

Diseño de filtros analógicos de capacidades conmutadas con Optimización por Enjambre de partículas

Switched-capacitor analog filter design with Particle Swarm Optimization

Presentación: 11/10/2019

Doctorando:

Gonzalo Tomás VodanovicGrupo de Estudio en Calidad Mecatrónica (UTN-FRVM) y Grupo de Desarrollo Electrónico e Instrumental (UNC-FAMAF)
gvodanovic@famaf.unc.edu.ar

Director/es:

Dr. Eduardo Romero**Dra. Gabriela Peretti**

Resumen

En este trabajo se aplica el método Optimización por Enjambre de Partículas para el diseño de filtros analógicos de capacidades conmutadas embebidos en la plataforma configurable de señales mixtas PSoC1 de *Cypress Semiconductor*. Se realiza una evaluación intensiva del algoritmo para un amplio rango de filtros. Los resultados obtenidos muestran que el optimizador es capaz de encontrar al menos una solución que cumple con las especificaciones de diseño para todos los casos evaluados.

Palabras claves: Optimización por enjambre de partículas, filtro, capacidades conmutadas, biquad, PSoC1.

Abstract

In this work the Particle Swarm Optimization algorithm is used for the design of switched-capacitor analog filter embedded in the mixed signal configurable platform PSoC1 of the firm Cypress Semiconductor. An intensive evaluation of the algorithm is performed for a wide range of filter specifications. The result obtained show that the optimizer is able to find at least one solution that meets the design objective for every case evaluated

Keywords: Particle Swarm Optimization, filter, switched-capacitor, biquad, PSoC1

Introducción

Los filtros activos son utilizados intensivamente en sistemas electrónicos que requieren manipulación de señales en el dominio de la frecuencia. Su aplicación en diferentes campos ha motivado, tanto a las comunidades académica como industrial, al desarrollo de topologías de circuitos capaces de sintetizar respuestas conocidas como, por ejemplo, Chebyshev, Butterworth y Bessel. Los filtros biquadráticos o biquads, son ampliamente utilizados en la industria debido, principalmente, a la facilidad que ofrecen para construir filtros de alto orden al conectarse etapas de segundo orden en cascada. Los biquads pueden implementarse utilizando diferentes técnicas de circuitos. Un caso muy interesante lo constituyen los filtros de capacidades conmutadas (SC, Switched Capacitors), que serán adoptados como caso de estudio en este trabajo. Estos filtros pueden ser implementados con bancos de capacitores programables, relativamente fáciles de implementar. Las diversas configuraciones de estos bancos permiten obtener, de la misma estructura, diferentes respuestas, otorgando gran flexibilidad al circuito. Este tipo de estructuras pueden encontrarse en algunas plataformas configurables analógicas y de señales mixtas [1] [2] [3]. Existe también la alternativa de que el diseñador conciba un diseño propio con arreglos de capacidades conmutadas acordes a sus necesidades específicas.

La literatura ofrece procedimientos de diseño para biquads (referidos de aquí en más como convencionales) que facilitan el proceso, recurriendo a simplificaciones, pero restringiendo los grados de libertad para lograr la satisfacción de las especificaciones [4] [5] [6] [7]. El procedimiento debe brindar los valores de componentes (para una topología dada) que permiten el cumplimiento de los objetivos de diseño. Sin embargo, el conjunto de valores de componentes es finito, lo cual hace que el resultado final sea solo una aproximación a lo deseado. En general, es necesario iterar el diseño, volviendo al procedimiento convencional costoso y muchas veces inconveniente. Adicionalmente, si la aplicación requiere la

reprogramación en campo para cumplir con diferentes requisitos de filtrado durante la operación del sistema, se hace necesario aplicar métodos computacionales de diseño que permitan resolver el problema de una manera eficiente. Resulta obvia la necesidad de buscar alternativas a los métodos tradicionales de diseño. Una alternativa es recurrir a métodos heurísticos, capaces de brindar una solución casi óptima en tiempos razonablemente cortos. Otra ventaja de estas alternativas computacionales de diseño radica en que pueden obtenerse respuestas diferentes a las realizables con los métodos convencionales, ampliándose el rango de filtros factibles (que cumplen los objetivos de diseño).

En este trabajo se aplica el método denominado Optimización por Enjambre de Partículas (PSO, Particle Swarm Optimization) para efectuar el diseño de un filtro de capacidades conmutadas. Si bien la formulación del algoritmo de optimización es general y puede extenderse a diferentes topologías con diversos bancos de capacitores, en este trabajo se ha adoptado como estructura a la disponible en la plataforma PSOC 1 de Cypress Semiconductor. La elección se fundamenta en la posibilidad de obtener resultados experimentales a futuro. Adicionalmente, estas plataformas permiten ser utilizadas en aplicaciones en las que se requiere cambiar la función del hardware durante la operación en campo. En este sentido, la aplicación de los métodos aquí explorados puede hacer viable la reconfiguración para cumplir con distintas especificaciones de filtros en aplicaciones como la referida.

Se plantea una evaluación intensiva del algoritmo, sobre un amplio abanico de posibilidades de diseño, diferente a las planteadas previamente por otros autores [8][9], en las cuales el desempeño de los algoritmos se evalúa en pocos casos de estudio. La exploración efectuada se orienta a la determinación de la habilidad del método para resolver los casos presentados. Los resultados obtenidos en el este trabajo son promisorios y permiten visualizar una aplicación del mismo en futuros trabajos.

Descripción de la plataforma

La familia PSoC1 cuenta con un microcontrolador embebido de 8 bits, recursos analógicos y digitales configurables agrupados en forma de bloques, y entradas/salidas globales configurables. Los bloques pueden acoplarse entre sí por medio de una red de interconexiones programable. La plataforma PSoC 1 puede utilizarse en una gran cantidad de aplicaciones, dada la flexibilidad de sus módulos configurables analógicos y digitales.

La arquitectura de este dispositivo, mostrada la figura 1, consiste de cuatro áreas principales: PSoC Core (1), Sistema Digital (2), Sistema Analógico (3) y Recursos del Sistema (4). Buses configurables globales permiten la combinación de los recursos del dispositivo en sistemas personalizables complejos.

El PSoC elegido como caso de estudio es el CY8C29466-PXI. Este dispositivo cuenta con 16 bloques digitales y 12 bloques analógicos. Los distintos EACCs (Embedded Analog Configurable Circuits) se separan en dos categorías: de tiempo continuo (CT, continuous time) y de capacidades conmutadas (SC, switched-capacitor). A su vez, los bloques SC se subdividen en dos tipos, C y D, los cuales se diferencian entre sí por la cantidad de elementos que los componen y las interconexiones posibles con otros bloques analógicos.

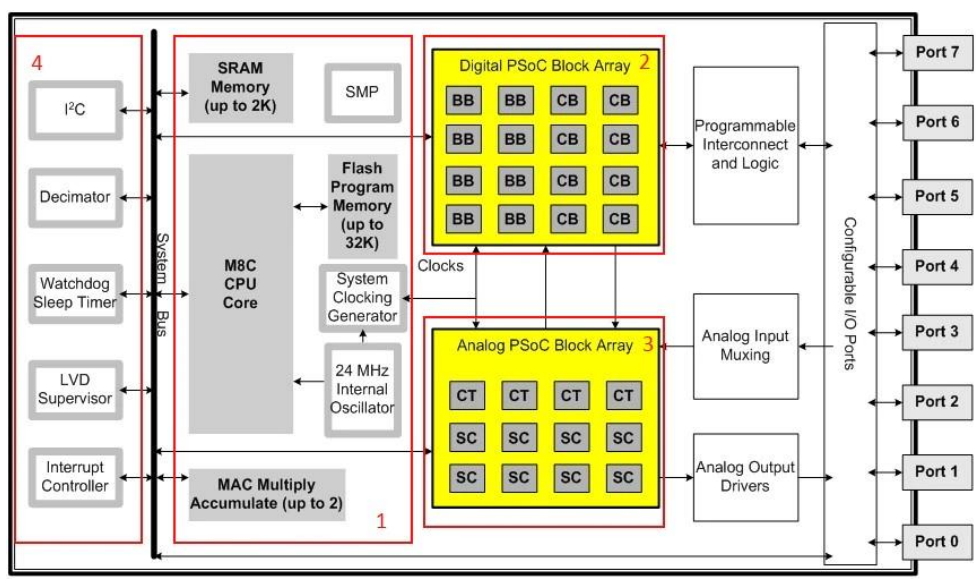


Figura 1: Arquitectura PSoC1

Los bloques analógicos pueden configurarse independientemente para implementar distintas funcionalidades, como amplificadores de ganancia programable, comparadores integradores, derivadores, o asociarse entre sí y con recursos digitales para implementar funcionalidades más complejas como filtros analógicos, conversores analógico - digital de hasta 14 bits, etc.

Filtros de capacidades conmutadas en PsoC1

Los recursos analógicos SC disponibles en este dispositivo cuentan con un amplificador operacional, capacitores y llaves programables de diferentes formas para realizar la conmutación de los capacitores. Existen también diferentes conexiones programables que le otorgan una gran versatilidad. En la figura 2 se muestra, a título de ejemplo, el esquema de uno de los bloques de capacidades conmutadas que se utilizará en este trabajo. Para conformar un filtro, es necesario interconectar dos bloques SC. La topología final del filtro, una vez programados los recursos de las celdas intervinientes, se muestra de manera simplificada en la figura 3. Como puede verse, el filtro cuenta con cuatro capacitores cuyo valor puede modificarse entre 32 posibilidades (0 a 31) y otros dos cuyos valores puede modificarse entre dos posibilidades (16 o 32). Estos valores, multiplicados por 70 da como resultado las capacidades de cada uno en pF. Considerando los 6 elementos a elegir, se determina que el espacio de búsqueda de la heurística es de 4.194.304 de combinaciones posibles.

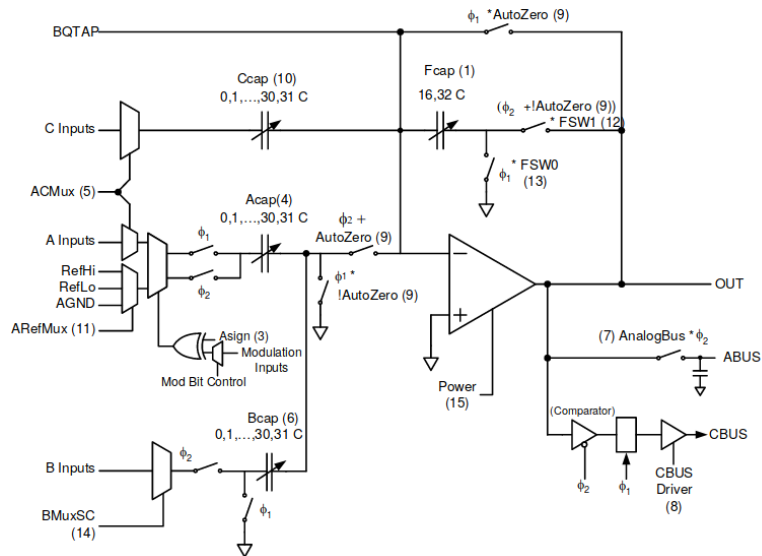


Figura 2: Esquema del bloque de capacidades conmutadas tipo C

A partir del esquema de la figura 3 se desarrolla la función de transferencia en dominio discreto y mediante la transformada bilineal, se obtiene en el dominio de Laplace la ecuación 1.

$$H(s) = -\frac{\frac{C_{PP}C_A}{C_2C_3}\left(\frac{s}{f_s}\right)^2 + \frac{C_P s}{C_2 f_s}\left(1 - \frac{s}{f_s}\right) + \frac{C_1}{C_2}\left(1 - \frac{1}{4}\left(\frac{s}{f_s}\right)^2\right)}{\left(\frac{C_B C_A}{C_2 C_3} - \frac{1}{2C_2} \frac{1}{4}\right)\left(\frac{s}{f_s}\right)^2 + \frac{C_4 s}{C_2 f_s} + 1} \quad (1)$$

Con esta topología, es posible diseñar filtros pasa bajo, pasa alto, pasa banda y rechaza banda. En el presente trabajo, se realizará el estudio de filtros pasa bajo, cuya función de transferencia se muestra en la ecuación 2.

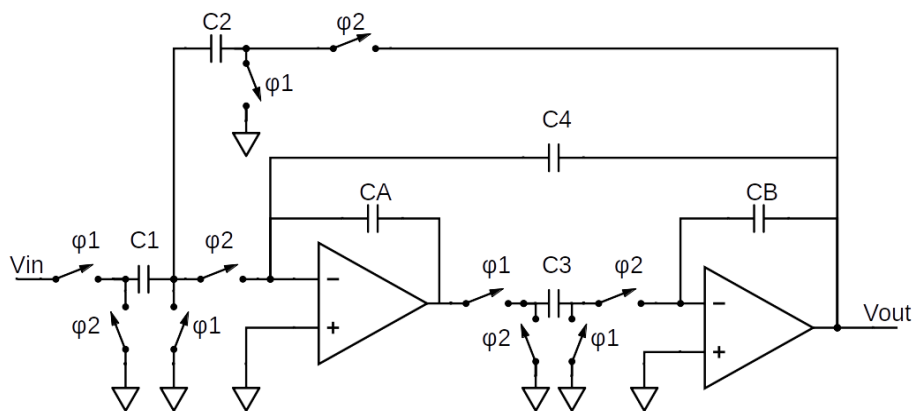


Figura 3: Filtro de capacidades conmutadas implementado en PSoC1

$$H(s) = \frac{K \cdot \omega_p}{\omega_p^2 + s \frac{\omega_p}{Q_p} + s^2} \quad (2)$$

En (2), K es la ganancia (G), $\omega_p = \frac{f_0}{2\pi}$ es la frecuencia de polo y $Q_p = \frac{1}{d}$ del factor de selectividad (siendo d es el factor de amortiguamiento). A partir de la ecuación 1, es posible obtener las ecuaciones que relacionan las especificaciones de un filtro con el valor de los capacitores seleccionados (ecuaciones 3) [10].

$$f_0 = \frac{f_s}{2\pi} \frac{\sqrt{C_2 C_3}}{\sqrt{(C_A C_B - \frac{1}{2} C_4 C_3 + \frac{1}{4} C_2 C_3)}} \quad d = \frac{c_4}{\sqrt{(C_A C_B - \frac{1}{2} C_4 C_3 + \frac{1}{4} C_2 C_3)}} \quad G = -\frac{c_1}{c_2} \quad (3)$$

Optimizador por enjambre de partículas

El optimizador por enjambre de partículas (PSO, Particle Swarm Optimization) es un método estocástico de optimización inspirado en el comportamiento social de animales que viven en grupos [11]. El comportamiento (velocidad y posición) de cada individuo (partícula) es el resultado de una combinación de influencias cognitivas, sociales y estocásticas. La función de *fitness* es una medida de la calidad de una partícula como resultado. En cada iteración, el algoritmo computa esta función para diferentes puntos del espacio de búsqueda multidimensional, utilizando un conjunto de partículas. Luego determina nuevas posiciones a explorar a partir de su conocimiento individual y del colectivo. Esto se repite hasta alcanzar el número máximo de iteraciones o hasta encontrar una solución adecuada.

En este caso en particular, la función de *fitness* evalúa el desempeño del filtro analógico respecto al deseado. Para cada partícula $K = [C1, C2, C3, C4, CA, CB]$, se calcula las especificaciones del filtro: frecuencia de corte (f_0), ganancia (G) y factor de amortiguamiento (d) con las ecuaciones 3. La función de *fitness* es una suma ponderada de los errores entre las especificaciones calculadas [f_0, d, G] y los valores deseados [$espec(f_0), espec(d), espec(G)$], como puede verse en la ecuación 4.

$$fitness = \frac{f_0 - espec(f_0)}{espec(f_0)} + \frac{d - espec(d)}{espec(d)} + \frac{G - espec(G)}{espec(G)} \quad (4)$$

Cuando el error en las tres especificaciones es menor al 5%, se la considera una solución válida y el proceso de optimización se finaliza. Luego de comprobar el desempeño del algoritmo, se decidió detener la optimización luego de 300 iteraciones como máximo. Debido a que existen algunos casos con soluciones difíciles de localizar por el algoritmo, se propone utilizar el mismo 100 veces sucesivas, cambiando la semilla para la generación de la población aleatoria inicial de partículas. En lo que sigue, se considerará como una corrida a 100 ejecuciones sucesivas del algoritmo de la forma antes descrita.

Existen diversas implementaciones de este optimizador, el presente trabajo utiliza la última versión, estandarizada en 2011 [12] denominada SPSO2011 [13]. Los hiperparámetros de este algoritmo se configuran de acuerdo a esta misma estandarización y se detallan en la tabla 1.

Tabla 1: Hiperparámetros del optimizador por enjambre de partículas

Número de partículas	48
Peso inercial	$1 / (2 \times \log(2))$
Aceleración Cognitiva	$0,5 + \log(2)$
Aceleración Social	$0,5 + \log(2)$

Resultados

Con el objeto de evaluar el optimizador en su amplio espacio de búsqueda, se implementaron filtros de diversas especificaciones. Se utilizaron 3 valores de ganancias, 3 factores de amortiguamiento y 200 frecuencias de polo distintas. Consecuentemente, se evalúa la capacidad del algoritmo con 1800 diseños diferentes. Dado el amplio rango de frecuencias de polo disponibles en el dispositivo, se seleccionó una muestra, de las mismas, con distribución logarítmica. Como se expresó anteriormente, esto implica una evaluación del espacio de diseño intensiva, que supera por mucho las presentadas anteriormente por otros autores.

Una ejecución exitosa del algoritmo implica que este encuentra por lo menos una solución viable. En nuestras evaluaciones se ha observado que todas las ejecuciones han sido satisfactorias, llevando la tasa de éxitos al 100%. Este es un resultado relevante ya que permite establecer un grado de confianza muy elevado sobre la robustez de la metodología aplicada para la resolución del problema bajo estudio.

En el gráfico de la figura 4, se muestra la distribución del número de iteraciones que el optimizador necesita para encontrar un filtro viable, considerando los 1800 filtros distintos y las 100 corridas para cada uno de ellos. Puede verse que en el 90% de los casos, el optimizador necesita 63 iteraciones o menos, mientras que la mediana es de 33 iteraciones. En una PC estándar (CPU: Intel i5-8265U, RAM: 8Gb, lenguaje de programación: Matlab) el optimizador requiere 0.002s, en promedio, para ejecutar una iteración, por lo que, para el 90% de los casos necesitaría menos de 126s. Por otro lado, para la mitad de los casos (dado por la mediana) se requieren menos de 66s.

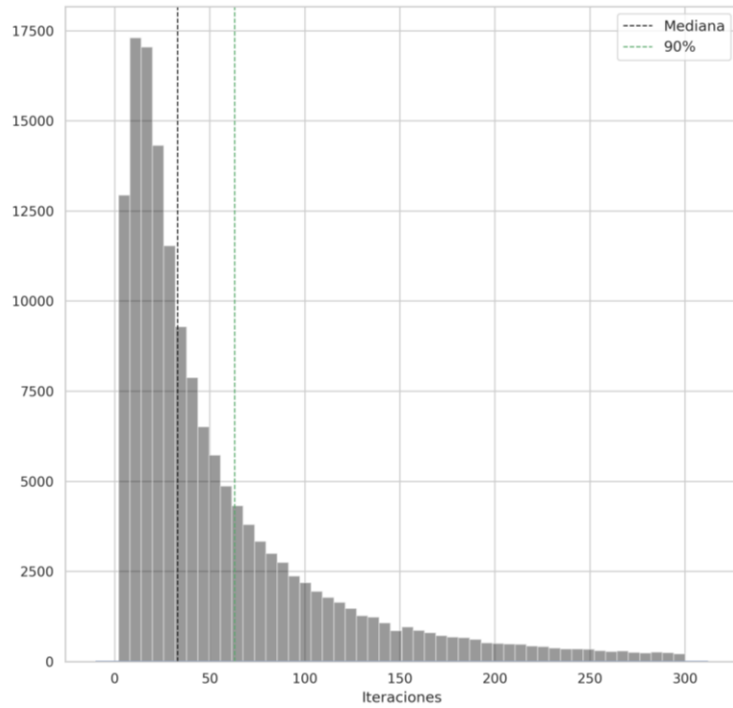


Figura 4: Histograma de iteraciones

En la figura 5 se muestra la distribución de iteraciones que requiere el algoritmo en una corrida. Debido al extenso número de casos que se evaluaron, se seleccionaron los tres casos que se consideraron representativos de los resultados. Los filtros elegidos tienen una frecuencia de polo de 643.5 Hz y factor de amortiguamiento de 1, pero diferentes ganancias. Como puede verse en la figura, en el caso en que la ganancia es 0.5 el optimizador requiere más iteraciones para lograr un diseño válido. Por otro lado, en el caso en que la ganancia es 2, en la mayoría de las corridas, logra encontrar filtros dentro de las especificaciones con unas pocas iteraciones.

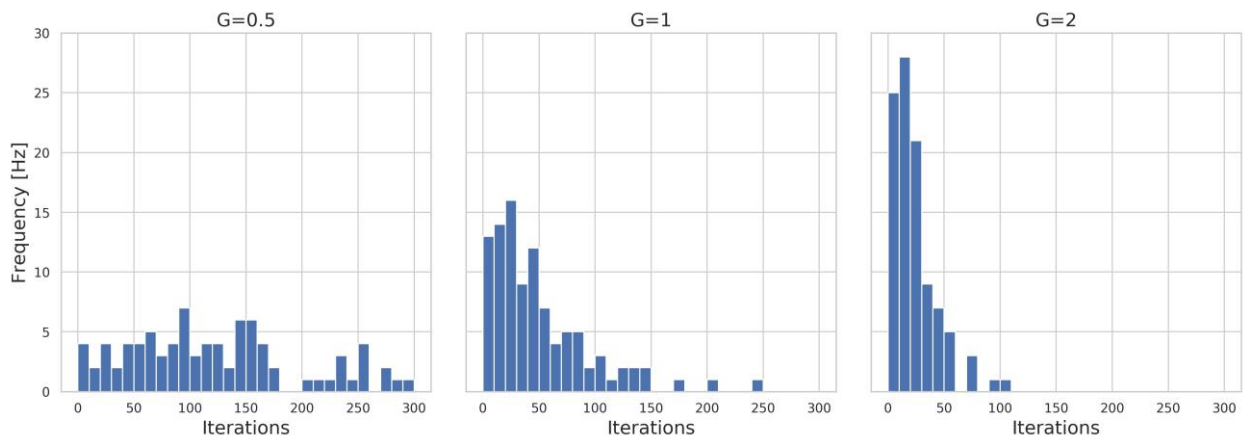


Figura 5: Histograma de iteraciones para una corrida con $d=1$ y $f_0=643.5$

En la tabla 2, se muestra la mediana de iteraciones que requiere el optimizador para encontrar un filtro válido en una corrida (100 ejecuciones con 300 iteraciones como máximo). Esta tabla se realizó considerando una frecuencia de polo de 643.52 Hz.

G \ d	d		
	0.25	1	2
0.5	78	110	174.5
1	23	37.5	47
2	17	19	31

Tabla 2: Mediana de iteraciones para la convergencia

Con el objeto de verificar estos resultados, se implementó uno de los filtros diseñados mediante PSO en un modelo de simulación SPICE de la estructura configurable analógica del PSoc1, donde se midió la relación tensión de entrada respecto a la de salida para distintas frecuencias. Las especificaciones del filtro implementado pueden verse en la tabla 3. En la figura 6, se muestra la respuesta en frecuencia teórica (línea azul) y, en puntos verdes, las mediciones SPICE. En los puntos simulados, el error entre respecto al cálculo teórico se encuentra en el orden del 0.04%, en promedio. Como puede observarse, el circuito diseñado usando las heurísticas se comporta excelentemente.

Frecuencia de polo	4989 [Hz]
Ganancia	1
Factor de amortiguamiento	1

Tabla 3: Especificaciones del filtro simulado

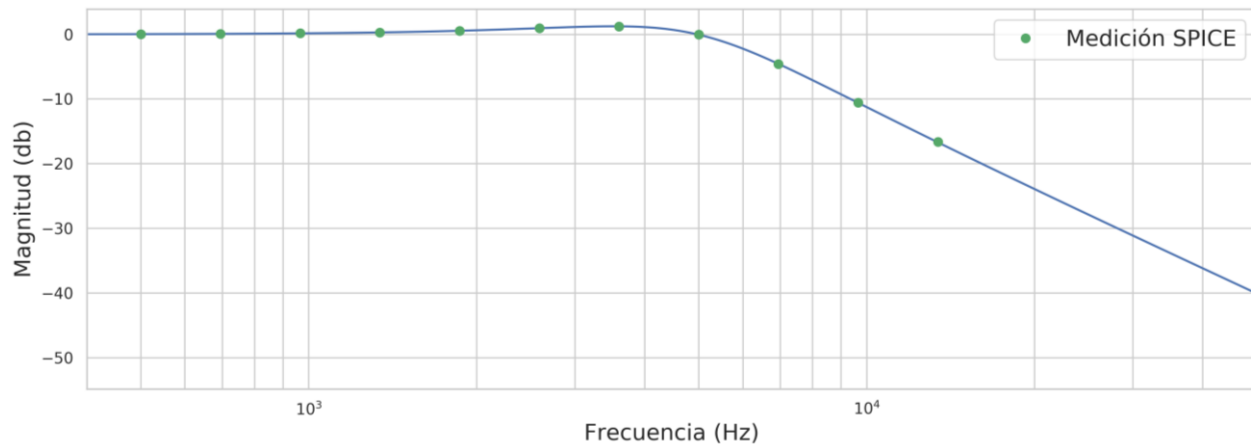


Figura 6: Respuesta en frecuencia teórica y simulada para un filtro implementado en PSoC1

Conclusión

El diseño de filtros analógicos de capacidades conmutadas mediante heurísticas permite obtener resultados cercanos al óptimo en un corto periodo de tiempo. En el presente trabajo se evaluó la capacidad de obtener diseños válidos del optimizador por enjambre de partículas para una amplia variedad de especificaciones de filtros. Se evaluaron 1800 filtros con diversas características, la tasa de éxito para todo el rango evaluado fue del 100%. Para la mitad de los casos analizados el optimizador logra converger en menos de 66s, corriendo en una PC estándar. Con el objeto de verificar los resultados obtenidos, se realizó la simulación una de las soluciones obtenidas utilizando un modelo SPICE del filtro, lográndose una alta correlación entre los simulado y el comportamiento teórico.

Referencias

- 1 Murphy, R., & Family, A. P. (2012). Getting Started with PSoC 1. Cypress Semiconductor.
- 2 Cypress Semiconductor. (2015). PSoC® 5LP: CY8C58LP. Family Datasheet. Prod. Cypress Semiconductor.
- 3 FPAA, D. R. (2003). AN221E04 Datasheet Dynamically Reconfigurable FPAA With Enhanced I/O.
- 4 Chen WK. (2009). Passive, Active, and Digital Filters, CRC Press. 2009.
- 5 Raut R, Swamy MNS. (2010). Modern Analog Filter Analysis and Design: A Practical Approach, Wiley-VCH.
- 6 Dimopoulos H. Analog Electronics Filters, Theory, Design and Synthesis, Springer, 2012.
- 7 Deliyannis T, Yichuang S, Kel Fidler J. (1999). Continuous-Time Active Filter Design. CRC Press.
- 8 Náhlík, J., Hospodka, J., Šubrt, O., & Chávez, A. B. (2015, September). Optimization methods for switched capacitor circuits. In 2015 International Conference on Applied Electronics (AE) (pp. 165-168). IEEE.
- 9 Vural, R. A., Yildirim, T., Kadioglu, T., & Basargan, A. (2011). Performance evaluation of evolutionary algorithms for optimal filter design. IEEE transactions on evolutionary computation, 16(1), 135-147.
- 10 Van Ess, D. (2007). AN2168. PSoC® 1 Understanding Switched Capacitor Filter
- 11 Eberhart, R., & Kennedy, J. (1995, October). A new optimizer using particle swarm theory. In MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science (pp. 39-43). Ieee.
- 12 Zambrano-Bigiarini, M., Clerc, M., & Rojas, R. (2013, June). Standard particle swarm optimisation 2011 at cec-2013: A baseline for future pso improvements. In 2013 IEEE Congress on Evolutionary Computation (pp. 2337-2344). IEEE.
- 13 PSC, "Particle Swarm Central," 2013, <http://www.particleswarm.info/>. [Online. Last accessed 14-Mar-2013].