

Aplicación del algoritmo de Optimización por Enjambre de Partículas en el dimensionamiento óptimo de componentes para Filtros Activos

Mónica Lovay, Eduardo Romero, y Gabriela Peretti

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

Resumen. En este trabajo se presenta un método basado en el algoritmo de optimización por enjambre de partículas (PSO, Particle Swarm Optimization) para realizar el dimensionamiento óptimo de componentes en filtros activos. PSO debe resolver un problema de optimización multiobjetivo para determinar los valores de las resistencias y capacitores 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. Con el propósito de evaluar la viabilidad del método de diseño propuesto se considera como caso de estudio un filtro Sallen-Key pasabanda de segundo orden. Se asume un escenario donde los valores posibles para las resistencias y capacitores corresponden a los de las series E96 y E24, respectivamente. Los resultados de las simulaciones realizadas muestran que PSO permite obtener configuraciones del filtro bajo estudio con sensibilidades bajas y errores de diseño que satisfacen las especificaciones.

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

1. Introducción

Los filtros analógicos son circuitos utilizados en un amplio rango de aplicaciones. Dependiendo del tipo de elementos empleados en su construcción, estos filtros pueden clasificarse en pasivos o activos. Los filtros pasivos se implementan con resistencias, capacitores e inductores. Los filtros activos usan adicionalmente transistores o amplificadores operacionales.

En el proceso de diseño de un filtro activo la selección de los componentes discretos es un aspecto de gran importancia, dado que de su correcta elección dependerá en gran medida el cumplimiento de las especificaciones. A los fines de realizar un diseño confiable, los valores de los componentes pasivos se escogen de las series industriales E12, E24, E48, E96 o E192. Cada una de estas series limita los valores que pueden asumir dichos componentes.

En el enfoque convencional de diseño, generalmente se proponen valores iguales para algunos componentes [1, 2, 3]. Esta consideración simplifica el proceso pero también limita la libertad de diseño. Por otro lado, la aproximación de los valores de

los componentes obtenidos por los disponibles en las series E impacta en la calidad del filtro.

Una alternativa para evitar los problemas derivados del método convencional consiste en enumerar todas las posibles combinaciones de valores para las resistencias y capacitores que se encuentran en la topología del filtro y encontrar aquella que mejor satisface las especificaciones. Sin embargo, debido a que los valores que pueden adoptar los componentes pertenecen a una determinada serie E y que se considera generalmente para los mismos un rango de varias décadas, esta búsqueda exhaustiva no es factible en tiempos razonables debido a la gran cantidad de alternativas que presenta. En este escenario, son necesarios métodos de búsqueda inteligente que logren cumplir los objetivos en tiempos computacionales cortos.

El algoritmo de optimización por enjambre de partículas (PSO, Particle Swarm Optimization) se inspira en el comportamiento social de organismos tales como las bandadas de aves o los bancos de peces. Este método, que surgió hace poco más de una década, es empleado para resolver diferentes problemas de optimización [4, 5, 6].

Algunos autores han propuesto la utilización de PSO para efectuar el dimensionamiento de componentes en filtros activos. Particularmente, en [7] se utiliza PSO para efectuar el diseño de un filtro pasabajo de variable de estado de segundo orden, considerando que las resistencias y capacitores deben ser compatibles con las series E24 y E96. El desempeño de PSO junto a otros dos algoritmos evolutivos es evaluado en [8] para realizar el diseño de un filtro Butterworth pasabajo de cuarto orden y del filtro de segundo orden adoptado en [7]. Tres versiones mejoradas del algoritmo PSO: Simplex PSO, CRPSO y ALC-PSO son propuestas en [9, 10, 11], respectivamente, para efectuar el diseño de filtros considerando las dos topologías utilizadas en [8]. En todos los trabajos referenciados el proceso de diseño propuesto está enfocado hacia la minimización de los errores de diseño respecto de las especificaciones. Adicionalmente, en [9] y [11] se efectúa, para la mejor solución arrojada por el algoritmo, un análisis de sensibilidad de la función objetivo empleada con respecto a variaciones en los valores de los componentes pasivos. En este sentido, resulta interesante destacar que si las sensibilidades fueran incorporadas y consideradas en el proceso de diseño, se deberían poder obtener diseños más robustos. Esto significa que el filtro exhibirá una mayor estabilidad en sus parámetros funcionales respecto a variaciones en los valores de los componentes. Otro aspecto común en los trabajos citados está relacionado con el ajuste que debe realizarse sobre la solución encontrada por el algoritmo para que la misma resulte compatible con la serie E considerada. Para tal fin, los autores de estos trabajos proponen redondear el valor obtenido para algunos de los componentes al valor de la serie E más próximo. Con el propósito de mantener bajo el error de diseño, los componentes restantes son obtenidos mediante la conexión de dos o más componentes que pertenezcan a la correspondiente serie E adoptada. Esto implica que el método no es totalmente automático y es susceptible de obtener soluciones que compliquen la implementación real debido a placas de circuito impreso complejas.

En este trabajo, se propone aplicar PSO para realizar el diseño de filtros activos. Se lleva a cabo el proceso de selección de los valores de los componentes para cumplir los objetivos de diseño y al mismo tiempo minimizar las sensibilidades de los parámetros funcionales frente a variaciones en los componentes. Además, el algoritmo limita su espacio de búsqueda solamente a configuraciones de filtro cuyos

valores de componentes pertenezcan a las series E consideradas. Esta característica implica que la solución arrojada por PSO puede ser directamente implementada, no requiriendo de ningún tipo de ajuste de valores y/o de conexiones adicionales. Con el propósito de comprobar la viabilidad del método propuesto se adopta como caso de estudio un filtro Sallen-Key pasabanda de segundo orden.

2. Filtros activos

Los filtros activos cumplen un rol esencial en diferentes tipos de sistemas. Particularmente, los filtros activos de segundo orden son utilizados con frecuencia en el diseño de filtros de orden superior mediante conexiones en cascada. La implementación de filtros de segundo orden puede ser realizada empleando alguna de las diferentes topologías existentes para tal fin. Entre ellas, la topología Sallen-Key (SK) es una de las más usadas. Esta configuración requiere de una mínima cantidad de componentes. No obstante, las sensibilidades de estos filtros pueden ser altas en algunos casos.

La sensibilidad es un parámetro que permite medir la variación de la performance de un filtro como resultado de cambios en los valores de sus componentes. Estas variaciones pueden ocurrir debido al envejecimiento de los mismos, tolerancias de fabricación, condiciones ambientales (temperatura), entre otros factores [2, 3]. 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.

De manera general, si P es una función de varias variables, $P=f(x_1, x_2, \dots, x_n)$, entonces la sensibilidad de P con respecto a x_i , está definida por:

$$S_{x_i}^P = \frac{\% \text{ cambio en } P}{\% \text{ cambio en } x_i} = \frac{\partial P/P}{\partial x_i/x_i} \quad (1)$$

Se considera que un filtro tiene baja sensibilidad cuando todas sus sensibilidades adoptan valores inferiores a la unidad [3].

En la Fig. 1 se muestra el circuito correspondiente al filtro SK pasabanda de segundo orden, propuesto como caso de estudio. Los componentes pasivos son cinco resistencias y dos capacitores.

La función de transferencia $F(s)$ para este tipo de filtro, 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, se muestra en (2).

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

Los valores de ω_p , G y Q_p se relacionan con los valores de los componentes pasivos mediante las siguientes expresiones:

$$\omega p = \sqrt{\frac{R_1 + R_2}{R_1 R_2 R_4 C_5 C_3}} \quad (3)$$

$$G = \frac{G_a}{R_1 C_5} \left[\frac{1}{R_1 C_5} + \frac{1}{R_4 C_5} + \frac{1}{R_4 C_3} + \frac{1 - G_a}{R_2 C_5} \right]^{-1}, \quad (4)$$

$$Qp = \omega p \left[\frac{1}{R_1 C_5} + \frac{1}{R_4 C_5} + \frac{1}{R_4 C_3} + \frac{1 - G_a}{R_2 C_5} \right]^{-1}, \quad (5)$$

donde: $G_a = 1 + \frac{R_a}{R_b}$.

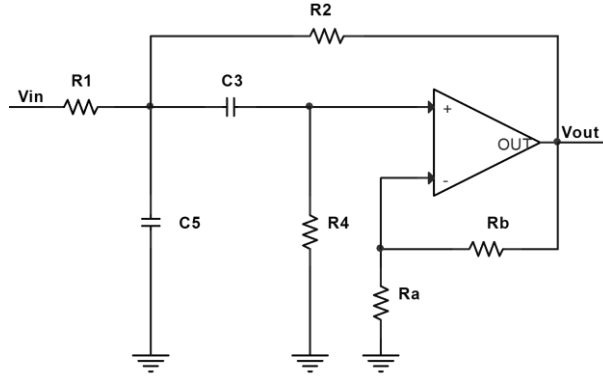


Fig. 1. Filtro SK pasabanda de segundo orden.

Las sensibilidades de ωp y Qp con respecto a los componentes pasivos son:

$$S_{R_4}^{\omega p} = S_{C_3}^{\omega p} = S_{C_5}^{\omega p} = -(1/2) \quad (6)$$

$$S_{R_1}^{\omega p} = -R_2 / (2(R_1 + R_2)) \quad (7)$$

$$S_{R_2}^{\omega p} = -R_1 / (2(R_1 + R_2)) \quad (8)$$

$$S_{R_1}^{Qp} = \frac{R_2(C_3 R_1 R_4 - C_3 R_1 R_2 - C_5 R_1 R_2 + C_3 R_2 R_4 + C_3 G_a R_1 R_4)}{(2(R_1 + R_2)(C_3 R_1 R_2 + C_3 R_1 R_4 + C_5 R_1 R_2 + C_3 R_2 R_4 - C_3 G_a R_1 R_4))} \quad (9)$$

$$S_{R_2}^{Qp} = -\frac{R_1(R_1 R_2(C_3 + C_5) + R_1 R_4 C_3(G_a - 1) + R_2 R_4 C_3(2G_a - 1))}{(2(R_1 + R_2)(C_3 R_1 R_2 + C_3 R_1 R_4 + C_5 R_1 R_2 + C_3 R_2 R_4 - C_3 G_a R_1 R_4))} \quad (10)$$

$$S_{R_4}^{Qp} = \frac{C_3 R_1 R_2 + C_5 R_1 R_2}{C_3 R_1 R_2 + C_3 R_1 R_4 + C_5 R_1 R_2 + C_3 R_2 R_4 - C_3 G_a R_1 R_4} - \frac{1}{2} \quad (11)$$

$$S_{C_3}^{Qp} = -S_{C_5}^{Qp} = \frac{C_5 R_1 R_2}{C_3 R_1 R_2 + C_3 R_1 R_4 + C_5 R_1 R_2 + C_3 R_2 R_4 + C_3 G_a R_1 R_4} - \frac{1}{2} \quad (12)$$

Como puede verse en (6), las sensibilidades de ωp con respecto a R_4 , C_3 y C_5 son independientes de los valores de los componentes del filtro. Por el contrario, las sensibilidades dadas por (7)-(12) son dependientes de los mismos, por lo que deben ser consideradas en la selección de los valores de resistencias y capacitores.

En el presente trabajo, la especificación elegida para el filtro que se desea diseñar es $G_F=1$, $\omega p_F=795*2*\pi=4995,13$ rad/s y $Qp_F=1,11$.

3. Diseño de filtros con PSO

3.1. Definición del problema de optimización

Para el caso de estudio propuesto, la tarea de PSO consiste en encontrar los valores de los componentes pasivos para los cuales las sensibilidades sean mínimas, y los errores en G , ωp y Qp sean inferiores a un error máximo especificado.

El problema de optimización multiobjetivo (POM) que debe ser resuelto por el algoritmo puede representarse mediante la siguiente expresión:

$$\begin{aligned} \min S(i) &= S_1(i), S_2(i), \dots, S_n(i) \\ \text{sujeto a: } & \text{Error}_G(i) \leq E_{\max G} \\ & \text{Error}_{\omega p}(i) \leq E_{\max \omega p} \\ & \text{Error}_{Qp}(i) \leq E_{\max Qp} \end{aligned} \quad (13)$$

En (13), i es una posible configuración de filtro determinada por los valores de resistencias y capacitores. S_1 a S_n representan las sensibilidades que deben ser minimizadas. Por otro lado, Error_G , $\text{Error}_{\omega p}$ y Error_{Qp} representan los errores en G , ωp y Qp , respectivamente, y se calculan de acuerdo a las expresiones (14) a (16). $E_{\max G}$, $E_{\max \omega p}$ y $E_{\max Qp}$ definen el error máximo tolerable para cada una de las características del filtro.

$$\text{Error}_G(i) = |(G(i) - G_F)/G_F| \quad (14)$$

$$\text{Error}_{\omega p}(i) = |(\omega p(i) - \omega p_F)/\omega p_F| \quad (15)$$

$$\text{Error}_{Qp}(i) = |(Qp(i) - Qp_F)/Qp_F| \quad (16)$$

3.2. Implementación de PSO

PSO es un método de búsqueda iterativo y colectivo, con énfasis en la cooperación, inspirado en el movimiento coordinado de los animales que viven en grupos. El comportamiento (dirección y velocidad) de cada individuo es el efecto de influencias cognitivas, sociales y estocásticas. El objetivo común de todos los miembros de la población consiste en encontrar la ubicación más favorable dentro de un determinado espacio de búsqueda.

En PSO, cada individuo es considerado una partícula que se mueve en un espacio de búsqueda multidimensional. La población o grupo también se denomina enjambre (swarm), y la trayectoria de cada partícula en el espacio de búsqueda es ajustada dinámicamente mediante la alteración de su velocidad. Estas partículas vuelan a través del espacio de búsqueda, teniendo dos capacidades de razonamiento esenciales: la memoria de la mejor posición individual alcanzada (influencia cognitiva) y el conocimiento de la mejor posición global (influencia social) [4, 5, 6]. El diagrama de flujo de la Fig. 2 muestra las operaciones realizadas por el algoritmo PSO implementado, el cual ha sido desarrollado en Matlab™.

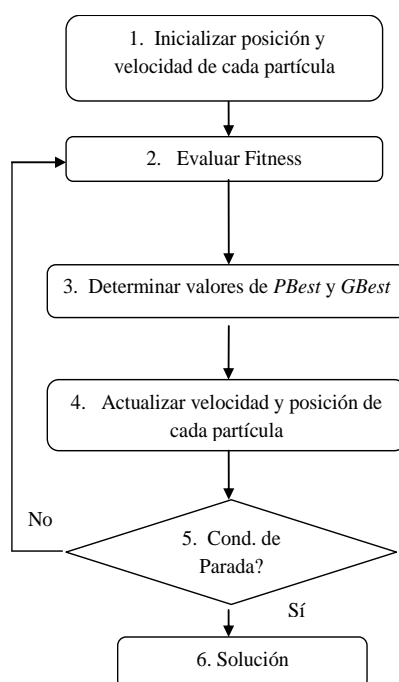


Fig. 2. Diagrama de flujo de PSO.

A continuación, se describe brevemente cada bloque del diagrama.

1. *Inicializar posición y velocidad de cada partícula*: El algoritmo comienza generando de manera aleatoria una población (o enjambre) inicial de partículas que son soluciones potenciales para el problema de optimización que se desea resolver. En este caso de estudio, cada partícula representa una posible configuración de filtro caracterizada por los valores de resistencias y capacitores. El número de resistencias y capacitores permite determinar la cantidad de dimensiones de las partículas con las que debe trabajar el algoritmo. Las expresiones (17) y (18), representan, respectivamente, la posición y velocidad de cada partícula i , en un espacio de búsqueda que en este caso es de siete dimensiones:

$$x_i = (x_{iR1}, x_{iR2}, x_{iR4}, x_{iRa}, x_{iRb}, x_{iC3}, x_{iC5}) \quad (17)$$

$$v_i = (v_{iR1}, v_{iR2}, v_{iR4}, v_{iRa}, v_{iRb}, v_{iC3}, v_{iC5}) \quad (18)$$

2. *Evaluar Fitness*: En cada generación (iteración), las partículas de la población actual son evaluadas de acuerdo a un criterio de calidad predefinido, denominado función de aptitud o fitness. El POM formulado en (13) es convertido en un problema de optimización con un único objetivo. Para ello, es utilizado el método de suma ponderada, el cual emplea una función de agregación para efectuar la transformación [12, 13, 14]. De esta manera, la función de *fitness* f que asigna un valor de aptitud a cada partícula i , se define mediante la expresión (19).

$$f(i) = w_1 |S_{R1}^{op}(i)| + w_2 |S_{R2}^{op}(i)| + w_3 |S_{R1}^{Qp}(i)| + w_4 |S_{R2}^{Qp}(i)| + w_5 |S_{R4}^{Qp}(i)| + w_6 |S_{C3C5}^{Qp}(i)|, \quad (19)$$

donde: $S_{C3C5}^{Qp}(i) = |S_{C3}^{Qp}(i)| = |S_{C5}^{Qp}(i)|$.

En (19), w_j es el peso asignado a cada una de las sensibilidades que deben ser minimizadas. Debido a que se considera que no existe prioridad entre las mismas, w_j adopta el mismo valor (1/6) para todas las sensibilidades.

PSO aplica las restricciones, penalizando en cada iteración a aquellas partículas que no las cumplen, mediante la expresión (20) [15]. En consecuencia, éstas adoptan un valor de fitness más alto de acuerdo al grado de violación de cada restricción.

$$f(i) = f(i) + \frac{|E_{maxG} - Error_{G,i}|}{Error_{G,max}} + \frac{|E_{maxop} - Error_{op,i}|}{Error_{op,max}} + \frac{|E_{maxQp} - Error_{Qp,i}|}{Error_{Qp,max}} \quad (20)$$

En (20), $Error_{G,i}$, $Error_{op,i}$, $Error_{Qp,i}$ representan, respectivamente, los errores en G , op y Qp , para la partícula i . Por otro lado, $Error_{G,max}$, $Error_{op,max}$ y $Error_{Qp,max}$ representan el máximo error alcanzado en la generación actual para G , op y Qp , respectivamente.

3. *Determinar valores de PBest y GBest*: Teniendo en cuenta el valor de fitness de cada partícula, el algoritmo determina el mejor valor de fitness alcanzado por cada partícula i hasta la generación actual k , y lo almacena en $PBest_i^k$. Por otro lado, computa el mejor valor de fitness entre todas las partículas hasta la generación actual y lo almacena en $GBest^k$.

4. *Actualizar velocidad y posición de cada partícula*: En cada generación k , cada partícula i modifica su posición usando su velocidad actual v_i^k y la distancia hacia $PBest_i^k$ y $GBest^k$. La velocidad v_i^{k+1} de cada partícula y su nueva posición x_i^{k+1} se calculan utilizando, respectivamente, las siguientes ecuaciones:

$$v_i^{k+1} = \underbrace{\phi v_i^k}_{\text{Inercia}} + \underbrace{c_1 r_{1,i} (PBest_i^k - x_i^k)}_{\text{Influencia cognitiva}} + \underbrace{c_2 r_{2,i} (GBest^k - x_i^k)}_{\text{Influencia social}} \quad (21)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (22)$$

En (21), ϕ es la masa inercial, c_1 y c_2 son constantes positivas conocidas como coeficientes de aceleración, que determinan la influencia relativa de $PBest_i^k$ y $GBest^k$, respectivamente, en el cálculo de la velocidad de cada partícula. Por otro lado, $r_{1,i}$ y $r_{2,i}$ son variables aleatorias independientes distribuidas uniformemente en el rango (0,1). El primer término de la ecuación (21) representa el impacto de la velocidad actual de la partícula en la velocidad con que la partícula se moverá a su siguiente

posición. El segundo término, denominado por algunos autores como componente de influencia cognitiva, representa la experiencia personal de cada partícula. El tercer término, también conocido como componente de influencia social, representa el efecto colaborativo de las partículas, que permite dirigir cada partícula hacia la mejor solución global del enjambre. Esto significa que cuando una partícula alcanza una nueva posición prometedora las otras partículas tratan de moverse hacia esa nueva posición.

Con respecto a la masa inercial ϕ , la misma es empleada con el propósito de balancear la búsqueda global y local durante el proceso evolutivo. De manera general, valores altos de ϕ ayudan a efectuar una exploración más exhaustiva durante el proceso, beneficiando la búsqueda global. Por el contrario, valores más bajos de ϕ favorecen la búsqueda local en torno al espacio de búsqueda actual. Algunos autores proponen decrementar linealmente el valor de la masa inercial durante el proceso de búsqueda con el propósito de evitar la convergencia prematura del algoritmo en la etapa inicial del proceso y potenciar la convergencia del mismo a una solución óptima global durante la última etapa de la búsqueda. Teniendo en cuenta estos aspectos, (23) representa la función utilizada en este caso de estudio para decrementar el valor de la masa inercial ϕ en el proceso evolutivo [16, 17].

$$\phi = \phi_{max} - \frac{\phi_{max} - \phi_{min}}{K_{max}} k \quad (23)$$

En (23), k es la generación actual, K_{max} es el máximo número de generaciones, ϕ_{max} y ϕ_{min} definen los extremos (superior e inferior, respectivamente) del rango de valores entre los que se decrementa la masa inercial durante la búsqueda.

5. *Condición de parada:* El proceso descrito se repite hasta que se satisface la condición de parada, que consiste en alcanzar un máximo número de generaciones.

6. *Solución:* Cuando se cumple el criterio de parada el algoritmo se detiene, arrojando como solución aquella partícula que alcanzó el mejor valor de fitness durante el proceso evolutivo ($GBest^k$).

4. Resultados de las simulaciones

El desempeño del método de diseño basado en PSO es evaluado teniendo en cuenta las sensibilidades y los errores en los parámetros funcionales de las configuraciones de filtro obtenidas. Para ello, se considera que las resistencias pueden adoptar valores en el rango 10^3 - $10^6\Omega$, de acuerdo a la serie E96. Con respecto a los capacitores el rango definido es 10^{-9} - $10^{-6}F$, de acuerdo a la serie E24. Se asume que valores fuera de estos rangos conducirían a efectos negativos debido a capacidades parásitas o señales de corriente muy grandes. De esta manera, PSO dispone de 288 valores posibles para cada resistencia y 72 valores posibles para cada capacitor. El espacio de búsqueda total asciende a 1,03E16 configuraciones.

Con fines demostrativos se establece un valor de 5,00E-03 (0,5%) para el error máximo tolerable en cada uno de los parámetros funcionales del filtro (E_{maxG} , E_{maxOp} y E_{maxQp}) definidos en (13).

El tamaño de la población (N), los coeficientes de aceleración (c_1 y c_2), los valores entre los que varía la masa inercial (ϕ_{max} y ϕ_{min}) y la cantidad máxima de generaciones (K_{max}) son parámetros de PSO que influyen en el desempeño del mismo. Teniendo en cuenta [7, 8, 9, 10, 11, 16, 17], se efectúan simulaciones considerando diferentes combinaciones de valores para los mismos y se seleccionan aquellos que permiten alcanzar al algoritmo la mejor performance. En este caso de estudio, los valores elegidos para cada uno de los parámetros son: $N=75$, $c_1=2$, $c_2=2$, $\phi_{max}=0,9$, $\phi_{min}=0,4$, $K_{max}=10000$.

Debido a que PSO realiza un proceso estocástico, los resultados pueden variar de acuerdo a la distribución estadística de la población inicial. Con el propósito de verificar si el algoritmo encuentra soluciones favorables independientemente del seteo que se efectúe para la población inicial, se efectúan 50 corridas del mismo, cambiando la semilla en la generación aleatoria de la población inicial de cada corrida.

En la Tabla 1 se reportan los valores de los componentes, sensibilidades y errores de diseño correspondientes a la mejor solución encontrada por el algoritmo en las 50 corridas efectuadas con los parámetros elegidos. La tabla muestra que las sensibilidades son inferiores a la unidad en todos los casos. Por otro lado, los errores de diseño no superan el error máximo establecido (0,5%).

Tabla 1. Características de la mejor solución alcanzada por PSO.

Parámetro	Valor obtenido
R_1	5230 Ω
R_2	2610 Ω
R_4	453000 Ω
R_a	100000 Ω
R_b	20000 Ω
C_1	1,00E-09 F
C_2	5,10E-08 F
$ S_{R1}^{wp}(y) $	0,1665
$ S_{R2}^{wp}(y) $	0,3335
$ S_{R1}^{qp}(y) $	0,6672
$ S_{R2}^{qp}(y) $	0,6676
$ S_{R4}^{qp}(y) $	0,0005
$ S_{C3}^{qp}(y) = S_{C5}^{qp}(y) $	0,0092
$Error_G$	0,034%
$Error_{op}$	0,183%
$Error_{qp}$	0,122%

Teniendo en cuenta las especificaciones se utiliza (2) para obtener la función de transferencia de un filtro al que denominamos filtro nominal. Este filtro no tiene en cuenta las sensibilidades, debido a que no es obtenido utilizando los valores de los componentes del mismo. La respuesta en frecuencia del filtro nominal es utilizada para observar de manera gráfica el cumplimiento de las restricciones por parte de las soluciones encontradas por PSO. En la Fig. 3 se muestra la respuesta en frecuencia del filtro nominal y la respuesta de la mejor configuración de filtro alcanzada por PSO

(Tabla 1). En la figura es posible observar que la respuesta del filtro obtenido es considerablemente similar a la respuesta del filtro nominal.

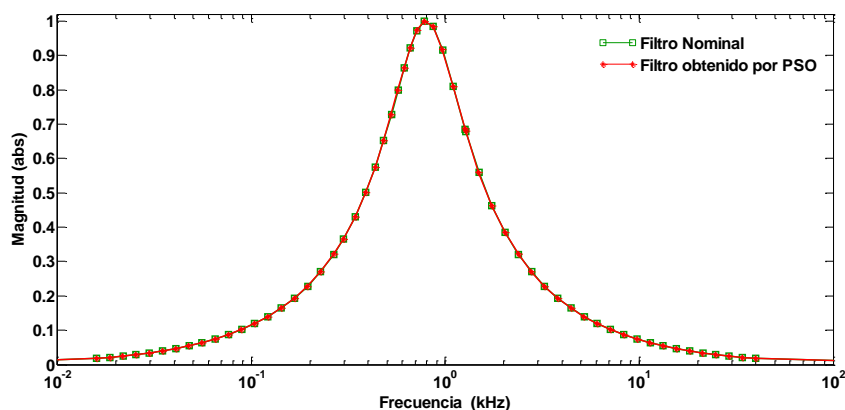


Fig. 3. Resposta en frecuencia del filtro nominal y de la mejor configuración de filtro encontrada por PSO.

Para complementar el análisis de resultados, se muestran en la Fig. 4 los valores de sensibilidad correspondientes a la solución alcanzada por el algoritmo en cada corrida. Puede observarse que las sensibilidades varían de una corrida a otra. No obstante, las mismas adoptan siempre valores inferiores a 1. Debe destacarse que la cantidad de corridas no tiene un impacto significativo en la performance del algoritmo. S_{R2}^{op} , S_{R4}^{Qp} , S_{R1}^{op} , S_{C3}^{Qp} y S_{C5}^{Qp} se mantienen casi constantes, mientras que S_{R1}^{Qp} , S_{R2}^{Qp} y S_{R4}^{Qp} varían levemente entre una corrida y otra. Por otro lado, las sensibilidades que adoptan valores más altos son S_{R1}^{Qp} y S_{R2}^{Qp} .

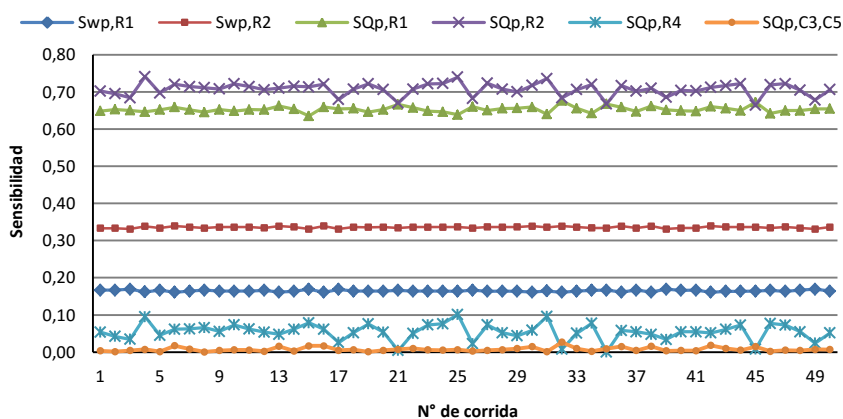


Fig. 4. Sensibilidades correspondientes a las soluciones encontradas por PSO en 50 corridas.

En la Tabla 2 se presenta una caracterización de los errores de diseño y valores de fitness alcanzados. Con respecto a los errores de diseño, en ningún caso el error

máximo supera el 0,5%. Para la mediana, el valor más bajo corresponde a $Error_G$ y los dos errores restantes presentan valores similares entre sí. Por otro lado, el valor mínimo más alto corresponde a $Error_{op}$, mientras que los restantes errores adoptan valores iguales. Con referencia a los valores de fitness, la variación entre el máximo, el mínimo y la mediana es muy pequeña. Esta característica demuestra que a pesar de la naturaleza estocástica del algoritmo, el mismo presenta un desempeño regular a lo largo de las corridas. Esta particularidad se traduce en una importante ventaja para los diseñadores debido a que tienen la posibilidad de elegir entre variadas opciones de diseño que presentan características de sensibilidad similares.

Tabla 2. Caracterización de errores y fitness en 50 corridas de PSO.

	Máximo	Mínimo	Mediana
$Error_G$	0,401%	0,001%	0,067%
$Error_{op}$	0,488%	0,024%	0,146%
$Error_{Op}$	0,461%	0,001%	0,122%
$Fitness$	0,285	0,265	0,276

Con respecto a los tiempos de ejecución, PSO emplea como máximo 11,902 segundos. Se estima que si se codifica en Matlab™ el método de búsqueda exhaustiva demandaría aproximadamente 4,46E10 segundos para encontrar la mejor solución. Esta estimación se realiza reduciendo el espacio de búsqueda y considerando un comportamiento lineal del método. PSO encuentra una solución favorable empleando tiempos computacionales considerablemente más bajos que la búsqueda exhaustiva, lo cual justifica la utilización del método propuesto.

No es el propósito del presente trabajo efectuar una comparación de tiempos de ejecución con otros algoritmos evolutivos. No obstante, teniendo en cuenta las experiencias llevadas a cabo por los autores con algoritmos genéticos [12, 13, 14], se destaca que la puesta a punto resulta ser notablemente más rápida para PSO. Esta característica es importante para medio ambientes industriales.

5. Conclusiones

En este trabajo se presenta un método basado en un algoritmo PSO para realizar el diseño de filtros activos. Se considera como caso de estudio un filtro Sallen-Key pasabanda de segundo orden, planteando un escenario posible para los valores que pueden adoptar las resistencias y los capacitores. El algoritmo debe determinar los valores de los componentes pasivos del filtro con el objetivo de obtener una configuración lo más robusta posible a las variaciones en los mismos, mediante la minimización de las sensibilidades. Además, debe considerarse que los errores de diseño deben ser menores a un valor máximo, con respecto a determinadas especificaciones. Los resultados de simulación muestran que PSO permite obtener configuraciones de filtro que, además de satisfacer las restricciones planteadas, presentan sensibilidades bajas. Por otro lado, estos algoritmos proveen, debido a su naturaleza estocástica, diferentes alternativas de diseño y emplean tiempos de ejecución considerablemente más bajos que los de la búsqueda exhaustiva.

Referencias

1. Dimopoulos, H.: Analog Electronics Filters: Theory, Design and Synthesis. Springer (2012)
2. Raut, R., Swamy, M. N. S.: Modern Analog Filter Analysis and Design: A Practical Approach. Wiley-VCH (2010).
3. Vural, R. A., Yildirim, T., Kadioglu, T., Basargan, A.: Performance evaluation of evolutionary algorithms for optimal filter design. IEEE Trans. on Evolutionary Computation, vol. 16, no.1, pp. 135-147 (2012).
4. El-Ghazali Talbi: Metaheuristics From Design to Implementation, Wiley (2009).
5. Yu, X. and Gen, M.: Introduction to Evolutionary Algorithms, Springer (2010).
6. Parsopoulos, K. E., Vrahatis, M. N.: Particle Swarm Optimization and Intelligence: Advance and Applications. Information Science Reference. (2010). EUA.
7. 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 on (pp. 1-4). IEEE (2010).
8. 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).
9. 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), 621-636 (2015).
10. 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), 593-609 (2015).
11. 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), 707-737 (2015).
12. Lovay, M., Peretti, G., Romero, E., Marqués, C.: Tolerancia a fallas en un filtro de alto orden mediante una estrategia de hardware evolutivo, in Proc. Fourth Congress of Applied Microelectronics, Bahía Blanca, Argentine, pp. 140-145 (2013).
13. Lovay, M., Peretti, G., Romero, E., Marqués, C.: Comparación experimental de algoritmos genéticos en un filtro tolerante a fallas, in Proc. Third Congress of Applied Microelectronics, Rosario, Argentine (2012).
14. Lovay, M., Peretti, G., Romero, E.: Implementation of an Adaptive Filter using an Evolvable Hardware Strategy, IEEE Latin America Transactions, vol.13, issue 4, pp. 927-934 (2015).
15. Pulido, G. T., Coello, C.: A constraint-handling mechanism for particle swarm optimization. En Evolutionary Computation, 2004. CEC2004. Congress on. IEEE, 2004. p. 1396-1403.
16. Gang, Ma, Wei, Zhou, Xiaolin, Chang: A novel particle swarm optimization algorithm based on particle migration. Applied Mathematics and Computation, vol. 218, no 11, p. 6620-6626 (2012).
17. Perez, R. E., Behdinan, K.: Particle swarm approach for structural design optimization. Computers & Structures, 2007, vol. 85, no 19, p. 1579-1588.