirB.20090115-060100.UTC.anon		8	16	0,282	[0.248, 0.317]	0,643	[0.624, 0.659]

6.3 Discusión de la estimación mediante la script varianzaW.m

La principal ventaja de la script “varianzaW.m”, sobre el método Varianza-Tiempo en el cual está basada, es que el análisis y la regresión lo efectúa en el dominio de la escala (frecuencia) y debido a las características de la descomposición wavelets (análisis multiresolución); con esta script, la estimación se desacopla de la fuerte dependencia que hay en el dominio temporal, donde los promedios están fuertemente correlacionados, mientras que en el dominio de la escala, los coeficientes wavelet están incorrelados. Si bien es superior en este aspecto, a los métodos cualitativos, cabe aclarar que sus resultados carecen de la exactitud de LDestimate, principalmente debido a que no se toma ninguna medida de selección de octavas y se utilizan todas las disponibles para efectuar la regresión o cual introduce un importante sesgo. Su ventaja radica entonces, en la facilidad de uso e inmediatez de implementación lo que la hace adecuada para una rápida estimación de comportamiento o preselección de trazas en una simulación. A manera de ejemplo, en la siguiente figura, se muestra un análisis efectuado con esta script.



En la figura se observa la salida de la script VarianzaW.m, para la traza de 10 Gigabit Ethernet 20040213-160000-1 que corresponde al Cluster Teragrid estudiada en [6.2.1](#), y que se mostró con

un comportamiento bi-escala, aquí ese comportamiento no se hace evidente sin embargo, resulta interesante observar que a partir de la octava 10 existe una fuerte oscilación de la varianza con respecto a la recta de regresión, como así también es posible observar una alineación con poco apartamiento desde la octava 2 hasta la 10, resultando la estimación de $H = 0,7946$, que es aproximada a la estimación de LDestimate si se consideran las octavas 2 a 16, como primera zona de alineación.

6.4 Tabla Comparativa de Resultados Obtenidos con Distintos Métodos de Estimación de “H”.

La siguiente tabla, muestra los resultados de la estimación de H , utilizando las aplicaciones desarrolladas en este trabajo, junto al resultado obtenido mediante la script de Darryl Veitch.

Metodo de Estimación de H Traza	V a r i a n z a	R e s c a l a d o	L D i g - R e s c a l a	V O c t a v a s
10 Mbit. pAug89.TL	0,802	0,737	0,821	0,756
10 Mbit. pOct89.TL	0,844	0,839	0,755	0,785
100 Mbit. Tokio (200701090800)	0,675	0,663	0,657	0,667
100 Mbit. Tokio (200701091200)	0,696	0,686	0,652	0,625
Gigabit-Ethernet (20040130-132000-0)	0,69	0,729	0,778	0,682
10 Gigabit Ethernet (20040212-130000-0)	0,93	0,979	0,754	0,883
ampath-oc12.20070109.dag0.20070109-0000.anon	0,685	0,727	0,847	0,683
ampath-oc12.20070109.dag0.20070109-1500.anon	0,638	0,678	0,651	0,684
OC48-20020814-103000-0-anon.pcap	0,58	0,651	0,598	0,626
OC48-20020814-115000-0-anon.pcap	0,66	0,645	0,588	0,637
equinix-chicago.dirA.20090115-060100.UTC.anon	0,653	0,667	0,73	0,642
equinix-chicago.dirB.20090115-060100.UTC.anon	0,693	0,698	0,643	0,637

Resultados en cursiva, se encuentran los resultados de la estimación de H , por el método del diagrama log-escala implementado mediante la script LDestimate de Darryl Veitch, que es el de mayor precisión y menor sesgo.

6.4.1 Acerca de la Exactitud

Con respecto al tema de la exactitud y sesgo de los distintos métodos, existen detallados estudios comparativos como se menciona en el capítulo 1, entre los que se encuentra el trabajo efectuado por Jeong, H-D. [28], el cual detalla una serie de resultados que ponen de manifiesto la ventaja de los métodos basados en wavelets con relación a estos tópicos. Si bien no es objetivo de este trabajo profundizar sobre los pormenores del análisis de la exactitud y precisión en si mismas, si se manifestó claramente, que la elección de los métodos se basaba en las conclusiones de aquellos trabajos que se consideraban más completos y pertinentes, estableciendo las referencias debidamente, a los efectos de ser consultados por quienes quieran profundizar en el tema.

Entonces, a continuación, se muestran a manera de resumen los cuadros comparativos extraídos de [28], que contienen los resultados del análisis de la exactitud y sesgo para los distintos métodos que se abordaron en el mismo.

Methods	Mean Values of Estimated H and ΔH				Methods	Mean Values of Estimated H and ΔH			
	.6		.7			.8		.9	
	\hat{H}	$\Delta H(\%)$	\hat{H}	$\Delta H(\%)$		\hat{H}	$\Delta H(\%)$	\hat{H}	$\Delta H(\%)$
Wavelet-based	.6002 (.573,.628)	+0.026	.7006 (.673,.728)	+0.092	Wavelet-based	.8009 (.773,.828)	+0.110	.9010 (.874,.929)	+0.108
Whittle's MLE	.6003 (.591,.610)	+0.043	.7003 (.691,.710)	+0.044	Whittle's MLE	.8004 (.791,.810)	+0.044	.9004 (.891,.909)	+0.043
Periodogram	.6008 (.599,.603)	+0.128	.7025 (.700,.705)	+0.358	Periodogram	.8040 (.802,.806)	+0.496	.9054 (.903,.908)	+0.598
R/S-statistic	.6277 (.624,.632)	+4.623	.7118 (.708,.716)	+1.689	R/S-statistic	.7916 (.787,.796)	-1.053	.8621 (.857,.867)	-4.210
Variance-time	.5964 (.594,.599)	-0.608	.6917 (.689,.695)	-1.192	Variance-time	.7814 (.778,.785)	-2.327	.8600 (.856,.864)	-4.439
IDC(t)	.5968 (.594,.600)	-0.528	.6919 (.689,.695)	-1.153	IDC(t)	.7812 (.778,.785)	-2.349	.8593 (.855,.863)	-4.520

Estos cuadros muestran el error relativo de las estimaciones de H , de los distintos métodos, tomando como base una secuencia patrón de 2^{15} elementos, generada mediante el algoritmo de Durbin-Levinson,⁴³ que permite obtener secuencias de FGN con valores muy exactos de H , y en este caso se utilizaron secuencias con $H = 0,6; 0,7; 0,8$ y $0,9$ respectivamente, y se obtuvieron los valores mostrados.

6.4.2 Acerca de la Precisión

En su trabajo Jeong [28], manifiesta que los métodos de estimación basados en Wavelets y el estimador de Whittle MLE son substancialmente mas precisos que los otros métodos habitualmente utilizados para la estimación de H , pues un buen estimador no solamente debe producir valores cercanos al real si no que además las estimaciones deben tener valores pequeños de varianza. A continuación se muestra un cuadro extraído de [28], que contiene las varianzas de la estimación de H , para cada uno de los métodos.

Methods	Variances of Estimated H			
	.6	.7	.8	.9
Wavelet-based	1.892e-04	1.922e-04	1.951e-04	1.985e-04
Whittle's MLE	1.093e-05	1.145e-05	1.179e-05	1.217e-05
Periodogram	1.288e-04	1.306e-04	1.291e-04	1.279e-04
R/S-statistic	3.971e-04	4.724e-04	5.400e-04	5.669e-04
Variance-time	1.928e-04	2.571e-04	3.405e-04	3.826e-04
IDC(t)	2.227e-04	2.812e-04	3.613e-04	3.947e-04

Para generar los valores de este cuadro se utilizaron 100 secuencias para cada valor de H . Aquí se observa que la varianza de los estimadores de Whittle y Periodograma son mas bajas que el de wavelets, lo cual es consistente pues el estimador de Whittle a su vez se basa en el Periodograma pero en general la varianza de los tres es mucho menor que la de los demás métodos.

⁴³ Si bien tiene un alto costo computacional tiene mucha exactitud en el valor de H de la secuencia generada, y en el apéndice C1 del trabajo citado se encuentra el código fuente escrito en C para la generación de secuencias FGN con un determinado H .

6.5 Discusión sobre la Síntesis de Procesos Autosimilares

Si bien es conocido que existen muchos métodos para la generación de secuencias autosimilares con mucha exactitud en cuanto al parámetro H , tienen el defecto de tener costos computacionales extremadamente altos, esto tiene utilidad si se van hacer mediciones muy precisas o bien contrastar con resultados numéricos la validez de procedimientos de análisis como se describió mas arriba; pero en el ámbito de la ingeniería muchas veces no tiene sentido una extrema exactitud si con esto se sacrifica la aplicación en tiempo real de de una determinada secuencia o bien disponer de las mismas en tiempos razonables para determinadas simulaciones.

La selección de un generador de trazas sintéticas con determinadas características se centra en dos aspectos.

1. Cuan fiable es para representar las características escogidas.
2. Cuan rápido o eficiente es para generar trazas de un determinado tamaño.

Los dos modelos mas frecuentemente utilizados como generadores son los F-ARIMA (fraccional autorregresivos integrados de media móvil), y el FGN (Ruido Gaussiano Fraccional) como se mostró en los capítulos anteriores, el FGN presenta ventajas en cuanto a parsimonia y el control de las distribuciones marginales por ser gaussiano. También, como se mencionó arriba, para la generación, se utilizan otros algoritmos como el RMA (Randon Mid-Point) conocido como algoritmo de punto medio, Durbin-Levinson que genera secuencias autosimilares exactas de FGN y otros que tienen como principal desventaja, la de tener costo computacional excesivamente elevado. Como ejemplo se muestra una tabla extraída de [28], que contiene los tiempos de generación de una secuencia de números que representan un proceso autosimilar, generados con el algoritmo de Durbin-Levinson utilizando una PC con S.O. SunOS 5.7, procesador Pentium II @233 MHz, y 512 MB de memoria RAM.

Sequence of					
32,768 Numbers	65,536 Numbers	131,072 Numbers	262,144 Numbers	524,288 Numbers	1,048,576 Numbers
Mean running time (<i>hour:minute:second</i>)					
00:01:44	00:07:28	00:39:08	02:41:09	10:55:04	44:13:37

Si observamos en la tabla, vemos que la secuencia para 2^{20} números, tarda 44 horas en generarse. Como contrapartida los métodos basados en wavelet para la generación de secuencias basadas en FGN están el orden de los segundos, para equipos de similares características.

6.5.1 Comparando tiempos para generación de trazas sintéticas

Para ilustrar un poco lo comentado en el punto anterior, a continuación se muestran los tiempos de generación para una secuencia autosimilar exponente $H = 0,8$ y 2^{20} números, calculados mediante tres scripts distintas:

1. Script algoritmo de punto medio extraído de [24], llamada PuntoMedioMBF.m
2. Script FGNW.m
3. Script FGNIFourier.m

El cálculo se hace con tres computadoras distintas con las siguientes características:

1. PC1: PC procesador Pentium Celeron @2,53 MHz. 1 GB RAM S.O: Windows XP SP3.
2. PC2: PC procesador Pentium 4 Core2 Duo @3 GHz. 4 GB. RAM S.O: Windows XP SP3.
3. PC3 :Notebook procesador Pentium T7200 Core2 Duo @2 GHz. 4 GB. RAM S.O. Windows Vista Home Premium SP1.

Los resultados se muestran en la siguiente tabla:

Tiempo en segundos para calcular 2^{20} números				
METODO \ EQUIPO	PC1	PC2	PC3	
PuntoMedioMBF.m	$3,00 \times 10^4$	$7,57 \times 10^3$	$1,45 \times 10^3$	
FGNW.m	7,797	2,094	3,494	
FGNIFourier.m	2,843	1,125	1,451	

Con el método basado en desplazamiento de punto medio y tomando desde la PC mas lenta a la mas rápida, se tardan: 5 horas, 2 horas con 6 minutos, y 24 minutos respectivamente; mientras que los otros métodos, tardan apenas 8 segundos, si tomamos el caso mas desfavorable con la PC mas lenta. Esto no deja lugar a dudas en cuanto la eficiencia del método a elegir para simulación. Lógicamente si lo que se quiere es validar un método de detección o medición donde la exactitud y la precisión se imponen, sería otro el criterio de selección.

En el capítulo 3 punto [3.11.5](#), se comentó acerca de las ventajas que ofrece el FGN como modelo de tráfico. Además, la estructura de la descomposición mediante wavelets, coincide en forma natural con la estructura de dependencia de rango largo de los procesos autosimilares, cuyo caso típico son precisamente los procesos FGN. Ambas circunstancias nos permiten atender, por una parte el criterio de parsimonia, pues con el valor de H , se controla la persistencia del comportamiento, y por otra parte, la descomposición wavelet asegura la autosimilitud, que permite mantener esta persistencia a distintas escalas temporales, por último, gracias a la implementación de los filtros wavelets, se mantiene un costo computacional razonable. Evidentemente la síntesis del FGN mediante wavelets, cumple con los criterios enunciados en el capítulo 1 punto [1.2](#) con relación a las características deseables de un modelo de tráfico.

6.5.2 Acerca del uso de las script FGNW.m y FGNIFourier.m

En el capítulo 5 en los puntos [5.3.1](#) y [5.3.2](#), se plantea la utilización de dos aplicaciones llamadas FGNW.m y FGNIFourier.m, las cuales sirven para la generación de secuencias autosimilares con un determinado H ; ambas cumplen con los criterios mencionados arriba. Para demostrar la validez de esta afirmación, y también la similitud del comportamiento de las trazas generadas con relación al tráfico que representan, se desarrollo una script llamada “correlacionFBM.m”,⁴⁴ que muestra de manera gráfica el comportamiento de la correlación de un proceso cualquiera, y al mismo tiempo, la curva de correlación teórica de un FGN para un valor determinado de H , esto permite evaluar visualmente cuanto se aparta el proceso bajo estudio del comportamiento teórico, y por ende permite validar la utilización de las trazas. Se discute a continuación, en que manera puede contribuir al estudio y análisis de procesos, la simulación de trazas con FGNW.m y FGNIFourier.m. Para esta discusión se utiliza también la script correlacionFBM.m.

6.5.2.1 La script correlacionFBM.m

La utilización de la script correlacionFBM.m, solo requiere el ingreso de tres argumentos, el nombre de la traza o vector de datos que queremos analizar, el valor teórico o supuesto de H , y por último el valor de “lag” o longitud que tomará el valor máximo de holgura temporal $k = (t-s)$ ⁴⁵. La script correlacionFBM.m entrega como salida, un grafico conteniendo la curva de autocorrelación de un FGN teórico con un valor de H indicado por el usuario, junto a la curva de autocorrelación de la serie ingresada como dato. A continuación se muestran análisis de trazas sintéticas generadas con FGNW.m y FGNIFourier.m y otras trazas pertenecientes a muestras de tráfico real, para poner en contexto la utilidad de las scripts o aplicaciones.

⁴⁴ Se encuentra disponible en el Apéndice 1, punto 9.1.13

⁴⁵ Recordar la expresión de la correlación para FGN capítulo 4 punto 4.6

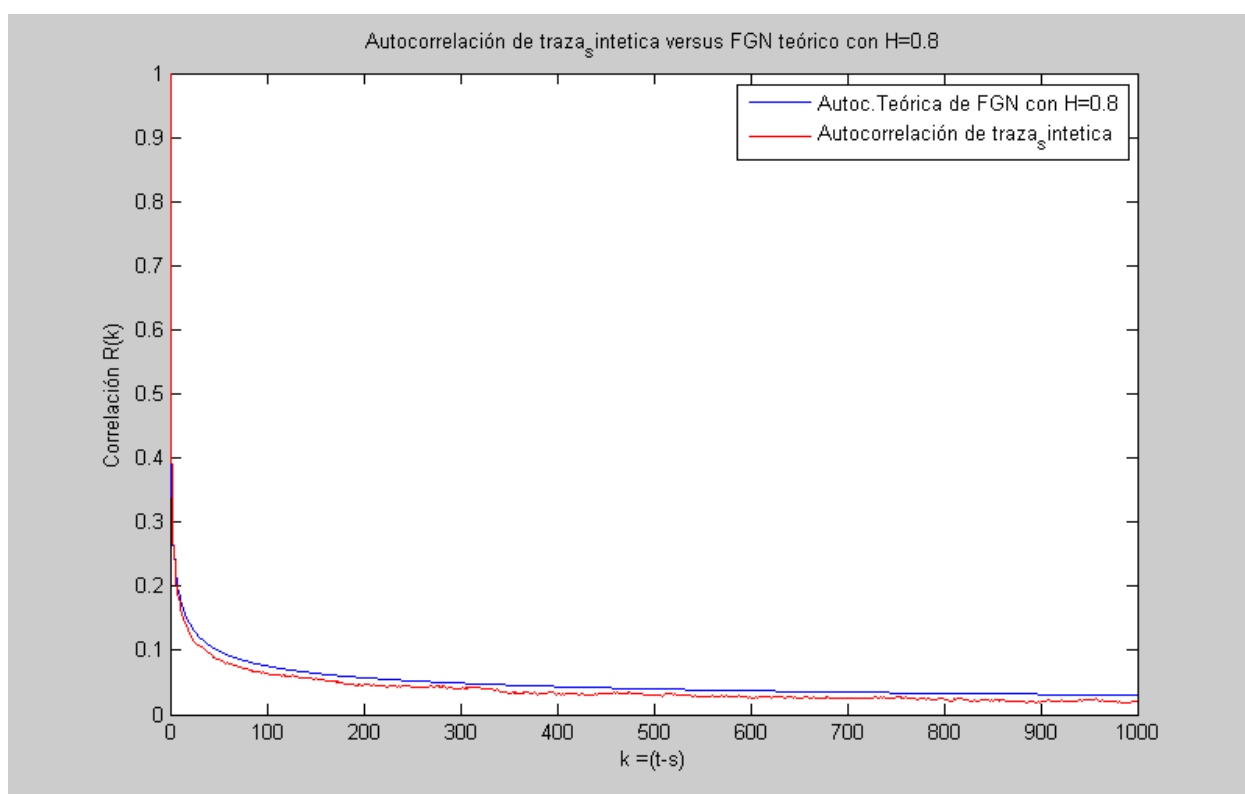
6.5.2.2 Analizando FGNW.m

Con un vector sintético simulando un FGN de $H=0,8$ con 2^{20} números, generado mediante FGNW.m; y al que llamamos “traza_sintética”, le aplicamos la script correlacionFBM.m de la siguiente forma:

1. Colocamos a la script en el directorio de trabajo del Matlab
2. Sobre el nombre de la script, hacemos un clic con el botón derecho, y luego “run”
3. Contestamos las preguntas como sigue:

Vector de datos? traza_sintetica
Ingrese el valor de k (lags) 1000
ingrese el valor de H (teórico) .8

Obtenemos entonces la siguiente salida:



Como se puede apreciar sigue en forma casi exacta el comportamiento de autocorrelación del FGN teórico.

6.5.2.3 Analizando FGNIFourier.m

Si aplicamos la script a vectores sintéticos elaborados con la script FGNIFourier.m, vemos que aproxima bastante bien el comportamiento teórico, con la ventaja que significa su mayor velocidad, y sin necesidad ninguna script accesoria o de terceros (como es el caso de FGNW.m), utilizando solamente los recursos de MATLAB®, esta característica convierte a esta script, en una buena elección para las simulaciones.

En primer lugar generamos la traza con la siguiente sintaxis:

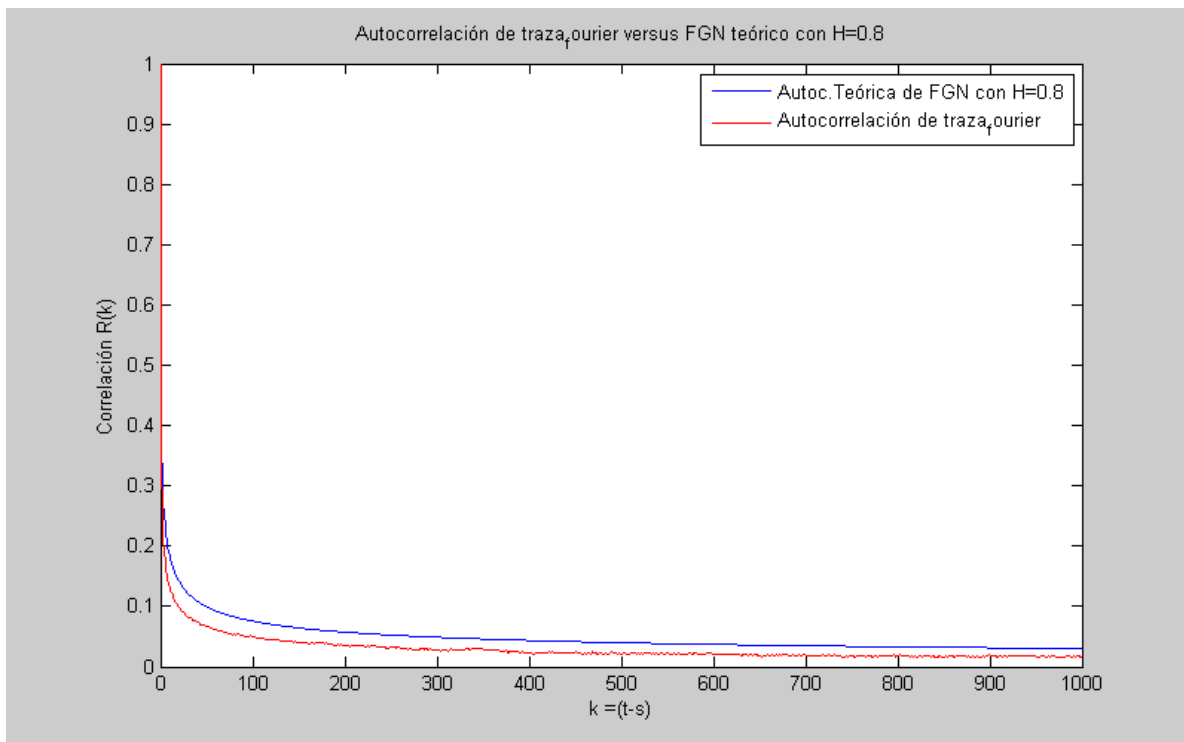
```
traza_fourier=FGNIFourier(2^20,.8);
```

En segundo lugar, aplicamos correlacionFBM.m como sigue:

Vector de datos? traza_fourier

Ingrese el valor de k (lags) 1000
ingrese el valor de H (teórico) .8

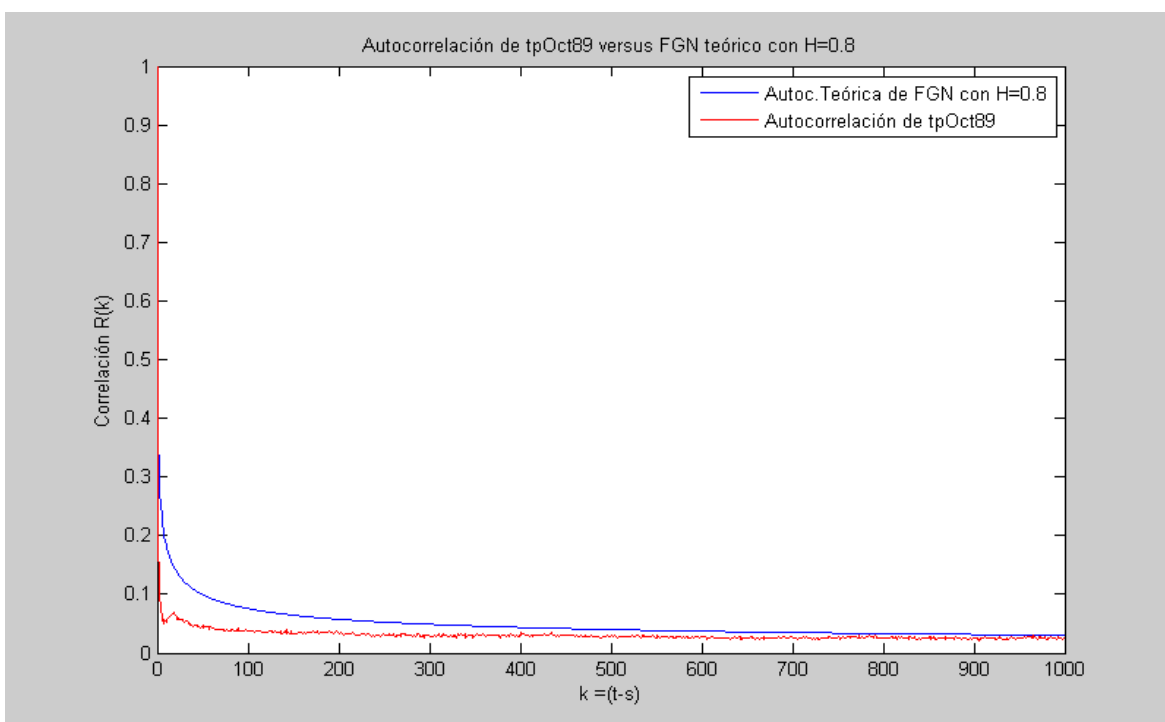
Obtenemos la siguiente gráfica:



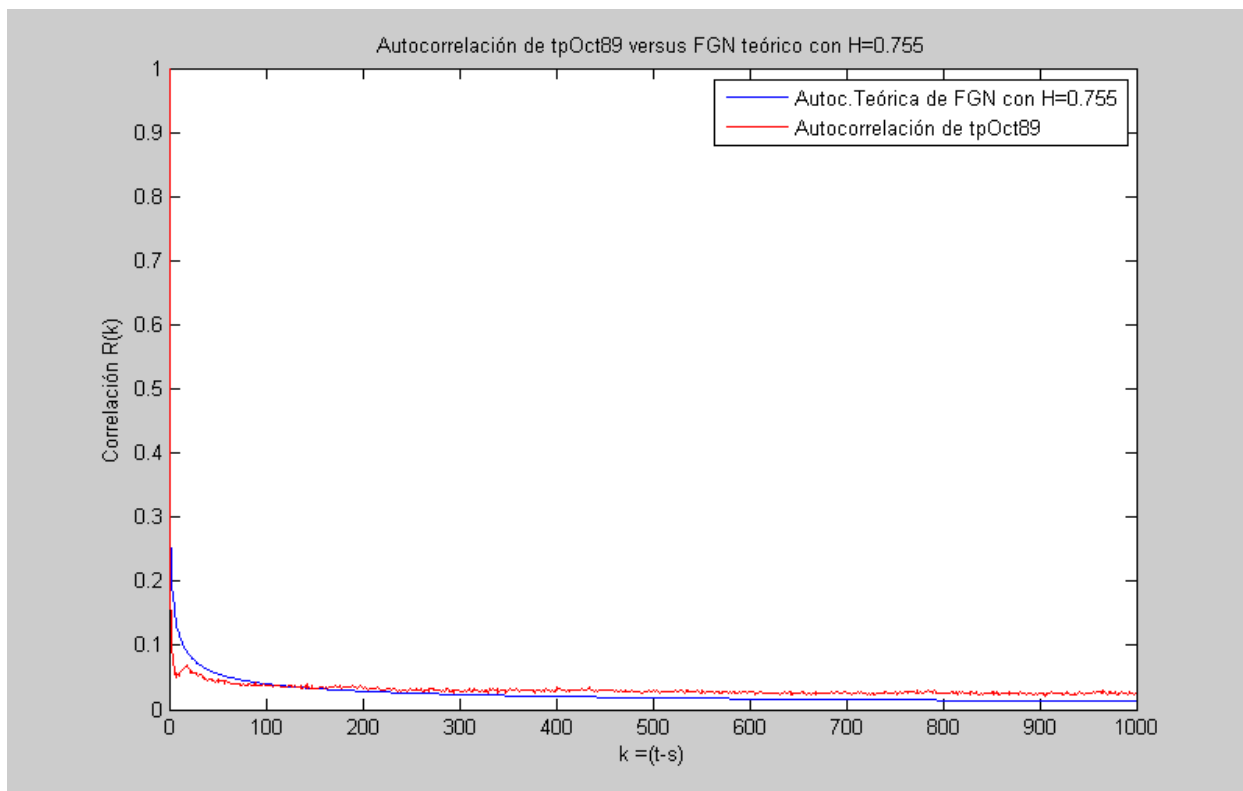
Al aplicar `LDestimate` al vector `traza_fourier`, obtenemos una estimación de $H = 0,786$, que significa un error relativo de 1,768 % para el $H = 0.8$, que fijamos `traza_fourier`, este valor es compatible con lo que se observa en la curva de correlación, que en este caso sigue o copia a la curva teórica levemente por debajo de la misma.

6.5.3 La script `correlacionFBM.m` en el contexto de tráfico real

Si se aplica `correlacionFBM.m` a un vector que contiene los tiempos entre arribos de la traza `pOct89`, se nombra como `tpOct89` a dicho vector y se obtiene la siguiente salida:



Como se puede apreciar el comportamiento de la autocorrelación del proceso de tiempos entre arribos sigue la misma forma que el teórico, pero muestra una menor divergencia en el origen, y por debajo del teórico, y este comportamiento es consistente con el hecho de que el parámetro H , estimado para este vector arroja un valor menor a 0.8, de hecho el valor estimado por `LDestimate.m`, es 0,755 como se puede ver en la tabla del punto 6.4. Entonces si a la script le indico que el valor teórico de H para graficar, es de 0,755, obtenemos la siguiente salida:



La conclusión que podemos sacar de lo observado en este análisis, es que evidentemente el FGN es una excelente aproximación como modelo de tráfico en estas circunstancias, y además la script `correlacionFBM.m`, nos puede ayudar con información adicional para una traza de comportamiento “dudoso”, pues si el comportamiento sigue como sucede en este ejemplo, la misma forma que la teórica, se puede asumir la gaussianidad de los datos.

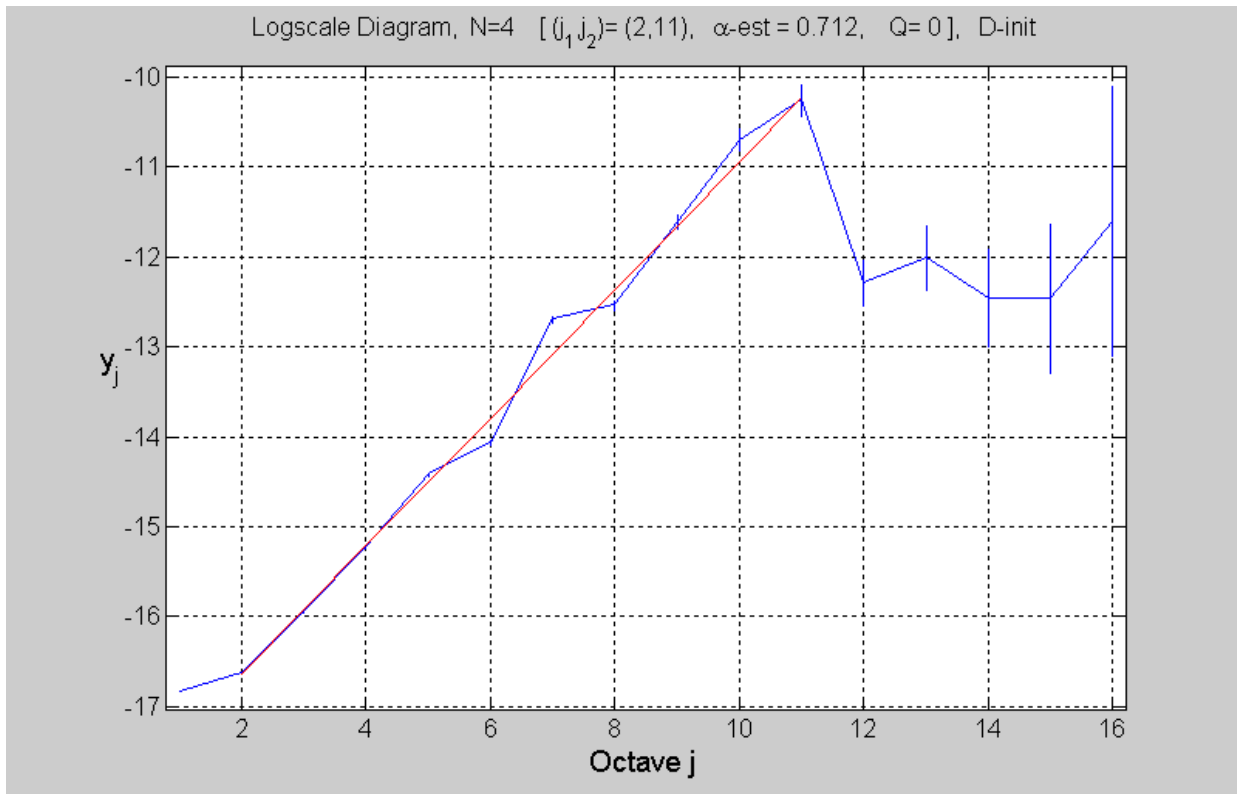
Para ilustrar un poco el escenario del anterior comentario, se puede tomar la traza de 10 Gigabit Ethernet (20040213-160000-1)⁴⁶, perteneciente al cluster TeraGrid del SDSC que al aplicar la script `LDestimate`, se ve claramente que tiene dos zonas de alineación, una primer zona con un fuerte comportamiento autosimilar y luego otra en la que de hecho parece un proceso Poisson y da muestras de SRD o proceso incorrelado. Esto se evidencia en las gráficas que se muestran a continuación aplicando la misma metodología que se mostró mas arriba.

Entonces dividimos el análisis en dos partes:

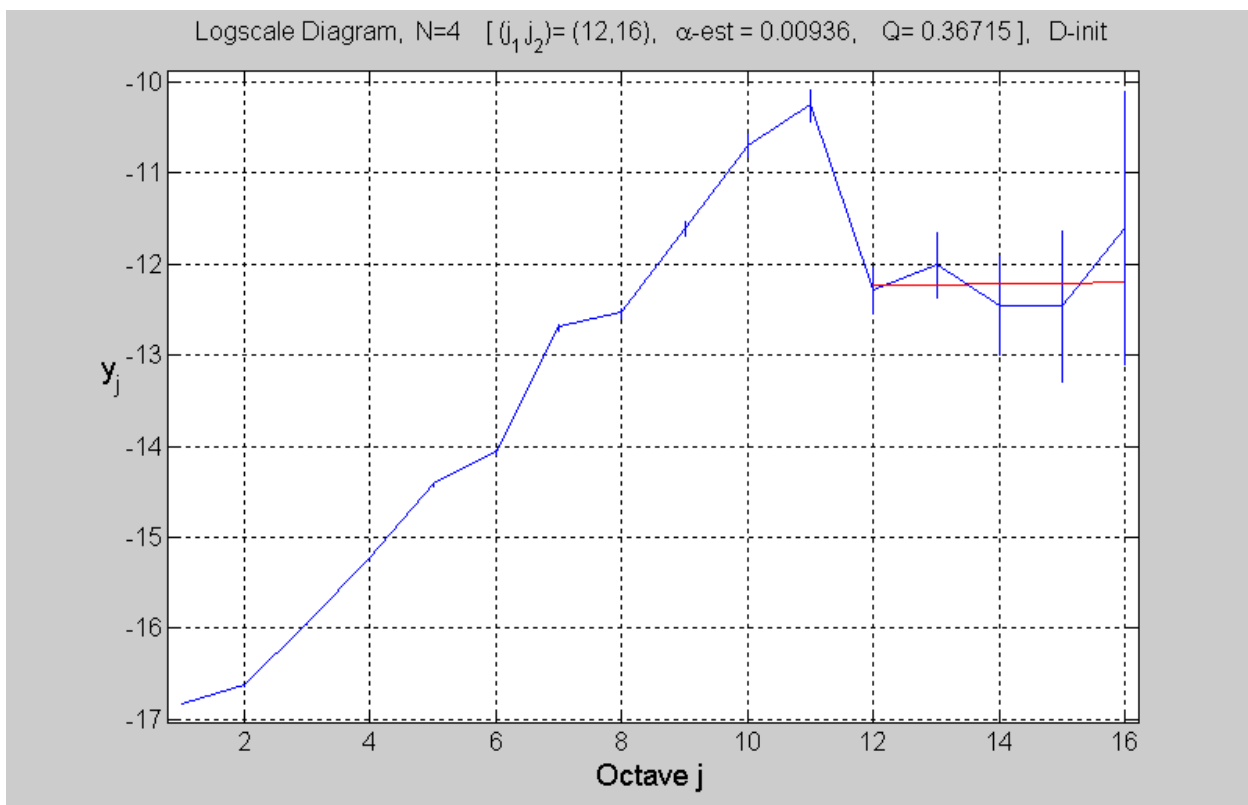
1. Se estudia la primer zona de alineación que va desde la octava 2 hasta la 11
2. Se estudia la segunda zona de alineación que va desde la octava 12 hasta la 16

Tomamos la primer zona de alineación que va de la octava 2 hasta la 11; nos arroja una estimación de $H = 0,856$, esto significa también LRD en como muestra la siguiente figura:

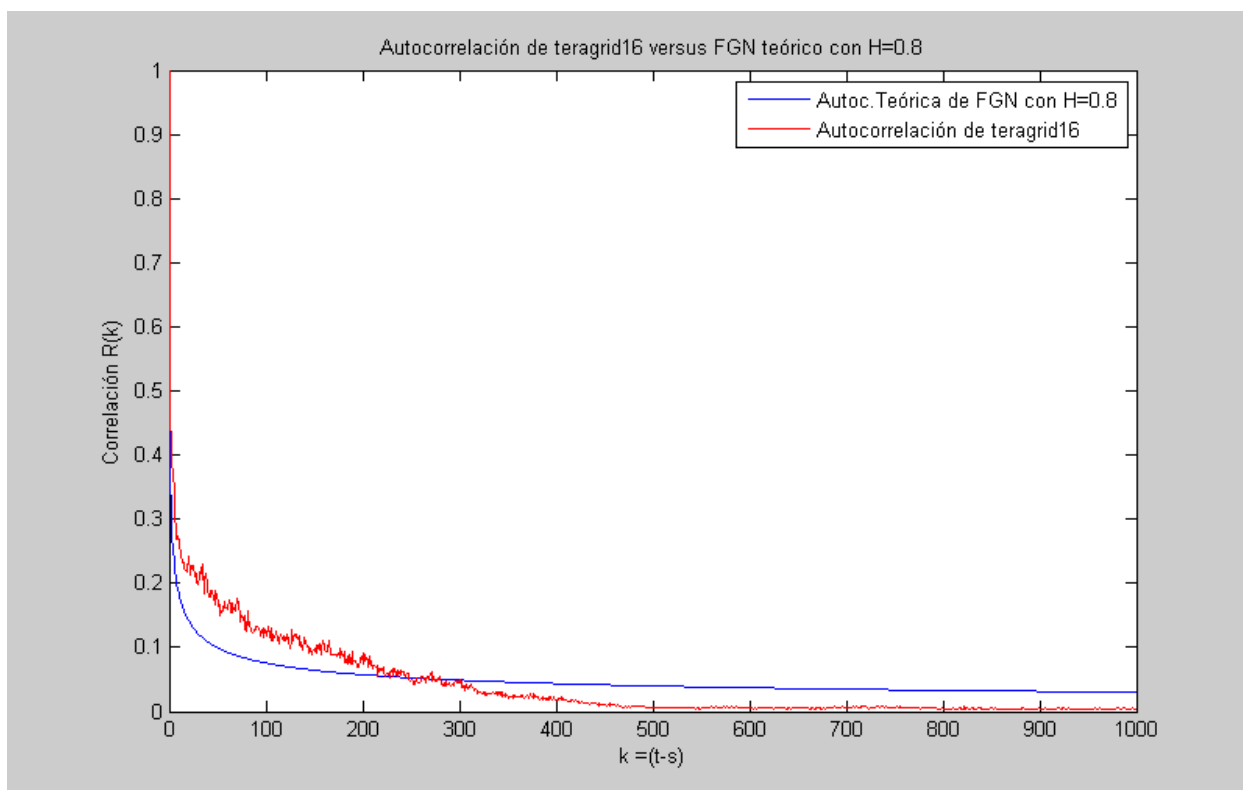
⁴⁶ <ftp://pma.nlanr.net/traces/long/tera/1/20040208-160000-1.gz>



Ahora, tomamos la segunda zona de alineación que desde la octava 12 hasta la 16 nos arroja una estimación de $H = 0,505$, o sea un proceso incorrelado o SRD.



Ahora bien si utilizamos la script `correlacionFBM.m` obtenemos la siguiente gráfica:



Se observa claramente una divergencia en el origen, que manifiesta la presencia de LRD; pero luego, a partir de $k \sim 300$ vemos que rápidamente converge a cero, resultando compatible con un comportamiento SRD. Esto es consistente con la grafica de LDestimate y los datos estimados arriba; también confirma que el proceso no presenta el mismo comportamiento en todas sus partes, evidenciando el fenómeno bi-escala que se comentaba anteriormente. Nuevamente en este caso se puede apreciar la utilidad de la script `correlacionFBM.m` para añadir información adicional a un análisis, y clarificar el comportamiento del proceso a los efectos de caracterizar en forma correcta una determinada muestra de tráfico.

Otro aporte que ofrecen estos resultados, es que, en general se puede afirmar que el FGN, se revela como un buen “candidato” para modelar monofractales, brindando una excelente aproximación para todos los procesos gaussianos o cuasi-gaussianos.

6.6 Concluyendo la discusión

La discusión realizada a lo largo de este capítulo, que discurre sobre la utilización y los detalles de las aplicaciones, muestra como el uso de las mismas, representa un aporte al estudio, análisis y síntesis de los fenómenos de escala en el tráfico de las redes telemáticas.

Queda planteada también la circunstancia, de que la experimentación con las aplicaciones sobre diversos escenarios, puede conducir a distintas formas de utilizarlas, e inclusive como consecuencia de eso, puedan surgir otros usos de las mismas, aun en modos no discutidos en este capítulo, además las scripts o aplicaciones, presentan la suficiente flexibilidad para ser adaptadas y modificadas en función de cualquier necesidad puntual o escenario particular, como también, eventualmente puedan servir de base al desarrollo de nuevas aplicaciones.

7 Conclusiones

Recordando el punto [1.4](#) del capítulo 1, al declarar los objetivos de esta tesis se manifestaba textualmente:

“Este trabajo, esta orientado al estudio del trafico autosimilar en las redes telemáticas, fundamentado en la revisión crítica, de diversos trabajos científicos realizados en este campo hasta el presente, y al desarrollo de aplicaciones practicas basadas en las wavelets u onditas utilizando el software MATLAB®. El objetivo de este trabajo se formaliza a través de dos ejes:

1. Brindar un marco o estructura didáctica que permita el desarrollo del tema como una unidad de estudio y aprendizaje.
2. El desarrollo de scripts o aplicaciones, para abordar el estudio de distintas características y/o propiedades en el contexto de la autosimilitud, y de esta manera, permitir el análisis, detección de comportamiento, estimación de parámetros, y la producción de trafico sintético, de manera que puedan utilizarse en el ámbito de la simulación.

Se considera este trabajo como un aporte curricular razonablemente autocontenido, con miras a facilitar el estudio de la temática considerada, y servir de base a otros estudios que contemplen la autosimilitud en el contexto del tráfico en las redes telemáticas.”

En los siguientes puntos, se expone como se consideran alcanzados los objetivos mencionados, mediante un repaso de los aportes producidos por el trabajo.

7.1 Aporte Como Marco o Estructura Didáctica

Con fundamento en la lectura de trabajos relacionados al tema de esta tesis, que datan aproximadamente desde 1990 hasta la fecha; y con los cuales se elaboró una selección que figura en las referencias bibliográficas de esta tesis, se exponen los temas y conceptos involucrados en el estudio del trafico autosimilar, la temática es abordada desde al análisis de los fenómenos de escala en los procesos. En particular, se hace el estudio a través del valor que toma el exponente de autosimilitud H , o exponente de Hurst, mediante distintas técnicas de análisis de sus propiedades estadísticas, y también mediante el uso de la simulación.

Partiendo del capítulo 1, donde se especifica el contexto del trabajo, el capítulo 2, expone el estado del arte del estudio del tráfico autosimilar. En el capítulo 3 donde se expone el marco teórico, los conceptos y definiciones, siguen una secuencia lógica, que comienza por la definición de tráfico como un proceso estocástico, y a medida avanza el desarrollo del tema, se van definiendo las propiedades estadísticas involucradas conforme va surgiendo la necesidad de utilizarlas, hasta llegar al tratamiento de los procesos que cumplen con las condiciones de autosimilitud, y se utilizan como modelos de tráfico, completando de esta manera el marco teórico para el abordar el estudio del tráfico autosimilar.

La elección del análisis basado en las wavelets u onditas como herramienta de aplicación, se fundamenta en que de por si, proveen de una completa estructura teórica que se ajusta en forma natural al comportamiento autosimilar del tráfico. En particular la transformada wavelet discreta, que a través del análisis multiresolución, comparte una afinidad estructural con los fenómenos de escala observados en el tráfico, esta característica convierte al uso de wavelets u onditas, en una herramienta ideal para las tareas de detección, estimación y síntesis de comportamiento autosimilar. El capítulo 4 aborda el estudio de las wavelets, desde la óptica del análisis de señales mediante el concepto de “transformadas”, y siguiendo la misma metodología utilizada en el capítulo 3, avanza en forma progresiva desde la transformada de Fourier, pasando por la transformada de Gabor hasta la transformada Wavelet y el Análisis Multiresolución, brindando las principales definiciones y los desarrollos teóricos para facilitar la comprensión del análisis wavelet.

En el capítulo 5, se expone el contenido y desarrollo de las aplicaciones, y en el capítulo 6, se discute la utilización de las mismas, para el estudio, análisis, y síntesis de procesos autosimilares.

Se asume que con el aporte de estos capítulos, se cumple con lo enunciado como primera parte o eje de los objetivos

7.2 Aporte de las Aplicaciones

Con relación al aporte de las aplicaciones desarrolladas en esta tesis, son una respuesta al desafío mencionado en el capítulo 1 donde se menciona:

1. El desarrollo de modelos generales que abarquen las principales características del tráfico a estudiar.
2. El desarrollo de aplicaciones con los modelos para obtener conclusiones válidas.

El contexto de esta tesis es el del segundo punto; y se plantea la simulación como el paradigma de estudio mas viable; motivo por el cual se desarrollaron las aplicaciones para MATLAB®, de las cuales algunas son de mi autoría y otras no, en cuyo caso se nombran sus respectivos autores, debidamente citados y documentados. El fundamento, utilización y validez de las aplicaciones, quedaron debidamente documentados en los capítulos 5 y 6, contando con la discusión de los resultados obtenidos, y demostrando al mismo tiempo, la utilidad y posibilidad de experimentación de las mismas.

En particular para el caso de las aplicaciones o scripts de mi autoria, desarrolladas como parte de esta tesis, sirven para realizar y/o complementar estudios, análisis, y síntesis de procesos autosimilares o muestras de tráfico (trazas), y tienen como ventaja de que hacen uso de los recursos de MATLAB®, sin necesitar librerías o scripts de terceros, facilitando el estudio a quienes no puedan conseguir o disponer de las fuentes citadas para las scripts pertenecientes a los otros autores, las cuales como se comentó en el capítulo 5, necesitan de material auxiliar que no tiene incluido MATLAB®.

7.2.1 Aplicaciones desarrolladas

Como aporte original de este trabajo se elaboraron las aplicaciones que muestra la siguiente tabla:

Script	Utilidad
estudioGabor.m	Estudio del análisis por transformada de Gabor
estudioWav.m	Análisis de una señal de audio mediante transformada de Gabor
cargaVectores.m	Carga de datos y generación de procesos sintéticos equivalentes
serieTescalada.m	Graficador de comportamiento
comparaGrafica.m	Graficador de comportamiento para dos procesos
rangoRescalado.m	Estimación de H , por el método gráfico de rango reescalado (Efecto Hurst)
rangoRescaladocomparado.m	Idem anterior para dos procesos
rangoRescaladopromedio.m	Estimación de H , por el método gráfico de rango reescalado (promediando los bloques)
varianzaTiempo.m	Estimación de H , por el método gráfico de varianza versus tiempo
varianzaTiempoComparado	Idem anterior para dos procesos
varianzaW.m	Estimación de H , por método de regresión de Varianza versus octavas (wavelets)
FGNIFourier.m	Generación de secuencias autosimilares sintéticas con exponente H determinado
correlacionFBM.m	Estudio gráfico del comportamiento de la autocorrelación de dos procesos.

Además, como se expuso en el capítulo 5 se trabajaron y analizaron un conjunto de scripts correspondientes a distintos autores, conforme se muestra en la siguiente tabla:

Scripts	Autor	Utilidad
LDestimate_code	Darryl Veitch	Conjunto de Scripts para estimación de parámetros
Uvi_Wave300	Universidad de Vigo	Tool Box con un conjunto de utilidades wavelet para Matlab
FGNDW.m	Jeong H-D	Síntesis de FGN con un determinado H

Se asume que con el aporte de estas aplicaciones, se cumple con lo enunciado como segunda parte o eje de los objetivos

7.3 La visión del estudio del tráfico autosimilar desde la óptica de este trabajo

En el capítulo 2, que refiere al estado del arte dentro del campo de estudio de tráfico autosimilar, se alude a que si bien en un principio se tendió a pensar que la presencia y ubicuidad de la autosimilitud, tornaría inutilizable el andamiaje de soluciones analíticas aportada por la teoría de colas, el paso del tiempo y el estudio más profundo demostró que esto no es así, de hecho muchos trabajos ponen en evidencia que tanto la detección espuria de LRD y/o autosimilitud, como el efecto o impacto que realmente causan en el desempeño de las redes, relativizan esta cuestión. Para corroborar esta afirmación, se pueden consultar las referencias bibliográficas [13], [29], [41], [42], y [51]; donde se exponen de manera crítica los alcances de ciertos estudios acerca de la autosimilitud y su contexto, es por eso precisamente que trabajos como el efectuado en esta tesis, pretenden facilitar el estudio y la comprensión del fenómeno, colaborando en la tarea de detectar y cuantificar la presencia y efectos del mismo.

En realidad, el panorama se plantea de una manera, donde no todo es autosimilar, ni claramente “Poissonian”, evidentemente se observan comportamientos compatibles en uno y otro sentido. En este punto adquiere gran importancia el conocimiento del contexto donde se hace el estudio, para poder justipreciar los resultados obtenidos de los análisis experimentales; y con este escenario se evidencia la utilidad de contar con aplicaciones como las desarrolladas.

En general y como resultado de la experimentación con diversas trazas, aparecen como válidas las afirmaciones adelantadas en el punto 2.4, corolario del capítulo 2; donde se mencionan evidencias de comportamiento con tendencias SRD, en trazas tomadas de grandes backbones con elevados niveles de multiplexación, y como comportamiento contrapuesto las trazas tomadas en redes LAN, son compatibles con LRD, verificando como válidos los análisis y resultados descriptos en las referencias bibliográficas [37] y [38].

En los procesos abordados a lo largo de este trabajo, se trata a la LRD como sinónimo de autosimilitud, que como se aclaró oportunamente, son cosas distintas, además, también existen distintos tipos de autosimilitud, pero lo que relaciona y agrupa en la misma categoría a todos estos conceptos, es lo siguiente.

En aquellos procesos que presentan LRD, o sea son persistentes en su comportamiento, aún en importantes separaciones temporales, y que en cierta manera al cambiar la escala en la que se miden esos tiempos u holguras temporales, vemos que se preserva la estructura de esta persistencia, decimos que el proceso presenta un comportamiento autosimilar, o que estamos ante un fenómeno de escala; la representación de estos procesos son curvas fractales donde la dimensión fractal está relacionada con el factor de escala del proceso, si analizamos la dimensión fractal en el dominio de las frecuencias, vemos que se relaciona con la pendiente logarítmica de la densidad de potencia del proceso, es decir que al graficar el espectro de potencia en un gráfico logarítmico los puntos se agruparán en torno a una línea recta cuya pendiente es la dimensión fractal. Este comportamiento particular se denomina monofractalidad (la pendiente es constante), y esta circunstancia es la que origina el nombre de “ruido $1/f$ ”, a los procesos LRD o Autosimilares siempre y cuando la pendiente de la recta mencionada se encuentre en el intervalo abierto $(0,1)$.

En este trabajo todos los análisis, desarrollos, y conclusiones siempre son expuestos desde la óptica de procesos con exponente de escala constantes en el tiempo, es decir no se aborda el estudio de procesos multifractales. En el capítulo 6, en la parte final del punto [6.2](#), se expuso que los procesos multifractales son una generalización de los procesos monofractales (H constante), donde el comportamiento ante el cambio de escala presenta una estructura mucho mas compleja (H como función del tiempo).

No obstante lo expuesto, en muchas de las trazas estudiadas en este trabajo, se demostró como válida la naturaleza monofractal como así también la gaussianidad de los procesos que representan, quedando plenamente justificada la utilidad de disponer de aplicaciones como las desarrolladas en esta tesis, para corroborar o contrastar resultados.

Como ejemplo de validez de la discusión en esta tesis como así también su ámbito de aplicación, se muestran trabajos recientes mencionados en las referencias bibliográficas como:

- [48] Xianhai Tan; Yuanhui Huang; Weidong Jin, Modeling and Performance Analysis of Self-Similar Traffic Based on FBM, Network and Parallel Computing Workshops, 2007. NPC Workshops. IFIP International Conference 18-21 Sept. 2007 Page(s):543 – 548.
- [49] William Rea, Les Oxley, Marco Reale y Jennifer Brown, Estimators for Long Range Dependence: An Empirical Study Electronic Journal of Statistics. Vol. 0 (2009) ISSN: 1935-7524.
- [50] Ginno Millán, Gastón Lefranc, Presentation of an Estimator for the Hurst Parameter for a Self-Similar Process Representing the Traffic in IEEE 802.3 Networks. Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844 Vol. IV (2009), No. 2, pp. 137-147.
- [51] Clegg Richard G.; Landa Raul; Rio Miguel, Criticisms of modelling packet traffic using long-range dependence. Presented at the PMECT Oct 2009 workshop (part of ICCCN 2009) in San Francisco.

Concluyendo, se considera alcanzado en forma global el objetivo de la tesis, al contar por una parte, con un marco teórico suficientemente autocontenido, que posibilita abordar el estudio y comprensión del tema; y por otra parte las aplicaciones desarrolladas, las cuales complementan el estudio y permiten las tareas de experimentación, análisis, estimación de parámetros y síntesis de procesos autosimilares.

7.4 Algunas Líneas de Trabajos Futuros

- Debido a que los modelos multifractales permiten capturar los efectos multiplicativos que introducen las pilas de protocolos en los intercambios en una red, son muy estudiados en la actualidad y también en ese contexto son particularmente útiles las wavelet, y por lo tanto este trabajo también puede ser punto de partida para estudiar esos procesos.
- También para quienes estén orientados al desarrollo de software, es posible la creación de aplicaciones integradas que arrojen como producto un “Frame Work” para trabajo con muestras de tráfico o inclusive un entorno de simulación que abarque el análisis y síntesis de procesos autosimilares.
- Debido a su gran poder de calculo, el MATLAB® es un excelente candidato para desarrollar un “Front End”, por aquellos que estén familiarizados con el desarrollo de interfaces con este software.

7.5 Unas últimas palabras

Estimado Lector, si el tiempo que gentilmente dedicó a la lectura de este trabajo, le significó en alguna medida, un aporte para, comprender, estudiar, o experimentar los temas tratados; el esfuerzo y empeño que puse en la elaboración del mismo, quedarán plenamente justificados.

R. Scappini

8 Referencias Bibliográficas

- [1] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, [On the self-similar nature of ethernet traffic \(extended version\)](#), IEEE/ACM Transactions on Networking, vol.2, pp.1–15, Feb. 1994.
- [2] P. Prithi and A. Erramilli. [Heavy-tailed on/off source behavior and self-similar traffic](#). In: Proc. ICC '95, Seattle, WA (1995) pp. 445- 450W.-
- [3] C. Lau, A. Erramilli, J. L. Wang, and W. Willinger, [Self-similar traffic generation: The random midpoint displacement algorithm and its properties](#), Proceedings of the IEEE International Conference on Communications, Seattle, WA, June 1995, vol. 1, pp.466–472.
- [4] Vern Paxson and Sally Floyd, [Wide-Area Traffic: The Failure of Poisson Modeling](#) IEEE/ACM Transactions on Networking, Vol. 3 No. 3, pp. 226-244, June 1995.
- [5] Ilka Norros, “[On the use of fractional brownian motion in the theory of connectionless networks](#)” [J], IEEE J. Select. Areas Commun, vol. 13, no. 6, Aug. 1995: 953–962.
- [6] Mark E. Crovella and Azer Bestavros, [Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes](#) in IEEE/ACM Transactions on Networking, 5(6):835--846, December 1997. (An earlier version of this work appeared under the BU technical report [TR-95-015](#) Boston University Computer Science Department, Revised, October 12, 1995).
- [7] B. K. Ryu and A. Elwalid, [The Importance of Long-range Dependence of VBR Video Traffic in ATM Traffic Engineering: Myths and realities](#) ACM Computer Communication Review, vol. 26, pp. 3-14, Oct. 1996.
- [8] Kihong Park, Gi Tae Kim, and Mark E. Crovella, [On the Relationship Between File Sizes, Transport Protocols, and Self-Similar Network Traffic](#). In Proceedings of the International Conference on Network Protocols, pages 171-180, October, 1996. (Also available as [Technical Report BU-CS-96-016](#), Boston University, Computer Science Dept., July, 1996).
- [9] Walter Willinger, Murad Taqqu, and Ashok Erramilli, [A Bibliographical Guide to Self-Similar Traffic and Performance Modeling for Modern High-Speed Networks](#) Stochastic Networks: Theory and Applications, Royal Statistical Society Lecture Notes Series, Vol. 4, Oxford University Press, 1996.
- [10] V. Paxson. [Fast, Approximate Synthesis of Fractional Gaussian Noise for Generating Self-Similar Network Traffic](#). ACM SIGCOMM, Computer Communication Review, 27(5):5–18, 1997.
- [11] Patrice Abry and Darryl Veitch, [Wavelet Analysis of Long-Range-Dependent Traffic](#), IEEE Transactions On Information Theory, Vol. 44, No. 1, January 1998.
- [12] Walter Willinger, Vern Paxson, and Murad Taqqu, [Self-similarity and Heavy Tails: Structural Modeling of Network Traffic](#). In “A Practical Guide to Heavy Tails: Statistical Techniques and Applications”, Adler, R., Feldman, R., and Taqqu, M.S., editors, Birkhauser, 1998.
- [13] Ronn Ritke, Xiaoyan Hong, Mario Gerla, [Contradictory Relationship between Hurst Parameter and Queueing Performance](#). UCLA Computer Science Department Hilgard Ave. Los Angeles CA.IEEE/ACM Transactions on Networking Vol 159 pp.467--478 1998.
- [14] Feldmann, A. C. Gilbert, W. Willinger and T. G. Kurtz, [The Changing Nature of Network Traffic: Scaling Phenomena](#), ACM Computer Communication Review, vol. 28, pp. 5-29, Apr. 1998.
- [15] Veitch, D.; Abry, P. [A wavelet-based joint estimator of the parameters of long-range dependence](#) Information Theory, IEEE Transactions on Volume 45, Issue 3, Apr 1999 Page(s):878 – 897.

- [16] Stéphane Mallat “A Wavelet Tour of Signal Processing”, ISBN : 0-12-466606-X Academic Press, 1999 - <http://www.cmap.polytechnique.fr/~mallat/book.html>.
- [17] Jeong, H.-D.J., McNickle, D., and Pawlikowski, K. (1999), [Fast Self-Similar Teletraffic Generation Based on FGN and Wavelets](#), In Proceedings of IEEE International Conference on Networks (ICON'99), Brisbane, Australia.
- [18] Kihong Park, Walter Willinger Self-Similar Network Traffic and Performance Evaluation Copyright 2000 John Wiley & Sons, Inc. ISBNs: 0-471-31974-0 (Hardback); 0-471-20644-X (Electronic).
- [19] Jim W. Roberts, France Telecom R&D ["Traffic Theory and the Internet"](#), IEEE Communications Magazine • January 2001.
- [20] Hei Xiaojun, [The Self-Similar Traffic Modeling in the Internet](#) Department of Electronic & Computer Engineering Hong Kong University of Science and Technology [Enlínea] URL: <http://www.ee.ust.hk/~heixj/publication/comp660f/comp660f.html> February 2001.
- [21] Mark E. Crovella, [Performance Evaluation with Heavy Tailed Distributions](#). In: Lecture Notes in Computer Science 1786. pp. 1--9. Slightly revised version appeared in Job Scheduling Strategies for Parallel Processing: 7th International Workshop, pp. 1-10, June 16, 2001.
- [22] Quan-Lin Li, [Heavy-tailed distributions and Communication Networks](#), at Carleton University, Canada. (2001).
- [23] M. Alzate y N. Peña, Modelos de Tráfico en Análisis y Control de Redes de Comunicaciones. 2004 Universidad Distrital Francisco José de Caldas.
- [24] M. Alzate y A. Monroy, [Introducción al Tráfico Autosimilar en Redes de Comunicaciones](#). Revista Ingeniería Vol. 6 No. 2 2001 Páginas: 6 –17. Universidad Distrital Francisco José de Caldas.
- [25] Pawlikowski Krzysztof, Jeong Hae–Duck Joshua, and Lee Jong–Suk Ruth, [On Credibility of Simulation Studies of Telecommunication Networks](#). University of Canterbury. IEEE Communications Magazine, pp. 132-139 January, 2002.
- [26] M. Alzate y A. Monroy, Uso de la transformada wavelet para el estudio de tráfico fractal en redes de comunicaciones. Revista Ingeniería Vol. 7 No. 1 Junio 2002 Páginas: 11 –24. Universidad Distrital Francisco José de Caldas.
- [27] Xavier Hesselbach Serra, Jordi Altés Bosch – “Análisis de redes y sistemas de comunicaciones”, Edicions UPC de la Universitat Politècnica de Catalunya, 2002 SL Jordi Girona Salgado 31, 08034 Barcelona - Depósito legal: B-41612-2002 ISBN: 84-8301-611-7.
- [28] Jeong, H.-D.J, [Modelling Of Self-Similar Teletraffic For Simulation](#), PhD Thesis, University of Canterbury July 2002.
- [29] Thomas Karagiannis, Michalis Faloutsos, Rudolf H. Riedi, [Long-Range Dependence: Now you see it, now you don't!](#) In IEEE GLOBECOM, Global Internet Symposium 2002
- [30] P. Abry, P. Flandrin, M. S. Taqqu, and D. Veitch. [Self-similarity and longrange dependence through the wavelet lens](#). In P. Doukhan, G. Oppenheim, and M. S. Taqqu, editors, Theory and Applications of Long-Range Dependence, pages 526–556. Birkh user, 2003.
- [31] Barakat, C., Thiran, P., Iannaccone, G., Diot, C., and Owezarski, P.. [Modeling Internet backbone traffic at the flow level](#). IEEE Transactions on Signal Processing, 51, 2111-2124. (2003).
- [32] Kenneth Falconer [“FRACTAL GEOMETRY Mathematical Foundations and Applications Second Edition”](#), University of St Andrews, UK ISBN: 0-470-84862-6 Wiley-VCH Paperback, 366 p. September 2003.
- [33] Dana Mackenzie: "Wavelets: ver el bosque y los  rboles"; con la colaboraci3n de los Dres. Ingrid Daubechies, Daniel Kleppner, St phane Mallat, Yves Meyer, Mary Beth Ruskai y Guido Weiss para Beyond Discovery®: The Path from Research to Human Benefit [M s all  del descubrimiento: El camino desde la investigaci3n hasta el

- beneficio humano], un proyecto de la National Academy of Sciences (Academia Nacional de las Ciencias). Se puede acceder en:
http://www7.nationalacademies.org/spanishbeyonddiscovery/mat_008276-04.html.
- [34] H. D. J. Jeong, D. McNickle, and K. Pawlikowski, Generation of Self-Similar Time Series for Simulation Studies of Telecommunication Networks, Proceedings of the 1st Western Pacific/3rd Australia-Japan Workshop on Stochastic Models, September 1999.
- [35] Clegg, R. G., The Statistics of Dynamic Networks PhD Thesis. Dept of Maths. University of York. June 2004.
- [36] Stallings Williams, Redes e Internet de Alta Velocidad. Rendimiento y Calidad De Servicio Segunda edición. PEARSON EDUCACIÓN. S.A., Madrid, 2004 ISBN: 84-205-3921-X.
- [37] Thomas Karagiannis, Mart Molle, and Michalis Faloutsos University of California Long-Range Dependence Ten Years of Internet Traffic Modeling, IEEE INTERNET COMPUTING 1089-7801/04/\$20.00 © 2004 IEEE Published by the IEEE Computer Society SEPTEMBER • OCTOBER 2004 .
- [38] ALARCON-AQUINO, V., GUERRERO-OJEDA, L.G., RODRIGUEZ-ASOMOZA, J. et al. Análisis de Tráfico Auto-similar en Redes de Comunicaciones Usando Onditas (Wavelets). Inf. tecnol., 2005, vol.16, no.2, p.61-66. ISSN 0718-0764.
- [39] Xiaofeng Bai, A. Shami, Modeling Self-Similar Traffic for Network Simulation, Technical report, NetRep-2005-01, April 2005.
- [40] Allen B. Downey, Lognormal and Pareto distributions in the Internet. In Computer Communications, Volume 28, Issue 7, 2 May 2005, Pages 790-801.
- [41] Carla Di Cairano-Gilfedder, Richard G. Clegg, A decade of internet research: advances in models and practices, BT Technology Journal, Volume 23, Number 4, October 2005, pp. 115-128(14).
- [42] Gennady Samorodnitsky “Long memory and self-similar processes” Annales de la faculté des sciences de Toulouse Sér. 6, 15 no. 1 (2006), p. 107-123 [Article PDF](#) | Analyses MR 2225749.
- [43] Clegg, R. G., A Practical Guide to Measuring the Hurst Parameter (Extended version), International Journal of Simulation: Systems, Science & Technology 7(1) (2006) 3-14
- [44] Thomas M. Chen, Network Traffic Modeling, Chapter in The Handbook of Computer Networks, Hossein Bidgoli (ed.), Wiley, to appear 2007.
- [45] G. Horn, Amund Kvalbein, J. Blomskøld, E. Nilsen: An empirical comparison of generators for self similar simulated traffic. Perform. Eval. 64(2): 162-190 (2007).
- [46] P. Abry y D. Veitch, Matlab Code for the estimation of MultiScaling Exponents [Enlínea] URL, http://www.cubinlab.ee.mu.oz.au/~darryl/MS_code.html Mon April 16 15:33:41 2007.
- [47] Oleg I. Sheluhin Moscow State Technical University of Service (MSTUS), Russia Sergey M. Smolskiy Moscow Power Engineering Institute (MPEI), Russia Andrey V. Osin Moscow State Technical University of Service (MSTUS), Russia SELF-SIMILAR PROCESSES IN TELECOMMUNICATIONS ISBN 978-0-470-01486-8 (HB) John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England England Hardcover 334 pages April 2007.
- [48] Xianhai Tan; Yuanhui Huang; Weidong Jin, Modeling and Performance Analysis of Self-Similar Traffic Based on FBM, Network and Parallel Computing Workshops, 2007. NPC Workshops. IFIP International Conference 18-21 Sept. 2007 Page(s):543 – 548.
- [49] William Rea, Les Oxley, Marco Reale y Jennifer Brown, Estimators for Long Range Dependence: An Empirical Study Electronic Journal of Statistics. Vol. 0 (2009) ISSN: 1935-7524.
- [50] Ginno Millán, Gastón Lefranc, Presentation of an Estimator for the Hurst Parameter for a Self-Similar Process Representing the Traffic in IEEE 802.3 Networks. Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844 Vol. IV (2009), No. 2, pp. 137-147.

[51] Clegg Richard G.; Landa Raul; Rio Miguel, [Criticisms of modelling packet traffic using long-range dependence](#). Presented at the PMECT Oct 2009 workshop (part of ICCCN 2009) in San Francisco.

9 Apéndice 1 - Código Fuente de las Aplicaciones

9.1 Código Fuente de las Scripts para MATLAB®

A continuación se expone el código fuente de las scripts desarrolladas en esta tesis, al final de este apéndice se encuentran los links de donde se pueden obtener todas las scripts de terceros que se mencionan en el trabajo.

El código que se encuentra dentro de los recuadros, se puede copiar y pegar directamente en el editor de scripts del MATLAB®, a los efectos de generar los archivos de scripts correspondientes.

9.1.1 Script estudioGabor.m

```
% Esta script calcula el espectro de la señal mediante FFT y además realiza
% un grafica de tiempo-frecuencia mediante un análisis de Gabor,
% permitiendo modificar los parámetros de ancho de ventana y cantidad de
% muestras del espectro
senal=(' señal ');
ts1=0:1/128:1; ts2=1:1/128:2; %entre 0 y 1 vale t1 y entre 1 y 2 vale t2 con
ts=[ts1 ts2]; % vector tiempo (eje x)
f1=2*sin(2*pi*32*ts1); f2=sin(2*pi*8*ts2); % f1 y f2 son las dos partes de la
% señal
f=[f1 f2]; %vector señal
fs=input(' Frec. de muestreo? ');
T=1/fs; % Intervalo de tiempo entre muestras
N=length(f); % Número o cantidad de muestras de la señal
t=0:T:(N-1)*T; % eje de tiempos
espectro=abs(fft(f));
ejeF=0:1/N:.5-1/N; % eje de frecuencias para dibujar espectro de la señal
W=input(' parametro W alfa=1/W de la ventana de gabor ');
alfa=1/W; %aquí introducir
% el parámetro W para calcular alfa a los efectos de probar con distintos
valores y
% observar el gráfico resultante (se hace de esta forma pues así se usan
% numeros grandes sin comas)
n=input(' Ingrese desplazamiento como % de la extensión total de la señal ');
V=n*N/100*T; % Desplazamiento de la ventana Del ta t
g1=(1/(2*sqrt(pi*alfa))*exp(-(t-V).^2/(4*alfa))); % Función ventana para
% dibujarla en el grafico
% de tiempo
G=[]; % se define un vector vacío para poder usarlo en la grafica
% a continuación se define un lazo donde se calcula el producto de la señal
% f por la ventana g, y se desliza la ventana controlado por el valor de j
% me=input(' Cantidad de muestras del espectro? '); % Permite modificar
% la cantidad de muestras que se graficarán
ct=100/n; % la cantidad de desplazamientos está dada por ct que a su vez esta
% relacionada con el ancho de la señal (ver línea 18)
for j=1:ct % significa que se haran ct traslaciones de la ventana de ancho V
g=(1/(2*sqrt(pi*alfa))*exp(-(t-j)*V).^2/(4*alfa)));
a=f.*g; d=abs(fft(a)); G=[G;d(1:floor(length(d)/2))]; % length(d)/2 por el
teorema de muestreo
end
figure % Muestra la señal y la ventana en el dominio temporal
plot(t, f, 'b', t, g1, 'r')
title([' \ti Señal de estudio s(t)= ',senal, ' y ventana g(t) con \alpha=
',num2str(alfa), ' y n= ',num2str(ct), ' desplazamientos con \Del ta t=
',num2str(V), ' seg. ']); xlabel(' \ti Tiempo'); ylabel(' \ts(t)')
figure % Muestra el espectro de la señal estudiada
plot(ejeF, espectro(1:floor(N/2)), 'r')
title([' \ti Espectro de Frecuencias para s(t)= ',senal ])
figure % Muestra el grafico en el plano frecuencia - tiempo
surf(G)
title([' \ti Análisis de Gabor para s(t)= ',senal, ' y ventana g(t) con \alpha=
',num2str(alfa), ' y ',num2str(ct), ' desplazamientos con \Del ta t=
',num2str(V), ' seg. ']);
xlabel(' \ti Frecuencia'); ylabel(' \Del ta t'); zlabel(' \ti Amplitud')
```

9.1.2 Script estudioWav.m

```

% A partir de un archivo de audio en formato wav
% Esta script calcula el espectro de la señal mediante FFT y además realiza
% un grafico de tiempo-frecuencia mediante un análisis de Gabor,
% permitiendo modificar los parámetros de ancho de ventana y cantidad de
% muestras del espectro
senal=input('Ingrese señal para analizar ','s');
st=40*(wavread(senal));
fs=44100; % frecuencia de muestreo archivo .wav
T=1/fs; % Intervalo de tiempo entre muestras
f=st(:,1);f=f'; % Carga canal izquierdo del archivo .wav estereo como vector fila,
% si se quiere analizar el canal derecho se debe utilizar st(:,2)
N=length(f);% Número o cantidad de muestras de la señal
t=0:T:(N-1)*T;% eje de tiempos
espectro=abs(fft(f));
ejef=0:1/N:.5-1/N; % eje de frecuencias para dibujar espectro de la señal
W=input(' parametro W alfa=1/W de la ventana de gabor '); alfa=1/W; %aquí introducir
% el parámetro W para calcular alfa a los efectos de probar con distintos valores y
% observar el gráfico resultante (se hace de esta forma pues así se usan
% numeros grandes sin comas)
n=input('Ingrese desplazamiento como % de la extensión total de la señal ');
V=n*N/100*T;
g1=(1/(2*sqrt(pi*alfa))*exp(-(t-V).^2/(4*alfa))); % Función ventana para
% dibujarla en el grafico
% de tiempo
G=[]; % se define un vector vacío para poder usarlo en la grafica
% a continuación se define un lazo donde se calcula el producto de la señal
% f por la ventana g, y se desliza la ventana controlado por el valor de j
ct=100/n; % la cantidad de desplazamientos está dada por ct que a su vez está
% relacionada con el ancho de la señal (ver líneas 19 y 20)
for j=1:ct % significa que se harán ct traslaciones de la ventana de ancho V
g=(1/(2*sqrt(pi*alfa))*exp(-(t-j*V).^2/(4*alfa)));
a=f.*g; d=abs(fft(a)); G=[G;d(1:floor(length(d)/2))];
end
figure % Muestra la señal y la ventana en el dominio temporal
plot(t,f,'b',t,g1,'r')
title(['\tSeñal de estudio s(t)= ',senal,' y ventana g(t) con \alpha= ',num2str(alfa),' y
n= ',num2str(ct),' desplazamientos con \Delta t=
',num2str(V),' seg.']); xlabel ('\tTiempo'); ylabel ('\ts(t)')
figure % Muestra el espectro de la señal estudiada
plot(ejef,espectro(1:floor(N/2)),'r')
title(['\tEspectro de Frecuencias para s(t)= ',senal])
figure % Muestra el grafico en el plano frecuencia - tiempo
surf(G)
title(['\tAnálisis de Gabor para s(t)= ',senal,' y ventana g(t) con \alpha=
',num2str(alfa),' y ',num2str(ct),' desplazamientos con \Delta t= ',num2str(V),' seg.'])
xlabel ('\tFrecuencia'); ylabel ('\tDel t'); zlabel ('\tAmplitud')

```

9.1.3 Script cargaVectores.m

```

% El vector debe estar en formato ASCII conteniendo los timestamps en la
% primera columna y la longitud en bytes de la trama ethernet en la
% segunda columna, de esta forma carga en "vec" los datos correspondientes
% a la longitud de la trama, en "vet" los timestamp correspondientes y en
% "tiea", los tiempos entre arribos.
dato=input('Nombre archivo de datos? ','s');
%Aquí podemos poner el nombre de cualquier archivo que
% contenga los datos que queremos estudiar
datos=load(strread(dato,'%c'));
vec=datos(:,2); % Vector longitud en bytes de los de los paquetes ethernet
vet=datos(:,1); % Vector conteniendo los timestamp de los paquetes
k=length(vec); % Se carga en k la cantidad de valores o longitud del
% vector datos
tiea=(vet(2:k)-vet(1:k-1)); % vector tiempo entre arribos del vector datos
% A continuación generamos un proceso de Poisson con los datos del vector
% bajo estudio
lambda=poissfit(vec); % Con esto generamos el parámetro de intensidad del
proceso
% sintético lo llamamos lambda por regresión MLE
% de los datos del vector considerando las llegadas en
% bytes por segundo como una distribución de Poisson
xpois=poissrnd(lambda,1,k); % Aquí generamos el vector xpois con k datos y dist.
% de Poisson con parámetro lambda, este vector es
% el equivalente Poisson teórico del vector datos
tieap=zeros(1,k); % Inicializa un vector para los tiempos entre arribos del
% proceso Poisson
tieap(1)=exprnd(1/lambda); % genera el primer tiempo para el vector tiempo
% teórico (Poisson)
% El lazo que sigue genera el vector con tiempos distribuidos exponencialmente
;ki=1;
while ki <=k-1
    tieap(ki+1)=tieap(ki)+exprnd(1/lambda); %Cálculo del vector de tiempos
%con distribución exponencial
    ki=ki+1;
end
k=length(tieap);
tieap=(tieap(2:k)-tieap(1:k-1)); %Calcula el tiempo entre arribos del
%vector tiempos y lo carga en tieap.

```

9.1.4 Script serieTescalada.m

```

% Esta script grafica los primeros 90 elementos de series de
% tiempo a partir de un vector de datos escalando el eje temporal como
% potencias de un factor de escala "fe", permitiendo ver el "aspecto" o comportamiento
% de la serie a distintas escalas temporales. Conviene que "fe" sea 10 de forma
% tal que si se elige por ejemplo un vector que tiene espaciado los datos
% en órden de milisegundos (0.001 seg.) el siguiente grafico se hace con un
% eje del orden de las centésimas de seg. y el siguiente con décimas de
% segundo, pues el factor original es de 10, y el siguiente será de
%  $10^2=100$  y el siguiente 1000 y así hasta terminar los graficos, el
% exponente esta gobernado por la variable "graf" del bucle el número de
% gráficos está gobernado por la variable "cs", indicada como cantidad de
% series, y por último la variable "cp" gobierna la cantidad de elementos
% del vector a graficar por cada serie m-escalada que calcula la script.
% Las restantes variables están comentadas dentro de la script
clear xf fe cs m d i s xm c1 cp % Con esto se limpian las variables a utilizar, esto es
util en caso de ejecutar la script varias veces
nombre=input('Nombre del vector a estudiar? ','s'); % a los efectos de colocar el nombre
xf=eval(nombre); % del vector estudiado en el grafico
d=length(xf); % Guarda la cantidad de valores que tiene el vector dato
ut=input('Unidad de tiempo (en seg.)? '); % Fija la unidad de tiempos en seg de los datos
del vector estudiado
fe=input('valor de "m" para escalar la serie? '); % Factor para escalar el tiempo
cs=input('Cantidad de Series? '); % Especifica cuantas series se van a graficar
cp=90; %La cantidad de puntos (eje x) que se graficaran por cada escala de tiempo
csmax=floor(log(d/cp)/log(fe)); % Calcula el máximo N° de series que se pueden graficar en
función de los datos disponibles
% El siguiente lazo es para verificar que la cantidad de series a dibujar
% no exeda el límite de los datos disponibles
if (csmax < cs)
    prompt = ['Con los datos disponibles la cant. de series debe ser menor o igual a
', num2str(csmax)];
    dlg_title = 'Atención!!!';
    num_lines = 1;
    def = {' '};
    p = inputdlg(prompt,dlg_title,num_lines,def);
    cs = str2num(char(p));
end
m=fe; % Se fija el valor inicial de "m" para utilizarlo en la 1° serie m-escalada
cut=ut; % Esta variable se utiliza en la leyenda del eje x para poner la unidad de tiempo
que corresponde a cada serie
subplot(cs,1,1); bar(xf(1:cp)); title(['Vector estudiado:
', nombre]); axis('tight'); xlabel(['Unidad de Tiempo = ', num2str(cut), ' seg.']) % Dibuja la
primer serie correspondiente al vector de datos
for graf=2:cs % Inicia el cálculo de las serie para la nueva escala de tiempo
    cut=ut*m; % Fija la unidad de tiempo para ponerla en la leyenda del eje x
    i=1; % las variables i y s controlan la posición de los datos para calcular la nueva
serie
    s=m;
    xm=zeros(i, floor(d/m)); % Inicializa el vector que representa la serie para esta
escala "m"
    for c=1: floor(d/m)
        xm(c)=sum(xf(i:s)); % Calcula los valores de la serie en esta escala
        i=i+m;
        s=s+m;
    end
    subplot(cs,1,graf); h=bar(xm(1:cp)); axis('tight'); xlabel(['Unidad de Tiempo =
', num2str(cut), ' seg.']) % Grafica la serie calculada para esta escala temporal
m=fe^graf; % se indexa el factor "m" estableciendo una nueva escala de tiempo
end

```

9.1.5 Script comparaGrafica.m

```

% Esta script grafica y compara el comportamiento de dos vectores ante el
% escalamiento temporal por un factor "m"
clear xd1 fe cs m d i s xm xd2 c1 c2 cp % se limpian las variables a utilizar
datos1=input('Nombre del vector datos 1 ? ','s');xd1=eval(datos1);
datos2=input('Nombre del vector datos 2 ? ','s');xd2=eval(datos2);
% Con el siguiente bucle me aseguro que la longitud de datos para calcular
% los puntos del gráfico no exceda la condición fijada en la línea 20
if length(xd1) <= length(xd2)
    d=length(xd1);
else
    d=length(xd2);
end
% Guarda la cantidad de elementos que tiene el vector de menor longitud a
% los efectos de correr sin errores de dimension la rutina de calculo
% de las series "m" escaladas que se inicia con el lazo de la línea 40
ut=input('Unidad de tiempo (en seg.)? '); % Fija la unidad de tiempos en
%seg de los datos del vector estudiado
fe=input('Valor de "m" para escalar la serie agregada ');
cs=input('Cantidad de Series? '); % esta variable especifica cuantas series
% m-escaladas se van a graficar
cp=90; %La cantidad de puntos (eje x) que se graficaran por cada escala de tiempo
csmax=floor(log(d/cp)/log(fe));% Calcula el máximo N° de series que se pueden graficar
% en función de los datos disponibles
% El siguiente lazo es para verificar que la cantidad de series a graficar
% no exeda el límite en función de los datos disponibles
if (csmax < cs)
    prompt = ['Con los datos disponibles la cant. de series debe ser menor o igual a
'], num2str(csmax)];
    dlg_title = 'Atención!!!';
    num_lines = 1;
    def = {'';
    p = inputdlg(prompt,dlg_title,num_lines,def);
    cs = str2num(char(p));
end
m=fe; % Se fija el valor inicial de "m" para utilizarlo en la 1° serie m-escalada
cut=ut; % esta variable se utiliza en la leyenda del eje x para poner la unidad
% de tiempo que corresponde a cada serie
c1=1;c2=2; % Estas variables son para controlar la posición de los graficos en la ventana
subplot(cs,2,1);bar(xd1(1:cp));title(['Vector ',datos1]);axis('tight');xlabel(['Unidad de
Tiempo = ',num2str(cut),' seg.']) % Dibuja la primer serie m-agregada correspondiente al
vector de datos
subplot(cs,2,2);bar(xd2(1:cp));title(['Vector ',datos2]);axis('tight');xlabel(['Unidad de
Tiempo = ',num2str(cut),' seg.'])
for graf=2:cs % Se inicia el cálculo de las series m-escaladas
    cut=ut*m; % Fija la unidad de tiempo para ponerla en la leyenda del eje x
    c1=c1+2; % c1 y c2 se utilizan para controlar la posición de grafico en la pantalla
    c2=c2+2;
    i=1; % las variables i y s controlan los datos de la serie m-escalada
    s=m; % en el siguiente bucle
    xm=zeros(i,floor(d/m)); % se inicializan los vectores a calcular
    xpm=zeros(i,floor(d/m));
    for c=1:floor(d/m)
        xm(c)=sum(xd1(i:s)); % se calculan cada uno de los valores de la serie m-escalada
        xpm(c)=sum(xd2(i:s)); % Idem anterior,pero con el 2° vector
        i=i+m;
        s=s+m;
    end
% A continuación se grafican las series calculadas para la escala "m"
    subplot(cs,2,c1);bar(xm(1:cp));axis('tight');xlabel(['Unidad de Tiempo =
'],num2str(cut),' seg.'])
    subplot(cs,2,c2);bar(xpm(1:cp));axis('tight');xlabel(['Unidad de Tiempo =
'],num2str(cut),' seg.'])
    m=fe^graf; % se indexa el factor "m" escalando la base de tiempo
end

```

9.1.6 Script rangoRescalado.m

```
% Script para calcular el valor del parámetro de Hurst, por el método del
% Rango Reescalado R/S, también llamado estadístico R/S
% load DST
clear Datos RVec d mmax m prom Vd f S AVd f R RS M
Datos=input('Nombre del vector con los datos? ','s');
RVec =eval(Datos);
% cr=input('Cantidad de Rangos a evaluar? ');
d = length(RVec);% Cantidad de elementos en el vector con los datos
m=d;graf=1;
while m > 8 % se considera 8 como la mínima cantidad de elementos para
    % calcular un rango
    Vd f=zeros(m,1);
    AVd f=zeros(m,1);
    prom=mean(RVec(1:m));% Calcula el promedio de bloque
    Vd f=RVec(1:m)-prom;% Calcula el apartamiento puntual de la media
    S=std(Vd f);% Calcula la desviación estándar del bloque
    AVd f=cumsum(Vd f);% Calcula las diferencias acumuladas
    R=(max(AVd f)-min(AVd f));% Calcula el rango
    if (R<=0)
        break
    else
        RS(graf)=log(R/S);% calcula el cociente R/S para el bloque
        M(graf)=log(m); % guarda el tamaño de la muestra m
        end
        m=floor(m/2);% Diminuye a la mitad la cantidad de datos por bloque
        graf=graf+1;% Incremento el índice del punto gráfico calculado
end
p=polyfit(M,RS,1);
plot(M,RS,'r',[0 M],[p(2) p(2)+M*p(1)],'-b');axis equal
title(['Gráfico Rango Reescalado para el vector ',Datos]);xlabel('X=log
(m)');ylabel('Y=log(R/S)')
key(1)={(['Tomando ',num2str(graf),' rangos del vector ',Datos])};
key(2)={(['se estima H = ',num2str(p(1)),'; valor obtenido por'])};
key(3)={(['regresión (cuadrados mínimos)'])};
text(M(graf-1)-2,RS(floor(graf/5)),key)
```

9.1.7 Script rangoRescaladocomparado.m

```
% Script para calcular el valor del parámetro de Hurst, por el método del
% Rango Reescalado R/S, también llamado estadístico R/S
clear Datos RVec d mmax m prom Vd f S AVd f R RS M key
Datos1=input('Nombre del 1er. vector con los datos? ','s');
RS1vec =eval(Datos1);
Datos2=input('Nombre del 2do. vector con los datos? ','s');
RS2vec=eval(Datos2);
% De acuerdo a la cantidad de elementos en los vectores con los datos, se
% calcula la dimensión de los rangos eligiendo el menor
if length(RS1vec) <= length(RS2vec)
    d=length(RS1vec);
else
    d=length(RS2vec);
end
m=d;graf=1;
while m > 8 % se considera 8 como la mínima cantidad de elementos para
    % calcular un rango
    Vd f1=zeros(m,1);Vd f2=zeros(m,1);
    AVd f1=zeros(m,1);AVd f2=zeros(m,1);
    prom1=mean(RS1vec(1:m));% Calcula el promedio de bloque para el vector 1
    prom2=mean(RS2vec(1:m));% Calcula el promedio de bloque para el vector 2
    Vd f1=RS1vec(1:m)-prom1;% Calcula el apartamiento puntual de la media
    Vd f2=RS2vec(1:m)-prom2;% Calcula el apartamiento puntual de la media
    S1=std(Vd f1);S2=std(Vd f2);% Calcula la desviación estándar de ambos bloques
    AVd f1=cumsum(Vd f1);% Calcula las diferencias acumuladas
    AVd f2=cumsum(Vd f2);
    R1=(max(AVd f1)-min(AVd f1));% Calcula el rango
    R2=(max(AVd f2)-min(AVd f2));
    if (R1<=0)||(R2<=0)% Con esto se evita la indeterminación causada por el
        break % logaritmo de cero
    else
        RS1(graf)=log(R1/S1);% calcula el cociente R/S para el bloque
        M1(graf)=log(m); % guarda el tamaño de la muestra m
        RS2(graf)=log(R2/S2);% calcula el cociente R/S para el bloque
        M2(graf)=log(m); % guarda el tamaño de la muestra m
        end
        m=floor(m/2);% Diminuye a la mitad la cantidad de datos por bloque
        graf=graf+1;% Incremento el índice del punto gráfico calculado
end
p=polyfit(M1,RS1,1);q=polyfit(M2,RS2,1);
plot(M1,RS1,'r',[0 M1],[p(2) p(2)+M1*p(1)],'-b',M2,RS2,'r',[0 M2],[q(2) q(2)+M2*q(1)],'-
g');axis equal
title(['Gráfico Rango Reescalado para los vectores ',Datos1,' y ',Datos2]);xlabel('X=log
(m)');ylabel('Y=log(R/S)')
key(1)={(['Tomando ',num2str(graf-1),' rangos de los vectores'])};
key(2)={(['se estima H',Datos1,' = ',num2str(p(1)),'; y H',Datos2,' = ',num2str(q(1))])};
key(3)={(['calculados por regresión (cuadrados mínimos)'])};
text(M1(graf-1)-2,RS1(floor(graf/5)),key)
```


9.1.8 Script rangoRescaladopromedio.m

```

% Script para calcular el valor del parámetro de Hurst, por el método del
% Rango Reescalado R/S, también llamado estadístico R/S pero promediando el
% valor obtenido en cada bloque y considerando todos los bloques, es decir
% siempre calcula tomando la totalidad de los datos
clear Datos RSvec d m bloque m prom Vdif AVdif key
Datos=input('Nombre del vector con los datos? ','s');
RSvec =eval (Datos);
d = length(RSvec);% Cantidad de elementos en el vector con los datos
m=d;graf=1; % Condiciones iniciales se toma la cantidad total de elementos
while m > 8 % se considera 8 como la minima cantidad de elemntos para
    % calcular un rango
    cb=floor(d/m);% Cantidad de bloques a iterar por el lazo
    SB=zeros(cb,1);% Se inializa el vector para las std de los bloques
contiguos
    RB=zeros(cb,1);% Se inicializa el vector para los rangos de bloques
contiguos
    RSB=zeros(cb,1);% Se iniciliza el vector para la razon R/S para cada bloque
contiguos
    Vdif=zeros(m,1);% Se inicializa el vector de las diferencias puntuales
    AVdif=zeros(m,1);% Se inicializa el vector para las diferencias acumuladas
    i=1;s=m; % Variables del control de bucle
    for bloque=1:cb
        prom=mean(RSvec(i:s));% Calcula el promedio de bloque
        Vdif=RSvec(i:s)-prom; % Calcula el apartamiento puntual de la media
        SB(bloque)=std(Vdif);% Calcula la desviación standar del bloque
        AVdif=cumsum(Vdif);% Calcula las diferencias acumuladas
        RB(bloque)=(max(AVdif)-min(AVdif));% Calcula el rango
        if RB(bloque)<=0 % con esta condición se evita calcular log de cero
            % deteniendo la ejecución en este punto.
            break
        else
            RSB(bloque)=(RB(bloque)/SB(bloque));% calcula el cociente R/S para
el bloque
            i=i+m;% Se incrementan las variables del control de puntero para
            s=s+m; % pasar al siguiente bloque de datos de tamaño m
        end
    end
    RS(graf)=log(mean(RSB));% calcula la ordenada para el graficolog(R/S)
    % promediando todos los bloques de tamaño m
    M(graf)=log(m); % guarda el tamaño de la muestra para la absisa del grafico
    m=floor(m/2);% Diminuyo a la mitad la cantidad de datos por bloque
    graf=graf+1; % Incremento el indice del punto grafico calculado
end
p=pol yfit(M, RS, 1);
plot(M, RS, 'r', [0 M], [p(2) p(2)+M*p(1)], '-b');axis equal
title(['Gráfico Rango Reescalado para el vector ',Datos]);xlabel ('X=log
(m)');ylabel ('Y=log(R/S)')
key(1)={(['Tomando ',num2str(graf-1),' rangos promediados del vector ',Datos])};
key(2)={(['se estima H = ',num2str(p(1)),' valor obtenido por'])};
key(3)={(['regresión (cuadrados mínimos)'])};
text(M(graf-1)-2, RS(floor(graf/5)),key)

```

9.1.9 Script varianzaTiempo.m

```

% Esta script calcula el parametro de Hurst para un vector de datos por el
% método de Varianza/Tiempo y lo grafica en pantalla
clear xf d fe m mg X Y i s p H key% se limpian del area de trabajo las
% variables utilizadas en esta script por si se utiliza mas de una vez
datos1=input('Nombre del vector que contiene los datos ? ','s');xf=eval(datos1);
d=length(xf);
fe=input('"m" nivel de agregación '); % corresponde a "m" o escala de
fe=floor(fe); % agregación de las series a calcular
if (fe<=1) % este lazo es para verificar que m sea >= 2 de lo contrario la
% script entra en loop infinito, actua poniendo una advertencia
% en la pantalla
prompt = ('"m" debe ser un entero mayor o igual a 2');
dlg_title = 'Atención!!!';
num_lines = 1;
def = {' '};
dia = inputdlg(prompt,dlg_title,num_lines,def);
fe = floor(str2num(char(dia)));
end
cs=(floor(log(d)/log(fe))); % Esto asegura que la cantidad de series
% calculada no sobrepase el máximo permitido
% por la long. del vector dato
X(1)=0; % calculo del primer punto con todos los valores del vector datos
Y(1)=log(var(xf));
m=fe;
mg(1)=m; % Guarda el primer valor de "m"
for graf=2:cs
i=1; % i y s son variables indices de control de posición del bucle
s=m; % para promediar los m valores de la nueva serie
xm=zeros(i,floor(d/m)); % inicializa el vector "xm" de la serie "m"
for c=1:floor(d/m) % calculo del vector "xm" la serie m-agregada
xm(c)=sum(xf(i:s))/m;
i=i+m; % se indexan los valores de los índices para formar bloques
s=s+m; % no adyacentes
end
if (m>=d);break;end
Y(graf)=log(var(xm)); % calculo del 1º punto de gráfico, correspondiente al
X(graf)=log(m); % vector de la serie m-agregada
m=fe^graf; % calculo del nuevo valor de "m" para el siguiente vector "xm"
mg(graf)=m; % Guarda el nuevo valor de "m"
end
p=polyfit(X,Y,1); % Ajuste lineal de los datos calculados me da los
% coeficientes p(1) y P(2) que son la pendiente y
% ordenada al origen de la recta ajustada
H=1-abs(p(1))/2; % Calcula el valor del parametro de Hurst a partir de
% la pendiente p(1) de la recta ajustada
% A continuación, la rutina que grafica los valores calculados
plot(X,Y,'.r',[0 X],[p(2) p(2)+X*p(1)],'-b');axis equal % Dibuja el grafico de
los datos y su correspondiente regresión
title(['Gráfico Varianza Tiempo para el vector ',datos1]);xlabel('log
(m)');ylabel('Y=log Varianza X(m)')
text(X(4)+.2,Y(4),['\rightrightarrows pendiente \beta = ',num2str(p(1))],'FontSize',8)
key(1)={(['Tomando ',num2str(d),' elementos de ',datos1,' y "m" inicial =
',num2str(fe),]);}
key(2)={(['se estima H = ',num2str(H),' para \beta = ',num2str(p(1)),' obtenido
por ']);}
key(3)={(['regresión (cuadrados mínimos) de ',num2str(cs),' puntos calculados
por la script.']);}
text(X(1)+.1,Y(cs)+1,key)

```

9.1.10 Script varianzaTiempocomparado.m

```

% Esta script calcula el parametro de Hurst para dos vectores de datos
% por el método de Varianza/Tiempo y pone los resultados en pantalla
clear xd1 xd2 d fe m mg XD1 XD2 YD1 YD2 i s P1 P2 H1 H2 % se limpian del area de trabajo
las
% variables utilizadas en esta script por si se utiliza mas de una vez
datos1=input('Nombre del vector datos 1 ? ','s');xd1=eval(datos1);
datos2=input('Nombre del vector datos 2 ? ','s');xd2=eval(datos2);
% Con el siguiente bucle me aseguro que la longitud de datos para calcular
% los puntos del gráfico no exceda la condición fijada en la línea 22
if length(xd1) <= length(xd2)
    d=length(xd1);
else
    d=length(xd2);
end
fe=input(' "m" nivel de agregación '); % corresponde a "m" o nivel de
% agregación de la serie resultante
fe=floor(fe);
if (fe<=1) % este lazo es para verificar que m sea >= 2 de lo contrario
% la script entra en loop infinito
prompt = (' "m" debe ser un entero mayor o igual a 2');
dlg_title = 'Atención!!!';
num_lines = 1;
def = {' '};
dia = inputdlg(prompt,dlg_title,num_lines,def);
fe = floor(str2num(char(dia)));
end
cs=(floor(log(d)/log(fe))); % Esto asegura que la cantidad de series
% calculadas no sobrepase el máximo permitido
% por la long. del vector dato
% A continuación se calcula el primer punto con todos los valores de ambos
% vectores
XD1(1)=0;
YD1(1)=log(var(xd1));
XD2(1)=0;
YD2(1)=log(var(xd2));
m=fe;
mg(1)=m; % Guarda el primer valor de "m"
% Los bucles anidados que están a continuación generan los vectores de las
% series "m" agregadas
for graf=2:cs
    i=1; % i y s son indices de control de posición del bucle para
    s=m; % promediar los m valores de la nueva serie
    xm=zeros(1,floor(d/m)); % inicializa el vector "xm" de la serie "m"
    xp=zeros(1,floor(d/m)); % Idem anterior pero con xp
    for c=1:floor(d/m) % calculo de los puntos para la serie m-agregada
        xm(c)=sum(xd1(1:s))/m;
        xp(c)=sum(xd2(1:s))/m;
        i=i+m;
        s=s+m;
    end
    if (m>=d);break;end
    YD1(graf)=log(var(xm)); % calculo de los puntos de gráfico correspondientes
    YD2(graf)=log(var(xp)); % a los vectores de la series m-agregada
    XD1(graf)=log(m);
    XD2(graf)=log(m);
    m=fe^graf; % calculo del nuevo valor de "m" para el siguiente vector "xm"
    mg(graf)=m; % Guarda el nuevo valor de "m"
end
P1=polyfit(XD1,YD1,1); % Ajuste lineal de los datos calculados me da los
P2=polyfit(XD2,YD2,1); % coeficientes, que son la pendiente y
% ordenada al origen de la recta ajustada
H1=1-abs(P1(1))/2; % Calcula el valor del parametro de Hurst a partir
H2=1-abs(P2(1))/2; % de la pendiente de la recta ajustada
% A partir de aquí se dibuja el gráfico de los datos y su correspondiente regresión
plot(XD1,YD1,'r',[0 XD1],[P1(2) P1(2)+XD1*P1(1)],'-b',XD2,YD2,'r',[0 XD2],[P2(2)
P2(2)+XD2*P2(1)],'-g');axis equal
title(['Gráfico Varianza Tiempo para ',datos1,' y ',datos2]);xlabel('log
(m)');ylabel('Y=log Varianza X(m)')
text(XD1(4)+.2,YD1(4),['\rightrightarrows pendiente \beta',datos1,' = ',num2str(P1(1))]);
text(XD2(4)+.2,YD2(4),['\rightrightarrows pendiente \beta',datos2,' = ',num2str(P2(1))]);
key(3)={'Tomando ',num2str(d),' elementos de cada vector, con "m" inicial =
',num2str(fe),' y '};
key(1)={'Se estima H',datos1,' = ',num2str(H1),' para \beta = ',num2str(P1(1))};
key(2)={'Se estima H',datos2,' = ',num2str(H2),' para \beta = ',num2str(P2(1))};
key(4)={'regresión (cuadrados mínimos) de ',num2str(cs),' puntos calculados por la
script.'};
menor1=min(YD1);menor2=min(YD2);
if menor1<=menor2
    text(0+.1,menor1+1.5,key,'FontSize',8)
else
    text(0+.1,menor2+1.5,key,'FontSize',8)
end
end

```

9.1.11 Script varianzaW.m

```
% Esta script calcula el valor de H por el metodo de Varianza / escala,
% calculando la varianza de los coeficientes del vector detalles a
% diferentes octavas y haciendo una regresión lineal de forma de conocer la
% pendiente de la recta

datos=input('Nombre del vector que contiene los datos? ','s');y=eval(datos);
lo=length(y);
% cc=input('Cantidad de coeficientes en último nivel ');% se debe
% descomentar esta línea si se desea especificar el Numero de coeficientes
% del último nivel de detalles
niv=floor(log2(lo/4));% si se utiliza cc, reemplazar el 4 por cc
cd=zeros(1,niv);
oct=zeros(1,niv);
for i=1:niv
    [y,d]=dwt(y,'db10');
    cd(i)=log2(var(d));
    oct(i)=i;
end
p=polyfit(oct,cd,1);
if p(1)<1
    H=(p(1)+1)/2;
else
    H=(p(1)-1)/2;
end
plot(oct,cd,'r',[0 oct],[p(2) p(2)+oct*p(1)],'-b');axis equal % Dibuja el grafico de los
datos y su correspondiente regresión
title(['Gráfico Varianza [dx(j,k)] para el vector ',datos]);xlabel('Octavas
j');ylabel('Varianza [dx(j,k)]')
text(oct(4)+.2,cd(4),['\rightarrow pendiente \gamma = ',num2str(p(1))],'FontSize',8)
key(1)={'Tomando ',num2str(lo),' elementos de ',datos};
key(2)={'se estima H = ',num2str(H),' para \gamma = ',num2str(p(1)),' obtenido por'};
key(3)={'regresión (cuadrados mínimos) de ',num2str(niv),' puntos calculados por la
script. '};
text(oct(1)+.1,cd(niv)-1,key)
```

9.1.12 Script FGNIFourier.m

```
% Esta función calcula una secuencia de números complejos a partir del
% espectro de frecuencias de un Ruido Gaussiano Fraccional. La fase de los
% números complejos es aleatoria y uniformemente distribuida (gaussiana)
% de esta forma se preserva la densidad espectral
function FGNFourier=FGNIFourier(n,H)
% Se crean n frecuencias, y se calcula el espectro de potencia aproximado
lambda = ((1:n)*pi)/n;
cf = (1/(2*pi))*var(lambda)*sin(pi*H). *gamma(2*H+1);
FGNS = cf*(abs(lambda)).^(1-2*H);
% Ajuste de la estimación via periodograma
r = random('Exponential',1,1,n);
f_r = FGNS.*r;
% Contrucción de la serie compleja con fase aleatoria.
re = sqrt(f_r);
im = random('Uniform',0,2*pi,1,n);
% Calculo de la parte real e imaginaria de la densidad espectral
real_part = re.*cos(im);
imag_part = re.*sin(im);
zFGN = real_part + imag_part*i;
FGNFourier=real(fft(zFGN));
```

9.1.13 Script correlaciónFBM.m

```
%Esta script grafica el comportamiento de la correlación de un proceso de
% FGN teórico y la de un vector que representa un proceso FGN sintético o
% tomado de trazas reales
datos=input('Vector de datos? ','s'); datos=eval(datos);
lags=input('Ingrese el valor de k (lags) ');
R=zeros(1,lags);
H=input('Ingrese el valor de H (teórico)');
for i=1:lags+1
    R(i)=.5*(((i+1)^(2*H))+((i-1)^(2*H))-(2*i^(2*H)));
end
[acf,nlags,bound]=autocorr(datos,lags,2);
% plot(R,'-r'),title(['H=',num2str(H)]);xlabel('k =(t-s)');ylabel('Correlación
R(k)');axis([0 Inf 0 Inf])
plot(nlags,R,'-b',nlags,acf,'-r'),title(['Autocorrelación de ',datos,' versus FGN teórico
con H=',num2str(H)]);xlabel('k =(t-s)');ylabel('Correlación R(k)');
legend(['Autoc. Teórica de FGN con H=',num2str(H)], ['Autocorrelación de ',datos]);axis([0
Inf 0 Inf])
```

9.1.14 LDestimate.m

Para acceder al conjunto de scripts para la estimación de parámetros de escala desarrolladas por Darryl Veitch, que pertenecen a un paquete llamado [second order estimation code](#), se sugiere visitar la siguiente página:

<http://www.cubinlab.ee.unimelb.edu.au/~darryl/index.html>

En la misma se puede ubicar el enlace [Download](#) donde se encuentran las aplicaciones desarrolladas por él. Es muy instructivo leer todas las explicaciones y comentarios que hace el autor sobre las aplicaciones, de los cuales algunos han sido transcriptos en este trabajo.

9.1.15 FGNDW.m

Para obtener el código fuente de esta script y sus accesorias se debe acceder al trabajo [26] Jeong, H.-D.J, [Modelling Of Self-Similar Teletraffic For Simulation](#), PhD Thesis, University of Canterbury July 2002, y el código se encuentra disponible en el Apéndice D, en particular punto D.2 en la página 248. El link para conseguir el documento es:

http://www.cosc.canterbury.ac.nz/research/reports/PhdTheses/2003/phd_0302.pdf

Una importante aclaración; para que este código funcione es necesaria la utilización de un toolbox llamado Uvi_Wave.300, que fue desarrollado en la Universidad de Vigo, y que se puede conseguir en el enlace mencionado en el siguiente punto. Se recomienda también leer la aclaración que hice en el capítulo 5, punto [5.3.1.1](#) Utilizando FGNDW.m, pues si se utilizan así como están las scripts, el MATLAB® pondrá mensajes de error en la pantalla.

9.1.16 Uvi_Wave.300

Uvi_Wave.300, es un completo paquete de utilidades (scripts) para trabajar con wavelets y el código escrito para MATLAB®, fue desarrollado por la Universidad de Vigo; originalmente el paquete estaba en el siguiente link:

http://www.tsc.uvigo.es/~wavelets/uvi_wave.html.

Aparentemente ese link ya no está disponible pero el paquete completo se puede conseguir en Internet en el siguiente link:

http://www.control.auc.dk/~alc/Uvi_Wave300.zip

Es conveniente bajarlo y descomprimirlo en un directorio y también resulta muy instructiva la lectura de los comentarios que poseen todas las scripts disponibles (que son muchas y no todas se utilizan, dado que este es un completo paquete de utilidades para trabajar con wavelets, y aquí solamente se utilizan aquellas que la script FGNDW.m, requiere como subrutinas).

10 Apéndice 2 – Procesos Estocásticos

10.1 Definición de Proceso Estocástico

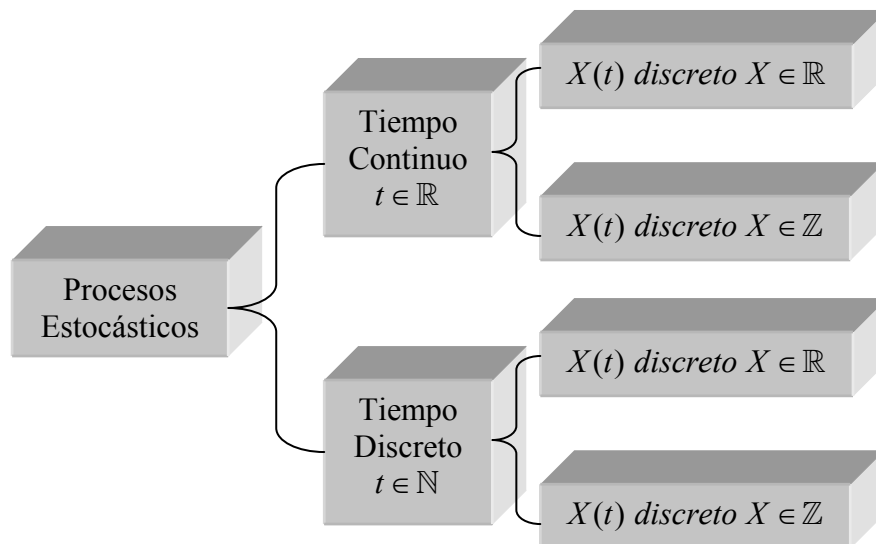
Un proceso estocástico es un conjunto de variables aleatorias $\{X(t)\}, t \in \Gamma$ definido en un espacio de probabilidades indexado por un parámetro t , este parámetro es generalmente el tiempo y toma valores del conjunto temporal Γ , que puede ser continuo o discreto.

Los procesos estocásticos pueden ser continuos o discretos en el tiempo (t), en el primer caso t varía en forma continua, y en el segundo, t adopta valores discretos, generalmente enteros positivos aunque en algunos casos se pueda considerar el rango de enteros entre $-\infty$ y $+\infty$.

Conforme la definición de variable aleatoria, un proceso estocástico se puede interpretar de varias formas:

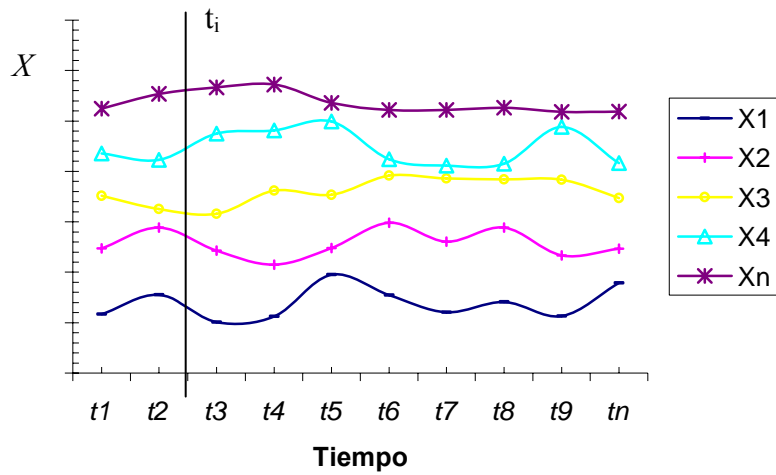
1. Una familia de funciones del tiempo, t variable, todos los resultados posibles.
2. Una única función temporal, t variable, un resultado.
3. Una variable aleatoria, t fijo, todos los resultados posibles.
4. Un único número, t fijo, un resultado.

Un proceso estocástico puede ser continuo o discreto en el tiempo, además la variable aleatoria puede tomar valores continuos o discretos como se resume en la siguiente figura:



10.2 Ergodicidad

Como se vio mas arriba un proceso estocástico podía tener distintas interpretaciones dependiendo del contexto, y lo podemos ilustrar viéndolo como distintas realizaciones de un proceso aleatorio en el siguiente grafico:



- Como variable aleatoria tiene su media, varianza y demás propiedades derivadas de su distribución. Si se calculan todos los valores posibles para todos los tiempos, es decir para todo su espacio de eventos se lo llama conjunto de valores posibles, entonces si calculamos la media de todo el espacio, será una media de conjunto.
- Si consideramos un cierto tiempo t_i constante, vemos que la variable X puede tomar todos los valores posibles a lo largo de la recta t_i y su intersección con X_1, X_2, X_3 etc.
- Ahora bien, si tomamos una sola realización de la variable X_i y consideramos variable a t (una única función del tiempo), podemos considerar su valor medio a lo largo del tiempo, siendo esta una media o promedio temporal de esa realización.

Se dice que un proceso estocástico es ergódico, si la media de su conjunto es igual a la media temporal de una sola realización, esto es muy importante, dado que nunca se dispone del conjunto total de las realizaciones de la variable, es mas en la mayoría de los casos solamente disponemos de una sola serie, entonces la suposición de ergodicidad nos permite la estimación de parámetros, analizando una única función temporal a lo largo de un período de tiempo.

11 Apéndice 3 – Ingeniería de Tráfico

11.1 Definición de Ingeniería de Tráfico

Citando el manual de la UIT (Union Internacional de Telecomunicaciones)⁴⁷, llamado “Sobre la Ingeniería de Tráfico”⁴⁸, el cual fue elaborado por la Comisión de Estudio 2/16 del UIT-D y Congresos Internacionales de Teletráfico donde dice:

“La teoría de teletráfico se define como la aplicación de la teoría de las probabilidades a la solución de problemas concernientes a la planificación, evaluación de la calidad de funcionamiento, operación y mantenimiento de sistemas de telecomunicación. En forma más general, la teoría de teletráfico se puede considerar como una disciplina de planificación en la que los medios (procesos estocásticos, teoría de puesta en fila y simulación numérica) se toman de la investigación de las operaciones.

El término teletráfico abarca todo tipo de tráfico de comunicación de datos y de tráfico de telecomunicaciones. La teoría estará primordialmente ilustrada con ejemplos de sistemas de comunicación telefónica y datos. Sin embargo, los medios formulados son independientes de la tecnología y aplicables en otras áreas como tráfico de caminos, tráfico aéreo, cintas de fabricación y montaje, distribución, gestión de talleres y almacenamiento, y toda clase de sistemas de servicio.

El objetivo de la teoría del teletráfico puede formularse así:

Lograr calcular el tráfico en unidades bien definidas mediante modelos matemáticos y determinar la relación existente entre calidad de servicio y capacidad del sistema, de tal manera que la teoría se convierta en una herramienta útil para la planificación de las inversiones.”

El Sector UIT-T está dividido en Comisiones de Estudio. La Comisión de Estudio 2 (CE2) se ocupa de los aspectos operacionales de la prestación de servicios, redes y calidad de funcionamiento. Cada Comisión de Estudio se divide en Grupos de Trabajo. El Grupo de Trabajo 3 de la Comisión de Estudio 2 (GT 3/2) es responsable de la ingeniería de tráfico. En las Recomendaciones producidas por el Grupo de Trabajo 3/2. Están comprendidas en la Serie E (numeradas entre E.490 y E.799) y constituyen el cuerpo principal de las Recomendaciones del UIT-T sobre ingeniería de tráfico.

Estas Recomendaciones se pueden clasificar conforme a las cuatro tareas de ingeniería de tráfico principales:

- Caracterización de la demanda de tráfico.
- Objetivos de grado de servicio (gos).
- Controles y dimensionamiento de tráfico.
- Supervisión de calidad de funcionamiento.

11.2 Teoría de Colas

La teoría de colas es el estudio matemático de los fenómenos que se producen en las líneas de espera, también llamadas colas de espera o sencillamente colas. Existen diversos procesos

⁴⁷ <http://www.itu.int/net/home/index-es.aspx>

⁴⁸ En español en: <http://clusterfie.epn.edu.ec/ibernal/html/CURSOS/Oct06Marzo07/Trafico/Libros/handbook.pdf> y en inglés : http://clusterfie.epn.edu.ec/ibernal/html/CURSOS/Oct06Marzo07/Trafico/Libros/telenookpdf_2006.pdf

relacionados, la llegada o arribo al final de la cola, el tiempo de espera en la cola, y el tiempo de servicio ocupado por el servidor o proceso de servicio.

Puede decirse que el estudio de los fenómenos de espera o colas tuvieron su génesis con el trabajo del ingeniero danés Agner Krarup Erlang, que trabajó para la Copenhagen Telephone Exchange y se encargó del estudio del problema de dimensionamiento de líneas y centrales de conmutación telefónica para el servicio de llamadas., publicó el primer artículo sobre la teoría de colas en 1909. David G. Kendall introdujo una notación de colas A/B/C en 1953. La notación de Kendall para describir las colas y sus características puede encontrarse en Tijms, H.C, "Algorithmic Analysis of Queues", Capítulo 9 en A First Course in Stochastic Models, Wiley, Chichester, 2003. Ha sido desde entonces extendida a 1/2/3/(4/5/6) donde los números se reemplazan con:

1) Un código que describe el proceso de llegada. Los códigos usados son:

- M para "Markoviano", significando una distribución exponencial para los tiempos de servicio o tiempos entre llegadas.
- D para unos tiempos de servicio "deterministas".
- G para una "distribución general". (tiempos de espera o servicio con distribuciones no exponenciales como ser weibull, o pareto, etc.

2) Un código similar que representa el proceso de servicio. Se usan los mismos símbolos.

3) El número de canales de servicio (o servidores).

4) La capacidad del sistema, o el número máximo de clientes permitidos en el sistema incluyendo esos en servicio. Cuando el número está al máximo, las llegadas siguientes son rechazadas.

5) El orden de prioridad en la que los trabajos en la cola son servidos:

- First Come First Served (FCFS) ó First In First Out (FIFO).
- Last Come First Served (LCFS).
- Service In Random Order (SIRO).

6) El tamaño del origen de las llamadas. El tamaño de la población desde donde los clientes vienen.

Otro aporte importante en este campo viene de la mano de John D. C. Little con su trabajo titulado: A Proof of the Queuing Formula : $L=\lambda W$. Al cual se puede acceder en: <http://www.doc.ic.ac.uk/~uh/for-paul/A%20PROOF%20FOR%20THE%20QUEUEING%20FORMULA%20L=IW.%207689045.pdf>. Una serie de expresiones analíticas derivadas de este trabajo son muy utilizadas en el análisis de colas y son conocidas como Fórmulas de Little.

11.3 Fórmula Erlang B

Esta fórmula se puede mostrar como ejemplo clásico de la aplicación de teoría de colas, en una expresión compacta, que expresa la relación existente entre el tráfico y el desempeño, mediante la probabilidad de bloqueo P_B , que puede tener un cliente en una red de N circuitos sobre la que los usuarios imponen una intensidad de tráfico ρ , que es el producto entre; la tasa de llegadas λ =cantidad promedio de llamadas por unidad de tiempo, y el tiempo promedio de servicio $\frac{1}{\mu}$ de las llamadas donde μ = llamadas servidas por unidad de tiempo .

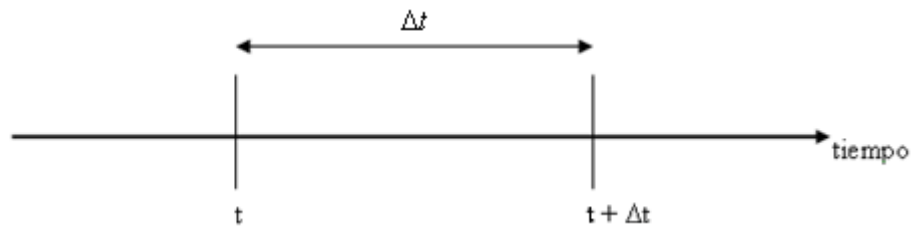
$$\rho = \lambda / \mu \quad ; \quad P_B = \frac{\rho^N / N!}{\sum_{n=0}^N \rho^n / n!}$$

Esta fórmula es válida ante la suposición de que las llamadas se producen según un proceso estacionario de Poisson es decir, se producen en forma aleatoria y con una distribución de tiempos entre llamadas exponencial.

11.4 Proceso de Poisson

Si consideramos las siguientes hipótesis

1. Un proceso en el cual cada llegada sea independiente de cuando se produjo la anterior. Llamando a esto “proceso sin memoria”.
2. Población infinita de forma que el número de fuentes sea tan grande que se pueda considerar que la tasa media de llegada de unidades no depende de la ventana temporal y es por tanto una constante cuyo valor es “ λ ” llegadas por unidad de tiempo.
3. Que la probabilidad de que se produzca una llegada sea proporcional al tiempo Δt , es decir, que sea $\lambda \Delta t + O(\Delta t)$, donde $O(\Delta t)$, hace referencia a los términos de orden superior a Δt los cuales tienden a cero en forma mas rápida conforme Δt tiende hacia cero.



Bajo estas hipótesis, la probabilidad de que se produzcan n llegadas en un tiempo T es:

$$P_n(T) = \frac{(\lambda T)^n}{n!} e^{-\lambda T}$$