

FILTRO ADAPTIVO BASADO EN DISPOSITIVOS PROGRAMABLES

Mónica A. Lovay

Grupo de Estudio en Calidad en Mecatrónica, Facultad Regional Villa María,
Universidad Tecnológica Nacional, mlovay@gmail.com

Resumen

En este trabajo se aborda el problema de proveer tolerancia a fallas a un filtro pasabajo de octavo orden, el cual tiene la capacidad de adaptarse a cambios que puedan motivar un apartamiento de las características deseadas del mismo. Para ello, se utilizan los principios de hardware evolutivo. El filtro bajo estudio se implementa a través de la conexión en cascada de cuatro filtros biquadráticos. Si se detecta que la característica de transferencia del filtro sobrepasa los límites aceptables para la aplicación, se propone reconfigurar el filtro, utilizando los valores de capacidades y ganancias disponibles en el chip. Para poder establecer si es necesaria la reconfiguración del filtro se utiliza el método de análisis transitorio. La evaluación de la performance del esquema presentado se efectúa a través de un modelo de fallas paramétricas. Los resultados de las simulaciones realizadas muestran que el filtro es capaz de mantener su funcionalidad bajo la presencia de fallas.

1. Introducción

Cuando un filtro se encuentra operando en una aplicación, pueden existir determinados agentes que contribuyan a disminuir su performance, por ejemplo, a través de la degradación de algunos de los componentes del filtro (capacidades, resistencias, amplificadores operacionales, interconexiones, etc.). Estas degradaciones pueden hacer que la característica de transferencia del mismo sobrepase los límites aceptables para la aplicación. En este contexto, si la aplicación es crítica, puede requerirse de características de funcionamiento seguro, adaptación a un entorno cambiante, o la capacidad para compensar las degradaciones en sus propios circuitos.

El hardware evolutivo (EHW), en particular, es una metodología que combina hardware reconfigurable con algoritmos evolucionarios con el propósito de adaptar un sistema a cambios en el ambiente o proveer recuperación de fallas. El diseñador establece los objetivos de performance y el algoritmo genético (AG) [1] busca las posibles configuraciones para alcanzarlos [2, 3].

En este trabajo, se aborda el problema de proveer tolerancia a fallas a un filtro pasabajo de octavo orden, el cual tiene la capacidad de adaptarse a cambios (en el medio ambiente de operación o en sus propios parámetros) que puedan motivar un apartamiento de las características deseadas del mismo. El filtro bajo estudio se implementa a través de la conexión en cascada de cuatro filtros biquadráticos en ispPAC10 de Lattice Semiconductor [4]. Para poder establecer si es necesaria la reconfiguración del filtro se requiere de un sistema de test. En nuestro caso, se utiliza el Método de Análisis Transitorio [5], el cual puede aplicarse a secciones de segundo orden. Si se detecta que el sistema no cumple con las especificaciones, el AG determina las posibles soluciones y asigna un nuevo valor para cada uno de los parámetros configurables del filtro.

2. Descripción del Sistema

ispPAC10 es un dispositivo analógico programable de Lattice Semiconductor. La Figura 1 muestra el diagrama de bloques del mismo. Como puede apreciarse, ispPAC10 presenta cuatro celdas analógicas, cada una de las cuales está compuesta por un amplificador operacional (OA), una resistencia y un capacitor en el lazo de realimentación, y dos amplificadores de entrada (IA) que se conectan al amplificador operacional a través de resistencias.

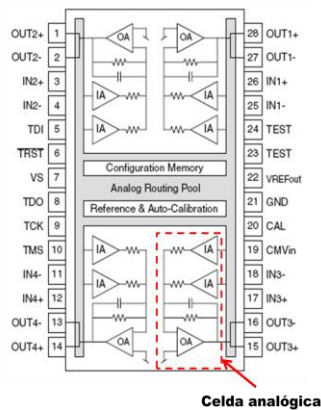


Figura 1. Diagrama de bloques de ispPAC10.

En la figura también puede observarse una zona sombreada, que representa los recursos de interconexión internos que permiten programar conexiones entre las celdas. Existen, además, un sistema de referencia y auto calibración, y una memoria.

La resistencia en el lazo de realimentación tiene un valor fijo y sólo puede conectarse o desconectarse. Sin embargo, el capacitor que tiene en paralelo, puede adoptar 128 valores posibles (desde $1.07E-12$ hasta $61.59E-12$), es decir, es programable. Por otro lado, los IA pueden ser programados desde ganancia -10 a +10 en pasos de 1. Las resistencias que conectan a los IA con el OA son fijas.

El diseño del sistema analógico propuesto se realizó utilizando Pac Designer [6], el cual permite conectar y desconectar los recursos entre sí, además de poder establecer el valor correspondiente para aquellos que son programables. Por medio de esta herramienta, se diseñó un filtro Butterworth pasabajo de orden ocho, con ganancia en continua igual a la unidad. La característica de este filtro se eligió a título de ejemplo y es el diseño de partida para el filtro que posteriormente se evolucionará en campo. El filtro se implementa mediante la conexión en cascada de cuatro secciones bicuadráticas. Este tipo de secciones puede sintetizarse con una gran cantidad de configuraciones basadas en OA; por lo que existen numerosos circuitos con los que puede obtenerse una función de transferencia bicuadrática. En cada chip ispPAC10 pueden implementarse hasta dos secciones bicuadráticas. Por este motivo, fue necesario utilizar dos chips.

La Figura 2 muestra las características del filtro diseñado, que corresponden a las llamadas respuestas nominales del filtro, las cuales deben mantenerse (dentro de ciertos parámetros de tolerancia establecidos), durante la operación en campo.

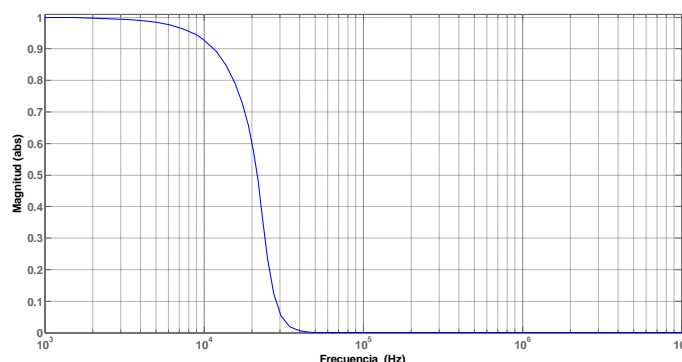


Figura 2. Respuesta en magnitud del filtro diseñado.

3. Test del filtro diseñado

Para efectuar el test del filtro diseñado, se utilizó el método de análisis transitorio. Este método de test puede aplicarse a secciones de segundo orden, lo cual no impide su uso en nuestro caso, ya

que como se mencionó anteriormente, el filtro diseñado está naturalmente particionado en secciones de segundo orden. Mediante el análisis de la respuesta transitoria, se pueden obtener los coeficientes de la ecuación del filtro bicuadrático diseñado. En la Figura 3 se muestran los parámetros de la respuesta transitoria.

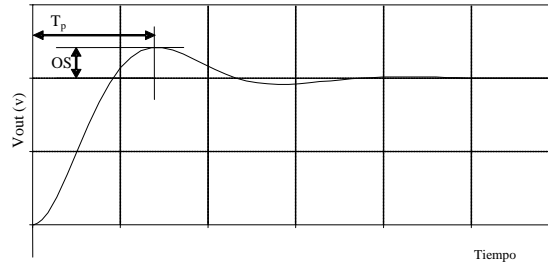


Figura 3. Parámetros de la respuesta transitoria.

Las fórmulas que permiten obtener los parámetros de interés para nuestro caso, son:

$$T_p = \frac{\pi}{\omega_n \cdot \sqrt{1-\zeta^2}} \quad (1)$$

$$OS = e^{-\pi \left(\frac{\zeta}{\sqrt{1-\zeta^2}} \right)} \quad (2)$$

En (1) y (2), T_p es el tiempo de ocurrencia del pico, OS es la sobre elongación o sobre pico, ω_n la frecuencia natural no amortiguada y ζ el amortiguamiento. Usando estas ecuaciones y obteniendo mediante medición los valores de T_p y OS, pueden obtenerse los valores de ζ y ω_n . Por otro lado, midiendo la tensión de salida cuando la señal ya es estable, puede obtenerse el valor de ganancia. Es decir, puede reconstruirse la ecuación de transferencia de cada etapa, y teniendo éstas, entonces, se puede obtener la función total. Esta función debe ser comparada con la función nominal (que surgió en los cálculos de diseño), con el propósito de poder calcular el error entre ambas, para determinar si es necesaria la invocación del proceso de reconfiguración.

4. Funcionalidad del AG

El AG es una técnica de búsqueda y optimización basada en los principios de la genética y la selección natural. Este algoritmo permite que una población formada por una cierta cantidad de individuos evolucione de acuerdo a las reglas especificadas, con el propósito de maximizar una función objetivo (fitness). Los fundamentos y detalles del AG pueden encontrarse en [1, 7-9].

En nuestro caso, el AG debe encontrar los valores de ganancia (k_i) de los IA y los valores (c_i) de los capacitores correspondientes a cada filtro bicuadrático, con el objetivo de mantener la respuesta del filtro dentro de las especificaciones.

Analizando la curva que representa la respuesta en Magnitud del filtro nominal (Fig. 2. b), se seleccionaron cinco puntos en los que el AG debe evaluar la función de transferencia de cada individuo (posibles valores de k_i y c_i que determinan un diseño de filtro) y compararla con la respuesta del filtro nominal para luego efectuar la asignación del valor de aptitud. La figura 4 muestra la distribución de los mismos.

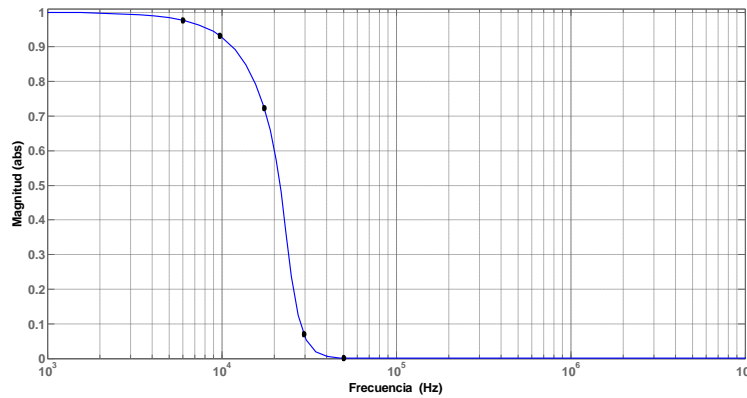


Figura 4. Ubicación de los puntos con los que el AG realiza la evaluación.

Como puede apreciarse en la figura, se consideraron cinco puntos: dos de ellos corresponden a la banda de paso, uno para el punto de corte y dos puntos para la banda de atenuación. En cada uno de ellos y para cada individuo, el algoritmo determina:

1. **Error relativo en magnitud:** representa el error en magnitud con el que el individuo se acerca al filtro nominal en un punto P determinado.

$$\text{ErrorRelativoMag}(P) = \text{abs} [(\text{MagnitIndividuo}(P) - \text{MagnitNominal}(P)) / \text{MagnitNominal}(P)] \quad (3)$$

2. **Error relativo en frecuencia:** representa el error en frecuencia con el que el individuo se acerca al filtro nominal para un punto P determinado.

$$\text{ErrorRelativoFrec}(P) = \text{abs} [(\text{FrecIndividuo}(P) - \text{FrecNominal}(P)) / \text{FrecNominal}(P)] \quad (4)$$

El AG debe encontrar un individuo, cuyos errores en magnitud y en frecuencia, satisfagan determinados criterios de performance en cada uno de los puntos considerados. La función fitness que permite asignarle un valor de aptitud a cada individuo se elaboró considerando:

- Los errores relativos máximos en magnitud y en frecuencia obtenidos de la evaluación de los puntos correspondientes a la banda de paso (EMMBP y EMFBP).
- El error relativo en magnitud y en frecuencia obtenido en el punto de corte (EMPC y EFPC).
- Los errores relativos máximos en magnitud y en frecuencia obtenidos de la evaluación de los puntos correspondientes a la banda de atenuación (EMMBA y EMFBA).

La función fitness (fit) utilizada para cada individuo (y), se construyó utilizando el Método de los Pesos [9,10], el cual permite su formulación a través de una suma ponderada:

$$\text{fit}(y) = B - [w_{BP} \cdot (\text{EMMBP}(y) + \text{EMFBP}(y)) + w_{PC} \cdot (\text{EMPC}(y) + \text{EFPC}(y)) + w_{BA} \cdot (\text{EMMBA}(y) + \text{EMFBA}(y))] \quad (5)$$

donde, B es una constante agregada para evitar números negativos en la función. Los valores de w_i se denominan pesos, y representan el grado de importancia concedido al error en cada banda. En nuestro caso: $w_{BP}=0.46$; $w_{PC}=0.46$ y $w_{BA}=0.08$.

Además, el AG fue implementado utilizando codificación real, por lo que debieron utilizarse estrategias de cruce y de mutación apropiadas para este tipo de codificación. Se utilizó el método Simplex Crossover (SPX) y el Método de Mutación Aleatorio Uniforme respectivamente [6].

5. Resultados experimentales

5.1. Operación libre de fallas

El AG fue implementado utilizando como parámetros, una población inicial de 200 individuos, probabilidad de cruce de 1 y probabilidad de mutación de 0.05. El algoritmo se detiene cuando alcanza 500 generaciones, o bien, cuando encuentra una solución que cumple con el siguiente criterio de performance: error en Banda de Paso $\leq 20\%$, error en Punto de Corte $\leq 10\%$ y error en Banda de Atenuación $\leq 40\%$.

Se efectuaron 50 corridas del algoritmo con los parámetros propuestos. Como el AG realiza un proceso estocástico, los resultados pueden cambiar de acuerdo a la distribución estadística de la población inicial. Para poder observar como los resultados pueden ser afectados por el seteo de la población inicial, hemos cambiado la semilla en la generación aleatoria de cada corrida, y consideramos que todos los valores que pueden adoptar los capacitares y IA están disponibles. La Tabla 1 muestra la mediana y el máximo correspondientes a la cantidad de generaciones bajo la cuales el algoritmo converge y el tiempo de ejecución.

Tabla 1. Cantidad de generaciones y tiempos de ejecución.

Mediana de cant. Generaciones	Máxima cant. Generaciones	Mediana de Tiempo (s)	Máximo Tiempo (s)
33,5	73	299,77	624,27

Para poder completar la evaluación de la performance del algoritmo propuesto, es importante analizar también, el error relativo del mismo en cada una de las bandas. La Tabla 2 muestra la caracterización del error relativo en frecuencia para la Banda de Paso, el Punto de Corte y la Banda de Atenuación. Como puede apreciarse, en todos los casos, el AG converge encontrando soluciones que cumplen con los criterios de performance establecidos.

Tabla 2. Caracterización del error relativo en frecuencia.

Error Banda de Paso			Error Punto de Corte			Error Banda de Atenuación		
Mediana	Mínimo	Máximo	Mediana	Mínimo	Máximo	Mediana	Mínimo	Máximo
10,71%	1,04%	19,33%	4,18%	0,12%	7,09%	23,82%	6,62%	37,94%

La Figura 5 muestra la respuesta en magnitud correspondiente al Filtro Nominal (trazo azul) y al mejor y peor diseño (trazos verde y rojo respectivamente) obtenidos a partir del AG.

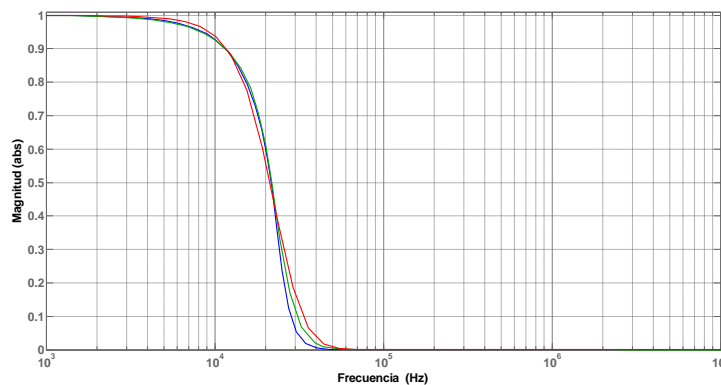


Figura 5. Respuesta en magnitud en condiciones normales.

5.2. Operación con Fallas

La performance del esquema tolerante a fallas presentado, se evaluó a través de la inyección de fallas paramétricas. Este tipo de falla considera desviaciones en los valores nominales de los capacitores de dos de los filtros bicuadráticos (de -50% a +50% en pasos de 10%). La Tabla 3 resume los resultados obtenidos en cuanto a cantidad de generaciones de convergencia y tiempos de ejecución. Comparando estos resultados con los obtenidos bajo condiciones normales (Tabla 1), puede observarse que la mediana y el máximo correspondientes a la cantidad de generaciones aumentan, lo cual provoca un incremento en los tiempos de ejecución.

Tabla 3. Cantidad de generaciones y tiempos de ejecución para operación con falla.

Mediana de cant. Generaciones	Máxima cant. Generaciones	Mediana de Tiempo (s)	Máximo Tiempo (s)
48,5	202	427,29	1779,62

Para completar la evaluación del sistema propuesto, en la Tabla 4 se visualizan los errores relativos en frecuencia correspondientes a la Banda de Paso, Punto de Corte y Banda de Atenuación. Con respecto a la operación normal (Tabla 2), puede apreciarse un incremento en el error mínimo cuando hay fallas, lo que provoca una disminución en el rango de error para la Banda de Paso y la Banda de Atenuación, debido a que en estos casos el error máximo disminuye levemente con respecto a la operación normal. Con respecto a la mediana del error relativo, observamos que aumenta levemente en la Banda de Paso, mientras que en el Punto de Corte y en la Banda de Atenuación disminuye.

Tabla 4. Caracterización del error relativo en frecuencia para operación normal y con falla.

Error Banda de Paso			Error Punto de Corte			Error Banda de Atenuación		
Mediana	Mínimo	Máximo	Mediana	Mínimo	Máximo	Mediana	Mínimo	Máximo
12,24%	6,16%	18,98%	3,27%	0,44%	9,25%	23,52%	13,33%	37,23%

No obstante, como puede apreciarse en la tabla, todas las soluciones encontradas por el AG bajo condiciones de falla, siguen respetando los parámetros de performance establecidos.

6. Conclusiones

En este trabajo se ha presentado un filtro de alto orden basado en dispositivos programables, con características de tolerancia a fallas a través de hardware evolutivo. Para determinar la necesidad de reconfiguración del filtro se utilizó el método de análisis transitorio. El AG que busca los valores de reconfiguración es robusto para las fallas consideradas en nuestra evaluación. Los resultados de las simulaciones realizadas muestran que el filtro es capaz de mantener su funcionalidad bajo la presencia de fallas.

REFERENCIAS

- [1] D. Goldberg, *“Genetic Algorithm. Search, optimization and machine learning”*, Addison-Wesley, 1989.
- [2] R. Salem Zebulum, M. Pacheco, and M. Vellasco, *“Evolutionary electronics: automatic design of electronic circuits and systems by genetic algorithms”*, CRC Press, 2002.
- [3] Q. Ji, Y. Wang, M. Xie, and J. Cui, *“Research on fault-tolerance of analog circuits based on evolvable hardware”*, in Proceedings 7th international conference on Evolvable systems: from biology to hardware, pp. 100-108, 2007.
- [4] *Designing Higher-order Filters*, Lattice Semiconductor Corporation, 2002.
- [5] J. Calvano, V. Alves, and M. Lubaszewski, *“Fault detection methodology for second order filters using compact test vectors transient analysis”*, 3rd. International Workshop on Design of Mixed-Mode Integrated Circuits and Applications, 18-24, 1999.
- [6] *Pac-Designer Software User Manual*, Lattice Semiconductor Corporation, 2011.
- [7] C. Reeves, and J. Rowe, *“Genetic algorithms: principles and perspective. A guide to GA theory”*, Kluwer Academic Publishers, 2002.
- [8] R. Salem Zebulum, M. Pacheco, and M. Vellasco, *“Evolutionary electronics: automatic design of electronic circuits and systems by genetic algorithms”*, CRC Press, 2002.
- [9] El-Ghazali Talbi, *“Metaheuristics From Design to Implementation”*, Wiley, 2009.
- [10] X. Yu, M. Gen, *“Introduction to Evolutionary Algorithms”*, Springer 2010.