

**EL ALGORITMO DEL MURCIÉLAGO VIRTUAL (BAT ALGORITHM) COMO
ESTRATEGIA PARA EL DISEÑO ÓPTIMO DE FILTROS PASA-BAJAS**

**HÉCTOR CASTRO RUEDA
MARCO OTERO CORZO**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES
BUCARAMANGA**

2014

**EL ALGORITMO DEL MURCIÉLAGO VIRTUAL (BAT ALGORITHM) COMO
ESTRATEGIA PARA EL DISEÑO ÓPTIMO DE FILTROS PASA-BAJAS**

**HÉCTOR FERNANDO CASTRO RUEDA
MARCO STEFANO OTERO CORZO**

**Trabajo de grado para optar al título de
Ingeniero electrónico**

**CARLOS RODRIGO CORREA CELY, Ph.D.
DIRECTOR TRABAJO DE GRADO**

**IVÁN AMAYA, Ph.D.(c)
CODIRECTOR DEL TRABAJO DE GRADO**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES
BUCARAMANGA**

2014

DEDICATORIA

A mis padres, para quienes nada es imposible; a mi hermano Miguel, por su ejemplo; a todos aquellos que con una palabra, un gesto o una oración me animaron para cumplir este sueño, en especial a mi tía Mabel por darme la oportunidad de enfrentarme a nuevos retos y a mi tío Hugo por sus consejos; a Karen por los buenos tiempos.

Héctor Fernando Castro Rueda

A mis padres por brindarme su apoyo incondicional, los mejores consejos y la fuerza para alcanzar esta meta; a Jessica por estar a mi lado durante estos años y a todos aquellos que de alguna manera contribuyeron e hicieron parte en la realización de este logro.

Marco Stefano Otero Corzo

AGRADECIMIENTOS

Especial agradecimiento a los profesores Rodrigo Correa e Iván Amaya, por sus valiosas recomendaciones y sugerencias. Gracias a todo el grupo de investigación CEMOS por los espacios prestados para el desarrollo de este proyecto de grado. Adicionalmente al profesor Carlos Niño por sus apreciaciones.

Los Autores

TABLA DE CONTENIDO

1. INTRODUCCIÓN	16
2. DESCRIPCIÓN DEL TRABAJO DE GRADO	18
2.1. PLANTEAMIENTO DEL PROBLEMA	18
2.2. OBJETIVOS	19
2.2.1. OBJETIVO GENERAL	19
2.2.2. OBJETIVOS ESPECIFICOS	19
3. MARCO TEÓRICO	20
3.1. FILTROS ANALÓGICOS	20
3.2. FILTROS DE PRIMER ORDEN Y SEGUNDO ORDEN	21
3.3. FILTROS CLASICOS	23
3.4. APROXIMACIÓN COMPUTACIONAL AL DISEÑO DE FILTROS	23
3.5. ALGORITMO DEL MURCIELAGO VIRTUAL	26
4. IMPLEMENTACIÓN DEL ALGORITMO DEL MURCIELAGO VIRTUAL	30
4.1. CRITERIO DE PARADA	31
4.2. VALIDACIÓN DEL ALGORITMO	32
4.2.1. FUNCIÓN DE JONG	32
4.2.2. FUNCIÓN RASTRIGIN	37
4.2.3. FUNCIÓN MICHALEWICZ	40
4.2.4. FUNCIÓN SHUBERT	43
4.2.5. FUNCIÓN ROSENBROCK	46
4.2.6. COMPARACIÓN CON RESULTADOS ENCONTRADOS EN LA LITERATURA.	48
5. DESARROLLO DE FILTROS CON RESTRICCIONES EN TIEMPO Y FRECUENCIA BASADO EN EL ALGORITMO DEL MURCIELAGO VIRTUAL	51
5.1. FUNCIÓN OBJETIVO	51

5.2.	EL ESQUEMA DE TOLERANCIA-----	51
5.3.	FUNCIÓN DE PENALIZACIÓN-----	53
5.4.	MEDIDAS DE DESEMPEÑO-----	54
	5.4.1. RETARDO DE GRUPO Y RETRASO DE FASE-----	54
	5.4.2. SOBREPASO MÁXIMO-----	54
6.	SELECCIÓN DE PARAMETROS OPTIMOS PARA EL ALGORITMO DEL MURCIELAGO VIRTUAL-----	56
6.1.	EFECTO DE LOS PESOS EN LA FUNCIÓN OBJETIVO-----	57
7.	RESULTADOS-----	61
7.1.	COMPARACIÓN CON EL DISEÑO DE UN FILTRO BUTTERWORTH.	61
7.2.	DISEÑO DE FILTROS DE SÉPTIMO Y DÉCIMO ORDEN CON RESTRICCIONES EN FRECUENCIA.-----	63
7.3.	COMPARACIÓN FILTRO BUTTERWORTH SOBREPICO MÍNIMO----	66
	7.3.1. FILTROS DE SEGUNDO ORDEN-----	66
	7.3.2. FILTROS DE CUARTO ORDEN-----	68
	7.3.3. FILTROS DE OCTAVO ORDEN-----	70
7.4.	FILTRO CON FASE MÁXIMAMENTE LINEAL.-----	72
7.5.	TRANSFORMACIÓN A FILTRO PASABANDA-----	74
8.	CONCLUSIONES-----	78
9.	RECOMENDACIONES PARA TRABAJOS FUTUROS-----	80
10.	REFERENCIAS BIBLIOGRÁFICAS-----	82
11.	BIBLIOGRAFÍA-----	83
12.	ANEXOS-----	84

LISTA DE FIGURAS

Figura 1. Filtro pasabajas	21
Figura 2. Filtro de primer orden con frecuencia de corte unitaria.....	22
Figura 3. Comparación gráfica de los tipos clásicos de filtros	23
Figura 4. Esquema de tolerancia	25
Figura 5. Diagrama de bloques BAT ALGORTIHM, primera parte.	27
Figura 6. Diagrama de bloques BAT ALGORTIHM, segunda parte.....	28
Figura 7. Diagrama de bloques BAT ALGORTIHM, tercera parte.	29
Figura 8. Función De Jong con $d=2$	34
Figura 9. Convergencia de los murciélagos $d=2$ para $\alpha=0.8$ y $Y=0.9$	34
Figura 10. Convergencia para De Jong para $d=3$, $\alpha=0.8$ y $Y=0.9$	36
Figura 11. Función Rastrigin $d=2$, $\alpha=0.8$ y $Y=0.9$	38
Figura 12. Convergencia para Rastrigin $d=2$ con $\alpha=0.8$ y $Y=0.9$	39
Figura 13. Aumento a la zona de convergencia.....	40
Figura 14. Función Michalewicz.....	41
Figura 15. Primeras Veinte iteraciones para Michalewicz.....	42
Figura 16. Función de Shubert.....	44
Figura 17. Convergencia para Shubert en dos simulaciones (rojo y azul) con $\alpha=0.8$ y $Y=0.9$	45
Figura 18. Función Rosenbrock.....	47
Figura 19. Convergencia para Rosenbrock $d=2$	47
Figura 20. Esquema de tolerancia muestreado con diferentes pasos	52
Figura 21. Efecto de los pesos en la optimización	59
Figura 22. Comparativa de la magnitud del filtro Butterworth con el filtro desarrollado a partir del algoritmo.....	62
Figura 23. Respuesta en frecuencia de la magnitud para filtros de séptimo orden del ejemplo 7.2.....	64
Figura 24. Detalle A de la Figura 23	64
Figura 25. Respuesta en frecuencia de la magnitud para filtros de décimo orden del ejemplo 7.2.....	65
Figura 26. Detalle A de la Figura 25.	65
Figura 27. Respuesta en frecuencia para los filtros del ejemplo 7.3.1	67
Figura 28. Respuesta al escalón para los filtros del ejemplo 7.3.1	68
Figura 29. Respuesta en magnitud para filtros del ejemplo 7.3.2	69
Figura 30. Respuesta al escalón para filtros del ejemplo 7.3.2.....	70
Figura 31. Respuesta en magnitud para filtros del ejemplo 7.3.3	71
Figura 32. Respuesta al escalón para filtros del ejemplo 7.3.3.....	71
Figura 33. Respuesta en magnitud para el ejemplo 4.....	73
Figura 34. Respuesta en fase para los filtros del ejemplo 4	73
Figura 35. Retardo de grupo para los filtros del ejemplo 4.	74

Figura 36. Respuesta en magnitud para la transformación del filtro del ejemplo 7.1.	75
Figura 37. Respuesta en magnitud para la transformación del filtro del ejemplo 7.3.3.....	76
Figura 38. Respuesta en fase para la transformación del filtro del ejemplo 7.3.3..	76
Figura 39. Respuesta al escalón para para la transformación del filtro del ejemplo 7.3.3.....	77

LISTA DE TABLAS

Tabla 1. Terminología del algoritmo.....	30
Tabla 2. Parámetros utilizados para la función De Jong con $\alpha=0.8$ y $\gamma=0.9$	33
Tabla 3. Resultados para De Jong $d=2$ $\alpha=0.8$ y $\gamma=0.9$	35
Tabla 4. Resultados para De Jong $d=3$	36
Tabla 5. Resultados para De Jong $d=256$	37
Tabla 6. Resultados para Rastrigin $d=2$ y $d=3$	40
Tabla 7. Resultados para Michalewicz $d=2$ y $d=5$	42
Tabla 8. Resultados para Shubert	44
Tabla 9. Resultados para Rosenbrock.....	46
Tabla 10. Resultados para Rosenbrock $d=16$	48
Tabla 11. Parámetros de la prueba de comparación BA.	49
Tabla 12. Resultados del algoritmo original y el algoritmo propuesto.	49
Tabla 13. Valores de α y γ para las simulaciones.....	56
Tabla 14. Especificaciones de diseño.....	57
Tabla 15. Parámetros del algoritmo	58
Tabla 16. Pesos para los filtros de la figura 18.	59
Tabla 17. Variables informativas para calcular los pesos	60
Tabla 18. Variables informativas de los errores para diferentes pesos de la ecuación	60
Tabla 19. Resultados de la validación del algoritmo con un filtro Butterworth como referencia.....	62
Tabla 20. Especificaciones para filtros de orden superior.....	63
Tabla 21. Especificaciones de diseño para el ejemplo del numeral 7.3.1 y 7.3.2 ..	66
Tabla 22. Polos resultantes para los filtros del ejemplo 7.3.1	66
Tabla 23. Polos obtenidos para los filtros del ejemplo 7.3.2	68
Tabla 24. Coeficientes del denominador para filtros del ejemplo 7.3.3.....	71
Tabla 25. Polos para los filtros del ejemplo 4	72
Tabla 26. Combinaciones para $\alpha=0.6$	84
Tabla 27. Combinaciones para $\alpha=0.8$	86
Tabla 28. Combinaciones para $\alpha=0.9$	88

LISTA DE ANEXOS

ANEXO A: Presentación de resultados de las combinaciones para la selección de parámetros óptimos del algoritmo. -----	84
ANEXO B: Características de los filtros clásicos. -----	90

RESUMEN

TÍTULO: EL ALGORITMO “DEL MURCIÉLAGO” VIRTUAL (BAT ALGORITHM) COMO ESTRATEGIA PARA EL DISEÑO OPTIMO DE FILTROS PASA-BAJAS.¹

AUTORES: Héctor Fernando Castro Rueda y Marco Stefano Otero Corzo.²

PALABRAS CLAVE: BAT ALGORITHM, algoritmo, optimización, estocástico, función objetivo, filtros analógicos.

CONTENIDO:

En el presente trabajo de investigación se realizó el diseño y desarrollo de filtros analógicos pasa-bajas con restricciones en tiempo y frecuencia, basado en la utilización del algoritmo del murciélago virtual BA, (BAT ALGORITHM), como método de optimización. Obteniendo como resultado un filtro que cumpla con las especificaciones dadas por el diseñador, y encontrando la función de transferencia que mejor se acople a las exigencias.

Inicialmente el algoritmo del murciélago BA fue utilizado en diferentes funciones de prueba, con el fin de verificar su efectividad en tiempo, precisión y número de iteraciones; ajustando adecuadamente los parámetros estocásticos de manera que el algoritmo converja y se restrinja el espacio de búsqueda al dominio de la función objetivo. Los resultados arrojados se contrastaron posteriormente con los publicados al hacer uso de otros algoritmos de optimización ya conocidos. Por último, se muestra el producto obtenido al aplicar el BAT ALGORITHM (BA) en el diseño de filtros analógicos pasa-bajas, bajo un esquema de tolerancia dado por especificaciones otorgadas por el usuario, la selección empírica de los parámetros óptimos para tener un mejor funcionamiento en esta aplicación, el empleo de restricciones de tiempo y frecuencia (sobrepico máximo y retardo de grupo), y diferentes comparaciones gráficas de los resultados arrojados respecto a otros métodos clásicos de diseño de filtros.

¹ Trabajo de grado

² Facultad: Fisicomecánicas Escuela: Ingeniería eléctrica, electrónica y de telecomunicaciones Director: Carlos Rodrigo Correa Cely Codirector: Iván Mauricio Amaya.

ABSTRACT

TITLE: THE BAT ALGORITHM AS STRATEGIC FOR THE OPTIMUM DESIGN OF LOW-PASS FILTERS.³

AUTHORS: Hector Fernando Castro Rueda and Marco Stefano Otero Corzo⁴

KEY WORDS: bat algorithm, algorithm, optimization, stochastic, objective function, analog filters.

CONTENT:

In this research work, the design and development of analog low-pass filters with restrictions in time and frequency was conducted based on the utilization of the “bat algorithm” as a method of optimization. As a result, a filter was obtained with the specifications given by the designer. In addition, an improved transference function that best matches the requirements was also found.

Initially, the bat algorithm (BA) was utilized in different test functions in order to verify its effectiveness in time, precision, and number of iterations. Adjusting adequately all the stochastic parameters made the algorithm to converge, and the search space was restricted to the objective function domain. The obtained results were later compared with the published results which were obtained with the use of different optimization algorithms that are already validated. Lastly, it shows the product obtained by applying the algorithm in the design of analog low-pass, under a scheme of tolerance specifications given by the user ratings, the empirical selection of optimum parameters in order to obtain the best use of this application, the utilization of time and frequency constraints (maximum overshoot and group delay) and different graphic comparisons with respect to other classic methods of filter design.

³ Bachelor thesis

⁴ Faculty: Physicomechanics. School: Electric, Electronic and the Telecommunications Engineering. Advisor: Carlos Rodrigo Correa Cely. Coadvisor: Iván Mauricio Amaya.

1. INTRODUCCIÓN

La observación del comportamiento de los sistemas biológicos y de los sistemas físicos en la naturaleza ha permitido el desarrollo de diferentes algoritmos de optimización, que replican fenómenos que van desde el recocido de metales (algoritmo *Simulated Annealing*), hasta el comportamiento de especies animales (algoritmo de colonia de hormigas, de luciérnagas y de optimización por enjambre de partículas, entre otros). La mayor parte de estos algoritmos, y en particular todos los mencionados anteriormente, implementan procesos estocásticos, y buscan explotar el espacio de búsqueda de forma global y local, para llegar a una solución final óptima. Este trabajo de grado se enfoca en el uso del algoritmo del murciélago (Bat Algorithm, BA), publicado por Xin-She Yang en el 2010 y que se basa en la ecolocación de los murciélagos.

El diseño de filtros analógicos es una de las tantas aplicaciones que tiene la ingeniería electrónica. Éstos se encuentran en múltiples dispositivos y pueden considerarse como dispositivos selectivos o diferenciadores de frecuencia. Su diseño tradicionalmente se basa en el ajuste de las bandas de paso y de supresión a través de polinomios y funciones racionales. Sin embargo, en este trabajo de grado se aborda desde la optimización de la función de transferencia de un filtro pasa-bajas, pues a partir del mismo se puede obtener cualquier diseño, buscando cumplir con restricciones de tiempo y frecuencia.

El presente documento se encuentra organizado de la siguiente manera: En el capítulo 1 se presenta la introducción al tema de investigación. En el capítulo 2 se presenta la descripción del trabajo de grado. En la sección 3 el marco teórico de los temas relevantes del trabajo. En la sección 4 se encuentra la implementación el algoritmo del murciélago virtual, la validación del mismo para varias funciones de prueba y sus respectivas comparaciones. En la sección 5 se muestra el desarrollo de filtros con restricciones de tiempo y frecuencia basado en el algoritmo del murciélago virtual. En el capítulo 6, se encuentra la selección de

parámetros óptimos para el algoritmo. En la sección 7 se presentan los resultados obtenidos para varios ejemplos de filtros. Finalmente, los capítulos 8, 9, 10 y 11 contienen las conclusiones, recomendaciones para trabajos futuros, referencias bibliográficas y los anexos, respectivamente.

2. DESCRIPCIÓN DEL TRABAJO DE GRADO⁵

2.1. PLANTEAMIENTO DEL PROBLEMA

La teoría clásica del diseño de filtros presenta varias técnicas tradicionales que poseen una eficiencia bastante aceptable [1], a tal punto, que algunos software como MATLAB® las han incorporado en sus paquetes de herramientas como instrucciones o sentencias, para así, facilitar al usuario un entorno de desarrollo y diseño de filtros mucho más agradable, logrando ahorrar tiempo en la implementación de los mismos.

Sin embargo, estos métodos limitan al diseñador a atenerse a utilizar ciertas características predeterminadas, pero al hacer uso de algoritmos metaheurísticos, se hace posible realizar una aproximación computacional para un diseño óptimo. El problema a abordar a lo largo de este trabajo de grado consiste en cómo implementar el diseño de filtros analógicos mediante la utilización del algoritmo del murciélago virtual; otorgando al usuario la libertad de proporcionar sus propias especificaciones de diseño, encontrando la función de transferencia que mejor cumpla las exigencias propuestas.

⁵ Corresponde a una contribución intelectual consensuada entre los estudiantes Héctor Castro y Marco Otero, con el profesor Rodrigo Correa y el ingeniero Iván Amaya.

2.2. OBJETIVOS

2.2.1. OBJETIVO GENERAL

Diseñar un filtro óptimo analógico con restricciones en tiempo y frecuencia mediante el algoritmo del murciélago virtual.

2.2.2. OBJETIVOS ESPECIFICOS

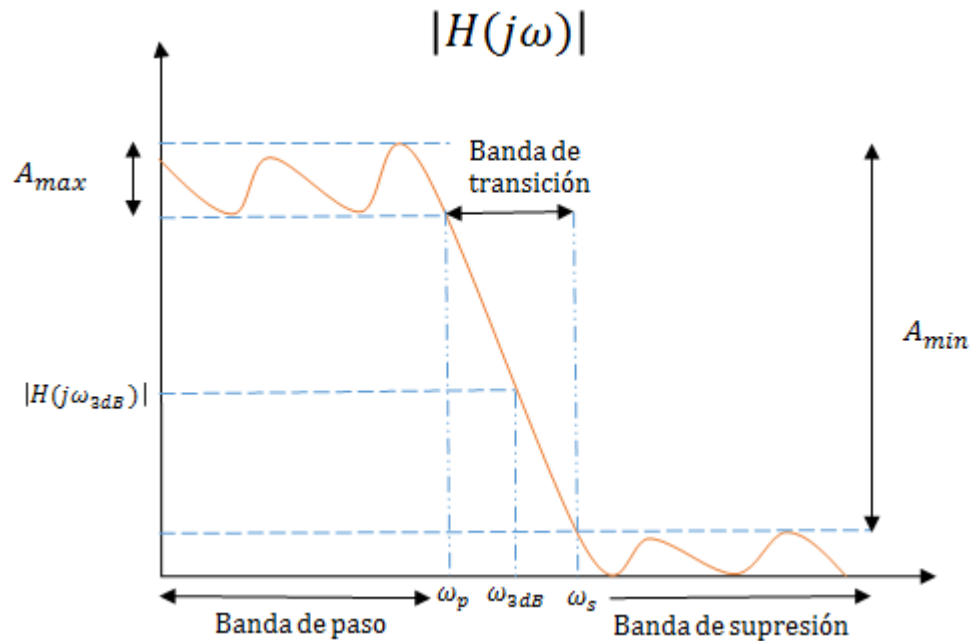
1. Implementar el algoritmo metaheurístico de optimización en MATLAB.
2. Verificar el funcionamiento del algoritmo frente a diferentes funciones de prueba comparando los resultados con los reportados en la literatura.
3. Aplicar el algoritmo del murciélago virtual en ejemplos de diseño óptimo de filtros analógicos con restricciones de parámetros de frecuencia y tiempo y compararlo con resultados frente a otros métodos.

3. MARCO TEÓRICO

3.1. FILTROS ANALÓGICOS

Los filtros permiten discriminar rangos de frecuencia, aceptando algunos y rechazando otros. Un filtro pasa bajas, por ejemplo, determina una banda de paso hasta una frecuencia ω_p , tal como lo muestra la Figura 1. Si el filtro tiene ganancia unitaria, la magnitud de las señales de entrada no se altera. Sin embargo, a partir de la frecuencia ω_s , banda de supresión, las señales se atenúan. Adicionalmente, existe una banda de transición ω_s y ω_p , debido a que los filtros no pueden cambiar inmediatamente entre cada una de las bandas. Por lo tanto, se define la frecuencia ω_{3dB} de corte o 3 dB, a partir de la cual se considera que las señales son rechazadas, pues en este punto la amplitud de salida es del 70.7% de la amplitud máxima posible. Los filtros estables o en general, sistemas estables, están acotados en amplitud, y matemáticamente son aquellos que tienen todos sus polos (raíces del denominador) de la función de transferencia en el semiplano izquierdo del plano complejo. En este documento solo se tendrán en cuenta sistemas estables.

Figura 1. Filtro pasabajas

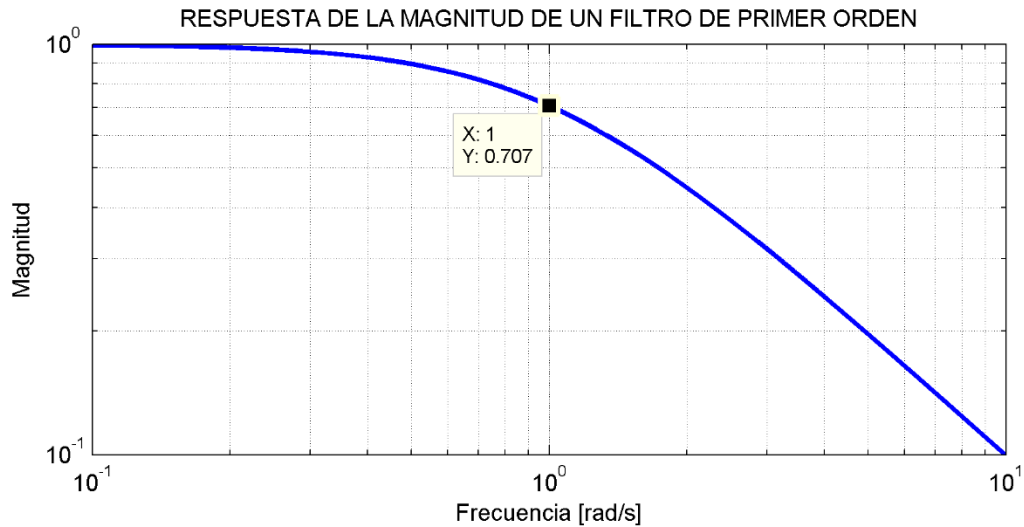


3.2. FILTROS DE PRIMER ORDEN Y SEGUNDO ORDEN

Los filtros de primer orden solo contienen un polo real, como lo muestra la ecuación (1), para un filtro pasa bajas con frecuencia de corte o de 3dB igual a ω_c , este tipo de filtros se caracteriza por tener una caída de 20 dB por década a partir de la frecuencia de corte, lo que significa que por cada década de frecuencia la magnitud se reduce 10 veces, como se muestra en la Figura 2, para este ejemplo $\omega_c = 1 [rad/s]$ por lo que en $10 [rad/s]$ la magnitud del filtro es $10^{-1} = 0.1$.

$$H(j\omega) = K * \frac{\omega_c}{\omega_c + j\omega} \quad (1)$$

Figura 2. Filtro de primer orden con frecuencia de corte unitaria



Por otro lado, los filtros de segundo orden tienen un par de polos complejos conjugados que les permite una caída de 40 dB por década, para un filtro de paso bajo su ecuación característica es (2), donde ω_o es la frecuencia natural del filtro y ξ es el factor de amortiguamiento.

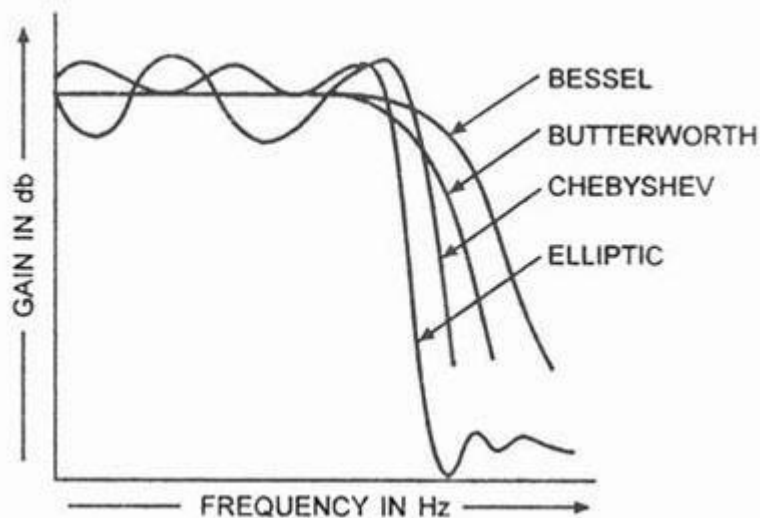
$$H(j\omega) = K * \frac{\omega_o^2}{(j\omega)^2 + 2 * \omega_o * \xi * j\omega + \omega_o^2} \quad (2)$$

De lo descrito anteriormente se deben resaltar dos hechos importantes. Primero que a partir de filtros de primer y segundo orden se pueden obtener fácilmente filtros de orden superior que permitan satisfacer los parámetros de diseño que sean necesarios para resolver cualquier problema. Igualmente, conectando en cascada filtros de primer orden se puede obtener un filtro de orden n . Sin embargo, estos filtros en cascada tiene una gran desventaja: la magnitud no es constante en la banda de paso. Segundo, a partir de los filtros pasa bajas se puede obtener cualquier otro tipo de filtro, ya sea pasa altas, rechaza banda, pasa banda, etc. Además los filtros pasa bajas no contienen ceros, esto será importante para desarrollar la función objetivo.

3.3. FILTROS CLASICOS

El diseño de filtros está abarcado prácticamente por cuatro métodos tradicionales (Butterworth, Chebyshev, filtros elípticos y Bessel), cada uno de ellos con una ventaja sobre los otros. De acuerdo con [2] el diseñador se tiene que amoldar a alguno de estos cuatro filtros, aun cuando lo ideal sería que el filtro cumpliera con todos los parámetros de diseño. En la Figura 3 se puede apreciar una gráfica con las respuestas típicas en el dominio de la frecuencia para cada tipo de filtro [3], y en el ANEXO B se detalla la información sobre ellos.

Figura 3. Comparación gráfica de los tipos clásicos de filtros



Fuente: [3]

3.4. APROXIMACIÓN COMPUTACIONAL AL DISEÑO DE FILTROS

Para realizar una aproximación computacional al diseño de filtros es necesario transformar la función de transferencia general de un filtro descrita en (3) en una función objetivo $f(x)$ que se pueda optimizar sujeta a restricciones $g_i(x)$. De

acuerdo con (3), se debe establecer un esquema de tolerancia que permita encontrar las inecuaciones para restringir la función $f(x)$.

$$H(s) = \frac{b_m s^m + \dots + b_2 s^2 + b_1 s + b_0}{a_n s^n + \dots + a_2 s^2 + a_1 s + a_0} \quad (3)$$

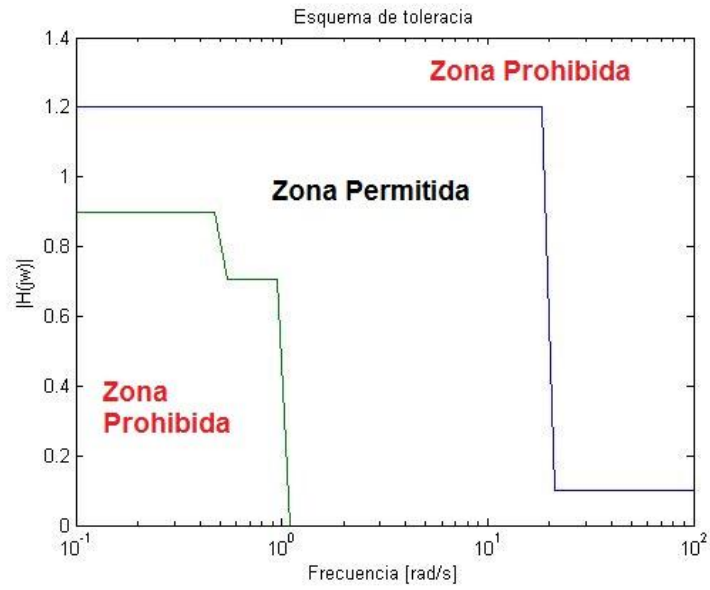
El esquema de tolerancia que se muestra en la Figura 4 permite establecer la magnitud deseada en valores de frecuencia específicos, generando tantas inecuaciones como desee el diseñador. De este modo se obtiene la magnitud deseada en las bandas de paso y supresión, así como las frecuencias que limitan estas bandas. Por otro lado, se tiene que la función de transferencia (3) se puede expresar en función de sus polos y ceros como se muestra en la ecuación (4), y en este caso el objetivo se transforma en hallar los polos y ceros en lugar de los coeficientes de (3).

Si bien estas dos representaciones son idénticas, con la segunda se obtienen ciertos beneficios. Entre ellos destaca el tener en cuenta restricciones en el tiempo con mayor facilidad, ya que su magnitud de respuesta en frecuencia se puede expresar de acuerdo con sus polos conjugados y reales como se muestra en (5).

$$H(s) = \frac{b_m * (s - z_1)(s - z_2) \dots (s - z_m)}{(s - p_1)(s - p_2) \dots (s - p_n)} \quad (4)$$

$$|H(jw)| = |H_{conj} * H_{real}| \quad (5)$$

Figura 4. Esquema de tolerancia



Donde, se presenta en [4] que:

$$|H_{conj}| = \left| \prod_{k=1}^p \frac{x_k^2 + y_k^2}{(j\omega)^2 + 2 * x_k * j * \omega + x_k^2 + y_k^2} \right| \quad (6)$$

$$|H_{real}| = \left| \prod_{k=1}^q \frac{z_k}{j\omega + z_k} \right| \quad (7)$$

3.5. ALGORITMO DEL MURCIELAGO VIRTUAL

El algoritmo del murciélago virtual (Bat algorithm) es un algoritmo metaheurístico propuesto para solucionar problemas de optimización de varias variables. Fue desarrollado por Xin-She Yang en 2010 [5], y se basa principalmente en el comportamiento de la eco-localización utilizada por los murciélagos con pulsos variables de emisión e intensidad.

Para el desarrollo del algoritmo se idealiza la eco-localización de murciélagos, que puede resumirse de la siguiente manera: Se inicializa la población de murciélagos virtuales aleatoriamente, cada murciélago tiene una velocidad, posición (solución) y frecuencia; también cuenta con una tasa de pulsos y una intensidad de sonido ajustables. Se establece el mínimo global $f(x)^*$, el cual sirve de referencia para guiar a los murciélagos virtuales. Adicionalmente se realiza una búsqueda local y aleatoria. A medida que se acerca a la presa, se actualizan los valores de la tasa de pulsos y la intensidad hasta que el mínimo global cumple ciertos criterios de parada.

Para la primera parte de la simulación descrita en la Figura 5, es necesario definir un rango de frecuencias $[f_{min}, f_{max}]$ para la simulación de los murciélagos virtuales, una posición $x_i (i = 1, 2, \dots, n)$ y velocidad v_i para un espacio de búsqueda d -dimensionado, Las soluciones dadas para posiciones x_i^t y velocidades v_i^t en el paso de tiempo t están dadas por la ecuaciones(8),(9) y (10):

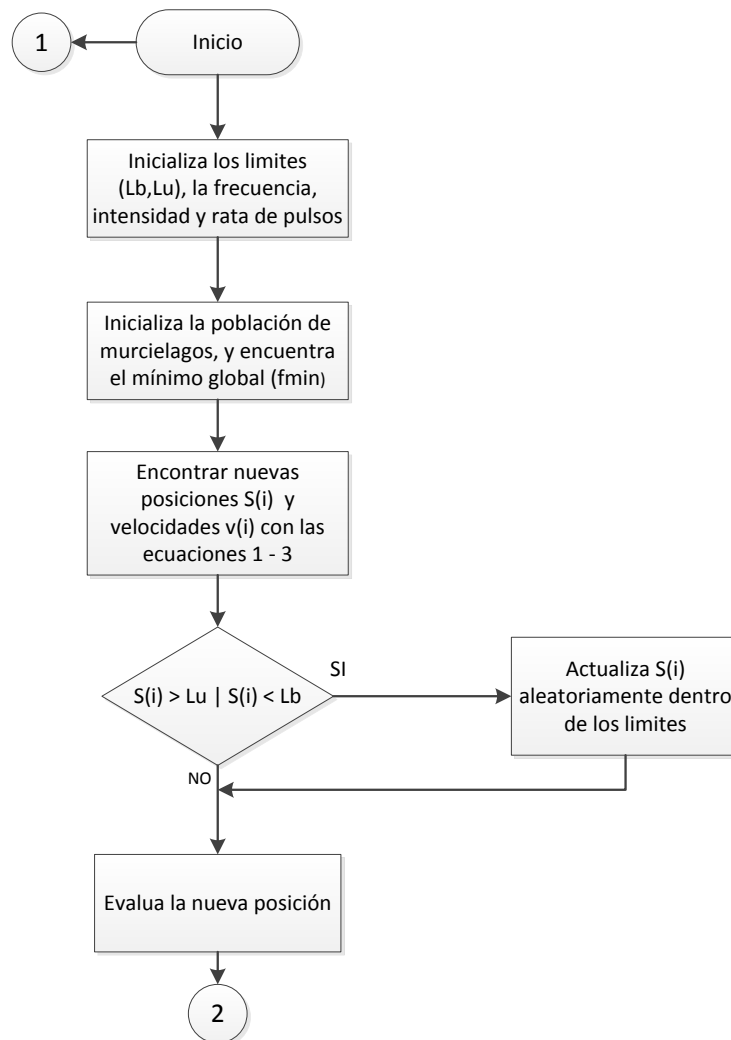
$$f_i = f_{min} + (f_{max} - f_{min})\beta \quad (8)$$

$$v_i^t = v_i^{t-1} + (x_i^t - x^*)f_i, \quad (9)$$

$$x_i^t = x_i^{t-1} + v_i^t \quad (10)$$

Donde $\beta \in [0,1]$ es un vector aleatorio de distribución uniforme; la mejor ubicación (solución) actual x^* se encuentra después de realizar la comparación de todas las soluciones arrojadas por n murciélagos.

Figura 5. Diagrama de bloques BAT ALGORITHM, primera parte.

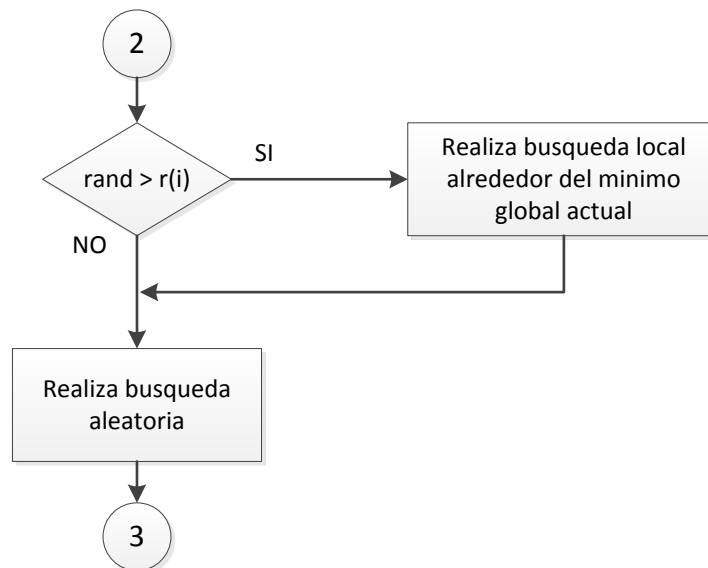


Para la búsqueda local señalada en la Figura 6, se seleccionada la mejor solución encontrada hasta el momento y se genera una nueva solución aleatoriamente alrededor de esta, como muestra la ecuación (11).

$$x_{new} = x_{old} + \epsilon * \rho * A^t \quad (11)$$

Donde ϵ esta entre $[-1, 1]$ es un número aleatorio, mientras que $A^t = \langle A_i^t \rangle$ es la intensidad promedio de todos los murciélagos en este paso de tiempo.

Figura 6. Diagrama de bloques BAT ALGORTIHM, segunda parte.



Por otra parte, es necesario controlar la intensidad A_i y la tasa de emisión de pulsos r_i , las cuales son actualizadas a medida que se encuentran mejores soluciones para cada uno de los murciélagos. La intensidad disminuye una vez al murciélago ha encontrado su presa, mientras que la tasa de las emisiones de impulsos aumenta. La intensidad puede ser escogida como un valor a conveniencia, al igual que la rata de pulsos. Sin embargo, y por comodidad, al inicializar la primera será cercana a la unidad, mientras que la segunda será cero. Posteriormente se actualizarán de acuerdo a las ecuaciones (12) y (13),

$$A_i^{t+1} = \alpha A_i^t \quad (12)$$

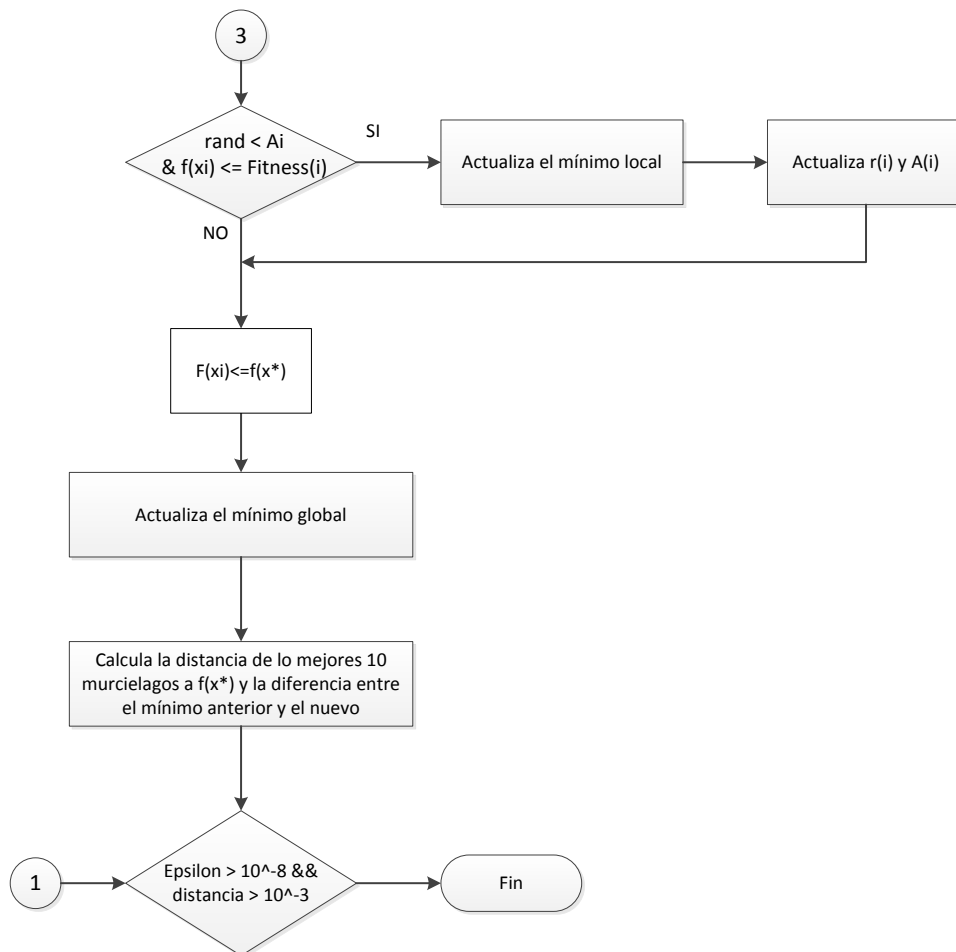
$$r_i^t = r_i^0 [1 - \exp(-Yt)] \quad (13)$$

donde α y Y son constantes. Para cualquier $0 < \alpha < 1$ y $Y > 0$, se tiene:

$$A_i^{t+1} \rightarrow 0, r_i^t \rightarrow r_i^0 \quad (14)$$

La selección de estas constantes se hace empíricamente, dependiendo de la función objetivo. Para una mejor interpretación del funcionamiento de esta parte del algoritmo se presenta en el diagrama de flujo de la Figura 7.

Figura 7. Diagrama de bloques BAT ALGORTIHM, tercera parte.



4. IMPLEMENTACIÓN DEL ALGORITMO DEL MURCIÉLAGO VIRTUAL

Para utilizar el algoritmo del murciélago en el diseño óptimo de filtros se debe asegurar su convergencia, razón por la cual se realizaron ciertas modificaciones en el criterio de parada. En este capítulo se explican las modificaciones y se valida el algoritmo con cinco funciones de prueba. Además, se comparará con resultados obtenidos en la literatura. Para una mejor comprensión de la validación y comparación se establecen las variables con su respectivo significado en la Tabla 1 para relacionarse con la terminología empleada en este capítulo. Cabe resaltar que el número de evaluaciones de la función es completamente diferente al número de iteraciones del algoritmo, ya que el número de iteraciones aumenta cada vez que la población de murciélagos ha realizado cierta cantidad de evaluaciones de la función.

Tabla 1. Terminología del algoritmo.

N	Población total de murciélagos
ε	Valor de referencia para comparar entre los mínimos globales encontrados
δ	Valor de referencia para comparar entre las distancias de los 10 primeros murciélagos
α	Constante de disminución de la intensidad
γ	Constante para el aumento de la rata de pulsos
ρ	Constante de proporcionalidad para la búsqueda local
Evaluaciones máx.	Numero de evaluaciones máxima de la función

4.1. CRITERIO DE PARADA

Un punto trascendental para el correcto funcionamiento del algoritmo es el criterio de parada. El método más simple [6] sería ejecutar el algoritmo un número de veces determinado por el usuario, pero esto no es conveniente, por lo que se utilizaron dos criterios de parada que son inclusivos, es decir, se deben cumplir ambos para que se detenga el algoritmo:

- Primer método: Se guarda en un registro el valor actual del mínimo global y cuando se obtiene un nuevo mínimo global se computa el valor absoluto de la resta y se compara con un valor de referencia llamado ε . Posteriormente se guarda el nuevo valor encontrado en el registro para una nueva comparación.
- Segundo método: Este método permite asegurar que todos los murciélagos converjan al mismo punto, evitando así que se caiga en un mínimo local y mejorando la efectividad en funciones multimodales. Además sirve como respaldo del primer método. Consiste en calcular la distancia entre la posición del mínimo global y la respectivas posiciones del 25% de murciélagos de la población total (valor seleccionado arbitrariamente). Después se compara con un valor δ y se decide si el algoritmo se detiene o si es necesario realizar nuevamente el procedimiento.

Los valores de referencia para el criterio de parada pueden variar de acuerdo a la función objetivo. Cabe resaltar que en el diseño de filtros la precisión no necesariamente debe ser muy exigente, con un valor de 10^{-3} es suficiente, pero si se debe obtener una elevada tasa de convergencia.

4.2. VALIDACIÓN DEL ALGORITMO

En esta sección se mostrará el desempeño del algoritmo del murciélago virtual utilizando el criterio de parada de la sección anterior, para encontrar el mínimo global de cinco funciones de prueba utilizadas ampliamente en la literatura [7][8]. Además se hará un análisis detallado del comportamiento del algoritmo, así como de algunas de sus ventajas y su convergencia para diferentes dimensiones de las funciones. En estas pruebas todas las funciones de prueba fueron ejecutadas 33 veces.

De acuerdo con [5] el algoritmo del murciélago virtual tiene un desempeño superior al que se obtiene con métodos metaheurísticos como PSO y los algoritmos genéticos (GA), sin embargo, el comportamiento del BA depende de varias constantes que se deben ajustar empíricamente. Por esta razón, al final de esta sección, en el numeral 4.2.6, se realizará la comparación del algoritmo del murciélago virtual propuesto contra los resultados presentados en [5].

4.2.1. FUNCIÓN DE JONG

La función de De Jong está determinada por la ecuación (15), se caracteriza por ser monótona y tener un mínimo global en $f_{min} = 0$ en $(0,0,\dots,0)$. Es ampliamente utilizada en la validación de algoritmos debido a su sencillez, lo cual permite realizar pruebas en dimensiones de un alto orden.

$$f(x) = \sum_{i=1}^d x_i^2 \quad (15)$$

Los valores utilizados por el algoritmo en este ejercicio se muestran en Tabla 2. La Figura 8 muestra gráficamente la función para $d = 2$, el caso más sencillo. En la Figura 8 también se muestran los 40 murciélagos virtuales alrededor del mínimo

(en rojo) en la última iteración. En la Figura 9 se presenta de una manera más detallada la convergencia de los murciélagos en la gráfica de contorno de la función.

Tabla 2. Parámetros utilizados para la función De Jong con $\alpha=0.8$ y $\gamma=0.9$.

Constante	Valor
Límite superior	10
Límite inferior	-10
Frecuencia máx.	0.1
Frecuencia mín.	2
ε	10^{-5}
δ	10^{-1}
α	0.8
γ	0.9
ρ	1
Evaluaciones máx.	5000000
N	40

Figura 8. Función De Jong con $d=2$.

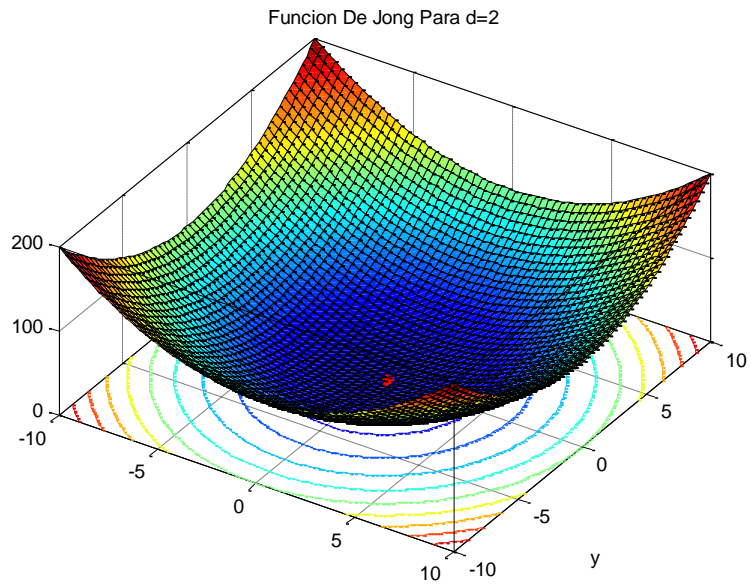
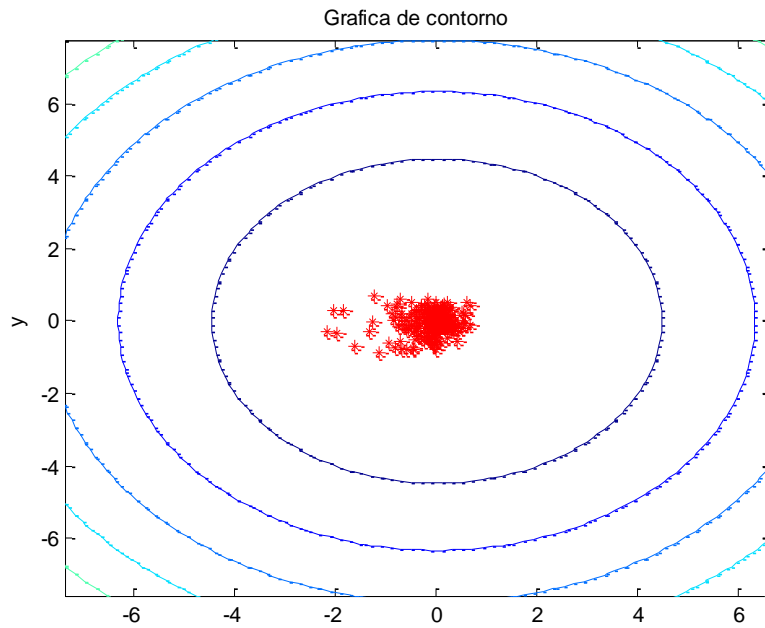


Figura 9. Convergencia de los murciélagos $d=2$ para $\alpha=0.8$ y $Y=0.9$



Al analizar los resultados de las simulaciones en la Tabla 3, se encuentra que la convergencia mejora pero que el costo computacional aumenta. Aun así, el efecto global se toma como positivo y por tanto se usa este criterio para las pruebas siguientes. . Por otro lado, si se desea un mínimo global mucho más preciso, es decir, alrededor de dos órdenes menos, esto se puede lograr variando los parámetros de la búsqueda local.

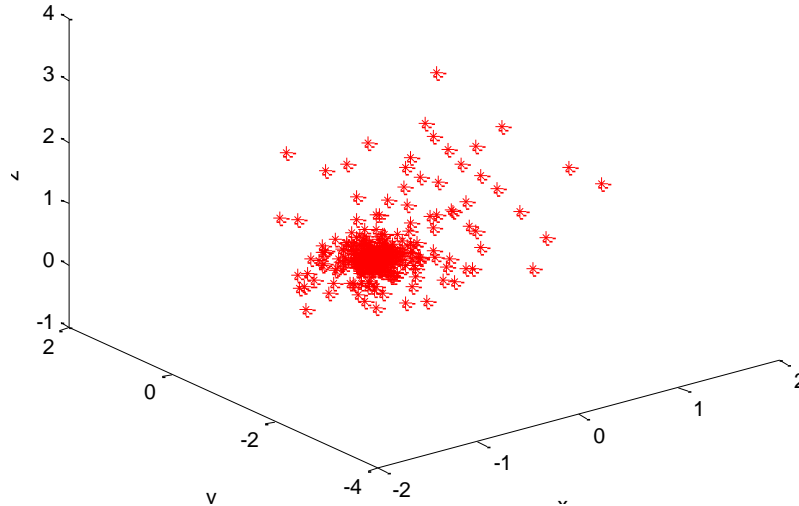
Tabla 3. Resultados para De Jong $d=2$ $\alpha=0.8$ y $\gamma=0.9$.

Evaluaciones promedio	49339
Desviación Estándar del número de evaluaciones de la función objetivo.	10776
Convergencia	100%
Promedio Mínimo Global	4.9518e-5
Dimensión	2

Para $d = 3$ se investigó acerca del movimiento de los murciélagos virtuales. La Figura 10 muestra un comportamiento en forma de enjambre, esto debido a las características que tiene este algoritmo de PSO y la cual es muy distinta al movimiento de otros algoritmos, como por ejemplo el de la espiral [9]. Este resultado permite mostrar la similitud del algoritmo aquí desarrollado con lo mencionado en [5].

Figura 10. Convergencia para De Jong para $d=3$, $\alpha=0.8$ y $\gamma=0.9$

Convergencia de los murcieagos para $d=3$



Al igual que en caso de dos dimensiones, en la Tabla 4 se presentan los resultados de las 33 simulaciones realizadas para $d=3$. Como era de esperarse, a medida que aumentan las dimensiones de la función objetivo, también aumenta el número de evaluaciones. Sin embargo, los resultados para dimensión igual a 256 muestran la principal ventaja del método de parada, en el cual la convergencia aumenta y se acerca mucho más al valor mínimo de la función (ver Tabla 5).

Tabla 4. Resultados para De Jong $d=3$

Evaluaciones promedio	178791
Desviación Estándar del número de evaluaciones de la función objetivo.	40916
Convergencia	100%
Promedio Mínimo Global	9.9079e-5

Dimensión	3
-----------	---

En el caso de $d = 256$, aunque la función de prueba es muy sencilla, el número de dimensiones tiende a dificultar la llegada del mínimo a valores alrededor de 10^{-5} , que es lo más común en este tipo de problemas, pero el criterio de parada permite que el algoritmo se detenga en un valor que es aceptable, teniendo en cuenta la aplicación en la que se utilizará. En la Tabla 5 se evidencia que el número de evaluaciones es similar al obtenido para tres dimensiones al igual que la desviación estándar.

Tabla 5. Resultados para De Jong $d=256$

Evaluaciones promedio	188310
Desviación Estándar del número de evaluaciones de la función objetivo.	5930
Convergencia	100%
Promedio Mínimo Global	0.093
Dimensión	256

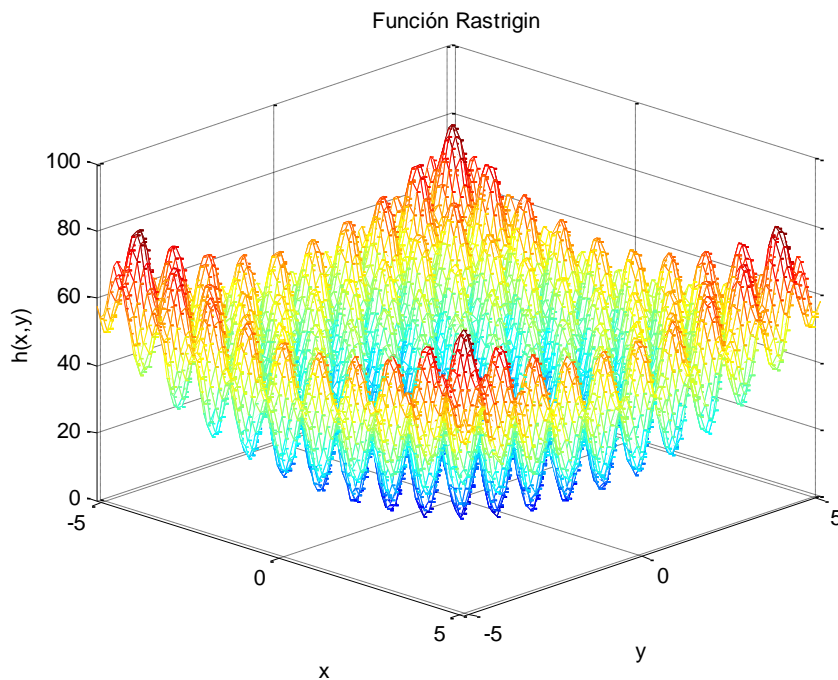
4.2.2. FUNCIÓN RASTRIGIN

La función de Rastrigin $h(x)$, tiene al igual que la función De Jong su mínimo en $(0,0,\dots,0)$, pero a diferencia de esta, la función Rastrigin es multimodal, tiene muchos mínimos globales lo cual dificulta ampliamente la tarea de optimización del algoritmo, aumentando el número de evaluaciones de funciones. Los parámetros para esta función son iguales a los de la Tabla 2, pero se modifican los

límites a $-5.12 < x_i < 5.12$ y $\alpha = 0.7$. En la Figura 11 se muestra una representación gráfica de la función Rastrigin, como se mencionó anteriormente esta función tiene varios picos que dificultan el trabajo del algoritmo, razón por la cual solo se suele probar en dimensión igual a 2.

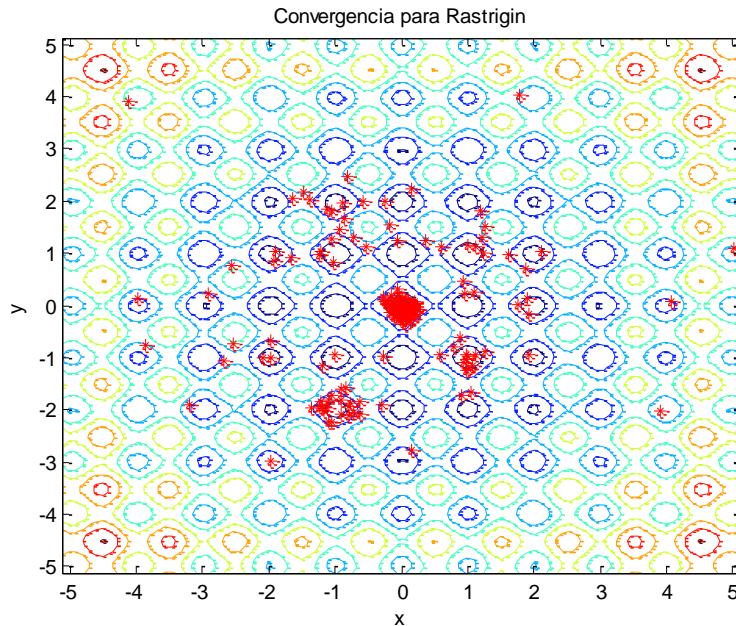
$$h(x) = \sum_{i=1}^d x_i^2 - 10 * \cos(2 * \pi * x_i) + 10 \quad (16)$$

Figura 11. Función Rastrigin d=2, $\alpha=0.8$ y $\gamma=0.9$



La característica multimodal de esta función puede ocasionar que el algoritmo se confunda y quede encerrado en uno de los mínimos locales. Para analizar mejor este efecto, considere la Figura 12. Si bien los murciélagos convergen a (0, 0), existen otras dos posiciones, (-1, -2) y (1, -1), en las cuales se realizaron bastantes evaluaciones, es decir, en algún momento fueron considerados mínimos globales, pero gracias a que los murciélagos se comunican entre sí, se logró encontrar el mínimo global.

Figura 12. Convergencia para Rastrigin $d=2$ con $\alpha=0.8$ y $Y=0.9$



Por otro lado, si se realiza un aumento a la zona de convergencia, como en la Figura 13, la aglomeración de murciélagos es sobresaliente debido a que el criterio de parada exige que para que el algoritmo se detenga, la diferencia entre la distancia de los murciélagos sea muy pequeña. Los resultados encontrados para la función Rastrigin en dimensión 2 y 3 se encuentran en Tabla 6, notándose que el promedio del mínimo es bastante preciso para el caso de $d=2$.

Por último se desea indagar acerca del comportamiento para dimensión igual a 3, a pesar que para esta dimensión no se tiene reporte en [5]. Dada la complejidad de esta función, los resultados obtenidos en la Tabla 6 son relativamente malos comparados con los obtenidos para $d=2$, pues el mínimo global está bastante alejado del ideal. Sin embargo, se tiene la ventaja de que el algoritmo se detuvo automáticamente en el 100% de las simulaciones, con un número de evaluaciones similar al caso de dimensión igual a 2 y como se ha mencionado anteriormente, para el diseño de filtros se desea una excelente tasa de convergencia con una precisión relativamente buena.

Figura 13. Aumento a la zona de convergencia

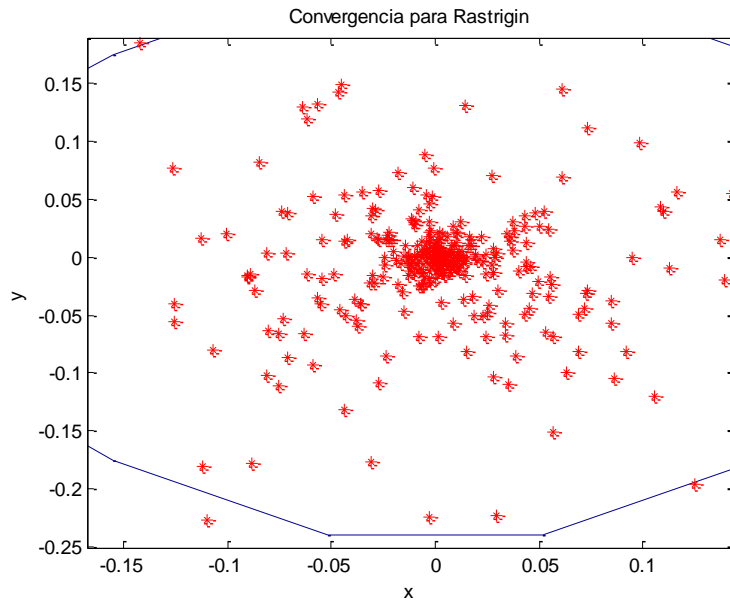


Tabla 6. Resultados para Rastrigin d=2 y d=3

	DIMENSIÓN=2	DIMENSION=3
Evaluaciones promedio	174114	195513
Desviación Estándar del número de evaluaciones de la función objetivo.	32900	24918
Convergencia	100%	100%
Promedio Mínimo Global	1.6067e-5	0.7245

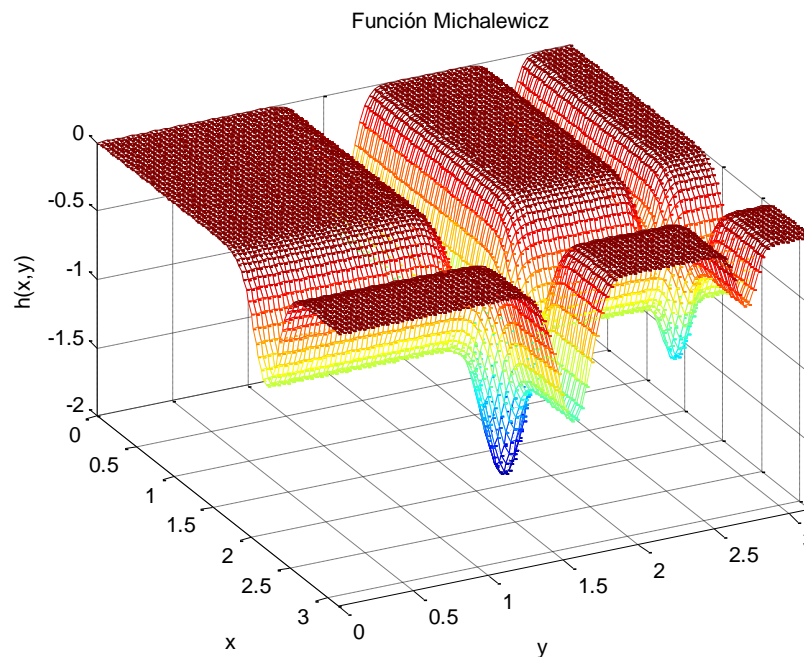
4.2.3. FUNCIÓN MICHALEWICZ

La función de Michalewicz, $g(x)$, con $m = 10$, posee distintos mínimos de acuerdo a la dimensión utilizada, lo cual es muy ventajoso para probar que el algoritmo no solo converja a mínimos que sean iguales a 0. Por ejemplo, para $d = 2$ se tiene

$g_{min} = -1.801$. Para estas simulaciones caso se mantienen los mismos parámetros que para la función Rastrigin, pero cambiando los límites a $0 < x_i < \pi$. En la Figura 14 se puede observar la representación gráfica, donde se garantiza que ni el mínimo está en el origen ni es igual a 0.

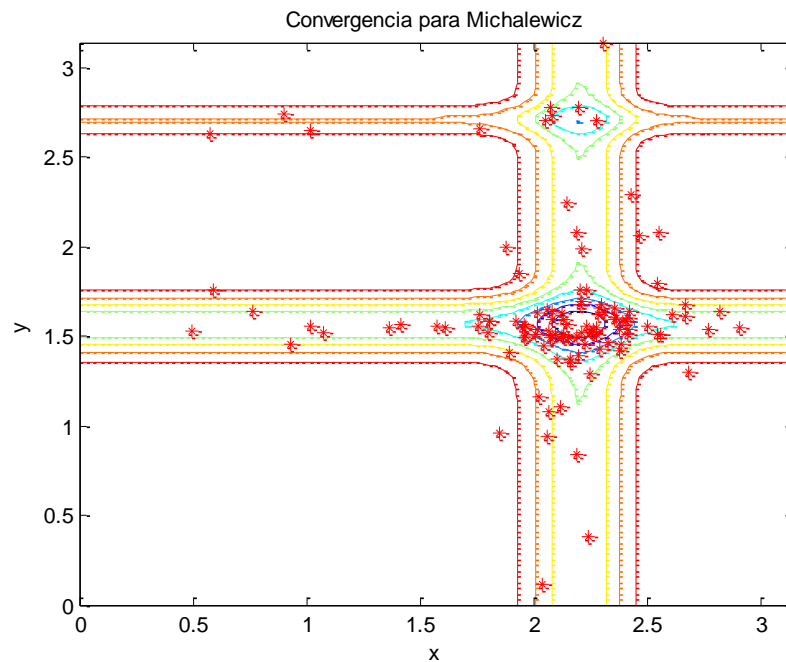
$$g(x) = - \sum_{i=1}^d \sin(x_i) * \left[\sin\left(\frac{ix_i^2}{\pi}\right) \right]^{2*m} \quad (17)$$

Figura 14. Función Michalewicz



Con el fin de observar el comportamiento de los murciélagos, se presenta en la Figura 15 el resultado final de las 20 primeras iteraciones para las mejores posiciones de los murciélagos virtuales, donde se evidencia como al terminar estas los murciélagos salen de las zonas donde se tiene un valor alto y la mayoría se posan sobre un punto mínimo.

Figura 15. Primeras Veinte iteraciones para Michalewicz.



Los resultados de la Tabla 7 muestran el buen comportamiento del algoritmo para esta función con $d=2$, pues en todas las simulaciones se encontró el mínimo ideal.

Por otro lado para $d = 5$, el comportamiento es muy distinto, puesto que se debe aumentar ε hasta 10^{-8} para lograr resultados suficientemente buenos, esto hace que aumente el número de evaluaciones de la función y aun así no se encuentre el mínimo ideal.

Tabla 7. Resultados para Michalewicz $d=2$ y $d=5$

	DIMENSION =2	DIMENSION =5
Evaluaciones promedio	114485	3434130
Desviación Estándar del número de evaluaciones de la función objetivo.	22900	245452

Convergencia	100%	97%
Promedio Mínimo Global	-1.8013	-4.5398

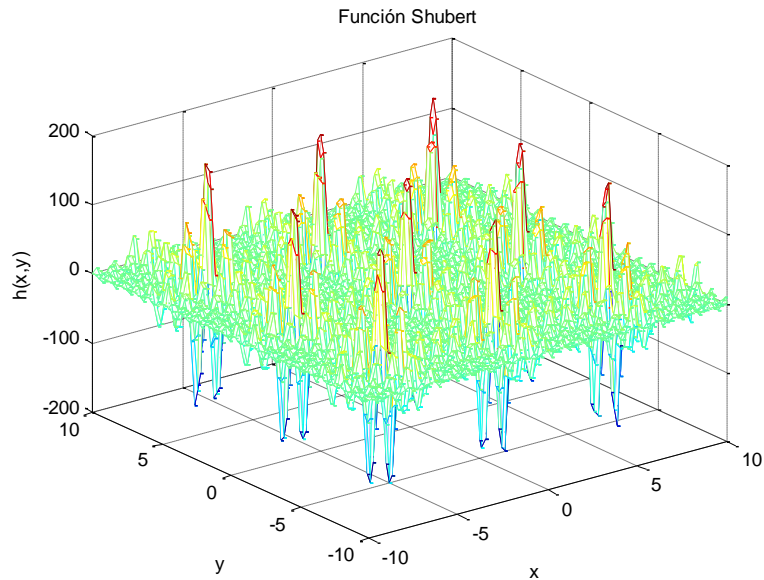
4.2.4. FUNCIÓN SHUBERT

La función de Shubert, $s(x)$, posee 18 mínimos globales; esta función permite evidenciar cómo se comporta el algoritmo para el caso en el que tiene múltiples posibles caminos. El dominio de la función se encuentra en $-10 < x_i < 10$, y para este caso se mantienen los valores de la Tabla 2 con excepción que se modifica $\delta = 0.6$.

$$s(x) = \sum_{i=1}^5 i * \cos((i + 1) * x_1 + i) \sum_{i=1}^5 i * \cos((i + 1) * x_2 + i) \quad (18)$$

En la Figura 16 se muestra la representación gráfica de la función, que está definida únicamente para dos dimensiones. Al igual que para Michalewicz en dimensión 2, los resultados de la Tabla 8, muestran que para esta función siempre se encontró el mínimo ideal, aunque el número de evaluaciones de la función si aumentó.

Figura 16. Función de Shubert



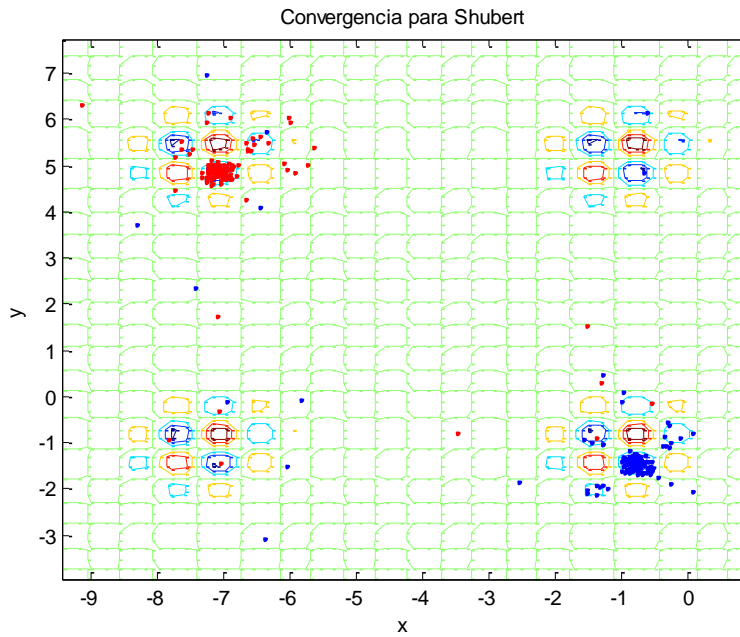
Por otro lado, al analizar el comportamiento de dos simulaciones por separado, se encontró que el algoritmo puede converger a cualquiera de los mínimos globales de la función, pero lo más importante es que como muestra la Figura 17, siempre se decidió por alguno de estos mínimos, por ejemplo, el enjambre rojo se aglutinó en el mínimo superior izquierdo mientras el azul en el inferior derecho, pero no se observan poblaciones en los otros mínimos. Esto es una gran ventaja del aumento de la tasa de pulsos y la disminución de la intensidad en el algoritmo, lo cual evita que el algoritmo tome dos caminos al mismo tiempo.

Tabla 8. Resultados para Shubert

Evaluaciones promedio	304163
Desviación Estándar del número de evaluaciones de la función objetivo.	34050
Convergencia	100%

Promedio Mínimo Global	-186.7309
Dimensión	2

Figura 17. Convergencia para Shubert en dos simulaciones (rojo y azul) con $\alpha=0.8$ y $Y=0.9$



4.2.5. FUNCIÓN ROSENBROCK

La función $r(x)$ tiene un mínimo global de 0, y su representación gráfica está en la Figura 18. Se realizaron pruebas para 2 y 16 dimensiones, consolidando los resultados en la Tabla 9 y en la Tabla 10. Adicionalmente, la Figura 19 muestra la convergencia de los murciélagos virtuales para el primer caso.

$$r(x) = \sum_{i=1}^{d-1} (1 - x_i^2)^2 + 100 * (x_{i+1} - x_i^2)^2 \quad (19)$$

Tabla 9. Resultados para Rosenbrock

Evaluaciones promedio	215036
Desviación Estándar del número de evaluaciones de la función objetivo.	43753
Convergencia	100%
Promedio Mínimo Global	4.4825e-05
Dimensión	2

Figura 18. Función Rosenbrock

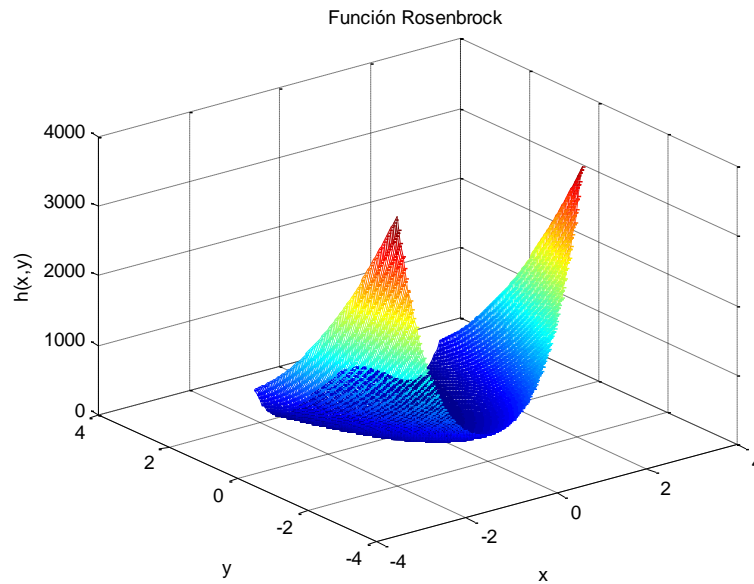
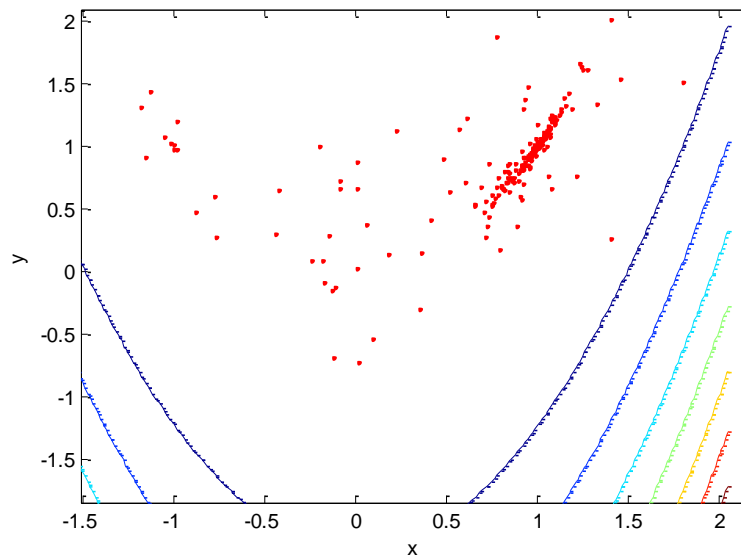


Figura 19. Convergencia para Rosenbrock d=2



Para el caso de 16 dimensiones se seleccionó $\alpha = 0.7$ con el fin de asegurar la convergencia. Los resultados mostrados en la Tabla 10 presentan $r_{min} = 4.4825 * 10^{-5}$ y lograron una tasa de convergencia del 97%.

Tabla 10. Resultados para Rosenbrock d=16

Evaluaciones promedio	1592729
Desviación Estándar del número de evaluaciones de la función objetivo.	221743
Convergencia	97%
Promedio Mínimo Global	1.6265e-05
Dimensión	16

4.2.6. COMPARACIÓN CON RESULTADOS ENCONTRADOS EN LA LITERATURA

Como ya se había mencionado al principio de este capítulo, los resultados obtenidos por el algoritmo implementado se compararon contra los presentados en [5], tanto para el algoritmo del murciélago propuesto por el autor, como para el algoritmo de optimización por enjambre de partículas (PSO); con el fin de que el algoritmo utilizado también sea comparado contra otros métodos de optimización. Se realizaron 100 simulaciones de cada función objetivo, al igual que en [5]. Los parámetros empleados en estas pruebas se encuentran en la Tabla 11, y los resultados obtenidos se muestran en la Tabla 12, que está dividida en dos partes, la primera para exhibiendo la comparación de resultados expuestos por el autor y la segunda con los resultados obtenidos por el algoritmo implementado, mostrando la dimensión utilizada para la función, junto con el promedio del número de evaluaciones de la función \pm la desviación estándar, redondeados al número entero más cercano. Por último entre paréntesis está el porcentaje de convergencia respecto a las 100 simulaciones. Cabe resaltar que las pruebas

realizadas con el algoritmo del murciélago propuesto se llevaron a cabo utilizando una dimensión distinta a la publicada, por esta razón se exhiben los resultados por separado.

Tabla 11. Parámetros de la prueba de comparación BA.

CONSTANTE	VALOR
Frecuencia máx.	0.1
Frecuencia mín.	2
ε	10^{-5}
α	0.8
γ	0.9
ρ	0.01
Evaluación máx.	1000000
N	40

Tabla 12. Resultados del algoritmo original y el algoritmo propuesto.

FUNCIÓN	ALGORITMO	
	PSO [5]	BA Yang [5]
Michalewicz	(d=16) 6922 ± 537 (98%)	(d=16) 4721 ± 753 (100%)
Rosenbrock	(d=16) 32756 ± 5325 (98%)	(d=16) 7923 ± 3292 (100%)
De Jong	(d=256) 17040 ± 1123 (100%)	(d=256) 5273 ± 490 (100%)
Schwefel*	(d=128) 14522 ± 1275 (97%)	(d=128) 8929 ± 729 (99%)
Ackley**	(d=128) 23407 ± 4325 (92%)	(d=128) 6933 ± 2317 (100%)

Rastrigin	79491 ± 3715 (90%)	12573 ± 3372 (100%)
Easom	17273 ± 2929 (90%)	7532 ± 1702 (99%)
Shubert's (18 minima)	23992 ± 3755(92%)	11925 ± 4049 (100%)

FUNCIÓN	BA UIS
Michalewicz	(d=2) 703 ± 39 (100%)
Rosenbrock	(d=2) 3215 ± 176 (100%)
De Jong	(d=3) 7097 ± 475 (100%)
Schwefel*	(d=2) 82918 ± 14638 (100%)
Ackley**	(d=2) 37159 ± 2841(100%)
Rastrigin	13337 ± 1380 (100%)
Easom	28361 ± 2922 (100%)
Shubert's (18 minima)	9724 ± 860 (100%)

En todas estas pruebas se conocía de antemano el valor del mínimo global, sin embargo para el problema de diseño de filtros este es totalmente desconocido, pues puede variar de acuerdo a las prioridades del diseñador, por lo tanto, se desarrolló el criterio de parada que asegura la convergencia del algoritmo, como se explicó en el numeral 4.1. ⁶

⁶ Para la función de Schwefel se modificó la frecuencia máxima y mínima a $2 * 10^{-3}$ y 10^{-6} respectivamente.

**Para la función de Ackley se utilizó $\alpha = 0.6$, pero se mantuvieron las frecuencias máximas y mínimas iniciales.

5. DESARROLLO DE FILTROS CON RESTRICCIONES EN TIEMPO Y FRECUENCIA BASADO EN EL ALGORITMO DEL MURCIELAGO VIRTUAL

5.1. FUNCIÓN OBJETIVO

Para establecer la función objetivo es necesario garantizar la frecuencia ω_{3dB} o frecuencia de corte, donde el módulo de la función de transferencia se reduce un factor de $\frac{1}{\sqrt{2}}$ con respecto al valor máximo [4], por lo tanto, se debe cumplir la ecuación (20), donde H_{max} es el valor máximo del módulo de la función de transferencia.

$$|H(j\omega_c)| = \frac{1}{\sqrt{2}} * H_{max} \quad (20)$$

Elevando al cuadro la ecuación (20) y reorganizando para utilizar el algoritmo de optimización se obtiene que la función objetivo (21).

$$f(\vec{x}) = ||H(j\omega_c)|^2 - 0.5|^2 \quad (21)$$

La ecuación está sujeta a diferentes restricciones del esquema de tolerancia, que permiten controlar la magnitud en la banda de paso y la de supresión, para incluir las restricciones (22) y (23) en la función objetivo se debe incluir una penalización.

$$g_i(x) = |H(j\omega_m)|^2 - B_u(\omega_m) \leq 0 \quad (22)$$

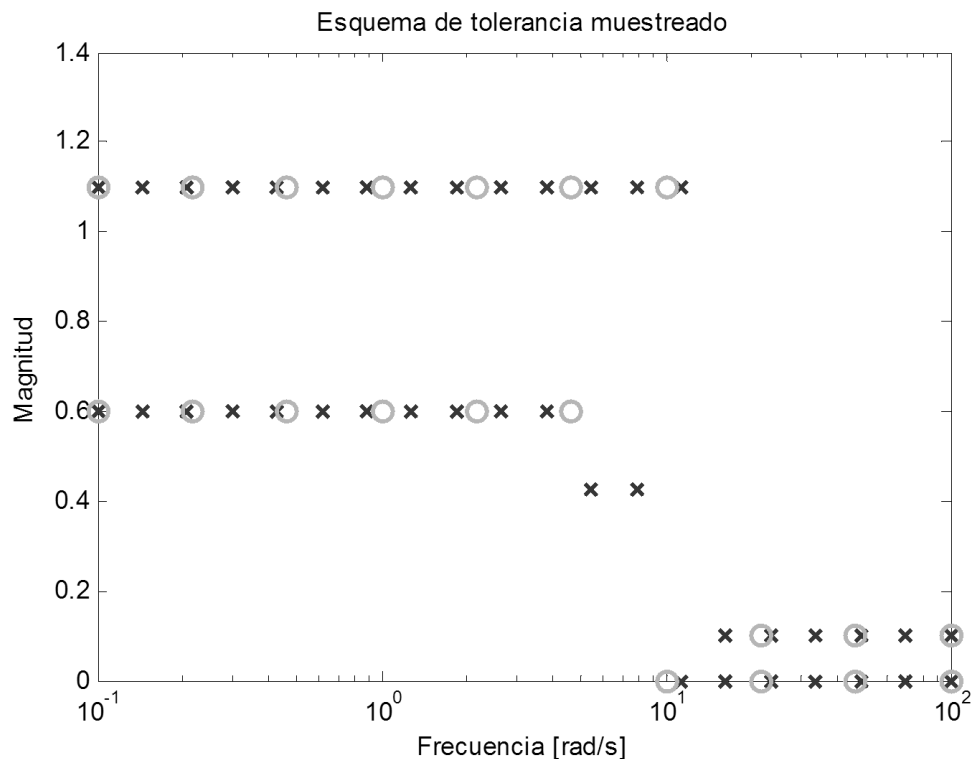
$$h_i(x) = -|H(j\omega_m)|^2 + B_l(\omega_m) \leq 0 \quad (23)$$

5.2. EL ESQUEMA DE TOLERANCIA

En la Figura 4, se presentó el esquema como un par de zonas prohibidas demarcadas por una línea continua. Esto genera una zona permitida para la

respuesta en frecuencia de la magnitud. Sin embargo, y dado que realmente se está trabajando con una representación discreta que permita su uso, computacionalmente el verdadero esquema de tolerancia depende del paso con el que se divida la frecuencia que se va a muestrear. La Figura 20 muestra el esquema de tolerancia para dos frecuencias de muestreo, una en negro (marcada con x) y otra en gris (marcada por círculos), siendo la primera el doble de la segunda. Es entonces en estos puntos donde se deben cumplir las desigualdades (22) y (23), por lo que en el resto de puntos pueden aparecer picos sin violar ningún parámetro. Es importante notar que si se desea un filtro muy exacto se debe realizar un esquema de tolerancia como muchos puntos, lo que conlleva un tiempo de procesamiento mayor y por lo tanto un mayor costo computacional.

Figura 20. Esquema de tolerancia muestreado con diferentes pasos



5.3. FUNCIÓN DE PENALIZACIÓN

La manera más práctica de incluir las restricciones de la función objetivo (21) es mediante el uso de funciones de penalización. Estas permiten utilizar algoritmos para problemas de optimización sin restricciones con el fin de resolver problemas que si están restringidos. Para esto se elige la función $\varphi(k, x)$, donde la constantes k y n se escogen arbitrariamente de acuerdo a los requisitos del problema.

$$\varphi(k, x) = \begin{cases} 0 & \text{si } x \leq 0 \\ k * x^n & \text{si } x > 0 \end{cases} \quad (24)$$

La idea es que para cumplir las ecuaciones (22) y (23), la ecuación (24) aumenta radicalmente si no se cumple la desigualdad, es decir, la función penaliza cualquier valor mayor a 0. Por lo tanto la función objetivo se modifica, adquiriendo la forma de (25), donde se tienen en cuenta las constantes de los pesos, de los que se hará una explicación detallada más adelante.

$$\begin{aligned} \tilde{f}(x) &= W_1 * f(x) + W_2 * \sum_{i=1}^m \varphi(k, g_i(x)) + W_3 * \sum_{i=1}^n \varphi(k, h_i(x)) \\ \tilde{f}(x) &= W_1 * ||H(j\omega_c)|^2 - 0.5|^2 \\ &\quad + W_2 \sum_{i=1}^m \varphi(k, |H(j\omega_m)|^2 - B_u(\omega_m)) \\ &\quad + W_3 \sum_{i=1}^n \varphi(k, -|H(j\omega_m)|^2 + B_l(\omega_m)) \end{aligned} \quad (25)$$

Además de cumplir con el esquema de tolerancia y la frecuencia de corte, el algoritmo debe tener en cuenta otras medidas de desempeño que se explican a continuación.

5.4. MEDIDAS DE DESEMPEÑO

5.4.1. RETARDO DE GRUPO Y RETRASO DE FASE

Para una función de transferencia $H(j\omega) = |H(j\omega)|\angle\theta(\omega)$ se define

$$t_p = \frac{\theta(\omega)}{\omega} \quad (26)$$

$$t_g = \frac{-d\theta(\omega)}{d\omega} \quad (27)$$

Donde t_p y t_g son respectivamente el retador de fase y el retardo de grupo.

5.4.2. SOBREPASO MÁXIMO

En la práctica se desea conocer el comportamiento de los filtros en el tiempo, aunque el diseño, por lo general, se lleva a cabo en el dominio de la frecuencia. El sobrepico o sobrepaso es una de las medidas que caracterizan el filtro en el tiempo, hace referencia al valor pico máximo de la respuesta del filtro ante una entrada escalón. Esta especificación es usada frecuentemente en sistemas de segundo orden y se encuentra plenamente determinada en la literatura [10], pero para sistemas de orden mayor se debe realizar una aproximación diferente.

De acuerdo con [11], la respuesta al escalón se puede obtener utilizando la transformación inversa de Laplace de la ecuación (6), como sigue:

$$H_{step}(s) = \frac{1}{s} \left[\prod_{k=1}^n \frac{a_k^2 + b_k^2}{(s - a_k + jb_k) * (s - a_k - jb_k)} \right] \quad (28)$$

Aplicando fracciones parciales,

$$H_{step}(s) = \frac{A}{s} + \sum_{k=1}^n \frac{B_k}{s - a_k + jb_k} + \frac{B_k^*}{s - a_k - jb_k} \quad (29)$$

Donde se obtiene fácilmente $A = 1$ y para B

$$B_k = \frac{1}{a_k + jb_k} * \prod_{k=1}^n \frac{a_k^2 + b_k^2}{j2b_k} \quad (30)$$

Finalmente, se sabe que $B_k = |B_k| \angle \theta$ y aplicando la transformada inversa para pasar al dominio del tiempo se llega a (31).

$$h_{step}(t) = 1 + 2 * B * e^{-a_k * t} * \cos(b_k * t - \theta) \quad (31)$$

6. SELECCIÓN DE PARAMETROS OPTIMOS PARA EL ALGORITMO DEL MURCIELAGO VIRTUAL

En esta sección se encontrarán los parámetros que muestren un mejor desempeño para el diseño de filtros. Para esto se utilizará el ejemplo del numeral 7.3 con el fin de realizar distintas pruebas para analizar el comportamiento del filtro de acuerdo a las especificaciones dadas. Por ahora, únicamente se debe tener en cuenta las especificaciones de diseño de la Tabla 21 y saber que se desea minimizar el efecto de sobrepico. El experimento consiste en realizar simulaciones con todas las posibles combinaciones de α y γ que se presentan en la Tabla 13, realizando 10 simulaciones para cada una. Los resultados de los polos encontrados, junto con las variables de error se encuentran en el Anexo A. A partir de los errores promedio encontrados, se seleccionó la mejor combinación, que es $\alpha = 0.9$ y $\gamma = 0.7$, con un error promedio de 0.192882775.

Tabla 13. Valores de α y γ para las simulaciones

α	γ
0.6	0.7
0.7	0.9
0.8	1.1
0.9	1.3

6.1. EFECTO DE LOS PESOS EN LA FUNCIÓN OBJETIVO

Es importante resaltar el efecto que pueden tener los pesos que se le den a función objetivo, buscando dar prioridad o no a algunas condiciones de diseño. Por esta razón la ecuación (25) tiene tres constantes llamadas “pesos” (W_1, W_2, W_3), y de acuerdo al valor que se les dé el algoritmo minimizará más un parámetro que otro. Adicionalmente la función de penalización tiene una constante (k) que permite evitar que se trabaje con números menores a 1, haciendo que la función sea más restrictiva. En esta sección se presentan ejemplos que no están optimizados ni tienen el comportamiento deseado por el diseñador, pero presentan características importantes para tener en cuenta en la ejecución del algoritmo.

Por ejemplo, en la Figura 21, se encuentra un ejemplo del efecto de distintos pesos para cumplir un esquema de tolerancia. En este caso hay una frecuencia de paso $\omega_p = 5$ [rad/s] y una frecuencia de atenuación $\omega_s = 30$ [rad/s]. Se desea diseñar un filtro de segundo orden, con dos polos complejos conjugados, que cumpla con este esquema. Además se debe cumplir que la frecuencia de corte o 3dB este en 10 [rad/s]. Estas especificaciones se encuentran en la Tabla 14 junto con los parámetros utilizados por el algoritmo en la Tabla 15.

Tabla 14. Especificaciones de diseño

Frecuencia de paso ω_p	5
Frecuencia de corte ω_c	10
Frecuencia de supresión ω_s	30
Amplitud pico-pico máxima en la banda de paso A_{max}	0.1
Amplitud máxima en la banda de supresión	0.01
Numero de polos	2

Tabla 15. Parámetros del algoritmo

Constante	Valor
Frecuencia máx.	0
Frecuencia mín.	2
ε	10^{-5}
α	0.9
γ	0.7
ρ	1
Evaluaciones máx.	1000000
N	40
n	2
Paso	1

Utilizando los pesos de la Tabla 16 y con $k = 1$, en todas las funciones de penalización, se obtienen los filtros de la Figura 21. En ellos se puede observar cualitativamente el efecto de los pesos. Por ejemplo, el filtro 1 cumple con la frecuencia de corte y el límite inferior y está muy cerca de cumplir con el límite superior en la frecuencia de paso, pero es el que peor atenuación tiene. Para cumplir con la atenuación en la frecuencia de supresión se aumenta el peso de esta variable y se genera el filtro 2, que claramente disminuye el error en la banda de supresión, pero deja de cumplir el límite inferior y la frecuencia de 3 dB. Finalmente, se le da prioridad a la frecuencia de supresión pero también al límite inferior, obteniendo el filtro 3 con un fuerte pico que viola el límite superior en la banda de paso, además de una atenuación menor a la del filtro 2.

Figura 21. Efecto de los pesos en la optimización

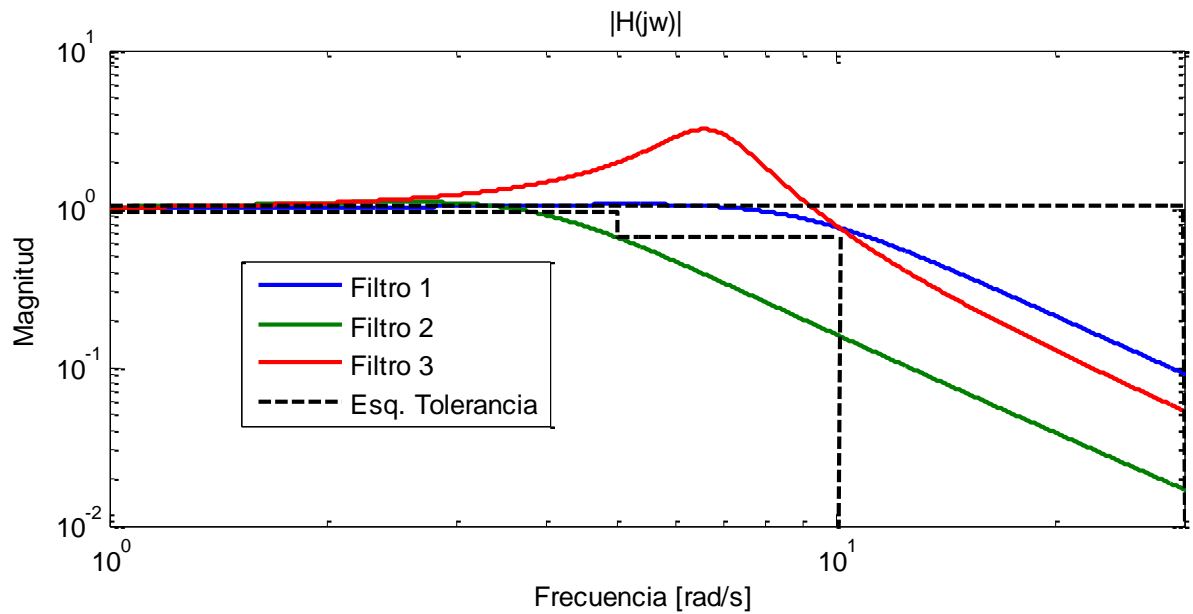


Tabla 16. Pesos para los filtros de la figura 18.

Condición	Filtro 1	Filtro 2	Filtro 3
Peso W_1	1	1	0.01
Peso W_2	1	1000	500
Peso W_3	1	0.1	10
Límite superior	200		
Límite inferior	0		

En conclusión, se deben escoger cuidadosamente las constantes de peso de acuerdo a las prioridades del diseñador, teniendo en cuenta que un filtro óptimo no es aquel que cumple estrictamente con todas las especificaciones (ya que esto probablemente no sea posible) sino es aquel que ajuste todos los parámetros de desempeño en el mejor equilibrio posible.

Con el fin de permitirle al diseñador tener una mayor información acerca del filtro encontrado, el algoritmo entrega las variables de la Tabla 17, las cuales permiten

obtener un valor cuantitativo de los errores del filtro respecto al esquema de tolerancia y de esta manera utilizar apropiadamente las constantes de las funciones de penalización. Para los filtros de la Figura 21 las variables de error se presentan en la Tabla 18.

Tabla 17. Variables informativas para calcular los pesos

Error_LS_ws	Guarda el error acumulado de la diferencia entre la respuesta de magnitud y el esquema de tolerancia superior hasta la frecuencia de supresión
Error_LS_wf	Igual que Error_LS_ws pero desde la frecuencia de supresión hasta la frecuencia final de muestreo.
Error_LI	Igual que Error_LS_ws, pero para el límite inferior.
Msp	Valor del sobrepico
fdex	Valor obtenido para la ecuación (21) para garantizar la frecuencia de corte.

Tabla 18. Variables informativas de los errores para diferentes pesos de la ecuación

	Filtro 1	Filtro 2	Filtro 3
Error_LS_ws	3.4193e-04	0.0028	0.0841
Error_LS_wf	0.0067	0.0502	0.9471
Error_LI	0.0178	0.2000	0.1296
fdex	0.0085	0.2246	0.0139
fmin	0.0334	0.4776	1.1746

7. RESULTADOS

En esta sección se realizan algunos ejemplos empleando el algoritmo del murciélago virtual. Primero se muestra cómo se diseñan filtros teniendo en cuenta el esquema de tolerancia, o simplemente ciertas especificaciones. Luego se optimizan los filtros teniendo en cuenta restricciones en tiempo, como el sobrepico, y otras características deseadas, como una fase lineal. Finalmente, se hace la transformación para obtener un filtro pasabanda a partir de un pasabajas diseñado con el algoritmo del murciélago.

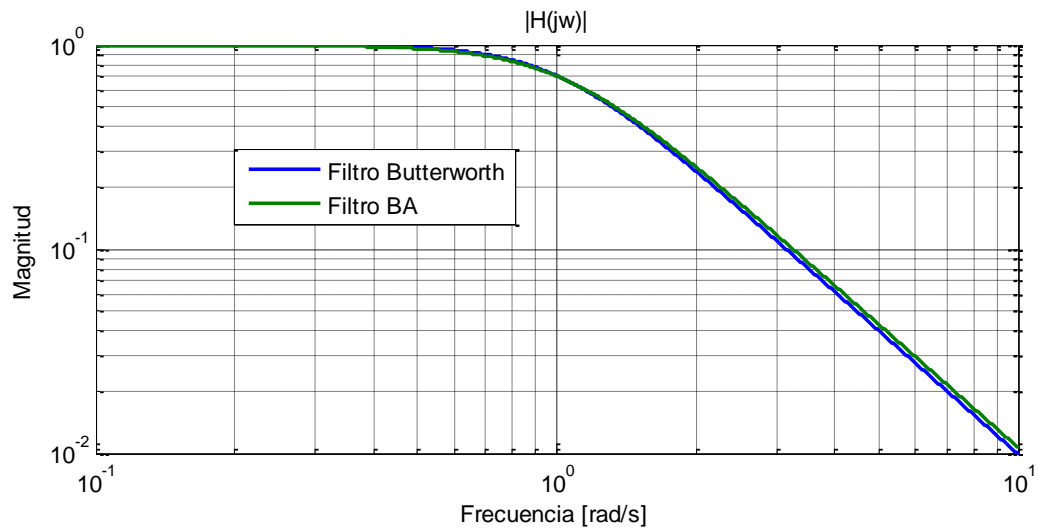
7.1. COMPARACIÓN CON EL DISEÑO DE UN FILTRO BUTTERWORTH

Para validar el correcto funcionamiento del algoritmo, se tiene en cuenta que un filtro Butterworth es por definición máximamente plano, es decir, este filtro será el óptimo si se requiere un filtro que sea lo más plano posible en su banda de paso. De esta manera sirve como referencia para validar nuevamente el algoritmo pero esta vez aplicado en el diseño de filtros. Para esto el algoritmo debe converger a los mismos polos del filtro Butterworth. En este ejemplo se supone un filtro Butterworth de segundo orden con una frecuencia de corte $\omega_c = 1 \text{ [rad/s]}$, la parte real e imaginaria de este filtro son los valores ideales. Se realizaron 33 pruebas y los resultados se presentan en la Tabla 19, y en la Figura 22 se muestra la respuesta del filtro Butterworth junto con la del filtro implementado, validando así el correcto funcionamiento del algoritmo para el diseño de un filtro maximente plano.

Tabla 19. Resultados de la validación del algoritmo con un filtro Butterworth como referencia

	a	jb
Valor ideal	0.707	0.707
Valor promedio	0.762	0.703
Error experimental (%)	7.835	0.452
Desviación estandar	1.260e-2	1.246e-3
Numero de evaluaciones promedio	2680	

Figura 22. Comparativa de la magnitud del filtro Butterworth con el filtro desarrollado a partir del algoritmo



7.2. DISEÑO DE FILTROS DE SÉPTIMO Y DÉCIMO ORDEN CON RESTRICCIONES EN FRECUENCIA

A continuación se valida el funcionamiento del algoritmo para diseñar filtros de hasta decimo orden. En este ejemplo se proponen dos filtros con las especificaciones de la Tabla 20: el primero puede oscilar su respuesta entre 0.8 y 1.2 en el límite superior, tal como muestra la Figura 23, donde se presenta adicionalmente un filtro Chebyshev de séptimo orden, nótese la similitud entre los dos filtros. En la Figura 24 se puede apreciar a más detalle que el filtro optimizado trata de converger hacia el filtro Chebyshev, y esto representa un indicio más de que el algoritmo funciona correctamente, pues el filtro Chebyshev es uno de los que mayor caída permite.

Por otro lado en la Figura 25 y la Figura 26 Se presenta un filtro de orden 10, para ratificar lo dicho anteriormente y para validar el algoritmo en órdenes superiores.

Tabla 20. Especificaciones para filtros de orden superior

Frecuencia de paso ω_p	0.5
Frecuencia de corte ω_c	1
Frecuencia de supresión ω_s	10
Amplitud pico-pico máxima en la banda de paso A_{max}	0.4
Amplitud máxima en la banda de supresión	1e-8

Figura 23. Respuesta en frecuencia de la magnitud para filtros de séptimo orden del ejemplo 7.2

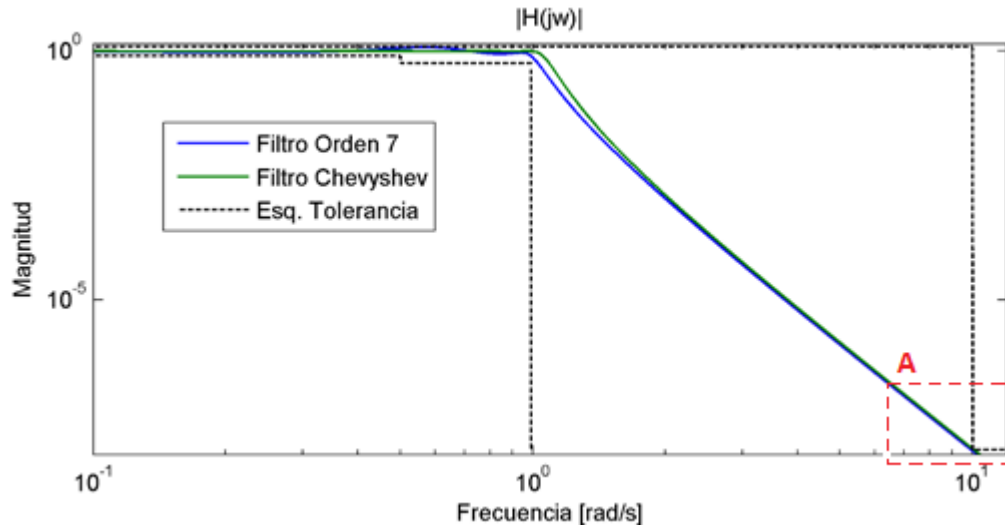


Figura 24. Detalle A de la Figura 23

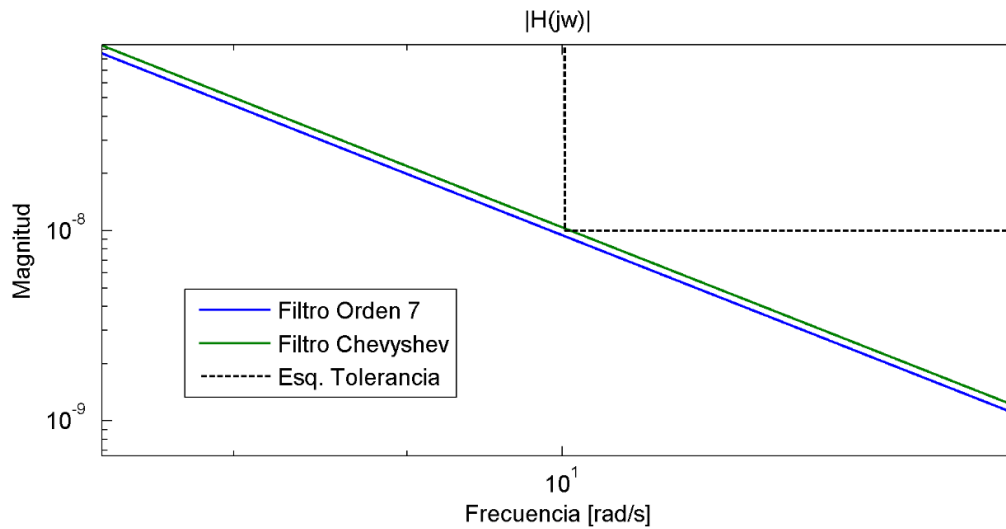


Figura 25. Respuesta en frecuencia de la magnitud para filtros de décimo orden del ejemplo 7.2

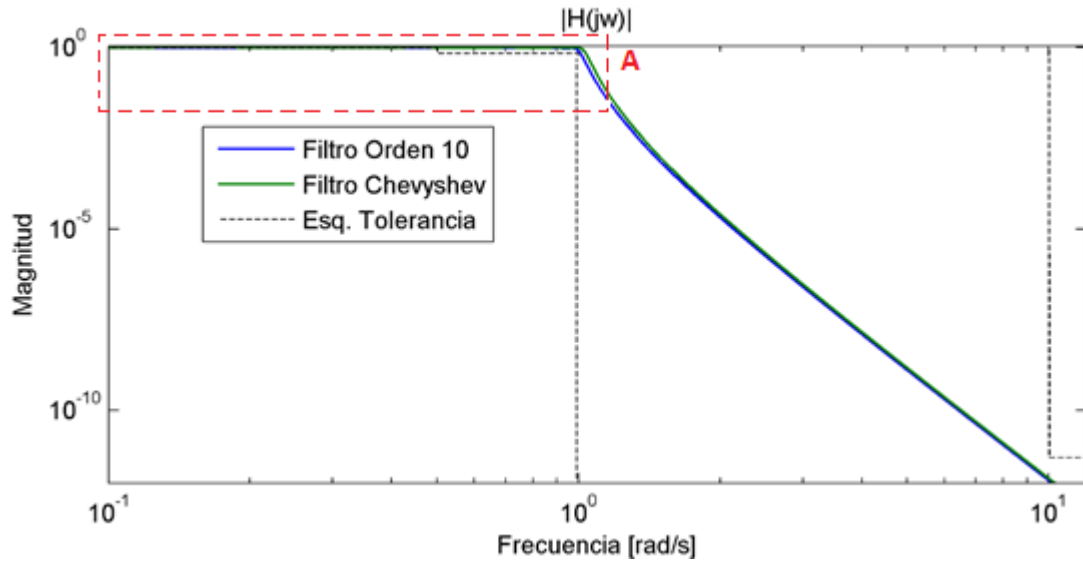
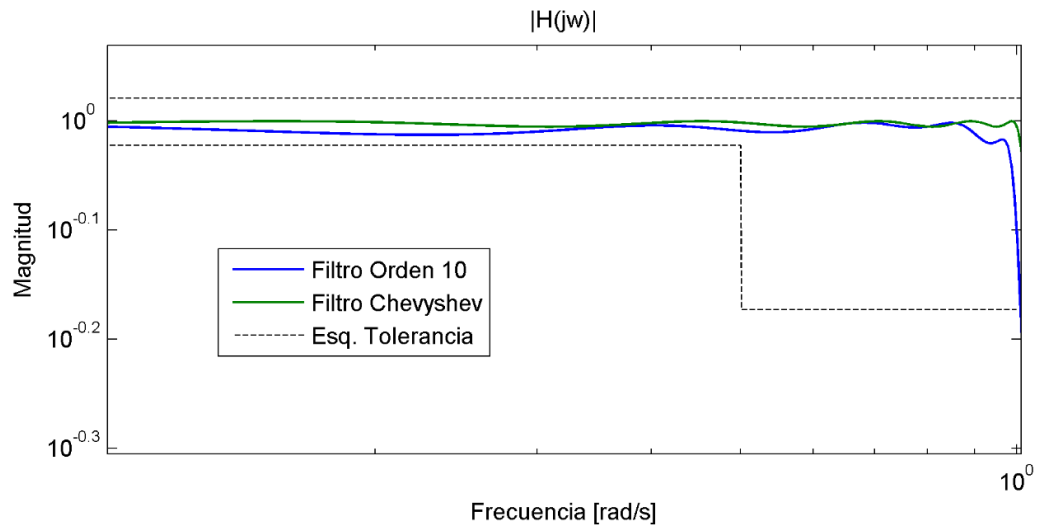


Figura 26. Detalle A de la Figura 25.



7.3. COMPARACIÓN FILTRO BUTTERWORTH SOBREPICO MÍNIMO

En este numeral se compara el desempeño de filtros Butterworth con los encontrados por el algoritmo para varios órdenes, Los filtros, además, deben presentar el mínimo sobrepico posible. La ecuación de transferencia del filtro Butterworth se encuentra con ayuda del comando *butter* de MATLAB®.

7.3.1. FILTROS DE SEGUNDO ORDEN

En este ejemplo se diseñan los dos filtros pasabajas con las especificaciones mostradas en la Tabla 21, uno empleando la función *butter* y el otro obtenido mediante el uso del algoritmo del murciélago. Los polos resultantes se encuentran en la Tabla 22.

Tabla 21. Especificaciones de diseño para el ejemplo del numeral 7.3.1 y 7.3.2

Frecuencia de paso ω_p	10
Frecuencia de corte ω_c	20
Frecuencia de supresión ω_s	100
Amplitud pico-pico máxima en la banda de paso A_{max}	0.2
Amplitud máxima en la banda de supresión	0.01

Tabla 22. Polos resultantes para los filtros del ejemplo 7.3.1

	a_1	jb_1
Filtro Butterworth	14.1421	14.1421
Filtro optimizado	28.0860	8.3066

En la Figura 27 se muestra la respuesta en frecuencia de la magnitud, donde tanto para el filtro Butterworth como para el filtro optimizado se cumplen todas las especificaciones de diseño. Claramente el filtro Butterworth tiene la respuesta más plana posible en la banda de paso y posteriormente cae más rápido que la del filtro optimizado. Por su parte, el filtro optimizado empieza la caída inmediatamente después de la frecuencia de paso, situada en 10 [rad/s] y cumple completamente con la atenuación. Por otro lado, la Figura 28 muestra el comportamiento de los filtros en el dominio del tiempo, y es aquí donde se ve la superioridad del filtro optimizado frente al Butterworth, pues este último tiene un sobrepico de aproximadamente 4% que el filtro optimizado no tiene; es decir, el filtro optimizado está sobreamortiguado; asimismo se observa que el filtro obtenido con el algoritmo del murciélago virtual responde más rápido.

Figura 27. Respuesta en frecuencia para los filtros del ejemplo 7.3.1

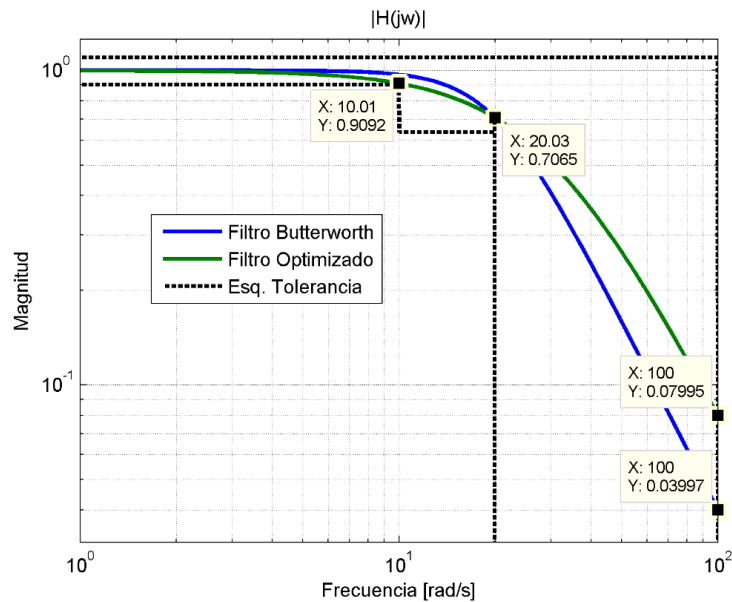
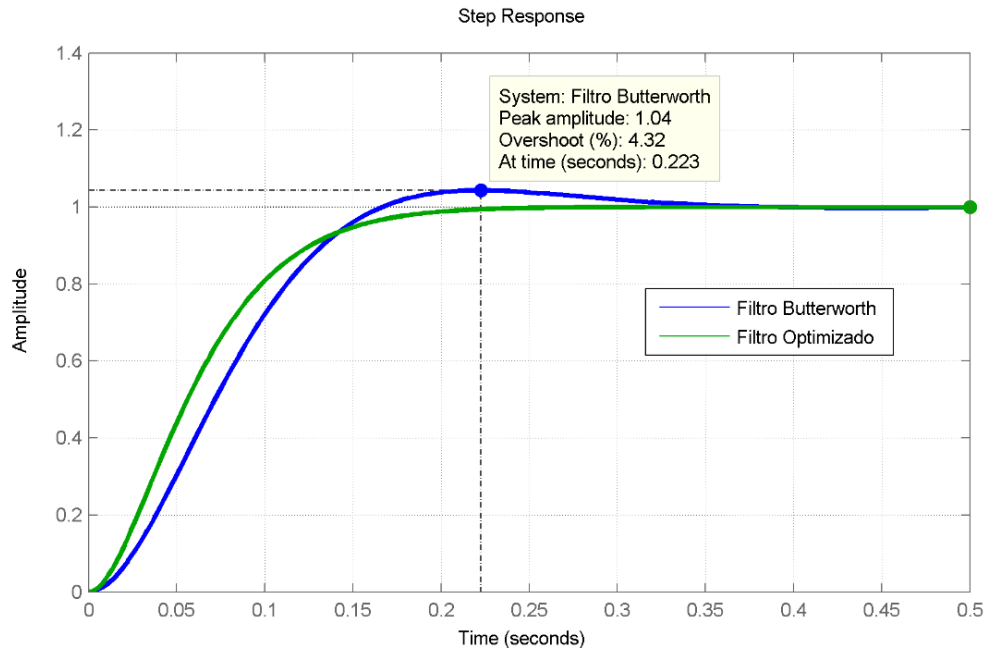


Figura 28. Respuesta al escalón para los filtros del ejemplo 7.3.1



7.3.2. FILTROS DE CUARTO ORDEN

Ahora, se desea verificar el correcto funcionamiento del algoritmo mediante un filtro de cuarto orden. Para este caso se toman las mismas especificaciones de la Tabla 21, pero se cambia la amplitud máxima en la banda de supresión a $A_s = 0.001$, en vista de que el orden superior permite una atenuación mucho mayor. Los polos de filtro Butterworth y de su correspondiente optimizado se encuentran en la Tabla 23.

Tabla 23. Polos obtenidos para los filtros del ejemplo 7.3.2

	a_1	jb_1	a_2	jb_2
Filtro butterworth	7.6537	18.4776	18.4776	7.6537
Filtro optimizado	2.9964	25.7766	8.3133	8.0285

La respuesta en frecuencia para la magnitud de los filtros se muestra en la Figura 29 y la respuesta en el dominio del tiempo se muestra en la Figura 30. En este caso el sobrepico para el filtro Butterworth fue de 10.8% mientras que para el filtro optimizado fue de 8.29%. Adicionalmente, el filtro optimizado cumple completamente todas las restricciones de frecuencia, mientras que el Butterworth no cumple la atenuación. Este hecho se debe a la posibilidad del filtro para aprovechar el esquema de tolerancia, ya que no se requiere de un filtro máximamente plano. En este caso la respuesta del filtro tiene casi la misma velocidad para ambos métodos de diseño.

Figura 29. Respuesta en magnitud para filtros del ejemplo 7.3.2

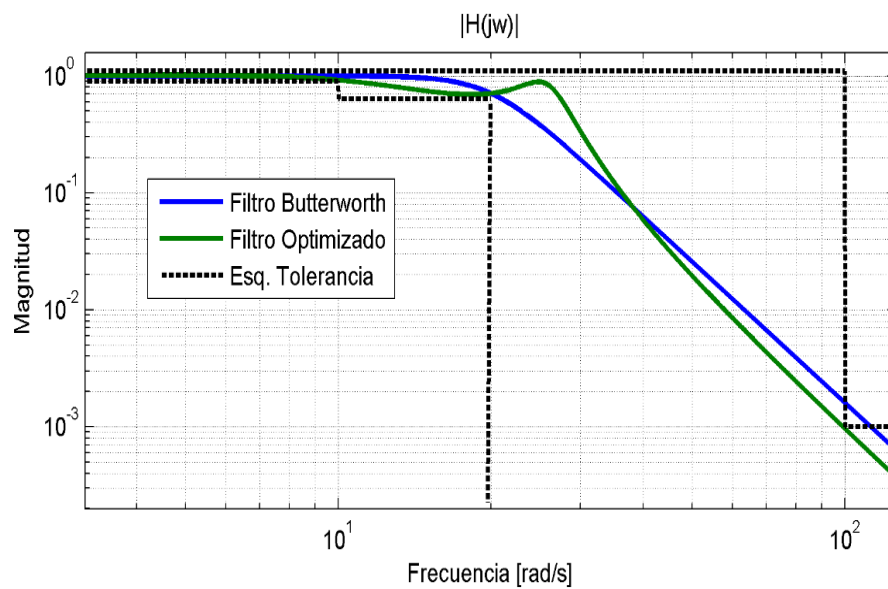
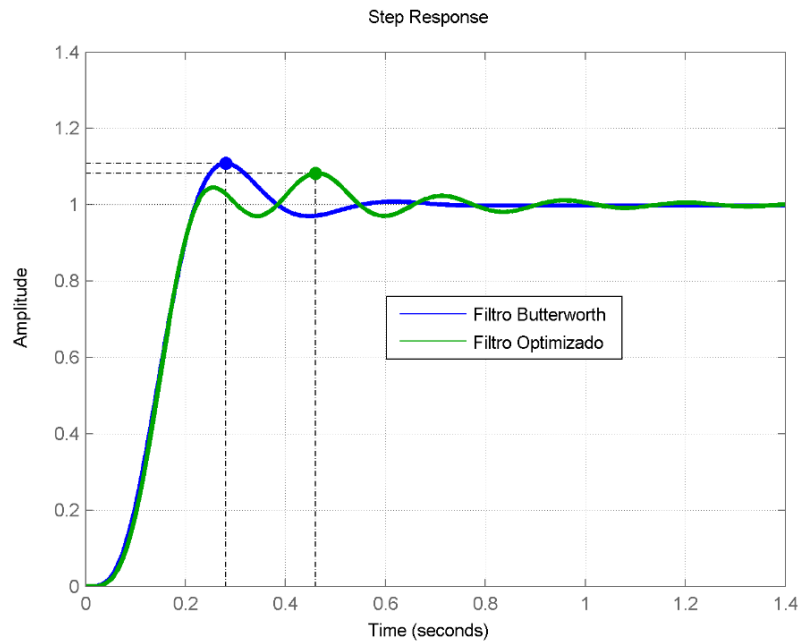


Figura 30. Respuesta al escalón para filtros del ejemplo 7.3.2



7.3.3. FILTROS DE OCTAVO ORDEN

Ya que el filtro de cuarto orden obtenido a partir del algoritmo del murciélago presenta una respuesta al escalón semejante a la del filtro Butterworth, en cuanto a velocidad de respuesta, se pretende observar si el aumento del orden del filtro afecta este tipo de respuesta o no. Para esto se diseña un filtro de orden 8 que cumpla con el esquema de tolerancia que se muestra en negro en la Figura 23 y que además tenga un sobrepico mínimo. El filtro Butterworth de octavo orden tiene los coeficientes del denominador que se muestran en la Tabla 24, donde también se muestran los coeficientes del filtro optimizado. Aunque la Figura 31 muestra que el comportamiento de los filtros es muy similar en el dominio de la frecuencia, la ventaja se aprecia en la Figura 32, donde se observa que en dominio del tiempo el filtro disminuye el sobrepico de 16.3% a 8.23% y a su vez la velocidad de respuesta es mejor en comparación a la del filtro Butterworth. Es así como se

puede verificar la mejora que aporta el algoritmo para filtros de órdenes superiores, en este caso octavo orden.

Tabla 24. Coeficientes del denominador para filtros del ejemplo 7.3.3

	Coeficientes a_n								
Filtro Butterworth	1.00	5.13	13.14	21.85	25.69	21.85	13.14	5.13	1.00
Filtro optimizado	1.00	3.34	10.33	17.92	24.58	23.23	15.68	7.07	1.44

Figura 31. Respuesta en magnitud para filtros del ejemplo 7.3.3

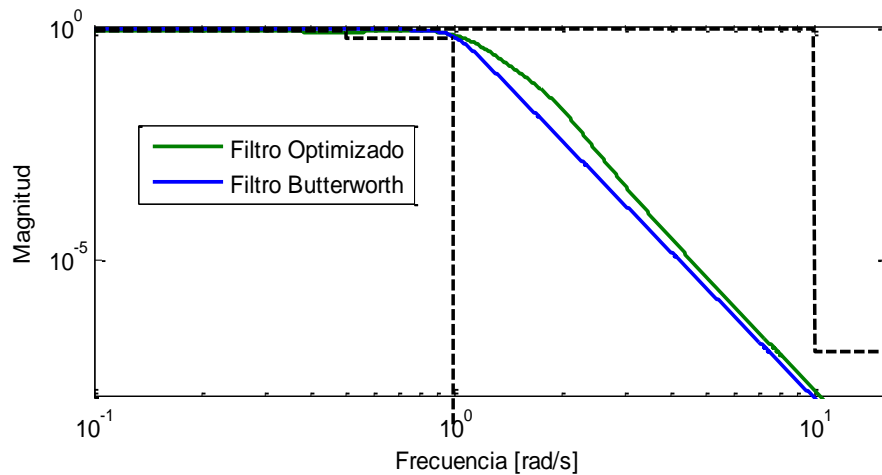
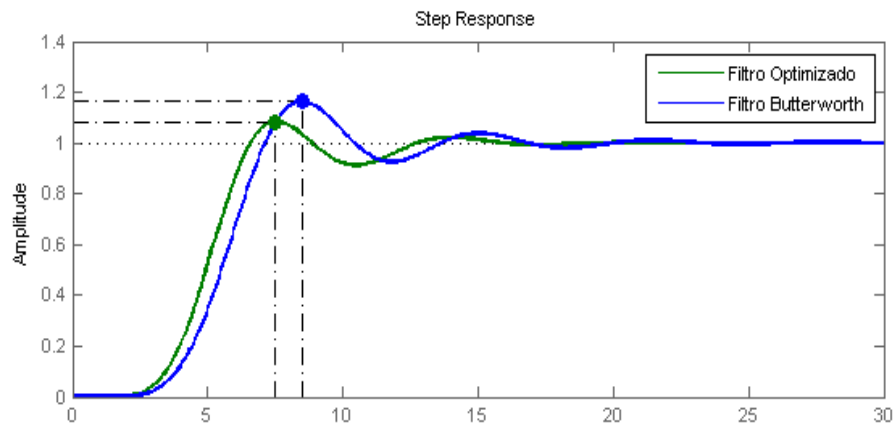


Figura 32. Respuesta al escalón para filtros del ejemplo 7.3.3



7.3.4. FILTRO CON FASE MÁXIMAMENTE LINEAL

En este ejemplo se desea optimizar un filtro de cuarto orden y cuya fase sea lo más lineal posible, para lo cual se utiliza un filtro de cuarto orden que cumpla con las especificaciones de la Tabla 21. Para lograr una fase lineal en la banda de paso, se utilizan restricciones al retardo de grupo. El filtro optimizado se compara con un filtro Butterworth y con un filtro que no esté restringido en el retardo de grupo, todos de orden 4, y los resultados de los polos obtenidos se muestran en la Tabla 25.

Tabla 25. Polos para los filtros del ejemplo 4

	a_1	jb_1	a_2	jb_2
Filtro Butterworth	7.6537	18.4776	7.6537	18.4776
Filtro optimizado	15.8098	25.8653	25.1611	4.4949
Filtro No optimizado	15.8684	12.4619	12.4917	20.8077

La respuesta en magnitud de los filtros se muestra en la Figura 33, y se observa que todos cumplen con la banda de supresión (10^{-2}) y con la banda de paso (10rad/s). Adicionalmente, el filtro Butterworth tiene la pendiente más pronunciada. En la Figura 34 se encuentra la fase correspondiente a la banda de paso de los filtros, donde se ve que la del filtro optimizado es más lineal y por tanto tiene un menor retardo de grupo (Figura 35).

Figura 33. Respuesta en magnitud para el ejemplo 4

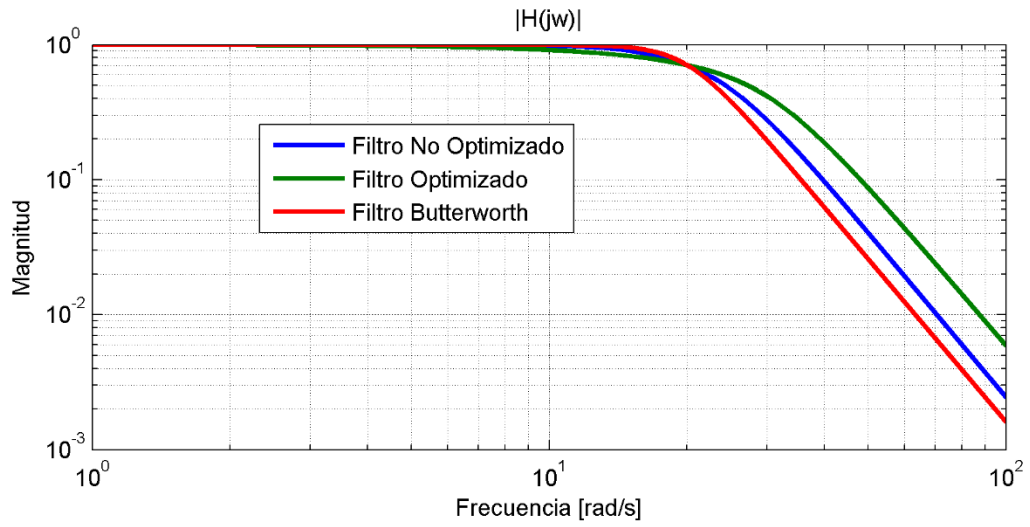


Figura 34. Respuesta en fase para los filtros del ejemplo 4

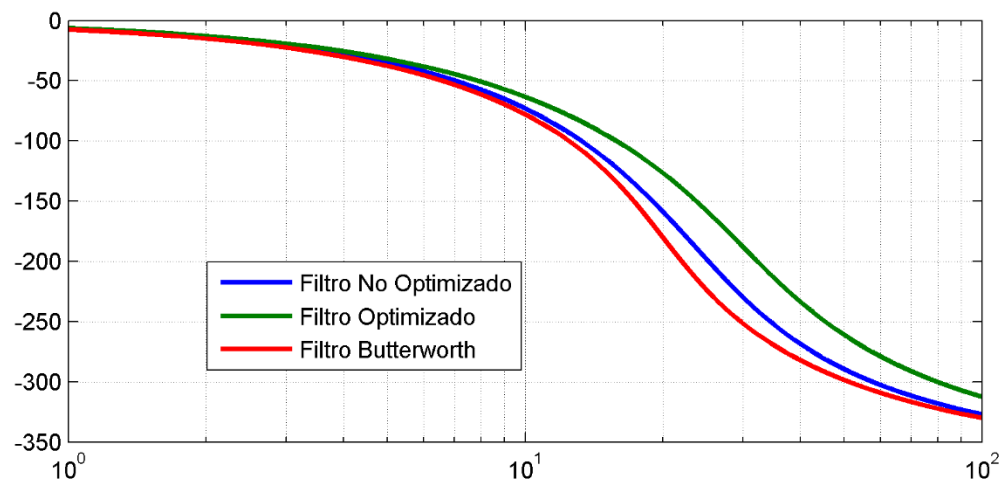
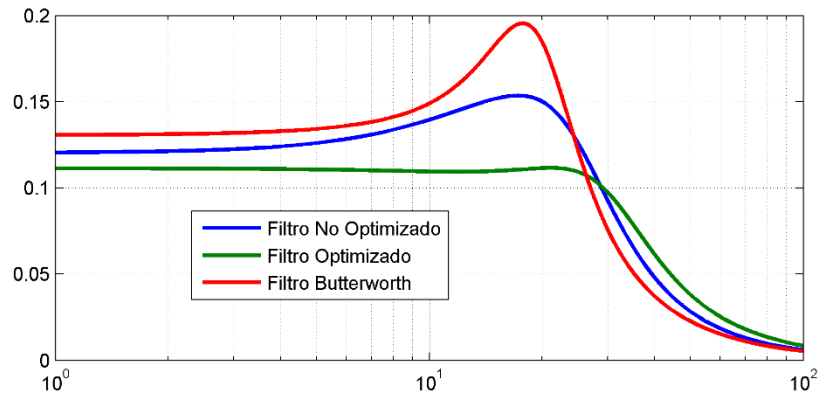


Figura 35. Retardo de grupo para los filtros del ejemplo 4



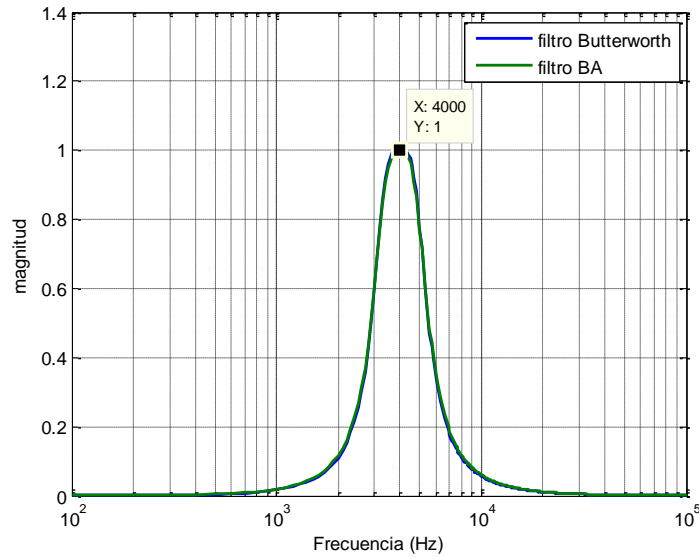
7.4. TRANSFORMACIÓN A FILTRO PASABANDA

En este ejemplo se compara la transformación de los filtros pasabajas expuestos en los numerales 7.1 y 7.3.3 al prototipo pasabanda. El objetivo para el primer caso simplemente es corroborar que la transformación del filtro es posible. En el segundo caso, se analiza si al hacer la conversión de filtros con orden superior, se afecta la respuesta en magnitud de los mismos. También se busca observar si el filtro obtenido al hacer la conversión sigue manteniendo las ventajas observadas en fase y en el dominio del tiempo del filtro pasabajas.

La transformación del filtro se realiza mediante el comando *lp2bp* de MATLAB®. Se asigna la frecuencia central y el ancho de banda deseados para el filtro pasabanda, que para el presente ejemplo serán de 4kHz y 2KHz, respectivamente.

La respuesta en magnitud para la transformación del filtro del numeral 7.1 se observa en la Figura 36, donde se valida que es posible realizar la transformación, y al igual que en la Figura 22 se encuentra bastante similitud respecto al filtro Butterworth.

Figura 36. Respuesta en magnitud para la transformación del filtro del ejemplo 7.1



En la Figura 37 se aprecia la respuesta en magnitud que arroja la transformación de los filtros desarrollados en el ejemplo 7.3.3. El filtro pasabanda resultante del algoritmo del murciélago presenta oscilación en la banda de paso en comparación con el filtro Butterworth transformado. Esto puede deberse, principalmente, al alto orden del filtro. Por otro lado, en la Figura 38 se observa que la fase de ambos filtros tiene un comportamiento equivalente, aunque tiende a ser un poco más lineal la obtenida por el algoritmo del murciélago. En la Figura 39 se observa que la respuesta en el dominio del tiempo es muy similar para ambos filtros.

Figura 37. Respuesta en magnitud para la transformación del filtro del ejemplo 7.3.3

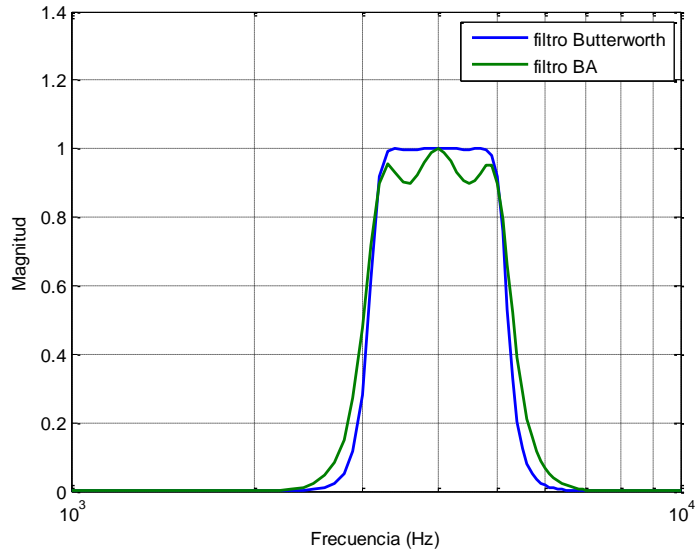


Figura 38. Respuesta en fase para la transformación del filtro del ejemplo 7.3.3

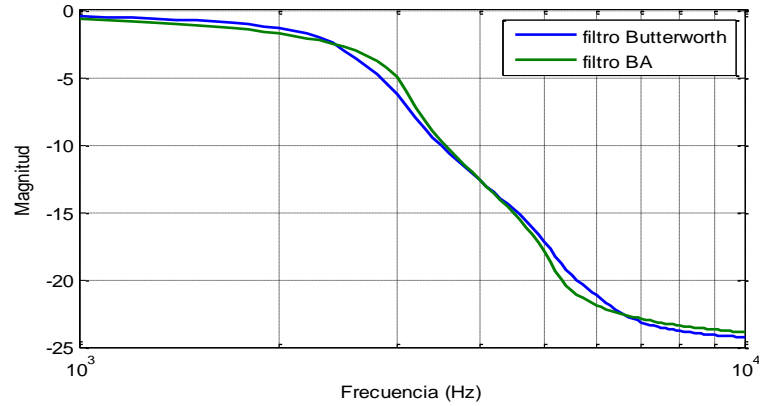
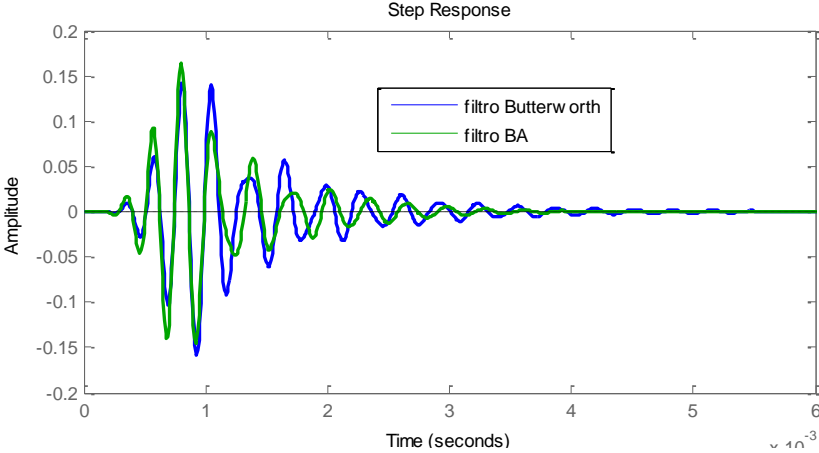


Figura 39. Respuesta al escalón para para la transformación del filtro del ejemplo 7.3.3



8. CONCLUSIONES

1. Se cumplieron satisfactoriamente todos los objetivos propuestos en este trabajo de investigación.
2. Se implementó el algoritmo del murciélago virtual y se propuso un marco de trabajo para el diseño óptimo de filtros, teniendo en cuenta la respuesta al escalón en el dominio del tiempo y la respuesta en magnitud y fase en el dominio de la frecuencia. Este marco de trabajo es fácil de implementar computacionalmente e incorpora todas las ventajas del algoritmo al ser comparado frente a otros métodos de diseño en el capítulo 7.
3. En la sección 4.2 se verificó el funcionamiento del algoritmo del murciélago virtual para diferentes funciones de prueba de varias dimensiones y altamente multimodales. Además, se realizó un análisis del comportamiento del algoritmo para un criterio de parada que permitió mejorar la convergencia del mismo, lo que permite utilizar el algoritmo en problemas de optimización en los cuales no se tenga un mínimo determinado con anterioridad.
4. El algoritmo permitió reducir el error de sobrepico en aproximadamente un 4% para un filtro de orden 2, en un 2.5% para orden 4 y en 8% para filtros de octavo orden sin violar ninguna restricción del esquema de tolerancia en el dominio de la frecuencia, y brindado una velocidad de respuesta igual o mejor a la de los filtros Butterworth en el dominio del tiempo (sección 7.2). Esto demuestra las ventajas de los filtros optimizados frente a los otros métodos de diseño, pues es el diseñador quien tiene control total sobre las especificaciones que desea obtener y no se ve obligado a encajar en alguno de los filtros clásicos.

5. En el capítulo 6 se realizó un análisis para diferentes combinaciones de los parámetros α y Y (ANEXO A: Presentación de resultados de las combinaciones para la selección de parámetros óptimos del algoritmo.), lo que permitió encontrar los valores que mejor se ajustan al problema de diseño de filtros. Cabe recordar que estos valores son netamente empíricos, por lo cual es muy importante realizar una correcta selección de los mismos para garantizar la convergencia del algoritmo.

6. Con los resultados obtenidos en el numeral 7.1 se demostró que el algoritmo efectivamente encontró los valores óptimos para el filtro. En el numeral 7.2 se comprobó la efectividad del algoritmo del murciélago virtual para el diseño de filtros pasabajas, en los cuales se satisfacen requisitos de diseño en el dominio de la frecuencia, evidenciando que el algoritmo cumple su funcionamiento para filtros de orden superior.

7. En la sección 7.4 se realizó satisfactoriamente el diseño de filtros pasabandas mediante la transformación de un filtro pasabajas encontrado a partir del algoritmo del murciélago, mostrando que a medida que el orden del filtro aumenta, algunas de las características apreciadas en el prototipo pasabajas pueden verse afectadas al hacer la transformación. Una posible forma de corregir esto, es agregando más restricciones al filtro diseñado. Por ejemplo, se puede agregar una restricción de banda plana para corregir la oscilación en la banda de paso en la transformación del filtro del numeral 7.3.3.

9. RECOMENDACIONES PARA TRABAJOS FUTUROS

1. Ya que en el trabajo presentado las constantes de peso se encuentran empíricamente, se recomienda realizar un número significativo de combinaciones, con el fin de encontrar los valores más óptimos para estas constantes y brindar la posibilidad de darle prioridad a alguna especificación de diseño del filtro.
2. Mejorar la penalización de las restricciones del esquema de tolerancia, y de esta manera ajustar el filtro en las bandas de paso y de corte, evitando sobrepicos en el dominio de la frecuencia.
3. Realizar investigaciones para encontrar el orden óptimo del filtro, pudiendo así comparar los distintos filtros clásicos de orden superior con filtros óptimos que además de cumplir restricciones en tiempo y frecuencia, también utilicen el mínimo orden posible.
4. Realizar estudios del espacio de búsqueda para el diseño de filtros, con el fin de minimizarlo y mejorar la eficiencia del algoritmo. Además esto permitiría conocer las ventajas de trabajar con polos cercanos al origen.
5. Indagar acerca de la implementación de los filtros aquí diseñados, de manera que estos puedan ser fabricados con componentes óptimos.
6. Al no encontrar en la literatura reportes con los cuales fuera posible comparar el desempeño de los filtros diseñados con el algoritmo del murciélago virtual contra estrategias como PSO, Harmony Search o Algoritmos Genéticos bajo el mismo esquema de tolerancia, se propone como trabajo futuro realizar la implementación del diseño de filtros con

alguno de estos otros algoritmos. Esto con el fin de estudiar su comportamiento y eficacia.

10. REFERENCIAS BIBLIOGRÁFICAS

- [1] A. AMBARDAR, *Procesamiento de señales analógicas y digitales*, Segunda Ed. Ciudad de México: Thomson Learning, 1999, p. 810.
- [2] V. AGGARWAL, W. O. JIN, and U.-M. O'REILLY, "Filter approximation using explicit time and frequency domain specifications," *Proc. 8th Annu. Conf. Genet. Evol. Comput. - GECCO '06*, p. 753, 2006.
- [3] Circuitstoday, "Types of Active Filters Butterworth: Chebyshev, Bessel and Elliptic filters.," 2009. [Online]. Available: <http://www.circuitstoday.com/active-filter-types>.
- [4] O. KOCA, H. KARL, and R. WEIGEL, "A Novel Method Based Upon Nonlinear Optimization for Analog Filter Design with Mask Constraints," *Signals, Syst. Electron. ...*, vol. 0, no. 2, pp. 9–12, 2007.
- [5] X.-S. YANG, "A new metaheuristic bat-inspired algorithm," *Nat. inspired Coop. Strateg. Optim. (NICSO 2010)*, pp. 65–74, 2010.
- [6] D. ROCHE, D. GIL, and J. GIRALDO, "Study on Evolutionary Computation Stopping Criteria," 2010.
- [7] M. MOLGA and C. SMUTNICKI, "Test functions for optimization needs," no. c, pp. 1–43, 2005.
- [8] X.-S. YANG, *Engineering Optimization*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2010.
- [9] J. A. PORTILLA GARCIA, "Solución de las Ecuaciones que Modelan un Circuito No Lineal De Corriente Directa Mediante el Método de la Espiral," Universidad Industrial de Santander, 2012.
- [10] K. OGATA, *Ingeniería De Control Moderna*, Tercera Ed. Pearson, 1998, p. 1015.
- [11] N. DAMERA-VENKATA, S. MEMBER, B. L. EVANS, and S. MEMBER, "An Automated Framework for Multicriteria Optimization of Analog Filter Designs," vol. 46, no. 8, pp. 981–990, 1999.
- [12] S. WINDER, *Analog and Digital Filter Design*, Segunda Ed. Newnes, 2002.

11. BIBLIOGRAFÍA

- A. AMBARDAR, *Procesamiento de señales analógicas y digitales*, Segunda Ed. Ciudad de México: Thomson Learning, 1999, p. 810.
- Circuitstoday, "Types of Active Filters Butterworth: Chebyshev, Bessel and Elliptic filters.," 2009. [Online]. Available: <http://www.circuitstoday.com/active-filter-types>.
- D. ROCHE, D. GIL, and J. GIRALDO, "Study on Evolutionary Computation Stopping Criteria," 2010.
- J. A. PORTILLA GARCIA, "Solución de las Ecuaciones que Modelan un Circuito No Lineal De Corriente Directa Mediante el Método de la Espiral," Universidad Industrial de Santander, 2012.
- K. OGATA, *Ingeniería De Control Moderna*, Tercera Ed. Pearson, 1998, p. 1015.
- M. MOLGA and C. SMUTNICKI, "Test functions for optimization needs," no. c, pp. 1–43, 2005.
- N. DAMERA-VENKATA, S. MEMBER, B. L. EVANS, and S. MEMBER, "An Automated Framework for Multicriteria Optimization of Analog Filter Designs," vol. 46, no. 8, pp. 981–990, 1999.
- O. KOCA, H. KARL, and R. WEIGEL, "A Novel Method Based Upon Nonlinear Optimization for Analog Filter Design with Mask Constraints," *Signals, Syst. Electron.*, vol. 0, no. 2, pp. 9–12, 2007.
- S. WINDER, *Analog and Digital Filter Design*, Segunda Ed. Newnes, 2002.
- V. AGGARWAL, W. O. JIN, and U.-M. O'REILLY, "Filter approximation using explicit time and frequency domain specifications," *Proc. 8th Annu. Conf. Genet. Evol. Comput. - GECCO '06*, p. 753, 2006.
- X.-S. YANG, "A new metaheuristic bat-inspired algorithm," *Nat. inspired Coop. Strateg. Optim. (NICSO 2010)*, pp. 65–74, 2010.
- X.-S. YANG, *Engineering Optimization*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2010.

12. ANEXOS

ANEXO A: Presentación de resultados de las combinaciones para la selección de parámetros óptimos del algoritmo.

Tabla 26. Combinaciones para $\alpha=0.6$

	$\alpha=0.6$ y $\gamma=0.7$										
	1	2	3	4	5	6	7	8	9	10	Promedio
Error_LS_ws	0.04385	0.04393	0.02794	0.00000	0.00000	0.01187	0.00000	0.04373	0.00000	0.00000	0.01713
Error_LS_wf	0.06270	0.06279	0.02661	0.01623	0.01391	0.00034	0.00081	0.06274	0.01920	0.00915	0.02745
Error_LI	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
Msp	0.60253	0.60238	0.60525	1.00000	0.99982	0.65583	0.80710	0.60268	1.00000	0.99517	0.78708
fdex	0.97591	0.97589	1.00467	0.01274	0.01166	0.00897	0.01821	0.97584	0.01247	0.00854	0.40049
msp-1	0.39747	0.39762	0.39475	0.00000	0.00018	0.34417	0.19290	0.39732	0.00000	0.00483	0.21292
Suma	1.47993	1.48023	1.45396	0.02898	0.02575	0.36534	0.21193	1.47963	0.03168	0.02252	0.65799

	$\alpha=0.6$ y $\gamma=0.9$										
	1	2	3	4	5	6	7	8	9	10	Promedio
Error_LS_ws	0.00000	0.00000	0.03086	0.04379	0.04379	0.00000	0.00000	0.00000	0.10712	0.00000	0.02256
Error_LS_wf	0.01923	0.00200	0.03878	0.06272	0.06270	0.01721	0.01665	0.00966	0.00019	0.00101	0.02301

Error_LI	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
Msp	1.00000	0.90876	0.61310	0.60260	0.60261	1.00000	1.00000	0.99627	0.51051	0.83567	0.80695
fdex	0.01507	0.02055	0.98925	0.97588	0.97589	0.01119	0.01447	0.01189	0.01055	0.01999	0.30447
msp-1	0.00000	0.09124	0.38690	0.39740	0.39739	0.00000	0.00000	0.00373	0.48949	0.16433	0.19305
Suma	0.03430	0.11379	1.44579	1.47979	1.47977	0.02840	0.03112	0.02528	0.60734	0.18533	0.54309

$\alpha=0.6$ y $\Upsilon=1.1$											
	1	2	3	4	5	6	7	8	9	10	Promedio
Error_LS_ws	0.04383	0.00000	0.00000	0.00000	0.00000	0.00000	0.08306	0.04202	0.00000	0.00000	0.01689
Error_LS_wf	0.06268	0.01194	0.01273	0.03316	0.00078	0.01597	0.00020	0.06379	0.01657	0.01553	0.02333
Error_LI	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
Msp	0.60256	0.99902	0.99945	0.99977	0.80129	0.99999	0.53337	0.60463	1.00000	0.99998	0.85401
fdex	0.97592	0.01320	0.00869	0.64945	0.03046	0.01338	0.03703	0.97461	0.01125	0.00895	0.27229
msp-1	0.39744	0.00098	0.00055	0.00023	0.19871	0.00001	0.46663	0.39537	0.00000	0.00002	0.14599
Suma	1.47987	0.02612	0.02196	0.68284	0.22994	0.02936	0.58693	1.47579	0.02782	0.02450	0.45851

$\alpha=0.6$ y $\Upsilon=1.3$											
	1	2	3	4	5	6	7	8	9	10	Promedio
Error_LS_ws	0.04652	0.04377	0.00064	0.01991	0.04379	0.04370	0.00000	0.00000	0.00000	0.02795	0.02263
Error_LS_wf	0.00024	0.06270	0.00049	0.00000	0.06272	0.06275	0.01829	0.01453	0.01157	0.02661	0.02599
Error_LI	0.00000	0.00000	0.00000	0.32382	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.03238

Msp	0.57945	0.60264	0.72659	0.00000	0.60260	0.60271	1.00000	0.99991	0.99874	0.60524	0.67179
fdex	0.02313	0.97588	0.01883	0.85597	0.97588	0.97583	0.01385	0.01242	0.01659	1.00467	0.48730
msp-1	0.42055	0.39736	0.27341	1.00000	0.39740	0.39729	0.00000	0.00009	0.00126	0.39476	0.32821
Suma	0.49044	1.47972	0.29337	2.19970	1.47979	1.47956	0.03214	0.02704	0.02942	1.45399	0.89652

Tabla 27. Combinaciones para $\alpha=0.8$

$\alpha=0.8$ y $\Upsilon=0.7$											
	1	2	3	4	5	6	7	8	9	10	Promedio
Error_LS_ws	0.04421	0.00000	0.00000	0.04466	0.00000	0.00000	0.03051	0.00000	0.03040	0.03080	0.01806
Error_LS_wf	0.06303	0.01647	0.01669	0.06359	0.01390	0.01652	0.03881	0.01781	0.02019	0.03882	0.03058
Error_Ll	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
Msp	0.60189	1.00000	1.00000	0.60102	0.99981	1.00000	0.61365	1.00000	0.60050	0.61320	0.80301
fdex	0.97587	0.01861	0.01824	0.97574	0.01504	0.02453	0.98903	0.03026	1.01640	0.98918	0.50529
msp-1	0.39811	0.00000	0.00000	0.39898	0.00019	0.00000	0.38635	0.00000	0.39950	0.38680	0.19699
Suma	1.48122	0.03509	0.03494	1.48298	0.02913	0.04105	1.44471	0.04806	1.46650	1.44560	0.75093

$\alpha=0.8$ y $\Upsilon=0.9$											
	1	2	3	4	5	6	7	8	9	10	Promedio
Error_LS_ws	0.01605	0.00000	0.00000	0.00000	0.04369	0.00000	0.00000	0.00000	0.00016	0.00000	0.00599
Error_LS_wf	0.01180	0.01021	0.01814	0.01565	0.06265	0.01808	0.01285	0.01450	0.00052	0.01924	0.01837

Error_LI	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
Msp	0.64356	0.99718	1.00000	0.99998	0.60277	1.00000	0.99951	0.99991	0.73857	1.00000	0.89815
fdex	1.00126	0.03283	0.01606	0.01448	0.97588	0.01301	0.02415	0.01867	0.03330	0.01892	0.21485
msp-1	0.35644	0.00282	0.00000	0.00002	0.39723	0.00000	0.00049	0.00009	0.26143	0.00000	0.10185
Suma	1.38555	0.04586	0.03420	0.03015	1.47944	0.03109	0.03749	0.03326	0.29541	0.03817	0.34106

$\alpha=0.8$ y $\gamma=1.1$											
	1	2	3	4	5	6	7	8	9	10	Promedio
Error_LS_ws	0.04328	0.04342	0.00670	0.00000	0.04382	0.00000	0.00000	0.04380	0.00000	0.00000	0.01810
Error_LS_wf	0.06280	0.06230	0.00038	0.01726	0.06266	0.01301	0.01873	0.06297	0.01802	0.01484	0.03330
Error_LI	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
Msp	0.60328	0.60330	0.67753	1.00000	0.60259	0.99957	1.00000	0.60247	1.00000	0.99994	0.80887
fdex	0.97564	0.97597	0.03664	0.02176	0.97592	0.00600	0.02697	0.97574	0.02502	0.02387	0.40435
msp-1	0.39672	0.39670	0.32247	0.00000	0.39741	0.00043	0.00000	0.39753	0.00000	0.00006	0.19113
Suma	1.47844	1.47840	0.36619	0.03902	1.47981	0.01944	0.04571	1.48005	0.04304	0.03877	0.64688

$\alpha=0.8$ y $\gamma=1.3$											
	1	2	3	4	5	6	7	8	9	10	Promedio
Error_LS_ws	0.03084	0.00000	0.00000	0.00000	0.00000	0.04081	0.00000	0.04343	0.00000	0.00000	0.01151
Error_LS_wf	0.03878	0.01659	0.01699	0.01630	0.01253	0.06697	0.00579	0.06269	0.01650	0.01634	0.02695
Error_LI	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

Msp	0.61312	1.00000	1.00000	1.00000	0.99936	0.60524	0.98039	0.60311	1.00000	1.00000	0.88012
fdex	0.98925	0.01295	0.02001	0.02148	0.01588	0.97250	0.03658	0.97576	0.02368	0.02372	0.30918
msp-1	0.38688	0.00000	0.00000	0.00000	0.00064	0.39476	0.01961	0.39689	0.00000	0.00000	0.11988
Suma	1.44575	0.02954	0.03700	0.03779	0.02905	1.47505	0.06197	1.47877	0.04018	0.04006	0.46752

Tabla 28. Combinaciones para $\alpha=0.9$

$\alpha=0.9$ y $\Upsilon=0.7$											
	1	2	3	4	5	6	7	8	9	10	Promedio
Error_LS_ws	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.04256	0.00000	0.00000	0.00426
Error_LS_wf	0.01921	0.01499	0.01385	0.01923	0.01911	0.01824	0.01242	0.06275	0.01507	0.01924	0.02141
Error_Ll	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
Msp	1.00000	0.99995	0.99980	1.00000	1.00000	1.00000	0.99931	0.60431	0.99996	1.00000	0.96033
fdex	0.03623	0.02230	0.01543	0.02187	0.10409	0.02649	0.01680	0.97538	0.02752	0.02938	0.12755
msp-1	0.00000	0.00005	0.00020	0.00000	0.00000	0.00000	0.00069	0.39569	0.00004	0.00000	0.03967
Suma	0.05544	0.03734	0.02947	0.04110	0.12320	0.04473	0.02991	1.47639	0.04264	0.04862	0.19288

$\alpha=0.9$ y $\Upsilon=0.9$											
	1	2	3	4	5	6	7	8	9	10	Promedio
Error_LS_ws	0.04359	0.04403	0.04338	0.04434	0.00000	0.00000	0.00000	0.00000			0.02192
Error_LS_wf	0.06259	0.01924	0.06200	0.06258	0.06300	0.06214	0.01403	0.01918	0.01735	0.00234	0.03845

Error_LI	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
Msp	0.60257	1.00000	0.60322	0.60233	0.60305	0.60211	0.99984	1.00000	1.00000	0.92278	0.79359
fdex	0.97597	0.03449	0.97620	0.97605	0.97557	0.97641	0.01823	0.03653	0.02003	0.03762	0.50271
msp-1	0.39743	0.00000	0.39678	0.39767	0.39695	0.39789	0.00016	0.00000	0.00000	0.07722	0.20641
Suma	1.47959	0.09777	1.47837	1.48064	1.43552	1.43644	0.03242	0.05571	0.03738	0.11718	0.76510

$\alpha=0.9$ y $\gamma=1.1$											
	1	2	3	4	5	6	7	8	9	10	Promedio
Error_LS_ws	0.00000	0.00000	0.04362	0.00000	0.00000	0.04397	0.04433	0.00000			0.01649
Error_LS_wf	0.00877	0.06276	0.01886	0.01837	0.06249	0.01370	0.01872	0.06275	0.06307	0.01772	0.03472
Error_LI	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
Msp	0.99427	0.60077	1.00000	1.00000	0.60294	0.99977	1.00000	0.60234	0.60170	1.00000	0.84018
fdex	0.03024	0.97636	0.02049	0.02347	0.97594	0.03111	0.02158	0.97593	0.97590	0.02030	0.40513
msp-1	0.00573	0.39923	0.00000	0.00000	0.39706	0.00023	0.00000	0.39766	0.39830	0.00000	0.15982
Suma	0.04474	1.43835	0.08298	0.04185	1.43549	0.08900	0.08463	1.43634	1.43727	0.03802	0.61287

$\alpha=0.9$ y $\gamma=1.3$											
	1	2	3	4	5	6	7	8	9	10	Promedio
Error_LS_ws	0.00000	0.00000	0.04359	0.00084	0.02774	0.00000	0.00000	0.00000	0.00000	0.00000	0.00722
Error_LS_wf	0.01681	0.06278	0.00048	0.02683	0.01329	0.01846	0.01513	0.01865	0.01714		0.02106
Error_LI	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

Msp	0.99964	1.00000	0.60286	0.72327	0.60574	0.99967	1.00000	0.99996	1.00000	1.00000	0.89311
fdex	0.01732	0.02788	0.97577	0.04253	1.00416	0.02521	0.02393	0.03392	0.02856	0.02029	0.21996
m _{sp} -1	0.00036	0.00000	0.39714	0.27673	0.39426	0.00033	0.00000	0.00004	0.00000	0.00000	0.10689
Suma	0.03449	0.09066	1.41697	0.34693	1.43945	0.04401	0.03906	0.05261	0.04570	0.02029	0.35302

ANEXO B: Características de los filtros clásicos.

FILTROS BUTTERWORTH

De acuerdo con [1] los filtros Butterworth también se conocen como máximamente planos, esto debido a las características matemáticas que tiene el polinomio correspondiente a este filtro que garantiza la respuesta más plana y la menor desviación con respecto a la ganancia en la banda de paso. Si bien se obtiene una respuesta en frecuencia monótona durante la banda de paso y libre de sobrepicos, es a costa de una caída lenta hacia la banda de supresión.

La función de transferencia del filtro butterworth está representada por la ecuación (32):

$$|H(s)| = \sqrt{\frac{Kpb}{1 + \left(\frac{s}{\omega_c}\right)^{2n}}} \tag{32}$$

Donde ω_c es la frecuencia de corte ($\omega \approx -3\text{dB}$) y n hace referencia al orden del filtro ($n=1, 2, 3, \dots$).

Para el caso en el que se requiere diseñar un filtro de acuerdo a las especificaciones de la Figura 1, es decir, los valores de A_{max} , A_{min} , ω_p y ω_s y por ahora excluyendo la frecuencia de corte o 3 dB, se tiene que encontrar el orden del filtro Butterworth que permita cumplir con estos requisitos de diseño, para esto se tiene(33) y para el caso en que $\omega_p = \omega_{3dB}$ la ecuación (34) la cual es una aproximación.

$$n = \frac{\log_{10} \frac{\sqrt{10^{-0.1 * A_{min}} - 1}}{\sqrt{10^{-0.1 * A_{max}} - 1}}}{\log_{10} \frac{\omega_s}{\omega_p}} \quad (33)$$

$$n = \frac{-0.05 * A_{min}}{\log_{10} \frac{\omega_s}{\omega_p}} \quad (34)$$

Una vez se determina el orden del filtro, se pueden utilizar tablas para determinar la función de transferencia del filtro a una frecuencia normalizada y posteriormente realizar un corrimiento en frecuencia.

- FILTRO DE CHEBYSHEV

A diferencia del filtro Butterworth, el Chebyshev presenta rizado en la banda de paso pero un incremento de la atenuación de la banda de rechazo. Al permitir este rizado en la banda de paso este filtro logra una caída más brusca luego de la frecuencia de corte [12]. Sin embargo, su retraso de grupo tiene un pico mucho mayor cerca de la banda de paso, en comparación al filtro Butterworth.

La función de transferencia del filtro Chebyshev está representada en la ecuación (35).

$$|H(s)| = \sqrt{\frac{1}{1 + \varepsilon^2 C_n^2\left(\frac{s}{\omega_c}\right)}} \quad (35)$$

Donde ω_c es la frecuencia de corte ($\omega \approx -3\text{dB}$), ε es la constante que determina la amplitud del rizado ($0 < \varepsilon \leq 1$) y C_n hace referencia al polinomio de Chebyshev de orden n ésimo. Este polinomio viene dado por (36).

$$C_{n+1}(\omega) = 2\omega C_n(\omega) - C_{n-1}(\omega) \quad (36)$$

Con $C_n(\omega)=1$ y $C_0(\omega)=\omega$

- FILTRO ELÍPTICO

Esta aproximación exhibe rizados en las dos bandas, tanto de paso como de supresión. También llamados filtros de Cauer, se utilizan cuando se necesita una transición muy rápida entre la banda de paso y la de atenuación. Cabe destacar que el filtro elíptico es quien presenta una mayor velocidad de transición, pero es también quien tiene el mayor rizado.

La Función de transferencia del filtro elíptico viene dada por (37).

$$|H(\omega)| = \sqrt{\frac{1}{1 + \varepsilon^2 R_n^2\left(\frac{\omega}{\omega_c}\right)}} \quad (37)$$

Donde ω_c es la frecuencia de corte ($\omega \approx -3\text{dB}$), ϵ es el factor de riple o constante de rizado ($0 < \epsilon \leq 1$) y R_n la función racional elíptica de orden n descrita por (38).

$$R_n(\omega) = \int_0^{2\pi} \frac{1}{1 - \omega^2 * \sin^2\theta} * d\theta \quad (38)$$

- FILTRO BESSEL

El filtro Bessel se utiliza cuando se necesita obtener una fase lineal o un retraso plano en la banda de paso. Por otro lado, la magnitud tiene una banda de paso plana, pero una caída hacia la banda de supresión bastante lenta y suave. Estas características permiten que el filtro obtenga un desempeño en el dominio del tiempo sobresaliente, pero no resulta muy eficiente para eliminar la frecuencias que no se desean.

La función de transferencia del filtro Bessel está representada por (39).

$$H(s) = \frac{\theta_n(0)}{\theta_n\left(\frac{s}{\omega_0}\right)} \quad (39)$$

Donde ω_c es la frecuencia de corte ($\omega \approx -3\text{dB}$) y $\theta_n(s)$ es el polinomio de Bessel inverso de orden n descrito por (40).

$$\sum_{k=0}^n a_k * s^k \quad (40)$$

Dónde:

$$a_k = \frac{(2n - k)!}{2^{n-k} * k! * (n - k)!}$$

Con $k=0,1,2,\dots,N$