

REDUCTOR DE RUIDO MEDIANTE LA TRANSFORMADA WAVELET

REFERENCIA PACS: 43.60.Bf.

Enric Gaus, Fabiola Ordoyo
Departamento de Acústica
Ingeniería La Salle
Universidad Ramon Llull
Pg. Bonanova nº 8
08022 Barcelona
E-mail: egaus@salleURL.edu

ABSTRACT

There are a lot of audio signals with noise, and there are a lot of noise reducers based on filtering techniques. We present a noise-reduction method based on the Wavelet Transform, with special mention on the Multiresolution Analysis. The system has only one input and only one output and there are some parameters accessible for the user. We will explain the fundamentals of Multiresolution Analysis and how the reducer works.

1. INTRODUCCIÓN

La transformada wavelet (WT), es una herramienta matemática que ha cobrado una grande importancia gracias a sus aplicaciones en el campo de la visión artificial y de la compresión de señales. Concretamente, las wavelets ortonormales están muy relacionadas los PR-QMF (Perfect Reconstruction Quadrature Mirror Filters) que nos servirán para hacer una descomposición multibanda de una señal de audio. Es posible aprovecharse de las ventajas que nos ofrece el análisis multiresolución para realizar un reductor de ruido a tiempo real sin necesidad de patrones previos, como sucede con los reductores de ruido off-line.

Descomposiciones tiempo-frecuencia

El objetivo principal en el análisis de una señal está en conseguir una representación tiempo-frecuencia que nos permita extraer los parámetros de nuestra señal. La Transformada de Fourier (TF) es un tipo de representación que nos da información de frecuencia, pero no nos proporciona información alguna sobre cuándo aparece una característica particular de una de las frecuencias.

$$F(\Omega) = \int_{-\infty}^{\infty} f(t)e^{-j\Omega t} dt \leftrightarrow f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\Omega)e^{j\Omega t} d\Omega$$

Esto no es ningún problema en un análisis con señales estacionarias, pero es totalmente inadmisibile en un análisis con señales transitorias. Para solucionar este problema, Gabor (1946) propone el eventaneamiento de la TF, más conocida como la Short-Time Fourier Transform (STFT).

$$F(\Omega, \mathbf{t}) = \int_{-\infty}^{\infty} f(t)g^*(t - \mathbf{t})e^{-j\Omega t} dt$$



La nueva base de funciones es generada por la modulación y translación de la ventana $g(t)$ a lo largo de la señal $f(t)$. donde Ω y τ representan la modulación y la translación de la ventana respectivamente. La principal limitación de la STFT la encontramos en la longitud fija de la ventana, hecho que nos proporciona una resolución tiempo-frecuencia fija. Esta es una consecuencia del principio de incertidumbre (Papoulis 1977), que define para un par transformado $g(t) \leftrightarrow G(\Omega)$:

$$\sigma_T \sigma_\Omega \geq \frac{1}{2}$$

cumpliendose la igualdad sólo cuando $g(t)$ es una función gaussiana y con σ_T y σ_Ω representando el valor RMS de las funciones $g(t)$ y $g(\Omega)$. La fig. 1 muestra un plano tiempo-frecuencia donde cada una de las celdas (representativas de la resolución) depende de σ_T y σ_Ω . Esto nos conlleva dos limitaciones: la primera, debido al principio de incertidumbre, si precisamos gran resolución temporal, sacrificamos resolución frecuencial, y viceversa. La segunda limitación consiste en que todas las celdas són idénticas, tanto en alta como en baja frecuencia, obteniendo un "banco de filtros de banda constante".

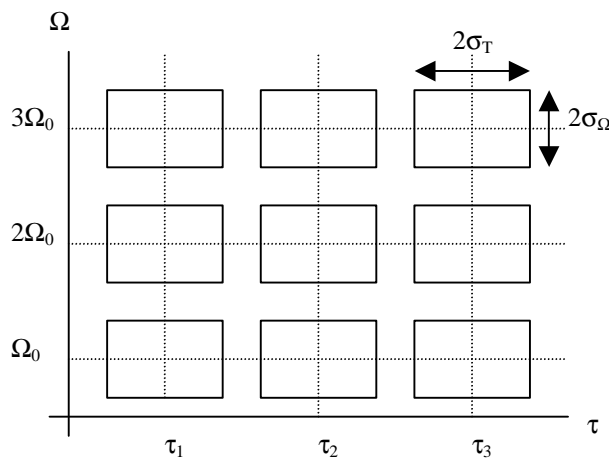


Fig. 1: Plano tiempo-frecuencia para la TF

Finalmente, la Discrete STFT y la Discrete-Time STFT no son más que un muestreo de la STFT en el dominio transformado ($\Omega \rightarrow m\Omega_0$, $\tau \rightarrow n\tau_0$) o en el dominio temporal ($t \rightarrow nT_0$) respectivamente, pero siguen manteniendo las limitaciones provocadas por el principio de incertidumbre.

2. TRANSFORMADA WAVELET

La WT es otro tipo de representación en un nuevo dominio transformado pero con una mayor localización tiempo-frecuencia. Para definir la WT precisamos de una función prototipo $\psi(t)$ (mother wavelet). Sea el par transformado $\psi(t) \leftrightarrow \Psi(\Omega)$ y aprovechandonos de la propiedad de escalaje de la TF podremos definir una familia de wavelets mediante la dilatación y translación de la wavelet madre (parámetros a y b respectivamente).

$$\frac{1}{\sqrt{a}} \mathbf{y} \left(\frac{t}{a} \right) \leftrightarrow \sqrt{a} \Psi(a\Omega)$$

$$y_{ab}(t) = \frac{1}{\sqrt{a}} y\left(\frac{t-b}{a}\right) \quad W(a,b) = \int_{-\infty}^{\infty} y_{ab}(t) f^*(t) dt = \langle y_{ab}, f \rangle$$

Esta nueva base del espacio $L^2(\mathbb{R})$ posee la característica de ser de corta duración para alta frecuencia y de larga duración para baja frecuencia. De este modo tenemos una mayor resolución tiempo-frecuencia permitiendo representar en un mismo plano tanto señales transitorias como señales de muy lenta variación. La fig.2 muestra el plano tiempo-escala (ya no hablamos de frecuencia, sino de escala) para la WT con celdas de distinta resolución en función de la escala. Nos encontramos con un “banco de filtros de porcentaje constante”.

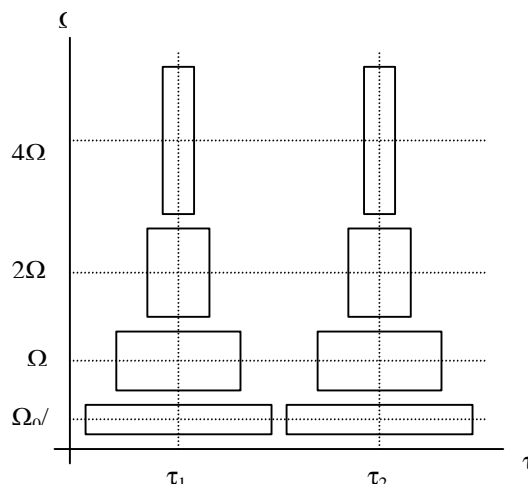


Fig. 2: Plano tiempo-escala para la TW

Que condiciones debe tener una función para poder convertirse en wavelet madre? La condición de admisibilidad impone que:

$$C_y = \int_0^{\infty} \frac{|\Psi(\Omega)|^2}{\Omega} d\Omega < \infty$$

Para que $C_y < \infty$ es necesario que $\Psi(0)=0$. Entonces, como que la TF de una señal en $\Omega=0$ es su valor medio, podemos imponer que $\int \psi(t) dt = 0$, es decir, que podemos escoger una función finita con valores tanto positivos como negativos que oscile alrededor del eje x. Además, deberemos escoger dicha función que tenga una buena localización tanto temporal como frecuencial. Ahora ya podremos reconstruir la señal original como

$$f(t) = \frac{1}{C_y} \int_{-\infty}^{\infty} \int_0^{\infty} \frac{1}{a^2} W(a,b) y_{ab}(t) da db$$

Pero la WT tiene dos grandes limitaciones: es redundante y de imposible implementación (es continua). De este modo, la Discrete Wavelet Transform (DWT) consiste en un muestreo de los parámetros a y b ($a=a_0^m$, $b=nb_0a_0^m$). Esta nueva familia de wavelets debe ser completa y no redundante. Escogeremos una discretización tal que sea mínima y nos asegure esas dos condiciones. Por otra parte, aparece un nuevo conjunto de funciones, las funciones de escalaje $\phi(t)$, que son ortonormales pero solo dentro de una misma escala. Estas funciones son tipo paso-bajo y complementan la función wavelet para representar una señal dentro de una misma escala, suavizándolas y proporcionando continuidad. Con una combinación lineal de las funciones de escalaje $\phi(t)$

escaladas y trasladadas podemos representar los niveles por debajo una resolución k (aproximaciones) mientras que con las combinaciones lineales de la wavelet madre $\psi(t)$ representaremos los niveles superiores a una resolución k (detalles).

3. ANÁLISIS MULTIRESOLUCIÓN

El análisis multiresolución (AMR) (S. Mallat i Y.Meyer, 1986) es otro tipo de representación donde una señal es la suma de una aproximación a una escala L y L señales de detalle a distintas resoluciones (superiores). Por tanto, es una representación implementable a la práctica y de ahí que muchas veces se hable de la transformada rápida de wavelet: la FFT es a la TF lo que el AMR es a la WT. Sin entrar en cuestiones matemáticas que justifiquen su funcionamiento, diremos que el análisis multiresolución es una secuencia $\{V_i\}_{i=-\infty}^{\infty}$ de subespacios $L^2(\mathbb{R})$ que satisfacen una serie de condiciones que pueden resumirse como que V_i representa el espacio de todas las funciones en $L^2(\mathbb{R})$ cuyos detalles de resolución menor que 2^{-i} han sido despreciados. Para su implementación usaremos una descomposición multibanda según un árbol piramidal y diádico, filtrando con filtros FIR PR-QMF (Perfect Reconstruction Quadrature Mirror Filters) y delmando la señal después de filtrarla (durante el proceso de análisis) o interpolandola antes de filtrarla (durante el proceso de síntesis).

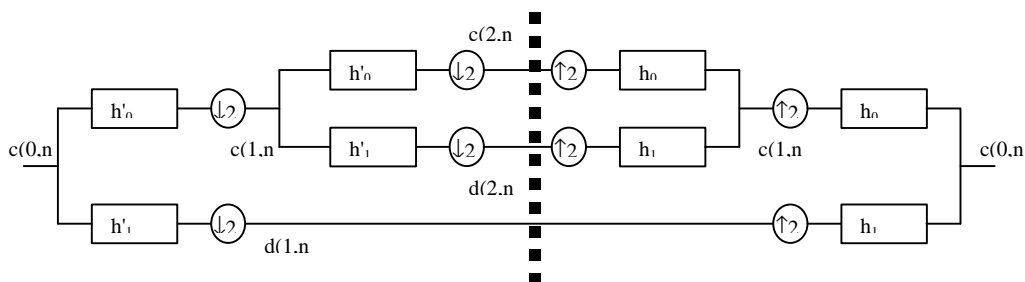


Fig. 3: Esquema Análisis multiresolución

En la fig. 3 puede verse un esquema de la descomposición y reconstrucción de una señal mediante AMR. Observamos que la frecuencia más alta (d_1) mantiene una frecuencia de muestreo más elevada, y por tanto un mayor número de puntos más elevado, que los detalles de los siguientes niveles de resolución. En el último escalón de la descomposición, nos encontramos con la aproximación a_m con la misma frecuencia de muestreo y número de puntos que el detalle a ese mismo nivel d_m . Con estos cambios de frecuencia de muestreo conseguimos obtener una señal descompuesta cuya longitud es idéntica a la de la señal original. Además, esta descomposición podría interpretarse como un análisis tiempo-frecuencia de una señal filtrada con filtros de octava y sin ocupar más memoria que la de la secuencia original. Increíble, no?

4. REDUCTOR DE RUIDO

Las aplicaciones más interesantes del AMR se encuentran en la compresión de datos y en los reductores de ruido. Nosotros, nos ocuparemos de estos últimos. Los reductores de ruido más comunes precisan de una estimación espectral del ruido a eliminar y mediante técnicas de filtraje modifican el espectro de la señal ruidosa. Nosotros trataremos de estudiar un sistema que sin aprendizaje previo sea capaz de reducir el ruido de una señal de forma apreciable. No debemos confundirnos con las técnicas de cancelación activa de ruido, puesto que estas precisan de una señal de error que tratarán de minimizar mediante algoritmos adaptativos (típicamente el LMS).

Veamos que entendemos como ruido. Un ruido es una señal no deseada que se ha sumado a una señal que queremos analizar. Para poder caracterizar una señal de ruido necesitamos su correlación, su espectro y su función de distribución. Como norma general, el ruido de alta frecuencia influye a los primeros niveles de detalle

del AMR, mientras que la baja frecuencia a los últimos niveles y a la aproximación. Un ruido blanco descompuesto por AMR, cuanto mayor sea el nivel de descomposición, menor es el nivel de ruido. Pero a la salida de cada nivel, no tendremos ruido blanco puesto que el filtro habrá coloreado su respuesta en frecuencia. Si a la entrada del sistema tenemos un ruido gaussiano, a la salida de cada nivel seguirá siendo gaussiano. Supongamos que tenemos una señal ruidosa $s(n)=f(n)+\sigma e(n)$. El objetivo de nuestro reductor de sonido es eliminar al máximo $s(n)$ para reconstruir $f(n)$. Los pasos a seguir son los siguientes: 1) Descomponer la señal ruidosa (AMR) hasta un nivel deseado N. 2) Para cada uno de los niveles, seleccionar un umbral y aplicar un algoritmo de Thresholding. 3) Reconstruir la señal.

4.1. Thresholding

El thresholding consiste en eliminar las muestras de una señal cuyo valor absoluto es inferior a un umbral establecido. El Hard Thresholding es el método más simple. El Soft Thresholding tiene unas propiedades matemáticas más interesantes y con él se obtienen mejores resultados. El Hard Thresholding consiste en forzar a 0 las muestras cuyo valor absoluto es inferior a un umbral establecido. El Soft thresholding es una extensión del Hard Thresholding, primero forzando a 0 las muestras cuyo valor absoluto es inferior a un umbral establecido y luego desplazando (offset) las muestras restantes alrededor del 0.



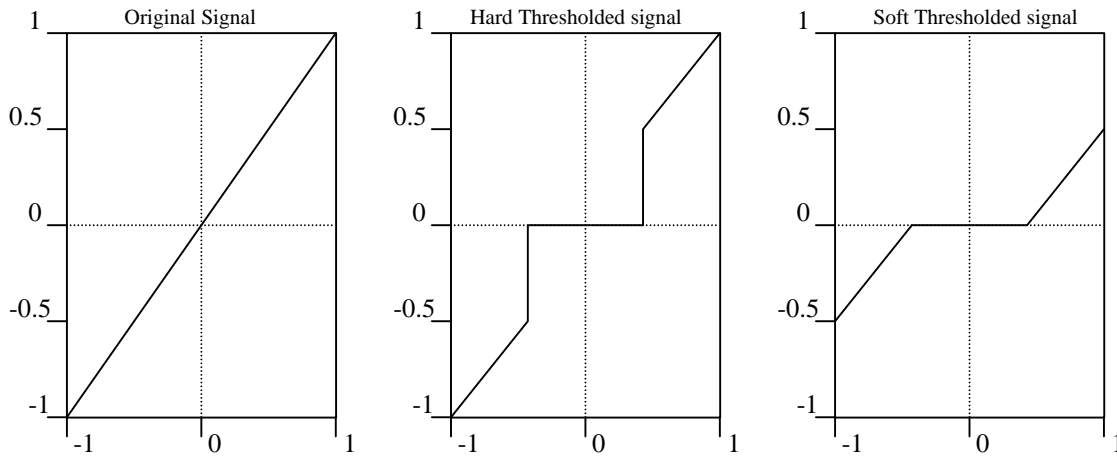


Fig. 5: Funcion de transferencia para los distintos métodos de Thresholding

5. RESULTADOS PRÁCTICOS

Hemos implementado todo este sistema en Matlab, con una descomposición AMR hasta un nivel $N=7$ usando filtros de daubechies db4 y un Soft Thresholding. La elección de umbrales es un parámetro de entrada para el usuario. Por ejemplo, una de las pruebas realizadas proporciona el siguiente resultado: para un tono puro a $f=1\text{KHz}$ y amplitud 1, se ha sumado un ruido blanco gaussiano de amplitud 0.2. Después de pasar la señal por el reductor de ruido se aprecia una clara disminución del ruido consiguiendo una relación S/N de unos 60dB, mientras que originariamente teníamos solo unos 30 dB de relación S/N.

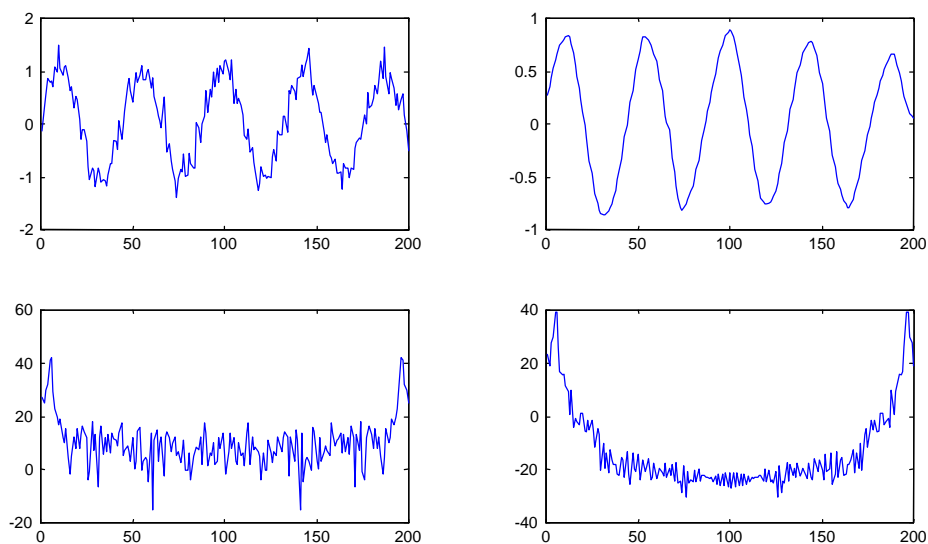


Fig. 6: Resultados prácticos



También se ha probado el sistema con señales de audio ruidosas (procedentes de una pletina cassette) y se ha notado una notable mejora al reducir el ruido de fondo (hiss). No se ha cuantificado la mejora puesto que para distintos pasajes (fortissimo o pianissimo, tipo de instrumento, etc.) es necesario modificar el valor de los umbrales escogidos para cada nivel.

6. CONCLUSIONES

Después de hacer el estudio de la TW y del AMR, hemos visto una aplicación dónde el AMR es de vital importancia. Si bien es cierto que tanto la TW y al AMR se basan en unos fundamentos matemáticos un tanto engorrosos, hemos dejado una idea intuitiva bastante sólida sin entrar en detalles de formulación. La técnica de reducción de ruido es bastante sencilla, una vez tenemos la señal descompuesta. Puesto que a priori nunca sabremos que clase de ruido entrará en nuestro sistema, dejamos la elección al usuario. Esto tampoco es ningún problema, ya que cómo se explica en la otra comunicación "Implementación de efectos de Audio a tiempo Real" el usuario tiene todos los umbrales accesibles en la pantalla del programa (Plug-in). Tan solo podríamos proponer una detección automática basada en histogramas u otras técnicas, pero esto ya formaría parte de otro estudio...

7. BIBLIOGRAFIA

- MULTIREOLUTION SIGNAL DECOMPOSITION, Ali N. Akansu, Richard A. Haddad, Academic Press, 1992
- THE COMPUTER MUSIC TUTORIAL, Curtis Roads, Ed. The MIT Press 1998, 3rd. Printing
- FRACTAL WAVELET COMPRESSION OF AUDIO SIGNALS, Robert A. Wannamaker, Edward R. Vrscay. J. Audio Eng. Soc. Vol 45, No. 7/8, 1997 July/August
- DISCRETE-TIME SIGNAL PROCESSING, Alan V. Oppenheim, Ronald W. Schaffer, Ed. Prentice Hall, 1989
- WAVELET TOOLBOX MANUAL, Matlab.

