

Algoritmo de Optimización por Enjambre de Partículas Aplicado en la Selección de Componentes de Filtros Bicuadráticos

Mónica Lovay, Gabriela Peretti, Eduardo Romero

Grupo de Estudios en Calidad en Mecatrónica

Facultad Regional Villa María, Universidad Tecnológica Nacional

Villa María, Argentina

gecam@frvm.utn.edu.ar

Abstract

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 de componentes discretos en filtros bicuadráticos. 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 presentado, se propone su aplicación en un filtro bicuadrático de variable de estado (KHN) pasabajo adoptado como caso de estudio. Los resultados muestran que PSO permite obtener configuraciones de filtro con sensibilidades bajas y errores en los parámetros funcionales que satisfacen las especificaciones consideradas. Por otro lado, este algoritmo provee, debido a su naturaleza estocástica, diferentes alternativas de diseño y emplea tiempos de ejecución considerablemente más bajos que los de la búsqueda exhaustiva.

1. Introducción

Los filtros bicuadráticos se encuentran presentes en numerosas aplicaciones de procesamiento de señales. Una de las principales ventajas de este tipo de filtros radica en que pueden ser conectados en cascada, permitiendo el diseño de filtros de orden superior. Su implementación se efectúa mediante dispositivos activos embebidos en un arreglo de componentes pasivos (resistencias y capacitores). Dada una determinada especificación de filtro, el dimensionamiento de dichos componentes resulta ser una tarea compleja. Este proceso comprende la selección de los valores de las resistencias y capacitores que permiten al filtro cumplir con determinadas especificaciones. No obstante, es necesario

considerar que estos valores deben estar contenidos en alguna de las series industriales E. Estas series contienen valores normalizados de resistencias y capacitores. Los fabricantes combinan los valores de las series E con la tolerancia de los componentes.

En el enfoque convencional de diseño, generalmente se proponen valores iguales para algunos componentes [1-3]. 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 diseño.

Con el propósito de evitar los problemas ocasionados por el método convencional, una alternativa 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 contexto, se necesitan métodos de búsqueda inteligente que logren cumplir los objetivos en tiempos computacionales aceptables.

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 [4]. Este método es empleado para resolver diferentes problemas de optimización [5, 6, 7].

Algunos autores han propuesto la utilización de PSO para efectuar el dimensionamiento de componentes en filtros bicuadráticos. Particularmente, en [8] 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 [9] para realizar

el diseño de un filtro Butterworth pasabajo de cuarto orden y del filtro de segundo orden adoptado en [8]. Tres versiones mejoradas del algoritmo PSO: Simplex PSO, CRPSO y ALC-PSO son propuestas en [10-12], respectivamente, para efectuar el diseño de filtros considerando las dos topologías utilizadas en [9]. 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 [10] y [12] 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.

En este trabajo, se presenta un método basado en el algoritmo PSO para realizar el dimensionamiento de componentes discretos en filtros bicuadráticos. El método propuesto lleva a cabo el proceso de selección de los valores de las resistencias y capacitores con el propósito de obtener configuraciones de filtro que permitan minimizar las sensibilidades y a la vez cumplir con los objetivos de diseño relacionados a los parámetros de desempeño. Además, el algoritmo limita su espacio de búsqueda solamente a configuraciones de filtro cuyos valores de resistencias y capacitores 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 formulado, se propone su aplicación para seleccionar los valores de los componentes pasivos de un filtro bicuadrático de variable de estado.

2. Filtros Bicuadráticos

Los filtros bicuadráticos, también referenciados como biquads, implementan funciones de transferencia de segundo orden. El filtro de variable de estado, también conocido como KHN, es frecuentemente utilizado para implementar biquads. Este tipo de filtro, cuyo diagrama de bloques se muestra en la Figura 1, provee tres salidas útiles: una pasabajo, una pasalto y una pasabanda. En este trabajo se considera que la salida pasabajo es la salida deseada (V_{o3} , en la Figura 1). Los filtros KHN suelen presentar sensibilidades bajas, característica que los hace muy atractivos, sobre todo en aplicaciones exigentes [2].

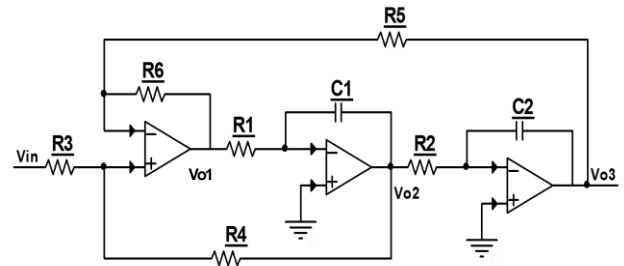


Figura 1. Filtro bicuadrático KHN.

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 (1).

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

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

$$G = \frac{1 + R_6/R_5}{1 + R_3/R_4} \frac{R_5}{R_6} \quad (2)$$

$$\omega_p = \sqrt{\frac{R_6/R_5}{R_1 R_2 C_1 C_2}} \quad (3)$$

$$Q_p = \left[\frac{1 + R_6/R_5}{1 + R_4/R_3} \sqrt{\frac{R_5 R_2 C_2}{R_6 R_1 C_1}} \right]^{-1} \quad (4)$$

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:

$$S_{x_i}^P = \frac{\% \text{ cambio en } P}{\% \text{ cambio en } x_i} = \frac{\partial P/P}{\partial x_i/x_i} \quad (5)$$

Se considera que un filtro tiene baja sensibilidad cuando todas sus sensibilidades adoptan valores inferiores a 1 [3].

Para el filtro KHN tomado como caso de estudio, las sensibilidades de ωp y Qp con respecto a los componentes pasivos son:

$$S_{R_1}^{Qp} = S_{C_1}^{Qp} = -S_{R_2}^{Qp} = -S_{C_2}^{Qp} = 1/2 \quad (6)$$

$$-S_{R_4}^{Qp} = -1/\{1 + (R_3/R_4)\} \quad (7)$$

$$S_{R_5}^{Qp} = -S_{R_6}^{Qp} = -\frac{Qp}{2} \frac{R_5 - R_6}{(1 + (R_4/R_3))} \sqrt{\frac{R_2 C_2}{R_5 R_6 R_1 C_1}} \quad (8)$$

$$S_{R_1}^{\omega p} = S_{R_2}^{\omega p} = S_{R_5}^{\omega p} = S_{C_1}^{\omega p} = S_{C_2}^{\omega p} = -S_{R_6}^{\omega p} = -(1/2)(9)$$

$$S_{R_3}^{\omega p} = S_{R_4}^{\omega p} = 0 \quad (10)$$

Es posible apreciar que en (6), (9) y (10) las sensibilidades adoptan valores fijos. Por otro lado, en (7) y (8) los valores de sensibilidad de Qp con respecto a variaciones en R_3 , R_4 , R_5 y R_6 son dependientes de los valores que posean los componentes del filtro. En consecuencia, si se desea diseñar un filtro KHN lo más robusto posible con respecto a las variaciones en los valores de sus componentes, las sensibilidades expresadas en (7) y (8) deben ser consideradas en el momento de la selección de estos valores.

En la Tabla 1 se presentan las especificaciones elegidas para el filtro bicuadrático KHN. Es necesario destacar que los métodos convencionales de diseño de filtros sólo permiten fijar los valores deseados para Qp y ωp , siendo el valor de G calculado a partir de ωp [1-3]. En este caso de estudio, se proponen valores

independientes para las tres especificaciones del filtro. Por este motivo, los métodos convencionales no pueden ser aplicados para efectuar el dimensionamiento de los componentes pasivos del filtro bajo estudio.

Tabla 1. Especificaciones elegidas para el filtro KHN.

ESPECIFICACIÓN	VALOR
G	3
ωp	$1000 * 2 * \pi = 6283,1853 \text{ rad/s}$
Qp	0,707

3. Selección de componentes usando 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 que dependen de los mismos sean mínimas, y los errores en G , ωp y Qp sean inferiores a un error máximo especificado. El algoritmo debe resolver un problema de optimización multiobjetivo (POM) que puede representarse mediante la siguiente expresión:

$$\begin{aligned} \min \mathbf{S}(i) &= S_1(i), S_2(i), \dots, S_n(i) \\ \text{sujeto a: } Error_G(i) &\leq E_{maxG} \\ Error_{\omega p}(i) &\leq E_{max\omega p} \\ Error_{Qp}(i) &\leq E_{maxQp} \end{aligned} \quad (11)$$

En (11), i es una posible configuración de filtro determinada por los valores de resistencias y capacitores. $\mathbf{S}(i)$ es el conjunto de 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 de acuerdo a las expresiones (12) a (14). E_{maxG} , $E_{max\omega p}$ y E_{maxQp} definen el error máximo tolerable para cada una de las características del filtro.

$$Error_G(i) = |(G(i) - G_F)/G_F| \quad (12)$$

$$Error_{\omega p}(i) = |(\omega p(i) - \omega p_F)/\omega p_F| \quad (13)$$

$$Error_{Qp}(i) = |(Qp(i) - Qp_F)/Qp_F| \quad (14)$$

3.2. Aplicación de PSO al problema de optimización

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) [4-7]. El diagrama de flujo de la Figura 2 muestra las operaciones realizadas por el algoritmo PSO implementado, el cual ha sido desarrollado en Matlab™.

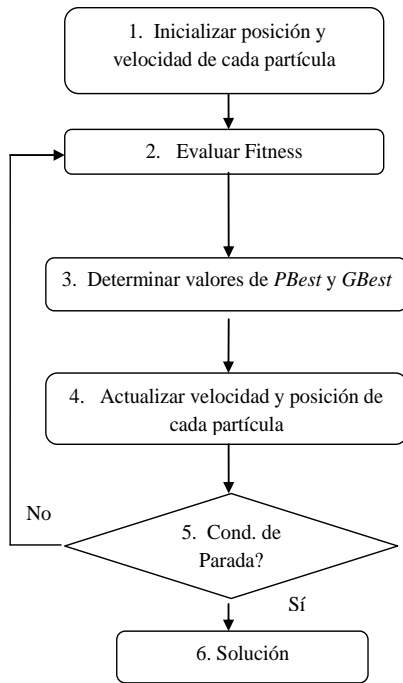


Figura 2. Diagrama de flujo de PSO.

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 (15) y (16), representan, respectivamente, la posición y velocidad de cada partícula i , en un espacio de búsqueda que en este caso es de ocho dimensiones:

$$x_i = (x_{iR1}, x_{iR2}, x_{iR3}, x_{iR4}, x_{iR5}, x_{iR6}, x_{iC1}, x_{iC2}) \quad (15)$$

$$v_i = (v_{iR1}, v_{iR2}, v_{iR3}, v_{iR4}, v_{iR5}, v_{iR6}, v_{iC1}, v_{iC2}) \quad (16)$$

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 (11) 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 [12-14]. De esta manera, la función de *fitness* f que asigna un valor de aptitud a cada partícula i , se define mediante la expresión (17).

$$f(i) = (w_1 |S_{R_3}^{Qp}(i)| + w_2 |S_{R_4}^{Qp}(i)| + w_3 |S_{R_5}^{Qp}(i)| + w_4 |S_{R_6}^{Qp}(i)|) \quad (17)$$

En (17), 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 (1/4) para todas las sensibilidades.

PSO aplica las restricciones, penalizando en cada iteración a aquellas partículas que no las cumplen, mediante la expresión (18) [16]. En consecuencia, éstas adoptan un valor de fitness más alto (f_h) de acuerdo al grado de violación de cada restricción.

$$f_h(i) = f(i) + \frac{|E_{maxG} - Error_{G,i}|}{Error_{G,max}} + \frac{|E_{maxop} - Error_{op,i}|}{Error_{op,max}} + \frac{|E_{maxQp} - Error_{Qp,i}|}{Error_{Qp,max}} \quad (18)$$

En (18), $Error_{G,i}$, $Error_{op,i}$, $Error_{Qp,i}$ 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 de la generación actual y lo almacena en $GBest_i^k$.

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 $GBest_i^k$. La velocidad v_i^{k+1} de cada partícula y su nueva posición x_i^{k+1} se calculan utilizando, respectivamente, las siguientes ecuaciones [3]:

$$v_i^{k+1} = \underbrace{\phi v_i^k}_{\text{Inercia}} + \underbrace{c_1 r_{1,i} (PBest_i^k - x_i^k)}_{\text{Influencia cognitiva}} + \underbrace{c_2 r_{2,i} (GBest_i^k - x_i^k)}_{\text{Influencia social}} \quad (19)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (20)$$

En (19), 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 $GBest_i^k$, 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 (19) 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 las mejores soluciones 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, (21) representa la función utilizada en este caso de estudio para decrementar el valor de la masa inercial ϕ en el proceso evolutivo [17-18].

$$\phi = \phi_{max} - \frac{\phi_{max} - \phi_{min}}{K_{max}} k \quad (21)$$

En (21), 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 decreta 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.

3.3. Parámetros de PSO

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 su desempeño. Teniendo en cuenta [8-12, 17-18], se efectúan simulaciones considerando diferentes combinaciones de valores para los mismos y se seleccionan aquellos con los que el algoritmo obtiene el mejor performance. Como PSO debe efectuar la minimización de la función fitness, los valores elegidos para cada parámetro son los que permiten al mismo alcanzar el valor más bajo de fitness. En este caso de estudio, los valores seleccionados son: $N=25$, $c_1=2$, $c_2=2$, $\phi_{max}=0,9$, $\phi_{min}=0,4$, $K_{max}=5000$.

Con fines demostrativos se establece un valor de 0,025 (2,5%) para el error máximo permitido en cada uno de los parámetros funcionales del filtro (E_{maxG} , E_{maxOp} y E_{maxQp}).

Por otro lado, se considera que las resistencias pueden adoptar valores en el rango 10^3 - $10^6\Omega$, de acuerdo a la serie E24. Con respecto a los capacitores el rango definido es 10^{-9} - $10^{-6}F$, de acuerdo a la serie E12. 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, el algoritmo dispone de 72 valores posibles para cada resistencia y 36 valores posibles para cada capacitor. El espacio de búsqueda total asciende a 1,81E14 configuraciones.

4. Resultados experimentales

Para evaluar el desempeño del método de diseño propuesto se consideran las sensibilidades y los errores en los parámetros de desempeño de las configuraciones de filtro obtenidas por PSO para el caso de estudio presentado.

Debido a que el algoritmo realiza un proceso estocástico, los resultados pueden variar de acuerdo a la distribución estadística de la población inicial. Con el propósito de verificar si PSO encuentra soluciones favorables independientemente del seteo que se efectúe para la población inicial, se realizan 100 corridas del mismo cambiando la semilla en la generación aleatoria de la población inicial de cada corrida.

En la Figura 3 se reportan los valores de sensibilidad pertenecientes a la configuración de filtro alcanzada por PSO en cada corrida. En el gráfico se visualizan las sensibilidades minimizadas por el algoritmo, es decir $|S_{R_3}^{Qp}|$, $|S_{R_4}^{Qp}|$, $|S_{R_5}^{Qp}|$ y $|S_{R_6}^{Qp}|$, las cuales adoptan en todos los casos valores inferiores a 1. Es posible observar que las sensibilidades de Qp respecto de R_3 y R_4 varían en un rango superior que las restantes. Por otro lado, la figura evidencia que en las configuraciones donde $|S_{R_3}^{Qp}|$ y $|S_{R_4}^{Qp}|$ alcanzan los valores más altos, $|S_{R_5}^{Qp}|$ y $|S_{R_6}^{Qp}|$ adoptan los valores más bajos, y viceversa. Esta particularidad revela que para esta topología de filtro existe una dualidad entre ambos grupos de sensibilidades. La característica mencionada puede ser visualizada de manera más clara en la Figura 4 donde se presentan los valores obtenidos para $|S_{R_3}^{Qp}|$ y $|S_{R_4}^{Qp}|$ contra los valores alcanzados para $|S_{R_5}^{Qp}|$ y $|S_{R_6}^{Qp}|$ para cada configuración de filtro encontrada como solución por PSO. Cada configuración obtenida está representada por un punto en el gráfico. Si se comparan dos configuraciones cualesquiera es posible verificar que si una de ellas posee para un grupo de sensibilidades un valor más bajo que la otra, indefectiblemente presenta un valor más alto para el otro grupo de sensibilidades, y viceversa.

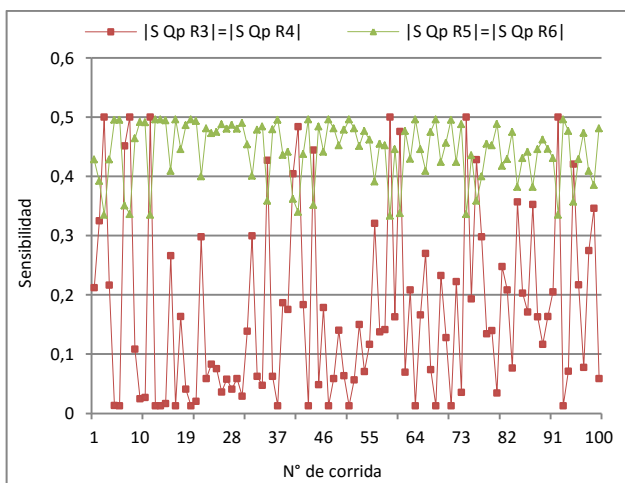


Figura 3. Sensibilidades correspondientes a las configuraciones de filtro encontradas por PSO en 100 corridas.

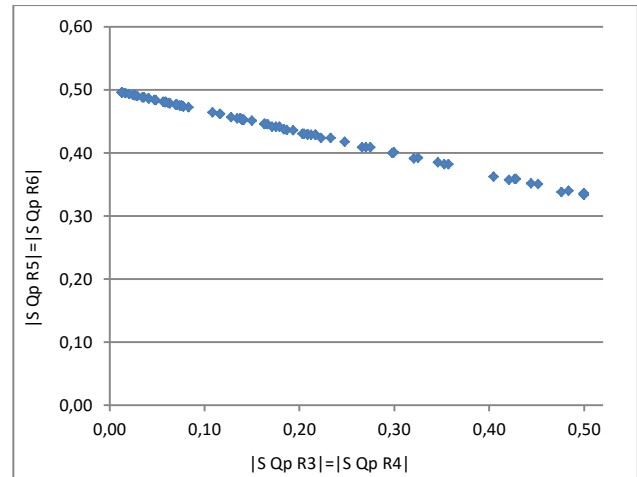


Figura 4. Valores de $|S_{R_3}^{Qp}| = |S_{R_4}^{Qp}|$ contra valores de $|S_{R_5}^{Qp}| = |S_{R_6}^{Qp}|$ para cada configuración de filtro encontrada por PSO.

En las Figuras 5, 6 y 7 se visualizan los valores que adoptan los componentes pasivos del filtro en cada una de las configuraciones encontradas por PSO en las 100 corridas efectuadas. Particularmente, en la Figura 5 se presentan los valores obtenidos para R_1 , R_2 y R_6 , en la Figura 6 se muestran los valores alcanzados para R_3 , R_4 y R_5 y en la Figura 7 se reportan los valores de C_1 y C_2 . Las figuras demuestran que en cada corrida se encuentran soluciones conformadas por diferentes combinaciones de valores de los componentes. De esta manera, el método propuesto proporciona al diseñador múltiples configuraciones de filtro permitiéndole elegir la más apropiada de acuerdo a sus recursos y necesidades particulares.

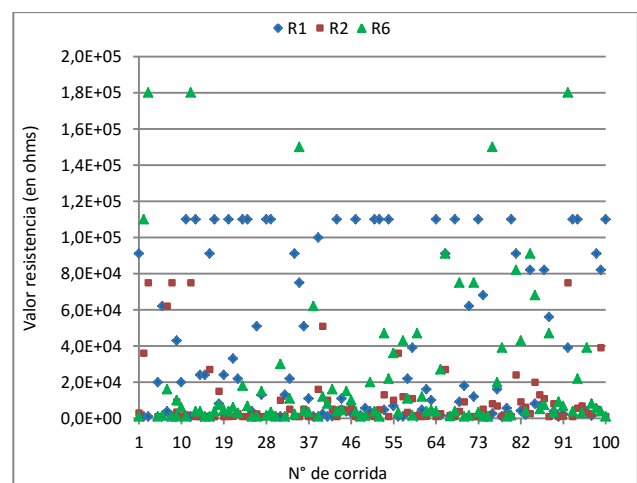


Figura 5. Valores que adoptan las resistencias R_1 , R_2 y R_6 en las soluciones encontradas por PSO.

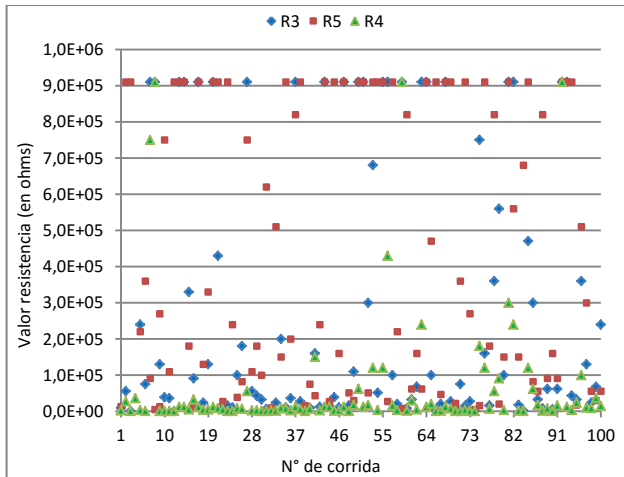


Figura 6. Valores que adoptan las resistencias R_3 , R_4 y R_5 en las soluciones encontradas por PSO.

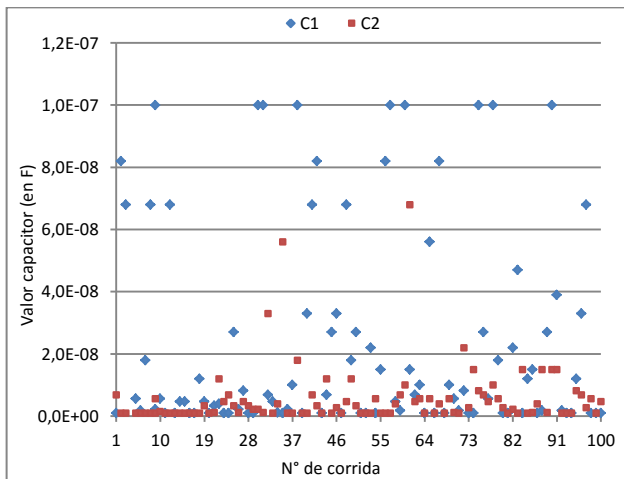


Figura 7. Valores que adoptan los capacitores C_1 y C_2 en las soluciones encontradas por PSO.

Para completar el análisis de resultados, en la Tabla 2 se presenta una caracterización de los valores obtenidos para las sensibilidades minimizadas, los errores en los parámetros de desempeño y el fitness de las configuraciones de filtro encontradas por PSO en las corridas realizadas. En la tabla se visualiza el mínimo, el máximo y la mediana. Esta medida de tendencia central es adoptada debido a que los datos presentan una distribución no normal. Con referencia a las sensibilidades, los valores de $|S_{R_3}^{Qp}|$ y $|S_{R_4}^{Qp}|$ están comprendidos en el rango $[0,0130; 0,5]$, mientras que los pertenecientes a $|S_{R_5}^{Qp}|$ y $|S_{R_6}^{Qp}|$ se encuentran en un rango menor, $[0,3333; 0,4957]$. Para la mediana, se observa que $|S_{R_3}^{Qp}|$ y $|S_{R_4}^{Qp}|$ presentan un valor más bajo que las dos restantes sensibilidades. Con respecto a los errores en los

parámetros de desempeño, los valores máximos son inferiores a 2,5% en todos los casos, lo cual significa que todas las configuraciones de filtro encontradas por PSO cumplen con las restricciones enunciadas en (11). Para los errores mínimos, $Error_{Qp}$ alcanza el valor cero, mientras que los otros dos errores son ligeramente superiores. Por otro lado, la mediana adopta el valor más bajo para $Error_{Qp}$, y el más alto para $Error_G$. Con referencia al fitness, el rango de variación del mismo es $[0,2543; 0,4180]$, siendo la mediana levemente más alta que el mínimo.

Tabla 2. Caracterización de las configuraciones encontradas por PSO.

	MÍNIMO	MÁXIMO	MEDIANA
$ S_{R_3}^{Qp} = S_{R_4}^{Qp} $	0,0130	0,5000	0,1393
$ S_{R_5}^{Qp} = S_{R_6}^{Qp} $	0,3333	0,4957	0,4531
$ERROR_G$	0,0653%	2,4904%	1,5045%
$ERROR_{op}$	0,0089%	2,4735%	1,4152%
$ERROR_{Qp}$	0%	2,4988%	1,3012%
FITNESS	0,2543	0,4180	0,2964

Con el propósito de complementar la caracterización de los resultados obtenidos utilizando el método basado en PSO, se visualiza en la Figura 8, la evolución del mejor fitness alcanzado por PSO a través de las generaciones en cuatro corridas seleccionadas al azar. La figura muestra que en todos los casos el mejor fitness mejora rápidamente en las primeras generaciones mientras el algoritmo explora el espacio de búsqueda, es decir, trata de localizar zonas prometedoras dentro del mismo. Con el transcurso de las generaciones, las mejoras se van atenuando cada vez más, debido a que en esta etapa el algoritmo intenta explotar el espacio de búsqueda, realizando una búsqueda en profundidad sobre las zonas prometedoras localizadas, con el objetivo de encontrar en ellas la mejor solución.

Con respecto a los tiempos de ejecución, PSO emplea como máximo 5,4584 segundos, con una mediana de 4,5430 segundos para efectuar cada corrida. Se estima que si el método de búsqueda exhaustiva es codificado en MatlabTM demandaría aproximadamente 7,84E8 segundos para encontrar la mejor solución. Esta estimación se realiza por extrapolación de los resultados obtenidos mediante búsqueda exhaustiva en un espacio de búsqueda reducido. PSO encuentra una solución favorable empleando tiempos computacionales considerablemente más bajos, lo cual justifica la utilización de la estrategia propuesta.

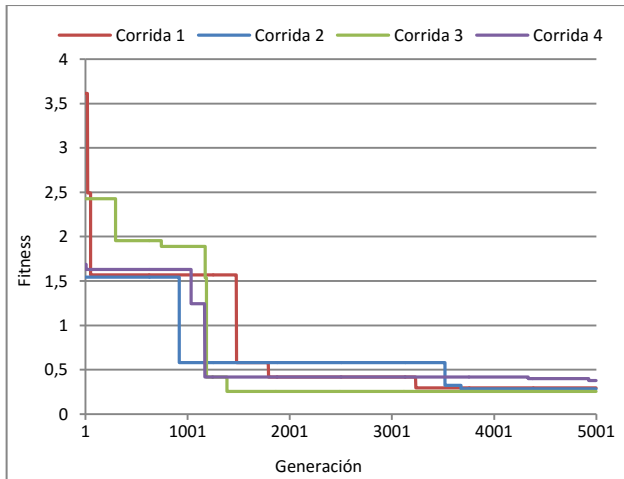


Figura 8. Evolución del mejor fitness alcanzado por PSO en cuatro corridas.

En la Tabla 3 se presentan los valores de los componentes correspondientes a la configuración encontrada por PSO que presenta el fitness más bajo. En la misma se muestran también los valores obtenidos para las sensibilidades minimizadas por el algoritmo y los errores en los parámetros de desempeño.

Tabla 3. Características de la solución alcanzada por PSO que presenta el fitness más bajo.

PARÁMETRO	VALOR OBTENIDO
R_1	11000 Ω
R_2	1000 Ω
R_3	910000 Ω
R_4	12000 Ω
R_5	820000 Ω
R_6	3600 Ω
C_1	1,00E-08 F
C_2	1,00E-09 F
$ S_{R_3}^{QP} = S_{R_4}^{QP} $	0,0130
$ S_{R_5}^{QP} = S_{R_6}^{QP} $	0,4956
ERROR _G	0,7472%
ERROR _{op}	0,5468%
ERROR _{Qp}	0,8464%
FITNESS	0,2543

Finalmente, teniendo en cuenta las especificaciones se utiliza (1) para obtener la función de transferencia de un filtro al que denominamos filtro nominal. Este filtro no considera 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 Figura 9 se muestra la respuesta en frecuencia (magnitud y fase) del filtro nominal y la respuesta de la configuración de filtro alcanzada por PSO que presenta el valor más bajo de fitness (Tabla 3). En la figura es posible observar que la respuesta del filtro obtenido por el algoritmo y la respuesta del filtro nominal son muy similares. En la Figura 10, se visualiza específicamente la magnitud de las respuestas en frecuencia para la zona correspondiente a la banda de paso y comienzo de la banda de transición. Puede apreciarse que las diferencias entre ambas respuestas son mínimas.

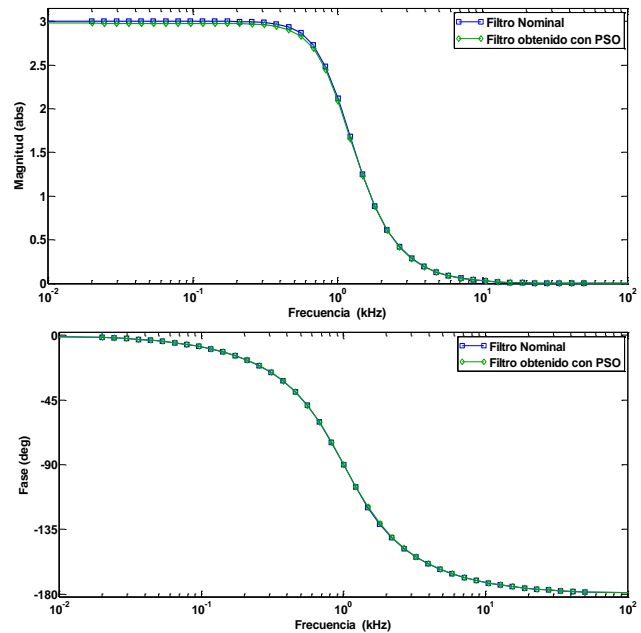


Figura 9. Respuesta en frecuencia (magnitud y fase) del filtro nominal y de la configuración de filtro alcanzada por PSO que presenta el valor más bajo de fitness.

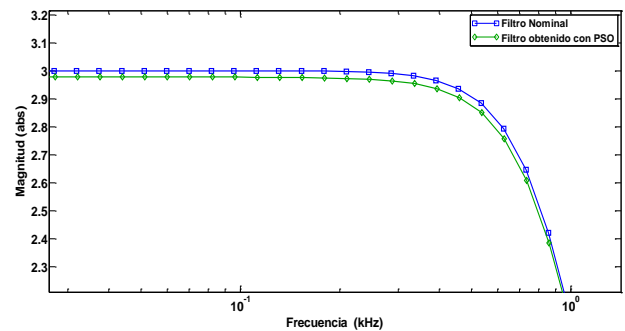


Figura 10. Magnitud de la respuesta en frecuencia del filtro nominal y de la configuración de filtro alcanzada por PSO en la zona de la banda de paso y comienzo de la banda de transición.

5. Conclusiones

En este trabajo se propone un método basado en el algoritmo PSO para efectuar la selección de componentes discretos en filtros bicuadráticos. Se considera como caso de estudio un filtro KHN pasabajo. 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 tenerse en cuenta que los errores en los parámetros de desempeño deben ser menores a un valor máximo, con respecto a determinadas especificaciones. Los resultados muestran que PSO permite obtener configuraciones de filtro que presentan sensibilidades bajas, además de satisfacer las restricciones planteadas. PSO, debido a su naturaleza estocástica, proporciona al diseñador diferentes alternativas de diseño en cada corrida. Con referencia a los tiempos de ejecución, el algoritmo resulta ser considerablemente más rápido que el método de búsqueda exhaustiva.

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

6. Referencias

- [1] Dimopoulos, H., *Analog Electronics Filters: Theory, Design and Synthesis*, Springer, 2012.
- [2] Raut, R. and Swamy, M. N. S., *Modern Analog Filter Analysis and Design: A Practical Approach*, Wiley-VCH, 2010.
- [3] Vural, R. A., Yildirim, T., Kadioglu, T. and 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] Kennedy J. and Eberhart R., "Particle Swarm Optimization". In Proc. of IEEE International Conference on Neural Networks (ICNN), Australia, p. 1942-1948, 1995.
- [5] El-Ghazali Talbi, *Metaheuristics From Design to Implementation*, Wiley, 2009.
- [6] Yu, X. and Gen, M., *Introduction to Evolutionary Algorithms*, Springer 2010.
- [7] Parsopoulos, K. E. and Vrahatis, M. N., *Particle Swarm Optimization and Intelligence: Advance and Applications*. Information Science Reference, 2010, EUA.
- [8] Vural, R. A. and 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*.
- [9] Vural, R. A., Yildirim, T., Kadioglu, T. and Basargan, A., "Performance evaluation of evolutionary algorithms for optimal filter design". *Evolutionary Computation, IEEE Transactions on*, 16(1), 135-147, 2012.
- [10] De, P., Kar, R., Mandal, D., and 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.
- [11] De, P., Kar, R., Mandal, D., and 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.
- [12] De, P., Kar, R., Mandal, D., and 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.
- [13] Lovay, M., Peretti, G., Romero, E., and 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.
- [14] Lovay, M., Peretti, G., Romero, E., and Marqués, C., "Comparación experimental de algoritmos genéticos en un filtro tolerante a fallas", in *Proc. Third Congress of Applied Microelectronics*, Rosario, Argentine, 2012.
- [15] Lovay, M., Peretti, G., and Romero, E., "Implementation of an Adaptive Filter using an Evolvable Hardware Strategy", *IEEE Latin America Transactions*, vol.13, issue 4, pp. 927-934, 2015.
- [16] Pulido, G. T. and Coello, C., "A constraint-handling mechanism for particle swarm optimization". *En Evolutionary Computation*, 2004. CEC2004. Congress on. IEEE, p. 1396-1403, 2004.
- [17] Gang, M., Wei, Z., and Xiaolin, C., "A novel particle swarm optimization algorithm based on particle migration". *Applied Mathematics and Computation*, vol. 218, no 11, p. 6620-6626, 2012.
- [18] Perez, R. E. and Behdinan, K., "Particle swarm approach for structural design optimization". *Computers & Structures*, vol. 85, no 19, p. 1579-1588, 2007.