

SEGUNDA PARTE

SEGUNDA PARTE

1 DISEÑO DE FILTROS FIR 2-D USANDO LA NDFT.....	1-1
1.1 INTRODUCCIÓN.....	1-1
1.2 MÉTODOS EXISTENTES DE MUESTREO EN FRECUENCIA PARA EL CASO 2-D	1-2
1.3 DISEÑO DE MUESTREO NO UNIFORME PROPUESTO	1-4
1.4 DISEÑO DE UN FILTRO CUADRADO MEDIANTE LA APLICACIÓN DE LA NDFT	1-5
1.4.1 <i>Ejemplo de diseño</i>	1-8
1.5 DISEÑO DE UN FILTRO CIRCULAR MEDIANTE LA APLICACIÓN DE LA NDFT	1-19
1.5.1 <i>Ejemplo de diseño</i>	1-20
1.6 DISEÑO DE UN FILTRO DIAMANTE MEDIANTE LA APLICACIÓN DE LA NDFT	1-28
1.6.1 <i>Ejemplo de diseño</i>	1-32
1.6.2 <i>Ejemplo: Diseño de un filtro ‘fan’ (abanico)</i>	1-39
1.7 APLICACIONES DE LOS FILTROS BIDIMENSIONALES.....	1-43
1.8 RESUMEN	1-43
 2 DECODIFICACIÓN DE SEÑALES DTMF MEDIANTE EL USO DE LA NDFT 2-1	
2.1 INTRODUCCIÓN.....	2-1
2.2 ANTECEDENTES	2-2
2.2.1 <i>Estándares DTMF</i>	2-2
2.2.2 <i>Implementación digital</i>	2-3
2.2.3 <i>Algoritmos existentes para la decodificación de señales DTMF</i>	2-5
2.3 ALGORITMO DE DECODIFICACIÓN DE SEÑALES DTMF USANDO LA SB-NDFT.....	2-6
2.4 RESULTADOS Y COMPARACIONES.....	2-10
2.4.1 <i>Figura de mérito</i>	2-10
2.4.2 <i>Factores que afectan al rendimiento de los algoritmos de detección</i>	2-11
2.4.3 <i>Comparación del rendimiento</i>	2-15
2.5 RESUMEN	2-20
2.6 APÉNDICE: PROGRAMAS REALIZADOS EN MATLAB PARA ESTE CAPÍTULO.....	2-20
 3 CONCLUSIONES.....	3-1

1 DISEÑO DE FILTROS FIR BIDIMENSIONALES USANDO LA NDFT

1.1 INTRODUCCIÓN

Los filtros digitales bidimensionales (2-D) tienen aplicación en diversas áreas como por ejemplo el procesamiento y codificación de imágenes, robótica, sismología, sonar, radar, astronomía, etc. En aplicaciones tales como el procesamiento de la imagen, la preservación de la información de la fase es importante. En este sentido, los filtros FIR ofrecen una ventaja significativa respecto de los filtros IIR, puesto que pueden ser fácilmente diseñados con características de fase lineal o fase cero. Además, se garantiza que son estables. En este capítulo, usaremos el concepto de la NDFT 2-D, definido y estudiado en la primera parte de este documento, para diseñar filtros FIR mediante un muestreo no uniforme en frecuencia de la respuesta en frecuencia deseada.

Existen cuatro métodos estándar para el diseño de filtros FIR bidimensionales, -enventanado, muestreo en frecuencia, transformación en frecuencia y métodos para el diseño óptimo de filtros -. En el método de enventanado, una ventana bidimensional se obtiene normalmente a partir de una ventana unidimensional mediante la utilización de técnicas separables o no separables. La respuesta impulsiva de longitud infinita de un filtro ideal bidimensional es multiplicada por esta ventana para obtener un filtro FIR bidimensional. Aunque sencillo, este método sufre de una falta de control sobre las especificaciones en el dominio frecuencial. Además, se pueden diseñar filtros únicamente con respuestas en frecuencia separables o circularmente simétricas. El método de muestreo en frecuencia es un método conceptualmente simple para diseñar filtros FIR bidimensionales. Incluye una variedad de métodos existentes que usan muestreo en frecuencia uniforme y no uniforme. El diseño de filtros mediante transformación en frecuencia no tiene equivalente en el caso unidimensional. Éste es un método atractivo y práctico de diseñar filtros bidimensionales mediante la aplicación de una función de transformación en frecuencia a un filtro unidimensional. Esta función controla la forma del contorno de la respuesta en frecuencia bidimensional. Asimismo, las características de amplitud del filtro unidimensional se preservan al filtro bidimensional. Estos tres métodos no son óptimos. Sin embargo, a diferencia del caso de los filtros unidimensionales, el desarrollo de un algoritmo práctico y eficiente para diseñar filtros óptimos bidimensional todavía constituye un área de investigación activa. El problema de diseñar un filtro óptimo bidimensional es mucho más complejo que para el caso unidimensional. El poderoso teorema de alternancia no es aplicable en el caso bidimensional y la solución minimax no es única. Consecuentemente, existen algoritmos iterativos como el algoritmo Remez que tienen una enorme carga computacional y que no siempre convergen a la solución óptima.

Los primeros esfuerzos en el muestreo en frecuencia no uniforme bidimensional han dado lugar a estructuras de muestreo restringidas que reducen la complejidad computacional o a métodos de mínimos cuadrados que garantizan una interpolación única. El método que vamos a desarrollar en este capítulo se basa en un muestreo en frecuencia generalizado, donde las muestras se sitúan en líneas de contorno que coinciden con la forma deseada de la banda de paso del filtro 2-D. El método propuesto produce filtros bidimensionales no separables con bandas de paso bastante buenas y con bajos picos de rizado. Se obtienen filtros de buena calidad incluso para pequeñas rejillas. Esto es extraordinariamente importante puesto que estos filtros son usados, principalmente, en aplicaciones de filtrados prácticas.

1.2 MÉTODOS EXISTENTES DE MUESTREO EN FRECUENCIA PARA EL CASO BIDIMENSIONAL

El método de muestreo en frecuencia es, probablemente, el menos estudiado entre los métodos tradicionales de diseño de filtros FIR bidimensionales. El método de muestreo en frecuencia uniforme fue el primero en aparecer. Este método implica muestrear la respuesta en frecuencia bidimensional del filtro ideal en los vértices de una rejilla cartesiana del plano frecuencial, y después, hallar la respuesta impulsiva correspondiente usando la DFT inversa de dos dimensiones. Este método es computacionalmente eficiente. Para el diseño de un filtro de tamaño $N \times N$, se requieren $O(N^3)$ operaciones si la inversa de la DFT es usada, o $O(N \log_2 N)$ operaciones si se aplica un algoritmo FFT (suponiendo que N es potencia de dos en este último caso). Sin embargo, si usamos este método para diseñar filtros que son muy exigentes, la estrechísima transición entre bandas provoca grandes rizados en la respuesta en frecuencia resultante. Esto mejora considerablemente si se introducen algunas muestras de transición. Calcular valores óptimos para estas muestras de transición requiere métodos de programación lineal que minimicen los picos de rizado. Esto supone una enorme carga computacional debido al elevado número de ecuaciones de restricción que se obtienen al muestrear la respuesta en frecuencia bidimensional sobre una rejilla densa. Es por esto por lo que, en la práctica, se usa una elección razonable de los valores de las muestras de transición, como por ejemplo, muestras mediante el uso de interpolación lineal. La desventaja inherente al muestreo en frecuencia uniforme es que no hay flexibilidad a la hora de elegir las muestras. Por ello, los filtros se desvían de la forma en la banda de paso deseada, especialmente para aquellos filtros de pequeñas dimensiones.

Se han hecho esfuerzos para mejorar este método mediante la colocación ‘no uniforme’ de las muestras. Curiosamente, el método de muestreo en frecuencia

generalizado es la base de algunos algoritmos iterativos usados para el diseño de filtros FIR óptimos bidimensionales. El método propuesto por Diamessis, Therrien, y Rozwood, se basa en una extensión de la interpolación de Newton al caso bidimensional. Con el fin de obtener una forma newtoniana bidimensional, se deben adoptar algunas restricciones topológicas en la rejilla. Este método particular usa una plantilla triangular para las muestras. Hay que resaltar que es difícil diseñar filtros con formas más complejas que la cuadrada usando una plantilla restringida para la localización de las muestras.

Angelides ha propuesto un método que sitúa las muestras en una rejilla rectangular no uniforme y usa un tipo de polinomios de Newton bidimensionales para diseñar filtros FIR de dos dimensiones. Esto reduce el problema del diseño 2-D a dos sistemas lineales 1-D. La complejidad computacional es $O(N^3)$.

En el método propuesto por Rozwood, Therrien y Lim, las localizaciones de las muestras están restringidas a situarse en líneas paralelas horizontales o verticales en el plano frecuencial bidimensional. Esto forma de muestreo reduce el sistema lineal general a pequeños conjuntos de ecuaciones más pequeños, reduciendo, así, la complejidad. Los autores dan dos ejemplos de diseño para filtros paso de baja y paso de banda circularmente simétricos en los que algunas muestras han sido situadas en los bordes de las bandas de paso y de rechazo para conseguir un mayor control sobre la forma del filtro. Estos filtros presentan unos bordes más circulares que aquellos filtros diseñados mediante el muestreo uniforme en frecuencia, pero, en cambio, picos de rizado mayores. Sin embargo, no existe ningún algoritmo eficiente que, automáticamente, coloque las muestras satisfaciendo las restricciones de este método, y diseñe un filtro de aceptable calidad. Por otro lado, este método tienen una complejidad computacional de $O(N^4)$.

Zakhor y Alvstad han desarrollado unas estrategias de muestreo no uniforme menos restrictivas que las citadas, en las cuales se restringe a las muestras a estar en curvas irreducibles del plano frecuencial bidimensional. Sus ejemplos de distribución de las muestras incluyen muestras polares y muestras en líneas rectas. Ello ha derivado a un teorema que proporciona las condiciones que se deben dar para que exista interpolación única. Sin embargo, en general, el número total de muestras requeridas para la interpolación única excede el número de coeficientes del filtro. Este método requiere de $O(N^6)$ operaciones. Una característica común de estos diseños es que pocas muestras están situadas en la intersección de las líneas de muestreo y los bordes de la banda de paso y banda de rechazo, para modelar los contornos de la forma deseada.

1.3 DISEÑO DE MUESTREO NO UNIFORME PROPUESTO

En el método de diseño de filtros FIR bidimensionales propuesto, la respuesta en frecuencia deseada es muestreada en N_i puntos no uniformemente localizados en el plano frecuencial bidimensional, siendo N_i el número de coeficientes independientes del filtro a diseñar. Todas las simetrías existentes en la respuesta impulsiva del filtro son utilizadas de tal manera que N_i es, generalmente, mucho más bajo que el número total de coeficientes del filtro a diseñar, N^2 . Esto reduce enormemente el tiempo de diseño, además de garantizar la existencia de simetría. El conjunto de N_i ecuaciones lineales, dadas por

$$\hat{X}(z_{1k}, z_{2k}) = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x[n_1, n_2] z_{1k}^{-n_1} z_{2k}^{-n_2},$$

tienen que ser resueltas para obtener los coeficientes independientes del filtro. El resto de coeficientes se hallará aplicando las simetrías existentes en la respuesta impulsiva deseada.

Para el diseño de filtros digitales bidimensionales mediante la aplicación de la NDFT debemos partir del diseño de filtros digitales unidimensionales mediante NDFT. El diseño de filtros para el caso unidimensional mediante aplicación de la NDFT se recoge en el PROYECTO FIN DE CARRERA, complementario a éste, del alumno JUAN ANTONIO MORÓN GARCÍA. Por tanto, haremos referencia a ese proyecto cuando algún paso para el diseño de filtros FIR bidimensionales necesite de lo ya explicado en el documento citado que, como se ha comentado, se encarga del diseño de filtros digitales FIR para el caso unidimensional.

En el caso del diseño de filtros unidimensionales, la elección de la localización de las muestras depende del tipo particular de filtro a diseñar. En general, el problema de colocar las muestras para el caso bidimensional es mucho más complicado que para el caso unidimensional. La experiencia en el diseño de filtros bidimensionales de distintas formas (cuadrado, circular, ...) indica que los mejores resultados se obtienen cuando las muestras se sitúan en líneas de contorno que coinciden con la forma de la banda de paso deseada. Por ejemplo, para diseñar un filtro cuadrado, colocaremos las muestras a lo largo de un conjunto de líneas de contorno cuadradas en el plano frecuencial bidimensional. Se debe advertir que estos resultados coinciden con los resultados de los métodos de diseño citados en el apartado anterior, donde un mejor control sobre la forma del filtro se obtenía mediante la colocación de las muestras en los bordes de la banda de paso y banda de rechazo. El número total de contornos y el número de muestras por contorno deben ser elegidos cuidadosamente con el fin de evitar singularidades. Se conoce una condición necesaria para no singularidad que ayuda como verificación aproximada. Sin embargo, esta condición no es suficiente para

garantizar no singularidad. Este teorema establece que si la suma de los grados de las curvas irreducibles, en las cuales las muestras están situadas, es pequeño comparado con el grado del polinomio del filtro, entonces el problema de interpolación es singular. Volviendo al ejemplo del diseño de un filtro cuadrado, es claro que el número de contornos cuadrados debe ser elegido apropiadamente con respecto al tamaño del filtro a diseñar.

Por tanto, vamos a colocar las muestras a lo largo de líneas de contorno que tendrán la forma del filtro deseado. Los parámetros que deben ser elegidos para el para el diseño del filtro son:

- ✓ El número de contornos y el espacio entre ellos.
- ✓ El número de muestras de cada contorno y el espaciado relativo entre ellas.
- ✓ Los valores de las muestras.

En los siguientes apartados, mostraremos cómo estos parámetros son elegidos para filtros con distintas formas, concretamente, nos centraremos en los filtros cuadrados, circulares, diamantes y fan (abanico). Una técnica común que se usa en el diseño de todos ellos es que una sección cruzada, concreta, de la respuesta en frecuencia bidimensional deseada es aproximada mediante funciones analíticas unidimensionales basadas en polinomios de Chebyshev, iguales a los explicados en el proyecto complementario a éste ya citado en párrafos anteriores. Las muestras serán colocadas en contornos que pasen a través de los extremos de esta sección cruzada. Todo esto quedará más claro cuando se vean los ejemplos de diseño de los apartados siguientes para algunas de las distintas formas de filtro bidimensional.

1.4 DISEÑO DE UN FILTRO CUADRADO MEDIANTE LA APLICACIÓN DE LA NDFT

Consideremos el diseño de un filtro paso de baja cuadrado $h[n_1, n_2]$, con unas especificaciones como las que se muestran en la figura 10. Sea el filtro de tamaño $N \times N$, con borde de la banda de paso situado en w_p y borde de la banda de rechazo en w_s tal y como se muestra en la mencionada figura 10.

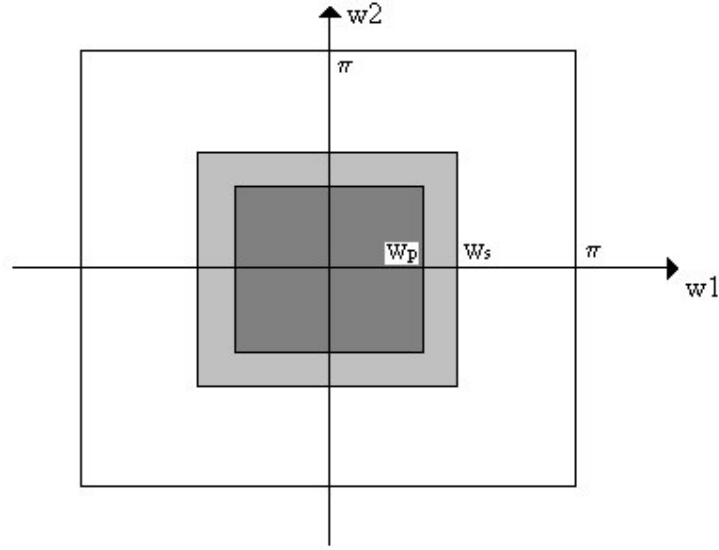


Figura 10. Especificaciones de un filtro cuadrado paso de baja. La región más oscura es la banda de paso, la región grisácea es la banda de transición y la región sin colorear es la banda de rechazo del filtro.

Consideramos que la respuesta en frecuencia, $H(e^{jw_1}, e^{jw_2})$, es de fase cero, con una respuesta en amplitud $A(w_1, w_2)$ que satisface la condición

$$A(w_1, w_2) = A^*(w_1, w_2).$$

Para una respuesta impulsiva real, $h[n_1, n_2]$, la ecuación anterior implica que $h[n_1, n_2]$ es simétrica con respecto al origen del plano espacial, por tanto:

$$h[n_1, n_2] = h[-n_1, -n_2].$$

Es claro que de la observación de la figura 10, la respuesta en amplitud exhibe una simetría del tipo

$$A(w_1, w_2) = A(-w_1, w_2) = A(w_1, -w_2).$$

En el dominio espacial, este hecho es equivalente a

$$h[n_1, n_2] = h[-n_1, n_2] = h[n_1, -n_2].$$

De todo ello, podemos expresar la respuesta en amplitud de la forma

$$A(w_1, w_2) = h[0,0] + \sum_{n_1=1}^{\frac{N-1}{2}} 2h[n_1,0] \cos w_1 n_1 + \sum_{n_2=1}^{\frac{N-1}{2}} 2h[0,n_2] \cos w_2 n_2 +$$

$$+ \sum_{n_1=1}^{N-1} \sum_{n_2=1}^{N-1} 4h[n_1, n_2] \cos w_1 n_1 \cos w_2 n_2 .$$

Así, el número de coeficientes independientes del filtro vendrá dado por la expresión

$$N_i = \frac{(N+1)^2}{4} .$$

Para hallar estos coeficientes, necesitamos N_i muestras de $A(w_1, w_2)$ localizadas en el primer cuadrante del plano (w_1, w_2) .

El método de diseño se basa en la siguiente idea. Si tomamos una sección cruzada de la respuesta en amplitud bidimensional a lo largo del eje w_1 (o del w_2), entonces tenemos la respuesta en frecuencia de un filtro paso de baja. Un ejemplo de una sección de este tipo se muestra en la figura 11.

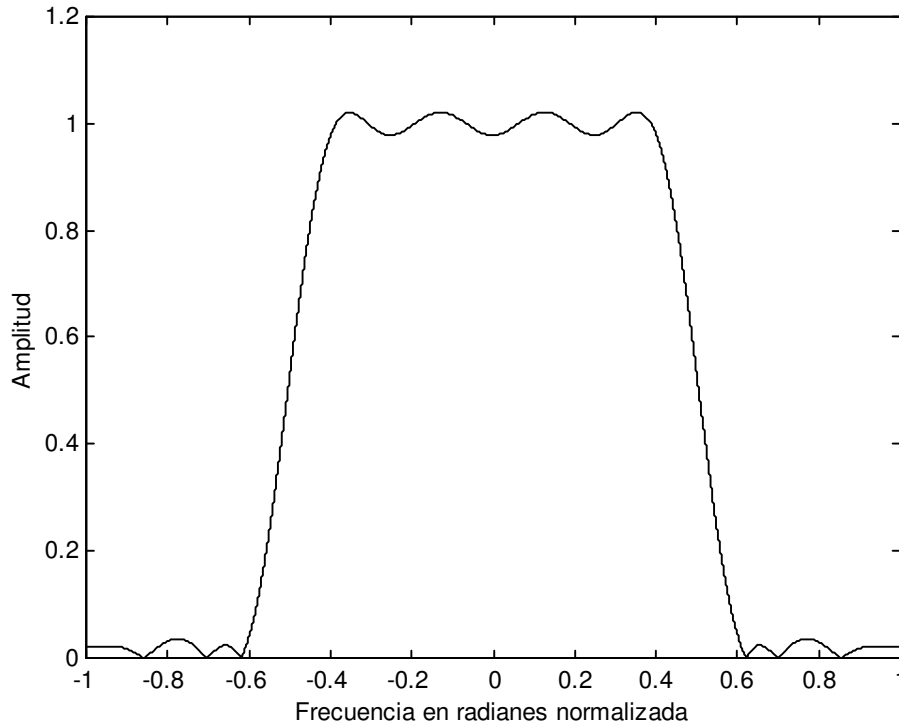


Figura 11. Sección cruzada de la amplitud del filtro cuadrado deseado a lo largo del eje w_1 .

Dadas las especificaciones de un filtro bidimensional tales como el tamaño de la rejilla y los bordes de las bandas, primero, debemos obtener la banda de paso y la banda

de rechazo de la sección cruzada (mostrada en la figura anterior) mediante la aplicación de funciones analíticas separadas, $H_p(w)$ y $H_s(w)$, tal y como se hace para el diseño de un filtro paso de baja unidimensional (esto se explica con detalle en el proyecto complementario a éste del alumno J. A. Morón García). Después, se colocan muestras a lo largo del eje w_1 , en los extremos de estas funciones $H_p(w)$ y $H_s(w)$. En el plano frecuencial bidimensional, las muestras se sitúan en contornos cuadrados que pasan por estos extremos. Se puede observar que todas las muestras situadas en un contorno cuadrado concreto tienen el mismo valor y están uniformemente espaciadas. El número total de contornos cuadrados es

$$\text{número de contornos cuadrados} = \frac{N+1}{2}.$$

El número muestras en el contorno k-ésimo empezando por el origen es

$$\text{número de muestras por contorno} = 2k - 1$$

con $k = 1, 2, \dots, N+1/2$. Estos números se han encontrado que funcionan bien, puesto que el número total de muestras, que viene dado por la suma de las $N+1/2$ series de números impares, es decir,

$$\sum_{k=1}^{N+1/2} (2k - 1) = \left(\frac{N+1}{2} \right)^2,$$

resulta igual al número de coeficientes independientes del filtro N_i .

Los coeficientes del filtro se hallan resolviendo las N_i ecuaciones obtenidas mediante el muestreo de la ecuación desarrollada de $A(w_1, w_2)$, citada anteriormente, en N_i puntos. El método de diseño expuesto se puede usar también para diseñar filtros cuadrados paso de alta. En ese caso, sólo es necesaria una modificación: la sección cruzada de la respuesta en amplitud bidimensional a lo largo del eje w_1 (o del eje w_2) tiene que ser aproximada por una respuesta unidimensional paso de alta.

1.4.1 EJEMPLO DE DISEÑO

Consideremos el diseño de un filtro cuadrado paso de baja con las siguientes especificaciones:

Tamaño de la rejilla = 9×9

Borde de la banda de paso, $w_p = 0.35\pi$

Borde de la banda de rechazo, $w_s = 0.65\pi$.

Mediante aplicación de las fórmulas expuestas anteriormente concluimos que, sólo 25 de los 81 coeficientes del filtro son independientes. Por tanto, necesitamos 25 muestras localizadas en el primer cuadrante del plano frecuencial bidimensional. De igual forma, el número total de contornos cuadrados es cinco. La sección cruzada de la respuesta en amplitud bidimensional a lo largo del eje w_1 se obtiene mediante funciones analíticas unidimensionales tal y como se explicó más arriba (ver proyecto complementario a éste). Una vez que tenemos la sección cruzada diseñada, nos quedamos con cinco muestras a lo largo del eje w_1 que se obtienen mediante el muestreo de las funciones analíticas unidimensionales en los puntos extremos de éstas. Por último, el resto de muestras se sitúan en cinco cuadrados consecutivos que pasan a través de estos extremos (en la figura 12 se mostrará con claridad dónde se han colocado exactamente las 25 muestras necesarias para el diseño del filtro). Estas muestras llevan a 25 ecuaciones lineales que, una vez resueltas, llevan a los 25 coeficientes independientes del filtro. La respuesta en amplitud y una gráfica que muestra en lugar donde hemos colocado las muestras así como las curvas de nivel del filtro se muestran a continuación. Claramente, se puede observar como la banda de paso tiene la forma cuadrada deseada.

Después de las gráficas con la característica del filtro y las curvas de nivel del mismo, se adjunta el programa de MATLAB diseñado para conseguir este filtro cuadrado bidimensional mediante la aplicación de la NDFT y en el que se ha hecho uso de los conceptos explicados en el proyecto complementario a éste para la obtención de la sección cruzada de la respuesta en amplitud bidimensional.

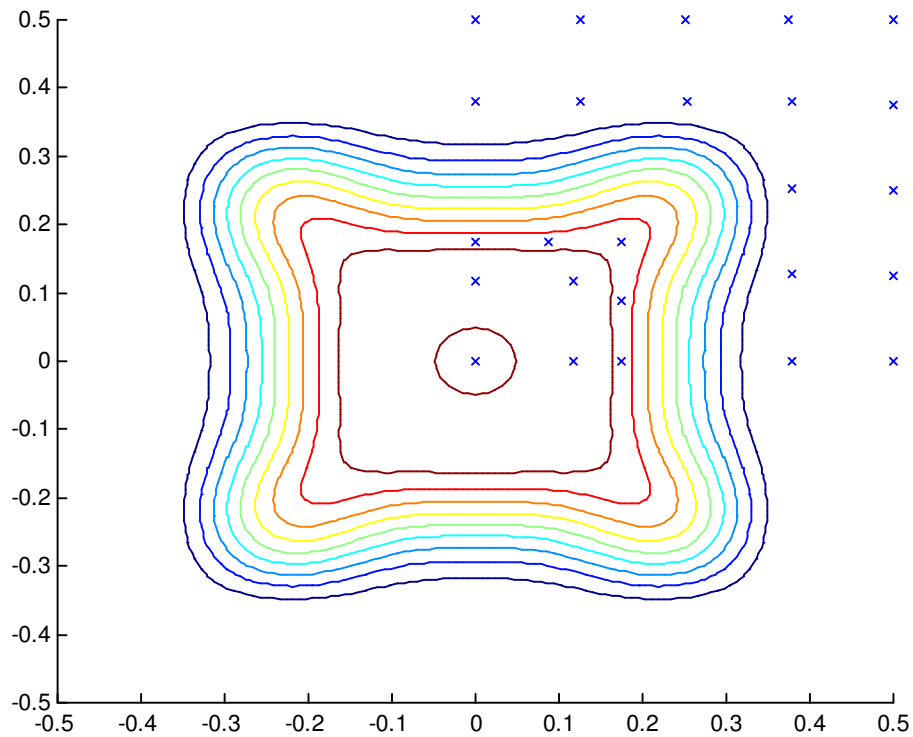


Figura 12. Filtro cuadrado de tamaño 9 x 9 diseñado mediante el método de la NDFT.

Gráfica que muestra las curvas de nivel del filtro.

Las muestras están colocadas en los lugares denotados por una 'x' en la gráfica.

Los ejes de frecuencias están normalizados por 2π .

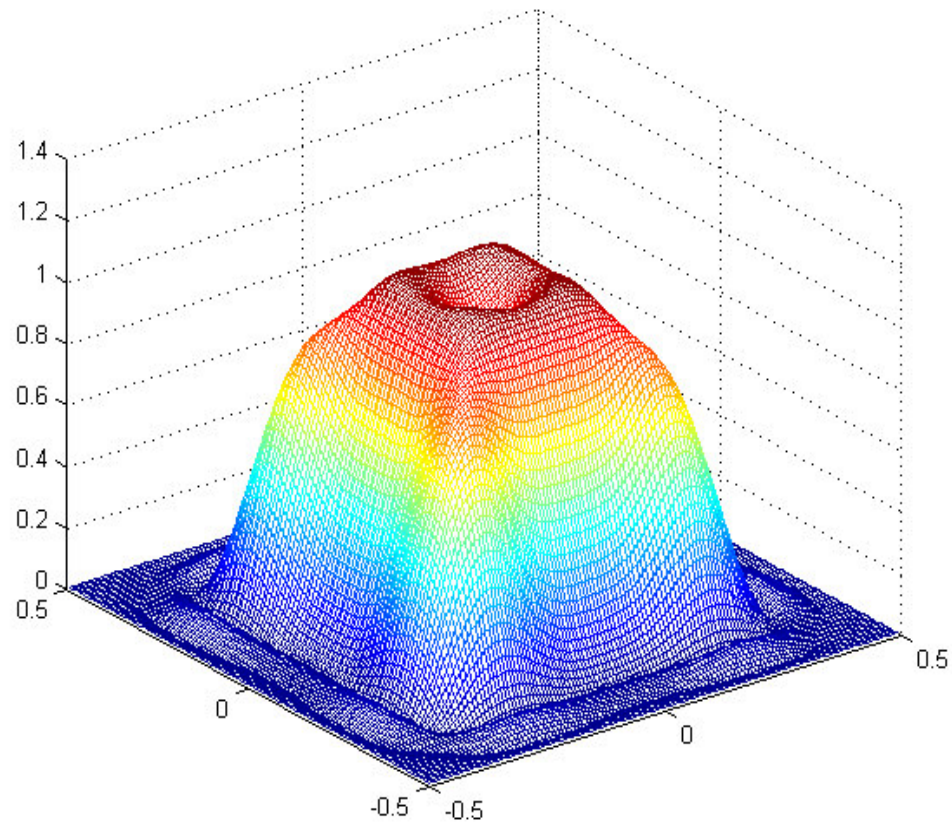


Figura 13. Filtro cuadrado de tamaño 9 x 9 diseñado mediante el método de la NDFT.
Dibujo de la respuesta en frecuencia del filtro.
Los ejes de frecuencias están normalizados por 2π .

CÓDIGO DEL PROGRAMA EN MATLAB PARA LA OBTENCIÓN DEL FILTRO CUADRADO MEDIANTE EL USO DE LA NDFT

```
% DISEÑO DE UN FILTRO CUADRADO MEDIANTE EL USO DE LA NON-UNIFORM DISCRETE
% FOURIER TRANSFORM

clear;

% Especificaciones

N1=9;           % Longitud del filtro en uno de los ejes
N2=9;           % Longitud del filtro en el otro eje
wp=0.35*pi;     % Frecuencia de corte de la banda de paso
ws=0.65*pi;     % Frecuencia de corte de la banda de rechazo
k=1;            % Relación entre los rizados de las bandas de paso y rechazo respect.

% DISEÑO DEL FILTRO CUADRADO, USAREMOS EL FILTRO LP DE TIPO 1 DE UNA DIMENSIÓN
% DESCRITO EN EL FICHERO NDFILPTIPO1.M ( PROYECTO COMPLEMENTARIO A ÉSTE ).

% Cálculo de parámetros necesarios para el diseño

Ni=(N1+1)^2/4;   % Número de coeficientes independientes del filtro
n_cont=(N1+1)/2; % Número de contornos de la rejilla de muestras

for k=1:(N1+1)/2
    n_mu_con(k)=2*k-1; % Vector que contiene el núm. de muestras de cada contorno
end

% DISEÑO DEL FILTRO LOW PASS TIPO I DE UNA DIMENSIÓN

% Cálculo de los valores P y S (órdenes de los polinomios de Chebyshev que servirán
% para generar las funciones Hp y Hs que son las respuestas en frecuencia en la banda
% de paso y en la de rechazo respectivamente)

x=(pi-ws)/wp;    % Factor de proporción entre las anchuras de las bandas
P=round((N1-1)/2/(1+x)); % Cálculo de P
S=(N1-1)/2-P;    % Cálculo de S

% Cálculo de los rizados en la banda de paso y banda de rechazo

dp=sqrt(k)*10^(-0.1162*(ws-wp)*(N1-1)-0.65);
ds=dp/k;

% Cálculo de los coeficientes necesarios para la generación de las funciones necesarias
% para el diseño del filtro

A=0.5*((cosh((1/P)*acosh((1+ds)/dp)))+1);
B=A-1;
C=0.5*((cosh((1/S)*acosh((1+dp)/ds)))+1);
D=C-1;
a=1/wp*acos((B-1)/A);
b=pi;
c=1/(pi-ws)*acos((D-1)/C);
d=pi*(1-c);

% A continuación, calculamos las frecuencias de muestreo óptimas según este tipo de
% diseño

for i=1:P
    wbp(i)=1/a*(acos((cos(i*pi/P)-B)/A)-b); % Frec. Muestreo en banda de paso
end
```

```

for i=1:S
    wbs(i)=1/c*(acos((cos(i*pi/S)-D)/C)-d); % Frec. Muestreo en banda de rechazo
end

% El vector que contiene las frecuencias de muestreo será tal y como se muestra
% a continuación. Hemos añadido una muestra más porque necesitamos (N+1)/2 muestras
% para tener determinados todos los coeficientes. También, debido a la simetría
% cambiamos el signo del vector wbp puesto que nos ha salido negativo como consecuencia
% del uso de funciones trigonométricas.

for i=1:P
    wbp(i)=-wbp(P-i+1);
end
wbp(P+1)=wp;

w=[wbp, wbs]; % VECTOR FINAL QUE CONTIENE LAS FRECUENCIAS DE MUESTREO

% Procedemos a la generación de A(w) tal y como se explica en la teoría

xpw=A*cos(a*wbp+b)+B;
xsw=C*cos(c*wbs+d)+D;

Apw=1-dp*cos(P*acos(xpw));
Asw=ds*cos(S*acos(xsw));

Aw=[Apw, Asw]; % VECTOR QUE CONTIENE EL VALOR EN MÓDULO DE LA RESPUESTA EN FRECUENCIA
% EN CADA UNA DE LAS FRECUENCIAS DE MUESTREO PREVIAMENTE CALCULADAS

% FIN DEL DISEÑO DEL FILTRO LP TIPO I DE UNA DIMENSIÓN

% CONSTRUCCIÓN DE LA REJILLA DE MUESTRAS

% Cálculo de las coordenadas donde se sitúan las muestras según este procedimiento.
% La hipermatriz de nombre 'matriz' contiene las coordenadas de las muestras para cada
% contorno.

matriz(:, :, 1)=[0 0]; % Coordenadas de la única muestra del primer contorno

% Cálculo de las coordenadas del resto de contornos (sólo aquellas pertenecientes a la
% mitad del primer cuadrante.

for i=2:n_cont
    for k=1:(n_mu_con(i)+1)/2
        matriz(k, :, i)=[(k-1)*w(i)/((n_mu_con(i)-1)/2) w(i)];
    end
end

% Completamos con las muestras simétricas respecto de la bisectriz del primer cuadrante.

for i=2:n_cont
    for k=1:(n_mu_con(i)+1)/2
        if (matriz(k, 1, i)~=matriz(k, 2, i))
            matriz(k+(n_mu_con(i)+1)/2, :, i)=[matriz(k, 2, i) matriz(k, 1, i)];
        end
    end
end

% Colocamos las muestras en un vector de tamaño Ni (número de muestras necesarias).

v_muestras(1, :)= [0 0]; % Colocamos la primera muestra que es siempre la misma
k=2; % Índice que recorre las hipermatrices
p=1; % Índice que recorre las filas de cada hipermatriz

for i=2:Ni
    if (matriz(p, :, k)~= [0 0]) % Si de la hipermatriz que extraemos sacamos
        k=k+1; % un valor [0 0] significa que ya está agotada
        p=1; % y, por tanto, debemos pasar a la siguiente
    end
    v_muestras(i, :)=matriz(p, :, k);
    p=p+1;
end

```

```

% Asignamos a cada muestra el valor de A(w) correspondiente, es decir, cada contorno
% de la rejilla tomará un único valor de A(w).

aux=0;
for i=1:n_cont
    for j=1:n_mu_con(i)
        Aw2d(j+aux)=Aw(i);      % Vector que contiene el valor de A(w) para cada muestra
    end
    aux=aux+n_mu_con(i);
end

% Cálculo de los coeficientes h(n1,n2) haciendo uso de la fórmula expuesta en la parte
% teórica.

for i=1:(N1-1)/2
    m1_aux(:,i)=2*cos(v_muestras(:,1)*i);
end

for i=1:(N1-1)/2
    m2_aux(:,i)=2*cos(v_muestras(:,2)*i);
end

m1_aux2=m1_aux/2;
m2_aux2=m2_aux/2;

for i=1:max(size(m1_aux2))
    for j=1:min(size(m1_aux2))
        m3_aux(i, ((j-1)*((N1-1)/2)+1):(j*((N1-1)/2)))=4*m1_aux2(i,j)*m2_aux2(i,:);
    end
end

% Construimos la matriz total D.
D=[ones(N1,1) m1_aux m2_aux m3_aux];

% Obtención de los Ni coeficientes independientes del filtro.
hn2d=inv(D)*Aw2d';

% Ordenamos hn2d matricialmente con el objeto de expresar los coeficientes independ.
% de una manera más apropiada para la obtención del resto de coeficientes simétricos.

for i=1:(N1+1)/2
    ind_aux=(i-1)*(N1-1)/2+(N1+3)/2;
    mat_hn((N1+1)/2-i+1,:)=hn2d(ind_aux:ind_aux+(N1-3)/2)';
end

mat_hn2d=[hn2d((N1+1)/2:-1:1) mat_hn];

% Ahora, aplicamos la simetría existente para hallar los coeficientes restantes

for i=1:length(mat_hn2d)-1
    mat_hn2d_inv(i,:)=mat_hn2d(length(mat_hn2d)-i,:);
end

mat_hn2d_half_1=[mat_hn2d;mat_hn2d_inv];      % Matriz con la mitad de los coeficientes

for i=2:min(size(mat_hn2d_half_1))
    mat_hn2d_half_2(:,i-1)=mat_hn2d_half_1(:,min(size(mat_hn2d_half_1))-i+2);
end

% La matriz 'MATRIZ' contiene los N1*N2 coeficientes del filtro diseñado
MATRIZ=[mat_hn2d_half_2 mat_hn2d_half_1];

% FIN DEL DISEÑO DEL FILTRO CUADRADO DE TAMAÑO N1*N2

% Por último, representamos la respuesta en frecuencia del filtro diseñado mediante
% el uso de la NDFT

l=128;
Hw1w2=fft2(MATRIZ,l,l);

w1=[-pi:2*pi/l:pi-2*pi/l]/(2*pi);
w2=[-pi:2*pi/l:pi-2*pi/l]/(2*pi);

hold on

```

```

contour(w1,w2,fftshift(abs(Hw1w2)),9)      % Gráfica de las curvas de nivel del filtro

% Localización de las muestras ind.
plot(v_muestras(:,1)/(2*pi),v_muestras(:,2)/(2*pi),'x')
axis([-0.5 0.5 -0.5 0.5])
pause
hold off

mesh(w1,w2,fftshift(abs(Hw1w2)))      % Gráfica del módulo del filtro

```

FIN DEL CÓDIGO DEL PROGRAMA EN MATLAB PARA LA OBTENCIÓN DEL FILTRO CUADRADO MEDIANTE EL USO DE LA NDFT

La respuesta en frecuencia del filtro cuadrado mostrado en la figura 10 es una respuesta claramente separable. Podemos diseñar un filtro cuadrado separable $h[n_1, n_2]$ diseñando en primer lugar dos filtros paso de baja unidimensionales $h_1[n]$ y $h_2[n]$, y luego multiplicando ortogonalmente sus respuestas impulsivas de la forma

$$h[n_1, n_2] = h_1[n_1]h_2[n_2].$$

En el dominio de la frecuencia, esto es totalmente equivalente a escribir lo siguiente

$$A(w_1, w_2) = A_1(w_1)A_2(w_2),$$

donde $A_1(w_1)$ y $A_2(w_2)$ son las funciones de amplitud de los filtros unidimensionales $h_1[n]$ y $h_2[n]$, respectivamente.

Los filtros separables son muy usados en la práctica debido a su diseño simple y a su fácil implementación. Sin embargo, estos filtros sufren del siguiente problema. Si δ_{p1}, δ_{s1} y δ_{p2}, δ_{s2} son los picos de rizado en las bandas de paso y de rechazo de los dos filtros unidimensionales respectivamente, entonces el filtro bidimensional resultante puede tener picos tan grandes como $(\delta_{p1} + \delta_{p2})$ en la banda de paso, y $\max(\delta_{s1}, \delta_{s2})$ en la banda de rechazo. Por tanto, el filtro bidimensional resultante puede tener rizados en la banda de paso mayores que sus originarios unidimensionales, lo cual es totalmente no deseable. En tales casos, resultados mucho mejores se pueden conseguir utilizando los N^2 grados de libertad disponibles en el diseño de filtros no separables en vez de usar solamente $2N$ grados en el caso de diseños separables. Vamos a comparar, a

continuación, el filtro cuadrado no separable diseñado anteriormente mediante la utilización de la NDFT con un filtro separable diseñado para las mismas especificaciones. Se ha usado el algoritmo de Parks-McClellan para diseñar el correspondiente filtro paso de baja unidimensional con longitud $N = 9$ y bordes de banda de paso y de rechazo iguales a los especificados para el ejemplo NDFT no separable que acabamos de ver. En las páginas siguientes, figuras 14 y 15, se muestran los resultados obtenidos mediante este proceso de diseño. Se puede observar que el rizado de la banda de paso es, en este caso, bastante mayor que para el caso NDFT no separable. Se comprueba, por tanto, que el diseño NDFT no separable tiene mejores características.

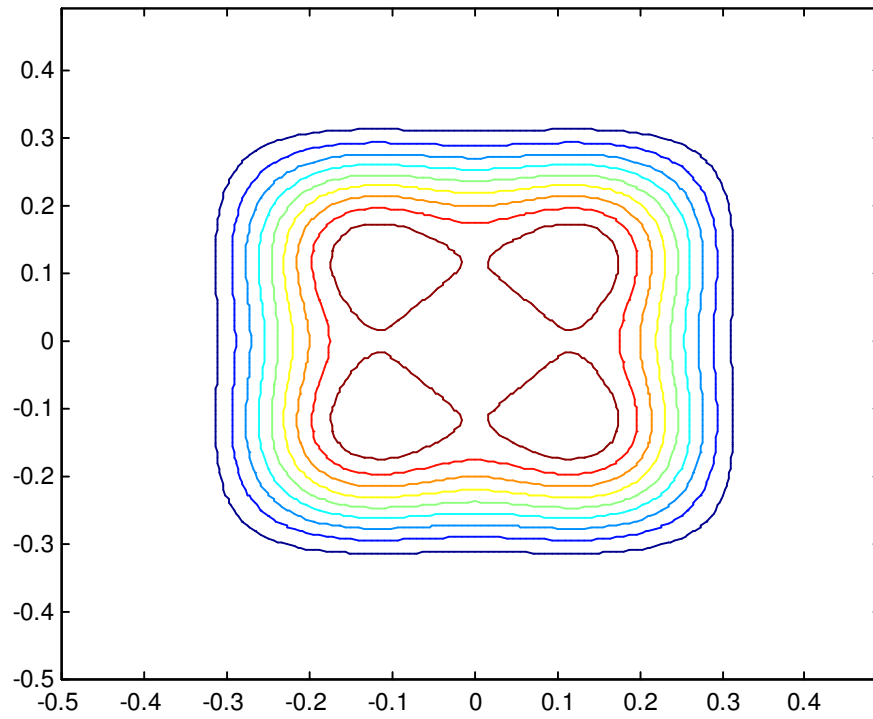


Figura 14. Curvas de nivel del filtro separable diseñado usando el algoritmo Parks-McClellan. Se puede observar los cuatro picos existentes en la banda de paso que son notablemente perjudiciales.

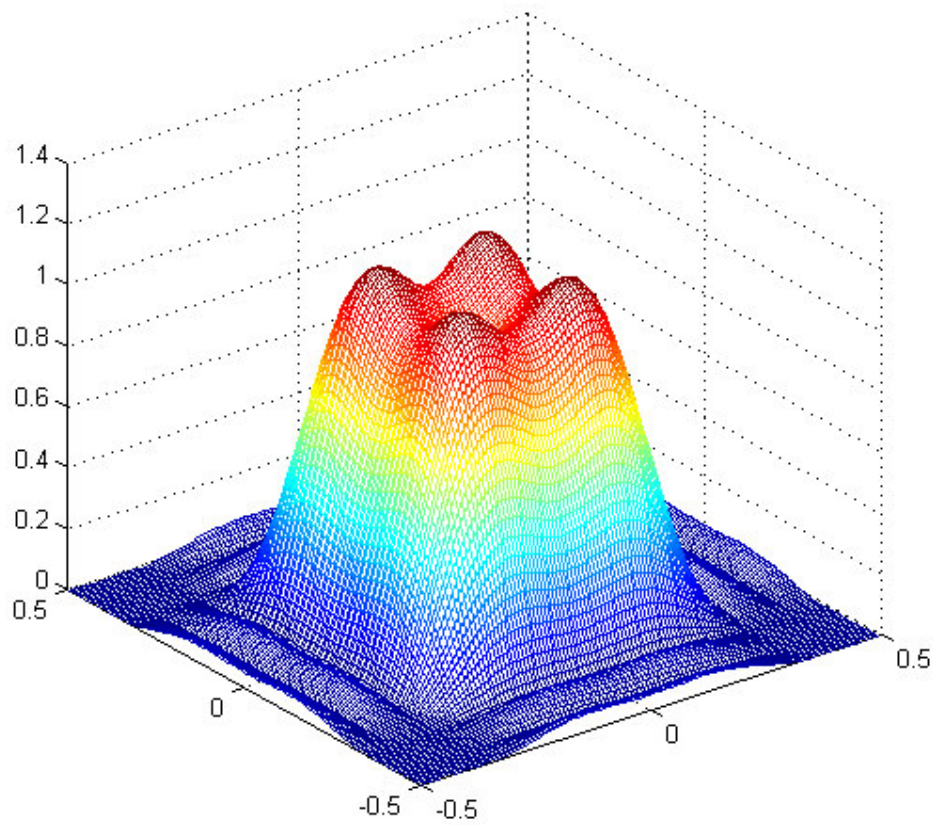


Figura 15. Respuesta en amplitud del filtro separable diseñado usando el algoritmo de Parks-McClellan. Se puede observar que el rizado en los bordes de la banda de paso es mucho mayor que para el filtro cuadrado diseñado anteriormente mediante el uso de la NDFT.

1.5 DISEÑO DE UN FILTRO CIRCULAR MEDIANTE LA APLICACIÓN DE LA NDFT

Seguidamente vamos a considerar el diseño de un filtro paso de baja circularmente simétrico de tamaño $N \times N$, que tiene unas especificaciones en cuanto a su respuesta en amplitud tal y como se muestra en la figura 16. El borde de la banda de paso está situado en w_p y el borde de la banda de rechazo está en w_s . Este tipo de filtros, al igual que los filtros cuadrados, presentan una clara simetría. De tal forma que se cumplen las siguientes expresiones:

$$h[n_1, n_2] = h[-n_1, n_2] = h[n_1, -n_2],$$

$$A(w_1, w_2) = A(-w_1, w_2) = A(w_1, -w_2),$$

por tanto, la respuesta en frecuencia $A(w_1, w_2)$ se puede expresar de la misma manera que para el filtro cuadrado, esto es,

$$\begin{aligned} A(w_1, w_2) = & h[0,0] + \sum_{n_1=1}^{\frac{N-1}{2}} 2h[n_1,0] \cos w_1 n_1 + \sum_{n_2=1}^{\frac{N-1}{2}} 2h[0,n_2] \cos w_2 n_2 + \\ & + \sum_{n_1=1}^{\frac{N-1}{2}} \sum_{n_2=1}^{\frac{N-1}{2}} 4h[n_1, n_2] \cos w_1 n_1 \cos w_2 n_2 \end{aligned}$$

Y el número de coeficientes independientes del filtro, N_i , viene dado por la misma expresión que vimos para el diseño del filtro cuadrado.

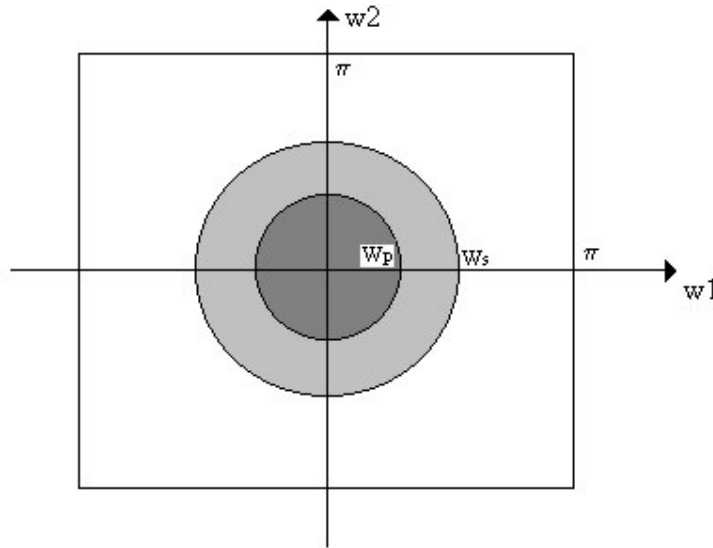


Figura 16. Especificaciones de un filtro circular paso de baja. La región más oscura es la banda de paso, la región grisácea es la banda de transición y la región sin colorear es la banda de rechazo del filtro.

Al igual que para el diseño del filtro cuadrado, necesitamos muestras de la respuesta deseada en N_f puntos del primer cuadrante del plano frecuencial. En este caso, las muestras van a estar situadas en contornos circulares (como ya se comentó, los mejores resultados se obtienen cuando los contornos donde van a ir situadas las muestras tienen la misma forma que la del filtro a diseñar). Por otro lado, la sección cruzada de la respuesta en amplitud bidimensional a lo largo del eje w_1 (o del eje w_2) es como la respuesta de un filtro paso de baja unidimensional, que se obtiene exactamente de la misma manera que para el filtro cuadrado, esto es, haciendo uso de lo explicado en el proyecto complementario a éste. Así, las $(N+1)/2$ muestras situadas a lo largo del eje w_1 están localizadas en los extremos de estas funciones analíticas (igual que para el caso del filtro cuadrado). El proceso de generación de las muestras restantes se puede ver como una rotación de las muestras localizadas en el eje w_1 . En el ejemplo siguiente se verá claramente como están situadas estas muestras.

Filtros paso de alta circularmente simétricos se pueden también diseñar mediante el uso de este método, generando funciones analíticas unidimensionales que proporcionen una respuesta paso de alta a lo largo del eje w_1 .

1.5.1 EJEMPLO DE DISEÑO

Considérese el diseño de un filtro paso de baja circularmente simétrico con las siguientes especificaciones:

Tamaño de la rejilla = 9×9

Borde de la banda de paso, $w_p = 0.4\pi$

Borde de la banda de rechazo, $w_s = 0.6\pi$.

De los 81 coeficientes del filtro, sólo 25 son independientes. Por tanto, necesitamos 25 muestras situadas sobre el primer cuadrante del plano frecuencial. Como ya se dijo, estas muestras están colocadas sobre contornos circulares del primer cuadrante. Los valores y las localizaciones de las 5 muestras a lo largo del eje w_1 son obtenidas al generar la respuesta unidimensional a lo largo de este eje. Después, estas muestras son usadas para generar 1, 3, 4, 5, 6 muestras emplazadas en cinco círculos consecutivos empezando por el origen. El resto de muestras, hasta llegar a las 25 necesarias, se sitúan en la esquina superior derecha del plano frecuencial, esto es, cerca de las coordenadas $(w_1, w_2) = (\pi, \pi)$. Estas muestras restantes están localizadas en arcos circulares uniformemente espaciados unos de otros. Como se puede observar en la figura 17, existen 3, 2, 1 muestras en 3 arcos circulares sucesivos y cada vez más

pequeños. Esta elección se ha hecho así porque ha funcionado con una gran cantidad de filtros diseñados de este tipo.

En general, para un filtro de tamaño $N \times N$, colocaremos 1, 3, 4, 5, ..., $(N+3)/2$ muestras en $(N+1)/2$ círculos empezando por el origen. Después, situaremos $(N-3)/2$, $(N-5)/2$, ... , 3, 2, 1 muestras en $(N-3)/2$ arcos circulares, uniformemente distanciados unos de otros, que van, radialmente, alejándose del origen. Las muestras, dentro de un círculo o arco concreto, están localizadas a distancias angulares constantes.

En las figuras siguientes, 17 y 18, mostramos las curvas de nivel del filtro diseñado con las características anteriormente citadas así como el lugar exacto donde se han colocado las 25 muestras para obtener el filtro. También, se muestra la respuesta en amplitud del filtro. Se puede observar que exhibe un alto grado de simetría circular así como un rizado muy pequeño.

Posteriormente, se adjunta el programa de MATLAB diseñado para conseguir este filtro paso de baja circularmente simétrico mediante la aplicación de la NDFT.

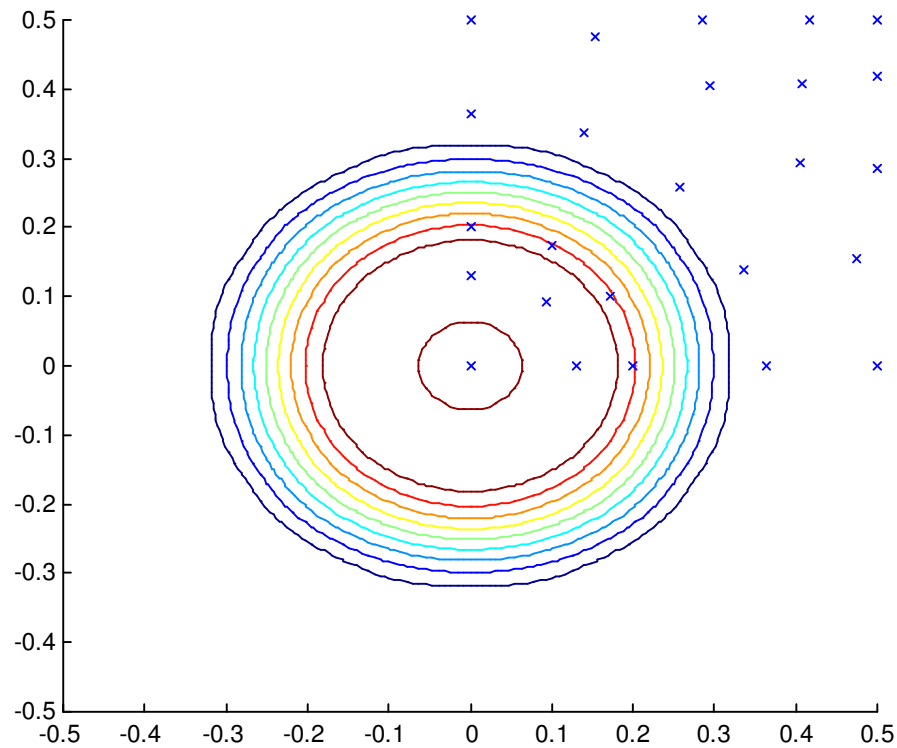


Figura 17. Filtro circular de tamaño 9 x 9 diseñado mediante el método de la NDFT.

Gráfica que muestra las curvas de nivel del filtro.

Las muestras están colocadas en los lugares denotados por una 'x' en la gráfica.

Los ejes de frecuencias están normalizados por 2π .

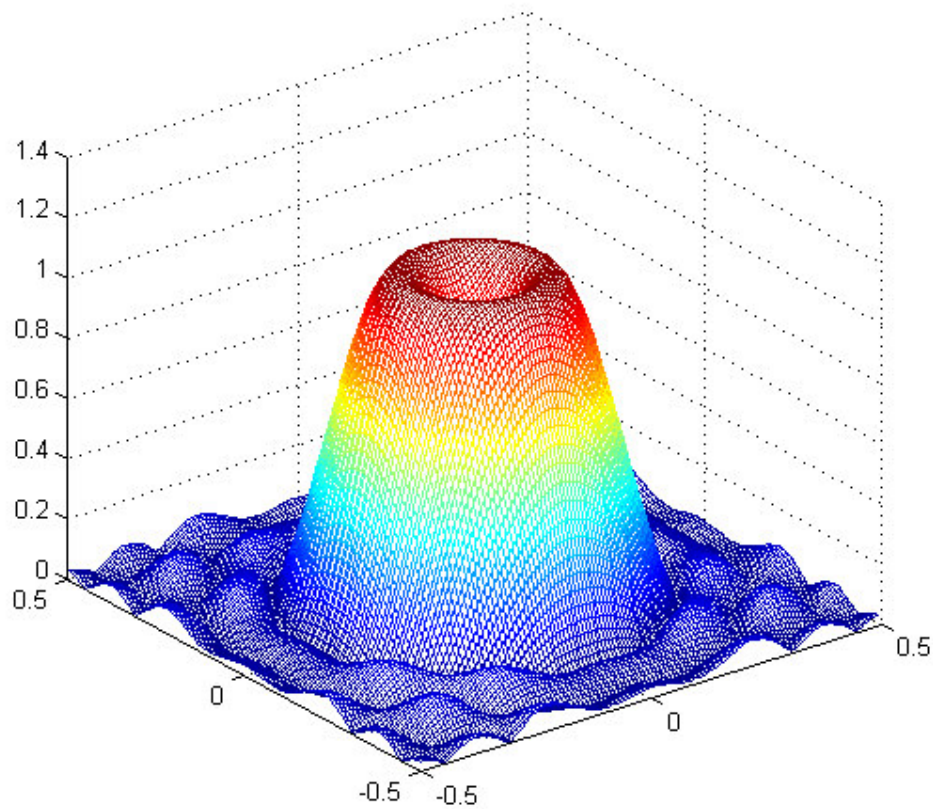


Figura 18. Filtro circular de tamaño 9 x 9 diseñado mediante el método de la NDFT.

Dibujo de la respuesta en frecuencia del filtro.

Los ejes de frecuencias están normalizados por 2π .

CÓDIGO DEL PROGRAMA EN MATLAB PARA LA OBTENCIÓN DEL FILTRO CIRCULAR MEDIANTE EL USO DE LA NDFT

```
% DISEÑO DE UN FILTRO CIRCULAR MEDIANTE EL USO DE LA NON-UNIFORM DISCRETE
% FOURIER TRANSFORM

clear;

% Especificaciones

N1=9;           % Longitud del filtro en uno de los ejes
N2=9;           % Longitud del filtro en el otro eje
wp=0.4*pi;      % Frecuencia de corte de la banda de paso
ws=0.6*pi;      % Frecuencia de corte de la banda de rechazo
k=3;           % Relación entre los rizados de las bandas de paso y rechazo respect.

% DISEÑO DEL FILTRO CIRCULAR, USAREMOS EL FILTRO LP DE TIPO 1 DE UNA DIMENSIÓN
% DESCRITO EN EL FICHERO NDFTLPTIPO1.M

% Cálculo de parámetros necesarios para el diseño

Ni=(N1+1)^2/4;   % Número de coeficientes independientes del filtro

% DISEÑO DEL FILTRO LOW PASS TIPO I DE UNA DIMENSIÓN

% Cálculo de los valores P y S (Órdenes de los polinomios de Chebyshev que servirán
% para generar las funciones Hp y Hs que son las respuestas en frecuencia en la banda
% de paso y en la de rechazo respectivamente)

x=(pi-ws)/wp;     % Factor de proporción entre las anchuras de las bandas
P=round((N1-1)/2)/(1+x)); % Cálculo de P
S=(N1-1)/2-P;     % Cálculo de S

% Cálculo de los rizados en la banda de paso y banda de rechazo

dp=sqrt(k)*10^(-0.1162*(ws-wp)*(N1-1)-0.65);
ds=dp/k;

% Cálculo de los coeficientes necesarios para la generación de las funciones necesarias
% para el diseño del filtro

A=0.5*((cosh((1/P)*acosh((1+ds)/dp)))+1);
B=A-1;
C=0.5*((cosh((1/S)*acosh((1+dp)/ds)))+1);
D=C-1;
a=1/wp*acos((B-1)/A);
b=pi;
c=1/(pi-ws)*acos((D-1)/C);
d=pi*(1-c);

% A continuación, calculamos las frecuencias de muestreo óptimas según este tipo de
% diseño

for i=1:P
    wbp(i)=1/a*(acos((cos(i*pi/P)-B)/A)-b); % Frec. Muestreo en banda de paso
end
```

```

for i=1:S
    wbs(i)=1/c*(acos((cos(i*pi/S)-D)/C)-d); % Frec. Muestreo en banda de rechazo
end

% El vector que contiene las frecuencias de muestreo será tal y como se muestra
% a continuación. Hemos añadido una muestra más porque necesitamos (N+1)/2 muestras
% para tener determinados todos los coeficientes. También, debido a la simetría
% cambiamos el signo del vector wbp puesto que nos ha salido negativo como consecuencia
% del uso de funciones trigonométricas.

for i=1:P
    wbp(i)=-wbp(P-i+1);
end
wbp(P+1)=wp;

w=[wbp, wbs]; % VECTOR FINAL QUE CONTIENE LAS FRECUENCIAS DE MUESTREO

% Procedemos a la generación de A(w) tal y como se explica en la teoría

xpw=A*cos(a*wbp+b)+B;
xsw=C*cos(c*wbs+d)+D;

Apw=1-dp*cos(P*acos(xpw));
Asw=ds*cos(S*acos(xsw));

Aw=[Apw, Asw]; % VECTOR QUE CONTIENE EL VALOR EN MÓDULO DE LA RESPUESTA EN FRECUENCIA
% EN CADA UNA DE LAS FRECUENCIAS DE MUESTREO PREVIAMENTE CALCULADAS

% FIN DEL DISEÑO DEL FILTRO LP TIPO I DE UNA DIMENSIÓN

% CONSTRUCCIÓN DE LA REJILLA DE MUESTRAS

% Cálculo de las muestras pertenecientes a los contornos de TIPO 1

n_circ1=(N1+1)/2; % Número de contornos de TIPO 1

% Cálculo de las coordenadas donde se sitúan las muestras según este procedimiento.
% La hipermatriz de nombre 'matrix1' contiene las coordenadas de las muestras para cada
% contorno de TIPO 1.

matrix1(:, :, 1)=[0 0]; % Coordenadas de la única muestra del primer contorno de TIPO 1

% Cálculo de las coordenadas del resto de contornos de TIPO 1

for i=2:n_circ1
    fase=(pi/2)/i;
    for j=1:(i+1)
        matrix1(j, :, i)=[w(i)*cos((j-1)*fase) w(i)*sin((j-1)*fase)];
    end
end

% Cálculo de las muestras pertenecientes a los contornos de TIPO 2
% La hipermatriz de nombre 'matrix2' contiene las coordenadas de las muestras para cada
% contorno de TIPO 2.

j=sqrt(-1);

% Muestra mitad del último contorno de TIPO 1
mid_sample=matrix1(floor(n_circ1/2)+1, :, n_circ1);

% Distancia entre (pi, pi) y mid_sample
dist=abs((pi-mid_sample(1))+j*(pi-mid_sample(2)));

inc=dist/((N1-3)/2); % Distancia entre los contornos de TIPO 2

n_circ2=(N1-3)/2; % Número de contornos de TIPO 2

for i=1:n_circ2-1
    v_modaux(i)=pi+i*inc; % Cálculo del radio de los contornos de este tipo
end

```

```

for i=1:n_circ2-1
    mat_aux1(i,:)=[pi sqrt(v_modaux(i)^2-pi^2)]; % Coordenadas extremas del tipo (pi,w2)
    mat_aux2(i,:)=[sqrt(v_modaux(i)^2-pi^2) pi]; % Coordenadas extremas del tipo (w1,pi)
end

% Pasamos de coordenadas vectoriales a coordenadas complejas

j=sqrt(-1);
for i=1:n_circ2-1
    v_aux1(i)=mat_aux1(i,1)+j*mat_aux1(i,2);
    v_aux2(i)=mat_aux2(i,1)+j*mat_aux2(i,2);
end

% Calculamos la diferencia de fase entre las muestras extremas de cada contorno TIPO 2
for i=1:n_circ2-1
    dif_fase(i)=angle(v_aux2(i))-angle(v_aux1(i));
end

% Calculamos el incremento de fase entre las muestras contiguas de cada contorno TIPO 2
for i=1:n_circ2-1
    inc_fase(i)=dif_fase(i)/(n_circ2-i);
end

% Ya estamos en disposición de calcular las coordenadas exactas de las muestras de cada
% contorno TIPO 2

clear j;
for i=1:n_circ2
    % Se hace ya que no hay ceros en la primera hipermatriz
    matrix2(:, :, i)=zeros((N1-1)/2,2); % de matrix2 y por tanto, el algoritmo de extracción
end % posterior de v_muestras no funcionaría
matrix2(1, :, n_circ2)=[pi pi];
for i=1:n_circ2-1
    for j=1:n_circ2+1-i
        matrix2(j, :, i)=[v_modaux(i)*cos(angle(v_aux1(i))+(j-1)*inc_fase(i))
        v_modaux(i)*sin(angle(v_aux1(i))+(j-1)*inc_fase(i))];
    end
end

% Colocamos las muestras en un vector de tamaño Ni (número de muestras necesarias)
% Lo hacemos en dos partes puesto que tenemos dos tipos de muestra en este caso
v_muestras(1,:)=[0 0]; % Colocamos la primera muestra que es siempre la misma
k=2; % Indice que recorre las hipermatrices
p=1; % Indice que recorre las filas de cada hipermatriz
n_samples_t1=sum(3:(N1+3)/2)+1; % Número de muestras de tipo I
for i=2:n_samples_t1
    if (matrix1(p, :, k)==[0 0]) % Si de la hipermatriz que extraemos sacamos
        k=k+1; % un valor [0 0] significa que ya está agotada
        p=1; % y, por tanto, debemos pasar a la siguiente
    end
    v_muestras(i,:)=matrix1(p, :, k);
    p=p+1;
end

k=1; % Indice que recorre las hipermatrices (=1, ya que no hay (0,0))
p=1; % Indice que recorre las filas de cada hipermatriz
for i=n_samples_t1+1:Ni
    if (matrix2(p, :, k)==[0 0]) % Si de la hipermatriz que extraemos sacamos
        k=k+1; % un valor [0 0] significa que ya está agotada
        p=1; % y, por tanto, debemos pasar a la siguiente
    end
    v_muestras(i,:)=matrix2(p, :, k);
    p=p+1;
end

% Ya tenemos en v_muestras las Ni muestras necesarias para generar el filtro.
% Ahora asignamos el valor que toma el filtro en cada una de las muestras.
% Nos ocupamos en primer lugar de los valores en las muestras TIPO 1.

aux=0; % Índice auxiliar
Aw2d(1)=Aw(1); % Valor en la muestra (0,0)
for i=2:n_circ1
    for j=1:i+1
        Aw2d(aux+j+1)=Aw(i);
    end
    aux=aux+(i+1);
end

```

```

% Asignamos valores a las muestras TIPO 2. Les vamos a asignar ds y -ds, rizado máx.
% en banda de rechazo, puesto que dichas muestras se encuentran en tal banda.

aux=0;
for i=1:n_circ2
    for j=1:n_circ2+1-i
        Aw2d(n_samples_t1+j+aux)=(-1)^(i+1)*ds;
    end
    aux=aux+n_circ2+1-i;
end

% Cálculo de los coeficientes h(n1,n2) haciendo uso de la fórmula expuesta en la parte
% teórica.

for i=1:(N1-1)/2
    m1_aux(:,i)=2*cos(v_muestras(:,1)*i);
end

for i=1:(N1-1)/2
    m2_aux(:,i)=2*cos(v_muestras(:,2)*i);
end

m1_aux2=m1_aux/2;
m2_aux2=m2_aux/2;

for i=1:max(size(m1_aux2))
    for j=1:min(size(m1_aux2))
        m3_aux(i,((j-1)*((N1-1)/2)+1):(j*((N1-1)/2)))=4*m1_aux2(i,j)*m2_aux2(i,:);
    end
end

% Construimos la matriz total D.
D=[ones(N1,1) m1_aux m2_aux m3_aux];

% Obtención de los Ni coeficientes independientes del filtro.
hn2d=inv(D)*Aw2d';

% Ordenamos hn2d matricialmente con el objeto de expresar los coeficientes independ.
% de una manera más apropiada para la obtención del resto de coeficientes simétricos.

for i=1:(N1+1)/2
    ind_aux=(i-1)*(N1-1)/2+(N1+3)/2;
    mat_hn((N1+1)/2-i+1,:)=hn2d(ind_aux:ind_aux+(N1-3)/2)';
end

mat_hn2d=[hn2d((N1+1)/2:-1:1) mat_hn];

% Ahora, aplicamos la simetría existente para hallar los coeficientes restantes

for i=1:length(mat_hn2d)-1
    mat_hn2d_inv(i,:)=mat_hn2d(length(mat_hn2d)-i,:);
end

mat_hn2d_half_1=[mat_hn2d;mat_hn2d_inv]; % Matriz con la mitad de los coeficientes

for i=2:min(size(mat_hn2d_half_1))
    mat_hn2d_half_2(:,i-1)=mat_hn2d_half_1(:,min(size(mat_hn2d_half_1))-i+2);
end

% La matriz 'MATRIZ' contiene los N1*N2 coeficientes del filtro diseñado
MATRIZ=[mat_hn2d_half_2 mat_hn2d_half_1];

% FIN DEL DISEÑO DEL FILTRO CIRCULAR DE TAMAÑO N1*N2

% Por último, representamos la respuesta en frecuencia del filtro diseñado mediante
% el uso de la NDFT

l=128;
Hw1w2=fft2(MATRIZ,l,l);

w1=[-pi:2*pi/l:pi-2*pi/l]/(2*pi);
w2=[-pi:2*pi/l:pi-2*pi/l]/(2*pi);

```

```

hold on
contour(w1,w2,fftshift(abs(Hw1w2)),9)      % Gráfica de las curvas de nivel del filtro

% Localización de las muestras ind.
plot(v_muestras(:,1)/(2*pi),v_muestras(:,2)/(2*pi),'x')

axis([-0.5 0.5 -0.5 0.5])
pause
hold off

mesh(w1,w2,fftshift(abs(Hw1w2)))          % Gráfica del módulo del filtro

```

FIN DEL CÓDIGO DEL PROGRAMA EN MATLAB PARA LA OBTENCIÓN DEL FILTRO CIRCULAR MEDIANTE EL USO DE LA NDFT

1.6 DISEÑO DE UN FILTRO DIAMANTE (ROMBO) MEDIANTE LA APLICACIÓN DE LA NDFT

Los filtros diamantes (o rombos) tienen importantes aplicaciones prácticas siendo una de las más importantes su uso en las señales de televisión. Un filtro diamante tiene unas especificaciones de amplitud tal y como se muestra en la figura 19, con el borde de la banda de paso situado en w_p y el borde de la banda de rechazo en w_s . Se puede observar que, la recta $w_1 = w_2$ intersecta el borde de la banda de paso en (w_p, w_p) y el borde de la banda de rechazo en (w_s, w_s) . También se puede advertir que, un filtro diamante presenta una simetría de orden 8 en el dominio frecuencial. Esto nos permite escribir

$$A(w_1, w_2) = A(-w_1, w_2) = A(w_1, -w_2) = A(w_2, w_1).$$

Equivalentemente, la respuesta impulsiva $h[n_1, n_2]$ satisface la relación

$$h[n_1, n_2] = h[-n_1, n_2] = h[n_1, -n_2] = h[n_2, n_1].$$

Además, un filtro diamante es un filtro bidimensional ‘half-band’. Su respuesta en amplitud, $A(w_1, w_2)$, es simétrica entorno al punto $(w_1, w_2, A(w_1, w_2)) = (\pi/2, \pi/2, 0.5)$ en el plano frecuencial:

$$A(w_1, w_2) + A(\pi - w_1, \pi - w_2) = 1.$$

Esto implica que la respuesta impulsiva tiene ceros alternantes. Para este tipo de filtros se cumple que

$$h[n_1, n_2] = \begin{cases} 0, & n_1 + n_2 = \text{par}, \\ 0.5, & n_1 = n_2 = 0. \end{cases}$$

Se puede deducir de esto que el número de coeficientes independientes de un filtro diamante es realmente pequeño.

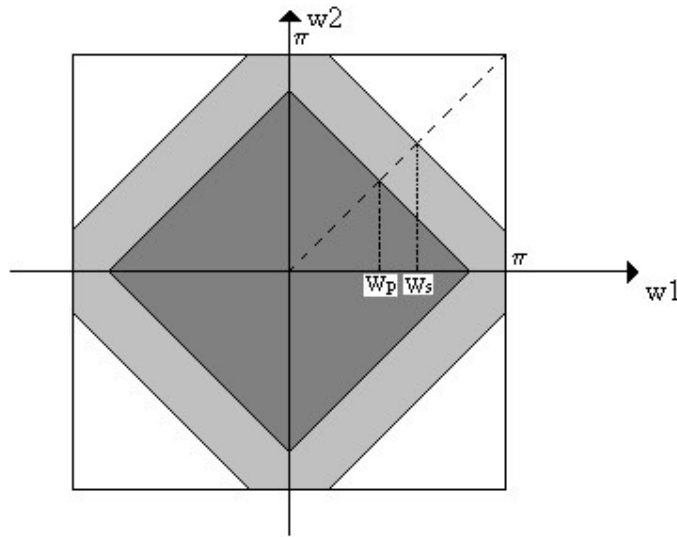


Figura 19. Especificaciones de un filtro diamante. La región más oscura es la banda de paso, la región grisácea es la banda de transición y la región sin colorear es la banda de rechazo del filtro.

Por ejemplo, la distribución de los valores de la respuesta impulsiva de un filtro ‘half-band’ de tamaño 9 x 9 se muestra en la figura 20. A causa de las propiedades de las ecuaciones anteriores, el número de coeficientes independientes de un filtro diamante de tamaño N x N se reduce a

$$N_i = \left\lfloor \frac{P+1}{2} \right\rfloor \left\lfloor \frac{P+2}{2} \right\rfloor,$$

donde

$$P = \frac{N-1}{2}.$$

Los N_i coeficientes independientes de la respuesta impulsiva yacen en una región en forma de cuña situada por debajo de la diagonal $n_1 = n_2$ en el primer cuadrante del plano espacial, (n_1, n_2) .

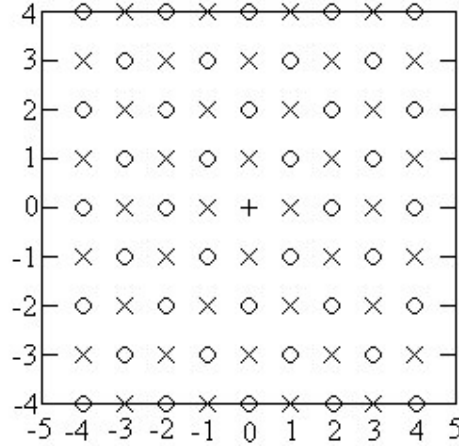


Figura 20. Distribución de los valores de la respuesta impulsiva de un filtro FIR Half-Band de dimensiones 9 x 9. Los coeficientes de valor cero se indican mediante 'O', los coeficientes distintos de cero se indican mediante 'X'. El valor del coeficiente central '+' es 0.5 .

Teniendo en cuenta todo lo comentado, la respuesta en amplitud de un filtro diamante se puede expresar mediante

$$A(w_1, w_2) = 0.5 + \sum_{n_1=1}^{\lfloor (P+1)/2 \rfloor} 2h[2n_1-1, 0] \{ \cos(2n_1-1)w_1 + \cos(2n_1-1)w_2 \} + \\ + \sum_{n_1=1}^{\lfloor (P+1)/2 \rfloor} \sum_{n_2=1}^{\lfloor P/2 \rfloor} 4h[2n_1-1, 2n_2] \{ \cos(2n_1-1)w_1 \cos(2n_2)w_2 + \cos(2n_2)w_1 \cos(2n_1-1)w_2 \}$$

Debido a la simetría de orden 8 y a la naturaleza 'half-band' del filtro, la única parte independiente de la respuesta en amplitud es un área triangular dentro de la banda de paso tal y como se muestra en la figura 21. En nuestro método de diseño, N_i muestras se colocan dentro de esta región del plano frecuencial. Si tomamos una sección cruzada de $A(w_1, w_2)$ a lo largo de la diagonal $w_1 = w_2$, ésta se parece a un filtro 'half-band' paso de baja unidimensional. Como siempre, aproximamos la banda de paso de esta respuesta por una función unidimensional $H_p(w)$, donde P, que es el orden del polinomio de Chebyshev $T_p(x)$, vale $(N-1)/2$. (Para ver el proceso exacto de cómo diseñar un filtro paso de baja 'half-band' unidimensional véase el proyecto complementario a éste que ya hemos citado en varias ocasiones). Las muestras se colocarán en $(N-1)/2$ líneas de pendiente (-1), que pasan por los extremos de $H_p(w)$. Todas las muestras pertenecientes a una misma línea tendrán el mismo valor y estarán equidistantemente espaciadas. El número de muestras en las líneas sucesivas,

comenzando desde el origen, viene dado en la tabla siguiente, que recoge desde filtros de tamaño 7×7 hasta filtros de tamaño 31×31 . Este rango de tamaños de filtros es suficientemente grande para cubrir todas la mayoría de las aplicaciones prácticas. La distribución de muestras dada se ha encontrado que funciona bien para varias elecciones de bandas de paso. Se debe advertir también que, si hay sólo una muestra para una determinada línea de contorno, ésta estará situada en el eje w_1 . Por último, las N_i muestras nos dan las ecuaciones necesarias para hallar los coeficientes independientes del filtro usando la expresión desarrollada de $A(w_1, w_2)$.

N	N_i	Número de muestras en los contornos sucesivos
7	4	1, 2, 1
9	6	1, 1, 2, 2
11	9	1, 1, 2, 3, 2
13	12	1, 1, 2, 3, 3, 2
15	16	1, 1, 2, 3, 3, 4, 2
17	20	1, 1, 2, 3, 3, 4, 4, 2
19	25	1, 1, 2, 3, 3, 4, 4, 4, 3
21	30	1, 1, 2, 3, 3, 3, 4, 4, 5, 4
23	36	1, 1, 2, 3, 3, 3, 4, 4, 5, 6, 4
25	42	1, 1, 2, 3, 3, 4, 4, 4, 5, 6, 6, 5
27	49	1, 1, 2, 3, 3, 4, 4, 4, 5, 6, 6, 6, 5
29	56	1, 1, 2, 3, 3, 4, 4, 4, 5, 6, 6, 6, 7, 5
31	64	1, 1, 2, 3, 3, 4, 4, 4, 5, 6, 6, 6, 7, 7, 6

Distribución de las muestras para el diseño de un filtro diamante. La última columna recoge el número de muestras para cada uno de los $(N - 1) / 2$ contornos sucesivos para el diseño de un filtro de tamaño $N \times N$, que tiene N_i coeficientes independientes.

El método de diseño propuesto proporciona filtros diamante de alta calidad con bajo rizado y una banda de paso muy bien perfilada.

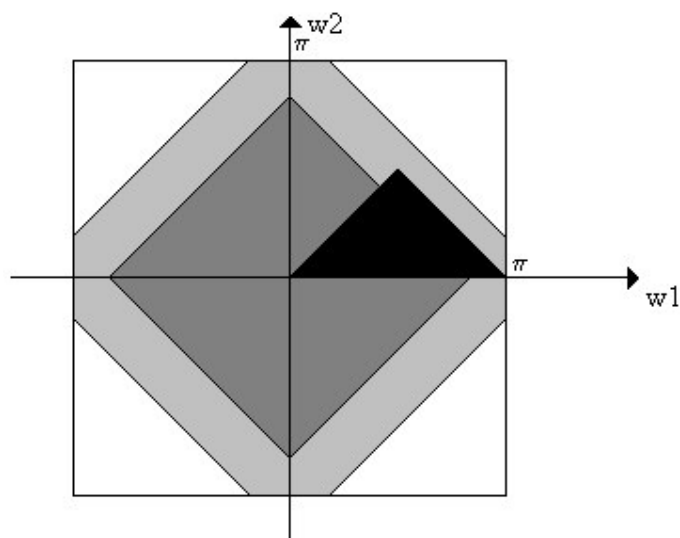


Figura 21. Respuesta en amplitud de un filtro diamante. La región más oscura es la única parte independiente debido a la simetría de orden 8 y a la naturaleza 'half-band' del filtro.

1.6.1 EJEMPLO DE DISEÑO

Considérese el diseño de un filtro diamante con las siguientes especificaciones:

Tamaño de la rejilla = 9×9

Borde de la banda de paso, $w_p = 0.36\pi$

Borde de la banda de rechazo, $w_s = 0.64\pi$.

Haciendo uso de la tabla anterior, comprobamos que sólo 6 de los 81 coeficientes del filtro son independientes. Las 6 muestras independientes están situadas, tal y como se muestra en la figura 22, en líneas que siguen la forma en rombo del filtro. La sección cruzada de la respuesta en amplitud bidimensional se obtiene a través de una función $H_p(w)$, que tiene, para este caso, 4 extremos. En consecuencia, las muestras se colocan en 4 líneas que pasan a través de estos 4 extremos. El número de muestras en estas líneas son 1, 1, 2, 2, respectivamente, tal y como se nos dice en la tabla anterior.

En las figuras siguientes, 22 y 23, mostramos las curvas de nivel del filtro diseñado con las características anteriormente citadas así como el lugar exacto donde se han colocado las 6 muestras necesarias para obtener el filtro. También, se muestra la respuesta en amplitud del filtro.

Posteriormente, se adjunta el programa de MATLAB diseñado para conseguir este filtro diamante mediante la aplicación de la NDFT.

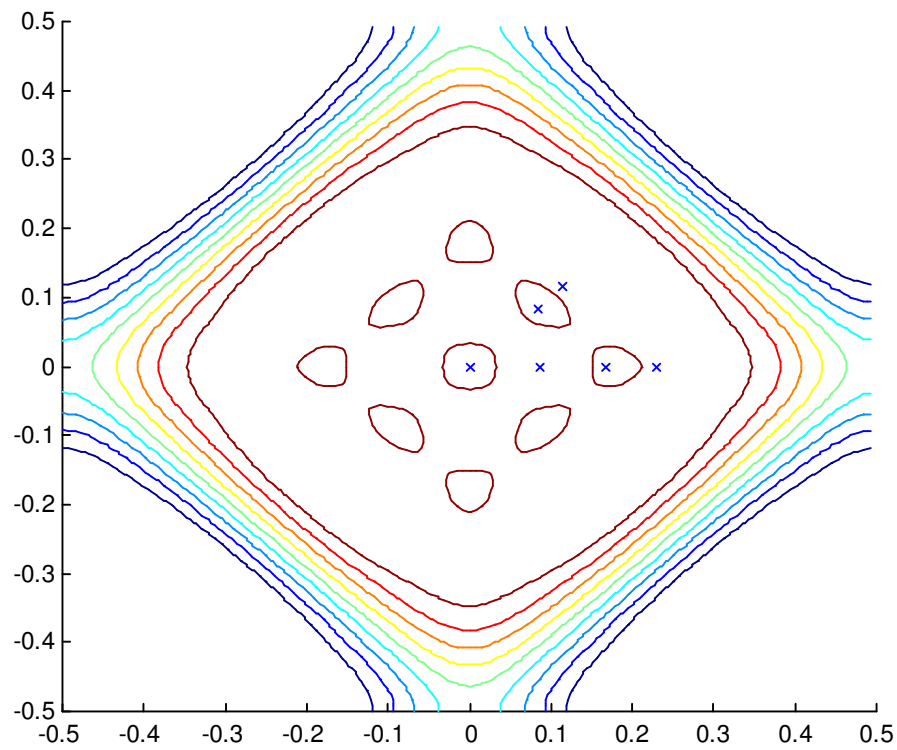


Figura 22. Filtro diamante de tamaño 9 x 9 diseñado mediante el método de la NDFT.

Gráfica que muestra las curvas de nivel del filtro.

Las muestras están colocadas en los lugares denotados por una 'x' en la gráfica.

Los ejes de frecuencias están normalizados por 2π .

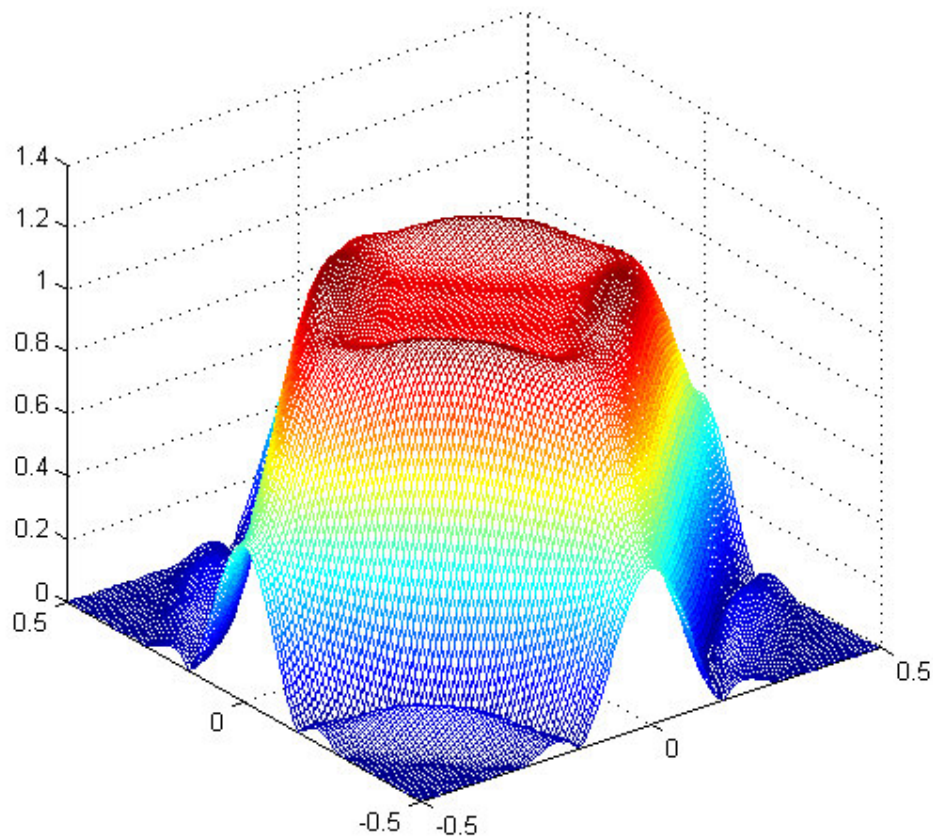


Figura 23. Filtro diamante de tamaño 9 x 9 diseñado mediante el método de la NDFT.

Dibujo de la respuesta en frecuencia del filtro.

Los ejes de frecuencias están normalizados por 2π .

CÓDIGO DEL PROGRAMA EN MATLAB PARA LA OBTENCIÓN DEL FILTRO DIAMANTE MEDIANTE EL USO DE LA NDFT

```
% DISEÑO DE UN FILTRO DIAMANTE MEDIANTE EL USO DE LA NON-UNIFORM DISCRETE
% FOURIER TRANSFORM

clear;

% Especificaciones

N1=9;           % Longitud del filtro en uno de los ejes
N2=9;           % Longitud del filtro en el otro eje
wp=0.36*pi;     % Frecuencia de corte de la banda de paso
ws=0.64*pi;     % Frecuencia de corte de la banda de rechazo
k=1/3;          % Relación entre los rizados de las bandas de paso y rechazo respect.

% TABLA EMPÍRICA QUE CONTIENE LA DISTRIBUCIÓN DE MUESTRAS EN LOS SUCEIVOS CONTORNOS
% SEGÚN SEA LA LONGITUD

TABLA(1,:)=[1 2 1 zeros(1,13)];           % PARA N1,N2=7
TABLA(2,:)=[1 1 2 2 zeros(1,12)];         % PARA N1,N2=9
TABLA(3,:)=[1 1 2 3 2 zeros(1,11)];       % PARA N1,N2=11
TABLA(4,:)=[1 1 2 3 3 2 zeros(1,10)];     % PARA N1,N2=13
TABLA(5,:)=[1 1 2 3 3 4 2 zeros(1,9)];    % PARA N1,N2=15
TABLA(6,:)=[1 1 2 3 3 4 4 2 zeros(1,8)];  % PARA N1,N2=17
TABLA(7,:)=[1 1 2 3 3 4 4 4 3 zeros(1,7)]; % PARA N1,N2=19
TABLA(8,:)=[1 1 2 3 3 3 4 4 5 4 zeros(1,6)]; % PARA N1,N2=21
TABLA(9,:)=[1 1 2 3 3 3 4 4 5 6 4 zeros(1,5)]; % PARA N1,N2=23
TABLA(10,:)=[1 1 2 3 3 4 4 4 5 6 6 5 zeros(1,4)]; % PARA N1,N2=25
TABLA(11,:)=[1 1 2 3 3 4 4 4 5 6 6 6 5 zeros(1,3)]; % PARA N1,N2=27
TABLA(12,:)=[1 1 2 3 3 4 4 4 5 6 6 6 7 5 zeros(1,2)]; % PARA N1,N2=29
TABLA(13,:)=[1 1 2 3 3 4 4 4 5 6 6 6 7 7 6 0]; % PARA N1,N2=31

% VECTOR QUE CONTIENE EL NÚMERO DE COEFICIENTES INDEPENDIENTES Y QUE HACE REFERENCIA
% A CADA FILA DE LA TABLA ANTERIOR, DEPENDIENDO TODO ELLO DE LA LONGITUD DEL FILTRO
% A DISEÑAR.

VECTOR_NI=[4 6 9 12 16 20 25 30 36 42 49 56 64];

% Cálculo del número de coeficientes independientes, se mete en Ni.

P1=(N1-1)/2;
Ni=floor((P1+1)/2)*floor((P1+2)/2);

% DISEÑO DEL FILTRO DIAMANTE, USAREMOS EL FILTRO HALF BAND (TIPO 1) DE UNA DIMENSIÓN
% DESCRITO EN EL FICHERO NDFTHALFBAND.M

% DISEÑO DEL FILTRO HALF BAND (TIPO I)

% Cálculo de los valores P y S (Órdenes de los polinomios de Chebyshev que servirán
% para generar las funciones Hp y Hs que son las respuestas en frecuencia en la banda
% de paso y en la de rechazo respectivamente)

P=(N1-1)/2;           % Cálculo de P
S=P;                  % Cálculo de S

% Cálculo de los rizados en la banda de paso y banda de rechazo

dp=sqrt(k)*10^(-0.1162*(ws-wp)*(N1-1)-0.65);
ds=dp/k;
```

```

% Cálculo de los coeficientes necesarios para la generación de las funciones necesarias
% para el diseño del filtro (en este caso, la mitad que para el filtro LP Tipo I debido
% a la simetría existente)

A=0.5*((cosh((1/P)*acosh((1+ds)/dp))+1);
B=A-1;
a=1/wp*acos((B-1)/A);
b=pi;

% A continuación, calculamos las frecuencias de muestreo. En este caso sólo P muestras
% debido a la simetría citada anteriormente

for i=1:P
    wbp(i)=1/a*(acos((cos(i*pi/P)-B)/A)-b);    % Frec. Muestreo en banda de paso
end

% El vector que contiene las frecuencias de muestreo será tal y como se muestra
% a continuación. Debido a la simetría cambiamos el signo del vector wbp
% puesto que nos ha salido negativo como consecuencia del uso
% de funciones trigonométricas

for i=1:P
    wbp(i)=-wbp(P-i+1);
end

w=sqrt(2)*wbp;    % VECTOR FINAL QUE CONTIENE LAS FRECUENCIAS DE MUESTREO
                  % MULTIPLICAMOS POR RAÍZ DE 2 PARA PASAR AL EJE 'w1'

% Procedemos a la generación de A(w) tal y como se explica en la teoría

xpw=A*cos(a*wbp+b)+B;

Apw=1-dp*cos(P*acos(xpw));

Asw=ds*cos(P*acos(xpw));

Aw=[Apw, Asw];    % VECTOR QUE CONTIENE EL VALOR EN MÓDULO DE LA RESPUESTA EN FRECUENCIA
                  % EN CADA UNA DE LAS FRECUENCIAS DE MUESTREO PREVIAMENTE CALCULADAS

% FIN DEL DISEÑO DEL FILTRO HALF BAND (TIPO I) DE UNA DIMENSIÓN

% CONSTRUCCIÓN DE LA REJILLA DE MUESTRAS

% En primer lugar, entramos en los vectores VECTOR_NI y TABLA para determinar
% tanto el número de contornos como el número de muestras por contorno

indice=find(VECTOR_NI==Ni);
num_cont=0;

for i=1:max(size(TABLA))
    if TABLA(indice,i)~=0
        num_cont=num_cont+1;    % Contiene el número de contornos
    end
end

% Vector que contiene el número de muestras de cada contorno
mues_cont=TABLA(indice,1:num_cont);

% Cálculo de las coordenadas donde se sitúan las muestras según este procedimiento.
% La hipermatriz de nombre 'matriz' contiene las coordenadas de las muestras para cada
% contorno.
for k=1:num_cont
    if mues_cont(k)==1
        matriz(1,:,k)=[w(k), 0];
    else
        for j=1:mues_cont(k)
            matriz(j,:,k)=[w(k)-w(k)/(2*(mues_cont(k)-1))*(j-1), w(k)/(2*(mues_cont(k)-
1))*(j-1)];
        end
    end
end

```

```

% Colocamos las muestras en un vector de tamaño N1 (número de muestras necesarias).
v_muestras(1,:)=[0 0];           % Colocamos la primera muestra que es siempre la misma

index=1;
for k=2:num_cont
    for p=1:mues_cont(k)
        v_muestras(index+1,:)=matriz(p,:,k); % Vamos colocando el resto de muestras
        index=index+1;
    end
end

% Asignamos a cada muestra el valor de A(w) correspondiente, es decir, cada contorno
% de la rejilla tomará un único valor de A(w).

aux=0;
for i=1:num_cont
    for j=1:mues_cont(i)
        Aw2d(j+aux)=Aw(i);           % Vector que contiene el valor de A(w) para cada muestra
    end
    aux=aux+mues_cont(i);
end

% Plateamos la ecuacion del filtro diamante que nos dara los coeficientes independ.
% Cálculo de los coeficientes h(n1,n2).

for i=1:floor((P1+1)/2)
    m1_aux(:,i)=cos(v_muestras(:,1)*(2*i-1));
end

for i=1:floor((P1+1)/2)
    m2_aux(:,i)=cos(v_muestras(:,2)*(2*i-1));
end

for i=1:floor(P1/2)
    m3_aux(:,i)=cos(v_muestras(:,2)*(2*i));
end

for i=1:floor(P1/2)
    m4_aux(:,i)=cos(v_muestras(:,1)*(2*i));
end

submat1=2*(m1_aux+m2_aux);

for i=1:max(size(m1_aux))
    for j=1:min(size(m1_aux))
        submat2(i,((j-1)*floor(P1/2)+1):(j*floor(P1/2)))=4*m1_aux(i,j)*m3_aux(i,:)+4*m2_aux(i,j)*m4_aux(i,:);
    end
end

% Restamos el 0.5 de Aw2d (Aplicación de la fórmula)
Aw2d=Aw2d-0.5;

% Construcción de D
D=[submat1 submat2];

% Despejamos los coeficientes
hn2d=inv(D)*Aw2d';

% Completamos la matriz MATRIZ que contiene todos los coeficientes del filtro teniendo
% en cuenta todas las propiedades de simetria del filtro
ind=1;
for i=0:(N1-1)/2
    for j=(i+1):(N1-1)/2
        if mod(i+j,2)==0
            MATRIZ1C((N1-1)/2+1-i,j+1)=0;
        else
            MATRIZ1C((N1-1)/2+1-i,j+1)=hn2d(ind);
            ind=ind+1;
        end
    end
end
end

```

```

% Completamos el resto del primer cuadrante
for j=0:(N1-1)/2-1
    for i=(j+1):(N1-1)/2
        MATRIZ1C((N1-1)/2-i+1,j+1)=MATRIZ1C((N1-1)/2-j+1,i+1);
    end
end

MATRIZ1C((N1-1)/2+1,1)=0.5; % Introducción del valor central

mat_hn2d=MATRIZ1C; % Asignación que permite usar código de otros ficheros ya realizados

% Ahora, aplicamos la simetría existente para hallar los coeficientes restantes

for i=1:length(mat_hn2d)-1
    mat_hn2d_inv(i,:)=mat_hn2d(length(mat_hn2d)-i,:);
end

mat_hn2d_half_1=[mat_hn2d;mat_hn2d_inv]; % Matriz con la mitad de los coeficientes

for i=2:min(size(mat_hn2d_half_1))
    mat_hn2d_half_2(:,i-1)=mat_hn2d_half_1(:,min(size(mat_hn2d_half_1))-i+2);
end

% La matriz 'MATRIZ' contiene los N1*N2 coeficientes del filtro diseñado

MATRIZ=[mat_hn2d_half_2 mat_hn2d_half_1];

% FIN DEL DISEÑO DEL FILTRO DIAMANTE DE TAMAÑO N1*N2

% Por último, representamos la respuesta en frecuencia del filtro diseñado mediante
% el uso de la NDFT

l=128;
Hw1w2=fft2(MATRIZ,l,l);

w1=[-pi:2*pi/l:pi-2*pi/l]/(2*pi);
w2=[-pi:2*pi/l:pi-2*pi/l]/(2*pi);

hold on
contour(w1,w2,fftshift(abs(Hw1w2)),9) % Gráfica de las curvas de nivel del filtro

% Localización de las muestras ind.
plot(v_muestras(:,1)/(2*pi),v_muestras(:,2)/(2*pi),'x')
axis([-0.5 0.5 -0.5 0.5])
pause
hold off

mesh(w1,w2,fftshift(abs(Hw1w2))) % Gráfica del módulo del filtro

```

FIN DEL CÓDIGO DEL PROGRAMA EN MATLAB PARA LA OBTENCIÓN DEL FILTRO DIAMANTE MEDIANTE EL USO DE LA NDFT

1.6.2 EJEMPLO: DISEÑO DE UN FILTRO ‘FAN’ (ABANICO)

Un filtro ‘fan’ de 90 grados es un filtro ‘half-band’ bidimensional cuya respuesta en amplitud se muestra en la figura 24. Los filtros de este tipo poseen una sensibilidad direccional, algo que es muy importante en algunas aplicaciones como por ejemplo el procesamiento de datos geosísmicos. Observando las figuras 19 y 24, podemos ver que la respuesta en amplitud del filtro ‘fan’ se obtiene desplazando la respuesta en amplitud del filtro diamante un factor de π a lo largo del eje w_2 . Así, si $h_d[n_1, n_2]$ es la respuesta impulsiva de un filtro diamante, podemos obtener la correspondiente al filtro ‘fan’, $h[n_1, n_2]$, mediante la expresión

$$h[n_1, n_2] = (-1)^{n_2} h_d[n_1, n_2].$$

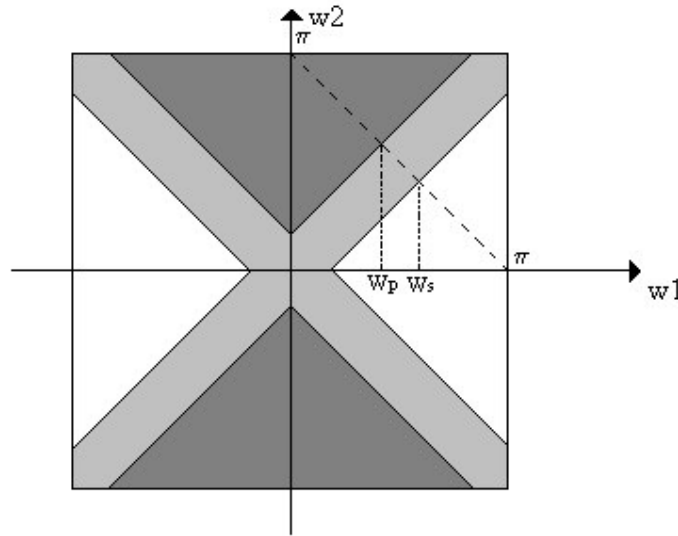


Figura 24. Especificaciones de un filtro ‘fan’. La región más oscura es la banda de paso, la región grisácea es la banda de transición y la región sin colorear es la banda de rechazo del filtro.

Puesto que el filtro diamante satisface la relación de simetría: $h_d[n_1, n_2] = h_d[n_2, n_1]$, podemos inferir la siguiente relación antisimétrica para el filtro ‘fan’:

$$h[n_1, n_2] = -h[n_2, n_1], \quad (n_1, n_2) \neq (0, 0).$$

El filtro ‘fan’ exhibe también una simetría de orden 4. Combinando este dato con la relación antisimétrica anterior y con la condición ‘half-band’, podemos expresar la respuesta en amplitud del filtro ‘fan’ de la siguiente forma

$$A(w_1, w_2) = 0.5 + \sum_{n_1=1}^{\lfloor (P+1)/2 \rfloor} 2h[2n_1-1, 0] \{ \cos(2n_1-1)w_1 - \cos(2n_1-1)w_2 \} +$$

$$+ \sum_{n_1=1}^{\lfloor (P+1)/2 \rfloor} \sum_{n_2=1}^{\lfloor P/2 \rfloor} 4h[2n_1-1, 2n_2] \{ \cos(2n_1-1)w_1 \cos(2n_2)w_2 - \cos(2n_2)w_1 \cos(2n_1-1)w_2 \}$$

Como era de esperar, el número de coeficientes independientes del filtro, N_i , es el mismo que para el filtro diamante. Podemos diseñar directamente el filtro ‘fan’ usando nuestro método de muestreo en frecuencia. Colocamos N_i muestras en un área triangular de la banda de paso, tal y como se muestra en la figura 25, puesto que es ésta la única parte independiente de la respuesta en amplitud.

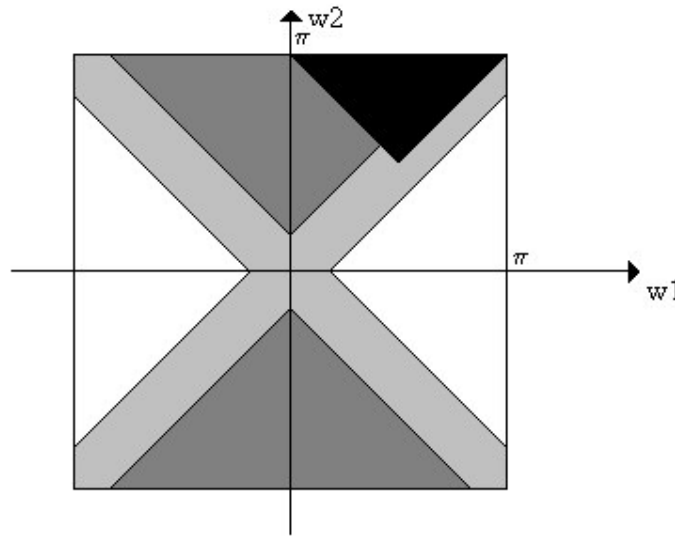


Figura 25. Respuesta en amplitud de un filtro diamante. La región más oscura es la única parte independiente debido a la simetría de orden 8 y a la naturaleza ‘half-band’ del filtro.

El proceso de generación de las muestras depende del hecho de que la sección cruzada de $A(w_1, w_2)$ a lo largo de la diagonal ($w_1 = \pi - w_2$) es como un filtro ‘half-band’ paso de baja unidimensional. El proceso de generación es el mismo que para un filtro diamante. Así, si H_k es el valor de una muestra para el filtro diamante correspondiente situada en la posición (w_{1k}, w_{2k}) , entonces H_k es el valor de una muestra para el filtro ‘fan’ en la posición $(w_{1k}, \pi - w_{2k})$. Así, se generan las N_i muestras, y después, se usa la expresión de $A(w_1, w_2)$ desarrollada para obtener los coeficientes independientes.

A continuación, mostramos un ejemplo de un filtro ‘fan’ de tamaño 9×9 , con $w_p = 0.43\pi$ y $w_s = 0.57\pi$.

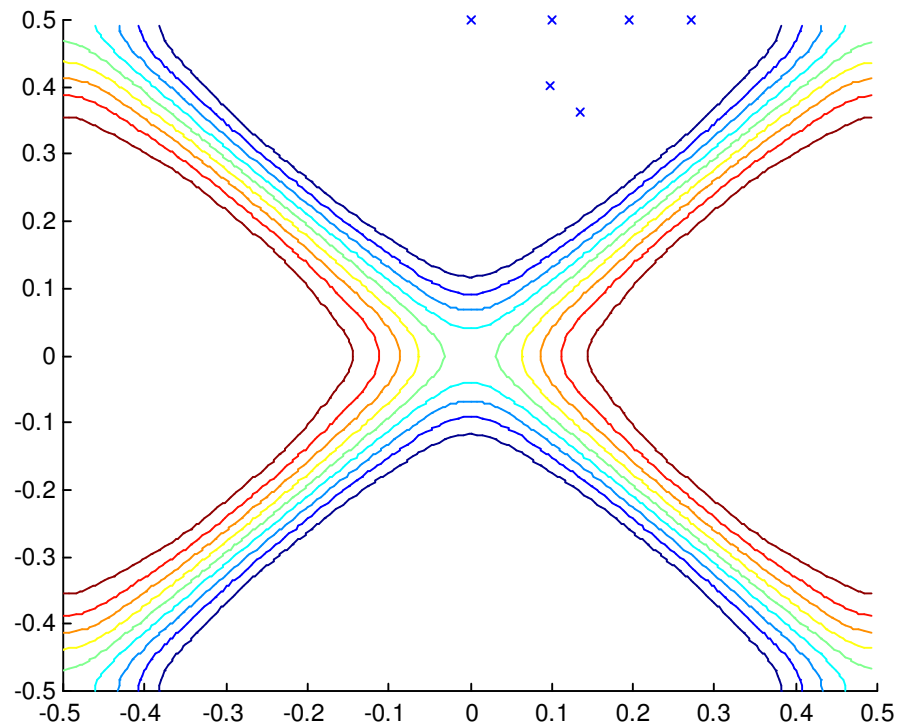


Figura 26. Filtro ‘fan’ de tamaño 9 x 9 diseñado mediante el método de la NDFT.

Gráfica que muestra las curvas de nivel del filtro.

Las muestras están colocadas en los lugares denotados por una ‘x’ en la gráfica.

Los ejes de frecuencias están normalizados por 2π .

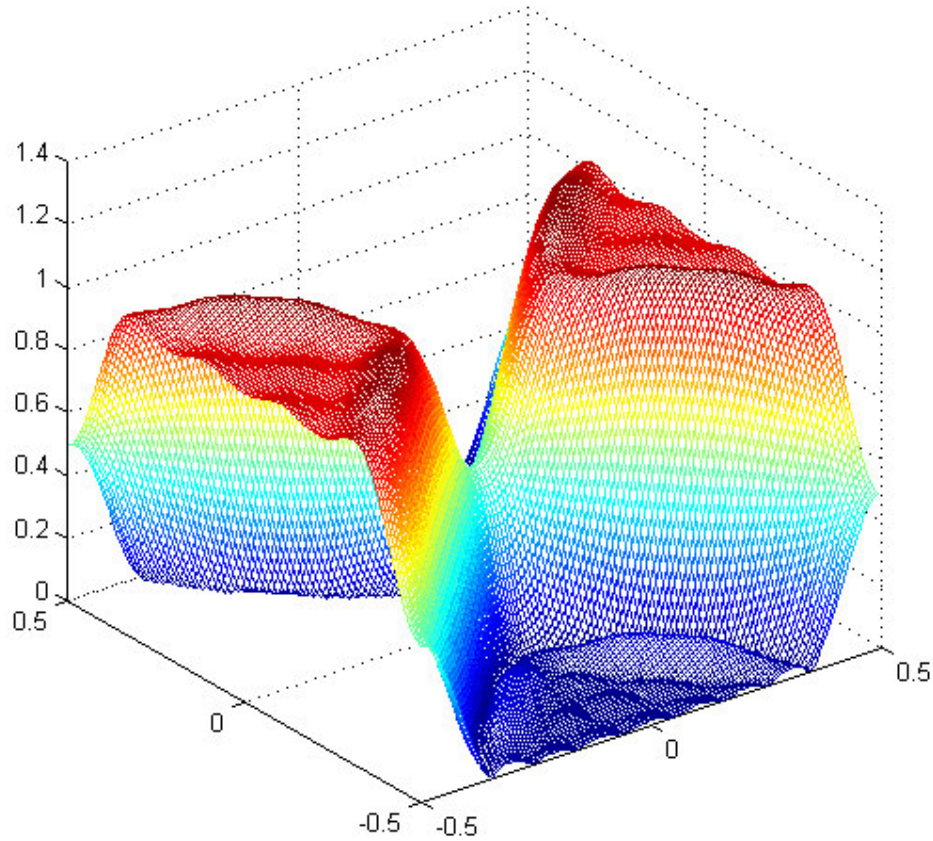


Figura 27. Filtro ‘fan’ de tamaño 9 x 9 diseñado mediante el método de la NDFT.
Dibujo de la respuesta en frecuencia del filtro.
Los ejes de frecuencias están normalizados por 2π .

1.7 APLICACIONES DE LOS FILTROS BIDIMENSIONALES

Los filtros bidimensionales tienen numerosas aplicaciones en el procesamiento digital de la señal. Son usados en el área de procesamiento de imagen para mejorar las imágenes de baja calidad, tales como las fotografías de rayos X o imágenes espaciales, y para la conversión entre diferentes estructuras de muestreo y formatos de vídeo. Tienen aplicación también como prefiltros en varios esquemas para codificación de la imagen y en bancos de filtros de sistemas subbandas de análisis y síntesis. También son usados para procesamiento multidimensional de datos que han sido obtenidos de exploraciones geofísicas, sismología, sonar, radar y radioastronomía. El aspecto principal a tener en cuenta en el uso de filtros bidimensionales en tiempo real es la gran cantidad de carga computacional y memoria que se necesitan para poder implementarlos. Recientemente, se han producido avances en la tecnología VLSI que permiten que el filtrado bidimensional en tiempo real sea factible.

1.8 RESUMEN

En este capítulo, hemos propuesto un método para diseñar filtros digitales bidimensionales mediante la utilización de la NDFT. Este método utiliza la libertad de colocación de las muestras en el plano frecuencial para generar filtros bidimensionales no separables con buenas características tanto de rizado como de forma de la banda de paso. Experimentalmente, se comprueba que los mejores resultados se obtienen al situar las muestras en contornos tales que concuerden con la forma de la banda de paso. Finalmente, hay que resaltar que este método puede ser extendido al diseño de filtros de tres dimensiones que son usados para el filtrado de señales de video.

2 DECODIFICACIÓN DE SEÑALES DTMF (DUAL-TONE MULTI-FREQUENCY) MEDIANTE EL USO DE LA NDFT

2.1 INTRODUCCIÓN

El sistema de señalización multifrecuencia de doble tono (DTMF, de Dual-Tone Multi-Frequency) está siendo ampliamente usado en todo el mundo en teléfonos de marcado por pulsación como mejora frente a la señalización clásica mediante roseta de los teléfonos más antiguos. Se consigue una velocidad de marcado mucho mayor gracias a los teléfonos de marcado por pulsación. Además de para la marcación en teléfonos, las señales DTMF están siendo usadas, cada vez más, en aplicaciones de control interactivas como por ejemplo, el buzón de voz, el correo electrónico, operaciones bancarias a través del teléfono, etc. En todas estas aplicaciones, el usuario elige una opción de un menú mandando señales DTMF desde el teclado de su teléfono.

Una señal DTMF consiste en la suma de dos tonos cuyas frecuencias están cogidas de dos grupos, mutuamente excluyentes, de frecuencias preasignadas. Para decodificar una señal DTMF, debemos extraer los dos tonos, y así, hallar el dígito que fue marcado. En este capítulo, vamos a aplicar los conceptos de la NDFT para decodificar señales DTMF. Los dos tonos serán extraídos mediante el empleo de la NDFT, la cual, servirá para evaluar el espectro de la señal DTMF en frecuencias específicas. Concretamente, usaremos el método NDFT Subbanda, ya comentado en el capítulo 2 de la primera parte de este proyecto. Si bien, partiremos de tal método para desarrollar un nuevo algoritmo específico para la detección de tonos DTMF. Mediante el uso de la información de que las frecuencias DTMF residen en la parte de bajas frecuencias del ancho de banda telefónico, podemos descartar las operaciones en altas frecuencias, y, de esta manera, conseguir importantes ahorros en cálculos sin sacrificar la calidad de la detección. También veremos una comparación entre algoritmos de detección de DTMF basados en el uso de la DFT y el propuesto aquí basado en la NDFT. Simulaciones que hemos realizado muestran que el algoritmo basado en la SB-NDFT requiere menor número de operaciones que el algoritmo basado en la SB-DFT para conseguir un determinado nivel de calidad. Puesto que la decodificación DTMF multicanal se realiza multiplexando en el tiempo los canales en un DSP, un tiempo menor en la decodificación implicará que un mayor número de canales DTMF se puedan decodificar simultáneamente. Esto es muy deseable en ambientes multicanal. En primer lugar, haremos un repaso de los estándares DTMF así como de los algoritmos de decodificación de DTMF existentes.

2.2 ANTECEDENTES

2.2.1 ESTÁNDARES DTMF

Los estándares para la señalización DTMF en teléfonos con marcado por pulsación aparece en las recomendaciones Q.23 y Q.24 del Libro Rojo de la CCITT. Este sistema es ‘dentro de banda’ puesto que las frecuencias de señalización están localizadas en la banda de frecuencias usadas para la comunicación vocal. En la recomendación Q.23, las frecuencias DTMF se asignan a los distintos dígitos y símbolos del teclado telefónico tal y como se muestra en la figura 28. Se debe advertir que las cuatro teclas de la última columna no aparecen en los teclados de los teléfonos normales. Dichas teclas están reservadas para uso futuro. Cada vez que un botón es pulsado, se genera una señal DTMF que, como ya sabemos, contiene dos frecuencias. Estas dos frecuencias pertenecen a dos grupos de frecuencias mutuamente excluyentes. Cada uno de esos dos grupos tiene cuatro frecuencias. Como se puede ver en la figura 28, una frecuencia corresponde a la fila y la otra corresponde a la columna del dígito marcado. Este código se conoce como ‘el código 2(1/4)’.

Fila ↓	Columna →			
	1209 Hz	1336 Hz	1477 Hz	1633 Hz
697 Hz	1	2	3	A
770 Hz	4	5	6	B
852 Hz	7	8	9	C
941 Hz	*	0	#	D

Figura 28. Asignación de frecuencias a los botones del teclado telefónico.

La recomendación Q.24 especifica valores operacionales para distintos parámetros técnicos, aplicables a la recepción de señales DTMF en intercambios locales. Puesto que factores tales como la pérdida de transmisión varían entre las redes de los distintos países, algunas administraciones tiene estándares ligeramente distintos. En la siguiente tabla, mostramos los parámetros más relevantes del estándar AT&T, TouchTone. Estos parámetros son muy importantes para asegurar una buena decodificación de las señales DTMF bajo condiciones de recepción extremas. Tal y como se especifica en la recomendación Q.24, el receptor DTMF debe comprobar la presencia simultánea de una y sólo una frecuencia de fila y de una y sólo una frecuencia de columna. También se exige que los receptores DTMF detecten las frecuencias operacionales con una tolerancia del $\pm 1.5\%$. Las tolerancias para los niveles de

potencia de la tabla indican que debe haber una diferencia mínima de $-25 - (-55) = 30 \text{ dBm}$ entre el nivel de los tonos operacionales y los no operacionales. La diferencia de potencia entre los tonos operacionales se conoce como *twist*. Así, hablamos de *twist normal* cuando el nivel de potencia del tono de columna es menor que el del tono de fila, debido a la atenuación de las altas frecuencias en la red telefónica. Por el contrario, llamamos *twist inverso* al que se produce cuando el nivel de potencia del tono de columna es superior al nivel del tono de fila. Los niveles máximos de twist permitidos son 8 dB para el twist normal y 4 dB para el twist inverso.

Parámetros		Valores
Frecuencias de las señales DTMF	Grupo Bajo	697, 770, 852, 941 Hz
	Grupo Alto	1209, 1336, 1477, 1633 Hz
Tolerancia en frecuencia $ \Delta f $	Operacional	$\leq 1.5\%$
	No Operacional	$\geq 3.5\%$
Niveles de potencia por frecuencia	Operacional	0 a -25 dBm
	No Operacional	Máx. -55 dBm
Diferencia de potencia entre frec.		+ 4 dB a -8 dB
Duración señal	Operacional	Mín. 40 ms
	No Operacional	Máx. 23 ms

2.2.2 IMPLEMENTACIÓN DIGITAL

Aunque existen chips de circuitos analógicos disponibles para la generación y la decodificación de señales DTMF en un único canal, estas funciones pueden ser también implementadas digitalmente en DSPs. Esta implementación digital mejora las prestaciones de la implementación analógica puesto que proporciona mayor precisión, estabilidad, versatilidad, reprogramabilidad, y la opción de operación multicanal. Algunos ejemplos de implementación digital usan DSPs como el TMS32010, WE DSP-32, o el ADSP-2100. Los chips codec (codificador/decodificador) o los convertidores analógico-digital y digital-analógico (A/D y D/A) proporcionan la interfaz analógica necesaria.

La generación digital de una señal DTMF es bastante simple puesto que supone simplemente añadir muestras digitales de dos sinusoides, las cuales son generadas mediante tablas 'look-up' o calculando una expresión polinómica.

La decodificación de señales DTMF es un problema más complicado de resolver. En los circuitos analógicos, la detección de un tono se realiza frecuentemente detectando y contando los pasos por cero de la señal de entrada. En el caso digital, la detección de un tono se puede realizar fácilmente transformando las muestras de entrada al dominio de la frecuencia mediante el uso de la transformada de Fourier. Esta transformación se puede efectuar de distintas maneras, como comentaremos más tarde. Sin embargo, sea cual sea la transformación usada, se obtiene finalmente un conjunto de muestras en el dominio de la frecuencia que miden la energía presente en las ocho frecuencias DTMF. El siguiente paso será discernir si los tonos detectados constituyen un dígito DTMF válido o no. Esto se consigue realizando algunos tests de validación, tales como:

- ✓ *Test de magnitud.* El tono de fila mayor y el tono de columna mayor deben estar por encima de un cierto umbral.
- ✓ *Test de no otros tonos.* Este test comprueba los niveles de potencia de los demás tonos de fila y columna. El resultado de dividir la magnitud de cada uno de ellos por la magnitud del tono de fila o columna mayor, según el grupo al que pertenezcan, debe ser menor que un cierto umbral.
- ✓ *Test de twist.* Las magnitudes de los tonos principales de fila y columna se comparan entre sí dividiendo la magnitud del tono de fila por la magnitud del tono de columna. Para twist normal, esta relación debería ser menor que un cierto umbral; para twist inverso debería ser mayor que un cierto umbral.
- ✓ *Test de energía total.* La energía total presente se calcula y se compara con la suma de las energías de los tonos de fila y columna mayores. Este previene de que un dígito pueda ser simulado a través de las señales de voz. Para una señal DTMF, estos dos valores son equivalentes. Sin embargo, para señales de voz, la energía total es mucho mayor.
- ✓ *Test de armónicos.* Este test también sirve para evitar simulación de dígitos. Se calculan las energías de los segundos armónicos de los tonos principales de fila y columna. Estos armónicos son prácticamente despreciables para una señal DTMF pero bastante altos, normalmente, para una señal vocal.

Si todos estos tests se superan, entonces el dígito correspondiente a la señal DTMF tratada es decodificado. Generalmente, un dígito es declarado válido si los dígitos decodificados en dos tramas sucesivas de la señal son el mismo.

El proceso de decodificación es, por tanto, realizado en dos pasos: transformación y validación. Nosotros nos vamos a centrar en la parte de la transformación usada para convertir la señal al dominio frecuencial. Sea N el número de muestras necesarias para detectar adecuadamente todos los dígitos DTMF bajo las condiciones de recepción recogidas en el estándar AT&T de la tabla anterior. Puesto

que la duración mínima de una señal DTMF es 40 ms, y la tasa de muestreo es 8 kHz, el número de muestras disponibles para decodificar cada dígito DTMF es $0.04 \times 8000 = 320$. Por tanto, tenemos que

$$N \leq 320.$$

El valor de N que se necesite dependerá del método de transformación usado. Una vez que N haya sido encontrado, los umbrales requeridos para la validación se pueden determinar considerando el peor caso de potencia de la señal bajo condiciones de recepción extremas. Compararemos los resultados de dos métodos de transformación en este capítulo. Concretamente, dos algoritmos para decodificación de DTMF basados en la SB-DFT y en la SB-NDFT.

2.2.3 ALGORITMOS EXISTENTES PARA LA DECODIFICACIÓN DE SEÑALES DTMF

La DFT ha sido usada para medir la energía de la señal DTMF recibida en los ocho tonos de frecuencia. Decodificadores de DTMF basados en la DFT están implementados para el TMS32010 y el ADSP-2100. Se calculan ocho muestras de la DFT, $X[k]$, que son las más próximas en frecuencia a los ocho tonos DTMF. El algoritmo de Goertzel necesita un filtro recursivo de segundo orden para cada frecuencia, y se usa para conseguir una implementación eficiente. El cálculo es ligeramente modificado puesto que sólo necesitamos realmente la magnitud al cuadrado de $X[k]$. Este algoritmo es un caso especial del algoritmo de Goertzel modificado que comentamos en el capítulo 2 de la primera parte del presente proyecto, donde $w_k = 2\pi k/N$. El problema de usar la DFT es que las frecuencias DTMF reales no coinciden exactamente con las muestras equiespaciadas de la DFT. Por tanto, debido a esta falta de coincidencia se introduce un cierto error.

En un algoritmo mejorado, este problema ha sido superado usando muestras en las frecuencias exactas de la DTMF. Este método es equivalente a calcular muestras NDFT en esas frecuencias y fue usado en una implementación para la familia WE DSP32. De nuevo, la magnitud al cuadrado de las muestras de la NDFT se calculan usando el algoritmo de Goertzel modificado que ya vimos. Por tanto, puesto que no hay error en cuanto a falta de coincidencia entre las muestras y las frecuencias reales DTMF, el algoritmo basado en la NDFT consigue mejores resultados que el basado en la DFT. Cada uno de estos algoritmos, necesitan de $N + 4$ multiplicaciones reales y de $2N + 2$ sumas reales para cada frecuencia, dando lugar a un total de $8N + 32$ multiplicaciones reales y $16N + 16$ sumas reales para las ocho frecuencias.

2.3 ALGORITMO PROPUESTO DE DECODIFICACIÓN DE SEÑALES DTMF USANDO LA NDFT SUBBANDA

Vamos a proponer un nuevo algoritmo para decodificar las señales DTMF basado en la NDFT Subbanda (de la que ya hablamos en el capítulo 2 de la primera parte del presente documento). Puesto que la tasa de muestreo usada en telefonía es de $f_s = 8kHz$, todas las frecuencias DTMF están localizadas en la banda de bajas frecuencias y por tanto, podemos obtener una buena aproximación de la NDFT mediante el uso de la NDFT Subbanda. De esta forma, descartaremos los cálculos para la banda de altas frecuencias. Este hecho reduce el número de operaciones necesarias a casi la mitad.

Consideremos una secuencia $x[n]$ con un número par de muestras, N . Tal y como se comentó en el capítulo 2 de la primera parte, podemos descomponer $x[n]$ en dos subsecuencias, $g_L[n]$ y $g_H[n]$, de longitud $N/2$ cada una, dadas por

$$g_L[n] = \frac{1}{2} \{x[2n] + x[2n+1]\},$$

$$g_H[n] = \frac{1}{2} \{x[2n] - x[2n+1]\},$$

con $n = 0, 1, \dots, N/2-1$. La NDFT de $x[n]$ se puede expresar, como ya vimos:

$$X(z_k) = (1 + z_k^{-1})G_L(z_k^2) + (1 - z_k^{-1})G_H(z_k^2),$$

donde $G_L(z_k^2)$ y $G_H(z_k^2)$ son las NDFTs de las subsecuencias, $g_L[n]$ y $g_H[n]$, evaluadas en los puntos $z = z_k^2$. Puesto que los ocho tonos DTMF están localizados en la banda de bajas frecuencias, podemos obtener una aproximación bastante buena de la NDFT si eliminamos el término de alta frecuencia $G_H(z_k^2)$. Nos queda entonces que

$$\hat{X}(z_k) = (1 + z_k^{-1})G_L(z_k^2), \quad k = 0, 1, \dots, 7,$$

donde $z_k = e^{jw_k}$, $w_k = 2\pi f_k / f_s$, y $f_0, f_1, \dots, f_7$ son las ocho frecuencias DTMF. Se advierte que el número de cálculos al aplicar esta aproximación es casi la mitad que para el caso general. La eliminación de la banda de altas frecuencias provoca una distorsión lineal de las muestras NDFT debido a una respuesta en frecuencia no constante en la banda de bajas frecuencias. Tal y como vimos en el capítulo 2 de la primera parte, podemos compensar esa distorsión lineal mediante un factor de escalado, así, escribimos

$$X(e^{jw_k}) = 2b_k \hat{X}(e^{jw_k}),$$

donde el parámetro b_k se define como

$$b_k = \frac{1}{1 + \cos w_k}.$$

Para la decodificación DTMF, sólo necesitamos las magnitudes al cuadrado de las muestras NDFT aproximadas. Por tanto:

$$|X(e^{jw_k})|^2 = 4b_k^2 |\hat{X}(e^{jw_k})|^2.$$

Operando, obtenemos

$$|\hat{X}(e^{jw_k})|^2 = |(1 + e^{-jw_k})|^2 |G_L(e^{j2w_k})|^2 = \frac{2}{b_k} |G_L(e^{j2w_k})|^2.$$

Llegando finalmente a:

$$|X(e^{jw_k})|^2 = 8b_k |G_L(e^{j2w_k})|^2.$$

El valor de $|G_L(e^{j2w_k})|^2$ se calcula usando el algoritmo modificado de Goertzel. Un breve resumen de este procedimiento se muestra a continuación. Expresemos $G_L(z_k^2)$ como

$$G_L(z_k^2) = \sum_{r=0}^{N/2-1} g_L[r] z_k^{-2r} = z_k^{-N} y_k[n] \Big|_{n=N/2},$$

donde $y_k[n]$ es el resultado de la convolución discreta

$$y_k[n] = g_L[n] \otimes z_k^{2n} u[n] = \sum_{r=0}^{N/2-1} g_L[r] z_k^{2(n-r)} u[n-r],$$

donde $u[n]$ denota la secuencia escalón unidad. Esto es equivalente a un filtro recursivo de segundo orden con una función de transferencia del tipo

$$H_k(z) = \frac{1}{1 - z_k^2 z^{-1}} = \frac{1 - e^{-j2w_k} z_k^{-1}}{1 - a_k z^{-1} + z^{-2}},$$

donde $z_k = e^{jw_k}$, y el parámetro a_k se define como

$$a_k = 2 \cos 2w_k.$$

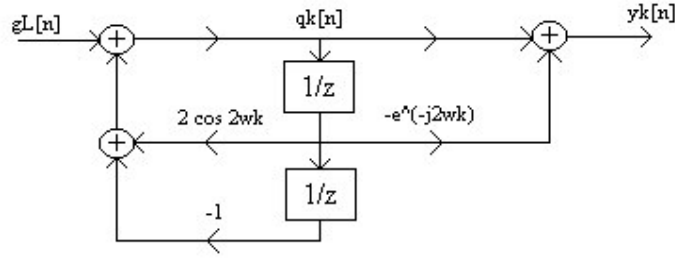


Figura 29. Algoritmo de Goertzel visto como un filtro digital recursivo de segundo orden.

Este sistema se muestra en la figura 29. Podemos calcular $y_k[N/2]$ calculando las señales intermedias $q_k[n]$ dadas por la ecuación en diferencias

$$q_k[n] = a_k q_k[n-1] - q_k[n-2] + g_L[n], \quad n = 0, 1, \dots, \infty,$$

con las condiciones iniciales

$$q_k[-1] = q_k[-2] = 0.$$

Puesto que $g_L[n] = 0$ para $n < 0$ y $n \geq N/2$, resulta que

$$y_k[N/2] = q_k[N/2] - e^{-j2w_k} q_k[N/2-1].$$

Tomando las magnitudes al cuadrado de la expresión anterior, obtenemos

$$|y_k[N/2]|^2 = q_k^2[N/2] + q_k^2[N/2-1] + a_k q_k[N/2] q_k[N/2-1].$$

Por otro lado, se tiene que

$$|G_L(e^{j2w_k})|^2 = |y_k[N/2]|^2.$$

Combinando las dos expresiones anteriores, llegamos a que la magnitud al cuadrado que buscamos vendrá dada por

$$|G_L(e^{j2w_k})|^2 = q_k^2[N/2] + q_k^2[N/2-1] + a_k q_k[N/2] q_k[N/2-1].$$

La figura 30 muestra las operaciones implicadas en este algoritmo. Hay que resaltar que el factor de $1/2$ de la expresión de $g_L[n]$ y el factor de 8 de la expresión de $|X(e^{jw_k})|^2$ se pueden despreciar en los cálculos porque solamente tienen importancia las magnitudes al cuadrado relativas de los ocho tonos DTMF. En este algoritmo, sólo necesitamos dos coeficientes, a_k y b_k , para cada frecuencia DTMF. Tal y como se

muestra en la figura 30, a_k es un multiplicador usado en el algoritmo de Goertzel, y b_k es un factor de escalado para la salida.

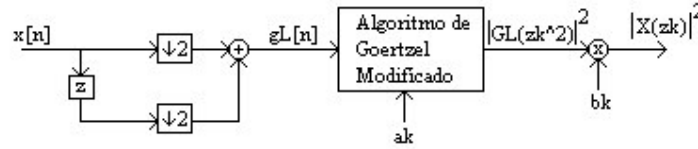


Figura 30. Algoritmo de decodificación de señales DTMF usando la NDFT Subbanda.

Para determinar la complejidad computacional del algoritmo propuesto, resumamos los pasos de dicho algoritmo:

- ✓ Obtener la subsecuencia $g_L[n]$ de la secuencia de entrada $x[n]$. Esto requiere $N/2$ sumas reales.
- ✓ Para cada tono DTMF, calculamos la magnitud al cuadrado de la NDFT de $g_L[n]$, evaluada en dos veces la frecuencia del tono, usando el algoritmo de Goertzel modificado. Esto requiere $N/2 + 4$ multiplicaciones reales y $N + 2$ sumas reales.
- ✓ Para cada tono DTMF, multiplicar la magnitud al cuadrado por un factor de escalado b_k . Esto requiere una multiplicación real.

Por tanto, el número total de operaciones requeridas para los ocho tonos DTMF es $4N + 40$ multiplicaciones reales y $8.5N + 16$ sumas reales. Puesto que los métodos existentes basados en la DFT y en la NDFT necesitan $8N + 32$ multiplicaciones reales y $16N + 16$ sumas reales para los ocho tonos, el algoritmo propuesto basado en la SB-NDFT tiene mucha menor carga computacional (casi la mitad).

Por simple observación del método propuesto, se puede advertir que podemos proponer otro algoritmo de decodificación de DTMF basado en la SB-DFT. Éste sería similar al aquí descrito salvo que ahora $w_k = 2\pi k/N$. Se deben elegir ocho valores enteros para k tales que correspondan a las frecuencias kf_s/N más próximas a las frecuencias DTMF. Aunque la complejidad computacional de este nuevo algoritmo basado en la SB-DFT es la misma que para el basado en la SB-NDFT, el error producido por el muestreo uniforme lleva a peores resultados tal y como veremos en los apartados siguientes.

2.4 RESULTADOS Y COMPARACIONES

En este apartado, nuestra intención es encontrar el mínimo número de muestras, N , necesario para detectar adecuadamente los dígitos DTMF bajo las condiciones de recepción especificadas en el estándar TouchTone. Simulando tales condiciones, compararemos los resultados de los dos algoritmos de decodificación aquí propuestos; esto es, el basado en la SB-NDFT y el basado en la SB-DFT. Comenzaremos definiendo una figura de mérito, que será usada como una medida de calidad. Después, comentaremos algunos factores que afectan al rendimiento de los algoritmos de detección de DTMF. Una comprensión de estos factores es esencial para interpretar los resultados de los algoritmos de detección. Por último, presentaremos los resultados de las simulaciones, que han sido usadas para encontrar el mínimo número de muestras necesario para cada algoritmo para obtener un cierta figura de mérito.

2.4.1 FIGURA DE MÉRITO

Para comparar el rendimiento de los algoritmos propuestos debemos definir una medida de calidad. Suponiendo que el dígito DTMF marcado fue decodificado con éxito, la *figura de mérito* corresponde al mínimo de las diferencias entre los niveles de potencia (dB) de los tonos mayor y segundo mayor de las filas y columnas. Matemáticamente:

$$F = \min(D_r, D_c),$$

donde

$$D_r = 10\log_{10} X_r - 10\log_{10} X_{rs},$$

$$D_c = 10\log_{10} X_c - 10\log_{10} X_{cs}.$$

X_r y X_c son los valores máximos de entre las energías detectadas en los tonos de las filas y columnas respectivamente; X_{rs} y X_{cs} son los segundos valores mayores de entre las energías detectadas en los tonos de las filas y columnas respectivamente. Claramente, para una decodificación correcta se necesita un valor positivo de la figura de mérito ($F > 0$). Es deseable contar con una figura de mérito elevada que garantice que los tonos operacionales sean detectados en niveles de potencia mucho mayores que los tonos no operacionales. Normalmente, el número de muestras a la entrada es elegido de tal forma que la figura de mérito en el peor caso sea en torno a 6 dB.

2.4.2 FACTORES QUE AFECTAN AL RENDIMIENTO DE LOS ALGORITMOS DE DETECCIÓN

Comentaremos, brevemente, los factores que afectan al rendimiento de los algoritmos de detección de DTMF. Las señales DTMF consisten en la suma de dos sinusoides. La entrada al algoritmo de detección es un segmento enventanado de la señal muestreada, dada por

$$v[n] = x[n]w[n],$$

donde

$$x[n] = A_r \cos w_r n + A_c \cos w_c n,$$

y

$$w[n] = \begin{cases} 1, & 0 \leq n \leq N-1, \\ 0, & \text{en otro caso} \end{cases}.$$

Aquí, A_r y A_c son las amplitudes, y w_r y w_c son las frecuencias de las sinusoides fila y columna respectivamente. La secuencia $w[n]$ es una ventana rectangular, con una transformada de Fourier en tiempo discreto

$$W(e^{jw}) = \sum_{n=0}^{N-1} e^{-jwn} = e^{-jw\frac{N-1}{2}} \frac{\text{sen}(wN/2)}{\text{sen}(w/2)}.$$

La transformada de Fourier en tiempo discreto de una senoide $A \cos w_0 n$ es un par de impulsos en $+w_0$ y en $-w_0$. El efecto de enventanar puede ser estudiado analizando la transformada de Fourier, $V(e^{jw})$, de la secuencia enventanada $v[n]$, que viene dada por

$$V(e^{jw}) = \frac{A_r}{2} W(e^{j(w-w_r)}) + \frac{A_r}{2} W(e^{j(w+w_r)}) + \frac{A_c}{2} W(e^{j(w-w_c)}) + \frac{A_c}{2} W(e^{j(w+w_c)}),$$

que consiste en la transformada de Fourier de la ventana rectangular repetida en las frecuencias $\pm w_r$ y $\pm w_c$ y escalada por las amplitudes A_r y A_c , de las sinusoides correspondientes, respectivamente. La figura de mérito obtenida dependerá de las condiciones de recepción así como del número de muestras de la señal de entrada, N .

Los valores relativos de las amplitudes de los tonos fila y columna en la señal DTMF recibida dependerá de la existencia o no de *twist*, que, como ya se comentó, es la diferencia de potencia entre los dos tonos operacionales. Para considerar el efecto del

twist, supongamos que el dígito '1' ha sido marcado, y, por tanto, las frecuencias $f_r = 697 \text{ Hz}$ y $f_c = 1209 \text{ Hz}$ han sido generadas. La figura 31 muestra la magnitud de la transformada de Fourier, $V(e^{j\omega})$, con $N = 100$, bajo tres condiciones de twist – no twist ($A_r = A_c$), twist inverso máximo (A_c mayor que A_r en 4 dB), twist normal máximo (A_c menor que A_r en 8 dB), respectivamente. De estos dibujos se saca la clara conclusión que el twist afecta enormemente al rendimiento del algoritmo de decodificación. Generalmente, la figura de mérito más baja se obtiene para el caso de máximo twist normal. Esto ocurre porque la altura del lóbulo principal del tono columna es relativamente pequeña, estando muy cerca de las alturas de los lóbulos secundarios más altos de los tonos columna. En consecuencia, se detectan altos valores en los tonos columna no operacionales y la figura de mérito decrece.

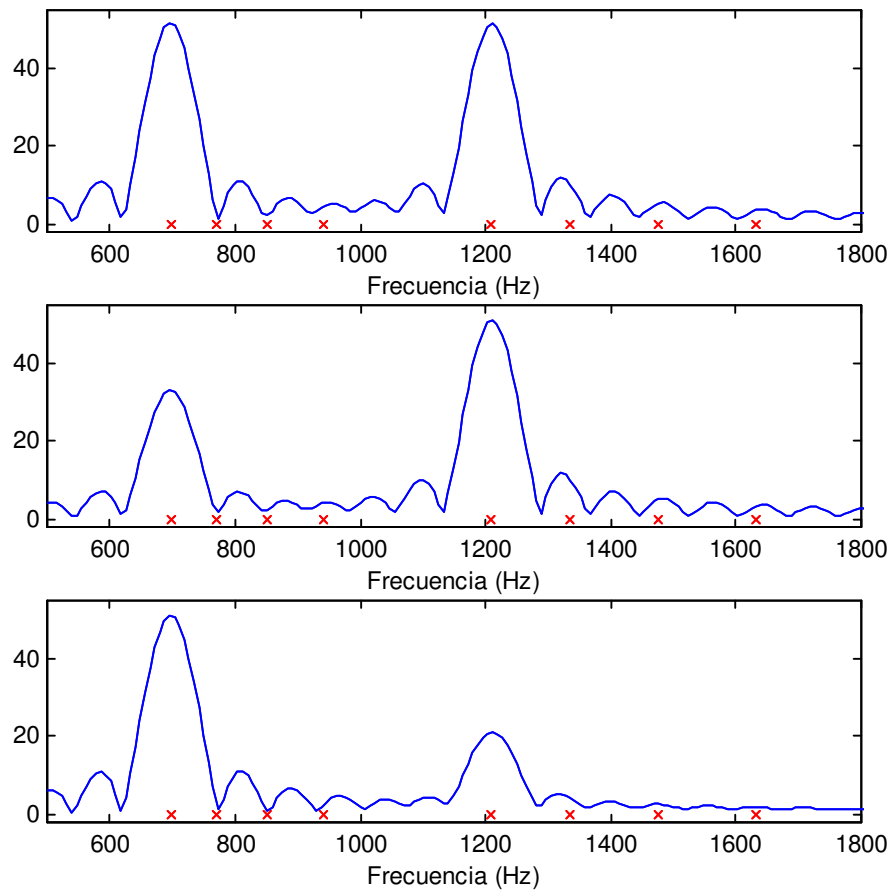


Figura 31. Magnitud de la transformada de Fourier de una señal DTMF con 100 muestras, generada cuando el dígito '1' es marcado. (Las ocho frecuencias DTMF aparecen con una 'x' en el eje de frecuencias). De arriba a abajo, no twist, máximo twist inverso y máximo twist normal.

Si vamos aumentando el número de muestras de entrada, N , la anchura del lóbulo principal ($\Delta\omega = 4\pi/N$) en los tonos operacionales decrece. Esto se muestra en la figura 32, para $N = 320$, que es el máximo número de muestras disponible para esta aplicación tal y como se comentó al inicio de este capítulo. Si comparamos esta figura

con la figura para $N = 100$, notaremos que para $N = 320$, los lóbulos secundarios están prácticamente unidos al lóbulo principal – esto es deseable puesto que ahora se detectarán niveles más bajos en las frecuencias no operacionales, y por tanto, la figura de mérito crecerá.

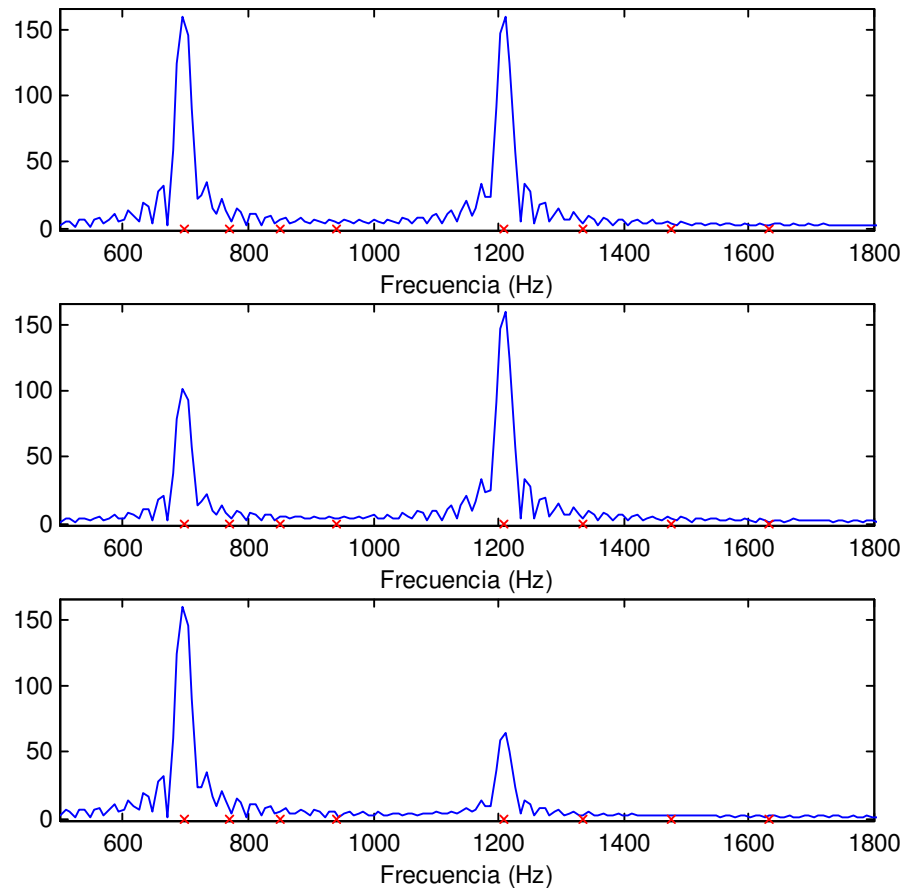


Figura 32. Magnitud de la transformada de Fourier de una señal DTMF con 320 muestras, generada cuando el dígito '1' es marcado. (Las ocho frecuencias DTMF aparecen con una 'x' en el eje de frecuencias). De arriba a abajo, no twist, máximo twist inverso y máximo twist normal.

Sin embargo, hay un *límite superior* en cuanto al número de muestras de la señal de entrada que se puede usar sin que decrezca la figura de mérito en el peor caso. Este límite surge debido a la tolerancia en frecuencia recogida en el estándar. Así, se exige que un receptor DTMF sea capaz de detectar tonos operacionales que se encuentren dentro de unas bandas con extremos $\pm 1.5\%$ alrededor de las frecuencias DTMF exactas. Es por ello que, si estamos muestreando el espectro en el centro de la banda y el tono en cuestión se mueve hacia el borde de esa banda, no podremos detectar la presencia del tono si el lóbulo principal es demasiado estrecho, y esto ocurre cuando N es grande. Esta situación se muestra en la figura 33. En el caso de la figura, $N = 320$ y el dígito pulsado es 'D'. Vemos que el tono columna está situado en el extremo inferior de la octava banda. Puesto que la muestra tomada en el centro de esa banda cae fuera del lóbulo principal, el tono columna operacional no será detectado y la detección fallará.

Por tanto, se infiere que un aumento en el número de las muestras provoca que la figura de mérito crezca en una primera etapa, pero, si ese aumento continúa, la figura de mérito comenzará a decrecer a partir de un cierto momento. Esto se muestra en la figura 34, la cual nos enseña cómo un aumento inicial del número de muestras produce un incremento de la figura de mérito, pero, si el aumento sigue, la figura de mérito comienza a decaer.

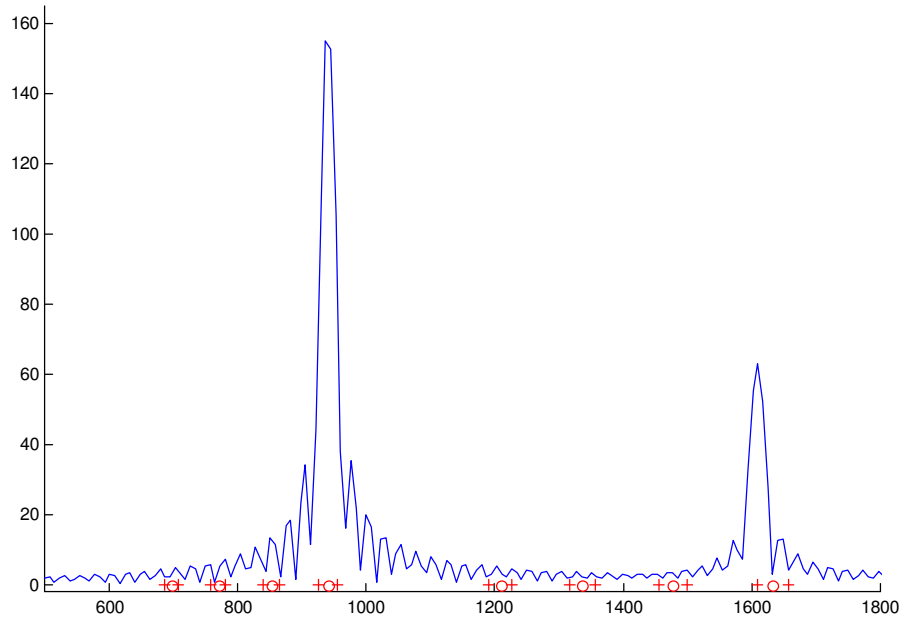


Figura 33. Efecto de la tolerancia en frecuencia en la detección de DTMF: Esta figura muestra la magnitud de la transformada de Fourier de una señal DTMF con 320 muestras, generada cuando el dígito 'D' es marcado. Las ocho frecuencias DTMF pueden variar dentro de las bandas delimitadas por '+', alrededor del centro denotado por 'o'. Aquí, el tono columna es recibido en el borde inferior de la octava a banda, y, al muestrear en el centro de la misma, la detección falla.

Por último, consideraremos el efecto de la presencia de tonos no operacionales, cuyos niveles serán, al menos, 30 dB, por debajo del más bajo de los dos tonos operacionales. La figura 34 muestra un ejemplo cuando el dígito '1' es pulsado. Para diferentes valores de N, el número de muestras de la secuencia de entrada, la figura de mérito F se obtiene calculando muestras de la transformada de Fourier enventanada de la señal DTMF en las ocho frecuencias DTMF. La línea azul representa el escenario cuando la señal DTMF consta únicamente de los tonos operacionales fila y columna correspondientes al dígito '1'. La línea roja corresponde al caso en el que se añaden seis sinusoides más, con amplitudes 30 dB por debajo de las amplitudes de las sinusoides de los tonos operacionales, correspondientes a los tonos no operacionales. Se puede observar que, en presencia de estos tonos no operacionales, la figura de mérito puede resultar por encima o por debajo de la figura de mérito para el caso de no existencia de tonos no operacionales, dependiendo de la longitud de la ventana N.

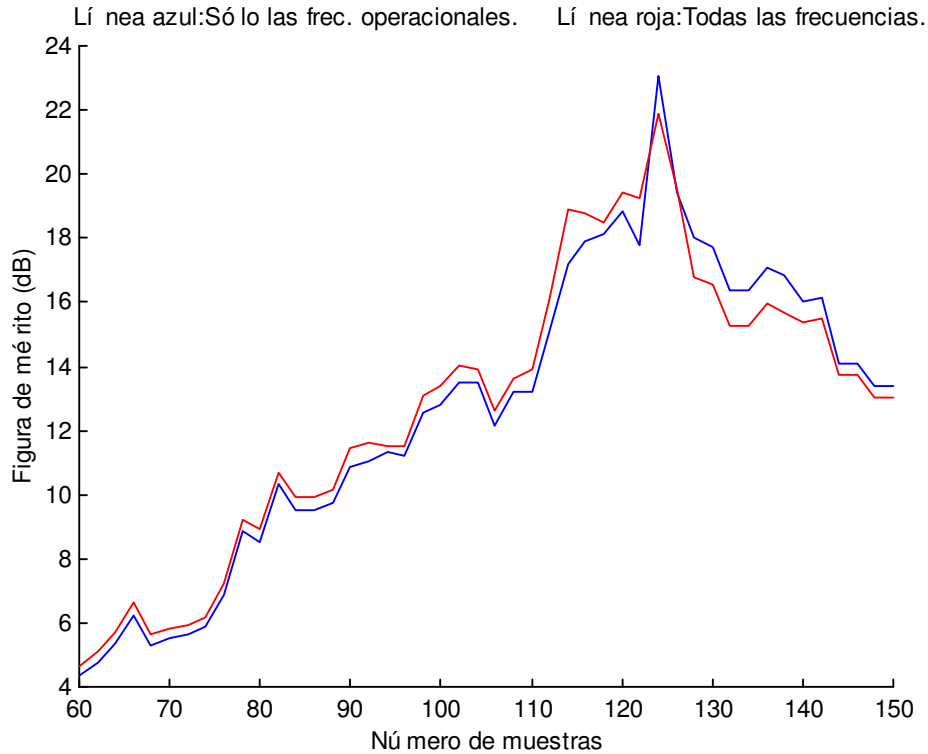


Figura 34. Efecto de la presencia de tonos no operacionales en la detección de DTMF: Variación de la figura de mérito con el número de muestras de la señal DTMF, cuando el dígito '1' es pulsado. La línea azul muestra el caso cuando sólo las dos sinusoides de los tonos operacionales están presentes en la señal. La línea roja muestra el caso en el que la señal también incluye sinusoides, 30 dB por debajo en amplitud con respecto a las sinusoides fundamentales, en las 6 frecuencias no operacionales.

2.4.3 COMPARACIÓN DEL RENDIMIENTO

En este apartado, vamos a comparar el rendimiento de dos algoritmos de decodificación de DTMF: el algoritmo propuesto basado en la SB-NDFT y el algoritmo 'modificado' basado en la SB-DFT. Así, calcularemos la figura de mérito, F , alcanzada por cada algoritmo para un número dado de muestras de entrada, N . Para cada valor de N , la figura de mérito en el peor caso se determina considerando los 16 dígitos DTMF, cada uno de los cuales bajo las tres condiciones de twist (no twist, twist inverso máximo y twist normal máximo), y quedándonos con el valor de F más bajo. La señal DTMF será generada añadiendo las dos sinusoides en las frecuencias operacional fila y columna, y seis sinusoides más con amplitudes 30 dB por debajo de las dos primeras en las seis frecuencias DTMF restantes. Hemos realizado las simulaciones considerando dos casos:

- ✓ Caso 1. Los tonos operacionales w_r y w_c están fijados a las frecuencias DTMF estándares.
- ✓ Caso 2. Las frecuencias operacionales varían dentro de unas bandas que comprenden $\pm 1.5\%$ alrededor de las frecuencias estándares.

En el caso 2, simulamos la condiciones de recepción especificadas en el estándar AT&T TouchTone variando las dos frecuencias operacionales dentro de las bandas de tolerancia, con extremos en $\pm 1.5\%$ alrededor de las frecuencias estándares. Concretamente, se aplican, para cada banda de tolerancia, frecuencias equidistantes con un incremento constante de 5 Hz. De esta manera, tenemos 5, 5, 5, 5, 7, 9, 9, 9 posibles elecciones de frecuencia en las 8 bandas de tolerancia, respectivamente (puesto que las anchuras de estas bandas son 20.9, 23.1, 25.6, 28.2, 36.3, 40.1, 44.3, 49 Hz). Para las posibles elecciones de frecuencias operacionales DTMF, se calcula la figura de mérito en el peor caso.

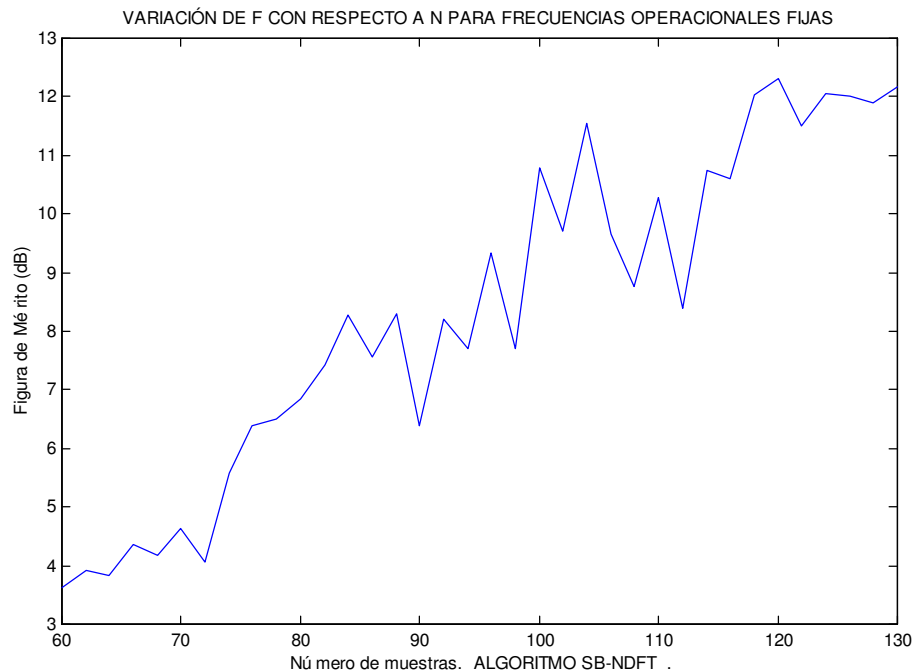


Figura 35. Variación de la figura de mérito con el número de muestras, para frecuencias operacionales fijas (caso 1). Algoritmo SB-NDFT.



Figura 36. Variación de la figura de mérito con el número de muestras, para frecuencias operacionales fijas (caso 1). Algoritmo SB-DFT.

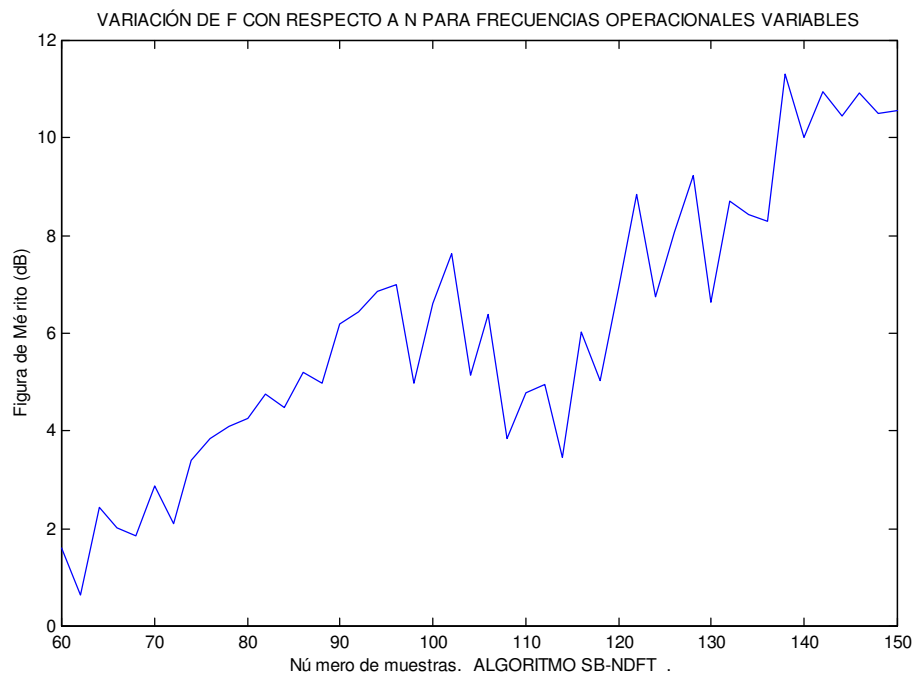


Figura 37. Variación de la figura de mérito con el número de muestras, para frecuencias operacionales variables (caso 2). Algoritmo SB-NDFT.

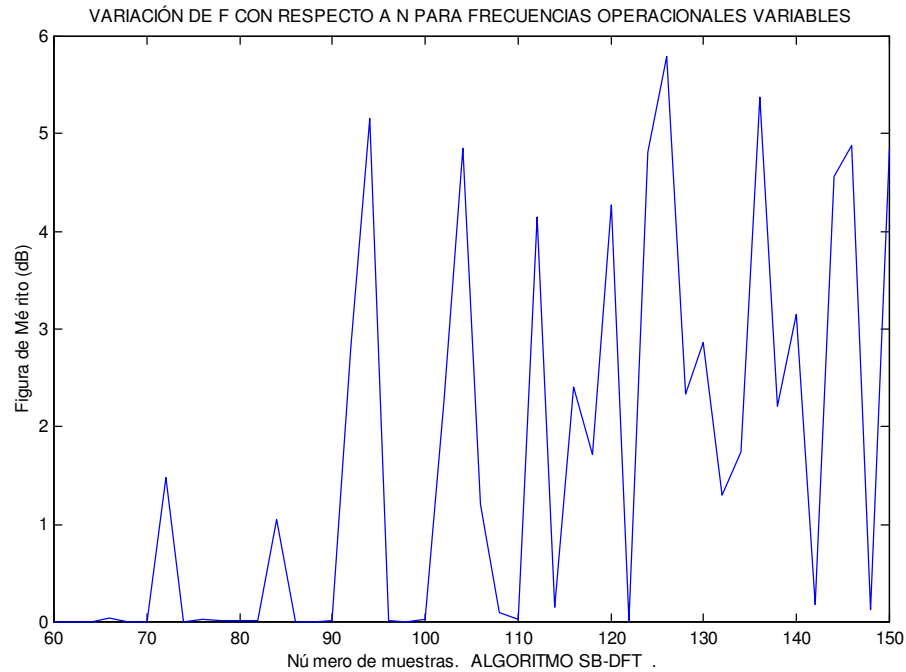


Figura 38. Variación de la figura de mérito con el número de muestras, para frecuencias operacionales variables (caso 2). Algoritmo SB-DFT.

Las figuras 35, 36, 37 y 38 muestran la variación de la figura de mérito en el peor caso con el número de muestras para los casos 1 (figs. 35 y 36) y 2 (figs. 37 y 38) , usando los dos algoritmos de decodificación de DTMF ya citados – SB-NDFT y SB-DFT.

En vista de los requerimientos en cuanto a tolerancia en frecuencia del estándar DTMF, el caso 2 es de alto interés práctico para elegir un algoritmo de decodificación. Como se puede observar, el algoritmo SB-DFT tiene un rendimiento muy pobre en el caso 2 mientras que el algoritmo SB-NDFT presenta unos resultados bastante más satisfactorios para el caso 2 estando, además, casi al mismo nivel que para el caso 1. En estas gráficas, un cero en el valor de F significa que la detección falla. Vemos como el algoritmo SB-DFT falla para algunos valores intermedios de N (en ambos casos, aunque sobre todo para el caso 2). Esto es producido debido a la falta de coincidencia entre las muestras en frecuencia de la DFT y las frecuencias reales DTMF, causada por el muestreo uniforme intrínseco a la DFT.

Ahora vamos a comparar los dos algoritmos de manera cuantitativa determinando el mínimo número de muestras, N, necesarias para conseguir una cierta figura de mérito. Esto se muestra en las tablas siguientes para los casos 1 y 2 respectivamente, y para dos valores de la figura de mérito, 6 dB y 9 dB.

ALGORITMO	Mínimo número de muestras necesarias	
	F = 6 dB	F = 9 dB
SB-NDFT	75	96
SB-DFT	92	102

Tabla Caso 1. Mínimo número de muestras necesarias para conseguir unas figuras de mérito de valores 6dB y 9dB mediante los algoritmos de decodificación de DTMF basados en la SB-NDFT y en la SB-DFT. Frecuencias operacionales fijas.

ALGORITMO	Mínimo número de muestras necesarias	
	F = 6 dB	F = 9 dB
SB-NDFT	90	128
SB-DFT	-	-

Tabla Caso 2. Mínimo número de muestras necesarias para conseguir unas figuras de mérito de valores 6dB y 9dB mediante los algoritmos de decodificación de DTMF basados en la SB-NDFT y en la SB-DFT. Frecuencias operacionales variables. La entrada “ - ” indica que esa figura de mérito no se alcanza nunca para ningún N.

ALGORITMO	F = 6 dB Caso 1		F = 9 dB Caso 1		F = 6 dB Caso 2		F = 9 dB Caso 2	
	M	S	M	S	M	S	M	S
SB-NDFT	340	654	424	832	400	781	552	1104
SB-DFT	408	798	448	883	-	-	-	-

Tabla de carga computacional. Número de operaciones necesarias para los algoritmos SB-NDFT y SB-DFT para detectar las energías presentes en las ocho frecuencias DTMF usando el mínimo número de muestras. (M y S denotan el número de multiplicaciones reales y el número de sumas reales respectivamente).

Tal y como se muestra en la Tabla Caso 2, no podemos conseguir un figura de mérito elevada usando el algoritmo SB-DFT si queremos cumplir, simultáneamente, con los requerimientos de tolerancia en frecuencia. En la Tabla de carga computacional, se ha calculado el número de operaciones necesarias para detectar las energías presentes en las ocho frecuencias DTMF, usando el mínimo número de muestras (dadas por las Tablas Caso 1 y Caso 2). Claramente, el algoritmo SB-NDFT requiere menos operaciones para obtener una cierta figura de mérito. Por tanto, queda demostrada la enorme ventaja que supone usar el algoritmo de detección de DTMF basado en la SB-NDFT frente a aquél basado en la SB-DFT.

2.5 RESUMEN

En este capítulo se ha propuesto un nuevo algoritmo para decodificar las señales DTMF basado en la NDFT Subbanda. Este algoritmo utiliza el hecho de que las frecuencias DTMF residen en la parte de baja frecuencia del ancho de banda telefónico. Así, se obtiene una buena aproximación de las energías presentes en las frecuencias DTMF descartando la banda de altas frecuencias una vez realizada una descomposición en dos bandas. Hemos comparado, mediante varias simulaciones que creaban el entorno descrito por el estándar AT&T TouchTone, el rendimiento de este algoritmo con aquél basado en la SB-DFT. Como resultado de dichas simulaciones, hemos encontrado que el algoritmo SB-NDFT requiere muchas menos operaciones que el SB-DFT (y, por supuesto, que los basados en la NDFT o en la DFT, a los que se hizo referencia en el punto 2.2.3) para conseguir un determinado nivel de calidad. Este resultado es importante porque, al ser implementado en un DSP, el algoritmo propuesto puede llevar a un aumento en el número de canales DTMF que pueden ser decodificados simultáneamente en un entorno multicanal.

2.6 APÉNDICE: PROGRAMAS REALIZADOS EN MATLAB PARA LA ELABORACIÓN DE ESTE CAPÍTULO

✓ Fichero espectrotecla1.m

```
% APLICACIÓN DE LA NDFT A LA Dual-Tone Multi-Frequency (DTMF)
% CONCRETAMENTE, APLICAREMOS EL MÉTODO SB-NDFT PARA DEMODULAR LAS SEÑALES DTMF

% Vamos a ver la magnitud de la transf.de Fourier de una señal generada DTMF cuando
% el dígito 1 es marcado. Variaremos el número de muestras usadas y también
%contemplaremos los casos de No Twist, Normal Twist y Reverse Twist.

% MATRIZ QUE CONTIENE LOS POSIBLES VALORES AL PULSAR EL TECLADO TELEFÓNICO
MATRIZ=['1' '2' '3' 'A'; '4' '5' '6' 'B'; '7' '8' '9' 'C'; '*' '0' '#' 'D'];

% VECTORES DE FRECUENCIAS DE LA DTMF (FRECUENCIAS DIGITALES)

fm=8000; % Frecuencia de muestreo
V_FILA=2*pi*[697 770 852 941]/fm;
V_COLUMNA=2*pi*[1209 1336 1477 1633]/fm;
wk=[V_FILA V_COLUMNA];

v_pulsado='1'; % Tecla pulsada
indice=1; % Índice que sirve para almacenar las 6 gráficas

for twist=1:3 % Contemplamos los tres casos de twist

for N=100:220:320 % Primero usaremos 100 muestras y luego 320

    if twist==1 % No Twist
        Ar=1;
        Ac=1;
    elseif twist==2 % Maximun Reverse Twist
        Ac=1;
    end
end
end
```

```

        Ar=Ac/(10^(4/20));
    else                                     % Maximun Normal Twist
        Ar=1;
        Ac=Ar/(10^(8/20));
    end

% Detectamos el elemento tecleado
for i=1:4
    for j=1:4
        if MATRIZ(i,j)==v_pulsado
            wr=V_FILA(i);
            wc=V_COLUMNA(j);
        end
    end
end

% Construimos la señal DTMF muestreada

xn=Ar*cos(wr*[0:N-1])+Ac*cos(wc*[0:N-1]);

l=1024;
grafica(indice,:)=abs(fftshift(fft(xn,l))); % Vamos almacenando las gráficas
indice=indice+1;

end
end

% Representamos las 6 gráficas obtenidas

w=[-pi:2*pi/l:pi-2*pi/l]; % Vector de frecuencias digitales
fc=w*fm/(2*pi);           % Vector de frecuencias (Hz)

fdtmf_cont=wk*fm/(2*pi); % Frecuencias DTMF en Hz

for indice=1:2:5
    subplot(3,2,indice),plot(fc,grafica(indice,:));
    xlabel('Frecuencia (Hz)');
    hold on
    subplot(3,2,indice),plot(fdtmf_cont,zeros(1,length(fdtmf_cont)),'rx');
    axis([500 1800 -2 55])
end

for indice=2:2:6
    subplot(3,2,indice),plot(fc,grafica(indice,:));
    xlabel('Frecuencia (Hz)');
    hold on
    subplot(3,2,indice),plot(fdtmf_cont,zeros(1,length(fdtmf_cont)),'rx');
    axis([500 1800 -2 165])
end
hold off

```

✓ Fichero toleranciafrecuencia.m

```

% APLICACIÓN DE LA NDFT A LA Dual-Tone Multi-Frequency (DTMF)

% Vamos a ver el efecto de la tolerancia en frecuencia en la detección de DTMF. Así,
% representaremos la magnitud de la transformada de Fourier de una señal DTMF con
% N=320 muestras y cuando el dígito 'D' es marcado. Contemplaremos el caso de Maximun
% Normal Twist.

clear;

% MATRIZ QUE CONTIENE LOS POSIBLES VALORES AL PULSAR EL TECLADO TELEFÓNICO
MATRIZ=['1' '2' '3' 'A'; '4' '5' '6' 'B'; '7' '8' '9' 'C'; '*' '0' '#' 'D'];

% VECTORES DE FRECUENCIAS DE LA DTMF (FRECUENCIAS DIGITALES)

fm=8000; % Frecuencia de muestreo

```

```
V_FILA=2*pi*[697 770 852 941]/fm;
V_COLUMNA=2*pi*[1209 1336 1477 1633]/fm;
wk=[V_FILA V_COLUMNA];

v_pulsado='D'; % Tecla pulsada
N=320; % Usaremos 320 muestras

Ar=1; % Maximun Normal Twist
Ac=Ar/(10^(8/20));

% Detectamos el elemento tecleado
for i=1:4
    for j=1:4
        if MATRIZ(i,j)==v_pulsado
            wr=V_FILA(i);
            wc=V_COLUMNA(j);
        end
    end
end

% Construimos la señal DTMF muestreada
%(El tono de la columna se desplaza hasta el límite permitido).

xn=Ar*cos(wr*[0:N-1])+Ac*cos(wc*(1-0.015)*[0:N-1]);

l=1024;
grafica=abs(fftshift(fft(xn,l)));

% Construimos un vector que contenga los 8 tonos DTMF así como las frecuencias en los
% bordes de la banda de tolerancia asociada a cada uno de los tonos.

fdtmf_cont=wk*fm/(2*pi); % Frecuencias DTMF en Hz

v_band_tole=[fdtmf_cont*(1-0.015) fdtmf_cont*(1+0.015)]; % Bandas de tolerancia

% Representamos la gráfica obtenida junto con las bandas de tolerancia

w=[-pi:2*pi/l:pi-2*pi/l]; % Vector de frecuencias digitales
fc=w*fm/(2*pi); % Vector de frecuencias (Hz)

hold on
plot(fc,grafica);
plot(fdtmf_cont,zeros(1,length(fdtmf_cont)),'ro');
plot(v_band_tole,zeros(1,length(v_band_tole)),'r+');
axis([500 1800 -2 165])
hold off
```

✓ Fichero efectotonosNOP.m

```
% APLICACIÓN DE LA NDFT A LA Dual-Tone Multi-Frequency (DTMF)

% En este caso, vamos a ver el efecto de la presencia de tonos 'NO OPERACIONALES' en el
% proceso de demodulación de la DTMF. Todo se hará suponiendo que es la tecla '1' la
% pulsada. Consideramos No Twist y variamos el número de muestras N para ver su efecto.

clear;

% MATRIZ QUE CONTIENE LOS POSIBLES VALORES AL PULSAR EL TECLADO TELEFÓNICO
MATRIZ=['1' '2' '3' 'A'; '4' '5' '6' 'B'; '7' '8' '9' 'C'; '*' '0' '#' 'D'];

fm=8000; % Frecuencia de muestreo

% VECTORES DE FRECUENCIAS DE LA DTMF (FRECUENCIAS DIGITALES)
V_FILA=2*pi*[697 770 852 941]/fm;
V_COLUMNA=2*pi*[1209 1336 1477 1633]/fm;
wk=[V_FILA V_COLUMNA];
```

```

aux1=1;           % Índ. auxiliar para la matriz que contendrá las figuras de mérito en
                  % cada iteración.

for iteracion=1:2

    aux2=1;       % Índ. auxiliar para la matriz que contendrá las figuras de mérito en
                  % cada iteración.

    v_pulsado='1'; % Tecla pulsada.

    for N=60:2:150 % Número de muestras de la señal DTMF generada

        Ar=1;     % Suponemos que no existe Twist
        Ac=1;

        % Detectamos el elemento tecleado
        for i=1:4
            for j=1:4
                if MATRIZ(i,j)==v_pulsado
                    wr=V_FILA(i);
                    wc=V_COLUMNNA(j);
                end
            end
        end

        % Construimos la señal DTMF muestreada y esa misma señal retrasada.
        % En la primera iteración NO añadiremos los restantes 6 tonos no generados por el
        % tecleo del 1 y en la segunda Sí los añadiremos (con 30 dB de menor amplitud) y
        % compararemos las figuras de mérito en ambos casos.

        xn=Ar*cos(wr*[0:N-1])+Ac*cos(wc*[0:N-1]);
        xn_1=Ar*cos(wr*[-1:N-2])+Ac*cos(wc*[-1:N-2]);

        if iteracion==2 % En la segunda iteración metemos los tonos no generados por el 1
            for i=1:8
                if ((wk(i)~=wr) & (wk(i)~=wc))
                    xn=xn+Ar/(10^(1.5))*cos(wk(i)*[0:N-1]);
                    xn_1=xn_1+Ar/(10^(1.5))*cos(wk(i)*[-1:N-2]);
                end
            end
        end

        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %%%%%%%%%% Construimos: DTMF decoding algorithm using the SB-NDFT %%%%%%%%%%
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

        for i=1:N/2
            x2n(i)=xn(2*i);
            x2n_1(i)=xn_1(2*i);
        end

        gl=1/2*(x2n+x2n_1);
        gh=1/2*(x2n-x2n_1);

        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %%%%%%%%%% CONSTRUIMOS EL ALGORITMO GOERTZEL %%%%%%%%%%
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        j=sqrt(-1);

        qk([1:8],1)=0; % CONDICIONES
        qk([1:8],2)=0; % INICIALES

        for i=1:8
            for k=3:length(gl)
                qk(i,k)=gl(k)+2*cos(2*wk(i))*qk(i,k-1)-qk(i,k-2);
                yk(i,k)=qk(i,k)-exp(-j*2*wk(i))*qk(i,k-1);
            end
        end

        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %%%%%%%%%% FIN DEL ALGORITMO GOERTZEL %%%%%%%%%%
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        for i=1:8
            GLW2(i)=(abs(yk(i,(N/2))))^2;
        end
    end
end

```

```

for i=1:8
    XW2(i)=GLW2(i)/(1+cos(wk(i)));
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Fin del DTMF decoding algorithm using the SB-NDFT %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% FIGURA DE MÉRITO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Xr=max(XW2(1:4)); % Máximo de los tonos de las filas
Xc=max(XW2(5:8)); % Máximo de los tonos de las columnas

ind_rmax=find(XW2==max(XW2(1:4))); % Índice de Xr
ind_cmax=find(XW2==max(XW2(5:8))); % Índice de Xc

% Buscamos ahora el segundo mayor valor de las filas y columnas.

XW2(ind_rmax)=0;
XW2(ind_cmax)=0;
Xrs=max(XW2(1:4)); % Segundo mayor valor de los tonos de las filas
Xcs=max(XW2(5:8)); % Segundo mayor valor de los tonos de las columnas

% Definimos la figura de mérito para medir la calidad de la decodificación

Dr=10*log10(Xr)-10*log10(Xrs);
Dc=10*log10(Xc)-10*log10(Xcs);

F_M(aux1,aux2)=min(Dr,Dc); % Matriz que contiene todas las figuras de mérito
calculadas
aux2=aux2+1;

end % Fin bucle N

aux1=aux1+1;

end % Fin bucle iteración

% Representamos las figuras de mérito obtenidas para ambos casos.
% En color rojo para el caso en el que sólo están las frecuencias operacionales.
% En color azul para el caso en el que están presentes las seis frecuencias restantes
DTMF.

hold on
plot([60:2:150],F_M(1,:), 'b', [60:2:150],F_M(2,:), 'r')
xlabel('Número de muestras');ylabel('Figura de mérito (dB)')
title('Línea azul:Sólo las frec. operacionales. Línea roja:Todas las
frecuencias.')
```

✓ Fichero Fsbndft_ffija.m

```

% GRÁFICA QUE MUESTRA LA VARIACIÓN DE LA FIGURA DE MÉRITO PARA EL CASO %% SB-NDFT %%
% SE VA A CALCULAR LA FIGURA DE MÉRITO PARA CADA TECLA PULSADA Y A SU VEZ, PARA LOS TRES
% CASOS DE TWIST POSIBLE, QUEDÁNDONOS CON EL PEOR F OBTENIDO PARA CADA CASO. TODO ELLO,
% VARIANDO N.

clear;

% MATRIZ QUE CONTIENE LOS POSIBLES VALORES AL PULSAR EL TECLADO TELEFÓNICO
MATRIZ=['1' '2' '3' 'A'; '4' '5' '6' 'B'; '7' '8' '9' 'C'; '*' '0' '#' 'D'];

fm=8000; % Frecuencia de muestreo

% VECTORES DE FRECUENCIAS DE LA DTMF (FRECUENCIAS DIGITALES)
V_FILA=2*pi*[697 770 852 941]/fm;
V_COLUMNA=2*pi*[1209 1336 1477 1633]/fm;
```

```

wk=[V_FILA V_COLUMNA];

for TECLA=1:16 % PARA CADA TECLA
    if mod(TECLA,4)==0
        col=4;
    else
        col=mod(TECLA,4);
    end

    v_pulsado=MATRIZ(ceil(TECLA/4),col);
    aux=0; % Índice auxiliar para las columnas de F

for N=60:2:130 % PARA CADA N

for twist=1:3 % PARA LOS TRES CASOS DE TWIST
    if twist==1 % No Twist
        Ar=1;
        Ac=1;
    elseif twist==2 % Maximun Reverse Twist
        Ar=1;
        Ac=Ac/(10^(4/20));
    else % Maximun Normal Twist
        Ar=1;
        Ac=Ar/(10^(8/20));
    end

% Detectamos el elemento tecleado.

for i=1:4
    for j=1:4
        if MATRIZ(i,j)==v_pulsado
            wr=V_FILA(i);
            wc=V_COLUMNA(j);
        end
    end
end

% Construimos la señal DTMF muestreada y esa misma señal retrasada
xn=Ar*cos(wr*[0:N-1])+Ac*cos(wc*[0:N-1]);
xn_1=Ar*cos(wr*[-1:N-2])+Ac*cos(wc*[-1:N-2]);

% Introducimos el resto de tonos a una amplitud 30 dB inferior
for i=1:8
    if ((wk(i)~=wr) & (wk(i)~=wc))
        xn=xn+Ar/(10^(1.5))*cos(wk(i)*[0:N-1]);
        xn_1=xn_1+Ar/(10^(1.5))*cos(wk(i)*[-1:N-2]);
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Construímos: DTMF decoding algorithm using the SB-NDFT %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for i=1:N/2
    x2n(i)=xn(2*i);
    x2n_1(i)=xn_1(2*i);
end

gl=1/2*(x2n+x2n_1);
gh=1/2*(x2n-x2n_1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% CONSTRUIMOS EL ALGORITMO GOERTZEL %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
j=sqrt(-1);

qk([1:8],1)=0; % CONDICIONES
qk([1:8],2)=0; % INICIALES

for i=1:8
    for k=3:length(gl)
        qk(i,k)=gl(k)+2*cos(2*wk(i))*qk(i,k-1)-qk(i,k-2);
    end
end

```

```

        yk(i,k)=qk(i,k)-exp(-j*2*wk(i))*qk(i,k-1);
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% FIN DEL ALGORITMO GOERTZEL %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i=1:8
    GLW2(i)=(abs(yk(i,(N/2))))^2;
end

for i=1:8
    XW2(i)=GLW2(i)/(1+cos(wk(i)));
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Fin del DTMF decoding algorithm using the SB-NDFT %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% FIGURA DE MÉRITO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Xr=max(XW2(1:4)); % Máximo de los tonos de las filas
Xc=max(XW2(5:8)); % Máximo de los tonos de las columnas

ind_rmax=find(XW2==max(XW2(1:4))); % Índice de Xr
ind_cmax=find(XW2==max(XW2(5:8))); % Índice de Xc

% Buscamos ahora el segundo mayor valor de las filas y columnas.

XW2(ind_rmax)=0;
XW2(ind_cmax)=0;
Xrs=max(XW2(1:4)); % Segundo mayor valor de los tonos de las filas
Xcs=max(XW2(5:8)); % Segundo mayor valor de los tonos de las columnas

% Definimos la figura de mérito para medir la calidad de la decodificación

Dr=10*log10(Xr)-10*log10(Xrs);
Dc=10*log10(Xc)-10*log10(Xcs);

FfijoN(twist)=min(Dr,Dc);

end % FIN BUCLE TWIST

aux=aux+1;
F(TECLA,aux)=min(FfijoN);

end % FIN BUCLE N

end % FIN BUCLE TECLA

% Nos quedamos con el peor caso de todos

for j=1:length(F) % Recorremos las columnas y suponemos que N>16 siempre (algo
normal)
    Fpeor(j)=min(F(:,j));
end

% Representamos

plot([60:2:130],Fpeor)
xlabel('Número de muestras. ALGORITMO SB-NDFT .');ylabel('Figura de Mérito (dB)');
title('VARIACIÓN DE F CON RESPECTO A N PARA FRECUENCIAS OPERACIONALES FIJAS')

```

✓ Fichero Fsbdfn ffija.m

% GRÁFICA QUE MUESTRA LA VARIACIÓN DE LA FIGURA DE MÉRITO PARA EL CASO %% SB-DFT %%

Proyecto Fin de Carrera. Ingeniería de Telecomunicación. Juan Antonio González Sánchez

```
% SE VA A CALCULAR LA FIGURA DE MÉRITO PARA CADA TECLA PULSADA Y A SU VEZ, PARA LOS TRES
% CASOS DE TWIST POSIBLE, QUEDÁNDONOS CON EL PEOR F OBTENIDO PARA CADA CASO. TODO ELLO,
% VARIANDO N.

clear;

% MATRIZ QUE CONTIENE LOS POSIBLES VALORES AL PULSAR EL TECLADO TELEFÓNICO
MATRIZ=['1' '2' '3' 'A'; '4' '5' '6' 'B'; '7' '8' '9' 'C'; '*' '0' '#' 'D'];

fm=8000; % Frecuencia de muestreo

% VECTORES DE FRECUENCIAS DE LA DTMF (FRECUENCIAS DIGITALES)
V_FILA=2*pi*[697 770 852 941]/fm;
V_COLUMNNA=2*pi*[1209 1336 1477 1633]/fm;
wk=[V_FILA V_COLUMNNA];

for TECLA=1:16 % PARA CADA TECLA
    if mod(TECLA,4)==0
        col=4;
    else
        col=mod(TECLA,4);
    end

    v_pulsado=MATRIZ(ceil(TECLA/4),col);
    aux=0; % Índice auxiliar para las columnas de F

    for N=60:2:130 % PARA CADA N

        for twist=1:3 % PARA LOS TRES CASOS DE TWIST
            if twist==1 % No Twist
                Ar=1;
                Ac=1;
            elseif twist==2 % Maximun Reverse Twist
                Ac=1;
                Ar=Ac/(10^(4/20));
            else % Maximun Normal Twist
                Ar=1;
                Ac=Ar/(10^(8/20));
            end

            % Detectamos el elemento tecleado.

            for i=1:4
                for j=1:4
                    if MATRIZ(i,j)==v_pulsado
                        wr=V_FILA(i);
                        wc=V_COLUMNNA(j);
                    end
                end
            end

            % Construimos la señal DTMF muestreada y esa misma señal retrasada

            xn=Ar*cos(wr*[0:N-1])+Ac*cos(wc*[0:N-1]);
            xn_1=Ar*cos(wr*[-1:N-2])+Ac*cos(wc*[-1:N-2]);

            % Introducimos el resto de tonos a una amplitud 30 dB inferior

            for i=1:8
                if (wk(i)~=wr) & (wk(i)~=wc)
                    xn=xn+Ar/(10^(1.5))*cos(wk(i)*[0:N-1]);
                    xn_1=xn_1+Ar/(10^(1.5))*cos(wk(i)*[-1:N-2]);
                end
            end

            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            % Construímos:DTMF decoding algorithm using the SB-DFT %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

            for i=1:N/2
                x2n(i)=xn(2*i);
                x2n_1(i)=xn_1(2*i);
            end
        end
    end
end
```

```

gl=1/2*(x2n+x2n_1);
gh=1/2*(x2n-x2n_1);

% Calculamos las frecuencias más próximas a las de la DTMF.
% Ésta es la diferencia con respecto al método desarrollado para la SB_NDFT.

wk_sbdft=2*pi/N*round(wk/(2*pi)*N);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% CONSTRUIMOS EL ALGORITMO GOERTZEL %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
j=sqrt(-1);

qk([1:8],1)=0;           % CONDICIONES
qk([1:8],2)=0;           % INICIALES

for i=1:8
    for k=3:length(gl)
        qk(i,k)=gl(k)+2*cos(2*wk_sbdft(i))*qk(i,k-1)-qk(i,k-2);
        yk(i,k)=qk(i,k)-exp(-j*2*wk_sbdft(i))*qk(i,k-1);
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% FIN DEL ALGORITMO GOERTZEL %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i=1:8
    GLW2(i)=(abs(yk(i,(N/2))))^2;
end

for i=1:8
    XW2(i)=GLW2(i)/(1+cos(wk_sbdft(i)));
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Fin del DTMF decoding algorithm using the SB-DFT %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% FIGURA DE MÉRITO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Xr=max(XW2(1:4));           % Máximo de los tonos de las filas
Xc=max(XW2(5:8));           % Máximo de los tonos de las columnas

ind_rmax=find(XW2==max(XW2(1:4))); % Índice de Xr
ind_cmax=find(XW2==max(XW2(5:8))); % Índice de Xc

% Buscamos ahora el segundo mayor valor de las filas y columnas.

XW2(ind_rmax)=0;
XW2(ind_cmax)=0;
Xrs=max(XW2(1:4));           % Segundo mayor valor de los tonos de las filas
Xcs=max(XW2(5:8));           % Segundo mayor valor de los tonos de las columnas

% Definimos la figura de mérito para medir la calidad de la decodificación

Dr=10*log10(Xr)-10*log10(Xrs);
Dc=10*log10(Xc)-10*log10(Xcs);

FfijoN(twist)=min(Dr,Dc);

end % FIN BUCLE TWIST

aux=aux+1;
F(TECLA,aux)=min(FfijoN);

end % FIN BUCLE N

end % FIN BUCLE TECLA

% Nos quedamos con el peor caso de todos

```

```

for j=1:length(F)      % Recorremos las columnas y suponemos que N>16 siempre (algo
normal)
    Fpeor(j)=min(F(:,j));
end

% Representamos

plot([60:2:130],Fpeor)
xlabel('Número de muestras. ALGORITMO SB-DFT .');ylabel('Figura de Mérito (dB)');
title('VARIACIÓN DE F CON RESPECTO A N PARA FRECUENCIAS OPERACIONALES FIJAS')

```

✓ Fichero Fsbndft_fvariable.m

```

% GRÁFICA QUE MUESTRA LA VARIACIÓN DE LA FIGURA DE MÉRITO PARA EL CASO %% SB-NDFT %%
% SE VA A CALCULAR LA FIGURA DE MÉRITO PARA CADA TECLA PULSADA Y A SU VEZ, PARA LOS TRES
% CASOS DE TWIST POSIBLE, QUEDÁNDONOS CON EL PEOR F OBTENIDO PARA CADA CASO. TODO ELLO,
% VARIANDO N. TAMBIÉN VARIAREMOS LAS FRECUENCIAS OPERACIONALES DENTRO DE LAS RESPECTIVAS
% BANDAS DE TOLERANCIA.

clear;

% MATRIZ QUE CONTIENE LOS POSIBLES VALORES AL PULSAR EL TECLADO TELEFÓNICO
MATRIZ=['1' '2' '3' 'A';'4' '5' '6' 'B';'7' '8' '9' 'C';'*' '0' '#' 'D'];

fm=8000;          % Frecuencia de muestreo

% VECTORES DE FRECUENCIAS DE LA DTMF (FRECUENCIAS DIGITALES)
V_FILA=2*pi*[697 770 852 941]/fm;
V_COLUMNA=2*pi*[1209 1336 1477 1633]/fm;
wk=[V_FILA V_COLUMNA];

for TECLA=1:16          % PARA CADA TECLA
    if mod(TECLA,4)==0
        col=4;
    else
        col=mod(TECLA,4);
    end

    v_pulsado=MATRIZ(ceil(TECLA/4),col)
    aux=0;              % Índice auxiliar para las columnas de F

    for N=60:2:150      % PARA CADA N

        for twist=1:3    % PARA LOS TRES CASOS DE TWIST
            if twist==1   % No Twist
                Ar=1;
                Ac=1;
            elseif twist==2 % Maximun Reverse Twist
                Ar=1;
                Ac=Ac/(10^(4/20));
            else           % Maximun Normal Twist
                Ar=1;
                Ac=Ar/(10^(8/20));
            end
        end

        % Detectamos el elemento tecleado.

        for i=1:4
            for j=1:4
                if MATRIZ(i,j)==v_pulsado
                    wr=V_FILA(i);
                    wc=V_COLUMNA(j);
                end
            end
        end
    end
end

```

```
% CONSTRUIMOS LA SEÑAL DTMF MUESTREADA (INCLUYENDO LAS DESVIACIONES EN FRECUENCIA
% ESPECIFICADAS EN EL STANDAR AT&T TOUCH TONE). Concretamente, tomamos frecuencias
% equidistantes respecto del valor nominal con variaciones de 5 Hz hasta llegar a una
% desviación total del 1.5% a ambos lados de dicha frec. nominal.

interv_tolerancia_wr=[(1-0.015)*wr,(1+0.015)*wr];
interv_tolerancia_wc=[(1-0.015)*wc,(1+0.015)*wc];

desv_frec_digital=5*2*pi/fm;

% Calculamos el número de frecuencias distorsionantes que caben en la banda de
% tolerancia

MIT_FREC_wr=floor((interv_tolerancia_wr(2)-
interv_tolerancia_wr(1))/(2*desv_frec_digital));
MIT_FREC_wc=floor((interv_tolerancia_wc(2)-
interv_tolerancia_wc(1))/(2*desv_frec_digital));

% Generamos todas las frecuencias distorsionantes

for desviacion=-MIT_FREC_wr:MIT_FREC_wr
    wdespwr(MIT_FREC_wr+desviacion+1)=5*2*pi/fm*(desviacion);
end

for desviacion=-MIT_FREC_wc:MIT_FREC_wc
    wdespwc(MIT_FREC_wc+desviacion+1)=5*2*pi/fm*(desviacion);
end

% Generamos ya las señales DTMF

for fila=1:length(wdespwr)
    for columna=1:length(wdespwc)
        xn=Ar*cos((wr+wdespwr(fila))*[0:N-1])+Ac*cos((wc+wdespwc(columna))*[0:N-1]);
        xn_1=Ar*cos((wr+wdespwr(fila))*[-1:N-2])+Ac*cos((wc+wdespwc(columna))*[-1:N-2]);

% Introducimos el resto de tonos a una amplitud 30 dB inferior

for i=1:8
    if ((wk(i)~=wr) & (wk(i)~=wc))
        xn=xn+Ar/(10^(1.5))*cos(wk(i))*[0:N-1];
        xn_1=xn_1+Ar/(10^(1.5))*cos(wk(i))*[-1:N-2];
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Construímos: DTMF decoding algorithm using the SB-NDFT %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for i=1:N/2
    x2n(i)=xn(2*i);
    x2n_1(i)=xn_1(2*i);
end

g1=1/2*(x2n+x2n_1);
gh=1/2*(x2n-x2n_1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% CONSTRUIMOS EL ALGORITMO GOERTZEL %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
j=sqrt(-1);

qk([1:8],1)=0;           % CONDICIONES
qk([1:8],2)=0;           % INICIALES

for i=1:8
    for k=3:length(g1)
        qk(i,k)=g1(k)+2*cos(2*wk(i))*qk(i,k-1)-qk(i,k-2);
        yk(i,k)=qk(i,k)-exp(-j*2*wk(i))*qk(i,k-1);
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% FIN DEL ALGORITMO GOERTZEL %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for i=1:8
    GLW2(i)=(abs(yk(i,(N/2))))^2;
```

```

end

for i=1:8
    XW2(i)=GLW2(i)/(1+cos(wk(i)));
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Fin del DTMF decoding algorithm using the SB-NDFT %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% FIGURA DE MÉRITO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Xr=max(XW2(1:4)); % Máximo de los tonos de las filas
Xc=max(XW2(5:8)); % Máximo de los tonos de las columnas

ind_rmax=find(XW2==max(XW2(1:4))); % Índice de Xr
ind_cmax=find(XW2==max(XW2(5:8))); % Índice de Xc

% Buscamos ahora el segundo mayor valor de las filas y columnas.

XW2(ind_rmax)=0;
XW2(ind_cmax)=0;
Xrs=max(XW2(1:4)); % Segundo mayor valor de los tonos de las filas
Xcs=max(XW2(5:8)); % Segundo mayor valor de los tonos de las columnas

% Definimos la figura de mérito para medir la calidad de la decodificación

Dr=10*log10(Xr)-10*log10(Xrs);
Dc=10*log10(Xc)-10*log10(Xcs);

Ffreq(fila,columna)=min(Dr,Dc);

end % FIN BUCLE COLUMNA
end % FIN BUCLE FILA

Ffijon(twist)=min(Ffreq(:));

end % FIN BUCLE TWIST

aux=aux+1;
F(TECLA,aux)=min(Ffijon);

end % FIN BUCLE N
end % FIN BUCLE TECLA

% Nos quedamos con el peor caso de todos

for j=1:length(F) % Recorremos las columnas y suponemos que N>16 siempre (algo normal)
    Fpeor(j)=min(F(:,j));
end

% Representamos

plot([60:2:150],Fpeor)
xlabel('Número de muestras. ALGORITMO SB-NDFT .');ylabel('Figura de Mérito (dB)');
title('VARIACIÓN DE F CON RESPECTO A N PARA FRECUENCIAS OPERACIONALES VARIABLES')

```

✓ Fichero Fsbdf_t_fvariable.m

```

% GRÁFICA QUE MUESTRA LA VARIACIÓN DE LA FIGURA DE MÉRITO PARA EL CASO %% SB-DFT %%
% SE VA A CALCULAR LA FIGURA DE MÉRITO PARA CADA TECLA PULSADA Y A SU VEZ, PARA LOS TRES
% CASOS DE TWIST POSIBLE, QUEDÁNDONOS CON EL PEOR F OBTENIDO PARA CADA CASO. TODO ELLO,
% VARIANDO N. TAMBIÉN VARIAREMOS LAS FRECUENCIAS OPERACIONALES DENTRO DE LAS RESPECTIVAS
% BANDAS DE TOLERANCIA.

```

```

clear;

% MATRIZ QUE CONTIENE LOS POSIBLES VALORES AL PULSAR EL TECLADO TELEFÓNICO
MATRIZ=['1' '2' '3' 'A'; '4' '5' '6' 'B'; '7' '8' '9' 'C'; '*' '0' '#' 'D'];

fm=8000; % Frecuencia de muestreo

% VECTORES DE FRECUENCIAS DE LA DTMF (FRECUENCIAS DIGITALES)
V_FILA=2*pi*[697 770 852 941]/fm;
V_COLUMNA=2*pi*[1209 1336 1477 1633]/fm;
wk=[V_FILA V_COLUMNA];

for TECLA=1:16 % PARA CADA TECLA
    if mod(TECLA,4)==0
        col=4;
    else
        col=mod(TECLA,4);
    end

    v_pulsado=MATRIZ(ceil(TECLA/4),col)
    aux=0; % Índice auxiliar para las columnas de F

    for N=60:2:150 % PARA CADA N

        for twist=1:3 % PARA LOS TRES CASOS DE TWIST
            if twist==1 % No Twist
                Ar=1;
                Ac=1;
            elseif twist==2 % Maximun Reverse Twist
                Ar=1;
                Ac=Ac/(10^(4/20));
            else % Maximun Normal Twist
                Ar=1;
                Ac=Ar/(10^(8/20));
            end

            % Detectamos el elemento tecleado.

            for i=1:4
                for j=1:4
                    if MATRIZ(i,j)==v_pulsado
                        wr=V_FILA(i);
                        wc=V_COLUMNA(j);
                    end
                end
            end

            % CONSTRUIMOS LA SEÑAL DTMF MUESTREADA (INCLUYENDO LAS DESVIACIONES EN FRECUENCIA
            % ESPECIFICADAS EN EL STANDAR AT&T TOUCH TONE). Concretamente, tomamos frecuencias
            % equidistantes respecto del valor nominal con variaciones de 5 Hz hasta llegar a una
            % desviación total del 1.5% a ambos lados de dicha frec. nominal.

            interv_tolerancia_wr=[(1-0.015)*wr, (1+0.015)*wr];
            interv_tolerancia_wc=[(1-0.015)*wc, (1+0.015)*wc];

            desv_frec_digital=5*2*pi/fm;

            % Calculamos el número de frecuencias distorsionantes que caben en la banda de
            % tolerancia

            MIT_FREC_wr=floor((interv_tolerancia_wr(2)-
            interv_tolerancia_wr(1))/(2*desv_frec_digital));
            MIT_FREC_wc=floor((interv_tolerancia_wc(2)-
            interv_tolerancia_wc(1))/(2*desv_frec_digital));

            % Generamos todas las frecuencias distorsionantes

            for desviacion=-MIT_FREC_wr:MIT_FREC_wr
                wdespwr(MIT_FREC_wr+desviacion+1)=5*2*pi/fm*(desviacion);
            end

            for desviacion=-MIT_FREC_wc:MIT_FREC_wc
                wdespwc(MIT_FREC_wc+desviacion+1)=5*2*pi/fm*(desviacion);
            end
        end
    end
end

```

```

end

% Generamos ya las señales DTMF

for fila=1:length(wdespwr)
    for columna=1:length(wdespwc)
        xn=Ar*cos((wr+wdespwr(fila))*[0:N-1])+Ac*cos((wc+wdespwc(columna))*[0:N-1]);
        xn_1=Ar*cos((wr+wdespwr(fila))*[-1:N-2])+Ac*cos((wc+wdespwc(columna))*[-1:N-2]);

% Introducimos el resto de tonos a una amplitud 30 dB inferior

for i=1:8
    if ((wk(i)~=wr) & (wk(i)~=wc))
        xn=xn+Ar/(10^(1.5))*cos(wk(i))*[0:N-1];
        xn_1=xn_1+Ar/(10^(1.5))*cos(wk(i))*[-1:N-2];
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Construímos:DTMF decoding algorithm using the SB-DFT %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for i=1:N/2
    x2n(i)=xn(2*i);
    x2n_1(i)=xn_1(2*i);
end

gl=1/2*(x2n+x2n_1);
gh=1/2*(x2n-x2n_1);

% Calculamos las frecuencias más próximas a las de la DTMF.
% Ésta es la diferencia con respecto al método desarrollado para la SB_NDFT.

wk_sbdf=2*pi/N*round(wk/(2*pi)*N);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% CONSTRUIMOS EL ALGORITMO GOERTZEL %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
j=sqrt(-1);

qk([1:8],1)=0;           % CONDICIONES
qk([1:8],2)=0;           % INICIALES

for i=1:8
    for k=3:length(gl)
        qk(i,k)=gl(k)+2*cos(2*wk_sbdf(i))*qk(i,k-1)-qk(i,k-2);
        yk(i,k)=qk(i,k)-exp(-j*2*wk_sbdf(i))*qk(i,k-1);
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% FIN DEL ALGORITMO GOERTZEL %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i=1:8
    GLW2(i)=(abs(yk(i,(N/2))))^2;
end

for i=1:8
    XW2(i)=GLW2(i)/(1+cos(wk_sbdf(i)));
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Fin del DTMF decoding algorithm using the SB-DFT %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% FIGURA DE MÉRITO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Xr=max(XW2(1:4));           % Máximo de los tonos de las filas
Xc=max(XW2(5:8));           % Máximo de los tonos de las columnas

ind_rmax=find(XW2==max(XW2(1:4))); % Índice de Xr

```

```

ind_cmax=find(XW2==max(XW2(5:8))); % Índice de Xc

% Buscamos ahora el segundo mayor valor de las filas y columnas.

XW2(ind_rmax)=0;
XW2(ind_cmax)=0;
Xrs=max(XW2(1:4)); % Segundo mayor valor de los tonos de las filas
Xcs=max(XW2(5:8)); % Segundo mayor valor de los tonos de las columnas

% Definimos la figura de mérito para medir la calidad de la decodificación

Dr=10*log10(Xr)-10*log10(Xrs);
Dc=10*log10(Xc)-10*log10(Xcs);

Ffreq(fila,columna)=min(Dr,Dc);

end % FIN BUCLE COLUMNA
end % FIN BUCLE FILA

FfijoN(twist)=min(Ffreq(:));

end % FIN BUCLE TWIST

aux=aux+1;
F(TECLA,aux)=min(FfijoN);

end % FIN BUCLE N

end % FIN BUCLE TECLA

% Nos quedamos con el peor caso de todos

for j=1:length(F) % Recorremos las columnas y suponemos que N>16 siempre (algo
normal)
    Fpeor(j)=min(F(:,j));
end

% Representamos

plot([60:2:150],Fpeor)
xlabel('Número de muestras. ALGORITMO SB-DFT .');ylabel('Figura de Mérito (dB)');
title('VARIACIÓN DE F CON RESPECTO A N PARA FRECUENCIAS OPERACIONALES VARIABLES')

```

3 CONCLUSIONES

Este Proyecto Fin de Carrera ha estudiado el concepto de la Transformada de Fourier Discreta No Uniforme, la cual, proporciona un acercamiento al muestreo no uniforme en el dominio frecuencial. La NDFT de una secuencia de longitud finita corresponde a muestrear su transformada Z en puntos arbitrariamente elegidos del plano Z . Por tanto, la NDFT se reduce a la clásica DFT cuando estos puntos de muestreo se sitúan en el círculo unidad en ángulos igualmente espaciados. La flexibilidad que proporciona la NDFT conduce a una resolución espectral variable que puede ser usada por el usuario. Esto es importante en muchas aplicaciones que necesiten del procesamiento de la señal, puesto que la mayoría de señales y sistemas tiende a tener su energía concentrada en cierta región del espectro y no uniformemente distribuida.

Después de definir la NDFT, demostramos que la NDFT inversa existe siempre que los puntos de muestreo elegidos sean distintos. El problema de calcular la inversa de la NDFT es matemáticamente equivalente al problema de la interpolación polinómica. El algoritmo de Goertzel se puede usar para calcular la NDFT. Sin embargo, algoritmos tan eficientes como la FFT no se pueden aplicar debido a que no hay una periodicidad implícita en la estructura de muestreo generalizada en que se basa la NDFT. También hemos presentado la NDFT Subbanda, que es un método para calcular la NDFT basado en una descomposición en subbandas de la secuencia de entrada. Este método lleva a un cálculo rápido y bastante aproximado de la NDFT para señales que tienen su energía concentrada en ciertas bandas concretas del espectro. El concepto de la NDFT fue extendido al caso de dos dimensiones.

En el proyecto complementario a éste, realizado por el alumno JUAN ANTONIO MORÓN GARCÍA, se ha estudiado el diseño de filtros digitales unidimensionales mediante la aplicación de la NDFT. Partiendo de los resultados obtenidos en tal proyecto, aquí se ha usado el concepto de la NDFT 2-D junto con tales resultados para desarrollar una técnica de muestreo no uniforme para diseñar filtros digitales bidimensionales. Una sección cruzada apropiada de la respuesta en frecuencia bidimensional deseada es aproximada mediante funciones analíticas unidimensionales basados en los polinomios de Chebyshev (con detalle en el proyecto complementario a éste). Las muestras se sitúan en contornos que tiene la misma forma que la banda de paso deseada del filtro bidimensional y que pasan a través de los extremos de la sección cruzada. *Experimentalmente* se demuestra que los mejores resultados se obtienen cuando los contornos en los que se sitúan las muestras tiene la misma forma que la banda de paso deseada del filtro 2-D. El método propuesto produce filtros no separables bidimensionales con buenas características en la banda de paso y poco rizado. Estos resultados adquieren más importancia si atendemos a la falta de un algoritmo práctico y eficiente para diseñar filtros óptimos bidimensionales.

Por último, se ha estudiado también un algoritmo bastante eficiente para la decodificación de señales multifrecuencia de doble tono o DTMF (Dual-Tone Multi-Frequency). Dicho algoritmo está basado en la SB-NDFT. Tal y como se especifica en el estándar de la CCITT, las frecuencias DTMF ocupan la parte de baja frecuencia del ancho de banda telefónico. Utilizando esta información, se ha desarrollado un algoritmo que calcula, de manera aproximada, las energías presentes en las frecuencias DTMF. En este algoritmo, se descompone la señal de entrada en dos subbandas, y nos quedamos con aquella que corresponde a la banda de bajas frecuencias para calcular las energías mediante el uso del algoritmo de Goertzel modificado. Se han realizado simulaciones aplicando los estándares AT&T TouchTone, para evaluar el rendimiento del algoritmo propuesto en comparación con otros. Los resultados han demostrado que el algoritmo SB-NDFT consigue un ahorro computacional bastante importante para lograr un determinado nivel de calidad.

Por tanto, es evidente que la NDFT es una herramienta bastante importante en muchas aplicaciones del procesamiento de la señal. En todas las aplicaciones tanto aquí consideradas como en el proyecto complementario a éste, los algoritmos basados en la NDFT consiguen unos mejores resultados que aquellos basados en la DFT. Esto demuestra la efectividad del muestreo en frecuencia no uniforme, siempre que la libertad de situar las muestras sea usada adecuadamente.

BIBLIOGRAFÍA

- S. Bagchi, S.K. Mitra: "The nonuniform discrete Fourier transform and its applications in signal processing". KLUWER ACADEMIC PUBLISHERS, 1999.
- J.S. Lim: "Two dimensional signal and image processing". PRENTICE-HALL, 1990.
- A.V. Oppenheim, R.W. Schaffer: "Discrete-time signal processing". PRENTICE-HALL INTERNATIONAL, ENGLEWOOD CLIFFS, 1989.
- J.G. Proakis, D.G. Manolakis: "Digital signal processing : principles, algorithms, and applications". UPPER SADDLE RIVER, NJ PRENTICE-HALL PTR, 1996.
- Bagchi, S. and Mitra, S. "The nonuniform discrete Fourier transform and its applications in filter design: Part I – 1-D". IEEE Trans. Circuits Syst. II: Analog and Digital Signal Processing, 43:422-433, 1996a.
- Bagchi, S. and Mitra, S. "The nonuniform discrete Fourier transform and its applications in filter design: Part II – 2-D". IEEE Trans. Circuits Syst. II: Analog and Digital Signal Processing, 43:434-444, 1996b.
- Bagchi, S. and Mitra, S. "Efficient robust DTMF decoding using the subband NDFT". Signal Processing, 56:255-267, 1997.
- Mitra, S., Shentov, O., and Petraglia, M. "A method for fast approximate computation of discrete-time transforms". In Proc. IEEE Int. Conf. on Acoust., Speech, Signal Processing, volume 4, pages 2025-2028, Albuquerque, NM, 1990.

- Goertzel, G. "An algorithm for the evaluation of finite trigonometric series". American Math. Monthly, 65:34-35, 1958.
- Cooley, J. and Tukey, J. "An algorithm for the machine computation of complex Fourier". Mathematics of Computation, 19:297-301, 1965.
- McClellan, J., Parks, T., and Rabiner, L. "A computer program for designing optimum FIR linear phase digital filters". IEEE Trans. Audio Electroacoust., AU-21:506-526, 1973.