

DISEÑO DE UNA TRANSICIÓN ENTRE GUÍAS COAXIAL Y RECTANGULAR EMPLEANDO ELEMENTOS FINITOS, SEGMENTACIÓN Y REDES NEURONALES ARTIFICIALES

Juan M. Cid Álvarez y Juan Zapata

Departamento de Electromagnetismo y Teoría de circuitos. E.T.S.I. de Telecomunicación

Ciudad Universitaria S/N, 28080 Madrid

jmcid@etc.upm.es, jz@etc.upm.es

Abstract— A transition between coaxial and rectangular is designed maximizing the adaptation. The free parameters of the system are adjusted using global optimization algorithms. The circuit response analysis is carried out using analytical models, based on artificial neural networks, of the sections forming the transition. But before it is needed to obtain these models. Firstly a small set of training points is generated by finite elements, and later the artificial neural networks are trained for learning their behavior. The design time decrease greatly with the use of artificial neural networks.

I. INTRODUCCIÓN

Se pretende diseñar una transición entre guías coaxial y rectangular (Fig. 1) en el rango de frecuencias que va de 10,7 a 12,8 GHz. Los parámetros libres del sistema son: $hp1$, $hp2$, hx , hg , rc , lpp , lcp y lcc . Se pide en el diseño tener en cuenta la tolerancia en los valores de: hx , hg , rc , lpp , lcp y lcc . Se busca que el resultado sea óptimo y para ello se usan algoritmos de optimización globales. En consecuencia, se hace necesario analizar la respuesta del circuito un número muy elevado de veces. Esto hace el uso de simulaciones electromagnéticas (EM) de onda completa prohibitivo computacionalmente. Por lo tanto, se emplean modelos basados en redes neuronales artificiales (RNA's) que mantienen la precisión de las simulaciones EM al tiempo que exhiben una gran eficiencia en cuanto a tiempos de computación requeridos [1]. En esta metodología, lo primero es entrenar las RNA's para que aproximen los parámetros S de todos los componentes a ser modelados, en el rango de valores de los parámetros libres para el cual se espera dar uso al modelo, empleando datos obtenidos de simulaciones EM. En nuestro caso, el método de simulación es el SFE [2] (segmentación y elementos finitos) que como se indica en [3] combina muy bien con las RNA's como consecuencia de la simplificación que se produce en el comportamiento a modelar al aplicar segmentación.

II. MODELADO

En primer lugar se crea un modelo de RNA's para cada una de las 3 secciones que se indican en la figura 1. Puesto que dos de estas secciones (correspondientes a los tornillos) son formalmente iguales, sólo se hace necesario obtener dos modelos de RNA's: para un tornillo en guía rectangular y para un conector coaxial también en rectangular.

Para modelar los parámetros S de una de esta secciones se emplean perceptrones multicapa (PMC's) con dos capas de pesos [1]. Cada uno de los parámetros S es modelado por uno de estos PMC's (con dos salidas reales) de manera independiente. Se hace uso de la simetría de la matriz de dispersión generalizada (MDG) para reducir el número de PMC's a usar. Los puntos de entrenamiento, obtenidos con SFE, se distribuyen de manera aleatoria en el rango de valores de los parámetros de entrada a los PMC's que nos interesa. El entrenamiento, o ajuste de los pesos de cada PMC, se realiza empleando un algoritmo cuasi-Newton. La función de coste a minimizar es la suma de las diferencias al cuadrado entre lo predicho por el PMC y el

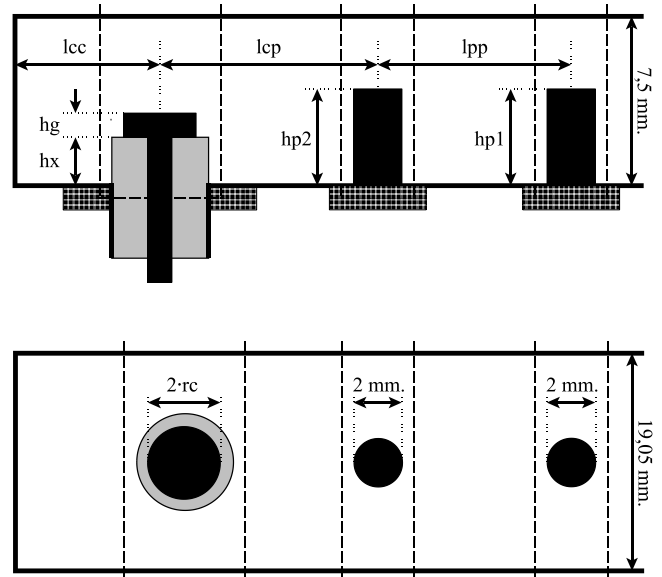


Fig. 1. Transición entre guías coaxial y rectangular. Se emplea un conector coaxial SMA. El sistema se considera sin pérdidas y en el vacío.

valor esperado en los puntos de entrenamiento. Calculándose las derivadas de esta función de coste (necesitadas por el algoritmo cuasi-Newton) mediante el método de retropropagación [1].

En la tabla I se indican, para el tornillo en guía rectangular, el número de modos considerado en cada puerta, el número de PMC's empleados, el número de neuronas intermedias por PMC, las entradas a cada PMC y su rango de variación, el número de puntos de entrenamiento empleados, el tiempo empleado en la generación de estos puntos y el tiempo empleado para entrenar los PMC's. Y en la tabla II se ofrecen los valores equivalentes para el conector coaxial. Los tiempo de CPU siempre se refieren a un Pentium II a 400 MHz y 412 MB RAM.

Después, la respuesta del sistema completo ya se puede obtener enlazando las MDG's de cada una de las secciones (cuyo modelo se ha obtenido anteriormente) mediante las correspondientes guíasondas de solución analítica.

III. DISEÑO

Para el diseño de la transición se emplea el algoritmo "simulated annealing" (SA) de [4]. En cada iteración del algoritmo se analiza la respuesta del sistema (con el modelo de RNA's) para 10 puntos de frecuencia distribuidos uniformemente en el rango de interés (de 10,7 a 12,8 GHz.) y para 13 combinaciones distintas de los parámetros libres: hx , hg , rc , lpp , lcp y lcc ; buscando la estabilidad de la solución frente a las tolerancias en

TABLA I
TORNILLO EN GUÍA RECTANGULAR

Nº de modos	
Puertas rectangulares	50 por puerta
Nº de PMC's	5050
Neuronas intermedias por PMC	9
Entrada 1	hp1 (hp2)
Rango de valores	0,05-0,70 cm.
Entrada 2	Frecuencia
Rango de valores	10,6-12,9 GHz.
Puntos de entrenamiento	
Nº de puntos	300
Tiempo de CPU	26.057,999 segundos
Entrenamiento de los PMC's	
Tiempo de CPU	6.531,011 segundos

TABLA II
CONECTOR COAXIAL EN GUÍA RECTANGULAR

Nº de modos	
Puertas rectangulares	50 por puerta
Puerta coaxial	11
Nº de PMC's	6216
Neuronas intermedias por PMC	14
Entrada 1	hx
Rango de valores	0,01-0,50 cm.
Entrada 2	hg
Rango de valores	0,01-0,24 cm.
Entrada 3	hx
Rango de valores	0,07-0,20 cm.
Entrada 4	Frecuencia
Rango de valores	10,6-12,9 GHz.
Puntos de entrenamiento	
Nº de puntos	700
Tiempo de CPU	184.467,020 segundos
Entrenamiento de los PMC's	
Tiempo de CPU	90.650,218 segundos

estos parámetros. En la función de coste se pide que las mayores pérdidas de retorno calculadas sean mínimas. El punto de partida se elige de manera aleatoria, estando los parámetros: hp1, hp2, hx, hg y rc limitados a variar dentro de los rangos dados en las tablas I y II. El tiempo de cálculo empleado por el SA fue de 20.063,29 segundos, con un total de 1.240 iteraciones realizadas.

Indicar que, a la hora de obtener la respuesta del sistema completo empleando RNA's, en las puertas de comunicación con el exterior sólo se considera el modo fundamental de cada una. Esto se puede hacer sabiendo que los parámetros S de las MDG's así obtenidas con las RNA's, mantienen la precisión del SFE con el mismo número de modos en las puertas que en los puntos de entrenamiento, como consecuencia de haber modelado cada parámetro S de manera independiente. De esta manera se evita calcular un número importante de parámetros S y se reduce el tamaño de las MDG a conectar, reduciendo el tiempo de cálculo. En las restantes puertas del circuito se consideran 50 modos porque las secciones pueden llegar a estar arbitrariamente próximas.

IV. RESULTADOS Y CONCLUSIONES

Al final del proceso de optimización se obtiene el resultado presentado en la figura 2 donde se muestran las pérdidas de retorno de la transición (en dB's) en el rango de frecuencias de interés. Con cruces están los puntos predichos por el modelo de RNA's y con círculos la comprobación con el SFE de la realización del diseño. Si bien se observa que en algunos puntos la diferencia entre los dos métodos es de varios dB's esto era de esperar como consecuencia de los valores tan reducidos que se están considerando. Aún así, se observa como el valor de las mayores pérdidas de retorno apenas sufre un ligero empeoramiento al analizar con SFE. Las pérdidas de retorno siguen siendo suficientemente bajas como para considerar bueno el diseño alcanzado.

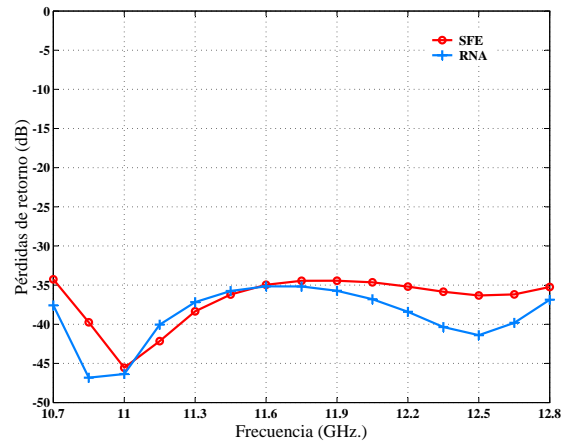


Fig. 2. Pérdidas de retorno en el punto de diseño. Calculadas con SFE y con el modelo de RNA's

Por otro lado, si en vez de haber empleado las RNA's en el SA se hubiera empleado el SFE (considerando fijo el número de iteraciones que el SA debería realizar) se estima el tiempo de cálculo necesario para la convergencia del algoritmo en más de 9 meses. En consecuencia, vemos como el uso de RNA's hace mucho más realizable la aplicación de los algoritmos de optimización global al diseño de circuitos de microondas.

REFERENCIAS

- [1] Q. J. Zhang and K. C. Gupta, *Neural networks for RF and microwave design*, Artech House, 2000.
- [2] J. Rubio, J. Arroyo and J. Zapata, "Analysis of passive microwave circuits by using a hybrid 2-D and 3-D finite element-mode matching method", *IEEE Trans. Microwave Theory Tech.*, vol. 47, No 9, pp. 1746-1749, Sep. 1999.
- [3] J. M. Cid and J. Zapata, "CAD of rectangular waveguide H-plane circuits by segmentation, finite elements and artificial neural networks", *Electronics Letters*, vol. 37, No 2, pp. 98-99, Jan. 2001.
- [4] W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery, *Numerical recipes in FORTRAN: the art of scientific computing*(second ed.), Cambridge University Press., 1992.