

Universidad Autónoma Metropolitana Unidad Azcapotzalco
División de Ciencias Básicas e Ingeniería
Licenciatura en Ingeniería en Computación

Nombre proyecto

Cálculo de parámetros en filtros de reconstrucción perfecta

Modalidad: Proyecto Tecnológico

Trimestre 2015 Primavera

María del Carmen Elizabeth Macías Sánchez
Matrícula 205306827
al205306827@alumnos.azc.uam.mx

Asesor:
Dr. Oscar Herrera Alcántara
Asociado D

Fecha de entrega:
02 de Septiembre del 2015

Yo, Oscar Herrera Alcántara, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.

A handwritten signature in black ink, consisting of a circle with a horizontal line through it and some stylized letters, positioned above a horizontal line.

Dr. Oscar Herrera Alcántara

Yo, María del Carmen Elizabeth Macías Sánchez, doy mi autorización a la Coordinación de Servicios de Información de la Universidad Autónoma Metropolitana, Unidad Azcapotzalco, para publicar el presente documento en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.



María del Carmen Elizabeth Macías Sánchez

Índice

Resumen.....	1
Introducción.....	1
Antecedentes	2
Justificación	3
Objetivos	4
Objetivo general.....	4
Objetivo específico	4
Marco teórico.....	4
<i>Parametrización de filtros de longitud 4.....</i>	<i>5</i>
<i>Ecuaciones inversas para parametrizaciones de filtros de longitud 4</i>	<i>6</i>
<i>Parametrizaciones de filtros de longitud 6</i>	<i>7</i>
<i>Ecuaciones inversas para parametrizaciones de filtros de longitud 6</i>	<i>10</i>
Desarrollo del proyecto.....	11
Resultados	12
<i>Resultados obtenidos para filtros de longitud 4</i>	<i>12</i>
<i>Resultados obtenidos para filtros de longitud 6:</i>	<i>16</i>
<i>Resultados obtenidos para filtros de longitud 4 al exceder el valor del umbral:</i>	<i>18</i>
<i>Resultados obtenidos para filtros de longitud 6 al exceder el valor del umbral:</i>	<i>21</i>
Conclusiones	23
Bibliografía.....	24
Entregables	25

Resumen

En el presente trabajo se calcula la respuesta al impulso de filtros ortogonales de reconstrucción perfecta dependientes de un solo parámetro para el caso de longitud 4, y de dos parámetros para filtros de longitud 6. Se calcularon las ecuaciones paramétricas inversas para filtros de longitud 4 y longitud 6.

También se calcularon los valores de los parámetros usando las ecuaciones paramétricas inversas a fin de obtener los correspondientes valores de respuestas al impulso que permitieran calcular el error entre la respuesta al impulso original y la calculada con los parámetros obtenidos.

Se calculó el error promedio de los parámetros dados como entrada y los obtenidos mediante las ecuaciones paramétricas inversas, sobre un número determinado de iteraciones.

Se observó que sí es posible determinar tanto los valores de los parámetros como de las respuestas al impulso, con una precisión aceptable al aplicar las ecuaciones paramétricas inversas.

Se observaron casos en los que se generaron soluciones negativas para los parámetros α en el caso de filtros de longitud 4, y α y β para filtros de longitud 6 que se obtuvieron al fijar un umbral de aproximación entre las respuestas al impulso de referencia y las calculadas con los valores de los parámetros usando las ecuaciones inversas.

Introducción

Uno de los resultados más interesantes en el ámbito de la Transformada Wavelet Discreta (TWD) es que se puede implementar utilizando un filtro pasabajas y un filtro pasaaltas. Los filtros ortogonales de reconstrucción perfecta suelen expresarse mediante un conjunto de coeficientes reales. Un ejemplo corresponde al filtro Haar [1] en donde el filtro pasabajas es $h = (h_0, h_1) = \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)$ y el filtro pasaaltas es $g = (g_0, g_1) = \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)$. Este conjunto de coeficientes representa la respuesta al impulso de los filtros (en el dominio temporal), que corresponden respectivamente a un filtro pasabajas y aun filtro pasaaltas (en el dominio de la frecuencia).

Un filtro ortogonal con mayor longitud puede contener cuatro, seis o un número par de coeficientes y, en general, existen una infinidad de filtros que cumplen con la propiedad que al aplicarlos apropiadamente a una señal (audio, imagen, voz, música, etc.) permiten recuperar fielmente una señal original. La etapa de análisis consiste en aplicar los filtros h y g a la señal original, y la fase de síntesis consiste en aplicar los filtros h y g al resultado de la fase de análisis, por lo que se puede decir que las fases de análisis y síntesis son procesos inversos.

En la práctica la señal original y la señal reconstruida (salida de la fase de síntesis) suelen diferir por el error de precisión de la aritmética de punto flotante, pero en la literatura se denominan filtros de reconstrucción perfecta. Se han propuesto varios filtros de reconstrucción perfecta como los así llamados *Daubechies* [2], *Coiflets* [3], y *Symlets* [4]. También se han

publicado ecuaciones de filtros que involucran funciones trigonométricas evaluadas en parámetros en intervalos $[0, 2\pi)$ lo cual permite generar un conjunto infinito de filtros de reconstrucción perfecta, pero continúa presente el problema siguiente: ¿Dado un filtro de reconstrucción perfecta que parámetros le corresponden? En el presente trabajo se propone implementar un algoritmo que devuelva parámetros en el intervalo de $[0, 2\pi)$ dada la respuesta al impulso de un filtro de longitud 4 y de longitud 6.

La parametrización de filtros ortogonales basados en wavelets permite reducir el espacio de búsqueda de conjuntos de coeficientes que cumplan con la condición de reconstrucción perfecta.

Se han propuesto varios métodos para calcular filtros de reconstrucción perfecta, el más popular fue presentado en un trabajo por I. Daubechies [5], quien presentó filtros que generan funciones *wavelet* que maximizan el número de momentos de desvanecimiento y que se reflejan en la maximización de la *regularidad* de los *wavelets*, es decir, son los más “suaves” posiblemente construibles para una longitud de filtro dada.

Otro método de diseño es el algebraico, en donde se proponen resolver ecuaciones derivadas de las condiciones de reconstrucción perfecta y la maximización de los momentos de desvanecimiento. Este método provee ecuaciones paramétricas que incluyen como casos especiales a los filtros obtenidos por I. Daubechies. El cálculo de los parámetros correspondientes para los filtros Daubechies se ha reportado para mostrar que son, en efecto, casos especiales. Sin embargo, poco se ha avanzado en proponer un método para calcular los parámetros dados los valores de coeficientes de filtros ortogonales. Este es justo el objeto de estudio del presente trabajo.

Antecedentes

La parametrización de filtros de reconstrucción perfecta ha sido abordada en trabajos previos y se han propuesto varias parametrizaciones así como métodos para parametrizarlos.

En [6] se discute la parametrización de coeficientes de funciones de escalamiento (ortogonales a las funciones wavelet) con varios momentos de desvanecimiento.

En [7] se presentan parametrizaciones de wavelets ortonormales con soporte en el intervalo $[0, 2N-1)$ en donde N es un entero par que involucra $N-1$ parámetros que varían en el intervalo $(0, 2\pi)$.

En [8] se utilizan parametrizaciones de filtros y se presenta una técnica para calcular la mejor wavelet para una imagen dada, la técnica usada calcula la relación señal a ruido y ajusta los coeficientes de los filtros (y por ende las funciones wavelet) dinámicamente.

En [9] se presenta un método algebraico para parametrizar filtros de longitud 4, 6, 8 y 10, y se presentan valores específicos de parámetros que corresponden a los filtros publicados por I. Daubechies.

En [10, 11] se aplica un algoritmo genético para determinar el conjunto de parámetros de filtros de longitud 4, 6, 8 y 10 que minimizan la cantidad de coeficientes wavelet

despreciables, y que garantizan el menor error de reconstrucción en imágenes en escalas de grises.

El problema abordado en este trabajo consiste en que dado un filtro de reconstrucción perfecta saber ¿qué parámetros le corresponden? Este problema aún no ha sido abordado ampliamente, y tampoco se aborda la existencia de múltiples soluciones.

El [12] se presentan algunas ecuaciones paramétricas inversas que procuran responder la pregunta anterior, sin embargo no se contempla el caso en el que los parámetros puedan estar fuera del rango $[0, 2\pi)$, y que por tanto requieran un “mecanismo de reparación”.

En el presente estudio se trabaja con filtros de longitudes 4 y 6, y se estudian varias alternativas de solución para calcular los parámetros. También se propone un mecanismo de “reparación” para que al final se presenten valores en el rango $[0, 2\pi)$.

Justificación

Existen varios métodos para parametrizar filtros ortogonales, pero poco se ha trabajado en el cálculo de los parámetros a partir de respuestas al impulso de un filtro dado.

Con el algoritmo propuesto se obtendrá un error de aproximación entre un filtro dado y uno generado con los parámetros calculados a partir de las ecuaciones correspondientes.

Es importante mencionar que no existe una única parametrización para un cierto filtro, y para una parametrización dada, se pueden generar varias opciones de “ecuaciones paramétricas inversas” las cuales ponen el valor de los parámetros en función de los coeficientes de la respuesta al impulso. Por ejemplo, para el filtro de longitud 4 se tienen cuatro coeficientes h_0 , h_1 , h_2 y h_3 dados por:

$$h_0 = \frac{1}{4} + \frac{1}{2\sqrt{2}} \cos \alpha \quad (1)$$

$$h_1 = \frac{1}{4} + \frac{1}{2\sqrt{2}} \sin \alpha \quad (2)$$

$$h_2 = \frac{1}{4} - \frac{1}{2\sqrt{2}} \cos \alpha \quad (3)$$

$$h_3 = \frac{1}{4} - \frac{1}{2\sqrt{2}} \sin \alpha \quad (4)$$

Que dependen del parámetro α .

De donde se puede despejar α en cada una de ellas mediante cálculo elemental, y obtener respectivamente:

$$\alpha = \arccos(2\sqrt{2} (h_0 - \frac{1}{4})) \quad (5)$$

$$\alpha = \arcsin(2\sqrt{2} (h_1 - \frac{1}{4})) \quad (6)$$

$$\alpha = \arccos(2\sqrt{2}(\frac{1}{4} - h_2)) \quad (7)$$

$$\alpha = \arcsin(2\sqrt{2}(\frac{1}{4} - h_3)) \quad (8)$$

Un ingeniero en computación en la UAM Azcapotzalco adquiere durante su formación los conocimientos para realizar este tipo de cálculos en las UEAs de Cálculo Diferencial y Cálculo Integral.

Cada una de las ecuaciones (5) a (8) puede dar un valor diferente para el parámetro α , e inclusive ese valor puede estar fuera de rango dada la periodicidad de las funciones trigonométricas involucradas, así que se requiere proponer e implementar un algoritmo que calcule el mejor valor de α^* que a su vez minimice el error de aproximación entre un filtro calculado con este valor de α^* , y el filtro h_0, h_1, h_2 y h_3 .

Objetivos

Objetivo general

Implementar un algoritmo que calcule los valores de parámetros de filtros ortogonales dada su respuesta al impulso.

Objetivo específico

1. Diseñar un algoritmo que calcule el error de aproximación entre un filtro dado y uno generado con los parámetros de las ecuaciones paramétricas inversas.
2. Hacer un programa que evalúe las ecuaciones paramétricas para filtros de longitud 4.
3. Hacer un programa que evalúe las ecuaciones paramétricas para filtros de longitud 6.
4. Calcular ecuaciones paramétricas inversas de filtros de longitud 6.
5. Hacer un programa que procure devolver valores de parámetros en el intervalo $[0, 2\pi)$.
6. Implementar un módulo de comparación, cálculo de errores y resultados estadísticos.
7. Identificar si existen algunas condiciones en la que el algoritmo proponga soluciones de parámetros con valores negativos.

Cabe mencionar que la medición de la eficiencia del algoritmo para este proyecto se hará al medir el número de soluciones con valores de parámetros positivos comparado con el número de parámetros con valores negativos.

Marco teórico

Como parte del marco teórico se revisa el método de parametrización algebraica [13] para obtener las ecuaciones para los filtros de reconstrucción perfecta de longitud 4 y 6.

Parametrización de filtros de longitud 4

Se parte de la ecuación de reconstrucción perfecta

$$|H(z)|^2 + |H(-z)|^2 = 1 \quad (9)$$

En donde $H(z) = h_0 + h_1z + h_2z^2 + h_3z^3 + \dots$ es un polinomio trigonométrico.

Entonces, al sustituir y desarrollar la ecuación (9) se obtiene

$$H_4(z) = h_0 + h_1z + h_2z^2 + h_3z^3 \quad (10)$$

De donde

$$2(h_0^2 + h_2^2 + h_1^2 + h_3^2) + 2(h_0h_2 + h_1h_3)z + 2(h_0h_2 + h_1h_3)z^2 = 1 \quad (11)$$

Entonces, para mantener la reconstrucción perfecta se hace

$$h_0^2 + h_2^2 + h_1^2 + h_3^2 = \frac{1}{2} \quad (12)$$

Y

$$h_0h_2 + h_1h_3 = 0 \quad (13)$$

Que sumar (12) con (13) multiplicado por (-2) se obtiene el cuadrado perfecto

$$(h_0 - h_2)^2 + (h_1 - h_3)^2 = \frac{1}{2} \quad (14)$$

En donde por analogía con la identidad trigonométrica $\sin^2x + \cos^2x = 1$ se deduce que:

$$(h_0 - h_2) = \frac{1}{\sqrt{2}} \cos \alpha \quad (15)$$

Y

$$(h_1 - h_3) = \frac{1}{\sqrt{2}} \sin \alpha \quad (16)$$

Puesto que

$$h_0 + h_1 + h_2 + h_3 = 1 \quad (17)$$

Y considerando que $H(z = -1) = 0$ se obtiene

$$h_0 - h_1 + h_2 - h_3 = 0 \quad (18)$$

Entonces, sumando (17) con (18) se tiene

$$h_0 + h_2 = \frac{1}{2} \quad (19)$$

Y sustrayendo (19) de (18) se tiene

$$h_1 + h_3 = \frac{1}{2} \quad (20)$$

Que al combinarlas se llega a:

$$\begin{aligned} h_0 &= \frac{1}{4} + \frac{1}{2\sqrt{2}} \cos \alpha & h_1 &= \frac{1}{4} + \frac{1}{2\sqrt{2}} \operatorname{sen} \alpha \\ h_2 &= \frac{1}{4} - \frac{1}{2\sqrt{2}} \cos \alpha & h_3 &= \frac{1}{4} - \frac{1}{2\sqrt{2}} \operatorname{sen} \alpha \end{aligned} \quad (21)$$

De manera análoga se puede llegar a las ecuaciones (8 a 13).

Ecuaciones inversas para parametrizaciones de filtros de longitud 4

Restando h_0 y h_2 se tiene $(h_0 - h_2) = \frac{1}{\sqrt{2}} \cos \alpha$

Y se despeja α :

$$\alpha = \operatorname{acos}(\sqrt{2}(h_0 - h_2))$$

Restando h_1 y h_3 se tiene $(h_1 - h_3) = \frac{1}{\sqrt{2}} \sin \alpha$

Y se despeja α :

$$\alpha = \operatorname{asin}(\sqrt{2}(h_1 - h_3))$$

Como se puede apreciar, existen al menos dos alternativas para calcular α .

Este trabajo plantea estudiar ¿cuál de estas ecuaciones inversas permite obtener el mejor valor de α que minimiza el error entre los coeficientes h_0 , h_1 , h_2 y h_3 y aquellos obtenidos cuando se vuelven a usar las ecuaciones (21)?

Otras alternativas para calcular α son mediante despeje directo en las ecuaciones (21).

De (21) se obtiene:

$$\alpha = \arccos\left(2\sqrt{2}\left(h_0 - \frac{1}{4}\right)\right)$$

$$\alpha = \arcsin\left(2\sqrt{2}\left(h_1 - \frac{1}{4}\right)\right)$$

$$\alpha = \arccos\left(-2\sqrt{2}\left(h_2 - \frac{1}{4}\right)\right)$$

$$\alpha = \arcsin\left(-2\sqrt{2}\left(h_3 - \frac{1}{4}\right)\right)$$

Parametrizaciones de filtros de longitud 6

Con las siguientes ecuaciones paramétricas se calcularon las ecuaciones paramétricas inversas de longitud 6 que son:

$$h_0 = \frac{1}{8} + \frac{1}{4\sqrt{2}} \cos \alpha + \frac{p}{2} \cos \beta \quad (22)$$

$$h_1 = \frac{1}{4} - \frac{1}{2\sqrt{2}} \cos \alpha \quad (23)$$

$$h_2 = \frac{1}{8} + \frac{1}{4\sqrt{2}} \cos \alpha - \frac{p}{2} \cos \beta \quad (24)$$

$$h_3 = \frac{1}{8} + \frac{1}{4\sqrt{2}} \sin \alpha + \frac{p}{2} \sin \beta \quad (25)$$

$$h_4 = \frac{1}{4} - \frac{1}{2\sqrt{2}} \sin \alpha \quad (26)$$

$$h_5 = \frac{1}{8} + \frac{1}{4\sqrt{2}} \sin \alpha - \frac{p}{2} \sin \beta \quad (27)$$

Donde $p = \frac{1}{2} \sqrt{1 + \sin\left(\alpha + \frac{\pi}{4}\right)}$

Restando las ecuaciones (22) y (24) se tiene

$$h_0 = \frac{1}{8} + \frac{1}{4\sqrt{2}} \cos \alpha + \frac{p}{2} \cos \beta \quad (22)$$

$$- \quad h_2 = \frac{1}{8} + \frac{1}{4\sqrt{2}} \cos \alpha - \frac{p}{2} \cos \beta \quad (24)$$

$$h_0 - h_2 = 2 \left(\frac{p}{2} \cos \beta\right)$$

$$h_0 - h_2 = p \cos \beta$$

Por lo tanto tenemos:

$$\cos \beta = \frac{h_0 - h_2}{p} \quad (28)$$

Restando las ecuaciones (25) y (27) tenemos

$$h_3 = \frac{1}{8} + \frac{1}{4\sqrt{2}} \sin \alpha + \frac{p}{2} \sin \beta \quad (25)$$

$$h_5 = \frac{1}{8} + \frac{1}{4\sqrt{2}} \sin \alpha - \frac{p}{2} \sin \beta \quad (27)$$

$$h_3 - h_5 = 2 \left(\frac{p}{2} \sin \beta \right)$$

$$h_3 - h_5 = p \sin \beta$$

Por lo tanto tenemos:

$$\sin \beta = \frac{h_3 - h_5}{p} \quad (29)$$

Dividiendo (28) y (29) tenemos

$$\tan \beta = \frac{\sin \beta}{\cos \beta}$$

$$\frac{\sin \beta}{\cos \beta} = \frac{\frac{h_3 - h_5}{p}}{\frac{h_0 - h_2}{p}}$$

$$\tan \beta = \frac{p (h_3 - h_5)}{p (h_0 - h_2)}$$

Despejando a β

$$\beta = \operatorname{atan} \left(\frac{h_3 - h_5}{h_0 - h_2} \right) \quad (30)$$

Sumando las ecuaciones (22) y (24) tenemos

$$h_0 = \frac{1}{8} + \frac{1}{4\sqrt{2}} \cos \alpha + \frac{p}{2} \cos \beta \quad (22)$$

+

$$h_2 = \frac{1}{8} + \frac{1}{4\sqrt{2}} \cos \alpha - \frac{p}{2} \cos \beta \quad (24)$$

$$h_0 + h_2 = \frac{1}{16} + 2 \left(\frac{1}{4\sqrt{2}} \cos \alpha \right)$$

$$h_0 + h_2 = \frac{1}{16} + \frac{1}{2\sqrt{2}} \cos \alpha$$

Por lo tanto tenemos:

$$\frac{1}{2\sqrt{2}} \cos \alpha = h_0 + h_2 - \frac{1}{16} \quad (31)$$

Sumando las ecuaciones (25) y (27) tenemos

$$h_3 = \frac{1}{8} + \frac{1}{4\sqrt{2}} \sin \alpha + \frac{p}{2} \sin \beta \quad (25)$$

+

$$h_5 = \frac{1}{8} + \frac{1}{4\sqrt{2}} \sin \alpha - \frac{p}{2} \sin \beta \quad (27)$$

$$h_3 + h_5 = \frac{1}{16} + 2 \left(\frac{1}{4\sqrt{2}} \sin \alpha \right)$$

$$h_3 + h_5 = \frac{1}{16} + \frac{1}{2\sqrt{2}} \sin \alpha$$

Por lo tanto:

$$\frac{1}{2\sqrt{2}} \sin \alpha = h_3 + h_5 - \frac{1}{16} \quad (32)$$

Dividiendo a (31) y (32) tenemos:

$$\frac{\frac{1}{2\sqrt{2}} \sin \alpha}{\frac{1}{2\sqrt{2}} \cos \alpha} = \frac{h_3 + h_5 - \frac{1}{16}}{h_0 + h_2 - \frac{1}{16}}$$

$$\frac{\sin \alpha}{\cos \alpha} = \frac{h_3 + h_5 - \frac{1}{16}}{h_0 + h_2 - \frac{1}{16}}$$

Entonces

$$\tan \alpha = \frac{h_3 + h_5 - \frac{1}{16}}{h_0 + h_2 - \frac{1}{16}}$$

Y despejando a α se tiene

$$\alpha = \operatorname{atan} \frac{h_3 + h_5 - \frac{1}{16}}{h_0 + h_2 - \frac{1}{16}} \quad (33)$$

Despejando la ecuación (23) obtenemos

$$h_1 = \frac{1}{4} - \frac{1}{2\sqrt{2}} \cos \alpha$$

$$h_1 - \frac{1}{4} = -\frac{1}{2\sqrt{2}} \cos \alpha$$

Despejando a α

$$\cos \alpha = 2\sqrt{2} \left(\frac{1}{4} - h_1 \right)$$

$$\alpha = \arccos 2\sqrt{2} \left(\frac{1}{4} - h_1 \right) \quad (34)$$

Despejando la ecuación (26) obtenemos

$$h_4 = \frac{1}{4} - \frac{1}{2\sqrt{2}} \sin \alpha$$

$$h_4 - \frac{1}{4} = -\frac{1}{2\sqrt{2}} \sin \alpha$$

Despejando a α se llega a

$$\sin \alpha = 2\sqrt{2} \left(\frac{1}{4} - h_4 \right)$$

$$\alpha = \arcsin 2\sqrt{2} \left(\frac{1}{4} - h_4 \right) \quad (35)$$

Dividiendo a (34) y (35) tenemos que:

$$\frac{\sin \alpha}{\cos \alpha} = \frac{2\sqrt{2} \left(\frac{1}{4} - h_4 \right)}{2\sqrt{2} \left(\frac{1}{4} - h_1 \right)}$$

Despejando a α

$$\tan \alpha = \frac{\left(\frac{1}{4} - h_4 \right)}{\left(\frac{1}{4} - h_1 \right)}$$

$$\alpha = \arctan \frac{\left(\frac{1}{4} - h_4 \right)}{\left(\frac{1}{4} - h_1 \right)} \quad (36)$$

Ecuaciones inversas para parametrizaciones de filtros de longitud 6

Por lo tanto las ecuaciones paramétricas inversas obtenidas son:

$$\beta = \arctan \left(\frac{h_3 - h_5}{h_0 - h_2} \right) \quad (30)$$

$$\alpha = \operatorname{atan} \frac{h_3 + h_5 - \frac{1}{16}}{h_0 + h_2 - \frac{1}{16}} \quad (33)$$

$$\alpha = \operatorname{acos} 2\sqrt{2} \left(\frac{1}{4} - h_1 \right) \quad (34)$$

$$\alpha = \operatorname{asin} 2\sqrt{2} \left(\frac{1}{4} - h_4 \right) \quad (35)$$

$$\alpha = \operatorname{atan} \frac{\left(\frac{1}{4} - h_4 \right)}{\left(\frac{1}{4} - h_1 \right)} \quad (36)$$

Como mecanismo de reparación, los valores de α en el intervalo $[0, 2\pi)$ que se encontraron para calcular las respuestas al impulso fueron los siguientes:

$$\alpha = t, \quad \text{con } t \text{ en el intervalo } [0, 2\pi) \quad (37)$$

$$\alpha = \pi + |t| \quad (38)$$

$$\alpha = \pi - |t| \quad (39)$$

$$\alpha = 2\pi - |t| \quad (40)$$

$$\alpha = \frac{3}{2}\pi - \left| t - \frac{1}{2}\pi \right| \quad (41)$$

Desarrollo del proyecto

Se calcularon las ecuaciones paramétricas inversas para filtros de longitud 4 dependientes de un solo parámetro α , de donde se obtuvieron sus respectivos valores de α .

Una vez obtenidas las ecuaciones paramétricas inversas para filtros de longitud 4 se implementó un módulo que generara números aleatorios para α .

Posteriormente se implementó un programa que al utilizar estos números aleatorios y dado un filtro se aplicaron las ecuaciones paramétricas inversas para calcular las respuestas al impulso de filtros de longitud 4.

Por último se implementó un programa que calcula el error promedio entre un filtro dado y uno generado con los parámetros de las ecuaciones paramétricas inversas, y así obtener la ecuación paramétrica que devolviera el valor de error mínimo.

Para los filtros de longitud 6 se calcularon el mayor número de ecuaciones paramétricas inversas dependientes de dos parámetros α y β .

Una vez obtenidas las ecuaciones paramétricas inversas para filtros de longitud 6 se implemento un módulo que generara números aleatorios para α y β . Posteriormente se implemento un programa que al utilizar estos números aleatorios y dado un filtro se aplicaron las ecuaciones paramétricas inversas para calcular las respuestas al impulso de filtros de longitud 6.

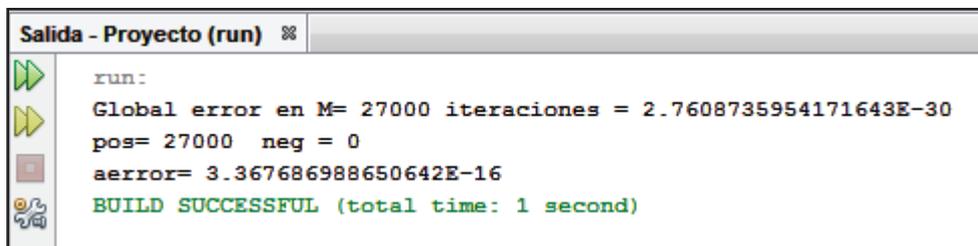
Por último se implemento un programa que calcula el error promedio entre un filtro dado de longitud 6 y uno generado con los parámetros de las ecuaciones paramétricas inversas, y así obtener la ecuación paramétrica que devolviera los valores de parámetros con error mínimo.

Resultados

Resultados obtenidos para filtros de longitud 4

Experimento 1.

El siguiente resultado se obtuvo con la ecuación paramétrica $h_0 = \frac{1}{4} + \frac{1}{2\sqrt{2}} \cos \alpha$ y su respectiva inversa $\alpha = \arccos(2\sqrt{2}(h_0 - \frac{1}{4}))$.



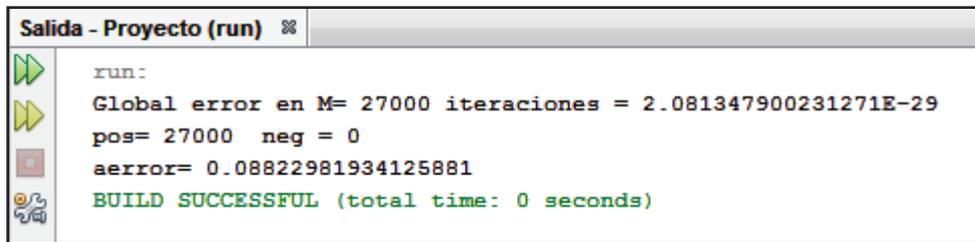
```
run:
Global error en M= 27000 iteraciones = 2.7608735954171643E-30
pos= 27000 neg = 0
aerror= 3.367686988650642E-16
BUILD SUCCESSFUL (total time: 1 second)
```

El “Global error” indica el promedio de los errores de aproximación entre el parámetro α de referencia y el calculado con las ecuaciones inversas, para 27,000 iteraciones. De esas iteraciones, en todos los casos se obtuvieron soluciones de parámetros en el intervalo deseado $[0, 2\pi)$ y el “Global error” es muy cercano a cero.

La variable aerror mide el error entre los valores de las respuestas al impulso (respuesta al impulso propuesta y respuesta al impulso calculada con los valores de parámetros alfa). Como puede verse es cercano a cero.

Experimento 2.

El siguiente resultado se obtuvo con la ecuación paramétrica $h_1 = \frac{1}{4} + \frac{1}{2\sqrt{2}} \operatorname{sen} \alpha$ y su respectiva inversa $\alpha = \operatorname{asin}(2\sqrt{2}(h_1 - \frac{1}{4}))$.



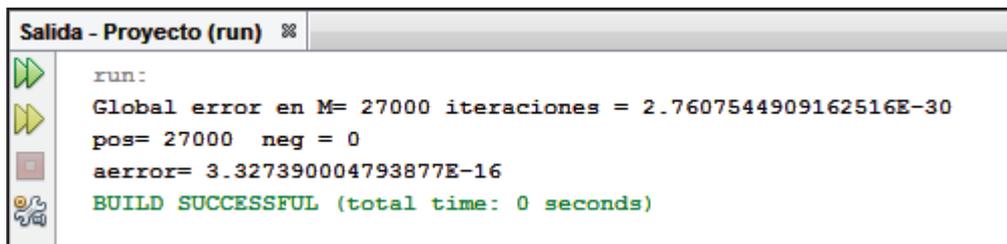
```
run:
Global error en M= 27000 iteraciones = 2.081347900231271E-29
pos= 27000 neg = 0
aerror= 0.08822981934125881
BUILD SUCCESSFUL (total time: 0 seconds)
```

El “Global error” indica el promedio de los errores de aproximación para las 27,000 iteraciones. De esas iteraciones, en todos los casos se obtuvieron soluciones de parámetros en el intervalo deseado $[0, 2\pi)$ y el “Global error” es muy cercano a cero.

La variable aerror mide el error entre los valores de las respuestas al impulso. Como puede verse, al igual que en el experimento 1, este valor es cercano a cero pero mayor que en el experimento 1.

Experimento 3.

El siguiente resultado se obtuvo con la ecuación paramétrica $h_2 = \frac{1}{4} - \frac{1}{2\sqrt{2}} \operatorname{cos} \alpha$ y su respectiva inversa $\alpha = \operatorname{acos}(2\sqrt{2}(\frac{1}{4} - h_2))$.



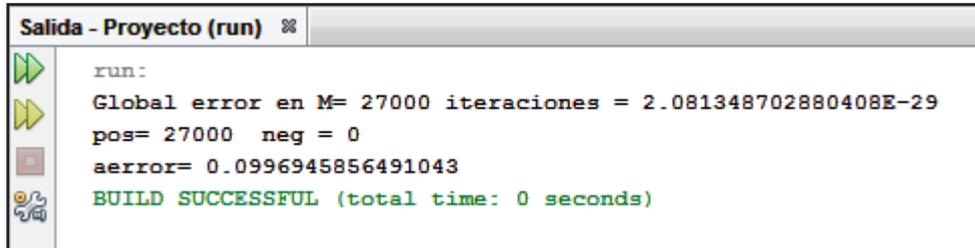
```
run:
Global error en M= 27000 iteraciones = 2.7607544909162516E-30
pos= 27000 neg = 0
aerror= 3.327390004793877E-16
BUILD SUCCESSFUL (total time: 0 seconds)
```

De esas iteraciones, en todos los casos se obtuvieron soluciones de parámetros en el intervalo deseado $[0, 2\pi)$ y el “Global error” es muy cercano a cero.

La variable aerror mide el error entre los valores de las respuestas al impulso. Como puede verse, al igual que en los experimentos 1, este valor es cercano a cero.

Experimento 4.

El siguiente resultado se obtuvo con la ecuación paramétrica $h_3 = \frac{1}{4} - \frac{1}{2\sqrt{2}} \text{sen } \alpha$ y su respectiva inversa $\alpha = \text{asin}(2\sqrt{2}(\frac{1}{4} - h_3))$.



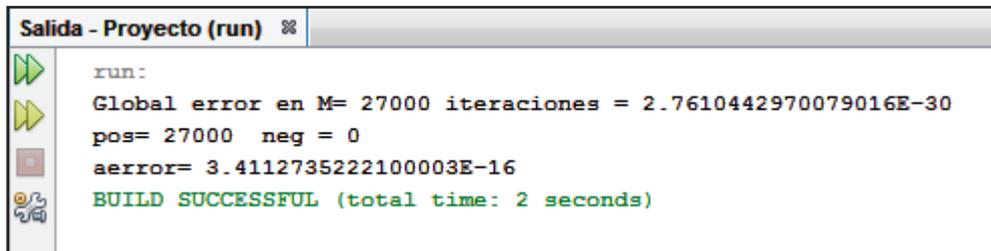
```
run:
Global error en M= 27000 iteraciones = 2.081348702880408E-29
pos= 27000 neg = 0
aerror= 0.0996945856491043
BUILD SUCCESSFUL (total time: 0 seconds)
```

De esas iteraciones, en todos los casos se obtuvieron soluciones de parámetros en el intervalo deseado $[0, 2\pi)$ y el “Global error” es muy cercano a cero.

La variable aerror mide el error entre los valores de las respuestas al impulso. Como puede verse, este valor es cercano a cero comparable con el experimento 2 y mayor que los experimentos 1, 2 y 3.

Experimento 5.

El siguiente resultado se obtuvo con la inversa $\alpha = \text{acos}(\sqrt{2}(h_0 - h_2))$.



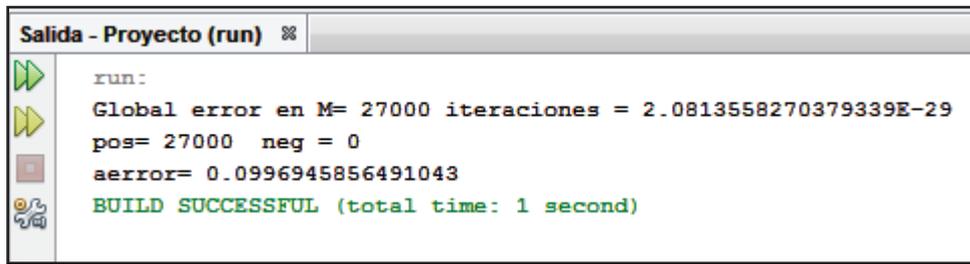
```
run:
Global error en M= 27000 iteraciones = 2.7610442970079016E-30
pos= 27000 neg = 0
aerror= 3.4112735222100003E-16
BUILD SUCCESSFUL (total time: 2 seconds)
```

De esas iteraciones, en todos los casos se obtuvieron soluciones de parámetros en el intervalo deseado $[0, 2\pi)$ y el “Global error” es muy cercano a cero.

La variable aerror mide el error entre los valores de las respuestas al impulso. Como puede verse, este valor es cercano a cero comparable con los experimentos 1 y 3.

Experimento 6.

El siguiente resultado se obtuvo con la inversa $\alpha = \text{asin}(\sqrt{2}(h_1 - h_3))$.



```

run:
Global error en M= 27000 iteraciones = 2.0813558270379339E-29
pos= 27000 neg = 0
aerror= 0.0996945856491043
BUILD SUCCESSFUL (total time: 1 second)

```

De esas iteraciones, en todos los casos se obtuvieron soluciones de parámetros en el intervalo deseado $[0, 2\pi)$ y el “Global error” es muy cercano a cero.

La variable aerror mide el error entre los valores de las respuestas al impulso. Como puede verse, este valor es comparable con los experimentos 2 y 4, y mayor que en los experimentos 1, 2 y 3.

En la siguiente tabla se muestran los resultados obtenidos para filtros de longitud 4.

Experimento	Ecuación paramétrica inversa	Error de Aproximación aerror	valores positivos	valores negativos
1	$\alpha = \text{acos}(2\sqrt{2}(h_0 - \frac{1}{4}))$	3.367686988650642 E-16	27000	0
2	$\alpha = \text{asin}(2\sqrt{2}(h_1 - \frac{1}{4}))$	0.088229819341258 81	27000	0
3	$\alpha = \text{acos}(2\sqrt{2}(\frac{1}{4} - h_2))$	3.327390004793877 E-16	27000	0
4	$\alpha = \text{asin}(2\sqrt{2}(\frac{1}{4} - h_3))$	0.099694585649104 3	27000	0
5	$\alpha = \text{acos}(\sqrt{2}(h_0 - h_2))$	3.411273522210000 3E-16	27000	0
6	$\alpha = \text{asin}(\sqrt{2}(h_1 - h_3))$	0.099694585649104 3	27000	0

Tabla 1. Resultados para filtros de longitud 4.

De la Tabla 1 se puede apreciar que:

Los peores casos se dan en los Experimentos 4 y 6, donde no se obtuvieron valores negativos y el error de aproximación es máximo respecto a las demás ecuaciones paramétricas inversas.

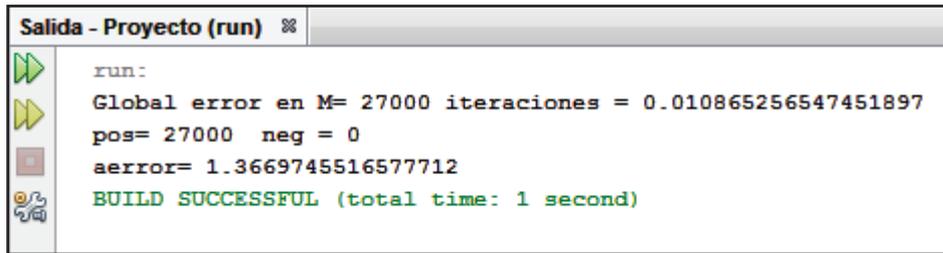
El mejor caso se da en el Experimento 3, donde no se obtuvieron valores negativos y el error de aproximación “aerror” es mínimo respecto a las demás ecuaciones paramétricas inversas.

Resultados obtenidos para filtros de longitud 6:

Experimento 7.

El siguiente resultado se obtuvo con el valor de las ecuaciones paramétricas inversas

$$\beta = \operatorname{atan} \left(\frac{h_3 - h_5}{h_0 - h_2} \right) \text{ y } \alpha = \operatorname{atan} \left(\frac{\frac{1}{4} - h_4}{\frac{1}{4} - h_1} \right).$$



```
Salida - Proyecto (run) ✖
run:
Global error en M= 27000 iteraciones = 0.010865256547451897
pos= 27000 neg = 0
aerror= 1.3669745516577712
BUILD SUCCESSFUL (total time: 1 second)
```

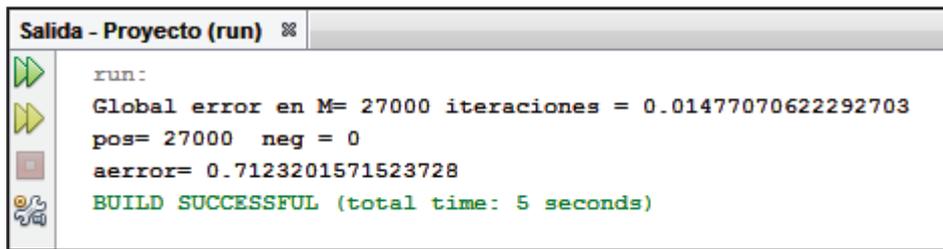
De esas iteraciones, en todos los casos se obtuvieron soluciones de parámetros en el intervalo deseado $[0, 2\pi)$ y el “Global error” es cercano a cero pero mayor que en los experimentos con filtros de longitud 4.

La variable aerror mide el error entre los valores de las respuestas al impulso. Como puede verse, este valor es cercano a cero pero es mayor al obtenido en todos los experimentos de filtros de longitud 4.

Experimento 8.

El siguiente resultado se obtuvo con el valor de las ecuaciones paramétricas inversas

$$\beta = \operatorname{atan} \left(\frac{h_3 - h_5}{h_0 - h_2} \right) \text{ y } \alpha = \operatorname{atan} \frac{h_3 + h_5 - \frac{1}{16}}{h_0 + h_2 - \frac{1}{16}}.$$



```
Salida - Proyecto (run) ✖
run:
Global error en M= 27000 iteraciones = 0.01477070622292703
pos= 27000 neg = 0
aerror= 0.7123201571523728
BUILD SUCCESSFUL (total time: 5 seconds)
```

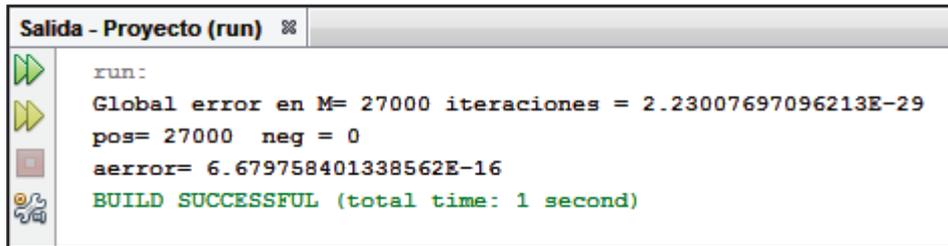
De esas iteraciones, en todos los casos se obtuvieron soluciones de parámetros en el intervalo deseado $[0, 2\pi)$ y el “Global error” es cercano a cero pero mayor que en los experimentos con filtros de longitud 4.

La variable aerror mide el error entre los valores de las respuestas al impulso. Como puede verse, este valor es cercano a cero, menor que en el Experimento 7, pero es mayor al obtenido en todos los experimentos de filtros de longitud 4.

Experimento 9.

El siguiente resultado se obtuvo con el valor de las ecuaciones paramétricas inversas

$$\beta = \operatorname{atan}\left(\frac{h_3 - h_5}{h_0 - h_2}\right) \text{ y } \alpha = \operatorname{acos} 2\sqrt{2}\left(\frac{1}{4} - h_1\right).$$



```
run:
Global error en M= 27000 iteraciones = 2.23007697096213E-29
pos= 27000 neg = 0
aerror= 6.679758401338562E-16
BUILD SUCCESSFUL (total time: 1 second)
```

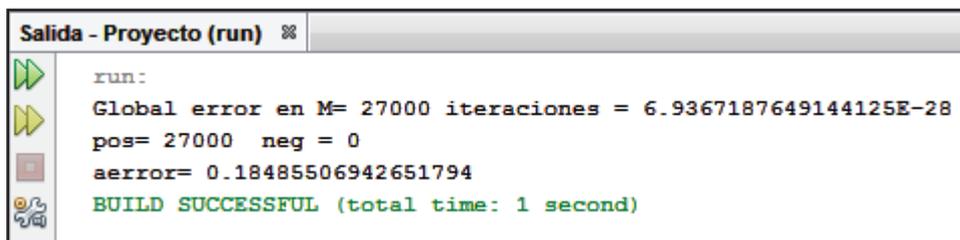
De esas iteraciones, en todos los casos se obtuvieron soluciones de parámetros en el intervalo deseado $[0, 2\pi)$ y el “Global error” es cercano a cero, y es comparable con los experimentos con filtros de longitud 4.

La variable aerror mide el error entre los valores de las respuestas al impulso. Como puede verse, este valor es cercano a cero comparable con experimentos de filtros de longitud 4.

Experimento 10.

El siguiente resultado se obtuvo con el valor de las ecuaciones paramétricas inversas

$$\beta = \operatorname{atan}\left(\frac{h_3 - h_5}{h_0 - h_2}\right) \text{ y } \alpha = \operatorname{asin} 2\sqrt{2}\left(\frac{1}{4} - h_4\right).$$



```
run:
Global error en M= 27000 iteraciones = 6.9367187649144125E-28
pos= 27000 neg = 0
aerror= 0.18485506942651794
BUILD SUCCESSFUL (total time: 1 second)
```

De esas iteraciones, en todos los casos se obtuvieron soluciones de parámetros en el intervalo deseado $[0, 2\pi)$ y el “Global error” es cercano a cero, y es comparable con los experimentos con filtros de longitud 4.

La variable aerror mide el error entre los valores de las respuestas al impulso. Como puede verse, este valor es cercano a cero y menor que los Experimentos 7 y 8, pero mayor que en el Experimento 9.

En la siguiente tabla se muestran los resultados obtenidos para filtros de longitud 6.

Experimento	Ecuación paramétrica inversa para α	Ecuación paramétrica inversa para β	Error de aproximación aerror	valores positivos	valores negativos
7	$\alpha = \operatorname{atan} \left(\frac{h_4 - \frac{1}{4}}{h_1 - \frac{1}{4}} \right)$	$\beta = \operatorname{atan} \left(\frac{h_3 - h_5}{h_0 - h_2} \right)$	1.366974551 6577712	27000	0
8	$\alpha = \operatorname{atan} \frac{h_3 + h_5 - \frac{1}{16}}{h_0 + h_2 - \frac{1}{16}}$	$\beta = \operatorname{atan} \left(\frac{h_3 - h_5}{h_0 - h_2} \right)$	0.712320157 1523728	27000	0
9	$\alpha = \operatorname{acos} 2\sqrt{2} \left(\frac{1}{4} - h_1 \right)$	$\beta = \operatorname{atan} \left(\frac{h_3 - h_5}{h_0 - h_2} \right)$	6.679758401 338562E-16	27000	0
10	$\alpha = \operatorname{asin} 2\sqrt{2} \left(\frac{1}{4} - h_4 \right)$	$\beta = \operatorname{atan} \left(\frac{h_3 - h_5}{h_0 - h_2} \right)$	0.184855069 42651794	27000	0

Tabla 2. Resultados para filtros de longitud 6.

El mejor valor de aerror se tiene en el Experimento 9, donde no se obtuvieron valores negativos y el error de aproximación es el mínimo obtenido entre las demás ecuaciones paramétricas.

Resultados obtenidos para filtros de longitud 4 al exceder el valor del umbral:

En estos experimentos se puso un umbral de aproximación entre los errores de las respuestas al impulso de referencia y los obtenidos con parámetros usando las ecuaciones inversas para filtros de longitud 4, de tal forma que si el umbral era excedido, se optaría por devolver también valores de parámetros sin ser “reparados” que dan lugar a valores negativos.

Experimentalmente se encontró que al hacer el umbral igual a $1e-25$, se obtuvieron con 27000 iteraciones lo siguiente:

Experimento 11.

El siguiente resultado se obtuvo con la ecuación paramétrica $h_1 = \frac{1}{4} + \frac{1}{2\sqrt{2}} \sin \alpha$ y su respectiva inversa $\alpha = \text{asin}(2\sqrt{2}(h_1 - \frac{1}{4}))$.

```
Salida - Proyecto (run) %
run:
Global error en M= 27000 iteraciones = 2.081347900231271E-29
pos= 26997 neg = 3
aerror= 0.08822981934125881
BUILD SUCCESSFUL (total time: 1 second)
```

Experimento 12.

El siguiente resultado se obtuvo con la ecuación paramétrica $h_3 = \frac{1}{4} - \frac{1}{2\sqrt{2}} \sin \alpha$ y su respectiva inversa $\alpha = \text{asin}(2\sqrt{2}(\frac{1}{4} - h_3))$.

```
Salida - Proyecto (run) %
run:
Global error en M= 27000 iteraciones = 2.081348702880408E-29
pos= 26997 neg = 3
aerror= 0.0996945856491043
BUILD SUCCESSFUL (total time: 2 seconds)
```

Experimento 13.

El siguiente resultado se obtuvo con la inversa $\alpha = \text{asin}(\sqrt{2}(h_1 - h_3))$.

```
Salida - Proyecto (run) %
run:
Global error en M= 27000 iteraciones = 2.0813558270379339E-29
pos= 26997 neg = 3
aerror= 0.0996945856491043
BUILD SUCCESSFUL (total time: 0 seconds)
```

Experimentalmente se encontró que al hacer el umbral igual a $1e-26$, se obtuvieron con 27000 iteraciones lo siguiente:

Experimento 14.

El siguiente resultado se obtuvo con la ecuación paramétrica $h_0 = \frac{1}{4} + \frac{1}{2\sqrt{2}} \cos \alpha$ y su respectiva inversa $\alpha = \arccos(2\sqrt{2}(h_0 - \frac{1}{4}))$.

```
Salida - Proyecto (run) ✖
run:
Global error en M= 27000 iteraciones = 2.7608735954171643E-30
pos= 26999 neg = 1
aerror= 3.367686988650642E-16
BUILD SUCCESSFUL (total time: 2 seconds)
```

Experimento 15.

El siguiente resultado se obtuvo con la ecuación paramétrica $h_2 = \frac{1}{4} - \frac{1}{2\sqrt{2}} \cos \alpha$ y su respectiva inversa $\alpha = \arccos(2\sqrt{2}(\frac{1}{4} - h_2))$.

```
Salida - Proyecto (run) ✖
run:
Global error en M= 27000 iteraciones = 2.7607544909162516E-30
pos= 26999 neg = 1
aerror= 3.327390004793877E-16
BUILD SUCCESSFUL (total time: 1 second)
```

Experimento 16.

El siguiente resultado se obtuvo con la inversa $\alpha = \arccos(\sqrt{2}(h_0 - h_2))$.

```
Salida - Proyecto (run) ✖
run:
Global error en M= 27000 iteraciones = 2.7610442970079016E-30
pos= 26999 neg = 1
aerror= 3.4112735222100003E-16
BUILD SUCCESSFUL (total time: 1 second)
```

En la siguiente tabla se muestran los resultados obtenidos para filtros de longitud 4.

Experimento	Ecuación paramétrica inversa	Valor del umbral	Casos con parámetros negativos	Casos con parámetros positivos
11	$\alpha = \text{acos} \left(2\sqrt{2} \left(h_0 - \frac{1}{4} \right) \right)$	1e-26	1	26999
12	$\alpha = \text{asin} \left(2\sqrt{2} \left(h_1 - \frac{1}{4} \right) \right)$	1e-25	3	26997
13	$\alpha = \text{acos} \left(2\sqrt{2} \left(\frac{1}{4} - h_2 \right) \right)$	1e-26	1	26999
14	$\alpha = \text{asin} \left(2\sqrt{2} \left(\frac{1}{4} - h_3 \right) \right)$	1e-25	3	26997
15	$\alpha = \text{acos} \left(\sqrt{2} \left(h_0 - h_2 \right) \right)$	1e-26	1	26999
16	$\alpha = \text{asin} \left(\sqrt{2} \left(h_1 - h_3 \right) \right)$	1e-25	3	26997

Tabla 3. Resultados para filtros de longitud 4.

Se puede observar que en los experimentos 11,13 y 15 solo hubo un caso de parámetros negativos con un valor del umbral de 1e-26, y para los experimentos 12,14 y 16 hubo 3 casos de parámetros negativos con un valor del umbral de 1e-25.

Resultados obtenidos para filtros de longitud 6 al exceder el valor del umbral:

En este experimento se puso un umbral de aproximación entre los errores entre las respuestas al impulso de referencia y los obtenidos con parámetros usando las ecuaciones inversas para filtros de longitud 6, de tal forma que si el umbral era excedido, se optaría por devolver también valores de parámetros sin ser “reparados” que dan lugar a valores negativos.

Experimentalmente se encontró que al hacer el umbral igual a 1e-2, se obtuvieron con 27000 iteraciones lo siguiente:

Experimento 17.

El siguiente resultado se obtuvo con el valor de las ecuaciones paramétricas inversas

$$\beta = \text{atan} \left(\frac{h_3 - h_5}{h_0 - h_2} \right) \quad \text{y} \quad \alpha = \text{atan} \left(\frac{\frac{1}{4} - h_4}{\frac{1}{4} - h_1} \right).$$

```

Salida - Proyecto (run) ✖
run:
Global error en M= 27000 iteraciones = 0.010865256547451897
pos= 17339 neg = 9661
aerror= 1.3669745516577712
BUILD SUCCESSFUL (total time: 1 second)
  
```

Experimento 18.

El siguiente resultado se obtuvo con el valor de las ecuaciones paramétricas inversas

$$\beta = \operatorname{atan}\left(\frac{h_3 - h_5}{h_0 - h_2}\right) \text{ y } \alpha = \operatorname{atan}\frac{h_3 + h_5 - \frac{1}{16}}{h_0 + h_2 - \frac{1}{16}}.$$

```
Salida - Proyecto (run) %
run:
Global error en M= 27000 iteraciones = 0.01477070622292703
pos= 10088 neg = 16912
aerror= 0.7123201571523728
BUILD SUCCESSFUL (total time: 2 seconds)
```

Experimentalmente se encontró que al hacer el umbral igual a $1e-23$, se obtuvieron con 27000 iteraciones lo siguiente:

Experimento 19.

El siguiente resultado se obtuvo con el valor de las ecuaciones paramétricas inversas

$$\beta = \operatorname{atan}\left(\frac{h_3 - h_5}{h_0 - h_2}\right) \text{ y } \alpha = \operatorname{asin} 2\sqrt{2}\left(\frac{1}{4} - h_4\right).$$

```
Salida - Proyecto (run) %
run:
Global error en M= 27000 iteraciones = 6.9367187649144125E-28
pos= 26999 neg = 1
aerror= 0.18485506942651794
BUILD SUCCESSFUL (total time: 4 seconds)
```

Experimentalmente se encontró que al hacer el umbral igual a $1e-25$, se obtuvieron con 27000 iteraciones lo siguiente:

Experimento 20.

El siguiente resultado se obtuvo con el valor de las ecuaciones paramétricas inversas

$$\beta = \operatorname{atan}\left(\frac{h_3 - h_5}{h_0 - h_2}\right) \text{ y } \alpha = \operatorname{acos} 2\sqrt{2}\left(\frac{1}{4} - h_1\right).$$

```
Salida - Proyecto (run) %
run:
Global error en M= 27000 iteraciones = 2.23007697096213E-29
pos= 26999 neg = 1
aerror= 6.679758401338562E-16
BUILD SUCCESSFUL (total time: 2 seconds)
```

En la siguiente tabla se muestran los resultados obtenidos para filtros de longitud 6.

Experimento	Ecuación paramétrica inversa para α	Ecuación paramétrica inversa para β	Valor del umbral	Casos con parámetros negativos	Casos con parámetros positivos
17	$\alpha = \text{atan} \left(\frac{\left(\frac{1}{4} - h_4\right)}{\left(\frac{1}{4} - h_1\right)} \right)$	$\beta = \text{atan} \left(\frac{h_3 - h_5}{h_0 - h_2} \right)$	1e-2	9661	17339
18	$\alpha = \text{atan} \frac{h_3 + h_5 - \frac{1}{16}}{h_0 + h_2 - \frac{1}{16}}$	$\beta = \text{atan} \left(\frac{h_3 - h_5}{h_0 - h_2} \right)$	1e-2	16912	10088
19	$\alpha = \text{asin} 2\sqrt{2} \left(\frac{1}{4} - h_4 \right)$	$\beta = \text{atan} \left(\frac{h_3 - h_5}{h_0 - h_2} \right)$	1e-23	1	26999
20	$\alpha = \text{acos} 2\sqrt{2} \left(\frac{1}{4} - h_1 \right)$	$\beta = \text{atan} \left(\frac{h_3 - h_5}{h_0 - h_2} \right)$	1e-25	1	26999

Tabla 4. Resultados para filtros de longitud 6.

Se puede observar que en los experimentos 19 y 20 solo hubo un caso de parámetros negativos con un valor del umbral de 1e-23 y 1e-25, para el experimento 17 hubo 9661 casos de parámetros negativos y para el experimento 18 hubo 16912 casos de parámetros negativos con un valor del umbral de 1e-2.

Conclusiones

Se observó que sí es posible determinar tanto los valores de los parámetros, y de las respuestas al impulso, con una precisión aceptable al aplicar las ecuaciones paramétricas inversas.

También se observó que tanto en los filtros de longitud 4 y longitud 6 no se obtuvieron valores negativos.

El “mecanismo de reparación” propuesto dio como resultado las ecuaciones paramétricas inversas que responden el problema dado un filtro de reconstrucción perfecta ¿qué parámetros le corresponden?

Existen casos en los que el programa implementado genera soluciones negativas para los parámetros α para el caso de filtros de longitud 4, y α y β para filtros de longitud 6 que se obtienen al fijar un umbral de aproximación entre las respuestas al impulso de referencia y las calculadas con los valores de los parámetros usando las ecuaciones inversas. Si el umbral se incrementa (mayor holgura) en todos los casos se pueden ofrecer soluciones positivas en el intervalo $[0, 2\pi)$.

Bibliografía

- [1] H. L. Resnikoff y R. O. N. Wells, *Wavelet Analysis: The Scalable Structure of Information : with 92 Figures*. Springer New York, 1998.
- [2] H. L. Resnikoff y R. O. N. Wells, *Wavelet Analysis: The Scalable Structure of Information : with 92 Figures*. Springer New York, 1998.
- [3] “Coiflet-Wikipedia, the free encyclopedia”. [En línea]. Disponible en: <http://en.wikipedia.org/wiki/Coiflet>. [Accedido: 22-jul-2015].
- [4] “Wavelet”. [En línea]. Disponible en: <http://www.mathworks.com/help/wavelet/ref/symaux.html>. [Accedido: 22-jul-2015].
- [5] Daubechies, I: *Ten Lectures on Wavelets*, Society for Industrial and Applied Mathematics, 1992.
- [6] G. Regensburger, “Parametrizing compactly supported orthonormal wavelets by discrete moments”, *AAECC*, vol. 18, no. 6, pp. 583-601, dic. 2007.
- [7] HehongZou; Tewfik, A.H., "Parametrization of compactly supported orthonormal wavelets," *Signal Processing, IEEE Transactions on*, vol.41, no.3, pp.1428, 1431, Mar 1993
- [8] Hereford, J.M., Roach, D.W., Pigford, R. Image compression using parameterized wavelets with feedback. pp. 267–277. SPIE (2003)
- [9] Lai, M.J. and D. W. Roach, “Parameterizations of univariate orthogonal wavelets with short support”, *Approximation Theory X*, pp. 369—384, *Innovations in Applied Mathematics*, Vanderbilt Univ. Press, 2002.
- [10] O. Herrera Alcántara y M. González Mendoza, “Optimization of Parameterized Compactly Supported Orthogonal Wavelets for Data Compression”, en *Advances in Soft Computing*, vol. 7095, I. Batyrshin y G. Sidorov, Eds. Springer Berlin Heidelberg, 2011, pp. 510-521.
- [11] O. Herrera Alcántara, “On the Best Evolutionary Wavelet Based Filter to Compress a Specific Signal”, en *Advances in Soft Computing*, vol. 6438, G. Sidorov, A. Hernández Aguirre, y C. Reyes García, Eds. Springer Berlin Heidelberg, 2010, pp. 394-405.
- [12] Oscar Herrera y Roman Mora, “Aplicación de Algoritmos Genéticos a la Compresión de Imágenes con Evolets”, en *Avances Recientes en Sistemas Inteligentes*, Miguel González Mendoza y Oscar Herrera Alcántara, SMIA, 2011, pp. 157 - 166.
- [13] Lai, M.J. and D. W. Roach, “Parameterizations of univariate orthogonal wavelets with short support”, *Approximation Theory X*, pp. 369-384, *Innovations in Applied Mathematics*, Vanderbilt Univ. Press, 2002.

Entregables

Clase Longitud4.java

Se implemento esta clase para realizar los experimentos deseados para parámetros de filtros de longitud 4.

```
package proyecto_14;

import static java.lang.Math.PI;
import static java.lang.Math.abs;
import static java.lang.Math.acos;
import static java.lang.Math.asin;
import static java.lang.Math.cos;
import static java.lang.Math.pow;
import static java.lang.Math.sin;
import static java.lang.Math.sqrt;

public class Longitud4 {

    public static void main(String[] args) {

        Demo d = new Demo();

    }
}

class Demo {
    int pos =0;
    int neg=0;
    double aerror=0;
    double eps = 1e-22; //valor del umbral dado
    int M = 27000;
    public Demo() {
        double gerror = 0;
        MyMath m = new MyMath();

        for (int i = 0; i < M; i++) {
            gerror += Run(m );
        }
        gerror /= M;
        aerror /= M;
        System.out.println("Global error en M= " + M + " iteraciones = " +
gerror);
        System.out.println("pos= " + pos + " neg = " + neg);
        System.out.println("aerror= " + aerror);
    }

    public double Run(MyMath m ) {
        double alfa;
        //generación de numeros aleatorios para alfa
        alfa = m.random() * 2 * Math.PI;
    }
}
```

```

double sq = 1 / (2 * Math.sqrt(2));
double[] h = new double[4];

//Ecuaciones paramétricas para filtros de longitud 4
h[0] = 0.125 + sq * cos(alfa);
h[1] = 0.125 + sq * sin(alfa);
h[2] = 0.125 - sq * cos(alfa);
h[3] = 0.125 - sq * sin(alfa);

//Ecuaciones parametricas inversas

//double alfap = acos(2 * sqrt(2) * (h[0] - 0.125));
//double alfap = asin(2 * sqrt(2) * (h[1] - 0.125));
//double alfap = acos(2 * sqrt(2) * (0.125 -h[2]));
//double alfap = asin(2 * sqrt(2) * (0.125 -h[3]));
//double alfap = acos(sqrt(2) * (h[0] - h[2]));
double alfap = asin(sqrt(2) * (h[1] -h[3]));

double[] hp = new double[4];
double t = alfap;
Ob[] ax = new Ob[5];
for (int i = 0; i < ax.length; i++) {
    ax[i] = new Ob();
}
for (int i = 0; i < ax.length; i++) {
    if (i == 0) {
        alfap = t;
    } else if (i == 1) {
        alfap = PI + abs(t);
    } else if (i == 2) {
        alfap = PI - abs(t);
    } else if (i == 3) {
        alfap = 2 * PI - abs(t);
    } else if (i == 4) {
        alfap = 1.5 * PI - abs(t - PI / 2);
    }
    hp[0] = 0.125 + sq * cos(alfap);
    hp[1] = 0.125 + sq * sin(alfap);
    hp[2] = 0.125 - sq * cos(alfap);
    hp[3] = 0.125 - sq * sin(alfap);
    double error = 0;
    for (int k = 0; k < 4; k++) {
        error += pow(hp[k] - h[k], 2);
    }
    error /= 4;
    ax[i].alfa = alfap;
    ax[i].error = error;
}
double errorT = ax[0].error;
double alfaT = 0;

for (int i = 0; i < ax.length; i++) {
    if (ax[i].alfa < 0) {
        continue;
    } else {
        if (errorT >= ax[i].error) {
            errorT = ax[i].error;
            alfaT = ax[i].alfa;
        }
    }
}

```

```

    }
}
if (errorT > eps) {
    for (int i = 0; i < ax.length; i++) {
        if (ax[i].alfa >= 0) {
            continue;
        } else {
            if (errorT > ax[i].error) {
                errorT = ax[i].error;
                alfaT = ax[i].alfa;
            }
        }
    }
    //Soluciones negativas
    neg++;
} else {

    //Soluciones positivas
    pos++;

}

for (int i = 0; i < ax.length; i++) {
    ax[i] = null;
}
ax = null;
aerror += abs(alfaT-alfa);
return errorT;
}
}

class Ob {

    double alfa;
    double error;

    public Ob() {
        alfa = 0;
        error = 0;
    }
}

```

Clase MyMath.java

Se implemento esta clase para poder generar números aleatorios para filtros de longitud 6

```

package proyecto_14;

import java.util.Random;

public class MyMath{
    Random r;
    public MyMath() {
        r = new Random(0);
        r.setSeed(0);
    }
    public double random() {
        return r.nextDouble();
    }
}

```

```
}  
}
```

Clase Longitud6.java

Se implemento esta clase para realizar los experimentos deseados para parámetros de filtros de longitud 6.

```
package proyecto_16;  
  
import static java.lang.Math.PI;  
import static java.lang.Math.abs;  
import static java.lang.Math.acos;  
import static java.lang.Math.asin;  
import static java.lang.Math.atan;  
import static java.lang.Math.cos;  
import static java.lang.Math.pow;  
import static java.lang.Math.sin;  
import static java.lang.Math.sqrt;  
  
public class Longitud6 {  
  
    /**  
     * @param args the command line arguments  
     */  
    public static void main(String[] args) {  
        Demo d = new Demo();  
    }  
  
}  
  
class Demo {  
  
    int pos = 0;  
    int neg = 0;  
    double aerror = 0;  
    double eps = 1e-1; //valor del umbral dado  
  
    int M = 27000;  
  
    public Demo() {  
        double gerror = 0;  
        MyMath m = new MyMath();  
        for (int i = 0; i < M; i++) {  
            gerror += Run(m);  
        }  
        gerror /= M;  
        aerror /= M;  
        System.out.println("Global error en M= " + M + " iteraciones = " +  
gerror);  
        System.out.println("pos= " + pos + " neg = " + neg);  
        System.out.println("aerror= " + aerror);  
  
    }  
  
    public double Run(MyMath m) {  
        double alfa;  
        double beta;
```

```

//generación de números aleatorios para alfa
alfa = m.random() * 2 * Math.PI;
//generación de números aleatorios beta
beta = m.random() * 2 * Math.PI;

double sq = 1 / (2 * Math.sqrt(2));
double p = 1 / 2 * sqrt(1 + sin(alfa + Math.PI * 0.25));
double[] h = new double[6];

//ecuaciones parametricas para filtros de longitud 6

h[0] = 0.125 + sq / 2 * cos(alfa) + p / 2 * cos(beta);
h[1] = 0.25 - sq * cos(alfa);
h[2] = 0.125 + sq / 2 * cos(alfa) - p / 2 * cos(beta);
h[3] = 0.125 + sq / 2 * sin(alfa) + p / 2 * sin(beta);
h[4] = 0.25 - sq * sin(alfa);
h[5] = 0.125 + sq / 2 * sin(alfa) - p / 2 * sin(beta);

//ecuaciones parametricas inversas

double betap = atan(h[3] - h[5] / h[0] - h[2]);
double alfap = atan(0.25 - h[4] / 0.25 - h[1]);
//double alfap= atan (h[3]+ h[5] - 0.0625/h[0]+h[2] - 0.0625);
//double alfap=acos(2 * Math.sqrt (2)* (0.25 - h[1]));
//double alfap=asin(2 * Math.sqrt (2)* (0.25 - h[4]));

double[] hp = new double[6];
double t = alfap;
double tb = betap;
Ob[] ax = new Ob[5];

for (int i = 0; i < ax.length; i++) {
    ax[i] = new Ob();
}
for (int i = 0; i < ax.length; i++) {
    if (i == 0) {
        alfap = t;
        for (int j = 0; j < ax.length; j++) {
            if (j == 0) {
                betap = tb;

            } else if (j == 1) {
                betap = PI + abs(tb);
            } else if (j == 2) {

                betap = PI - abs(tb);
            } else if (j == 3) {

                betap = 2 * PI - abs(tb);
            } else if (j == 4) {
                betap = 1.5 * PI - abs(tb - PI / 2);
            }
        }
    } else if (i == 1) {
        alfap = PI + abs(t);
        for (int j = 0; j < ax.length; j++) {
            if (j == 0) {
                betap = tb;
            }
        }
    }
}

```

```

    } else if (j == 1) {
        betap =  $PI$  + abs(tb);
    } else if (j == 2) {

        betap =  $PI$  - abs(tb);
    } else if (j == 3) {

        betap = 2 *  $PI$  - abs(tb);
    } else if (j == 4) {
        betap = 1.5 *  $PI$  - abs(tb -  $PI$  / 2);
    }
}
} else if (i == 2) {
    alfap =  $PI$  - abs(t);
    for (int j = 0; j < ax.length; j++) {
        if (j == 0) {
            betap = tb;

        } else if (j == 1) {
            betap =  $PI$  + abs(tb);
        } else if (j == 2) {

            betap =  $PI$  - abs(tb);
        } else if (j == 3) {

            betap = 2 *  $PI$  - abs(tb);
        } else if (j == 4) {
            betap = 1.5 *  $PI$  - abs(tb -  $PI$  / 2);
        }
    }
}
} else if (i == 3) {
    alfap = 2 *  $PI$  - abs(t);
    for (int j = 0; j < ax.length; j++) {
        if (j == 0) {
            betap = tb;

        } else if (j == 1) {
            betap =  $PI$  + abs(tb);
        } else if (j == 2) {

            betap =  $PI$  - abs(tb);
        } else if (j == 3) {

            betap = 2 *  $PI$  - abs(tb);
        } else if (j == 4) {
            betap = 1.5 *  $PI$  - abs(tb -  $PI$  / 2);
        }
    }
}
} else if (i == 4) {
    alfap = 1.5 *  $PI$  - abs(t -  $PI$  / 2);
    for (int j = 0; j < ax.length; j++) {
        if (j == 0) {
            betap = tb;

        } else if (j == 1) {
            betap =  $PI$  + abs(tb);
        } else if (j == 2) {

            betap =  $PI$  - abs(tb);
        } else if (j == 3) {

```

```

        betap = 2 * PI - abs(tb);
    } else if (j == 4) {
        betap = 1.5 * PI - abs(tb - PI / 2);
    }
}
}
hp[0] = 0.125 + sq / 2 * cos(alfap) + p / 2 * cos(betap);
hp[1] = 0.25 - sq * cos(alfap);
hp[2] = 0.125 + sq / 2 * cos(alfap) - p / 2 * cos(betap);
hp[3] = 0.125 + sq / 2 * sin(alfap) + p / 2 * sin(betap);
hp[4] = 0.25 - sq * sin(alfap);
hp[5] = 0.125 + sq / 2 * sin(alfap) - p / 2 * sin(betap);

double error = 0;
for (int k = 0; k < 6; k++) {
    error += pow(hp[k] - h[k], 2);
}
error /= 6;
ax[i].alfa = alfap;
ax[i].error = error;
}
double errorT = ax[0].error;
double alfaT = 0;
double betaT = 0;

for (int i = 0; i < ax.length; i++) {
    if (ax[i].alfa < 0) {
        continue;
    } else {
        if (errorT >= ax[i].error) {
            errorT = ax[i].error;
            alfaT = ax[i].alfa;
        }
    }
}
}
if (errorT > eps) {
    for (int i = 0; i < ax.length; i++) {
        if (ax[i].alfa >= 0) {
            continue;
        } else {
            if (errorT > ax[i].error) {
                errorT = ax[i].error;
                alfaT = ax[i].alfa;
            }
        }
    }
}

//Soluciones negativas
    neg++;
} else {

//Soluciones positivas

    pos++;

}

for (int i = 0; i < ax.length; i++) {
    ax[i] = null;
}
}

```

```

        ax = null;
        aerror += abs(alfaT - alfa);
        return errorT;
    }
}

class Ob {

    double alfa;
    double error;

    public Ob() {
        alfa = 0;
        error = 0;
    }
}

```

Clase MyMath.java

Se implemento esta clase para poder generar números aleatorios para filtros de longitud 6.

```

package proyecto_16;

import java.util.Random;

public class MyMath{
    Random r;
    public MyMath(){
        r = new Random(0);
        r.setSeed(0);
    }
    public double random(){
        return r.nextDouble();
    }
}

```