



UNIVERSIDAD
POLITECNICA
DE VALENCIA



Funciones Prolate para modulaciones multiportadoras

Autor: Milton Romero Quiroz

Director 1: Miguel Angel Rodriguez Hernandez

Director 2: Vicente Cazares

Fecha de comienzo: 6/04/2018

Lugar de trabajo: E.T.S.I. Telecomunicaciones

Objetivos

El objetivo principal de este trabajo es realizar una evaluación del uso de las funciones Prolate como modelado de pulso en modulaciones multiportadoras y realizar una comparativa con los métodos utilizados en la actualidad para determinar si a futuro pueden ser una opción viable en el uso de estas en modulaciones multiportadoras. Se pretende que este trabajo sirva como introducción a las secuencias Slepian, sus características principales, los métodos de generación de estas funciones y su uso en modulaciones multiportadoras. Además, se pretende realizar una comparación de este tipo de ventana con otras ventanas más utilizadas, como lo son la rectangular y la gaussiana, y determinar ventajas y desventajas de la utilización de estas nuevas funciones para el modulado de pulso.

Metodología

La metodología del presente trabajo será mediante pruebas de simulación realizadas en el entorno de Matlab, el cual permite la simulación de un sistema OFDM y mediante técnicas de modelado de pulso se evaluarán ambos escenarios. Se analizarán características importantes de la forma de onda de los pulsos y en base a los resultados obtenidos determinar qué tan factible es el uso de las funciones Prolate para modulaciones multiportadoras. Se realizará una introducción a las funciones Prolate, sus características principales y parámetros, se introducirá el método de generación de estas y sus diferentes características en función a los parámetros seleccionados. Luego se implementarán en el entorno de las modulaciones multiportadoras y se realizarán las pruebas propuestas anteriormente.

Desarrollo

Como punto de partida se realizará una introducción a las funciones Prolate y como generarlas en el entorno de Matlab, así como también se pondrán a prueba los distintos parámetros de estas y como afectan a estas funciones. Se realizará una comparativa con otros tipos de onda y podremos observar las diferencias y ventajas o desventajas que tienen sobre los pulsos rectangulares y gaussiano. Se realizará el diseño de un sistema OFDM con la implementación de un bloque de modelado de pulso donde se hará uso de las funciones Prolate y se analizará el desempeño del sistema mediante las gráficas de tasa de error de bit.

Resultados

Se pudo obtener como resultado una mejora en el rendimiento del sistema al introducir un filtro de modelado de pulso utilizando las secuencias Slepian. Se pudo determinar que la tasa de errores disminuyó al realizar la implementación del filtro de modelado de pulso y se pudo observar como afectaban distintos parámetros en la generación de las secuencias al rendimiento del sistema. Se realizó una comparativa con el sistema OFDM convencional y se determinó que el uso de estas funciones no conlleva una implementación tan complicada y son una variante para tener en cuenta en implementaciones futuras.

Líneas futuras

Como próximo paso se debe realizar un estudio más a fondo del fundamento teórico de este tipo de funciones, ya que su utilización puede llevar a mejoras en sistemas actuales. La documentación actual no es muy extensa y un poco escasa y con mayor desarrollo e investigación de estas se puede llegar a implementaciones interesantes.

Abstract

OFDM is a multi-carrier modulation scheme that has the characteristic of decomposing a frequency-selective channel into multiple narrow-band channels with seemingly flat fading. This system can be improved using various pulse modeling techniques. The objective of this work is to present a type of functions that can be used to achieve this objective. The Discrete Prolate Spheroidal Sequences (DPSSs) have the characteristic of being an optimal window, a sequence limited in time whose energy is more concentrated in a finite frequency interval. The properties of these sequences are analyzed, as well as a method to generate them and discuss the advantages of using pulse modeling in comparison with the conventional system

ÍNDICE

I.	Introducción	1
II.	Prolate Spheroidal Wave Functions	2
III.	Series Slepian	3
IV.	Escenario General	4
IV.1	Generación de señales DPSS.....	5
IV.2	Espectro de señales DPSS.....	6
IV.3	Características de las secuencias Slepian.....	11
IV.4	Comparación con otras formas de onda.....	16
V.	Modulaciones Multiportadoras	19
V.1	Comparativa pulso rectangular y pulso Slepian.....	21
VI.	Simulación	23
V.1	Modelado de pulso.....	28
VII.	CONCLUSIONES	33
	BIBLIOGRAFÍAS	34

I. Introducción

Los sistemas multiportadoras como Orthogonal Frequency Division Multiplexing (OFDM) funcionan gracias a que el ancho de banda disponible se divide en múltiples canales de menor ancho de banda que se solapan entre sí y que a su vez son ortogonales entre sí, la separación es clave a la hora de garantizar la ortogonalidad. Este tipo de esquema utiliza formas de pulso rectangulares lo que causa que la señal OFDM tenga una alta radiación fuera de banda debido a los grandes lóbulos laterales de las subportadoras sinusoidales, esto es una desventaja cuando el número de canales es pequeño, por ello se necesitan cientos de canales para tener un sistema eficiente en ancho de banda.

Si analizamos el canal radio en frecuencia podemos ver que, si analizamos un ancho de banda pequeño, el canal puede parecer plano en frecuencia, es decir no tiene variaciones, mientras que si aumentamos el ancho de banda de análisis en el sistema nos damos cuenta de que realmente tiene desvanecimientos en frecuencia. Estos desvanecimientos se deben al efecto multicarmino, debido a que un símbolo transmitido en el tiempo tiene ecos, por los distintos rayos que se van desplazando en el camino y esto crea interferencia en el símbolo siguiente.

Si el ancho de banda que estamos transmitiendo es pequeño, los símbolos son grandes, por lo que los ecos no tienen tanto impacto sobre los símbolos siguientes. En cambio, si el ancho de banda es mayor, los símbolos son más pequeños, por lo que los ecos crean una interferencia significativa. Esto OFDM lo soluciona utilizando múltiples subportadoras de un ancho de banda pequeño, solapadas entre sí, cada una de ellas portara un único símbolo. Esto consigue que se divida el flujo de datos en muchos flujos paralelos, cada uno de estos con tiempo de símbolo muy grande y por tanto mitigando la interferencia entre símbolos.

OFDM se basa en el concepto de la transformada de Fourier de una ventana temporal. La transformada de Fourier de un pulso rectangular en el tiempo, es una señal sinc en la frecuencia, que tiene nulos en $1 / \text{Tiempo del símbolo}$, lo que permite solapar las múltiples portadoras entre sí, siempre que se garantice que se va a mantener esa separación. La figura 1 muestra el espectro en frecuencia de un pulso rectangular, podemos observar entre una de sus características el ancho del lóbulo principal y el tamaño de los lóbulos laterales, más sobre esto será comentado más adelante.

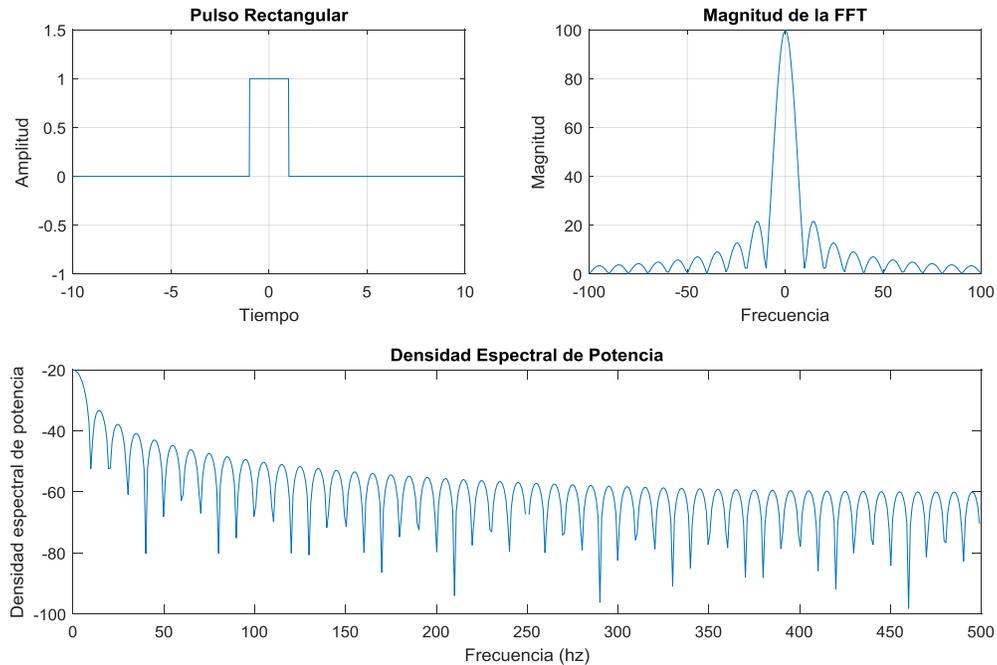


Fig.1. Pulso rectangular y su espectro

Durante mucho tiempo se ha planteado la pregunta: ¿hasta qué punto una función que esta confinada a un ancho de banda finito, también está concentrada en el dominio del tiempo? Las funciones Prolate descritas por David Slepian, son una familia de funciones que incluyen funciones ortogonales que son óptimas en términos de concentración de energía de una función limitada en banda [1].

Se propone un eventanado optimo en el dominio del tiempo para un esquema OFDM en el sentido de concentración espectral. En lugar de suprimir los lóbulos laterales a lo largo de la frecuencia, la potencia de los lóbulos laterales es concentrada en una banda de guarda mientras se suprime máximamente la potencia para un determinado rango de frecuencia. Esto se logra al implementar pulsos óptimos de duración finita, PSWFs, para dar forma al pulso OFDM transmitido [2].

II. Prolate Spheroidal Wave Functions

Limitar una señal en el tiempo disminuye su complejidad computacional, pero a la vez utilizar filtro más cortos o truncados puede causar lóbulos muy grandes en el dominio de la frecuencia. Las funciones Prolate spheroidal Wave functions (PSWFs) combaten este problema de concentración de energía al obtener un pulso limitado en tiempo con un mínimo de fuga espectral fuera de banda o dicho de otra manera un pulso limitado en banda con máxima concentración en un intervalo dado.

Las PSWFs son un set de funciones limitadas en banda que vamos a denotar como $\psi_n(t)$, estas dependen de un total de 4 parámetros: el tiempo continuo t , el orden de la función n , el intervalo para el cual la función es conocida t_o y el parámetro de ancho de banda c , este último parámetro es referido como la frecuencia Slepian. Donde c viene dado por:

$$c = t_o \Omega \quad (1)$$

Donde Ω es el ancho de banda o frecuencia de corte de $\psi_n(t)$ para cierto orden n .

Las PSWFs $\psi_n(t)$, concentradas en un intervalo de $[-t_o, t_o]$ son autofunciones normalizadas del sistema

$$\int_{-t_o}^{t_o} \psi_n(c, t) \frac{\sin \Omega(x-t)}{\pi(x-t)} dt = \psi_n(c, t) \lambda_n(c) \quad (2)$$

λ_n es el autovalor de las funciones y puede ser considerado como el índice de concentración en el intervalo $[-t_o, t_o]$. Según la teoría ψ_n es invariable a la transformada de Fourier. Esta invarianza tanto en el dominio de estado finito e infinito hace que ψ_n sea limitado en banda. Otra propiedad clave de ψ_n es que también obedecen una dualidad de ortogonalidad donde las eigenfunciones de (2) son ortogonales sobre un dominio finito e infinito.

III. Series Slepian

Las Discrete Prolate Spheroidal Sequences o DPSSs se han encontrado que son útiles en el procesamiento digital de señales. Se ha determinado que las DPSS de orden cero es una ventana óptima en el diseño de filtros de impulso finito (FIR). Tanto PSWFs y DPSS son muy similares pero las DPSSs son una derivativa de las PSWFs siendo las DPSS más fáciles de computar y solo son conocidas sobre un intervalo finito.

A medida que el orden N aumenta, la energía de las funciones ψ_n se vuelve menos concentrada en el intervalo $[-t_o, t_o]$ para valores fijos de c . Una buena medida de esto son los valores de los eigenvalores λ_n .

Los primeros $\frac{2c}{\pi}$ eigenvalores son cercanos a 1 por una pequeña diferencia δ , donde $\delta > 0$. Para n más allá de $\frac{2c}{\pi}$ los eigenvalores descienden cerca de 0. Para $\delta \cong 0$, casi toda la energía de ψ_n de un orden dado n para todo el $t \in [-t_o, t_o]$ esta contenida dentro de estos límites [3].

Las PSWFs también exhiben comportamientos de simetría como las series de Fourier, se puede observar que los órdenes pares de ψ_n son pares y los órdenes impares de ψ_n son impares. La figura

2a muestra los primeros 4 órdenes pares de las series Slepian, la figura 2b muestra los primeros 4 órdenes impares

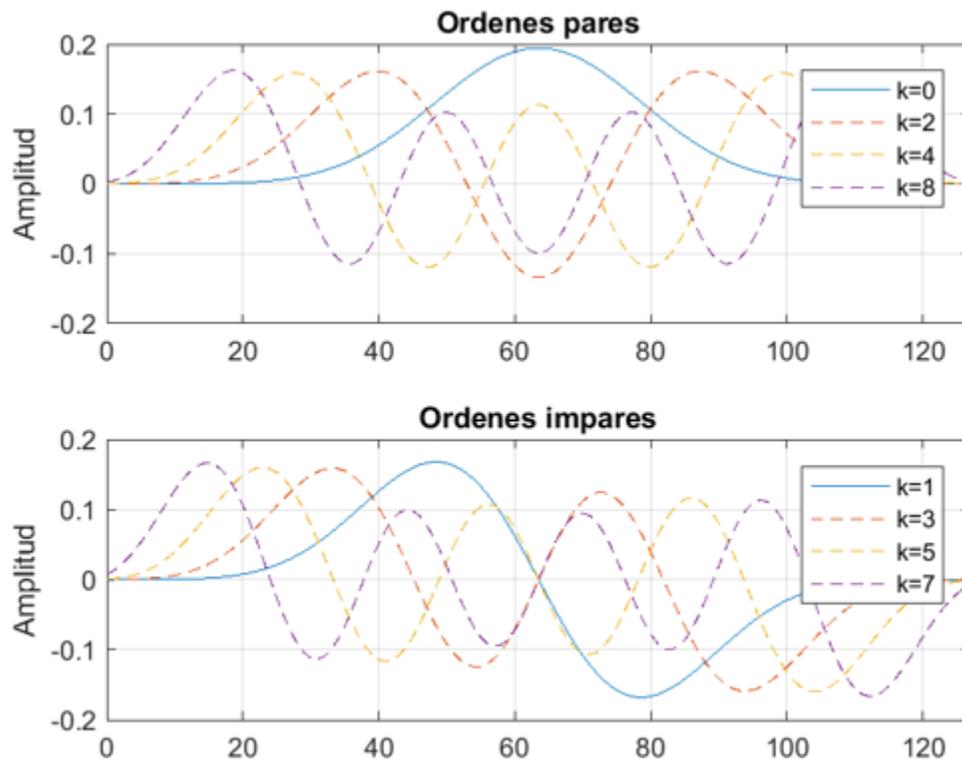


Fig. 2. Ordenes pares e impares de las secuencias Slepian

La función Prolate de orden cero, esta máximamente concentrada en un intervalo de tiempo dado. Las funciones trigonométricas son ortogonales y completas en un intervalo finito, sin embargo, las PSWFs tienen la libertad de formar una base ortonormal sobre infinito también. Otra propiedad interesante es que la transformada de Fourier sobre un intervalo finito e infinito es una versión escalada de sí misma. Además, son potencialmente óptimas por encima de otras expansiones ortogonales para funciones discontinuas tal como las ondas cuadradas. [4]

IV. Escenario General

El escenario general de este proyecto consiste en primero describir la generación de las funciones DPSS en el entorno de Matlab y analizar sus principales características, una vez estudiadas estas funciones, procederemos a realizar una comparativa con otro tipo de funciones para analizar sus ventajas y desventajas como ventana temporal en nuestro sistema de modulación multiportadoras y finalmente ponerlas a prueba en un sistema OFDM para analizar su desempeño.

IV.1 Generación de señales DPSS

Matlab nos facilita la generación de las DPSS, con una sola sentencia podemos generar la familia de funciones con una determinada longitud de secuencia y un ancho de banda determinado [5].

```
[dps_seq,lambda] = dpss(seq_length,time_halfbandwidth,num_seq)
```

Devuelve las primeras (2 veces *time_halfbandwidth*) funciones prolate de longitud *seq_length*. *dps_seq* es una matriz con *seq_length* filas y (2 veces *time_halfbandwidth*) columnas. *Time_halfbandwidth* debe ser menor a *seq_length/2*

Esta función devuelve las primeras **num_seq** funciones Prolate con una longitud **seq_length** y un ancho de banda **time_halfbandwidth/seq_length**. Nos devuelve también los ratios de concentración de energía en el dominio de frecuencia en el vector **dps_seq**. Estos ratios representan la cantidad de energía en la banda de paso $[-W, W]$ al total de energía de $[-Fs/2, Fs/2]$, donde F_s es la tasa de muestreo. **Lambda** es un vector columna igual en longitud al número de secuencias Slepian.

El producto *time_halfbandwidth* es NW donde N es la longitud de la secuencia y $[-W, W]$ es el ancho de banda efectivo de la secuencia. Al construir la secuencia Slepian se escoge la longitud de secuencia deseada (*seq_length*) y ancho de banda $2W$. Tanto la longitud de secuencia y el ancho de banda afectan cuantas secuencias Slepian tienen ratios de concentración cercanos a 1. Como regla, hay $2W-1$ secuencias Slepian con ratios de concentración de energía aproximadamente igual a 1. Más allá de esas $2W-1$ secuencias Slepian los ratios se empiezan a acercarse a 0. Valores comunes son 2.5, 3, 3.5 y 4.

El siguiente código vamos a generar los primeros 4 órdenes de la secuencia Slepian con un ancho de banda de $4/128$. La figura 3 nos muestra los primeros 4 órdenes de las secuencias Slepian.

```
[E,V] = dpss(128,4,10); % computa los 4 primeros ordenes
                        % de las DPSS con BW 4/128.
%Orden 0
figure(1);
plot(t,E(:,1),'k'); axis tight; ylim([-0.4 0.4]);
title('DPSS para k=0, N=128, W=4/N ');
%4 primeros ordenes
figure(5);
plot(t,E),axis tight;ylim([-0.4 0.4])
title('DPSS para k=0,1,2,3, N=128, W=4/N ');
legend('k=0','k=1','k=2','k=3')
```

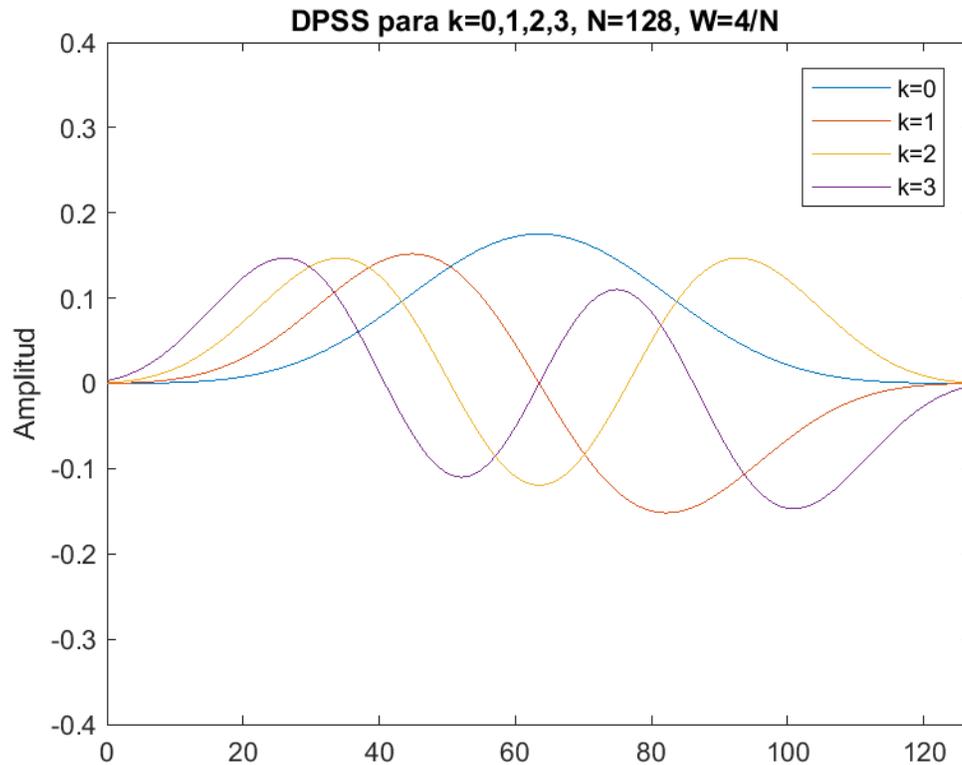


Fig. 3. 4 primeros ordenes de las DPSS.

IV.2 Espectro de señales DPSS

Por la teoría de Fourier, toda forma de onda puede ser representada como la suma de un número de sinusoides, cada una con una amplitud y fase. Esta representación se conoce como el espectro de una señal, o su representación en el dominio de la frecuencia. A veces es más fácil analizar las señales en termino de representación espectral.

El espectro de una señal nos puede indicar aspectos de una señal que no son tan obvios al ver su representación en el tiempo. Por ejemplo, la figura 4 nos muestra una señal representada en el dominio de tiempo que es periódica, podemos observar que tiene un periodo de 0.2seg, pero no podemos obtener más información sobre esta señal.

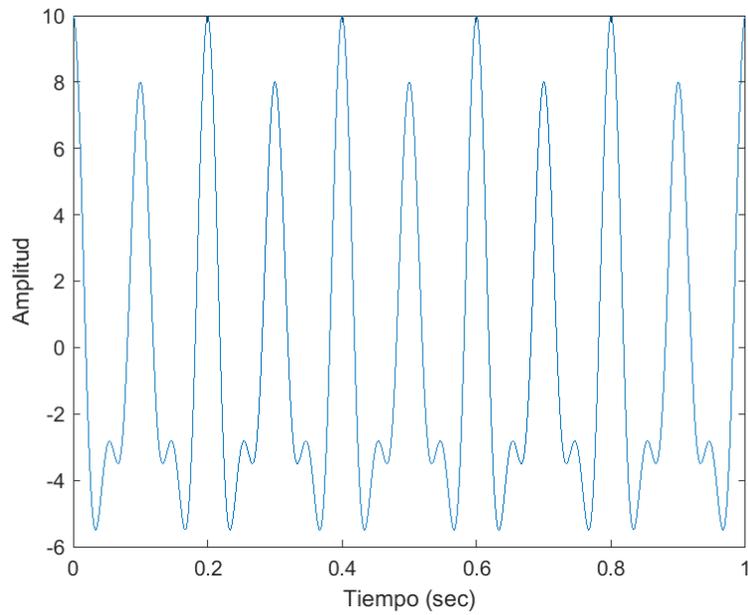


Fig. 4. Señal periódica en el tiempo.

La figura 5 en cambio nos muestra el espectro de magnitud de la señal, con esto nos damos cuenta de que la señal está compuesta por 3 componentes de frecuencia cada una con su respectiva magnitud conforme al peso de cada componente.

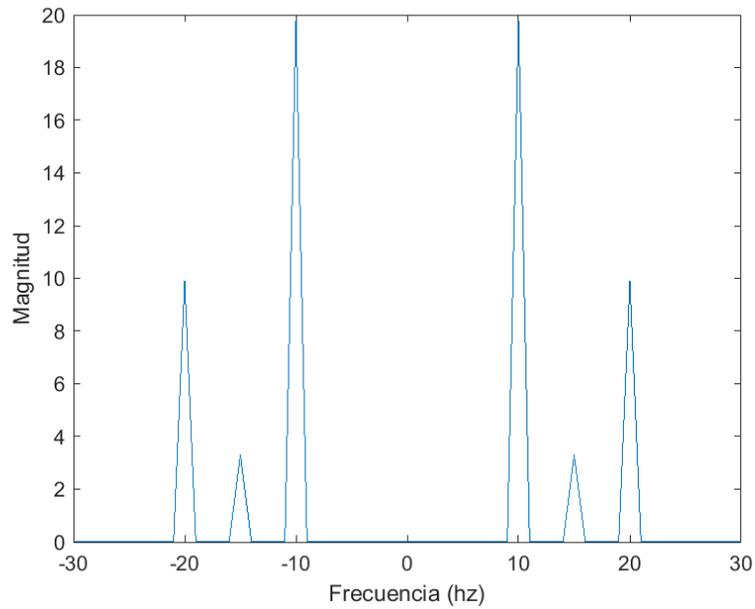


Fig. 5. Espectro de la señal.

El espectro se lo obtiene de la siguiente forma: Dada una señal discreta en el tiempo $x[n]$, podemos determinar su respuesta en frecuencia utilizando la transformada discreta de Fourier DFT.

$$X[k] \equiv \sum_{n=0}^{N-1} x[n] e^{-\frac{j2\pi kn}{N}}, k = 0, 1, 2, 3, \dots, N - 1 \quad (3)$$

La DFT determina los ‘pesos’ sinusoidales mediante el producto interno de sinusoides y la señal. A continuación, analizaremos algunos aspectos de las secuencias Slepian. La figura 6 nos muestra los 4 primeros ordenes de las secuencias Slepian. Podemos observar que el orden determina cuantos cruces por cero tiene la función. Esto se puede ver reflejado en la figura 7 que nos muestra el espectro de magnitud para los 4 órdenes.

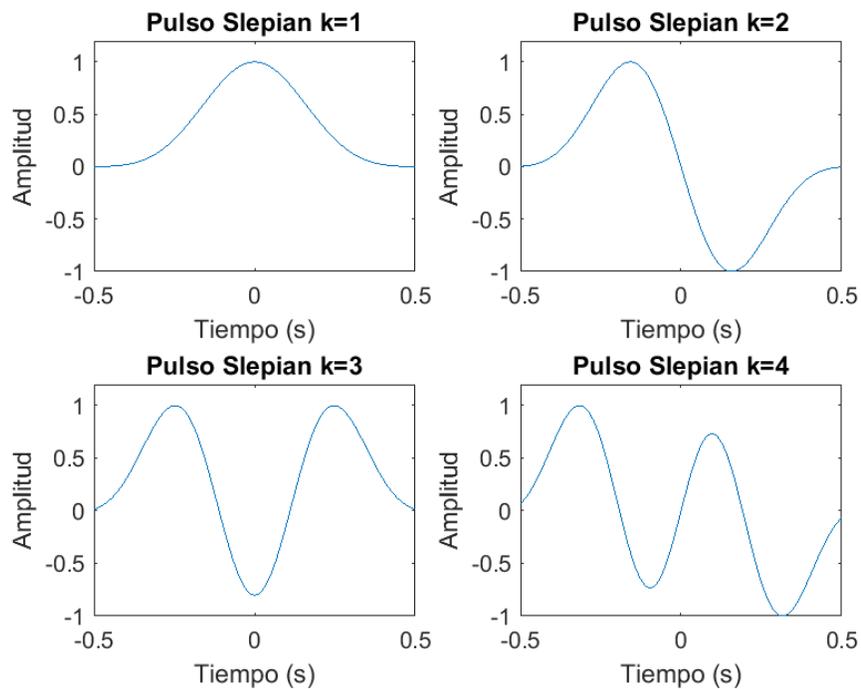


Fig. 6. Secuencia Slepian para k=0,1,2,3

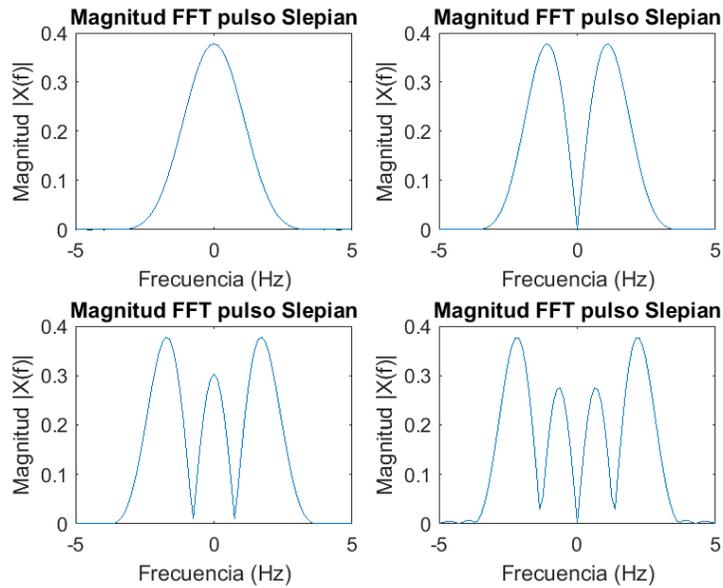


Fig. 7. Espectro de magnitud de las secuencias Slepian

Una vez definidas las señales, podemos graficar la densidad espectral de potencia. La densidad espectral de una señal es una función matemática que nos informa como está distribuida la potencia o energía de dicha señal sobre las distintas frecuencias de las que está formada. La figura 8 y 9 muestra la densidad espectral de potencia de los 4 primeros ordenes de las funciones Prolate, podemos observar como a medida que va aumentando el orden de las funciones, los lóbulos laterales van incrementando en nivel, por lo que el pulso óptimo es el de orden Zero, ya que es el que más concentra la energía en la banda de paso [5].

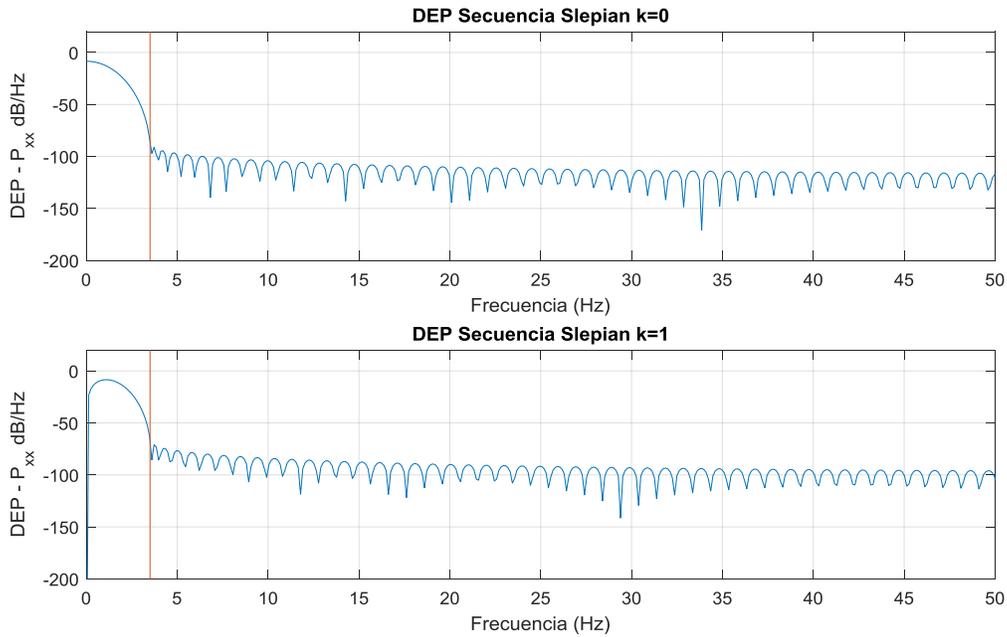


Fig. 8. Densidad Espectral de Potencia para k=0,1

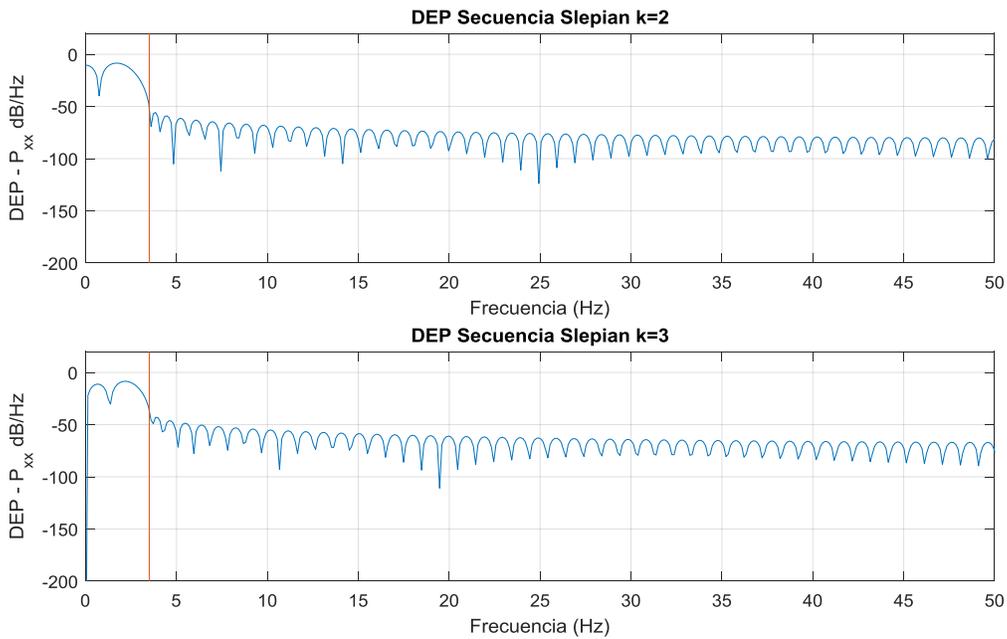


Fig. 9. Densidad Espectral de Potencia para k=2,3

La figura 10 nos muestra una superposición de los 4 órdenes de las secuencias, para poder observar de mejor manera como el nivel de los lóbulos laterales varía de acuerdo con el orden de las funciones.

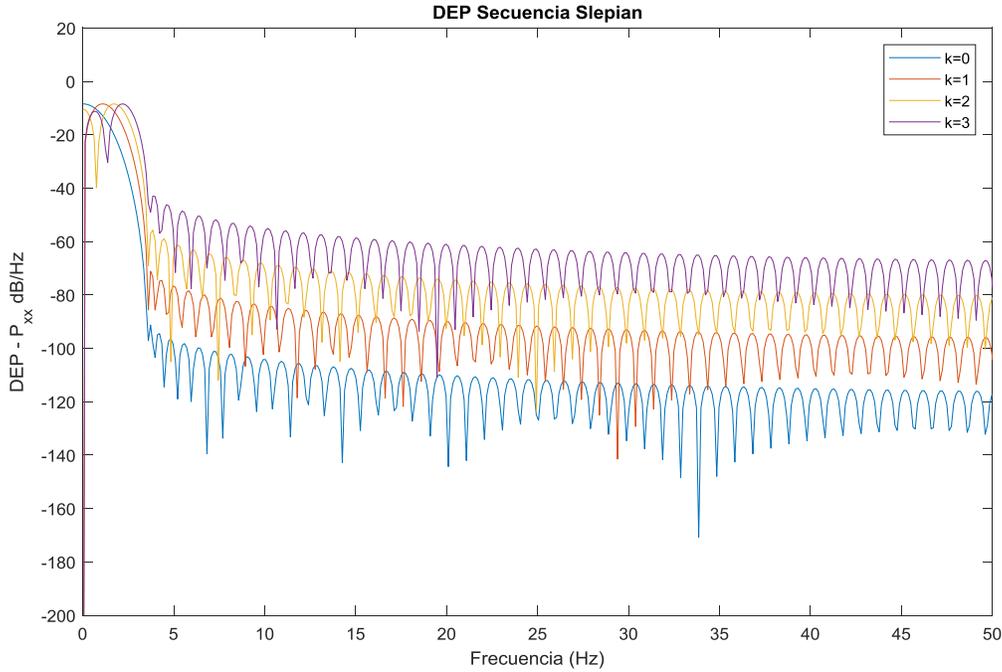


Fig. 10. Espectro combinado de los 4 primeros ordenes

IV.3 Características de las secuencias Slepian

Los parámetros más importantes que considerar en la ventana DPSS son la longitud de la ventana N , así como el ancho de banda de concentración. El ancho del lóbulo principal está directamente relacionado con el parámetro de concentración mas no es tan dependiente de la longitud de la ventana. El nivel de los lóbulos laterales es una función tanto de la longitud como del ancho de banda de concentración. Primero vamos a observar el caso de que dos secuencias tengan la misma longitud, pero distintos anchos de banda de concentración, en este caso se comparara con un ancho de banda de concentración de 2.5 y 4. La figura 11 y 12 nos muestran cómo afecta el parámetro el ancho de banda de concentración a nuestra secuencia. Mientras mayor sea el ancho de banda de concentración, el lóbulo principal será más ancho, pero a su vez los lóbulos laterales tienen un nivel de energía menor. El aumentar o disminuir el tamaño de la secuencia no influye en el nivel de concentración [6].

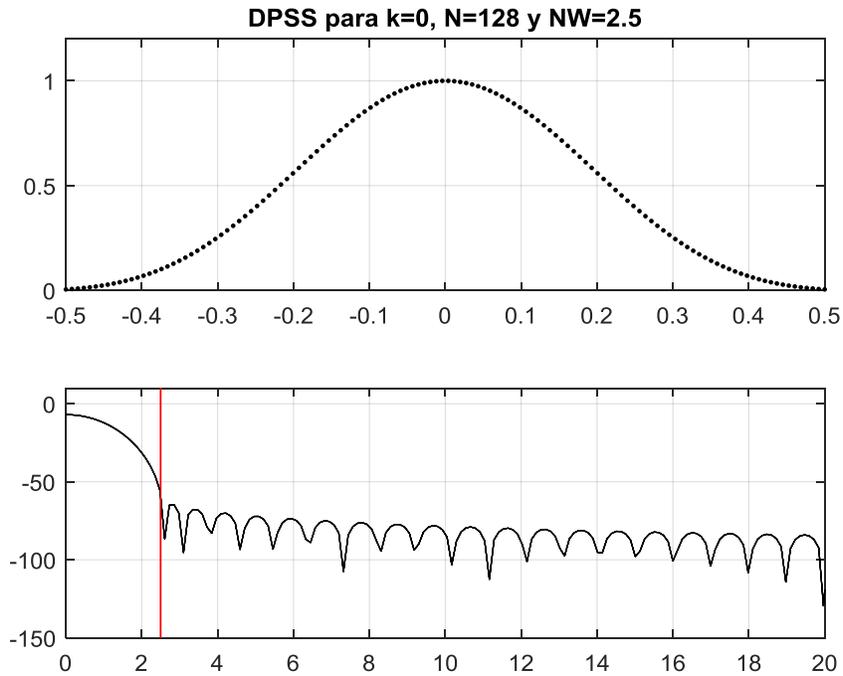


Fig. 11. Secuencia Slepian de longitud $N=128$ y $NW=2.5$

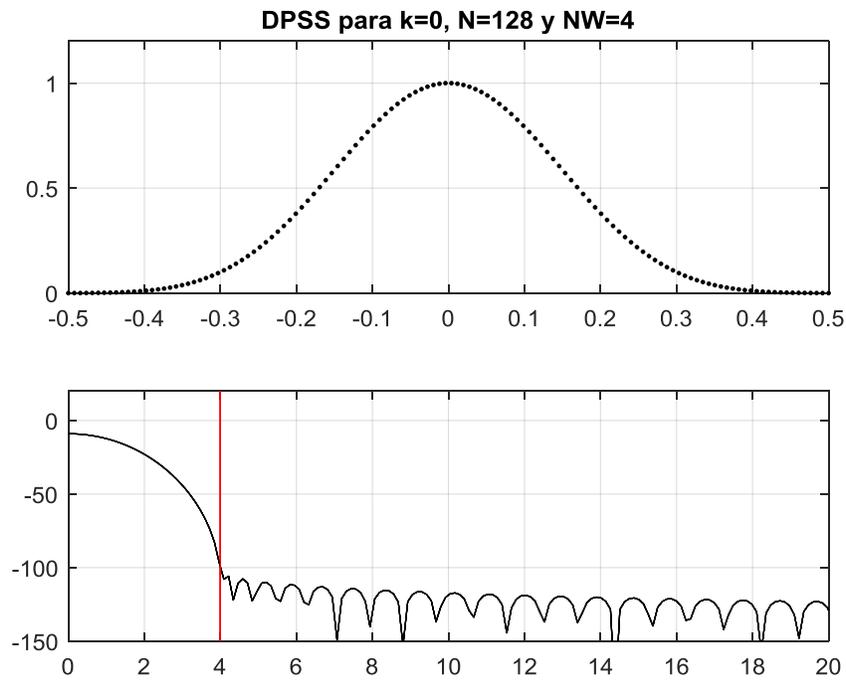


Fig. 12. Secuencia Slepian de longitud $N=128$ y $NW=4$

Como se mencionó al inicio, el producto NW nos determina el número de secuencias cuyos ratios de concentración se aproximan a la unidad. Dicho esto, vamos a analizar el efecto del producto NW en el siguiente escenario. Se generan los 10 primeros ordenes de las secuencia Slepian, con una longitud de secuencia de 128 muestras y utilizando los valores de $NW=2.5$ y $NW=4$. La figura 13 nos indica cuantas secuencias tienen una concentración cercana a la unidad, por teoría sabemos que $2NW-1$ secuencias cumplen con este criterio y como se observa en efecto 4 secuencias tiene una concentración por encima del 90%. La figura 14 nos demuestra que para un parámetro $NW=4$ también se cumple que el número de secuencias cercanas a la unidad en este caso es de 7 secuencias.

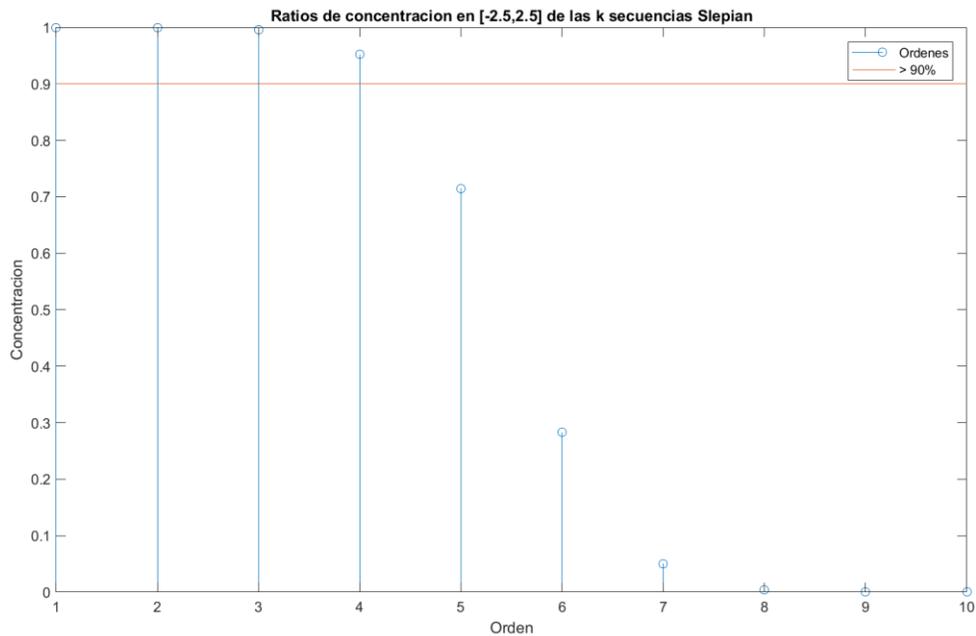


Fig. 13 Ratios de concentración para $NW=2.5$

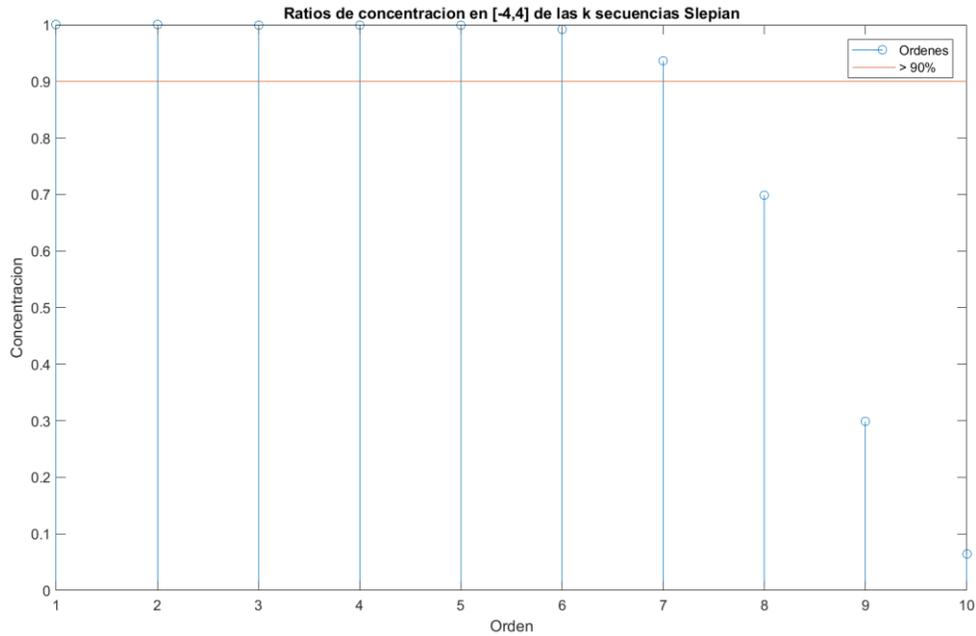


Fig. 14. Ratios de concentración para $NW=4$

Esto se puede apreciar graficando las 10 primeras secuencias para ambos casos. Podemos observar en la figura 15 como afecta el parámetro NW a la forma de la secuencia, al aumentar el factor NW , la secuencia se vuelve más angosta en el tiempo, lo contrario pasa si el factor NW disminuye. Esto tendrá un efecto en la frecuencia que será que el pulso este más concentrado en la banda de paso. Podemos también notar que el orden de las secuencias determina el número de cruces por cero que tiene cada secuencia [7].

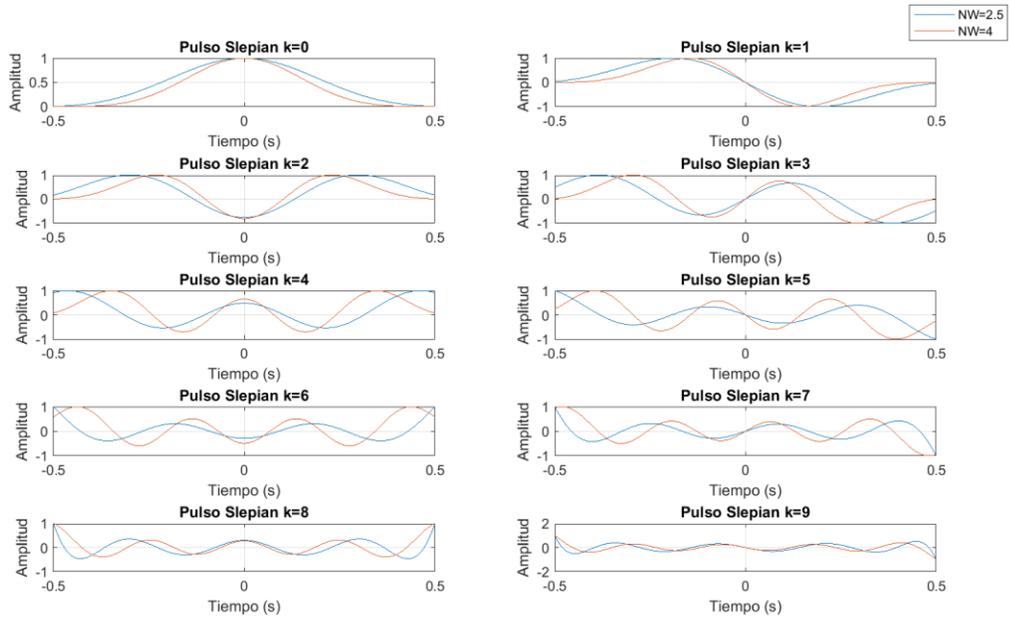


Fig. 15. Primeros 10 órdenes de las secuencias Slepian

La figura 16 nos muestra el espectro para las secuencias y como para los distintos órdenes y dependiendo del producto NW la concentración va desde valores cercanos a la unidad a volverse prácticamente 0.

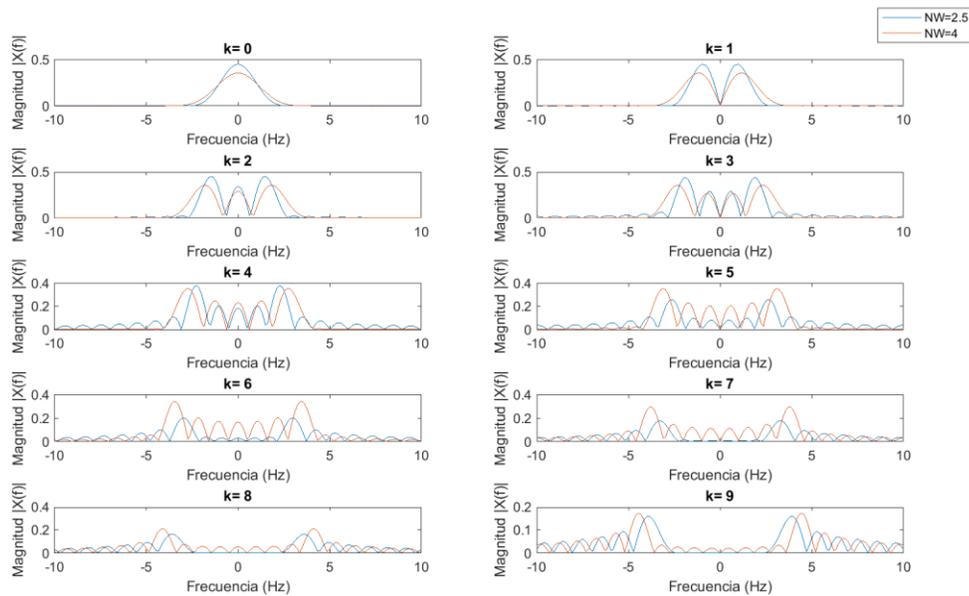


Fig. 16. Espectro primeros 10 órdenes de las secuencias Slepian

IV.4 Comparación con otras formas de onda

A continuación, vamos a comparar los diferentes tipos de pulsos, para esta comparativa se seleccionaron los pulsos gaussianos, rectangular y el primer orden de las Slepian. La figura 17 nos muestra los 3 tipos de ventana temporal que van a ser analizadas: gaussiana, Slepian y rectangular. La figura 18 nos indica el espectro de amplitud de las 3 ventanas temporales. Podemos observar que la transformada de Fourier de una función gaussiana es también una función gaussiana, de la secuencia Slepian podemos observar cómo es la que muestra mayor concentración dentro de la banda de paso y del pulso rectangular podemos ver los lóbulos laterales que ocurren cada N/T_s [8].

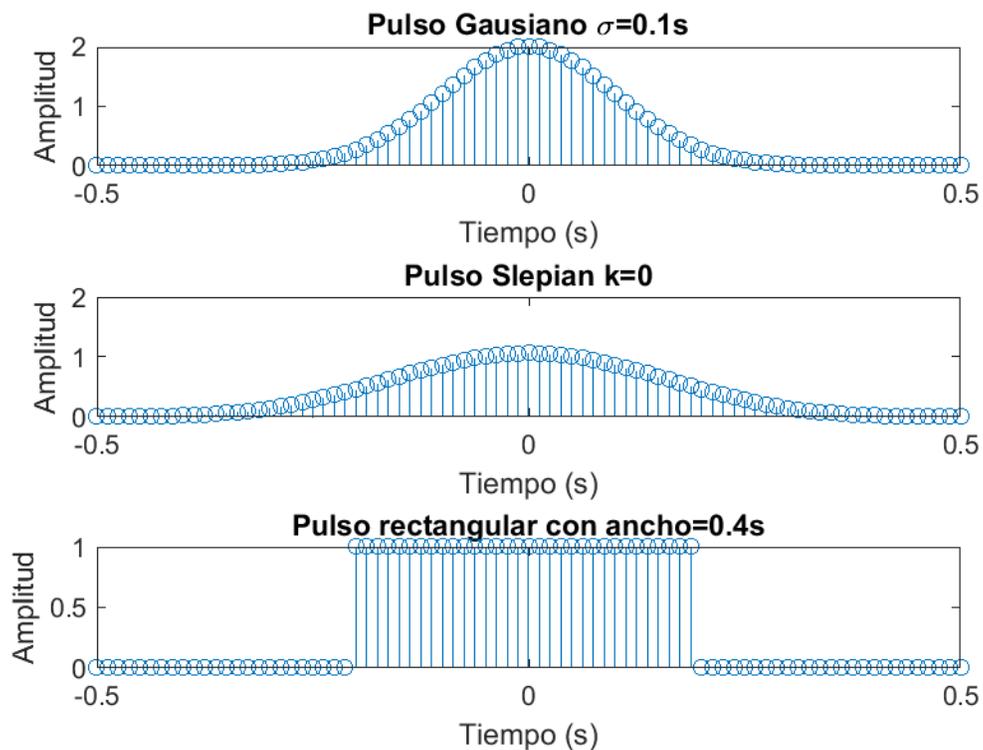


Fig. 17. Formas de pulso Gaussiano, Slepian y Rectangular

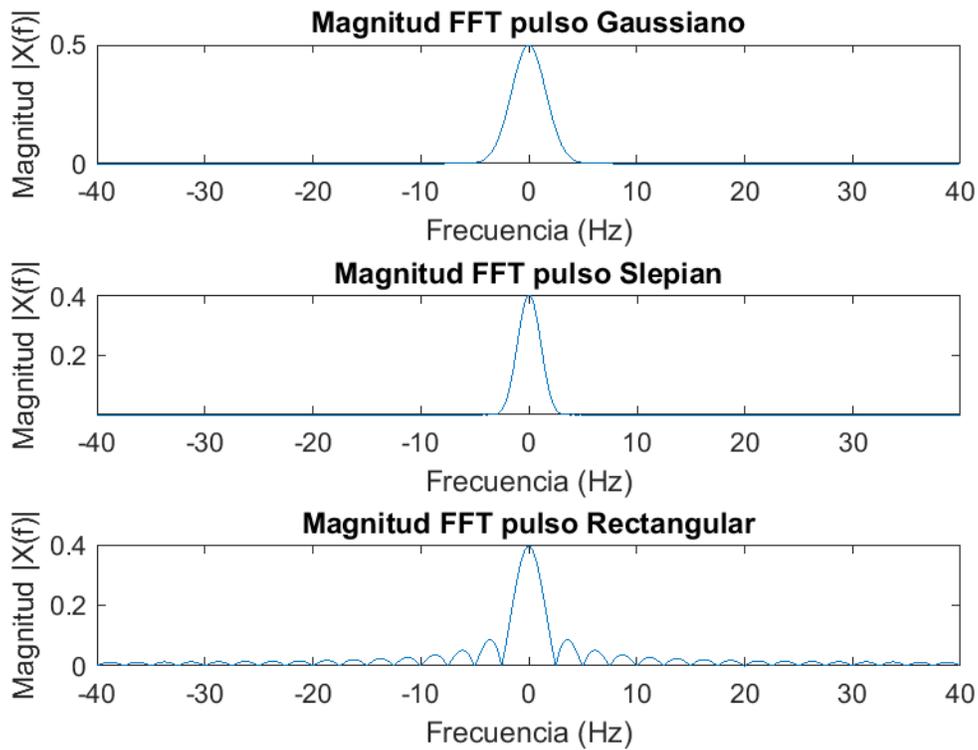


Fig. 18. Espectro de magnitud de los pulsos

Podemos comparar las relaciones entre el lóbulo principal y los lóbulos laterales de las 3 ventanas temporales utilizadas. El ancho del lóbulo principal limita la resolución de frecuencia de la señal eventanada. Por ende, la habilidad de distinguir dos componentes de frecuencia espaciados muy cercanamente aumenta a medida que el lóbulo principal se estrecha. A medida que el lóbulo principal se estrecha y la resolución espectral mejora, la energía de la ventana se extiende a los lóbulos secundarios aumentando la fuga espectral. La figura 19 nos muestra como el ancho del lóbulo principal se vuelve más estrecho para cada ventana temporal y también se observa como en el caso del pulso rectangular, la energía de los lóbulos secundarios es mayor con respecto al lóbulo principal. De esto podemos concluir que la secuencia Slepian es una muy buena alternativa ya que nos provee de una buena resolución espectral sin incrementar la fuga espectral en los lóbulos secundarios.

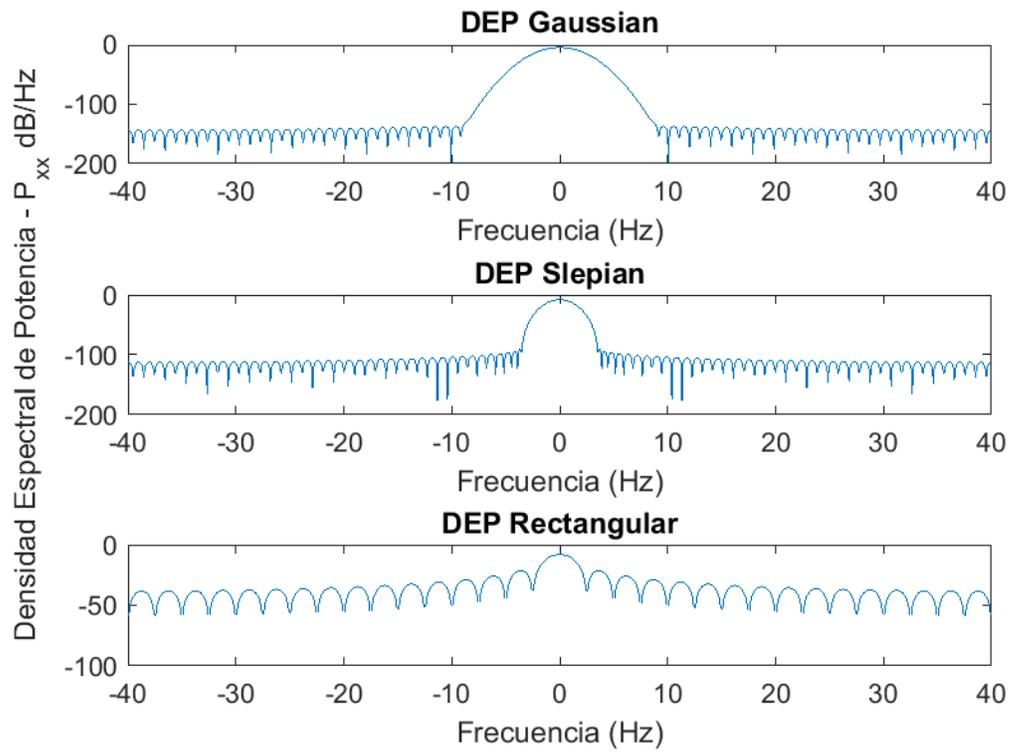


Fig. 19. Comparativa lóbulos principales

La figura 20 nos permite apreciar de mejor manera el ancho del lóbulo principal para cada uno de los pulsos utilizados, así como el nivel que alcanzan los lóbulos laterales. Estos nuevos pulsos permiten reducir la banda de guarda en los sistemas OFDM comparado con los sistemas que utilizan pulsos rectangulares.

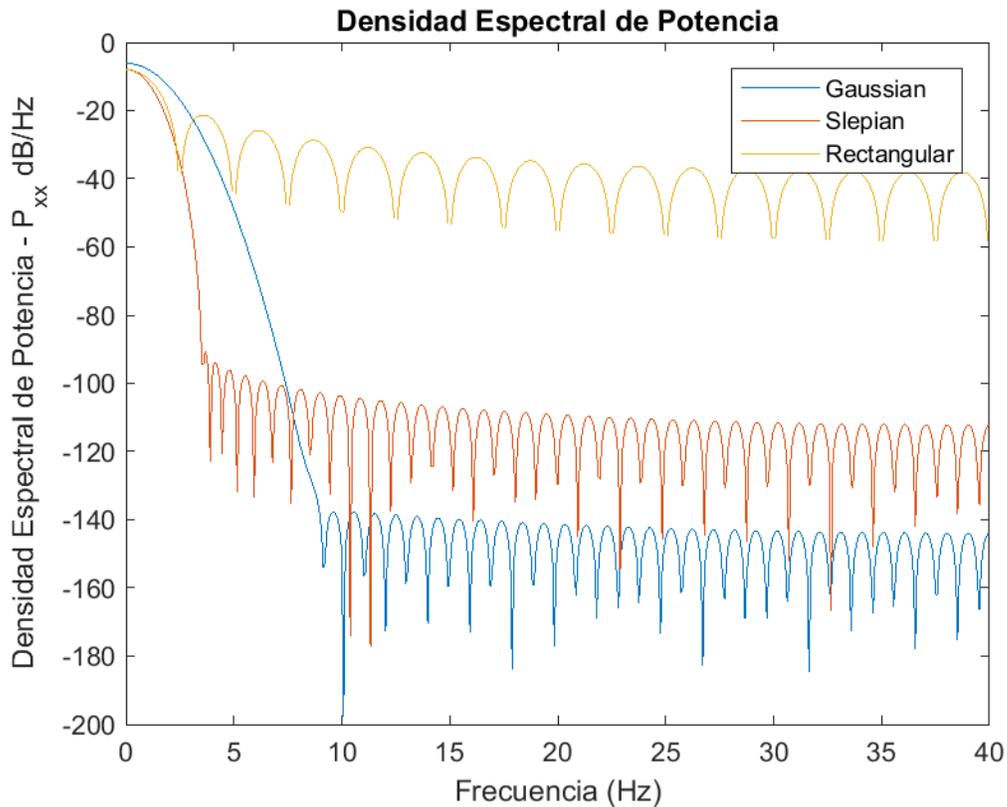


Fig. 20. Comparativa de la DEP de los pulsos.

V. Modulaciones Multiportadoras

La mayoría de los estándares de comunicaciones inalámbricas adoptaron OFDM como técnica de modulación. El concepto de OFDM es la transmisión multiportadoras, la cual consiste en subportadoras ortogonales. El principio del esquema OFDM es dividir los canales de alta tasa de datos en sub-canales de bajas tasas de datos, cada uno de estos luego es modulado con la ayuda de diferentes subportadoras. Uno de los mayores beneficios de un sistema OFDM es transformar canales selectivos en frecuencia a un número de canales aproximadamente planos de frecuencia, con una duración de símbolo larga y un retardo de propagación corto, lo cual ayuda a maximizar la eficiencia espectral.

Los sistemas OFDM necesitan una sincronización de frecuencia muy precisa, ya que una desviación en la frecuencia destruye la ortogonalidad entre las subportadoras lo que produce interferencia entre portadoras (ICI). En un esquema típico OFDM, cada símbolo está definido como la suma de N símbolos de datos que son modulados con la ayuda de varias técnicas de modulación, como lo son QAM o QPSK

El dominio de frecuencia de una señal OFDM está compuesto por múltiples sincs en cada subportadora de frecuencia. Para garantizar la ortogonalidad y por ende cero interferencias entre subportadoras, deben cumplirse que en el pico de cada subportadora exista un nulo espectral de las demás subportadoras. Cuando ocurre un desfase en frecuencia se pierde la ortogonalidad debido a que el pico de cada subportadora no coincide con un nulo espectral. Por ende, potencia de un lóbulo lateral existe en cada subportadora de frecuencia. El objetivo es lograr que los lóbulos laterales disminuyan en amplitud dejando el lóbulo principal sin ser afectado. Las funciones Prolate nos dan pulsos ortogonales y poseen un ancho de pulso constante sin importar el orden del pulso. Esta propiedad es muy importante ya que elimina la interferencia entre símbolos.

En OFDM convencional cada símbolo es transmitido mediante un pulso rectangular de duración T lo cual conlleva un espectro de sinc para cada subportadora. Cada subportadora debe estar espaciada $1/T$, para que los ceros de las demás subportadoras coincidan con el máximo de cada una y garantizar una transmisión sin interferencia con otras subportadoras. Esto lleva a un problema en caso de que exista alguna dispersión en frecuencia ya que no estará correctamente sincronizado causando Interferencia entre portadoras (ICI). Estas secuencias son menos susceptibles a la dispersión de frecuencia de las funciones sinc, por lo que son más apropiadas para una transmisión OFDM.

Un Sistema OFDM puede ser dividido en categorías de acuerdo con el tipo de forma de pulso utilizado. La primera categoría utiliza pulsos de longitud de un intervalo de símbolo. Esto da un solape en frecuencia entre todos los canales en el sistema, pero no interfiere debido a que los pulsos modulados son ortogonales entre sí. La mayoría de los sistemas en esta categoría utilizan pulsos rectangulares. Pulsos así de cortos tiene lóbulos laterales relativamente largo, lo cual es una desventaja, especialmente cuando el número de canales es pequeño. Para los pulsos rectangulares, el nivel de los lóbulos laterales decae $1/f$. Varios cientos de canales son necesarios para tener un sistema eficiente en ancho de banda.

La segunda categoría utiliza pulsos de duración infinita, los pulsos están limitados en banda tal que se solapan en frecuencia solo entre canales adyacentes. Los pulsos infinitamente largos deben ser truncados para que sea realizable. Se utilizaron una nueva clase de pulsos para OFDM. Estos pulsos son de duración finita, pero más largos que un intervalo de símbolo. Los pulsos son generados por un proceso de optimización en el cual la energía fuera de banda se minimiza bajo la restricción de cero interferencias entre canales y símbolos. [9]

V.1 Comparativa pulso rectangular y pulso Slepian

Una vez estudiado el comportamiento de las distintas secuencias Slepian podemos proceder a implementarlas como ventana temporal en un sistema de modulación multiportadoras. Para el diseño de filtros la ventana general que se suele elegir es una secuencia de duración finita, simétrica y discreta en el tiempo que muestra un decaimiento en los bordes. La transformada discreta de Fourier de dicha secuencia presenta un lóbulo principal y unos lóbulos laterales de menor amplitud que aparecen debido al truncado de la señal. La figura 21 nos muestra una secuencia rectangular en el dominio de tiempo, también podemos observar la respuesta en frecuencia de esta. La figura 22 nos muestra lo mismo, pero utilizando una secuencia Slepian, podemos notar como el lóbulo principal es más ancho que en la ventana rectangular, pero los lóbulos laterales tienen un nivel menor.

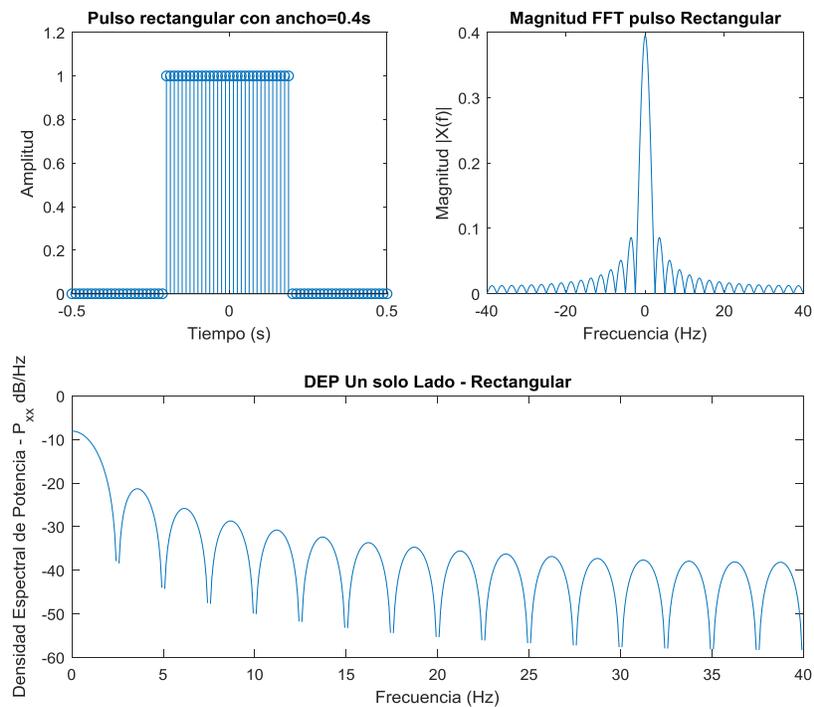


Fig. 21. Pulso rectangular y su espectro

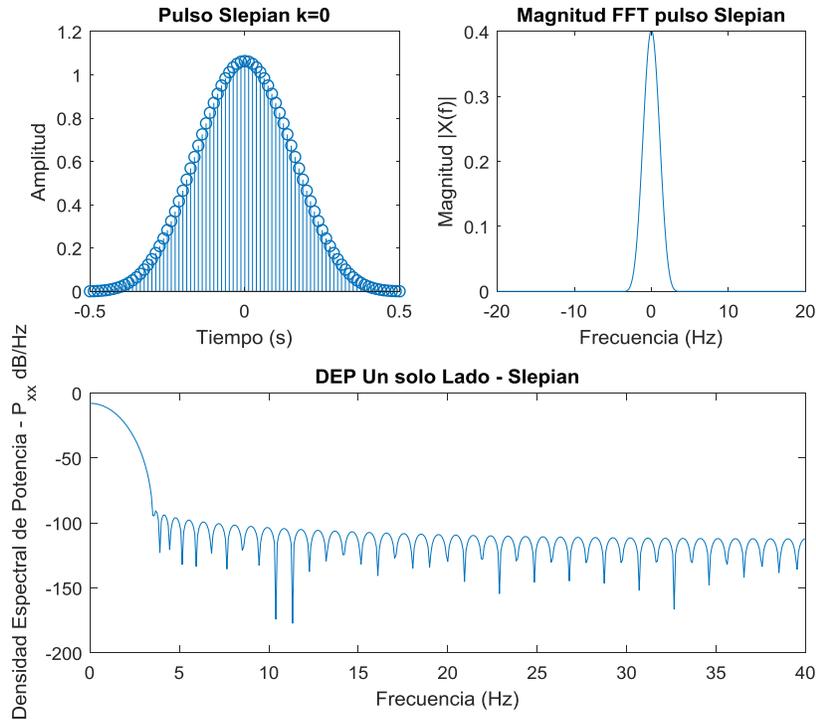


Fig. 22. Pulso Slepian y su espectro

Para el diseño de filtros la ventana general que se suele elegir es una secuencia de duración finita, simétrica y discreta en el tiempo que muestra un decaimiento en los bordes. La transformada discreta de Fourier de dicha secuencia presenta un lóbulo principal y unos lóbulos laterales de menor amplitud que aparecen debido al truncado de la señal. El pulso rectangular muestra una mala concentración de energía comparada con las secuencias Slepian, pero su ventaja yace en que la interferencia se puede eliminar simplemente con el uso de un intervalo de guarda. La figura 23 muestra una buena concentración de energía de las secuencias Slepian con respecto al pulso rectangular, alrededor de la frecuencia de portadora se obtiene un espectro prácticamente plano mientras que se puede observar un decaimiento mucho más rápido para las componentes espectrales fuera de banda, lo cual hace al sistema menos susceptible a interferencia de banda estrecha.

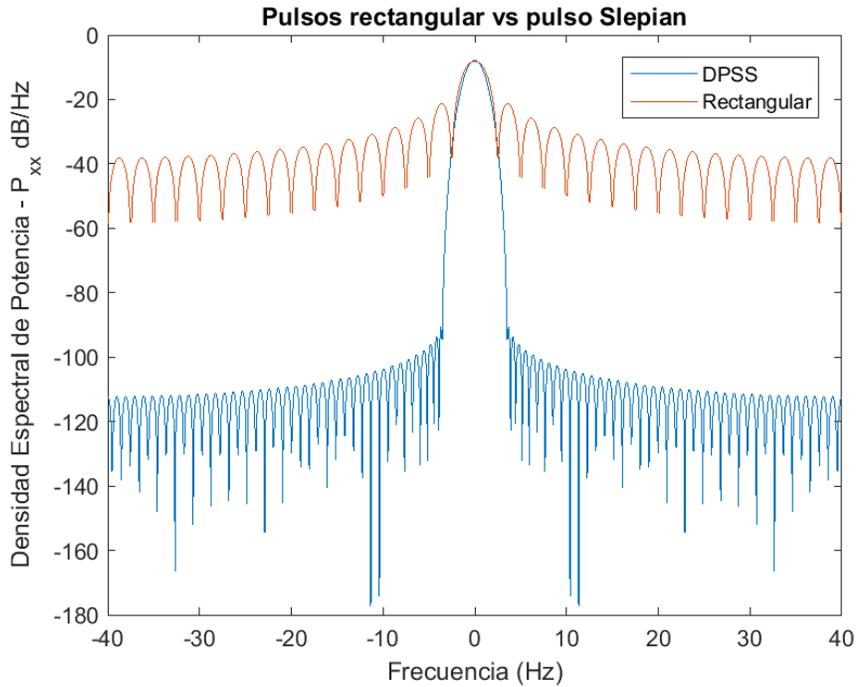


Fig. 23. Comparativa entre pulso rectangular y pulso Slepian

Cabe recalcar que existe una compensación que se debe tomar para tener pulsos más angostos las secuencias Slepian deben aumentar en longitud N y debemos mantener el número de símbolos M que vamos a transmitir fijos. Esto conlleva transmitir secuencias más largas, entonces es posible incrementar la eficiencia espectral de OFDM al incrementar el número de subportadoras cuyas frecuencias no se solapen y con menor ICI.

VI. Simulación

Para el escenario propuesto vamos a simular un sistema OFDM, la figura 24 muestra el diagrama de bloques de este sistema, mientras que la figura 25 nos muestra más en detalle los bloques utilizados y podemos observar cómo se añade un bloque de modelado de pulso, este será el responsable de implementar las secuencia Slepian en nuestro sistema.



Fig. 24 Bloques de simulación

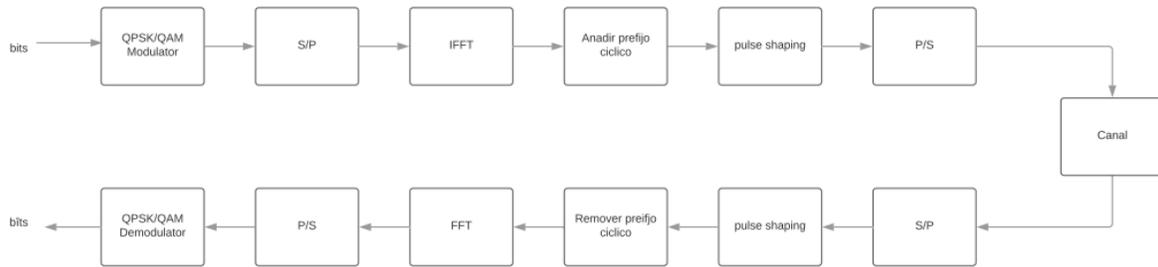


Fig. 25 Modulador y Demodulador OFDM

Para esta simulación utilizaremos los siguientes parámetros, especificados en la figura 26 y lo primero será simular un sistema OFDM convencional, sin modulado de pulso adicional.

Parámetros de la señal	valores
Puntos de la FFT	1024
Prefijo cíclico (muestras)	72
Numero de subportadoras de datos	600(50 Bloques de recursos con 12 subportadoras cada una)
Numero de subportadoras nulas	424
Modulación	QPSK,QAM

Fig. 26 parámetros de la simulación

Para la simulación lo primero que debemos hacer es definir los parámetros con los que se va a realizar.

```

NFFT=1024;      %Tamaño de la FFT
nRB=50;        %Numero de bloques de recursos
nSub=12;       %Numero de subportadoras por RB
cp=72;         %longitud del prefijo ciclico
M=4;          %Tamaño de la constelacion
k=log2(M);    %Numero de bits por simbolo
SNRdb=15;     %SNR
s=rng(111);
Lfiltro=513;  %Longitud del filtro

```

A continuación, debemos generar el flujo de bits que se van a transmitir, así como definir el modulador, en este caso QPSK y aplicar la modulación a los bits generados. La figura 27 muestra un extracto de los bits generados aleatoriamente, también nos muestra lo que obtenemos a la salida del modulador.

```

%% Generacion de bits
bits=randi([0 1],nSubDatos*k,1);

%% Modulador QAM
qamMapper = comm.RectangularQAMModulator( ...
    'ModulationOrder', M, 'BitInput', true, ...
    'NormalizationMethod', 'Average power');
bitsMod=qamMapper(bits);
scatterplot(bitsMod)

```

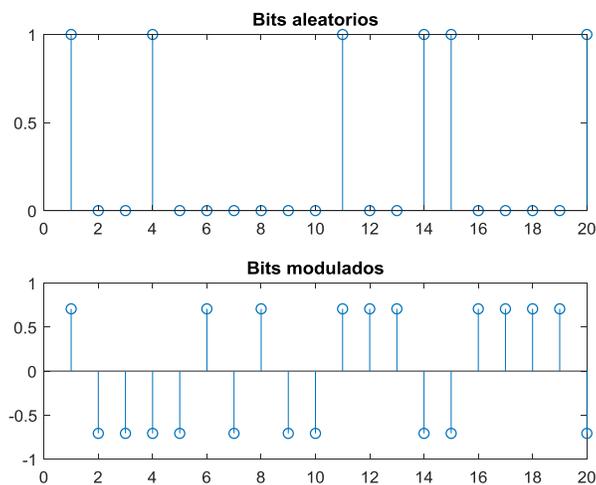


Fig. 27. Bits a transmitir

El siguiente paso será llenar con ceros donde corresponden las subportadoras nulas y aplicar la IFFT para generar los símbolos OFDM.

```

%% llenar con ceros
SimbolosOFDM=[zeros((NFFT-nSubDatos)/2,1);bitsMod;zeros((NFFT-nSubDatos)/2,1)];
%% IFFT
Ifftout=ifft(fftshift(SimbolosOFDM));

```

Luego se procede a agregar el prefijo cíclico, la figura 28 nos muestra el espectro OFDM.

```

%% Agregar prefijo ciclico
TxOFDMcp=[Ifftout(NFFT-cp+1:NFFT); Ifftout];
% Graficar el espectro
[dep,f]=periodogram(TxOFDMcp,rectwin(length(TxOFDMcp)), NFFT*2,1,'centered');
plot(f,10*log10(dep));

```

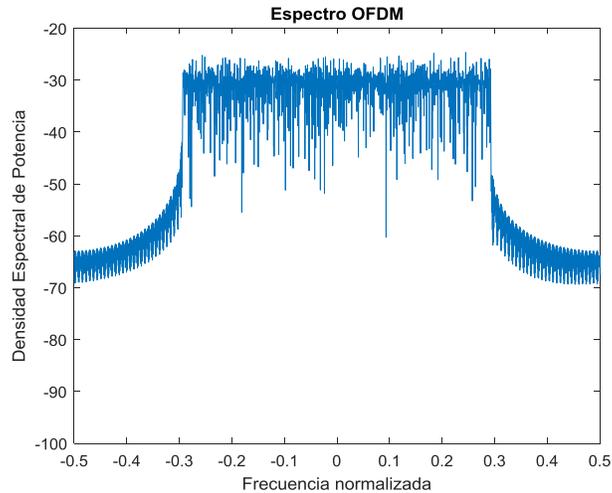


Fig. 28. Espectro OFDM

Lo siguiente será enviar los símbolos por el canal AWGN y realiza el proceso inverso. Primero se elimina el prefijo cíclico y luego se procede a realizar la FFT, una vez seleccionadas las subportadoras de datos se procede a realizar la demodulación y detección de símbolos y cálculo de errores. La figura 29 nos muestra la constelación obtenida en el receptor y la figura 30 nos muestra la gráfica de BER vs SNR

```

%% Eliminar prefijo ciclico
Rx_cp=Rx_OFDM(cp+1:end);

% Realizar la FFT
Rx_simbolos=fftshift(fft(Rx_cp));

% Seleccionar subportadoras de datos
Rx_datos=Rx_simbolos(213:NFFT-212);
scatterplot(Rx_datos)

%% Demodulador
qamDemod = comm.RectangularQAMDemodulator('ModulationOrder', ...
    M, 'BitOutput', true, ...
    'NormalizationMethod', 'Average power');
BER = comm.ErrorRate;

% Decisor y calculo de errores
Bits_rx = qamDemod(Rx_datos);
ber = BER(bits, rxBits);

disp(['BER = ' num2str(ber(1)) ' con una SNR = ' ...
    num2str(SNRdb) ' dB']);

```

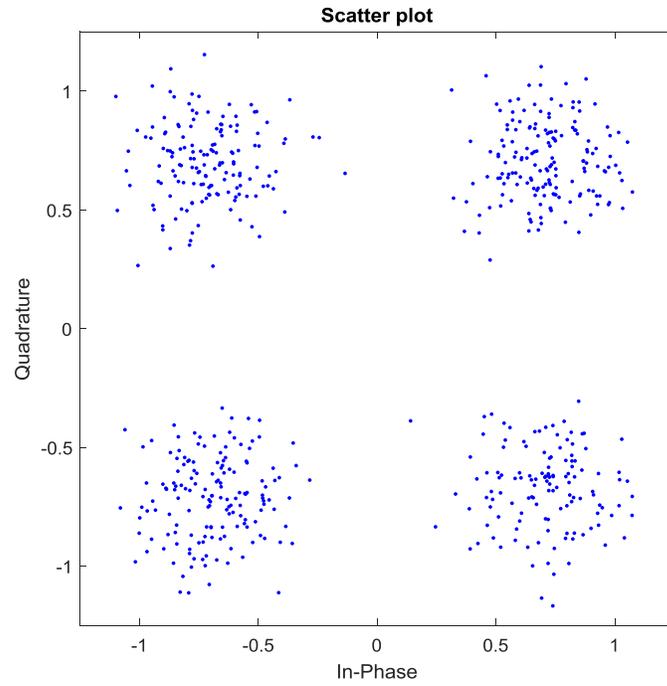


Fig. 29. Constelación en el receptor

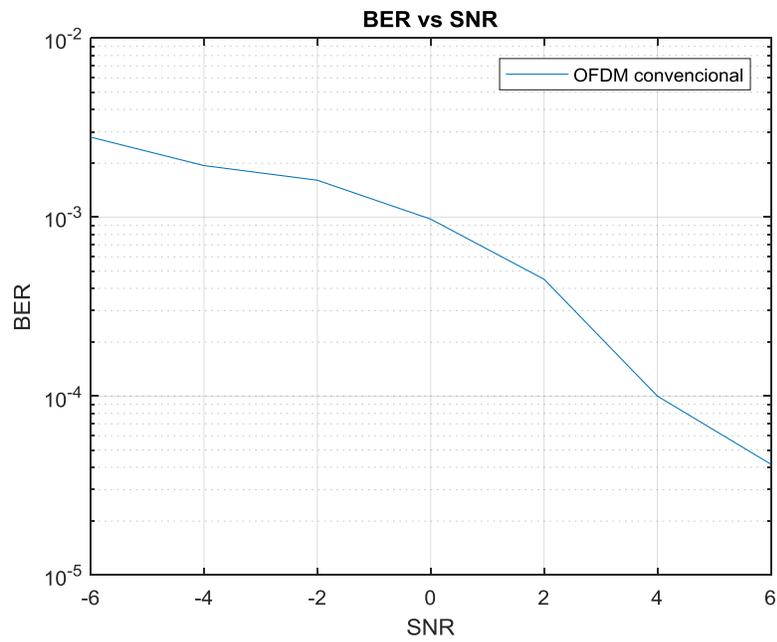


Fig. 30. Grafica VER vs SNR

V.1 Modelado de pulso

OFDM tiene la habilidad de descomponer un canal selectivo en frecuencia en múltiples canales de banda estrecha con un desvanecimiento plano. muchas de las ventajas que nos da OFDM son: tolerancia a la dispersión debido al multitrayecto, inmunidad a los canales selectivos en frecuencia, una alta eficiencia espectral, técnicas de modulación y demodulación eficiente. El desempeño del sistema se puede mejorar al utilizar técnicas de modelado de pulso eficientes [10].

Una vez analizado el comportamiento de nuestro sistema multiportadoras sin aplicar ningún tipo de modelado de pulso, el siguiente paso es diseñar nuestro filtro utilizando las funciones Prolate y observar la respuesta de este filtro. Antes de entrar en detalle con el modelado de pulso debemos entender el porqué de esta técnica. Hay 2 criterios para asegurar la no interferencia.

El primero es que la forma de pulso tenga un cruce por cero al momento de muestreo de todos los intervalos de pulso excepto el suyo, esto para garantizar que el efecto residual de los otros pulsos no introduzca errores. El segundo criterio es que la forma del pulso sea tal que la amplitud decaiga rápidamente fuera del intervalo del pulso. Esto es importante debido a que el sistema se verá afectado por interferencia entre símbolos en caso de que se mueva el pulso a la hora de ser muestreado y no concuerde con el cruce por cero, provocando errores en la decisión. Por eso un decaimiento más rápido fuera del intervalo del pulso es más probable que introduzca menos interferencia entre símbolos. Utilizando los mismos bloques descritos anteriormente, procedemos a crear un filtro que utilice el primer orden de las secuencias Slepian. Primero se genera la secuencia Slepian y luego se crea el filtro. La figura 31 nos muestra el primer orden de las secuencias Slepian con una longitud de 513 muestras [11].

```
%% Generacion de las secuencias Slepian
[e,v]=dpss(Lfiltro,4,1);
e=e./max(e);
figure(1)
plot(e)
e=e.';
%% Creación del filtro
h1=fir1(length(e)-1,0.58,e);
fvtool(h1)
```

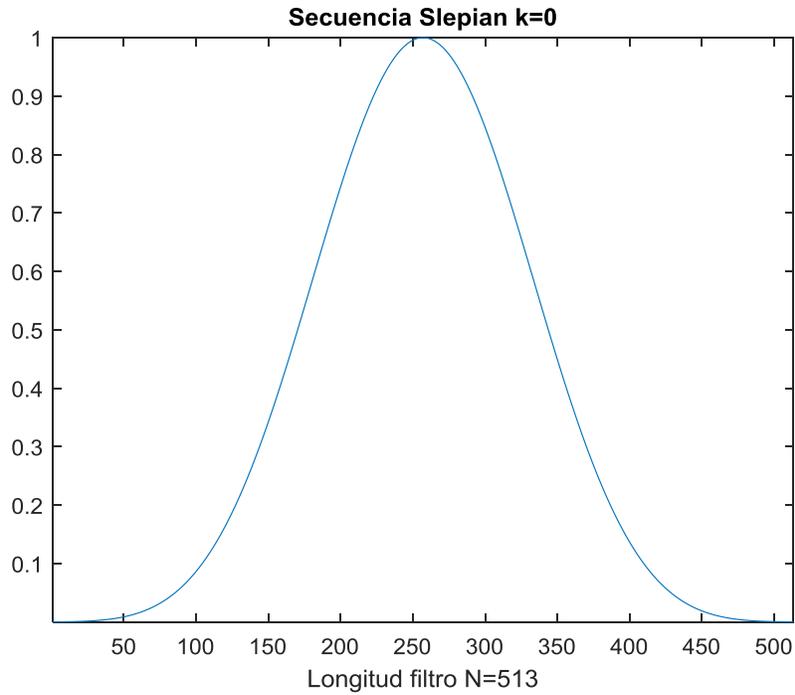


Fig. 31. Primer orden de las secuencias Slepian

La figura 32 nos muestra la respuesta en magnitud del filtro generado, así como también nos indica la respuesta al impulso del mismo filtro. Podemos notar el decaimiento rápido que se produce fuera del intervalo del pulso.

```
TxOFDMcp2=filter(h1,1,[TxOFDMcp; zeros(Lfiltro-1,1)])
```

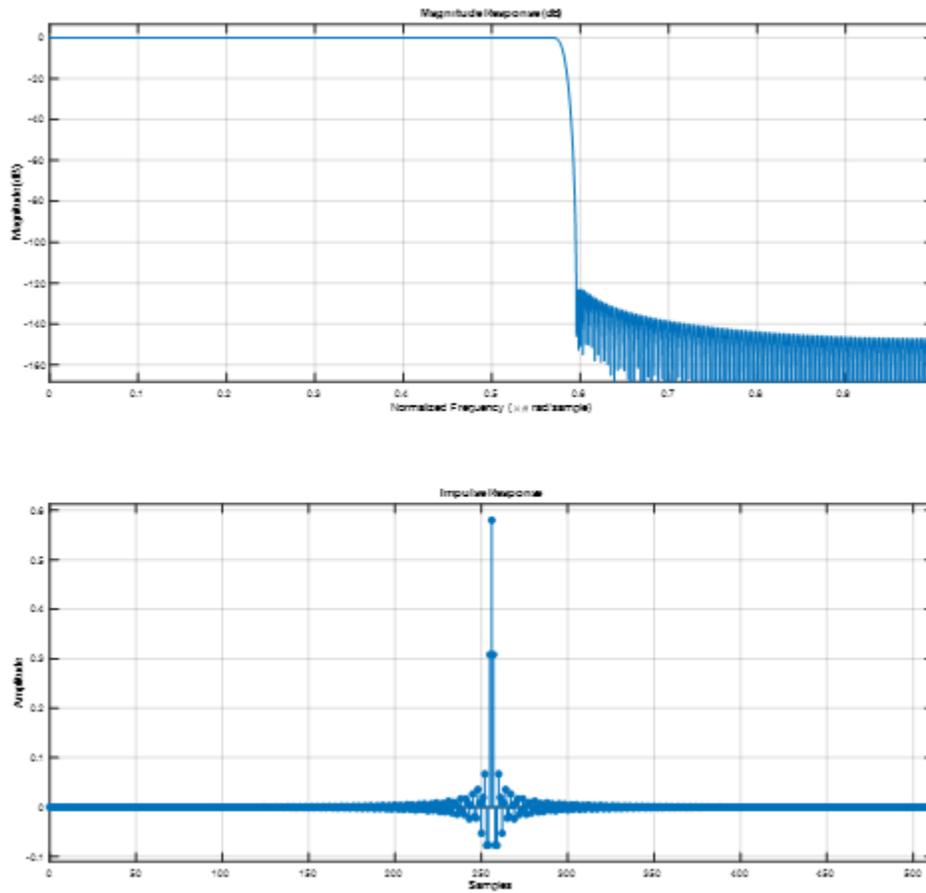


Fig. 32. Respuesta del filtro Slepian

Se procede a aplicar el filtro a la señal OFDM. Podemos observar el espectro de un sistema OFDM convencional y el sistema aplicando el modelado de pulso utilizando las funciones Slepian. A simple vista en la figura 33 se puede observar que, al utilizar el filtro, los lóbulos laterales son mucho menores, esto permite una mayor utilización del espectro, lo que conlleva un incremento en la eficiencia espectral.

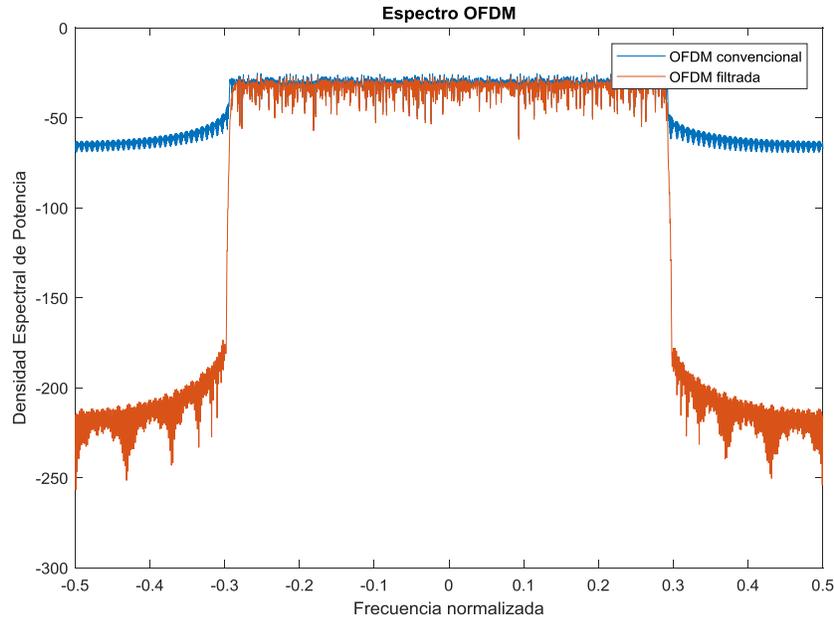


Fig. 33. OFDM convencional vs OFDM filtrada

Al aplicarse un filtro para el modelado de pulso en la transmisión hay que tener en cuenta que debemos implementar un filtro idéntico en la recepción, además de tomar en cuenta el retardo que provoca el filtro y contrarrestar dicho retardo. Luego se analiza como el sistema OFDM convencional. La figura 34 nos muestra las gráficas de BER vs SNR de ambos sistemas. Podemos observar como al utilizar un filtro de modelado de pulso y su filtro adaptado en el receptor se pudo mejorar el rendimiento del sistema, tomando en cuenta que ahora es más sencillo generar las secuencias Slepian, es una buena alternativa a otros tipos de filtros de modelado de pulso utilizados en la actualidad.

```
%% retardo de grupo
retardo=mean(grpdelay(h1));
%% Filtro Receptor
Rx_OFDMF2=filter(h1,1,Rx_OFDM2);
Rx_OFDMFretardo=Rx_OFDMF2(Lfiltro:end);
```

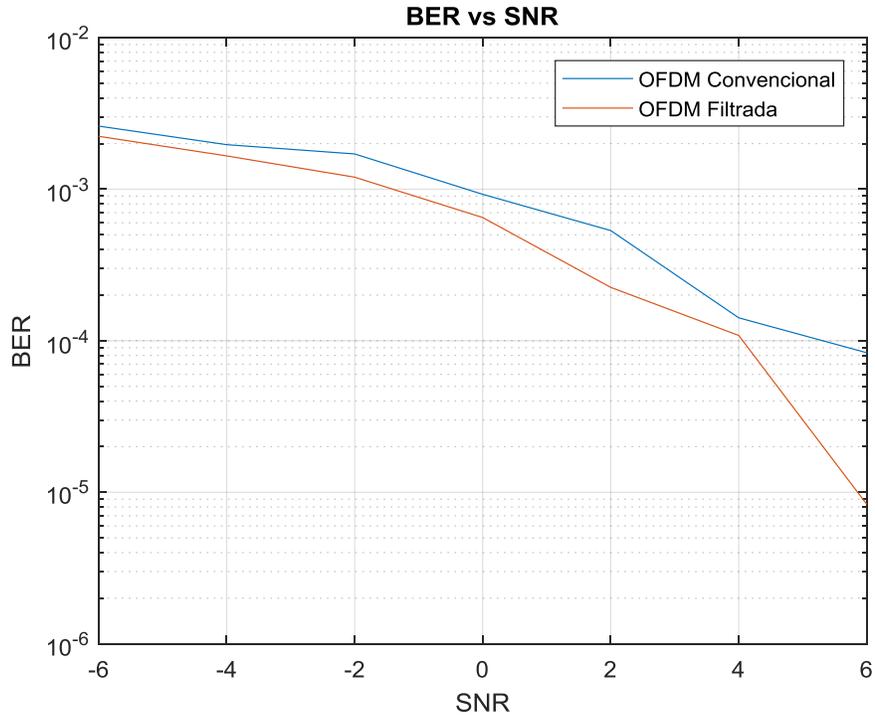


Fig. 34. BER vs OFDM combinada

Podemos observar un comportamiento similar al aumentar el tamaño de la constelación, la figura 35 nos muestra el grafico de BER vs SNR para una modulación 32-QAM y para una modulación 64-QAM. Se puede observar una ligera mejoría en el sistema, sin mucha complicación adicional a la hora de implementación.

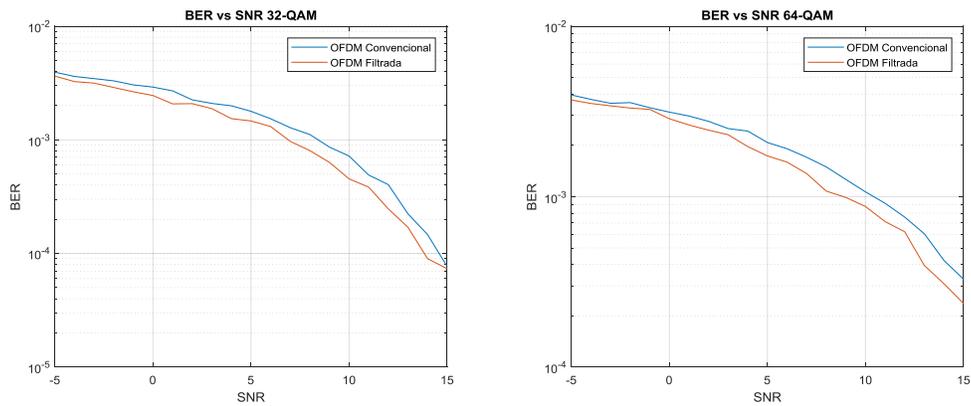


Fig. 35. BER vs SNR para 32-QAM y 64-QAM

VII. CONCLUSIONES

Se realizó el diseño y simulación de un sistema multiportadoras con la particularidad de que se estudió las ventajas y desventajas de utilizar secuencias Slepian para el modelado de pulso. De los espectros obtenidos, se puede observar que al utilizar un filtro implementado con las secuencias Slepian podemos conseguir que el tamaño de los lóbulos laterales se disminuya de gran manera, además existe un decaimiento muy rápido fuera del intervalo del pulso. Esto produjo una mejora en el rendimiento del sistema, lo cual se pudo evidenciar en las gráficas de BER vs SNR obtenidas.

Se pudo estudiar en detalle el comportamiento de las secuencias Slepian y su posible uso en las modulaciones multiportadoras. Los parámetros para tener en cuenta a la hora de generar este tipo de funciones nos determinan que tan concentradas están dentro de una banda determinada. Se determinó que se pueden generar un número fijo de secuencias cuya concentración está cerca de la unidad, el cual está determinado por el producto time-halfbandwidth, para las cuales, al analizar las secuencias generadas a partir de este intervalo, la concentración dentro de la banda del pulso se vuelve muy cercana a cero.

Aunque el fundamento teórico para comprender este tipo de secuencias puede ser algo avanzado, las herramientas como Matlab posee en sus librerías una función para poder generar este tipo de funciones, a futuro y con un mayor estudio en este tema se podrán lograr nuevas aplicaciones para este tipo de funciones

BIBLIOGRAFIAS

- [1] I. C. Moore, M. Cada. Prolate spheroidal wave functions, an introduction to the Slepian series and its properties. *Applied and Computational Harmonics Analysis*. 2004(16):208-230
- [2] E. Guvenkaya, H. Arslan, Inventores; Universidad del sur de Florida, solicitante. Orthogonal Frequency Division Multiplexing (OFDM) transmitter and receiver windowing for interference (ACI) suppression and rejection. US8908796B1. 2014.
- [3] A. Sahin, I. Guvenc, H. Arslan. A Survey on Multicarrier Communications: Prototype Filters, Lattice Structures, and Implementation Aspects. *IEEE COMMUNICATIONS SURVEYS & TUTORIALS*. 2014;16(3):1312-1338
- [4] J. Smith III. Spectral Audio Signal Processing: Spectrum Analysis Windows. [Internet]. Disponible en: https://www.dsprelated.com/freebooks/sasp/Spectrum_Analysis_Windows.html
- [5] Discrete prolate spheroidal (Slepian) sequences - MATLAB dpss- MathWorks España [Internet]. Es.mathworks.com. 2018 [citado 7 Septiembre 2018]. Disponible en: <https://es.mathworks.com/help/signal/ref/dpss.html#br6qo3z-5>
- [6] T. Verma, S. Bilbao, T. H. Y. Meng. The digital Prolate Spheroidal Window. *IEEE International Conference on Acoustics, Speech and Signal Processing*. 1996;1351-1354
- [7] T. Proietti, A. Luati. Low-Pass Filter Design using Locally Weighed Polynomial Regression and Discrete Prolate Spheroidal Sequences. *Munich Personal RePEX Archive*. 2009(6):1-25
- [8] Y. Jitsumatsu, M. Ogata, T. Kohda. A comparison between prolate spheroidal and Gaussian FIR pulse shaping filters. *ICSES International Conference on Signals and Electronic Systems*. 2008(9):14-17
- [9] A. Vahlin, N. Holte. Optimal Finite Duration Pulses of OFDM. *IEEE TRANSACTION ON COMMUNICATION*. 1996;44(1):10-14
- [10] D. Gandhi, S. Gupta, U. Dalal. Implementation of Pulse Shaping Techniques in OFDM System. *International Journal of Computer Applications (0975-8887)*. 2013;68(4):1-5
- [11] Z. Zhao, M. Schellmann, X. Gong, Q. Wang, R. Bohnke, Y. Guo. Pulse shaping design for OFDM systems. *EURASIP Journal on Wireless Communications and Networking*. 2017;74:1-25 DOI 10.1186/s13638-017-0849-8

Anexos

Generación secuencias par e impar

```
close all;
f= [0:0.5/1024:0.5]; t = [0:127];
[E,V] = dpss(128,6,8);

impares=E(:,2:2:end);
pares=E(:,1:2:end);
subplot(2,1,1)
plot(t,pares(:,1))
title('Ordenes pares');ylabel('Amplitud')
hold on; grid on;
plot(t,pares(:,2:4),'--')
axis([0 127 -0.2 0.2]);
legend('k=0','k=2','k=4','k=8')
subplot(2,1,2)
plot(t,impares(:,1))
hold on; grid on;
plot(t,impares(:,2:4),'--')
axis([0 127 -0.2 0.2])
legend('k=1','k=3','k=5','k=7')
title('Ordenes impares');ylabel('Amplitud')

print -dpng ParImpar.png
```

Espectro y DEP

```
fs=127;
sigma=0.1
t=-0.5:1/fs:0.5;
NFFT=1024;

%% Slepian
[e,v]=dpss(128,3.5,4);
Maximo=max(e);
mult=1./Maximo;
e=mult.*e;
BW=3.5;

for i=1:length(v)
subplot(2,2,i)
plot(t,e(:,i))
title(['Pulso Slepian k=', num2str(i)])
xlabel('Tiempo (s)')
ylabel('Amplitud')
ylim([-1 1.2])
end
print -dpng Slepian4ordenes.png

figure(2);
L=length(e);
f=fs*(-NFFT/2:NFFT/2-1)/NFFT;
E=fftshift(fft(e,NFFT));
```

```

subplot(2,2,1)
plot(f,abs(E(:,3))/(L))
title('Magnitud FFT pulso Slepian')
xlabel('Frecuencia (Hz)')
ylabel('Magnitud |X(f)|')
xlim([-5 5])

subplot(2,2,2)
plot(f,abs(E(:,4))/(L))
title('Magnitud FFT pulso Slepian')
xlabel('Frecuencia (Hz)')
ylabel('Magnitud |X(f)|')
xlim([-5 5])

subplot(2,2,3)
plot(f,abs(E(:,1))/(L))
title('Magnitud FFT pulso Slepian')
xlabel('Frecuencia (Hz)')
ylabel('Magnitud |X(f)|')
xlim([-5 5])

subplot(2,2,4)
plot(f,abs(E(:,2))/(L))
title('Magnitud FFT pulso Slepian')
xlabel('Frecuencia (Hz)')
ylabel('Magnitud |X(f)|')
xlim([-5 5])

print -dpng espectromagnitud.png
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
f1=fs*(0:NFFT/2)/NFFT;
figure(3);
E1=fft(e(:,1),NFFT);
E1=E1(1:NFFT/2+1);
E2=fft(e(:,2),NFFT);
E2=E2(1:NFFT/2+1);
E3=fft(e(:,3),NFFT);
E3=E3(1:NFFT/2+1);
E4=fft(e(:,4),NFFT);
E4=E4(1:NFFT/2+1);
Pxx1=E1.*conj(E1)/(L*L);
Pxx2=E2.*conj(E2)/(L*L);
Pxx3=E3.*conj(E3)/(L*L);
Pxx4=E4.*conj(E4)/(L*L);
subplot(2,1,1)
plot(f1,10*log10(Pxx1))
title('DEP Secuencia Slepian k=0')
xlabel('Frecuencia (Hz)')
ylabel('DEP - P_{xx} dB/Hz')
ylim([-200 20])
xlim([0 50])
grid on
hold on
plot([BW BW],[-200 20])
subplot(2,1,2)
plot(f1,10*log10(Pxx2))

```

```

title('DEP Secuencia Slepian k=1')
xlabel('Frecuencia (Hz)')
ylabel('DEP - P_{xx} dB/Hz')
ylim([-200 20])
xlim([0 50])
grid on
hold on
plot([BW BW],[-200 20])
print -dpng DEP4ordenesAnex1.png

```

```

figure()
subplot(2,1,1)
plot(f1,10*log10(Pxx3))
title('DEP Secuencia Slepian k=2')
xlabel('Frecuencia (Hz)')
ylabel('DEP - P_{xx} dB/Hz')
ylim([-200 20])
xlim([0 50])
grid on
hold on
plot([BW BW],[-200 20])
subplot(2,1,2)
plot(f1,10*log10(Pxx4))
title('DEP Secuencia Slepian k=3')
xlabel('Frecuencia (Hz)')
ylabel('DEP - P_{xx} dB/Hz')
ylim([-200 20])
xlim([0 50])
grid on
hold on
plot([BW BW],[-200 20])

print -dpng DEP4ordenesAnex2.png

```

Características secuencias Slepian

```

fs1=255;
fs2=127;
NFFT=1024;
f1=fs1*(0:NFFT/2)/NFFT;
f2=fs2*(0:NFFT/2)/NFFT;
sigma=0.1
t1=-0.5:1/fs1:0.5;
t2=-0.5:1/fs2:0.5;

```

```

%% Slepian
NW1=4;
NW2=4;
[e1,v1]=dpss(256,NW1,1);
[e2,v2]=dpss(128,NW2,1);
Maximo1=max(e1);
Maximo2=max(e2);
mult1=1./Maximo1;
mult2=1./Maximo2;
e1=mult1.*e1;

```

```

e2=mult2.*e2;

L1=length(e1);
L2=length(e2);
E1=fft(e1,NFFT);
E1=E1(1:NFFT/2+1);
E2=fft(e2,NFFT);
E2=E2(1:NFFT/2+1);
Pxx1=E1.*conj(E1)/(L1*L1);
Pxx2=E2.*conj(E2)/(L2*L2);

figure(1)
subplot(2,1,1)
plot(t1,e1,'k.')
ylim([0 1.2])
title(['DPSS para k=0, N=128 y NW=',num2str(NW1)])
grid on
subplot(2,1,2)
ylim([0 1.2])
plot(f1,10*log10(Pxx1),'k')
ylim([-150 10])
xlim([0 20])
grid on
hold on
plot([NW1 NW1],[-200 20],'r')

figure(2)
subplot(2,1,1)
plot(t2,e2,'k.')
title(['DPSS para k=0, N=128 y NW=',num2str(NW2)])
grid on
ylim([0 1.2])
subplot(2,1,2)
plot(f2,10*log10(Pxx2),'k')
ylim([-150 20])
xlim([0 20])
grid on
hold on
plot([NW2 NW2],[-200 20],'r')

```

Concentración pulsos Slepian

```

close all;
clear all;
fs=127;
sigma=0.1
t=-0.5:1/fs:0.5;
NFFT=1024;
f=fs*(-NFFT/2:NFFT/2-1)/NFFT;
NW1=2.5;
NW2=4;
[e1,v1]=dpss(128,NW1,10);
[e2,v2]=dpss(128,NW2,10);
Maximo1=max(e1);
mult1=1./Maximo1;
e1=mult1.*e1;
Maximo2=max(e2);
mult2=1./Maximo2;

```

```

e2=mult2.*e2;

h1 = figure(1); set(h1,'Position',[100 100 1100
650],'PaperPositionMode','auto');

for n=1:length(v1)
    if n<5
        subplot(length(v1)/2,2,n)      %Graficar cada orden de las secuencias
        plot(t,e1(:,n))
        title(['Pulso Slepian k=', num2str(n-1)])
        xlabel('Tiempo (s)')
        ylabel('Amplitud')
        grid on
    else
        subplot(length(v1)/2,2,n)
        plot(t,e1(:,n))
        title(['Pulso Slepian k=', num2str(n-1)])
        xlabel('Tiempo (s)')
        ylabel('Amplitud')
        grid on

    end
    hold on;

end%% Graficar para NW=2.5
for n=1:length(v2)
    if n<5
        subplot(length(v2)/2,2,n)      %Graficar cada orden de las secuencias
        plot(t,e2(:,n))
        title(['Pulso Slepian k=', num2str(n-1)])
        xlabel('Tiempo (s)')
        ylabel('Amplitud')
        grid on
    else
        subplot(length(v2)/2,2,n)
        plot(t,e2(:,n))
        title(['Pulso Slepian k=', num2str(n-1)])
        xlabel('Tiempo (s)')
        ylabel('Amplitud')
        grid on
    end
    legend({'NW=2.5','NW=4'},'Position',[0.92 0.95 0.01 0.02])

    hold on;

end%% Graficar para NW=4

print -dpng Slepian10Anex.png

h2 = figure(2); set(h2,'Position',[100 100 1100
650],'PaperPositionMode','auto');
stem(1:length(v1),v1)
hold on
plot(1:length(v1),0.90*ones(length(v1),1))

```

```

title(['Ratios de concentracion en [-',num2str(NW1),',',num2str(NW1),']
de las k secuencias Slepian'])
xlabel('Orden')
ylabel('Concentracion')
legend('Ordenes', '> 90%')
grid on
print -dpng Concentracion2.5Anex.png

h3 = figure(3); set(h3,'Position',[100 100 1100
650],'PaperPositionMode','auto');
stem(1:length(v2),v2)
hold on
plot(1:length(v2),0.90*ones(length(v2),1))
title(['Ratios de concentracion en [-',num2str(NW2),',',num2str(NW2),']
de las k secuencias Slepian'])
xlabel('Orden')
ylabel('Concentracion')
legend('Ordenes', '> 90%')
grid on
print -dpng Concentracion4Anex.png
%% Magnitud de la FFT
L2=length(e1);

h4 = figure(4); set(h4,'Position',[100 100 1100
650],'PaperPositionMode','auto');

for n=1:6
F(:,n)=fftshift(fft(e1(:,n),NFFT)); % Magnitud de la FFT
Pxx1(:,n)=F(:,n).*conj(F(:,n))/(NFFT*NFFT); % DEP
subplot(5,2,n)
plot(f,abs(F(:,n))/(L2))
xlim([-10 10])
xlabel('Frecuencia (Hz)')
title(['k= ', num2str(n-1)])
ylabel('Magnitud |X(f)|')
hold on
F(:,n)=fftshift(fft(e2(:,n),NFFT)); % Magnitud de la FFT
Pxx2(:,n)=F(:,n).*conj(F(:,n))/(NFFT*NFFT); % DEP
subplot(5,2,n)
plot(f,abs(F(:,n))/(L2))
xlim([-10 10])
xlabel('Frecuencia (Hz)')
title(['k= ', num2str(n-1)])
ylabel('Magnitud |X(f)|')
end
hold off

for n=6:length(v1)
F(:,n)=fftshift(fft(e1(:,n),NFFT));
Pxx1(:,n)=F(:,n).*conj(F(:,n))/(NFFT*NFFT);
subplot(5,2,n)
plot(f,abs(F(:,n))/(L2))
xlim([-10 10])
xlabel('Frecuencia (Hz)')
title(['k= ', num2str(n-1)])
ylabel('Magnitud |X(f)|')

```

```

legend({'NW=2.5','NW=4'},'Position',[0.92 0.95 0.01 0.02])
hold on
F(:,n)=fftshift(fft(e2(:,n),NFFT));
Pxx2(:,n)=F(:,n).*conj(F(:,n))/(NFFT*NFFT);
subplot(5,2,n)
plot(f,abs(F(:,n))/(L2))
xlim([-10 10])
xlabel('Frecuencia (Hz)')
title(['k= ', num2str(n-1)])
ylabel('Magnitud |X(f)|')
legend({'NW=2.5','NW=4'},'Position',[0.92 0.95 0.01 0.02])
end

print -dpng MagFFFTAnex.png

```

OFDM

```

close all;
clear all;
%% Comparacion OFDM filtrada y OFDM convencional
NFFT=1024;           %Tamano de la FFT
nRB=50;              %Numero de bloques de recursos
nSub=12;             %Numero de subportadoras por RB
cp=72;               %longitud del prefijo ciclico

M=4;                 %Tamano de la constelacion
k=log2(M);           %Numero de bits por simbolo
SNRdb=-6:2:6;        %SNR
s=rng(111);
BER_tot=zeros(1,length(SNRdb));
BER_tot2=zeros(1,length(SNRdb));
Lfiltro=513;         %Longitud del filtro

%% Filtro
nSubDatos=nRB*nSub;

%% Generacion de las secuencias Slepian
[e,v]=dpss(Lfiltro,4,1);
e=e./max(e);
figure(1)
plot(e)
axis tight
title('Secuencia Slepian k=0');
xlabel('Longitud filtro N=513')
e=e.';
%% Creacion del filtro
h1=fir1(length(e)-1,0.58,e);
h2=fir1(length(e)-1,0.58,e);
fvtool(h1)

%% Modulador QAM
qamMapper = comm.RectangularQAMModulator( ...
    'ModulationOrder', M, 'BitInput', true, ...
    'NormalizationMethod', 'Average power');

```

```

for i=1:length(SNRdb); %5 valores de SNR
    errores_acumulados=0;
    errores_acumulados2=0;
    for num=1:1
        %% Generacion de bits
        bits=randi([0 1],nSubDatos*k,1);
        bitsMod=qamMapper(bits);
        % figure(2)
        % subplot(2,1,1)
        % stem(bits(1:20))
        % title('Bits aleatorios')
        % subplot(2,1,2)
        % stem(bitsMod(1:20))
        % title('Bits modulados')

        %% llenar con ceros
        SimbolosOFDM=[zeros((NFFT-nSubDatos)/2,1);bitsMod;zeros((NFFT-
nSubDatos)/2,1)];

        %% IFFT
        Ifftout=ifft(iffshift(SimbolosOFDM));

        %% Agregar prefijo ciclico
        TxOFDMcp=[Ifftout(NFFT-cp+1:NFFT); Ifftout];
        TxOFDMcp2=filter(h1,1,[TxOFDMcp; zeros(Lfiltro-1,1)]);
        figure(4)
        if i==length(SNRdb)
            % %%Graficar el espectro
            [dep,f]=periodogram(TxOFDMcp,rectwin(length(TxOFDMcp)),
NFFT*2,1,'centered');
            plot(f,10*log10(dep));
            title('Espectro OFDM')
            ylabel('Densidad Espectral de Potencia')
            xlabel('Frecuencia normalizada')
            hold on
            [dep2,f2]=periodogram(TxOFDMcp2,rectwin(length(TxOFDMcp2)),
NFFT*2,1,'centered');
            plot(f2,10*log10(dep2));
            title('Espectro OFDM')
            ylabel('Densidad Espectral de Potencia')
            xlabel('Frecuencia normalizada')
            legend('OFDM convencional','OFDM filtrada')
        end

        %% Canal AWGN
        Rx_OFDM1= awgn(TxOFDMcp, SNRdb(i), 'measured');
        Rx_OFDM2 = awgn(TxOFDMcp2, SNRdb(i), 'measured');

        % if i==length(SNRdb)
        % [dep1,f1]=periodogram(Rx_OFDM2,rectwin(length(Rx_OFDM2)),
NFFT*2,1,'centered');
        % plot(f,10*log10(dep));
    end
end

```

```

%
% title('Espectro OFDM en el Rx')
% ylabel('Densidad Espectral de Potencia')
% xlabel('Frecuencia normalizada')
% end
% legend('1','2')
% RxFilt=filter(h2,1,Rx_OFDM)
%% retardo de grupo
% figure()
% grpdelay(h1)
retardo=mean(grpdelay(h1));

%% Filtro Receptor

Rx_OFDMF2=filter(h1,1,Rx_OFDM2);
Rx_OFDMFretardo=Rx_OFDMF2(Lfiltro:end);

%% Eliminar prefijo ciclico
Rx_cp=Rx_OFDM1(cp+1:end);
Rx_cp2=Rx_OFDMFretardo(cp+1:end);

%Realizar la FFT
Rx_simbolos=fftshift(fft(Rx_cp));
Rx_simbolos2=fftshift(fft(Rx_cp2));

%Seleccionar subportadoras de datos
Rx_datos=Rx_simbolos(213:NFFT-212);
Rx_datos2=Rx_simbolos2(213:NFFT-212);
% if i==length(SNRdb)
% scatterplot(Rx_datos)
% grid on
% scatterplot(Rx_datos2)
% grid on
% end
%% Demodulador
qamDemod = comm.RectangularQAMDemodulator('ModulationOrder', ...
    M, 'BitOutput', true, ...
    'NormalizationMethod', 'Average power');

% Decisor y calculo de errores
Bits_rx = qamDemod(Rx_datos);
Bits_rx2 = qamDemod(Rx_datos2);

[errores,BER]=biterr(bits,Bits_rx);
errores_acumulados=errores_acumulados+errores;

[errores2,BER2]=biterr(bits,Bits_rx2);
errores_acumulados2=errores_acumulados2+errores2;
end
BER_tot(i)=errores/(100*nSubDatos*k);
BER_tot2(i)=errores2/(100*nSubDatos*k);

```

```
end
```

```
figure  
semilogy( SNRdb, BER_tot) %Grafica en escala logaritmica del BER que se  
obtuvo  
grid on  
ylabel('BER')  
xlabel('SNR')  
title('BER vs SNR 32-QAM')  
hold on  
semilogy( SNRdb, BER_tot2)  
legend('OFDM Convencional', 'OFDM Filtrada')
```