Curso de Posgrado

Análisis y Procesamiento Avanzado de Señales

Guía de Trabajos Prácticos N^{o} 4

Transformada Ondita Discreta Diádica
Paquetes de Onditas

Curso acreditable en:
Doctorado en Ingeniería, UNL
Maestría en Ingeniería Biomédica, UNER

Junio de 2009

1. Objetivos

El objetivo general de esta guía de trabajos prácticos es que el alumno pueda implementar el algoritmo de la *Transformada Ondita Discreta Diádica* (TODD) y su generalización a Paquetes de Onditas, a partir de la aplicación de operaciones elementales de filtrado, submuestreo e interpolación. En particular, se pretende que el alumno pueda:

- Comprender los fundamentos teóricos del algoritmo de la TODD.
- Utilizar los filtros de *respuesta finita al impulso* (FIR) relacionados con las onditas más conocidas.
- Comprender la relación entre los coeficientes de estos filtros y las funciones ondita y escala.
- Implementar mediante operaciones sencillas las etapas de análisis y síntesis involucradas en la TODD.
- Generalizar el algoritmo de la TODD para el cálculo de la transformada Paquete de Onditas.

Presente los desarrollos requeridos en forma manuscrita, prolija y con letra **clara** y **legible**. No se aceptará en formato electrónico la parte teórica o conceptual de los trabajos. La parte informática deberá ser presentada en un archivo comprimido `TP#NombreAlumno.zip` o `TP#NombreAlumno.rar`, con el desarrollo en un archivo `TP#NombreAlumno.pdf` y un directorio con los correspondientes archivos `[filename.m]` y las salidas obtenidas, si correspondiera.

2. Ejercicios de laboratorio

Fecha: 23 de junio, 2009.

Ejercicio 1: Generación de señales.
En este ejercicio se generarán las señales que se utilizarán en el resto de la guía.

a) Escriba una función `low` para generar la señal discreta:

$$x[n] = 1, \;\; n = 1, 2, \ldots, L.$$

b) Escriba una función `high` para generar la señal discreta:

$$x[n] = (-1)^n, \;\; n = 1, 2, \ldots, L.$$

c) Grafique el resultado de las señales generadas por ambas funciones para $L = 2^5$, ¿Por qué las denominamos de este modo?.

d) Genere una señal sinusoidal muestreada a 1024 Hz, durante un segundo, cuya frecuencia aumente linealmente desde 0 a 500 Hz durante su duración. La amplitud de la señal debe decaer linealmente desde 1 a 0 en el lapso de tiempo indicado.

e) Utilizando escalas apropiadas, grafique la señal temporal, y grafique la magnitud de su transformada de Fourier.

Ejercicio 2: Generación de filtros y convolución.

a) Escriba una función que realice la convolución circular discreta entre dos señales $x[n]$ y $h[n]$, utilizando ciclos `for`[1]. En ésta se debe considerar a $x[n]$ periódica, pero $h[n]$ debe ser nula fuera de su rango de definición. Si la longitud de x es N y la longitud de h es M, la convolución circular se puede expresar mediante la siguiente ecuación:

$$y[k] = \sum_{l=1}^{M} h[l]x[((N + k - l) \mathrm{mod} N) + 1],$$

para $1 \leq k \leq N$, donde mod es la operación módulo entero (resto de la división entera).

b) Modifique la función anterior de forma que permita el agregado de un retardo. Tenga en cuenta que, dado que la convolución es circular, éste se manifiesta como un desplazamiento (shift) circular de la salida. Aplique esta función sobre señales sencillas y para diferentes retardos.

[1] Puede utilizar como base la función implementada en la Guía Nº1

c) Genere una función para implementar los filtros de promedio móvil y diferencia móvil. Estos filtros se definen en términos de sus coeficientes (y dan lugar a la ondita de Haar si se los normaliza):

$$h[1] = 1/2,\ h[2] = 1/2 \ \text{(promedio móvil)}$$

y

$$g[1] = 1/2,\ g[2] = -1/2 \ \text{(diferencia móvil)}.$$

La función debe permitir elegir si se normalizarán o no los coeficientes. Observe que el filtro de promedio móvil actúa como pasa-bajos. Para comprobarlo aplique $\mathbf{h}$ a las señales generadas por `low` y `high`, respectivamente (para $L = 2^5$). Grafique las señales y el correspondiente resultado. Repita el mismo procedimiento con $\mathbf{g}$ para ver que el filtro de diferencia móvil actúa como pasa-altos.

d) Escriba una función para generar los siguientes filtros, los que tras su normalización conducen a la famosa ondita Daubechies 4 (que tiene $p = 2$ momentos desvanecientes):

$$\mathbf{h} = 1/8 \left[1+\sqrt{3},\ 3+\sqrt{3},\ 3-\sqrt{3},\ 1-\sqrt{3}\right],$$

$$\mathbf{g} = 1/8 \left[1-\sqrt{3},\ -3+\sqrt{3},\ 3+\sqrt{3},\ -1-\sqrt{3}\right].$$

La función debe permitir elegir si se normalizarán o no los coeficientes.

e) En la ecuación 7.34 de [1] se establece que

$$\forall \omega \in \mathbb{R}, \quad |\hat{h}(\omega)|^2 + |\hat{h}(\omega+\pi)|^2 = 2.$$

Por otro lado, la ecuación 7.58 establece que

$$\hat{g}(\omega) = e^{-i\omega}\hat{h}^*(\omega+\pi).$$

Combinando estas dos (demuestrelo!), se obtiene:

$$\forall \omega \in \mathbb{R}, \quad |\hat{h}(\omega)|^2 + |\hat{g}(\omega)|^2 = 2.$$

Verifique numericamente la validez de esta ecuación para los filtros Haar normalizados y los Daubechies 4 normalizados. Es decir, estime la respuesta en frecuencia de los filtros $\mathbf{h}$ y $\mathbf{g}$ y verifique que la suma de las magnitudes al cuadrado da 2.

Ejercicio 3: Banco de filtros de análisis y submuestreo.

Este ejercicio constituye el primer paso hacia una descomposición ondita diádica. Se aplican ambos filtros pasa-bajos y pasa-altos a una señal, guardando sólo las muestras impares del resultado (submuestreo). Estos pasos se muestran en la Figura 1 parte a).

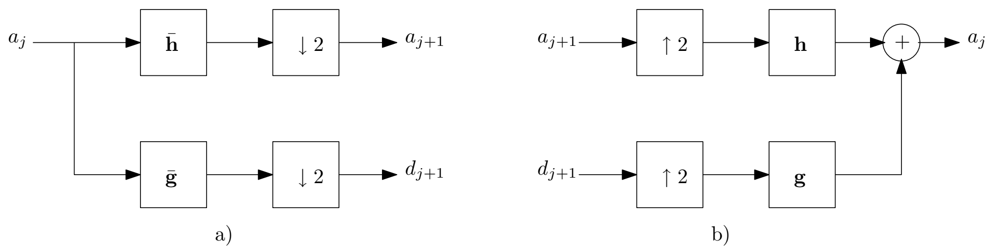

Figura 1: Algoritmo de banco de filtros para descomposición y reconstrucción diádica.

a) Escriba una función que realice el submuestreo de una señal, preservando las muestras de índice impar (1,3,5, etc)[2]. La función debe producir una señal con la mitad de la longitud que la señal de entrada.

b) Implemente la primer etapa de la descomposición con banco de filtros en la función `dstep`, de acuerdo a la Figura 1 parte a). Dada una señal de entrada, debe producir los coeficientes de aproximación y detalle. Para los de aproximación se debe filtrar la señal con $\bar{\mathbf{h}}$. Note que $\bar{h}[n] = h[N - n + 1]$, es decir que produce una inversión temporal de los coeficientes. Luego se submuestrea con la función del punto 3.a). Además, en el filtrado se debe compensar el retardo introducido por el filtro, que es igual a la mitad de su longitud. De la misma forma se obtienen los coeficientes de detalle, pero utilizando $\bar{\mathbf{g}}$ en lugar de $\bar{\mathbf{h}}$[3].

c) Utilice esta función sobre la señal sinusoidal generada en el ejercicio 1.d). Use los filtros Haar normalizados anteriormente diseñados. Grafique la magnitud de la transformada de Fourier de las señales después del filtrado, y después del submuestreo, y compárelas con la de la señal original ya analizada.

d) Repita el ejercicio anterior para los filtros Daubechies normalizados.

Ejercicio 4: Banco de filtros de síntesis y sobremuestreo.
Este ejercicio está relacionado con la reconstrucción de la señal (luego del sobremuestreo). Para la reconstrucción primero se sobremuestrea agregando ceros entre las muestras y luego se filtra. Estos pasos se muestran en la Figura 1 parte b).

a) Escriba una función que realice el sobremuestreo de una señal mediante el agregado de ceros. Por ejemplo, para la señal [1 2 3] debe producir [0 1 0 2 0 3]. La señal de salida debe tener el doble de muestras que la de entrada.

b) Implemente la primer etapa de la reconstrucción con banco de filtros en la función `rstep`, de acuerdo a la Figura 1 parte b). El método consiste en sobremuestrear con la función anteriormente desarrollada y luego filtrar con los

[2]Notese que se ha adaptado el método para la notación de Matlab/Octave en donde la primer muestra es 1 y no 0. En la bibliografia se puede encontrar que el submuestreo retiene las muestras pares, que serán las mismas para un sistema indexado desde 0.

[3]Verifique sus resultados comparándolos con los obtenidos mediante la función `dwt` del toolbox de onditas de Matlab, si lo tiene disponible.

filtros **h** y **g**. Nuevamente el filtrado se debe hacer con la convolución circular anteriormente implementada, y se debe incluir un retardo igual a la mitad de la longitud de los filtros. La reconstrucción se completa sumando ambas señales obtenidas.

c) Utilice esta función para reconstruir la señal sinusoidal anteriormente analizada con los filtros Haar normalizados. Grafique la magnitud de la transformada de Fourier de las señales después del sobremuestreo y después del filtrado (pero antes de sumar las señales).

d) Repita el ejercicio anterior pero para la descomposicion producida con los filtros Daubechies normalizados.

Ejercicio 5: Transformada Ondita Diadica Discreta (TODD).
En este ejercicio vamos a iterar el paso de filtrado y submuestreo. Más precisamente, luego de calcular las aproximaciones utilizando nuestro banco de filtro de análisis, volvemos a aplicarles el banco de filtros, y así sucesivamente. Utilizando este procedimiento dividiremos a la señal en escalas de diferente resolución.

a) Dada una señal de longitud N, el máximo nivel de descomposción que se puede aplicar es el número de veces (entero) que N es divisible por 2. Para el caso particular de señales con $N = 2^p$ se puede descomponer hasta nivel p. Si la señal tiene un número impar de muestras, no se puede aplicar este algoritmo en forma directa. Implemente una función `maxlevel` que determine el nivel máximo de descomposición admitido por una señal.

b) Implemente una función que realice la descomposición hasta el nivel deseado, para generar la TODD (iterada sobre los coeficientes de aproximación). Esto es, debe aplicar la función `dstep`, en el primer nivel a la señal original, y luego iterativamente sobre los coeficientes de aproximación obtenidos en la escala anterior. Se debe verificar utilizando la función del ejercicio anterior, que el nivel de descomposición deseado sea válido. Se sugiere organizar los coeficientes en un vector, por ejemplo para una descomposición hasta nivel 3, usar la estructura $[a_3 d_3 d_2 d_1]$.

c) Implemente una función que realice la reconstrucción iterativamente desde el nivel dado, es decir, iterando la función `rstep` para reconstruir los coeficientes de aproximación en la escala siguiente.

d) Aplique a una señal cualquiera las funciones de descomposición y reconstrucción en forma secuencial y compare la señal original con la reconstruida. Grafique la señal original, la señal reconstruida y el error de reconstrucción. Utilice los filtros Haar y Daubechies normalizados.

Ejercicio 6: Paquetes de Onditas.
Las funciones `dstep` y `rstep` pueden ser utilizadas para calcular la transformada

Paquete de Onditas (TPO), si se determina una iteración diferente en lugar de descomponer iterativamente los coeficientes de aproximación como en la TODD. En este punto se implementará dicha transformación.

a) Escriba una función que permita calcular la descomposición Paquetes de Onditas completa (es decir sobre el árbol completo), iterando con la función `dstep` tanto sobre los coeficientes de aproximación como en los de escala, hasta el nivel deseado.

b) Escriba una una función `wpdec` que permita ingresar la estructura del árbol de descomposición deseado y devuelva los coeficientes de la descomposición obtenida, iterando la función `dstep` sobre los coeficientes adecuados.

c) Escriba una función `wprec` que reconstruya mediante la iteración de la función `rstep`, una señal, dada su descomposición generada con `wpdec` y la estructura del árbol.

d) Utilice esta descomposición y reconstrucción sobre una señal cualquiera. Grafique la señal original, la reconstruida y el error de reconstrucción. Utilice los filtros Haar y Daubechies normalizados.

Ejercicio 7: El algoritmo cascada.
En este ejercicio estimaremos las funciones ondita y escala a partir de los coeficientes de los filtros correspondientes. Para facilitar el abordaje implementaremos un método que sólo utilice las funciones que ya tenemos a mano:

- El esquema multiresolución descompone una señal en versiones trasladadas y dilatadas de las funciones escala (aproximaciones) y ondita (detalles).
- El resultado de esta descomposición son los coeficientes de las funciones escala y ondita respectivamente.
- Consecuentemente si aplicamos el esquema de síntesis a un vector con un sólo uno en la posición adecuada (de longitud apropiada) reconstruiremos una versión trasladada y dilatada de la correspondiente función escala u ondita.

a) Genere una señal con todos ceros de longitud $N = 2^7$. Suponga que ésta es el resultado de una descomposición TODD hasta nivel 5, que tiene los coeficientes $[a_5 d_5 d_4 d_3 d_2 d_1]$. En la posición correspondiente a $a_5[2]$ (en el vector total será la posición 2) coloque un 1. Reconstruya usando la TODD inversa, y grafique la señal obtenida. Utilice para ello los filtros de Haar y los Daubechies. Esto permite obtener la función de escala para dichos filtros, compárelas con las que aparecen en la bibliografía (por ejemplo, en la pagina 23 de [2] para la Haar, y en página 253 de [1] para la Daubechies con $p = 2$ momentos desvanecientes).

b) Genere una señal con todos ceros de longitud $N = 2^7$. Suponga que ésta es el resultado de una descomposición TODD hasta nivel 5, que tiene los coeficientes $[a_5 d_5 d_4 d_3 d_2 d_1]$. En la posición correspondiente a $d_5[2]$ (en el vector total será la posición 6) coloque un 1. Reconstruya usando la TODD inversa, y grafique la señal obtenida. Utilice para ello los filtros de Haar y los Daubechies. Esto permite obtener la función de escala para dichos filtros, compárelas con las que aparecen en la bibliografía (por ejemplo en página 24 de [2] para la Haar, y en página 253 de [1] para la Daubechies con $p = 2$ momentos desvanecientes).

Referencias

[1] S.G. Mallat. *A Wavelet Tour of Signal Processing.* Academic Press, 2nd edition, September 1999.

[2] Gilbert Strang. *Course on Wavelets and Signal Processing.* Centre for Mathematical Sciences, Lund Institute of Technology, October 2002.