

Aplicación del algoritmo Evolución Diferencial en un método de dimensionamiento para filtros bicuadráticos

Mónica Lovay, Gabriela Peretti, y Eduardo Romero

Grupo de Estudio en Calidad en Mecatrónica,
Facultad Regional Villa María, Universidad Tecnológica Nacional,
Villa María, Argentina
gecam@frvm.utn.edu.ar

Resumen. Este trabajo presenta un método alternativo para el diseño de filtros bicuadráticos. El mismo utiliza un algoritmo evolutivo (EA, Evolutionary Algorithm) para determinar los valores de los componentes pasivos que le proporcionan al filtro sensibilidades mínimas con respecto a las variaciones de éstos, presentando además errores en los parámetros de desempeño inferiores a los establecidos en las especificaciones. Para este fin, se adopta el algoritmo Evolución Diferencial (DE, Differential Evolution), el cual se caracteriza por ser simple, eficiente y veloz. Para comprobar la viabilidad del método de diseño formulado, se propone su aplicación para seleccionar los componentes pasivos de un filtro bicuadrático pasabajo IGMFB (Infinite-Gain Multiple Feedback). Los resultados muestran que DE permite obtener configuraciones de filtro con sensibilidades bajas y errores en los parámetros funcionales que satisfacen las especificaciones consideradas. Por otro lado, la comparación de resultados con los obtenidos utilizando otros EA muestra que DE emplea una cantidad menor de evaluaciones de la función fitness y tiempos de ejecución más bajos, además de proporcionar soluciones que presentan, en general, mejores valores de fitness.

Palabras clave: optimización por enjambre de partículas, particle swarm optimization, diseño de filtros activos, optimización multiobjetivo.

1 Introducción

Los filtros bicuadráticos son circuitos electrónicos que se encuentran presentes en numerosas aplicaciones de procesamiento de señales. Una de las principales ventajas de este tipo de filtros radica en que los mismos pueden ser conectados en cascada, permitiendo el diseño de filtros de orden superior.

El dimensionamiento de los componentes pasivos en filtros bicuadráticos es una tarea compleja. Este proceso no se limita solamente a seleccionar los valores de resistencias y capacitores que permiten al filtro cumplir con determinadas especificaciones, sino que además es necesario considerar que estos valores deben estar contenidos en alguna de las series industriales E. Estas series limitan los valores que pueden asumir los componentes.

El enfoque convencional de diseño propone asignar valores iguales a algunas resistencias o capacitores del filtro [1, 2]. Esta estrategia, si bien simplifica el proceso de diseño, limita la libertad del mismo.

Por otro lado, efectuar una búsqueda exhaustiva de todas las combinaciones de valores posibles para las resistencias y capacitores del filtro, eligiendo la que mejor se ajusta a las especificaciones dadas puede ser una alternativa. No obstante, este método no resulta factible en tiempos razonables debido a la gran cantidad de alternativas que se presentan.

Por este motivo, es necesario el empleo de métodos de búsqueda inteligente que permitan llevar a cabo el proceso de diseño en tiempos computacionales aceptables. En este contexto, las técnicas evolutivas se convierten en una interesante opción.

Los algoritmos evolutivos (EA, Evolutionary Algorithms) son algoritmos informáticos que imitan las fuerzas de la evolución natural y la autoadaptación [3, 4]. Su característica principal es precisamente la eficiencia de búsqueda en espacios con una cantidad de soluciones considerablemente grande.

Algunos autores han propuesto la utilización de EA para obtener los valores de los componentes pasivos que mejor se ajustan a las especificaciones de filtros bicuadráticos. Particularmente, en [5] se propone utilizar el algoritmo CSA (Clonal Selection Algorithm) para efectuar la selección de componentes de un filtro Butterworth pasabajo de cuarto orden. El método restringe el espacio de búsqueda a los valores disponibles en las series E12. Los resultados obtenidos se comparan con el método convencional, GA (Genetic Algorithms) y TS (Tabu Search), demostrando que el algoritmo propuesto permite alcanzar un error de diseño menor. En [6] se utiliza el algoritmo PSO (Particle Swarm Optimization) para efectuar el diseño de un filtro pasabajo de variable de estado de segundo orden, considerando que los componentes deben ser compatibles con las series E24 y E96. Los resultados se contrastan con el método convencional demostrando que el error de diseño para PSO es considerablemente menor.

Otros autores emplean los filtros propuestos como casos de estudio en [5, 6] con el propósito de validar el desempeño de varios EA. Particularmente, en [7] son evaluados tres algoritmos: GA, ABC (Artificial Bee Colony) y PSO. En [8] se explora la aplicabilidad de los algoritmos DE (Differential Evolution) y HS (Harmony Search) en el diseño de filtros. En [9] se propone una variante del algoritmo ABC, denominada CRbABC_Dt. Tres versiones mejoradas del algoritmo PSO: Simplex PSO, CRPSO y ALC-PSO son presentadas en [10-12]. En [13] es propuesta una nueva metaheurística, el algoritmo VS (Vortex Search).

En todos los trabajos referenciados el proceso de diseño propuesto está enfocado hacia la minimización de los errores de diseño respecto de las especificaciones, sin incorporar las sensibilidades en dicho proceso. Por otro lado, tres algoritmos [5,8,13] proporcionan soluciones donde los valores de los componentes son directamente compatibles con las series E consideradas. No obstante, en [6,7,9-12], las soluciones deben ser ajustadas para ser compatibles con las series E adoptadas. En estos casos, los autores proponen redondear el valor obtenido para algunos de los componentes al valor de la serie E más próximo, con lo cual se incrementa el error en las especificaciones del filtro. Los componentes restantes son obtenidos mediante la implementación en paralelo o serie de dos componentes estándar. Otro aspecto común a todos los trabajos arriba mencionados está relacionado con el valor de la ganancia del filtro en la banda

de paso, el cual en ningún caso se encuentra restringido. Esto significa que la ganancia no es incluida en el proceso de optimización y no puede ser fijada como objetivo de diseño. Se observa que en numerosas ocasiones se obtienen filtros con ganancias extremadamente bajas, inconvenientes en la práctica.

En [13, 14], los autores del presente trabajo proponen incorporar la sensibilidad como parte del proceso de optimización, con el propósito de obtener diseños más robustos y menos variables a cambios en los valores de los componentes pasivos. El método de dimensionamiento propuesto en estos trabajos emplea un EA que lleva a cabo un proceso de optimización cuyo propósito es minimizar las sensibilidades del filtro con respecto a desviaciones en los valores de los componentes pasivos, y cumplir con las especificaciones del mismo, en las cuales se asume que la ganancia en la banda de paso también se encuentra restringida. Por otro lado, en ambos casos el EA limita su espacio de búsqueda solamente a configuraciones de filtro cuyos valores de componentes pertenezcan a las series E consideradas. Esta propiedad implica que la solución arrojada por el método puede ser directamente implementada, no requiriendo de ningún tipo de ajuste de valores y/o de conexiones adicionales. Los EA utilizados en estos trabajos son GA [13] y PSO [14].

En este trabajo, se presenta una nueva alternativa para implementar el método de dimensionamiento propuesto, empleando otro EA, el algoritmo Evolución Diferencial (DE, Differential Evolution). Con el propósito de comprobar la viabilidad de utilizar DE como parte de este método de diseño, se propone su aplicación para seleccionar los componentes pasivos de un filtro bicuadrático pasabajo IGMFB (Infinite-Gain Multiple Feedback). Los resultados obtenidos aplicando DE son presentados en este trabajo y comparados con los alcanzados en trabajos previos usando GA y PSO.

2 Filtros bicuadráticos: características generales

Los filtros bicuadráticos, también referenciados como biquads, implementan funciones de transferencia de segundo orden. Particularmente, la expresión (1) muestra la función de transferencia de un filtro bicuadrático pasabajo. La misma se encuentra expresada en función de la ganancia en la banda de paso (G), la frecuencia de polo ($\omega_p = 2\pi f_p$) y el factor de calidad (Q_p), denominados en general especificaciones del filtro.

$$F(s) = \frac{G \omega_p^2}{s^2 + \left(\frac{\omega_p}{Q_p}\right)s + \omega_p^2} \quad (1)$$

Por otro lado, la sensibilidad pasiva de un filtro es una medida de la variación de la performance del mismo como resultado de cambios en los valores de sus componentes pasivos. Estas variaciones pueden ocurrir debido al envejecimiento de éstos, tolerancias de fabricación, condiciones ambientales (temperatura), entre otros factores [2]. Mientras menos sensible es un filtro a los cambios en sus componentes, más estables permanecen sus características y, por lo tanto, existen más probabilidades de que pueda permanecer dentro de sus especificaciones. Se considera que un filtro tiene baja sensibilidad cuando todas sus sensibilidades adoptan valores inferiores a la unidad [2].

En los filtros bicuadráticos, las sensibilidades de ωp y Qp son las más críticas. De manera general, si S_j representa la especificación ωp o Qp de un filtro cuyos componentes pasivos son x_1, x_2, \dots, x_i , entonces la sensibilidad de S_j con respecto a x_i , está definida por:

$$S_{x_i}^{S_j} = \frac{\% \text{ cambio en } S_j}{\% \text{ cambio en } x_i} = \frac{\partial S_j / S_j}{\partial x_i / x_i} \quad (2)$$

3 Método de diseño para filtros bicuadráticos basado en DE

3.1 Formulación del problema de diseño

La tarea de DE se focaliza en encontrar los valores de los componentes pasivos (correspondientes a determinadas series E) que minimicen las sensibilidades de ωp y Qp a las desviaciones en los mismos. Por otro lado, las soluciones deben presentar errores inferiores a un límite tolerable definido para las especificaciones G_F , ωp_F y Qp_F .

Las consideraciones enunciadas permiten derivar que DE debe resolver un problema de optimización multiobjetivo (POM), el cual puede ser representado mediante la siguiente expresión:

$$\begin{aligned} \min S(X_i) &= (S_{x_1}^{S_j}(X_i), S_{x_2}^{S_j}(X_i), \dots, S_{x_n}^{S_j}(X_i)), \\ \text{sujeto a: } Error_G(X_i) &\leq E_{maxG} \\ Error_{\omega p}(X_i) &\leq E_{max\omega p} \\ Error_{Qp}(X_i) &\leq E_{maxQp} \end{aligned} \quad (3)$$

En (3), X_i es una posible configuración de filtro determinada por los valores de p resistencias y q capacitores. $S_{x_1}^{S_j}(X_i)$ a $S_{x_n}^{S_j}(X_i)$ representan las sensibilidades que deben ser minimizadas. Por otro lado, $Error_G$, $Error_{\omega p}$ y $Error_{Qp}$ representan los errores en G , ωp y Qp , respectivamente, y se calculan teniendo en cuenta las expresiones (4) a (6). E_{maxG} , $E_{max\omega p}$ y E_{maxQp} definen el error máximo tolerable para cada una de las características del filtro.

$$Error_G(X_i) = |(G(X_i) - G_F) / G_F| \quad (4)$$

$$Error_{\omega p}(X_i) = |(\omega p(X_i) - \omega p_F) / \omega p_F| \quad (5)$$

$$Error_{Qp}(X_i) = |(Qp(X_i) - Qp_F) / Qp_F| \quad (6)$$

3.2 Evolución Diferencial (DE)

DE es una técnica metaheurística basada en poblaciones de vectores numéricos. Entre las ventajas más destacadas de este algoritmo pueden mencionarse la simplicidad, eficiencia, propiedad de búsqueda local y velocidad. El proceso seguido por DE para resolver un problema de optimización se caracteriza por iterar sobre una población de vectores para hacer evolucionar soluciones candidatas con respecto a una función de aptitud llamada fitness [15,16].

En el diagrama de flujo de la Fig. 1 se muestran las principales operaciones realizadas por el algoritmo DE implementado, el cual ha sido desarrollado en Matlab™.

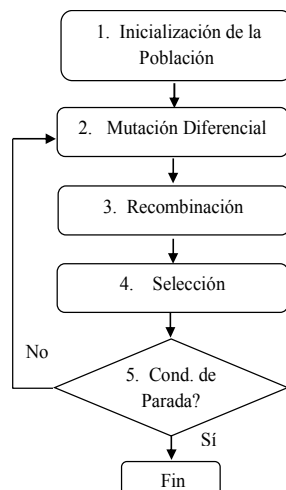


Fig. 1. Diagrama de flujo del algoritmo DE.

A continuación, se describe brevemente cada bloque del diagrama.

1. Inicialización de la Población. El algoritmo comienza generando de manera aleatoria una población inicial de NP vectores que son soluciones potenciales para el problema de optimización que se desea resolver. Particularmente, en este trabajo, cada vector representa una posible configuración de filtro caracterizada por los valores de sus p resistencias y q capacitores. El número de resistencias y capacitores permite determinar la cantidad de dimensiones ($D=p+q$) de los vectores con los que debe trabajar DE. En la expresión (7) se muestra la estructura de cada vector $X_{i,g}$.

$$X_{i,g} = [x_{1i,g}, x_{2i,g}, \dots, x_{Di,g}], \quad i = 1, 2, 3, \dots, NP \quad (7)$$

En (7), cada elemento $x_{ji,g}$ se corresponde con un valor de resistencia o capacitor y g indica la generación a la cual pertenece el vector.

2. Mutación Diferencial. Para cada vector $X_{i,g}$ (vector objetivo) en la generación g , se crea un vector mutado $V_{i,g}$, que se obtiene mediante perturbaciones de vectores que conforman la población actual. Para ello, el algoritmo emplea la estrategia de mutación representada en la expresión (8). De acuerdo a esta estrategia, cada vector mutado $V_{i,g} = [v_{1i,g}, v_{2i,g}, \dots, v_{Di,g}]$, se obtiene añadiendo a la diferencia proporcional de dos vectores ($X_{r1,g}, X_{r2,g}$) elegidos aleatoriamente, un tercer vector $X_{r3,g}$, también elegido aleatoriamente.

$$V_{i,g} = X_{r3,g} + F \cdot (X_{r1,g} - X_{r2,g}) \quad (8)$$

En (8), los índices $r1$, $r2$, $r3$ son enteros generados aleatoriamente en el rango $[1, NP]$, sin repetición. El factor de escala F controla la amplificación de la diferencia entre los individuos $X_{r1,g}$ y $X_{r2,g}$ y es usada para evitar el estancamiento en el proceso de búsqueda. Generalmente, F adopta valores en el rango $[0, 2]$ [15, 16].

3. *Recombinación.* Este proceso tiene como propósito agregar diversidad genética a la población. Para ello, el operador de recombinación genera vectores de prueba utilizando información que copia de dos vectores diferentes. En particular, DE cruza cada vector objetivo $X_{i,g}$ con su correspondiente vector mutado $V_{i,g}$ dependiendo de la probabilidad de recombinación de la población (Cr). En las expresiones (9) y (10) se muestran, respectivamente, el vector de prueba $U_{i,g}$ y el criterio empleado por DE para obtener cada elemento $u_{ji,g}$ del mismo.

$$U_{i,g} = [u_{1i,g}, u_{2i,g}, \dots, u_{Di,g}] \quad (9)$$

$$u_{ji,g} = \begin{cases} v_{ji,g} & \text{si } rand_j[0,1] \leq Cr \quad \text{ó } j = j_{rand} \\ x_{ji,g} & \text{en otro caso} \end{cases} \quad (10)$$

En (10), $j=1, 2, \dots, D$ y j_{rand} es un número entero aleatorio en el rango $[1, D]$ que permite garantizar que $U_{i,g}$ es diferente de $X_{i,g}$ en al menos un elemento. Por otro lado, Cr generalmente adopta valores en el rango $[0, 1]$ [15, 16].

4. *Selección.* Para conformar la siguiente generación ($g+1$), cada vector objetivo $X_{i,g}$ es comparado con su correspondiente vector de prueba $U_{i,g}$, usando una función de fitness f . El vector con el mejor valor de fitness es el que integrará la población de la siguiente generación. El proceso de selección descrito es representado en la expresión (11).

$$X_{i,g+1} = \begin{cases} U_{i,g} & \text{si } f(U_{i,g}) < f(X_{i,g}) \\ X_{i,g} & \text{de otra manera} \end{cases} \quad (11)$$

En este trabajo, la formulación de la función de fitness f es efectuada teniendo en cuenta el POM enunciado en (3). El mismo es convertido en un problema de optimización con un único objetivo, utilizando el método de suma ponderada, el cual emplea una función de agregación para efectuar la transformación [3, 4]. De esta manera, la función de fitness f que asigna un valor de aptitud a cada vector $X_{i,g}$, se define mediante la expresión (12).

$$f(X_{i,g}) = \sum_{k=1}^n w_k |S_{x_i}^{Sj}(X_{i,g})| \quad (12)$$

En (12), n representa la cantidad de sensibilidades $S_{x_i}^{Sj}$ que deben ser minimizadas. Se considera el valor absoluto de las mismas, con el propósito de evitar disminuciones en el valor de fitness ocasionadas por valores de sensibilidad negativos. w_k es el peso asignado a cada una de las sensibilidades. Teniendo en cuenta que $w_k \in [0, 1]$, que $\sum_{k=1}^n w_k = 1$ y que no existe prioridad entre las mismas, w_k adopta el mismo valor $1/n$ para todas las sensibilidades $S_{x_i}^{Sj}$. Por este motivo, f es reformulada de la siguiente manera:

$$f(X_{i,g}) = \frac{1}{n} \sum |S_{x_i}^{Sj}(X_{i,g})| \quad (13)$$

Con referencia a las restricciones expresadas en (3), DE las implementa mediante una estrategia de penalización. De acuerdo a esta estrategia [14], las restricciones son aplicadas penalizando en cada generación a aquellos vectores que no las cumplen, mediante la expresión (14). En consecuencia, éstas adoptan un valor de fitness más alto (f_i) de acuerdo al grado de violación de cada restricción.

$$f_h(X_{i,g}) = f(X_{i,g}) + n_1 \frac{|E_{maxG} - Error_{G,X_{i,g}}|}{Error_{G,max}} + n_2 \frac{|E_{max\omega p} - Error_{\omega p,X_{i,g}}|}{Error_{\omega p,max}} + n_3 \frac{|E_{maxQp} - Error_{Qp,X_{i,g}}|}{Error_{Qp,max}} \quad (14)$$

En (14), $Error_{G,X_{i,g}}$, $Error_{\omega p,X_{i,g}}$ y $Error_{Qp,X_{i,g}}$ representan, respectivamente, los errores en G , ωp y Qp , para el vector $X_{i,g}$. Por otro lado, $Error_{G,max}$, $Error_{\omega p,max}$ y $Error_{Qp,max}$ representan el máximo error alcanzado en la generación actual para G , ωp y Qp , respectivamente. n_1 , n_2 y n_3 son constantes agregadas por los autores para regular individualmente el impacto de las restricciones en el valor de fitness. La adopción de valores inferiores a la unidad para estas constantes implica una atenuación en el impacto de las mismas. Por el contrario, si se consideran valores superiores a la unidad se logrará un acrecentamiento de dicho impacto en los valores de fitness.

5. *Condición de parada:* La nueva generación atraviesa el proceso descrito desde la mutación hasta la selección. El ciclo se repite hasta que se satisface la condición de parada, que consiste en alcanzar un máximo número de generaciones (g_{max}).

4 Filtro bajo estudio

Con el propósito de comprobar la viabilidad del método de diseño formulado en este trabajo, se propone su aplicación para dimensionar los componentes pasivos de un filtro bicuadrático pasabajo IGMFB. En la Fig. 2 se muestra el circuito correspondiente a esta topología de filtro.

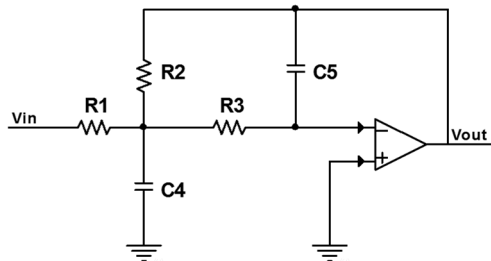


Fig. 2. Filtro IGMFB pasabajo.

En la Tabla 1 se presentan las expresiones que permiten obtener los valores de los parámetros funcionales del filtro a partir de los valores de los componentes pasivos. En la tabla también se muestran las ecuaciones correspondientes a las sensibilidades de estos parámetros que presentan dependencia con respecto a los componentes pasivos. Estas expresiones son las que permiten a DE realizar el cálculo del valor de fitness de cada vector, definido en la ecuación (14).

La especificación elegida para el filtro bicuadrático que se desea diseñar considera los siguientes valores para los parámetros funcionales: $G_F=3$, $\omega_{pF}=1000*2*\pi=6283,9478$ rad/s y $Q_{pF}=0,707$.

Con respecto a los valores que los componentes pasivos pueden asumir, se proponen dos escenarios. El escenario E1 considera que las resistencias y capacitores pueden

adoptar valores de acuerdo a las series E96 y E24, respectivamente, considerando que $E_{maxG}=E_{max\omega p}=E_{maxQp}=5,00E-03$ (0,5%). Por otro lado, el escenario E2 propone emplear la serie E24 para los valores de las resistencias y la serie E12 para los valores de los capacitores, adoptando el valor $2,50E-02$ (2,5%) para E_{maxG} , $E_{max\omega p}$ y E_{maxQp} . En ambos escenarios el rango de valores definidos para las resistencias y capacitores es $10^3-10^6 \Omega$ y $10^{-9}-10^{-6}$ F, respectivamente. Se asume que valores fuera de estos rangos conducirían a efectos negativos debido a capacidades parásitas o señales de corriente muy grandes.

Tabla 1. Ecuaciones para obtener los parámetros funcionales y las sensibilidades de un filtro pasabajo IGMFB.

Parámetros Funcionales	Sensibilidades
$G = R_2/R_1$	$S_{R_1}^{Qp} = Qp \left(\frac{1}{R_1} \sqrt{\frac{R_2 R_3 C_5}{C_4}} \right)$
$\omega p = \sqrt{\frac{1}{R_2 R_3 C_4 C_5}}$	$S_{R_2}^{Qp} = -\frac{Qp}{2} \left(\frac{1}{R_1} \sqrt{\frac{R_2 R_3 C_5}{C_4}} - \sqrt{\frac{R_3 C_5}{R_2 C_4}} + \sqrt{\frac{R_2 C_5}{R_3 C_4}} \right)$
$Qp = \left[\sqrt{\frac{C_5}{C_4}} \left(\sqrt{\frac{R_2 R_3}{R_1}} + \sqrt{\frac{R_3}{R_2}} + \sqrt{\frac{R_2}{R_3}} \right) \right]^{-1}$	$S_{R_3}^{Qp} = -\frac{Qp}{2} \left(\frac{1}{R_1} \sqrt{\frac{R_2 R_3 C_5}{C_4}} + \sqrt{\frac{R_3 C_5}{R_2 C_4}} - \sqrt{\frac{R_2 C_5}{R_3 C_4}} \right)$

5 Resultados experimentales

5.1 Seteo y comportamiento de DE

El tamaño de la población de vectores (NP), el factor de escala (F), la probabilidad de recombinación de la población (Cr) y la cantidad máxima de generaciones (g_{max}) son parámetros de DE que influyen en su desempeño. Teniendo en cuenta [3, 4, 15, 16], se efectúan simulaciones considerando diferentes combinaciones de valores para los mismos y se seleccionan aquellos con los que el algoritmo obtiene la mejor performance. Los valores elegidos para cada parámetro son los que permiten a DE alcanzar el valor de fitness más bajo. En este trabajo, los valores seleccionados para el caso de estudio direccionado son: $NP=50$, $F=0,7$, $Cr=0,5$, $g_{max}=2500$.

El desempeño del método de diseño basado en DE es evaluado teniendo en cuenta las sensibilidades y los errores en los parámetros funcionales de las configuraciones de filtro obtenidas para la especificación propuesta.

Debido a que DE realiza un proceso estocástico, los resultados pueden variar. Con el propósito de explorar como los resultados pueden ser afectados por el seteo de la población inicial, se realizan 50 corridas del mismo, cambiando la semilla usada en la generación aleatoria de esta población.

En la Tabla 2 se presenta una caracterización de los valores obtenidos para las sensibilidades minimizadas, los errores en los parámetros funcionales del filtro y el fitness, correspondientes a las soluciones alcanzadas por DE en las 50 corridas efectuadas. Para cada uno de estos parámetros se presentan en la tabla su valor máximo,

mínimo y mediana. La mediana ha sido adoptada como medida de tendencia central debido a que la evaluación de los datos obtenidos permitió determinar que no estaban normalmente distribuidos. Con respecto a las sensibilidades, los valores máximos muestran que las mismas son siempre inferiores a la unidad, presentando $|S_{R_1}^{Qp}|$ los valores más altos y $|S_{R_3}^{Qp}|$ los valores más pequeños. Con respecto a los mínimos, $|S_{R_3}^{Qp}|$ asume valores cercanos a cero, mientras que las otras dos sensibilidades presentan valores más grandes. Para la mediana, $|S_{R_1}^{Qp}|$ y $|S_{R_2}^{Qp}|$ son similares entre sí y siempre mayores que $|S_{R_3}^{Qp}|$. Con referencia a los errores de diseño, es posible observar que los errores máximos nunca superan el error máximo tolerable definido para cada característica del filtro. Para los errores mínimos y la mediana, $Error_{\omega p}$ y $Error_{Qp}$ adoptan los valores más bajos en E1 y los más altos en E2, ocurriendo lo contrario con $Error_G$, para el cual los dos parámetros asumen el valor cero en E2. Con respecto a los valores de fitness alcanzados, el máximo es superior en E1, mientras que el mínimo y la mediana presentan valores similares entre un escenario y otro.

Tabla 2. Caracterización de resultados alcanzados por DE.

	E1			E2		
	Mínimo	Máximo	Mediana	Mínimo	Máximo	Mediana
$ S_{R_1}^{Qp} $	0,1433	0,5917	0,3659	0,2889	0,4293	0,3739
$ S_{R_2}^{Qp} $	0,3032	0,4524	0,3785	0,3586	0,4037	0,3754
$ S_{R_3}^{Qp} $	0,0001	0,3091	0,0567	0,0011	0,1148	0,0015
$Error_G$ (%)	0,1783	0,4619	0,3401	0	2,4390	0
$Error_{\omega p}$ (%)	0,0278	0,4966	0,3028	0,4737	2,4985	2,4985
$Error_{Qp}$ (%)	0,0326	0,4885	0,1606	0,2813	2,4195	1,1448
$Fitness$	0,2502	0,3945	0,2601	0,2503	0,2862	0,2503

En la Tabla 3 se reportan los valores de los componentes correspondientes a la mejor solución encontrada por DE en las 50 corridas efectuadas. Esta solución es aquella que presenta el valor más bajo de fitness (0,2502 en E1 y 0,2503 en E2).

Tabla 3. Valores de los componentes para la mejor solución encontrada por DE.

Parámetro	E1	E2
R_1	18700	11000
R_2	56200	33000
R_3	14000	8200
C_4	1,60E-08	2,70E-08
C_5	2,00E-09	3,30E-09
$ S_{R_1}^{Qp} $	0,3747	0,3739
$ S_{R_2}^{Qp} $	0,3753	0,3754
$ S_{R_3}^{Qp} $	5,57E-04	0,0015
$Error_G$ (%)	0,1783	0
$Error_{\omega p}$ (%)	0,3028	2,4985
$Error_{Qp}$ (%)	0,0517	1,1448

Por otro lado, la ecuación (1) es utilizada para obtener la función de transferencia del filtro pasabajo adoptado como caso de estudio, considerando la especificación elegida (G_F , ω_{pF} , y Q_{pF}). El filtro correspondiente a la función de transferencia obtenida es denominado filtro nominal y puede ser utilizado como referencia para evaluar la bondad de las soluciones encontradas mediante el método propuesto. La Fig. 3 muestra la respuesta en frecuencia (sólo magnitud) del filtro nominal y las respuestas de los filtros obtenidos usando DE para los dos escenarios propuestos. La configuración de filtro adoptada en cada escenario corresponde a la peor solución encontrada por DE (solución con el fitness más alto) en las 50 corridas. La figura muestra que en ambos casos las curvas no muestran diferencias significativas.

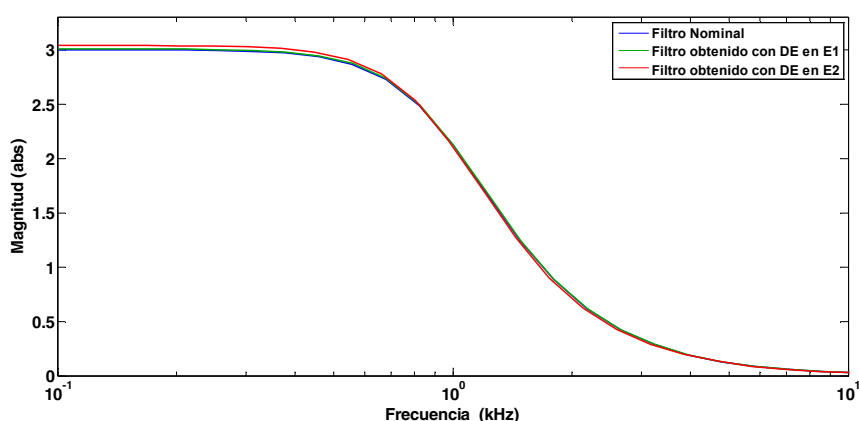


Fig. 3. Respuesta en frecuencia del filtro nominal y de los filtros obtenidos por DE (peor caso).

5.2 Comparación con otros algoritmos evolutivos

Con el propósito de complementar la evaluación del método de diseño basado en DE que se propone en este trabajo, se efectúa una comparación con resultados alcanzados en trabajos previos empleando otros EA. Particularmente, se considera la utilización de un GA [13] y de PSO [13]. Si bien la mejor solución encontrada por los tres algoritmos en 50 corridas es la misma, existen algunos otros aspectos que pueden ser considerados para evaluar el desempeño de los tres algoritmos. Específicamente, en este trabajo, la comparación de resultados se efectúa teniendo en cuenta los siguientes criterios: cantidad de evaluaciones de la función fitness (CEFF), tiempos de ejecución y valores de fitness alcanzados por cada algoritmo.

Con respecto a la CEFF, la misma se calcula para cada algoritmo mediante el producto entre el tamaño de la población y la cantidad máxima de generaciones definida. En la Tabla 4 se muestran los valores alcanzados por DE, GA y PSO para el caso de estudio abordado en el presente trabajo, bajo los dos escenarios propuestos. La tabla muestra que para DE, la CEFF es considerablemente menor a la empleada por los otros dos algoritmos. En la tabla también se visualizan los tiempos de ejecución máximos alcanzados por cada algoritmo en una corrida. Los tres EA han sido ejecutados en una PC Intel Core i7-4790K, 4,00GHz, 32,0 GB de RAM. Es posible observar que existe una congruencia entre los tiempos de cada algoritmo y la CEFF,

siendo en consecuencia DE el algoritmo que emplea el menor tiempo. Por otro lado, la tabla muestra que en E1, aun empleando la misma CEFF por el GA y PSO, el GA presenta tiempos considerablemente más altos. Esta característica refleja que los tiempos del GA, independientemente de la CEFF empleada, son muy superiores a los tiempos de los otros dos algoritmos.

Tabla 4. Comparación de resultados con GA y PSO.

Algoritmo	CEFF		Tiempos de ejecución (seg.)	
	E1	E2	E1	E2
<i>DE</i>	125000	125000	4,2276	4,1655
<i>GA</i>	400000	500000	59,9349	71,8789
<i>PSO</i>	400000	250000	8,3768	5,6256

Otro aspecto interesante para analizar en los EA está relacionado con el valor de fitness alcanzado por las soluciones encontradas en las diferentes corridas. En la Tabla 5 se muestra una caracterización de estos valores para DE, GA y PSO. Con respecto a los valores mínimos, en cada escenario los mismos coinciden para los tres algoritmos. No obstante, para el máximo y la mediana es posible apreciar que para el GA se presentan los valores más altos. DE y PSO presentan valores similares entre sí, sin embargo, los valores más bajos siempre corresponden a DE. Esta particularidad demuestra que, en general, DE puede encontrar soluciones que presentan valores más bajos de fitness. Por otro lado, el rango de variación de los valores de fitness siempre es menor para DE. Esta propiedad también es importante, ya que, si bien el método de diseño ofrece diferentes soluciones alternativas en cada corrida del EA, resulta también interesante que estas alternativas presenten variaciones pequeñas en sus valores de fitness, de manera que la calidad de las soluciones sea similar.

Tabla 5. Comparación de resultados con GA y PSO teniendo en cuenta los valores de fitness.

Algoritmo	E1				E2			
	Mínimo	Máximo	Rango	Mediana	Mínimo	Máximo	Rango	Mediana
<i>DE</i>	0,2502	0,3945	0,1443	0,2601	0,2503	0,2862	0,0359	0,2503
<i>GA</i>	0,2502	0,4517	0,2015	0,2887	0,2503	0,4739	0,2236	0,2597
<i>PSO</i>	0,2502	0,4064	0,1562	0,2631	0,2503	0,3122	0,0619	0,2513

6 Conclusión

En este trabajo se propone utilizar el algoritmo DE para el diseño de filtros bicuadráticos. El algoritmo debe determinar los valores de los componentes pasivos del filtro que brinden una configuración robusta frente a las variaciones de los mismos, mediante la minimización de las sensibilidades. Adicionalmente, el método de diseño debe satisfacer las especificaciones fijadas sobre los parámetros funcionales del filtro con un error predeterminado por el diseñador. Se considera como caso de estudio un filtro bicuadrático pasabajo IGMFB. Los resultados muestran que DE permite obtener configuraciones de filtro que presentan sensibilidades bajas, además de satisfacer las restricciones planteadas. Por otro lado, DE utiliza menos evaluaciones de la función

fitness y tiempos de ejecución más bajos, en comparación con otros EA propuestos anteriormente para implementar el método de diseño.

En trabajos futuros, se pretende extender la aplicación del método de dimensionamiento a configuraciones de filtros más complejas y/o de orden superior.

Referencias

1. Chen, W. K.: *Passive, Active, and Digital Filters*, CRC Press (2009).
2. Raut, R., Swamy, M. N. S.: *Modern Analog Filter Analysis and Design: A Practical Approach*. Wiley-VCH (2010).
3. El-Ghazali Talbi: *Metaheuristics from Design to Implementation*, Wiley (2009).
4. Yu, X. and Gen, M.: *Introduction to Evolutionary Algorithms*, Springer 2010.
5. Jiang, M., Yang, Z., Gan, Z.: Optimal components selection for analog active filters using clonal selection algorithms. In *Advanced Intelligent Computing Theories and Applications. With Aspects of Theoretical and Methodological Issues* (pp. 950-959). Springer Berlin Heidelberg (2007).
6. Vural, R. A., Yildirim, T.: State variable filter design using particle swarm optimization. In *Symbolic and Numerical Methods, Modeling and Applications to Circuit Design (SM2ACD)*, XIth International Workshop, pp. 1-4 (2010).
7. Vural, R. A., Yildirim, T., Kadioglu, T., Basargan, A.: Performance evaluation of evolutionary algorithms for optimal filter design. *Evolutionary Computation*, IEEE Transactions on, 16(1), 135-147 (2012).
8. Vural, R. A., Bozkurt, U., Yildirim, T.: Analog active filter component selection with nature inspired metaheuristics. *AEU-International Journal of Electronics and Communications*, 67(3), pp. 197-205 (2013).
9. Bose, D., Biswas, S., Vasilakos, A. V., Laha, S.: Optimal filter design using an improved artificial bee colony algorithm. *Information Sciences*, 281, pp. 443-461 (2014).
10. De, B. P., Kar, R., Mandal, D., Ghoshal, S. P.: Optimal selection of components value for analog active filter design using simplex particle swarm optimization. *International Journal of Machine Learning and Cybernetics*, 6(4), pp. 621-636 (2015).
11. De, B. P., Kar, R., Mandal, D., Ghoshal, S. P.: Optimal analog active filter design using craziness-based particle swarm optimization algorithm. *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, 28(5), pp. 593-609 (2015).
12. De, B. P., Kar, R., Mandal, D., Ghoshal, S. P.: Particle swarm optimization with aging leader and challengers for optimal design of analog active filters. *Circuits, Systems, and Signal Processing*, 34(3), pp. 707-737 (2015).
13. Lovay, M., Romero, E., Peretti, G.: Diseño de Filtros Activos Robustos usando Algoritmos Genéticos, *Proceedings of SII 2015, 4th Argentine Symposium on Industrial Informatics*, 44 JAIIO - 44th Argentine Conference on Informatics, September 2015, Facultad de Ciencias Exactas, Ingeniería y Agrimensura, Universidad Nacional de Rosario, Rosario, Santa Fe, Argentina, pp. 169-180, ISSN: 2451-7542 (2015).
14. Lovay, M., Romero, E., Peretti, G.: Diseño óptimo de filtros bicuadráticos mediante optimización por enjambre de partículas, *Libro de Memorias VII Congreso de Microelectrónica Aplicada 2016*, San Luis, Argentina, octubre de 2016, ISBN 978-987-733-068-7. Editado por Nueva Editorial Universitaria (2016).
15. Storn, R., Price, K.: Differential Evolution – a simple and efficient adaptive scheme for global optimization over continuous spaces, *J. Glob Optim* 1995, 11, pp.341-359 (1995).
16. Rout, U., Sahu, R., Panda, S.: Design and analysis of differential evolution algorithm based automatic generation control for interconnected power system, *Ahin Shams Engineering Journal*, (4), pp. 409-421 (2013).