

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN
UNIVERSIDAD POLITÉCNICA DE CARTAGENA



Proyecto Fin de Carrera

Desarrollo de técnicas de diseño de filtros en tecnología guiada para aplicaciones espaciales



AUTOR: Juan Antonio Rosell Franco
DIRECTOR: Alejandro Álvarez Melcón
CODIRECTORES: Fernando D. Quesada Pereira
Juan Pascual García
Cartagena, Diciembre 2007



Autor	Juan Antonio Rosell Franco
E-mail del Autor	jarosell@gmail.com
Director(es)	Alejandro Álvarez Melcón
E-mail del Director	alejandro.alvarez@upct.es
Codirector(es)	Fernando Daniel Quesada Pereira Juan Pascual García
Título del PFC	Desarrollo de técnicas de diseño de filtros en tecnología guiada para aplicaciones espaciales
Descriptor(es)	Diseño de filtros, Guía de onda
<p>Resumen</p> <p>El presente proyecto se engloba dentro del diseño de un radiotelescopio en la banda de 1420MHz, conocida como banda de emisión del hidrógeno. La parte desarrollada por este proyecto será la de diseñar un filtro en tecnología guía de onda para dicha banda. Para ello, hemos realizado una herramienta de diseño que es capaz de diseñar filtros de una forma rápida y eficiente en cualquier banda.</p> <p>Esta herramienta va a utilizar el programa FEST de la Agencia Espacial Europea, que permite el análisis de estructuras en guía de onda pero no su diseño.</p> <p>La principal novedad de este trabajo es la incorporación de redes neuronales para calcular las dimensiones necesarias del filtro de una forma muy precisa.</p> <p>Junto al método de desarrollo utilizado presentaremos ejemplos de filtros con diversas especificaciones para su utilización en la banda del hidrógeno u en otras bandas utilizadas para comunicaciones por satélite.</p>	
Titulación	Ingeniero de Telecomunicación
Intensificación	Sistemas y Redes de Telecomunicación
Departamento	Tecnología de la Información y la Comunicación
Fecha de Presentación	Diciembre de 2007

Agradecimientos

Quiero comenzar este trabajo agradeciendo a mi director de proyecto Alejandro Álvarez Melcón la ayuda que me ha prestado durante todo este tiempo. Ha sido tremendamente fácil trabajar con él; incluso cuando las cosas no funcionaban nunca perdía la esperanza y siempre animaba a probar cosas nuevas y continuar.

También quiero agradecer a Juan Pascual García la ayuda con las redes neuronales y los algoritmos de optimización. Sin él esto no hubiera sido posible.

No quiero olvidarme tampoco de agradecer a José Luis Gómez Tornero la ayuda brindada durante estos años.

Por último quiero agradecer a mi familia y amigos por estar siempre ahí y animarme cuando lo he necesitado.

Gracias a todos.

Índice de contenidos

1.	INTRODUCCIÓN	1
1.1	Aplicaciones espaciales. Radioastronomía	1
1.2	Filtros de microondas. Parámetros S	1
1.3	Guía de onda	4
1.4	Objetivo del proyecto.....	4
1.5	Organización del proyecto	6
2.	FEST (Fullwave Electromagnetic Simulation Tool)	7
3.	GENERACIÓN DE MEDIDAS.....	12
3.1.	Estructura de cada tipo de análisis.....	12
3.1.1.	Análisis de una cavidad: caso single ended.....	12
3.1.2.	Análisis de una cavidad: caso Double ended.....	13
3.1.3.	Análisis de dos cavidades	14
3.2	Análisis	16
3.2.1.	Single ended	16
3.2.2.	Double ended.....	20
3.2.3	Medida de acoplos	22
3.3	Ejemplos	25
4.	RED NEURONAL	31
4.1	¿Qué es una red neuronal?.....	31
4.2	Tipos de redes neuronales.....	32
4.3	Red neuronal utilizada	35
4.4	Entrenamiento de la red	35
4.5	Ejemplos de entrenamiento	37
5.	PROCESO DE DISEÑO	42

5.1 Interfaz para la generación de datos	42
5.2 Obtención del prototipo paso bajo del filtro	45
5.3 Obtención de los parámetros a sintetizar	47
5.4 Interfaz red neuronal.....	47
5.5 Obtención de las dimensiones finales.....	49
5.5.1 Ejemplo de filtro orden 3:.....	51
5.5.2 Ejemplo de filtro orden 4:.....	52
5.6 Obtención de la longitud del resonador sin cargar	53
5.7 Optimización automática.....	54
6. EJEMPLOS DE DISEÑO	57
6.1 Filtro pasa banda de orden 3 y 30 MHz de ancho de banda.....	57
6.2 Filtro pasa banda de orden 4 y 80 MHz de ancho de banda.....	60
6.3 Filtro pasa banda de orden 4 y 100 MHz de ancho de banda.....	63
6.4 Filtro pasa banda de orden 5 y 30 MHz de ancho de banda.....	66
6.5 Filtro pasa banda de orden 5 y 75 MHz de ancho de banda.....	69
7. CONCLUSIONES Y LÍNEAS FUTURAS	72
7.1 Conclusiones.....	72
7.2. Líneas Futuras.....	72
ANEXO I	74
ANEXO II.....	75
ANEXO III	77
BIBLIOGRAFÍA	79

Índice de figuras

Ilustración 1.1 Respuesta de un filtro pasa banda	3
Ilustración 1.2 Diagrama de bloques de la herramienta	5
Ilustración 2.1 Interfaz de la herramienta FEST.....	7
Ilustración 2.2 Propiedades del elemento Step	8
Ilustración 2.3 Ejemplo de conexionado	8
Ilustración 2.4 Opciones de configuración de FEST	9
Ilustración 2.5 Ejemplo de formato de un archivo .out	10
Ilustración 2.6 Representación gráfica del resultado de la simulación.....	10
Ilustración 3.1 Cavidad terminada en cortocircuito y dimensiones correspondientes ...	12
Ilustración 3.2 Representación de la estructura single ended en FEST.....	13
Ilustración 3.3 Cavidad y dimensiones correspondientes.....	13
Ilustración 3.4 Representación de la estructura double ended en FEST	14
Ilustración 3.5 Filtro de orden 2 y dimensiones correspondientes	15
Ilustración 3.6 Representación de la estructura de dos cavidades en FEST	16
Ilustración 3.7 Ajuste realizado sobre la fase	18
Ilustración 3.8 Fase y retardo de grupo de una cavidad analizada mediante el método single ended	18
Ilustración 3.9 Respuesta del S_{11} y retardo de grupo correspondiente	19
Ilustración 3.10 Respuesta de la fase del S_{11}	20
Ilustración 3.11 Parámetros S de una cavidad analizada mediante el método double ended.....	21
Ilustración 3.12 Evolución del parámetro S_{21}	22
Ilustración 3.13 Parámetros S de la estructura de dos cavidades	24
Ilustración 3.14 Evolución de la frecuencia de resonancia con las dimensiones	25
Ilustración 3.15 Evolución del factor de calidad con las dimensiones	26
Ilustración 3.16 Evolución de la frecuencia de resonancia con las dimensiones	27
Ilustración 3.17 Evolución del factor de calidad con las dimensiones	28
Ilustración 3.18 Evolución de la frecuencia de resonancia con las dimensiones	29
Ilustración 3.19 Evolución del acoplo entre-resonadores con las dimensiones	30
Ilustración 4.1 Estructura de una neurona	31
Ilustración 4.2 Estructura de red neuronal artificial	31
Ilustración 4.3 Función sigmoide	32
Ilustración 4.4 Función de base radial	33
Ilustración 4.5 Estructura de la red neuronal	34
Ilustración 4.6 Estructura de la red neuronal	35
Ilustración 4.7 Estructura de nuestra RBFN	35
Ilustración 4.8 Evolución del error con el entrenamiento	36
Ilustración 4.9 Evolución de la frecuencia de resonancia con las dimensiones	39
Ilustración 4.10 Evolución del factor de calidad con las dimensiones.....	39
Ilustración 4.11 Evolución de la frecuencia de resonancia con las dimensiones	40
Ilustración 4.12 Evolución del coeficiente de acoplo con las dimensiones.....	41
Ilustración 5.1 Interfaz utilizada para la generación de datos	42
Ilustración 5.2 Estructura de generación de archivos	43
Ilustración 5.3 Prototipo paso bajo del filtro	46
Ilustración 5.4 Respuesta filtro pasa-banda correspondiente	46
Ilustración 5.5 Interfaz utilizada para la red neuronal	48

Ilustración 5.6 Cavity terminada simplemente	49
Ilustración 5.7 Cavity terminada doblemente	49
Ilustración 5.8 Filtro de orden 2	50
Ilustración 5.9 Filtro de orden 3 y dimensiones correspondientes	51
Ilustración 5.10 Filtro de orden 4 y dimensiones correspondientes	52
Ilustración 5.11 Geometría de una cavity rectangular	54
Ilustración 6.1 Respuesta del filtro obtenido	58
Ilustración 6.2 Respuesta optimizada del filtro	59
Ilustración 6.3 Respuesta optimizada del filtro en detalle.....	59
Ilustración 6.4 Respuesta del filtro obtenido	61
Ilustración 6.5 Respuesta optimizada del filtro	62
Ilustración 6.6 Respuesta obtenida del filtro	64
Ilustración 6.7 Respuesta optimizada del filtro	65
Ilustración 6.8 Respuesta obtenida del filtro	67
Ilustración 6.9 Respuesta del filtro optimizado en detalle.....	68
Ilustración 6.10 Respuesta del filtro obtenido	70
Ilustración 6.11 Respuesta del filtro optimizado	71

Índice de tablas

Tabla 4.1 Resultados del entrenamiento para distintas configuraciones	38
Tabla 4.2 Resultados del entrenamiento para distintas configuraciones	40
Tabla 5.1 Comparación de resultados para distintos gradientes.....	55
Tabla 5.2 Comparación de resultados para distintas configuraciones del algoritmo genético.....	55
Tabla 6.1 Dimensiones iniciales, optimizadas y errores para el ejemplo 6.1	58
Tabla 6.2 Dimensiones iniciales, optimizadas y errores para el ejemplo 6.2.....	61
Tabla 6.3 Dimensiones iniciales, optimizadas y errores para el ejemplo 6.3.....	64
Tabla 6.4 Dimensiones iniciales, optimizadas y errores para el ejemplo 6.4.....	67
Tabla 6.5 Dimensiones iniciales, optimizadas y errores para el ejemplo 6.5.....	70

1. INTRODUCCIÓN

1.1 Aplicaciones espaciales. Radioastronomía

Cada día son más las aplicaciones de las telecomunicaciones relacionadas con el entorno espacial. El desarrollo de las comunicaciones por satélite ha hecho que nuestra sociedad disponga de servicios como GPS (Sistema de Posicionamiento Global), televisión y telefonía vía satélite... impensables hace unas décadas e imprescindibles en la actualidad.

Además estos avances han permitido un mayor estudio y conocimiento de nuestro entorno. Ciencias como la meteorología, geología,... utilizan a diario satélites para estudiar nuestro planeta. Por ejemplo, avances como los Radares de Apertura Sintética (S.A.R.), han supuesto una revolución en campos como la teledetección y la cartografía.

Otra de las ciencias en continuo desarrollo gracias a las telecomunicaciones es la radioastronomía, una rama perteneciente a la astronomía que al igual que ésta estudia los objetos celestes y fenómenos astrofísicos. La diferencia entre ambas es que la astronomía realiza las observaciones en el rango visible del espectro y la radioastronomía lo hace en la región radio. Mediante este tipo de observaciones ha sido posible el descubrimiento de nuevas galaxias, estrellas...

Otra aplicación de la radioastronomía es la búsqueda de Inteligencia Extraterrestre por medio de la recepción y análisis de señales electromagnéticas. Un ejemplo de este tipo de aplicación es el programa SETI ("Search for ExtraTerrestrial Intellingence") de la NASA.

Este creciente desarrollo de aplicaciones hace que cada vez sean más los servicios a incluir en el mismo espectro radioeléctrico y por tanto se deban utilizar sistemas de mejores características.

1.2 Filtros de microondas. Parámetros S

En todas las aplicaciones de telecomunicaciones, y especialmente en las espaciales, necesitaremos eliminar la información en otras bandas, que se traduce en ruido para nuestro sistema, añadiendo las mínimas pérdidas posibles. Para ello utilizaremos filtros de microondas en tecnología guía de onda, microstrip...dependiendo de la aplicación.

Los filtros son cuadripolos que permiten la transmisión de determinadas frecuencias, atenuando fuertemente las restantes. Dependiendo de sus componentes constitutivos, la naturaleza de las señales que tratan, respuesta en frecuencia y método de diseño, los filtros se clasifican en distintos grupos:

- **Según respuesta frecuencia**
 - Filtro paso bajo: Permite el paso de frecuencias bajas y atenúa fuertemente las altas frecuencias. Presentan ceros a alta frecuencia y polos a bajas frecuencia.
 - Filtro paso alto: Es el que permite el paso de frecuencias altas y atenúa fuertemente las bajas frecuencias. Presentan ceros a bajas frecuencias y polos a altas frecuencias.
 - Filtro paso banda: Son aquellos que permiten el paso de componentes frecuenciales contenidos en un determinado rango de frecuencias, comprendido entre una frecuencia de corte superior y otra inferior.
 - Filtro elimina banda (o rechaza banda): Es el que se opone al paso de componentes frecuenciales contenidos en un determinado rango de frecuencias, comprendido entre una frecuencia de corte superior y otra inferior.
- **Filtros activos y pasivos**
 - Filtro pasivo: Es el constituido únicamente por componentes pasivos como condensadores, bobinas y resistencias.
 - Filtro activo: Es aquel que puede presentar ganancia en toda o parte de la señal de salida respecto a la de entrada. En su implementación se combinan elementos activos y pasivos.
- **Filtros analógicos o digitales**
 - Filtro analógico: Diseñado para el tratamiento de señales analógicas.
 - Filtro digital: Diseñado para el tratamiento de señales digitales.

Los problemas de filtrado se hacen especialmente críticos en aplicaciones de tipo espacial, donde los requerimientos de rechazo, group delay, térmicos... son muy exigentes. Además, a menudo estos requerimientos vienen acompañados con la necesidad de manejar potencias altas, sobre todo en las etapas de los transmisores de los satélites.

En cuanto a los parámetros de dispersión o parámetros S podemos decir que son una herramienta para caracterizar un sistema de microondas. Estos parámetros relacionan las amplitudes normalizadas de las ondas reflejadas en función de las ondas positivas o incidentes.

$$\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{pmatrix} = \begin{pmatrix} S_{11} & S_{12} & \dots & S_{1N} \\ S_{21} & S_{22} & \dots & S_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ S_{N1} & S_{N2} & \dots & S_{NN} \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{pmatrix}$$

En el caso de un filtro, los parámetros S serán selectivos en frecuencia, lo que quiere decir que variarán con ésta. En estos dispositivos, al tener dos puertos, la matriz anterior quedará de la siguiente forma:

$$\begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}$$

Además se cumplirán una serie de propiedades:

- Dispositivo pasivo: $|S_{ij}| \leq 1$
- Simetría $S_{11} = S_{22}$
- Reciprocidad: $S_{12} = S_{21}$
- $|S_{11}| + |S_{21}| + \text{pérdidas} = 1$

A continuación mostramos la respuesta típica de un filtro pasa banda, en la que podemos ver las siguientes partes:

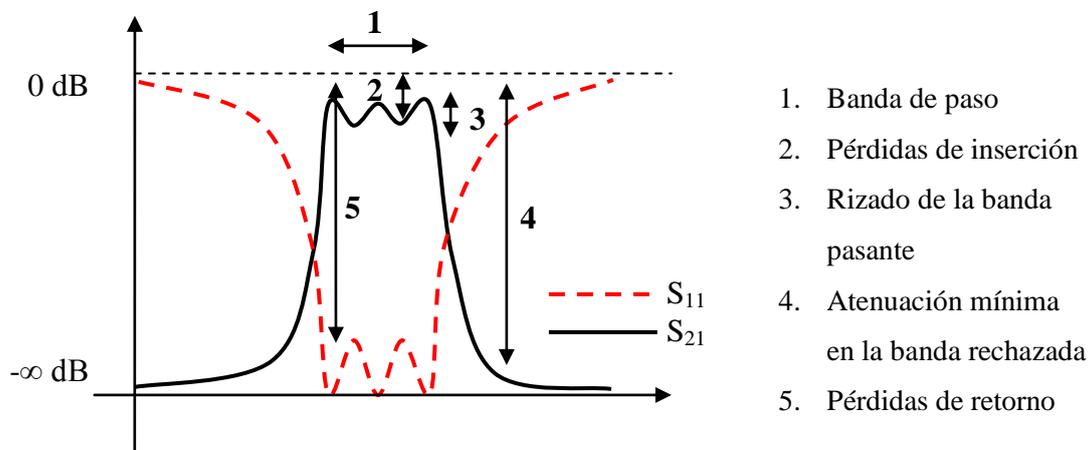


Ilustración 1.1 Respuesta de un filtro pasa banda

Como podemos ver en la imagen, en la banda de paso del filtro el parámetro S_{21} se acerca a la unidad (0 dB), lo que significa que toda la energía se transmite. En la

banda rechazada el módulo del parámetro S_{21} se acerca a cero ($-\infty$ dB), lo que significa que la señal no se transmite. Podemos verlo de forma equivalente en el parámetro S_{11} ya que en la banda de paso el parámetro S_{11} se acerca a 0 ($-\infty$ dB) lo que significa que nada se refleja. Para la banda rechazada el parámetro S_{11} se acerca a 1 (0 dB), luego la señal se refleja.

1.3 Guía de onda

La guía de onda es una estructura física capaz de guiar las ondas electromagnéticas. Este tipo de líneas de transmisión presentan características muy buenas como bajo ruido y gran ancho de banda. Además presentan una fácil fabricación. Como inconvenientes de esta tecnología podemos mencionar que las estructuras ocupan un alto volumen y elevado peso. También presentan una difícil caracterización

Las guías de onda se analizan resolviendo las ecuaciones de Maxwell. Estas ecuaciones tienen soluciones múltiples, conocidas como modos. Los modos de propagación dependen de la longitud de onda, de la polarización y de las dimensiones de la guía. El modo longitudinal de una guía de onda es un tipo particular de onda estacionaria formado por ondas confinadas en la cavidad. Los modos transversales se clasifican en tipos distintos:

- Modos TE (Transversal eléctrico): En ellos, la componente del campo eléctrico en la dirección de propagación es nula.
- Modos TM (Transversal magnético): la componente del campo magnético en la dirección de propagación es nula.
- Modos TEM (Transversal electromagnético), la componente tanto del campo eléctrico como del magnético en la dirección de propagación es nula.
- Modos híbridos, son los que sí tienen componente en la dirección de propagación tanto en el campo eléctrico como en el magnético.

En una guía multimodo, la energía se distribuye entre los modos que se propagan y la instrumentación requerida para detectar la energía total es más compleja y cara, por lo que lo más deseable es operar la guía de onda con un solo modo.

El orden en el que los diferentes modos comienzan a propagarse en la guía depende de su frecuencia de corte. En guías de onda rectangulares, el primer modo en propagarse es el TE_{10} y por ello se le conoce como modo fundamental de la guía. El ancho de banda de la guía vendrá limitado por la aparición de modos de orden superior.

1.4 Objetivo del proyecto

El objetivo de este trabajo será por tanto crear una herramienta que permita el diseño asistido de filtros en guía de onda para aplicaciones espaciales. La herramienta que proponemos consta de tres pasos:

1. Generación de medidas software, para lo cual utilizamos el programa FEST de la Agencia Espacial Europea (ESA). Este programa permite el análisis de estructuras en guía de onda pero no su diseño.
2. Red Neuronal. Una red neuronal es capaz de aproximar de una forma muy precisa cualquier función. Entrenando una red neuronal con las medidas software generadas con FEST conseguiremos relacionar los parámetros eléctricos con dimensiones físicas de una forma muy precisa.
3. En caso de que sea necesario utilizaremos algoritmos de optimización para conseguir una respuesta del filtro que se ajuste todo lo posible a las especificaciones.

A continuación podemos ver un diagrama de bloques de la herramienta que proponemos. Esta herramienta está implementada en MATLAB y se comunica con FEST para generar los medidas software necesarias y con FORTRAN para utilizar la red neuronal. La motivación de no utilizar una red neuronal ya implementada en MATLAB se basa en que las redes que utiliza MATLAB usan el mismo valor de varianza para ambas entradas. En nuestro caso queremos que cada entrada tenga un valor de varianza distinto, ya que variarán en distintos rangos.

En cuanto al programa FEST de la ESA, queremos destacar que sólo dispone de versión para el sistema operativo Linux por lo que realizaremos todo el desarrollo de la herramienta en dicho s.o.

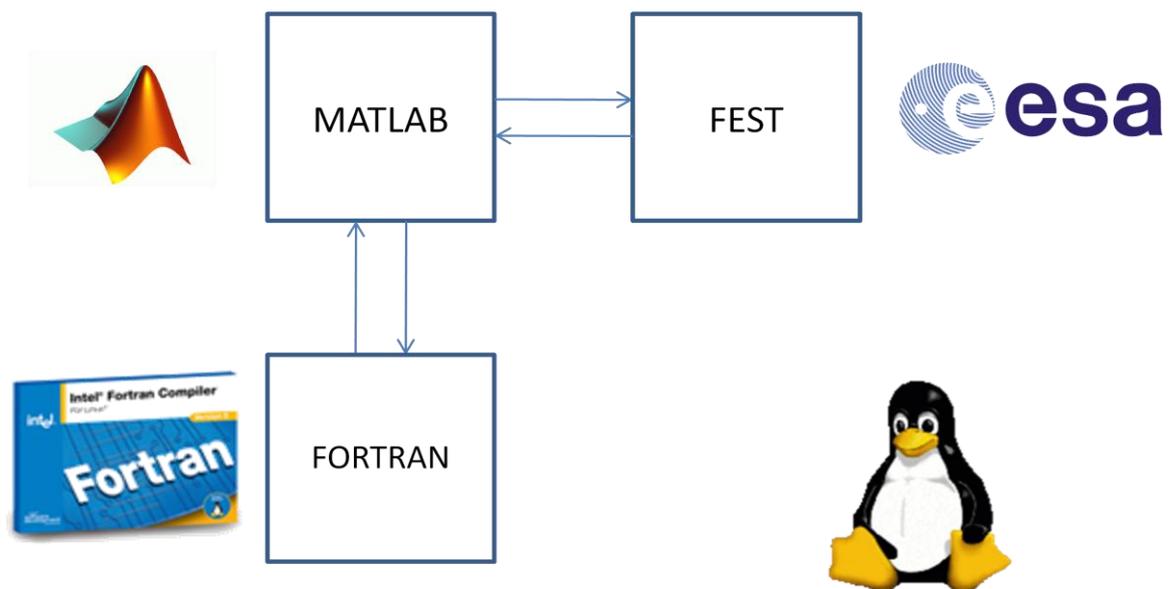


Ilustración 1.2 Diagrama de bloques de la herramienta

1.5 Organización del proyecto

Una vez explicado el objetivo de este trabajo, haremos un pequeño resumen sobre la estructura del mismo.

Comenzaremos el trabajo dedicando un capítulo al programa FEST, hablando de su interfaz, opciones de configuración, etc. Esta descripción servirá para dar una pequeña guía para su uso.

Continuaremos explicando la generación de las medidas software, así como los tipos de medidas utilizadas y los métodos para obtener los parámetros buscados en cada uno de ellos.

Dedicaremos el cuarto capítulo a las redes neuronales, introduciendo los tipos de redes, algoritmos de entrenamiento... Explicaremos también las motivaciones que nos han llevado a utilizar este tipo de solución y mostraremos algunos ejemplos de entrenamiento.

El siguiente capítulo englobará todo lo explicado para tratar el proceso de diseño, introduciendo también los métodos para obtener el equivalente paso bajo del filtro y la obtención de los parámetros a sintetizar. También se tratarán en este capítulo los algoritmos de optimización que hemos usado.

El capítulo 6 estará dedicado a ejemplos de diseño. Los ejemplos mostrados estarán centrados a una frecuencia de 1400 MHz, ya que el objetivo es recibir en la banda del hidrógeno. La motivación de utilizar esta banda se debe a que el hidrógeno es el elemento más abundante del universo, suponiendo más del 75% en masa y más del 90% en número de átomos. Además este elemento se encuentra en abundancia en las estrellas y los planetas gaseosos gigantes.

Finalizaremos el trabajo con las conclusiones y las líneas futuras que podríamos seguir.

2. FEST (Fullwave Electromagnetic Simulation Tool)

Comenzaremos hablando de la herramienta utilizada para realizar las medidas software necesarias. Se trata del programa FEST, desarrollado por la Agencia Espacial Europea (ESA), que permite analizar de una forma muy rápida y precisa estructuras en guía de onda.

Este programa está basado en la combinación de métodos modales y ecuación integral, dentro de la técnica llamada Representación en Red Multimodal Equivalente. La herramienta software es capaz de analizar un gran número de discontinuidades en guía-onda, tales como las inductivas, capacitivas, bifurcaciones, uniones, codos en guía, etc. Uniendo diferentes discontinuidades, se puede realizar el análisis de un gran número de componentes de microondas para aplicaciones espaciales. Uno de estos componentes, utilizado mucho en sistemas reales, son los filtros inductivos en guía-onda.

Dentro del contexto de filtros inductivos, hay que señalar que la herramienta FEST sólo puede realizar el análisis de las estructuras propuestas. En consecuencia, el proceso de diseño de filtros prácticos puede ser largo, especialmente para filtros de órdenes altos.

Podemos ver su interfaz a continuación:

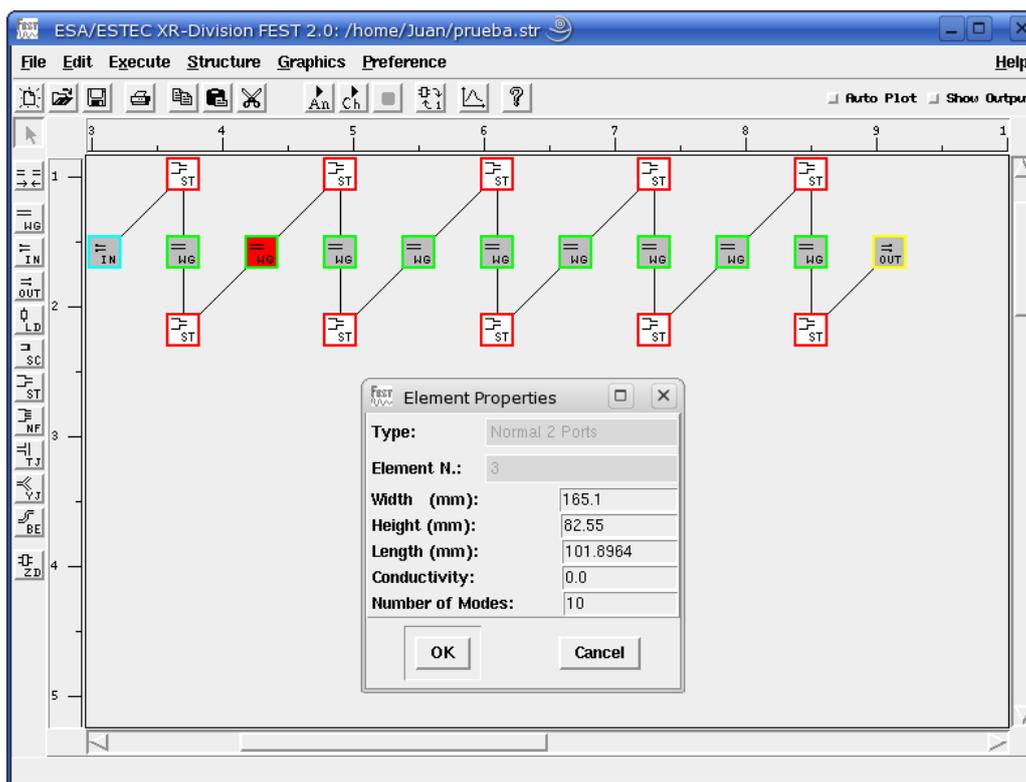


Ilustración 2.1 Interfaz de la herramienta FEST

Para modelar una estructura tendremos que ir añadiendo elementos de forma sucesiva, dotando a cada uno de sus características: altura, anchura y longitud de la guía, conductividad y número de modos, como se puede apreciar en la Ilustración 2.1.

Cuando deseemos introducir una ventana, ya sea inductiva o capacitiva deberemos usar un bloque *Step*, que introduce el offset entre los componentes. Podemos elegir una cantidad fija para dicho offset o seleccionar la opción *Offset Symmetrically*, que hará que el offset sea simétrico entre los elementos. Esta será la opción utilizada por todos nuestros análisis.



Ilustración 2.2 Propiedades del elemento Step

Una vez hemos introducido los elementos, deberemos conectar los bloques utilizando la opción *Connect Element*. Situándonos en un bloque *Step* uniremos primero éste con el extremo más grande y luego con el más pequeño. Debemos repetir este proceso sucesivamente hasta completar todas las conexiones. Podemos ver un ejemplo a continuación, donde primero conectaremos el elemento *In Port* y luego el elemento *Wave Guide*, ya que representa a la ventana y por tanto será el extremo más pequeño:

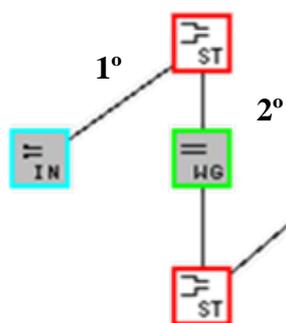


Ilustración 2.3 Ejemplo de conexionado

Las opciones de configuración que nos permite el programa son las siguientes:

- **Frequency Specs:**
 - **Start Frequency:** Frecuencia inicial del análisis expresada en GHz.
 - **End Frequency:** Frecuencia final del análisis expresada en GHz.
 - **Frequency Step:** Paso en frecuencia entre dos medidas consecutivas, expresada en GHz.
- **Execution Specs:**
 - **Functions:** Número de funciones base usadas en la solución de la ecuación integral. Debe ser como mínimo el doble del número de modos utilizados en la estructura. Este parámetro afecta a la precisión del cálculo.
 - **Elements:** Número de términos incluidos en las series infinitas para evaluar el núcleo de la ecuación integral. Debe ser como mínimo 10 veces superior al número de funciones base. Este parámetro afecta a la precisión del cálculo.
- **Overall Specs:**
 - **Override modes:** Número de modos en cada guía usados para el análisis o la caracterización. Si este número es mayor que cero se omitirá el número de modos especificados en cada guía. No puede exceder el valor de 20.
 - **Override conductivity:** Omite la conductividad en todas las guías de onda.

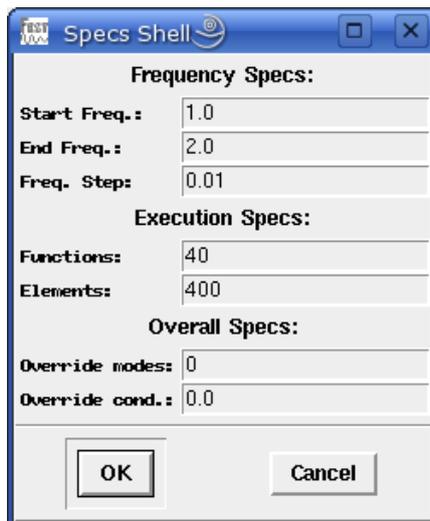
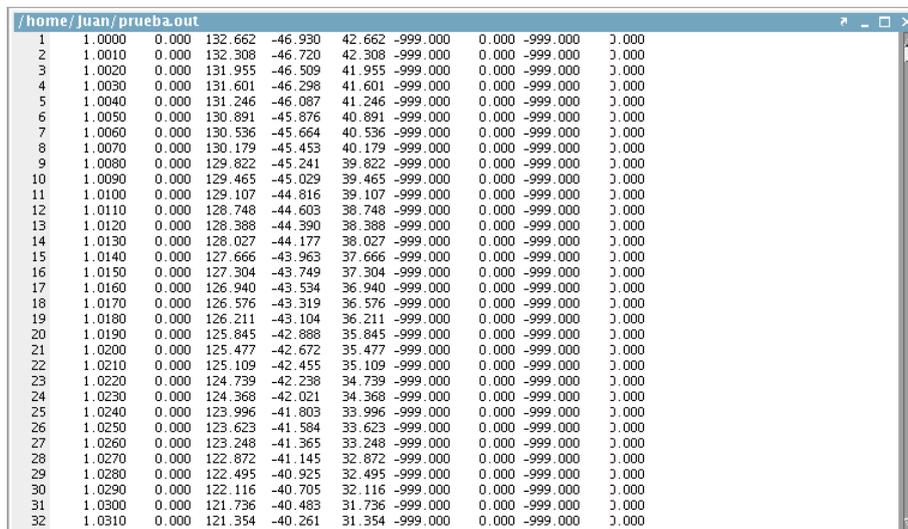


Ilustración 2.4 Opciones de configuración de FEST

Una vez representada la estructura el programa nos da diferentes opciones:

Capítulo 2 - FEST (Fullwave Electromagnetic Simulation Tool)

- **Analizar.** Será la opción utilizada por nuestro trabajo y nos devolverá un archivo de extensión .out que contiene el valor del módulo y la fase de los parámetros de dispersión para cada paso de frecuencia especificado. El programa nos permite representar gráficamente dichos parámetros pero no analizarlos para obtener datos como frecuencia de resonancia y ancho de banda. Por tanto, cargaremos el archivo .out desde MATLAB para poder obtener estos parámetros necesarios para el diseño de filtros.



```
1 1.0000 0.000 132.662 -46.930 42.662 -999.000 0.000 -999.000 0.000
2 1.0010 0.000 132.308 -46.720 42.308 -999.000 0.000 -999.000 0.000
3 1.0020 0.000 131.955 -46.509 41.955 -999.000 0.000 -999.000 0.000
4 1.0030 0.000 131.601 -46.298 41.601 -999.000 0.000 -999.000 0.000
5 1.0040 0.000 131.246 -46.087 41.246 -999.000 0.000 -999.000 0.000
6 1.0050 0.000 130.891 -45.876 40.891 -999.000 0.000 -999.000 0.000
7 1.0060 0.000 130.536 -45.664 40.536 -999.000 0.000 -999.000 0.000
8 1.0070 0.000 130.179 -45.453 40.179 -999.000 0.000 -999.000 0.000
9 1.0080 0.000 129.822 -45.241 39.822 -999.000 0.000 -999.000 0.000
10 1.0090 0.000 129.465 -45.029 39.465 -999.000 0.000 -999.000 0.000
11 1.0100 0.000 129.107 -44.816 39.107 -999.000 0.000 -999.000 0.000
12 1.0110 0.000 128.748 -44.603 38.748 -999.000 0.000 -999.000 0.000
13 1.0120 0.000 128.388 -44.390 38.388 -999.000 0.000 -999.000 0.000
14 1.0130 0.000 128.027 -44.177 38.027 -999.000 0.000 -999.000 0.000
15 1.0140 0.000 127.666 -43.963 37.666 -999.000 0.000 -999.000 0.000
16 1.0150 0.000 127.304 -43.749 37.304 -999.000 0.000 -999.000 0.000
17 1.0160 0.000 126.940 -43.534 36.940 -999.000 0.000 -999.000 0.000
18 1.0170 0.000 126.576 -43.319 36.576 -999.000 0.000 -999.000 0.000
19 1.0180 0.000 126.211 -43.104 36.211 -999.000 0.000 -999.000 0.000
20 1.0190 0.000 125.845 -42.888 35.845 -999.000 0.000 -999.000 0.000
21 1.0200 0.000 125.477 -42.672 35.477 -999.000 0.000 -999.000 0.000
22 1.0210 0.000 125.109 -42.455 35.109 -999.000 0.000 -999.000 0.000
23 1.0220 0.000 124.739 -42.238 34.739 -999.000 0.000 -999.000 0.000
24 1.0230 0.000 124.368 -42.021 34.368 -999.000 0.000 -999.000 0.000
25 1.0240 0.000 123.996 -41.803 33.996 -999.000 0.000 -999.000 0.000
26 1.0250 0.000 123.623 -41.584 33.623 -999.000 0.000 -999.000 0.000
27 1.0260 0.000 123.248 -41.365 33.248 -999.000 0.000 -999.000 0.000
28 1.0270 0.000 122.872 -41.145 32.872 -999.000 0.000 -999.000 0.000
29 1.0280 0.000 122.495 -40.925 32.495 -999.000 0.000 -999.000 0.000
30 1.0290 0.000 122.116 -40.705 32.116 -999.000 0.000 -999.000 0.000
31 1.0300 0.000 121.736 -40.483 31.736 -999.000 0.000 -999.000 0.000
32 1.0310 0.000 121.354 -40.261 31.354 -999.000 0.000 -999.000 0.000
```

Ilustración 2.5 Ejemplo de formato de un archivo .out

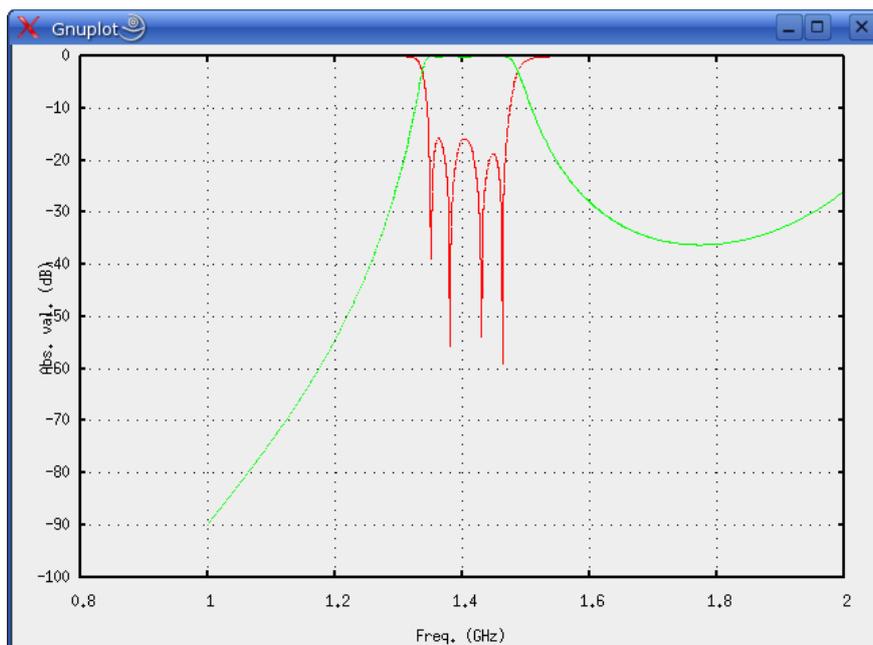


Ilustración 2.6 Representación gráfica del resultado de la simulación

- **Caracterizar**, que nos devuelve un archivo de extensión .chr que contiene la matriz de parámetros Z de la estructura a caracterizar. Esto nos permitiría poder agrupar varias matrices de parámetros Z para formar estructuras más complejas.

3. GENERACIÓN DE MEDIDAS

A continuación vamos a explicar detenidamente el método que hemos empleado para generar las medidas necesarias para el diseño de filtros.

3.1. Estructura de cada tipo de análisis

El primer paso va a consistir en modelar en FEST las estructuras en guía de onda que vamos a utilizar. Emplearemos dos métodos de medida para el caso de una cavidad (single ended y double ended) que nos proporcionaran medidas de frecuencia de resonancia y factor de calidad. Para el caso de dos cavidades sólo tendremos un método de medida que nos proporcionará medidas de frecuencia de resonancia y acoplo entre-resonadores.

3.1.1. Análisis de una cavidad: caso single ended

Para obtener medidas mediante este método utilizaremos una cavidad terminada en cortocircuito, donde variaremos la anchura de la ventana y la longitud de la cavidad. La estructura y su representación en FEST pueden verse en las figuras que se muestran a continuación:

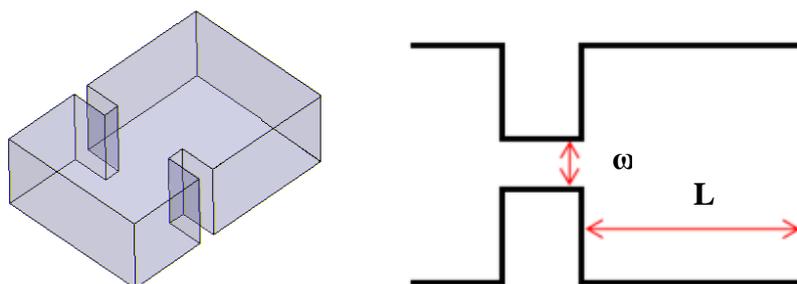


Ilustración 3.1 Cavidad terminada en cortocircuito y dimensiones correspondientes

Como podemos ver en la Ilustración 3.2 para modelar esta estructura en FEST primero colocamos un bloque *Input Port* y tras él un bloque *Wave Guide* de longitud pequeña, para representar la ventana. Utilizaremos un bloque *Step* entre ellos para representar el salto en amplitud ('offset'). Por último utilizaremos un bloque *Short Circuit Load* para representar la cavidad terminada en cortocircuito. Necesitaremos otro bloque *Step* entre ambos para volver a la amplitud de entrada. La conexión de los elementos debe realizarse de la forma explicada anteriormente y los offset serán simétricos.

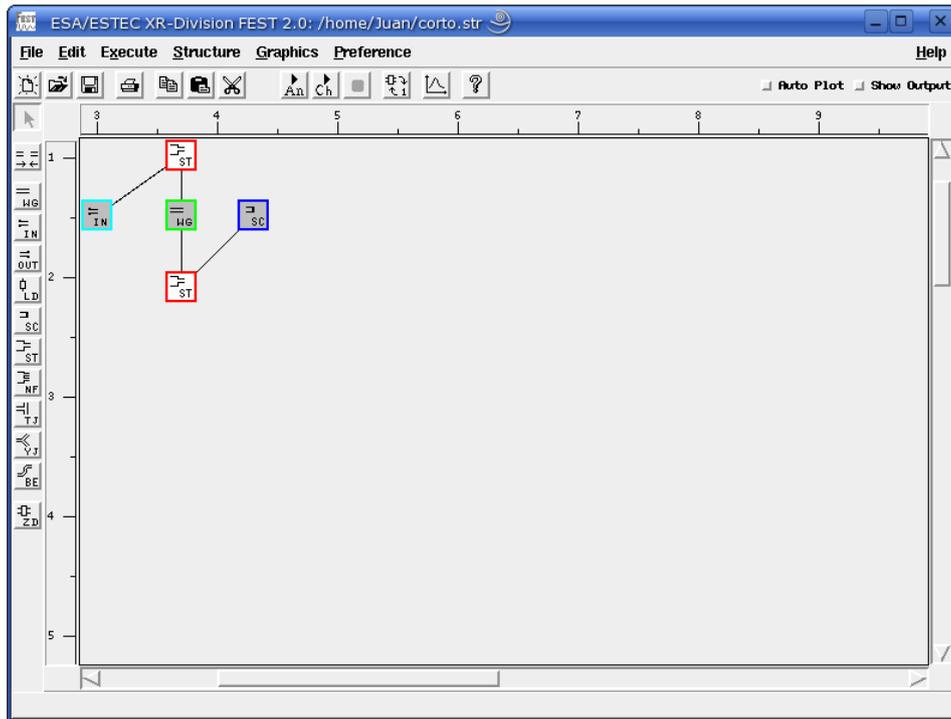


Ilustración 3.2 Representación de la estructura single ended en FEST

La estructura representada en FEST se guarda en un archivo, de extensión .str, que contiene cada uno de los elementos, dimensiones asociadas y conexiones entre ellos.

3.1.2. Análisis de una cavidad: caso Double ended

El otro método para la obtención de medidas utilizando una cavidad se trata del doble ended, cuya estructura y representación en FEST pueden verse en las figuras que se muestran a continuación:

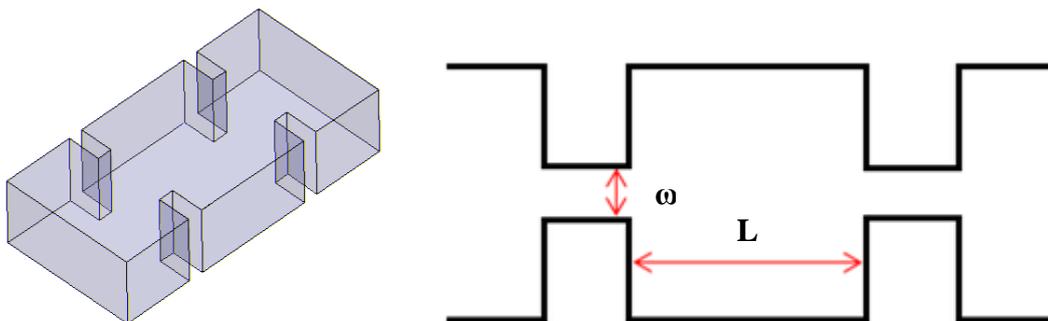


Ilustración 3.3 Cavidad y dimensiones correspondientes

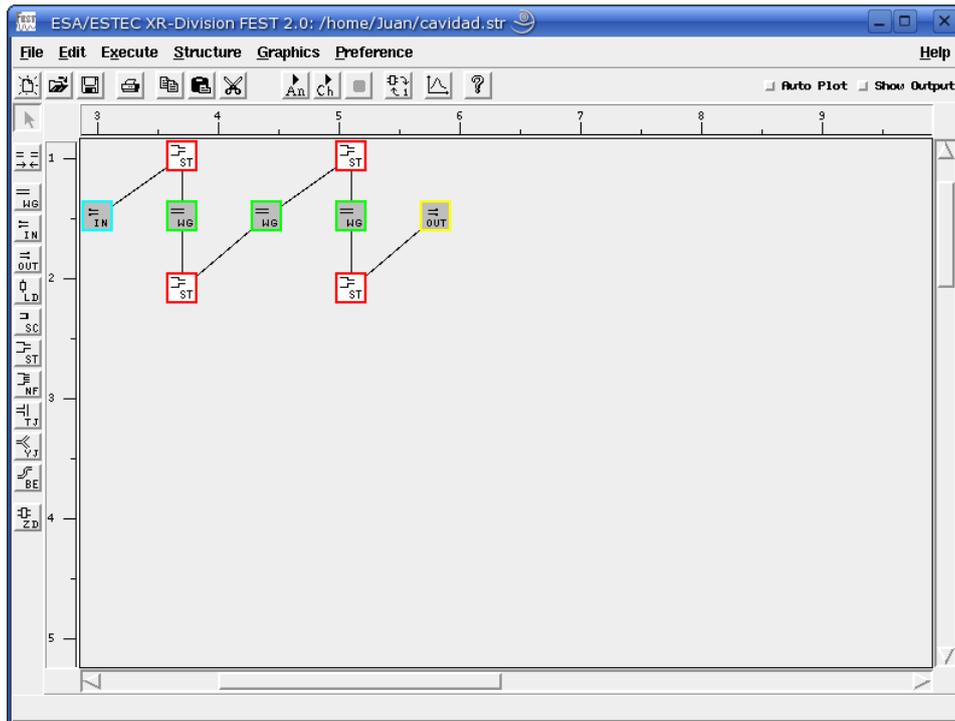


Ilustración 3.4 Representación de la estructura double ended en FEST

Como podemos ver en la imagen, primero colocamos un bloque *Input Port* y tras él utilizaremos un bloque *Step* para representar el salto en amplitud ('offset'). A continuación usaremos un bloque *Wave Guide* de pequeño tamaño, para representar la ventana y seguidamente otro bloque *Step* para representar el salto en amplitud que hay desde la ventana a la cavidad. El siguiente bloque utilizado será un bloque *Wave Guide* que representará la cavidad. En este caso tendremos dos ventanas, por lo que continuaremos con un bloque *Step* que representará el cambio de amplitud de la cavidad a la ventana, otro bloque *Wave Guide* de pequeña longitud que representará la ventana y otro bloque *Step* que representará el 'offset' entre la ventana y la guía. Por último utilizaremos un bloque *Output Port*. La conexión de los elementos debe realizarse del modo explicado anteriormente y todos los offset serán simétricos.

Al igual que en el caso anterior, la estructura representada en FEST se guarda en un archivo, de extensión `.str`, que contiene cada uno de los elementos, dimensiones asociadas y conexiones entre ellos.

3.1.3. Análisis de dos cavidades

Para la obtención de filtros no sólo necesitaremos medidas de frecuencia de resonancia y factor de calidad, también necesitaremos medidas de coeficiente de acoplo. Para ello utilizaremos un filtro de orden $N=2$, pudiendo ver su estructura a continuación:

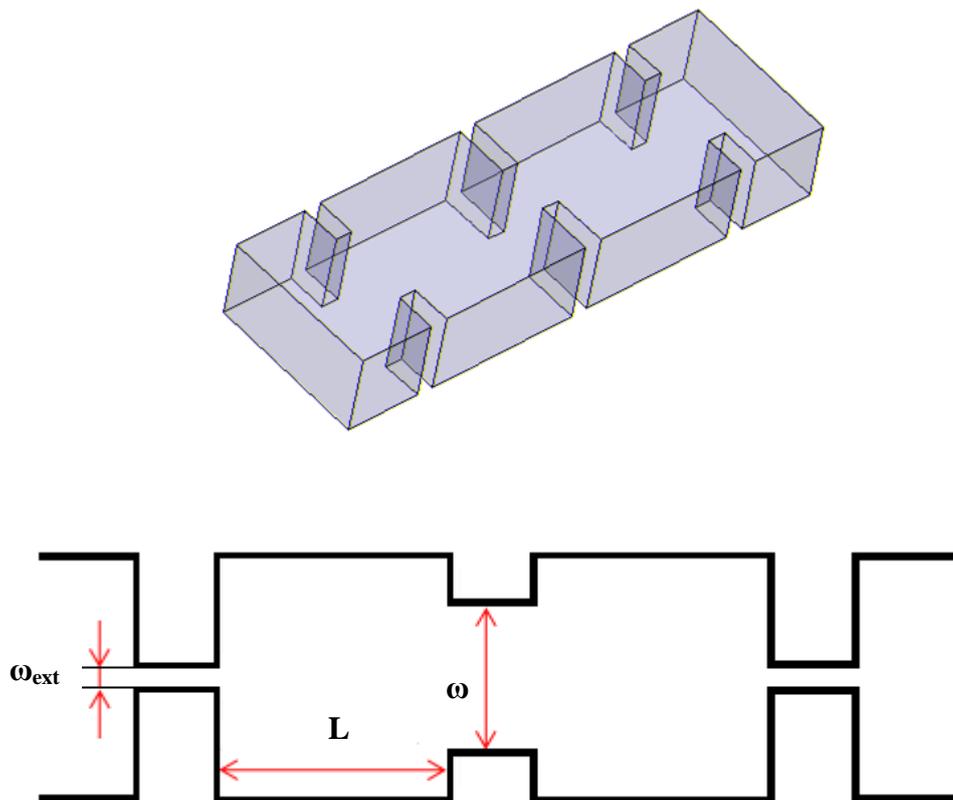


Ilustración 3.5 Filtro de orden 2 y dimensiones correspondientes

Como podemos ver las ventanas externas que hemos utilizado están muy cerradas: así conseguiremos que estas ventanas externas no influyan prácticamente en la longitud de la cavidad. Trataremos este tema con más profundidad en el Anexo II.

Como podemos ver en la Ilustración 3.6, primero colocamos un bloque *Input Port* y tras él utilizaremos un bloque *Step* para representar el salto en amplitud ('offset') entre la guía y la ventana. A continuación usaremos un bloque *Wave Guide* de pequeño tamaño, para representar la ventana externa y seguidamente otro bloque *Step* para representar el salto en amplitud de la ventana a la cavidad. El siguiente bloque utilizado será un bloque *Wave Guide* que representará la primera cavidad. Continuaremos con un bloque *Step* que representará el cambio de amplitud de la cavidad a la ventana, otro bloque *Wave Guide* de pequeña longitud que representará la ventana interna y otro bloque *Step* que representará el offset entre la ventana y la guía. Seguidamente introduciremos un bloque *Wave Guide* que representará la segunda cavidad, de la misma longitud que la anterior. Tras esto introduciremos un bloque *Step* que representará el offset entre la cavidad y la ventana. Tras él introduciremos un bloque *Wave Guide* de pequeño tamaño que representará la ventana externa y un bloque *Step* que de nuevo representa el offset entre ventana y guía. Por último utilizaremos un bloque *Output Port*. La conexión de los elementos debe realizarse del modo indicado en los apartados anteriores y todos los offset serán simétricos.

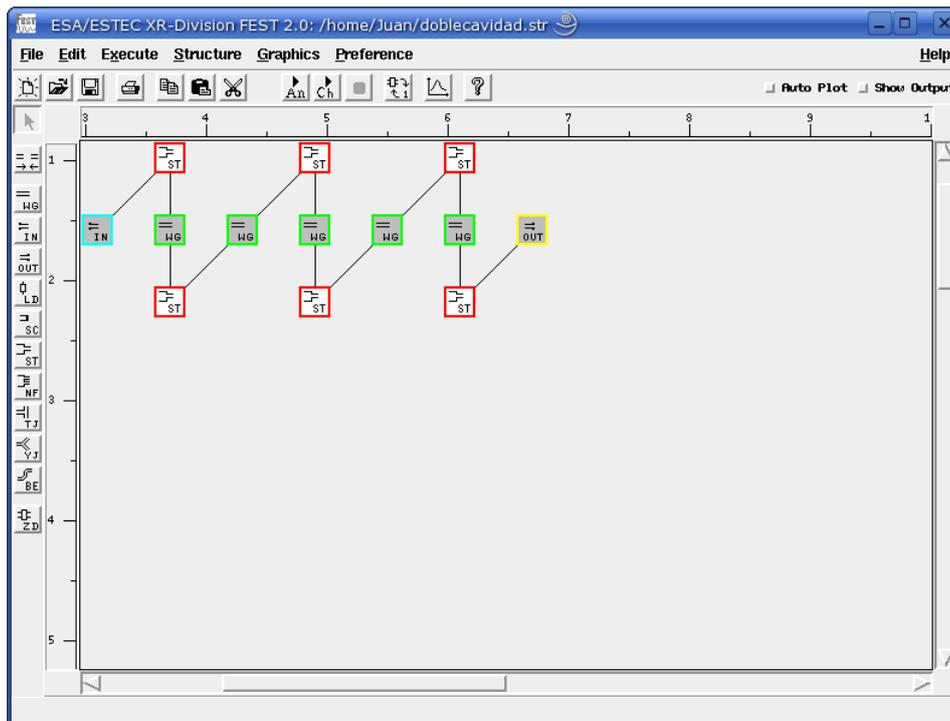


Ilustración 3.6 Representación de la estructura de dos cavidades en FEST

Al igual que en los casos anteriores la estructura se almacenará en un archivo de extensión `.str`.

3.2 Análisis

Debido a las diferencias en la estructura de los casos presentados hemos implementado tres funciones en MATLAB para cada uno de los tipos de análisis. Procederemos a explicar la funcionalidad de cada una de ellas:

3.2.1. Single ended

Para este tipo de análisis hemos implementado la función `calcula_se.m`, cuya estructura es la siguiente:

```
function [Cav, Vent, f, Qe] = calcula_se (ancho, alto, minCavidad, maxCavidad,
pntCav, minVentana, maxVentana, pntVent, minfrec, maxfrec, puntos, poste)
```

Donde, como parámetros de entrada tenemos:

- **Ancho y alto:** Dimensiones de la guía de onda utilizada.
- **minCavidad, maxCavidad y pntCav:** Valores mínimo y máximo de la longitud de la cavidad y número de puntos a analizar entre ambos extremos.

- **minVentana, maxVentana y pntVent:** Valores mínimo y máximo del tamaño de la ventana y número de puntos a analizar entre ambos extremos.
- **minfrec, maxfrec y puntos:** Valores mínimo y máximo de frecuencia para el análisis y número de puntos a analizar entre ambos extremos.
- **Poste:** Longitud de los postes utilizados para las ventanas.

Y, como parámetros de salida:

- **Cav y Vent:** Vectores generados con las longitudes de cavidad y tamaños de ventana simulados.
- **f y Qe:** Matrices en las que se almacenan los valores de la frecuencia de resonancia y factor de calidad para cada uno de los casos analizados.

A partir de los valores que se introducen como parámetros la función crea un vector de *pntCav* longitudes de cavidad linealmente espaciados entre *minCavidad* y *maxCavidad* utilizando la función *linspace*. También creamos un vector de *pntVent* tamaños de ventana equiespaciados entre *minVentana* y *maxVentana*. Veamos un ejemplo utilizando los siguientes datos:

```
minCavidad = 120
maxCavidad = 130
pntCav = 5
```

```
minVentana = 60
maxVentana = 70
pntVent = 5
```

Obtendríamos:

```
Cav = [120.0000 122.5000 125.0000 127.5000 130.0000]
```

```
Vent = [60.0000 62.5000 65.0000 67.5000 70.0000]
```

El análisis consiste en hallar la frecuencia de resonancia y factor de calidad de cada una de las combinaciones longitud de cavidad – tamaño de ventana que hemos creado utilizando la función *linspace*. Para ello se realizará un barrido en dos dimensiones utilizando dos bucles *for* anidados.

```
for m=1:pntCav
    for n=1:pntVent
        %Estructura de modificación del archivo.str
    end
end
```

Capítulo 3 - Generación de medidas

Para cada una de las combinaciones se modifica el archivo *corto.str* donde está almacenada la estructura de FEST que corresponde a este caso, modificando las dimensiones que tenía por las que correspondan. Tras esto se ejecutará FEST con el archivo modificado como parámetro, devolviéndonos los resultados de la simulación en un archivo .out. A partir de aquí podemos cargar el archivo.out desde MATLAB y comenzar a analizarlo.

En este caso, al tener un sólo puerto trabajaremos con el parámetro S_{11} , observando las variaciones en su fase. El módulo de dicho parámetro será igual a la unidad ya que consideremos el resonador sin pérdidas. Tras cargar los vectores de frecuencia y fase deberemos realizar un ajuste en la respuesta de la fase. Este ajuste consistirá en restar 360° a los valores situados tras -180° , como se muestra en la Ilustración 3.7. De esta forma podremos calcular el retardo de grupo.

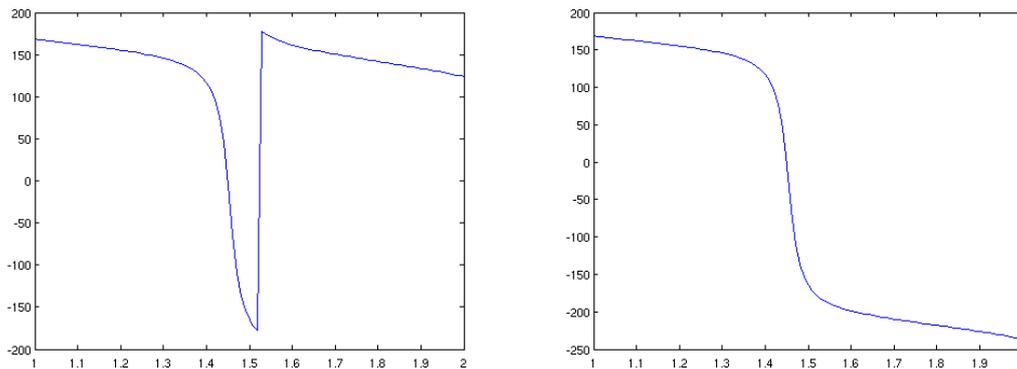


Ilustración 3.7 Ajuste realizado sobre la fase

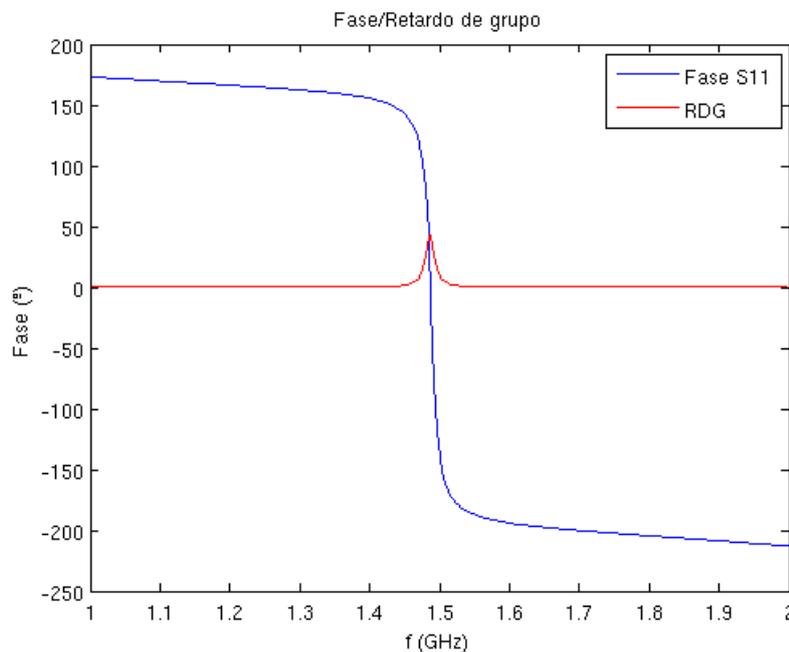


Ilustración 3.8 Fase y retardo de grupo de una cavidad analizada mediante el método single ended

Como podemos ver en la Ilustración 3.8 hallaremos el retardo de grupo, realizando la derivada numérica de de la fase. El máximo del retardo de grupo nos marcará la frecuencia de resonancia de la estructura. Para mejorar la precisión obtenida realizaremos un nuevo análisis con FEST de unos 50 puntos en un pequeño margen alrededor de la frecuencia de resonancia obtenida en el primer análisis.

Utilizaremos dos métodos para calcular el factor de calidad. El método para calcular la frecuencia de resonancia será el que hemos explicado, y será común a ambos. A partir de aquí tendremos dos opciones:

1. Tomar el valor de la frecuencia de resonancia y el del retardo de grupo en ese punto y aplicar la siguiente expresión:

$$Q_e = \frac{\omega_0 \cdot \tau_{s11}(\omega_0)}{4} \quad (3.1)$$

En la Ilustración 3.9 podemos ver un ejemplo gráfico de lo comentado:

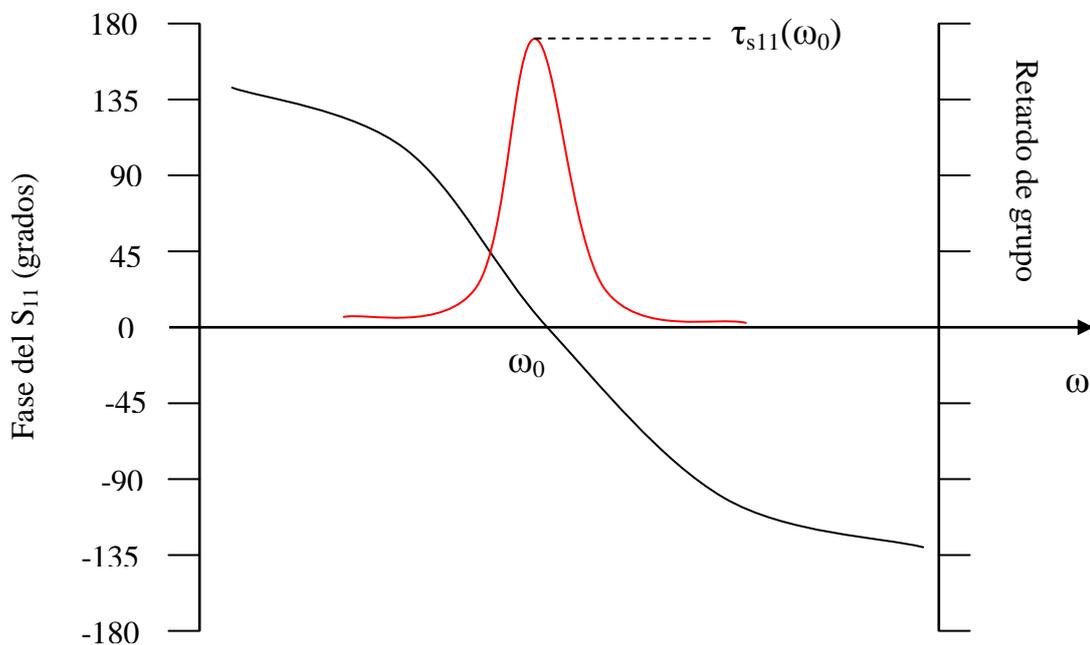


Ilustración 3.9 Respuesta del S₁₁ y retardo de grupo correspondiente

2. Tomar el valor de la fase a la frecuencia de resonancia (ph_0) y buscar las frecuencias situadas a $\pm 90^\circ$ de la fase ph_0 . Podemos verlo gráficamente en la Ilustración 3.10. Con esos valores tendremos que aplicar la siguiente expresión:

$$Q_e = \frac{\omega_0}{\Delta\omega_{\pm 90^\circ}} = \frac{\omega_0}{\Delta\omega_+ - \Delta\omega_-} \quad (3.2)$$

Y, como parámetros de salida:

- **Cav** y **Vent**: Vectores generados con las longitudes de cavidad y tamaños de ventana simulados.
- **f** y **Qe**: Matrices en las que se almacenan los valores de la frecuencia de resonancia y factor de calidad para cada uno de los casos analizados.

A partir de los valores que se introducen como parámetros la función crea un vector de *pntCav* longitudes de cavidad linealmente espaciadas entre *minCavidad* y *maxCavidad* utilizando la función *linspace*. También creamos un vector de *pntVent* tamaños de ventana equiespaciados entre *minVentana* y *maxVentana*. También es válido para este caso el ejemplo mostrado en el caso *single ended*.

El análisis consiste en hallar la frecuencia de resonancia y factor de calidad de cada una de las combinaciones longitud de cavidad – tamaño de ventana que hemos creado utilizando la función *linspace*. Para ello se realizará un barrido en dos dimensiones utilizando dos bucles *for* anidados.

Para cada una de las combinaciones se modifica el archivo *cavidad.str* donde está almacenada la estructura de FEST que corresponde a este caso, modificando las dimensiones que tenía por las que correspondan. Tras esto ejecutaremos FEST con el archivo modificado como parámetro, devolviéndonos los resultados de la simulación en un archivo de extensión *.out*. A partir de aquí podemos cargar ese archivo desde MATLAB y comenzar a analizarlo.

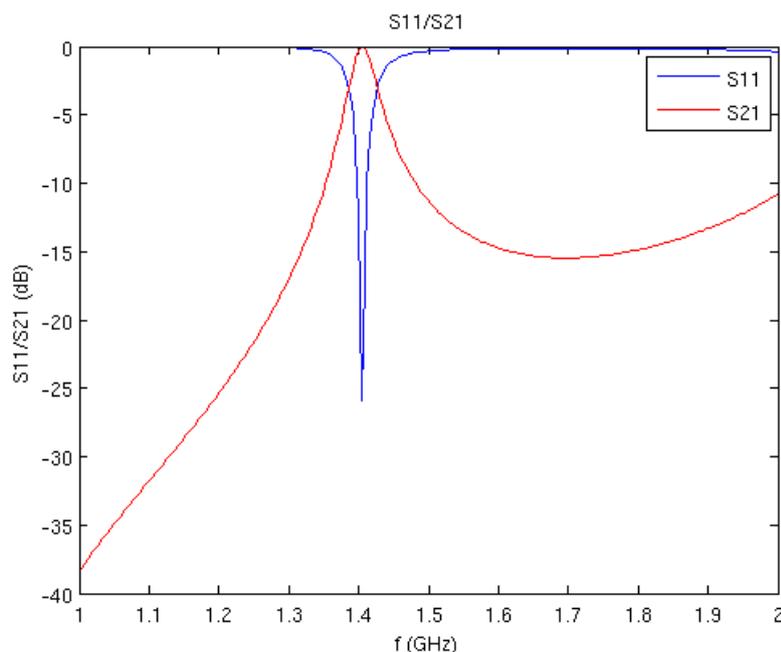


Ilustración 3.11 Parámetros S de una cavidad analizada mediante el método *double ended*

En este caso, al tener dos puertos trabajaremos con el módulo de los parámetros S_{11} y S_{21} . La forma de las curvas obtenidas será la misma que la representada en la Ilustración 3.11.

Para obtener la frecuencia de resonancia buscaremos el máximo del parámetro S_{21} . Para mejorar la precisión obtenida realizaremos un nuevo análisis con FEST de unos 50 puntos en un pequeño margen alrededor de la frecuencia de resonancia. Tras esto será necesario interpolar para hallar los dos valores a los que el módulo del parámetro S_{21} sea -3 dB. Su diferencia nos dará el ancho de banda. Podemos verlo gráficamente a continuación:

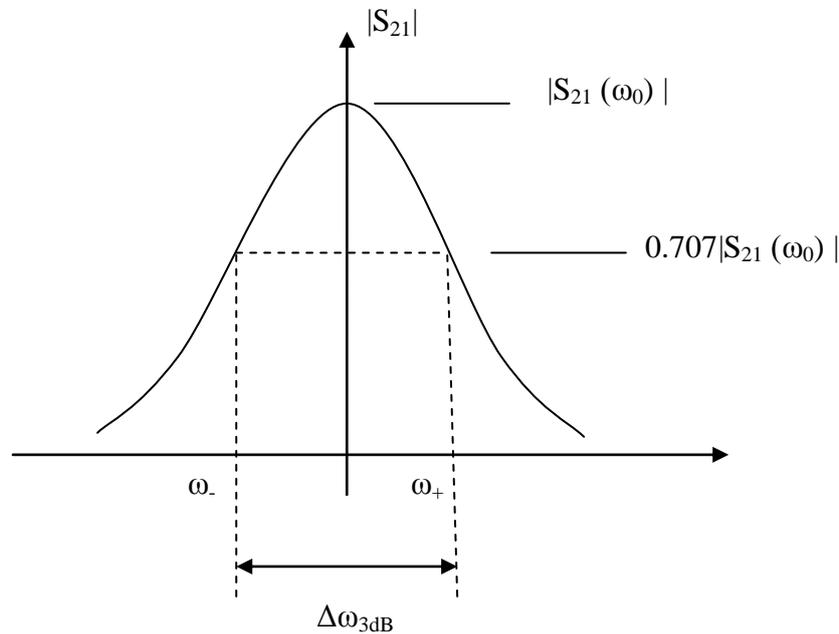


Ilustración 3.12 Evolución del parámetro S_{21}

Mediante la siguiente expresión obtendremos el factor de calidad:

$$Q'_e = \frac{Q_e}{2} = \frac{\omega_0}{\Delta\omega_{3dB}} \quad (3.3)$$

3.2.3 Medida de acoplos

Para este segundo tipo de análisis hemos implementado la función *calcula.m*, cuya estructura es la siguiente:

```
function [Cav, Vent, f, acoplo] = calcula( ancho, alto, minCavidad, maxCavidad,
pntCav, minVentana, maxVentana, pntVent, Ventext, minfrec, maxfrec, puntos,
poste)
```

Donde, como parámetros de entrada tenemos:

- **Ancho y alto:** Dimensiones de la guía de onda utilizada.
- **minCavidad, maxCavidad y pntCav:** Valores mínimo y máximo de la longitud de la cavidad y número de puntos a analizar entre ambos extremos.
- **minVentana, maxVentana y pntVent:** Valores mínimo y máximo del tamaño de la ventana interna y número de puntos a analizar entre ambos extremos.
- **Ventext:** Valor asignado al tamaño de las ventanas externas, constante para todos los análisis.
- **minfrec, maxfrec y puntos:** Valores mínimo y máximo de frecuencia para el análisis y número de puntos a analizar entre ambos extremos.
- **Poste:** Longitud de los postes utilizados para las ventanas.

Y, como parámetros de salida:

- **Cav y Vent:** Vectores generados con las longitudes de cavidad y tamaños de ventana simulados.
- **f y acoplo:** Matrices en las que se almacenan los valores de la frecuencia de resonancia y acoplo entre-resonadores para cada uno de los casos analizados.

A partir de los valores que se introducen como parámetros la función crea una matriz de *pntCav* longitudes de cavidad linealmente espaciadas entre *minCavidad* y *maxCavidad* utilizando la función *linspace*. También creamos una matriz de *pntVent* tamaños de ventan equiespaciados a entre *minVentana* y *maxVentana*.

El análisis consiste en hallar la frecuencia de resonancia y el coeficiente de acoplo entre-resonadores de cada una de las combinaciones longitud de cavidad – tamaño de ventana que hemos creado utilizando la función *linspace*. Para ello se realizará un barrido en dos dimensiones utilizando dos bucles *for* anidados.

Para cada una de las combinaciones se modifica el archivo *doblecavidad.str* donde está almacenada la estructura de FEST que corresponde a este caso, modificando las dimensiones que tenía por las que correspondan. Tras esto ejecutaremos FEST con el archivo modificado como parámetro, devolviéndonos los resultados de la simulación en un archivo *.out*. A partir de aquí podemos cargar el archivo.out desde MATLAB y comenzar a analizarlo.

En el caso de medida de coeficiente de acoplo tendremos dos puertos, al igual que en el caso anterior, por lo que realizaremos las medidas sobre el módulo de los parámetros S_{11} y S_{21} . Al representarlos gráficamente obtendremos curvas como la que sigue:

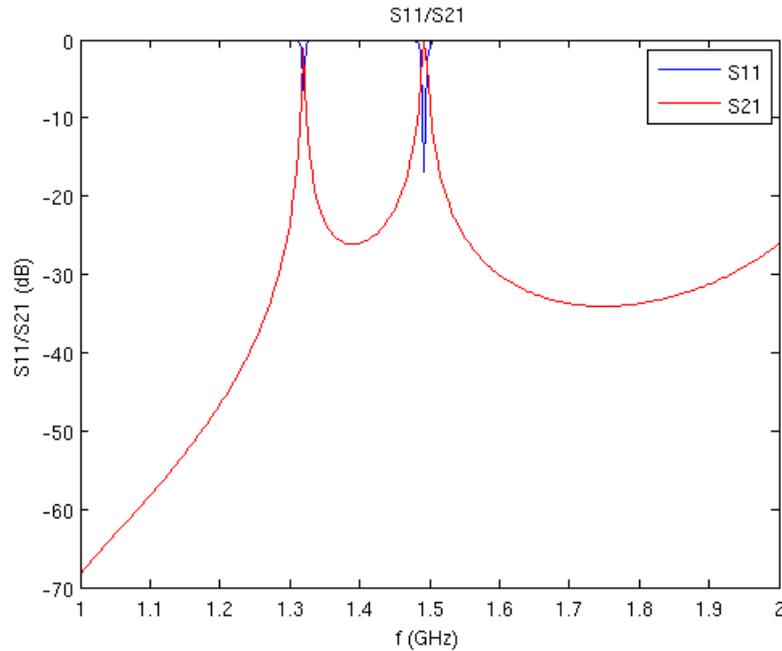


Ilustración 3.13 Parámetros S de la estructura de dos cavidades

Para hallar la frecuencia de resonancia buscaremos los máximos del módulo del parámetro S_{21} . Para cada uno de ellos realizaremos un nuevo análisis con FEST en torno a las frecuencias obtenidas en este primer análisis. Con este refinamiento mejoramos la precisión de los valores obtenidos.

La frecuencia de resonancia de la estructura será la media de los dos valores obtenidos, como podemos ver en la siguiente expresión:

$$f_0 = \frac{f_{p1} + f_{p2}}{2} \quad (3.4)$$

En cuanto al acoplo entre-resonadores utilizaremos la siguiente expresión que es capaz de obtener el coeficiente de acoplo de dos resonadores sintonizados asincrónamente, independientemente de que el acoplo sea eléctrico, magnético o mixto.

$$k = \pm \frac{1}{2} \left(\frac{f_{02}}{f_{01}} + \frac{f_{01}}{f_{02}} \right) \sqrt{\left(\frac{f_{p2}^2 - f_{p1}^2}{f_{p2}^2 + f_{p1}^2} \right)^2 - \left(\frac{f_{02}^2 - f_{01}^2}{f_{02}^2 + f_{01}^2} \right)^2} \quad (3.5)$$

En nuestro caso, las estructuras serán simétricas y las cavidades resonarán a la misma frecuencia. Nos encontraremos por tanto en el caso síncrono y la expresión (3.5) se reducirá a la siguiente:

$$k = \pm \frac{f_{p2}^2 - f_{p1}^2}{f_{p2}^2 + f_{p1}^2} \quad (3.6)$$

3.3 Ejemplos

Mostraremos a continuación un ejemplo gráfico de los resultados para cada tipo de análisis, explicando sus características.

- **Análisis de una cavidad mediante el método single ended:**
 - Longitudes de cavidad: 10 puntos entre 120 y 130 mm.
 - Tamaños de ventana: 10 puntos entre 60 y 70 mm.

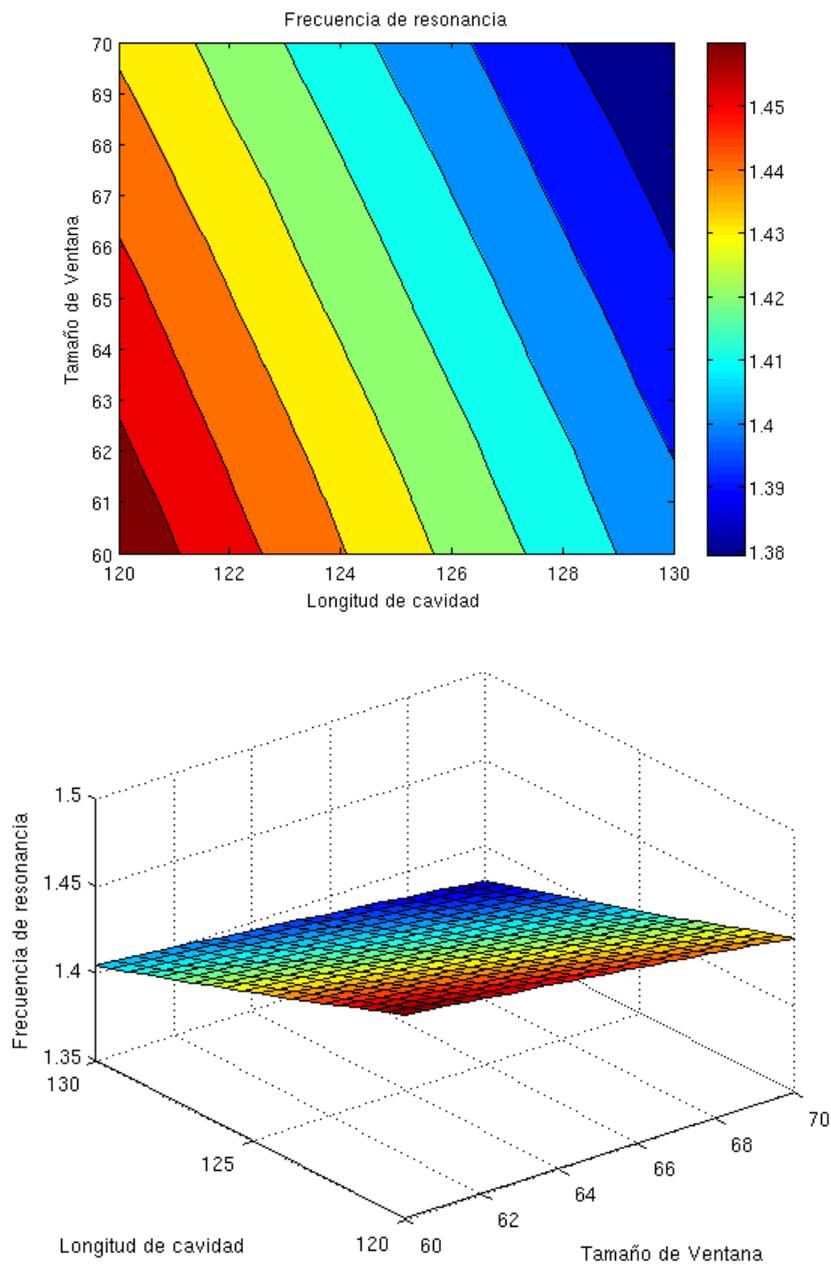


Ilustración 3.14 Evolución de la frecuencia de resonancia con las dimensiones

Podemos ver en la gráfica 3.14 como la frecuencia de resonancia disminuye al aumentar la longitud de la cavidad. También observamos este efecto al aumentar el tamaño de la ventana.

En cuanto al factor de calidad para el método single ended obtenemos gráficas como las siguientes:

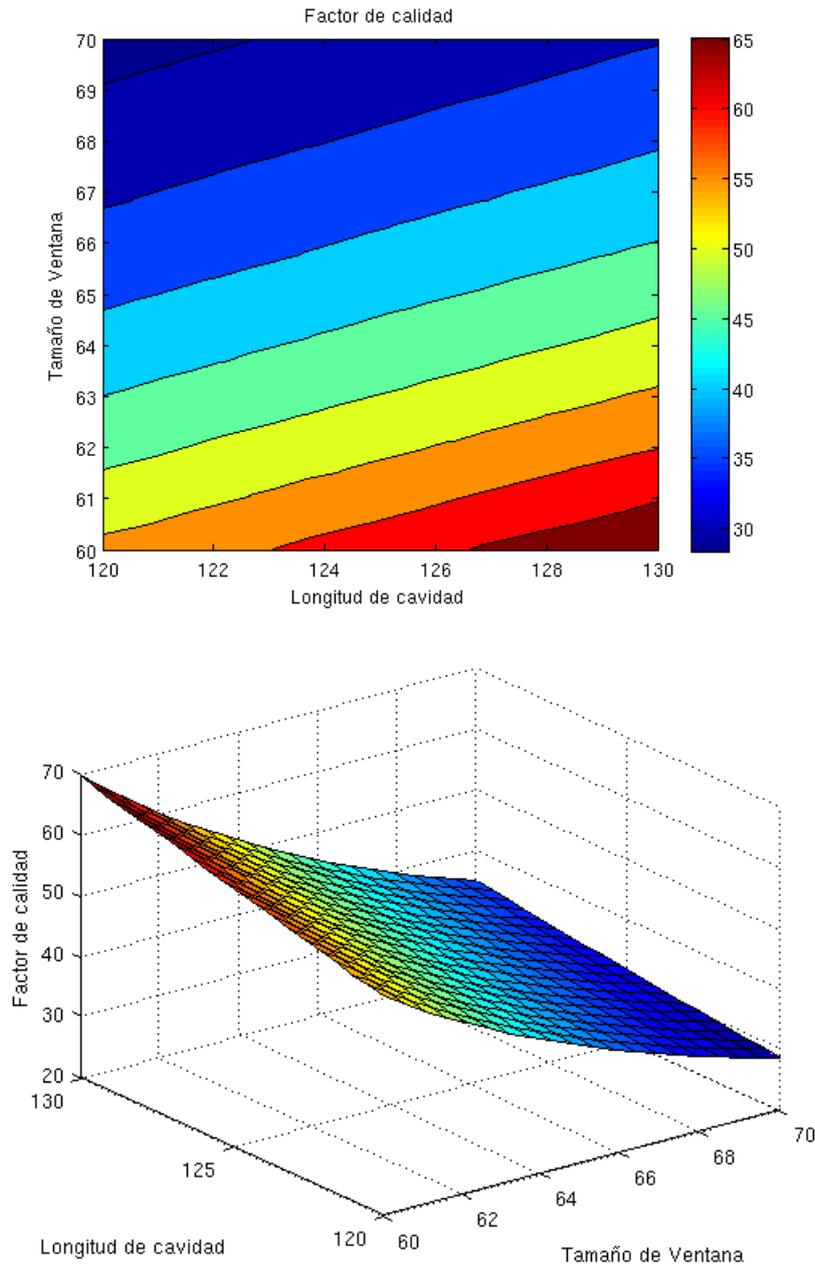


Ilustración 3.15 Evolución del factor de calidad con las dimensiones

Podemos ver en la Ilustración 3.15 como el factor de calidad aumenta al aumentar la longitud de la cavidad y disminuye al aumentar el tamaño de la ventana.

- **Análisis de una cavidad mediante el método double ended:**
 - Longitudes de cavidad: 10 puntos entre 120 y 130 mm.
 - Tamaños de ventana: 10 puntos entre 60 y 80 mm.

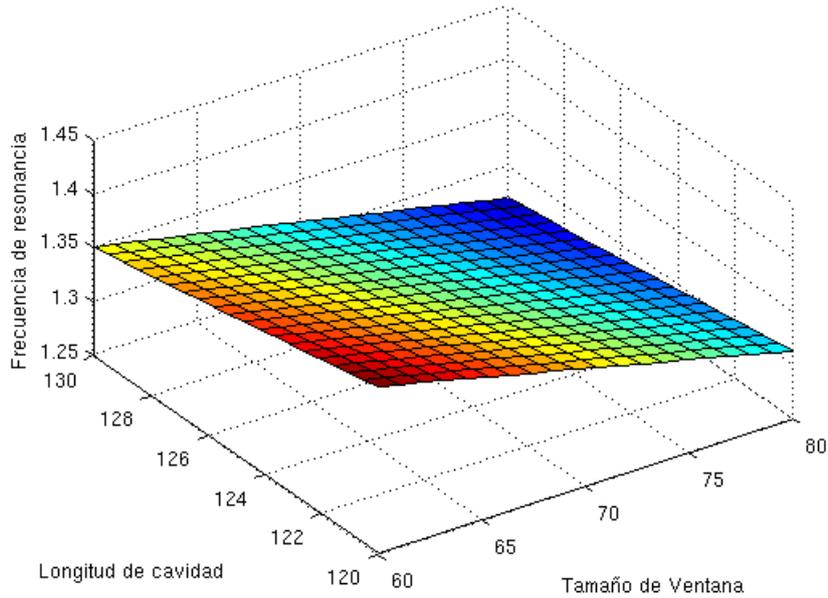
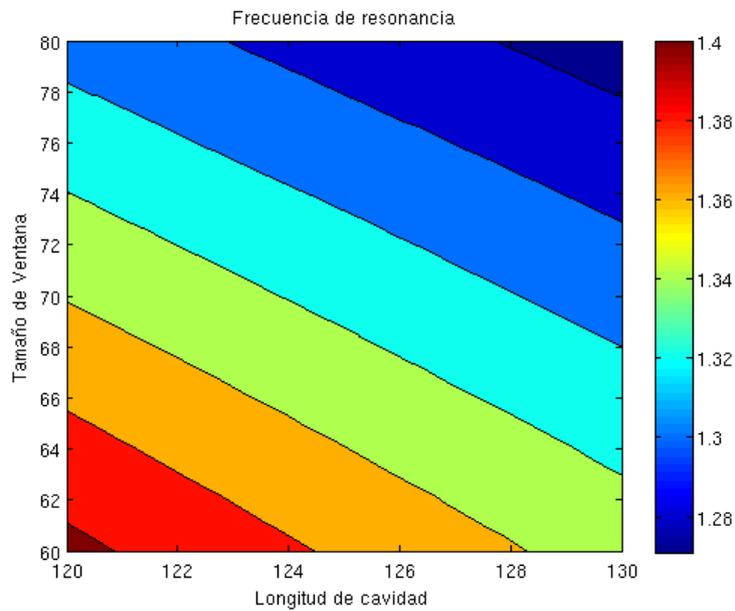


Ilustración 3.16 Evolución de la frecuencia de resonancia con las dimensiones

Podemos ver en la Ilustración 3.16 como la evolución de la frecuencia de resonancia con las dimensiones presenta las mismas características que en el caso single ended. La frecuencia de resonancia disminuye conforme aumenta la longitud de la cavidad y también disminuye conforme aumentamos el tamaño de la ventana.

Representaremos la evolución del factor de calidad en las ilustraciones mostradas a continuación:

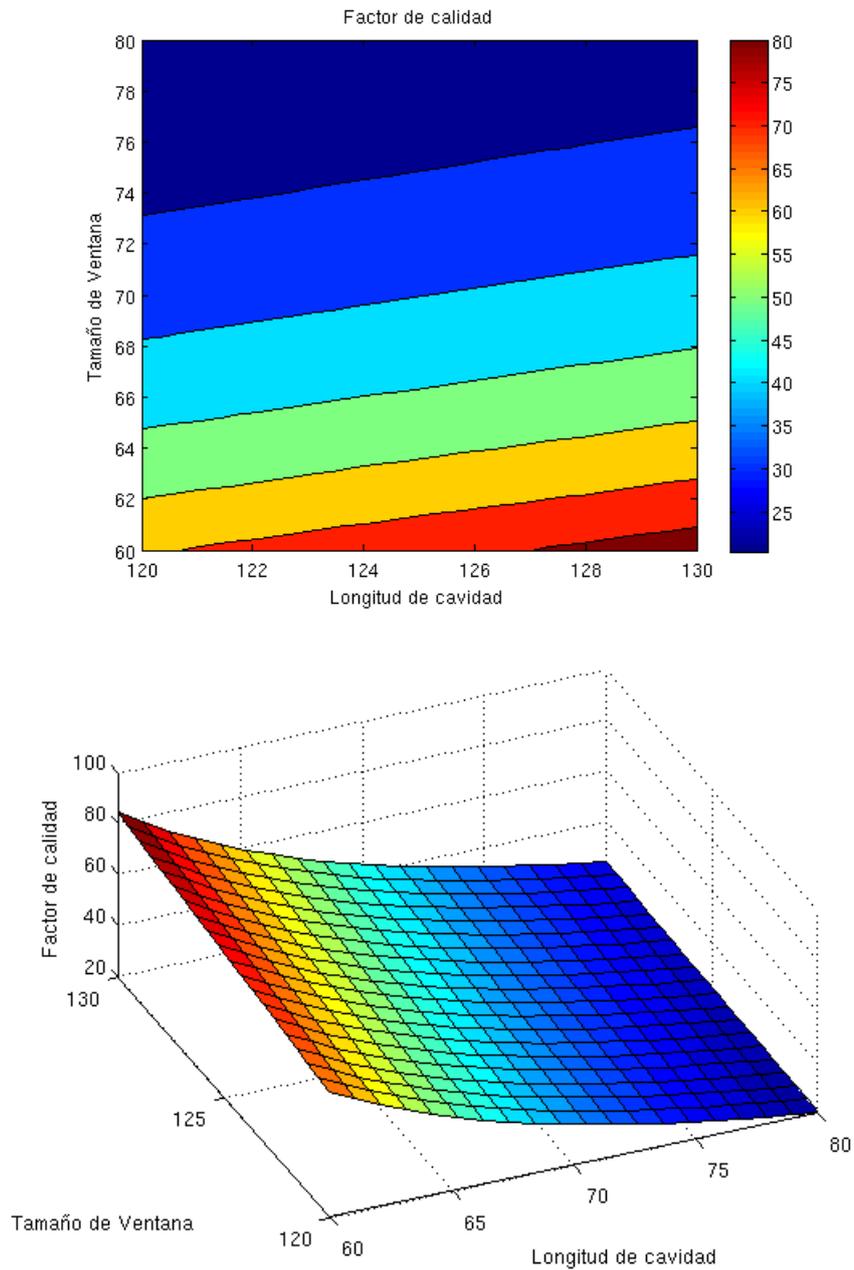


Ilustración 3.17 Evolución del factor de calidad con las dimensiones

Al igual que en el caso anterior y cómo podemos ver en las gráficas el factor de calidad aumenta al aumentar la longitud de la cavidad y disminuye al aumentar el tamaño de la ventana.

• **Análisis de dos cavidades.**

- Longitudes de cavidad: 10 puntos entre 120 y 130 mm.
- Tamaños de ventana: 10 puntos entre 60 y 70 mm.
- Ventana externa: 5 mm.

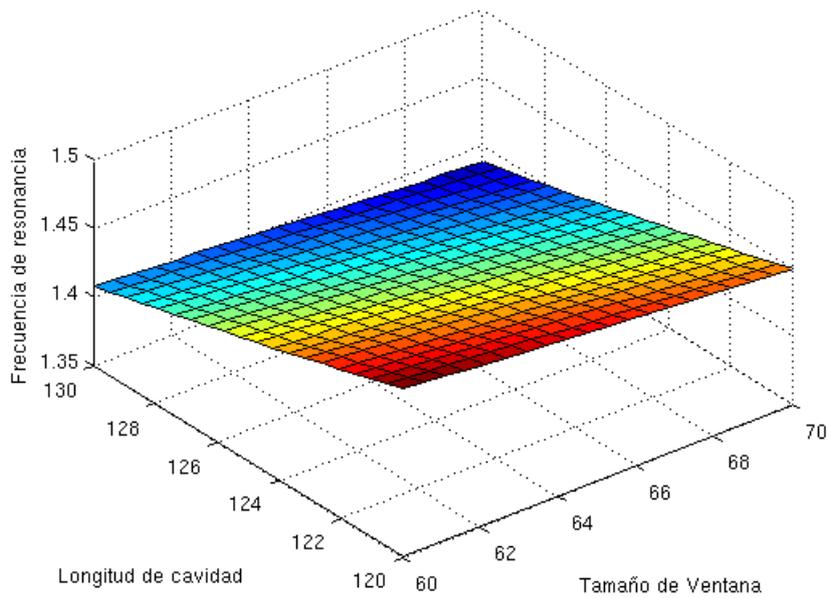
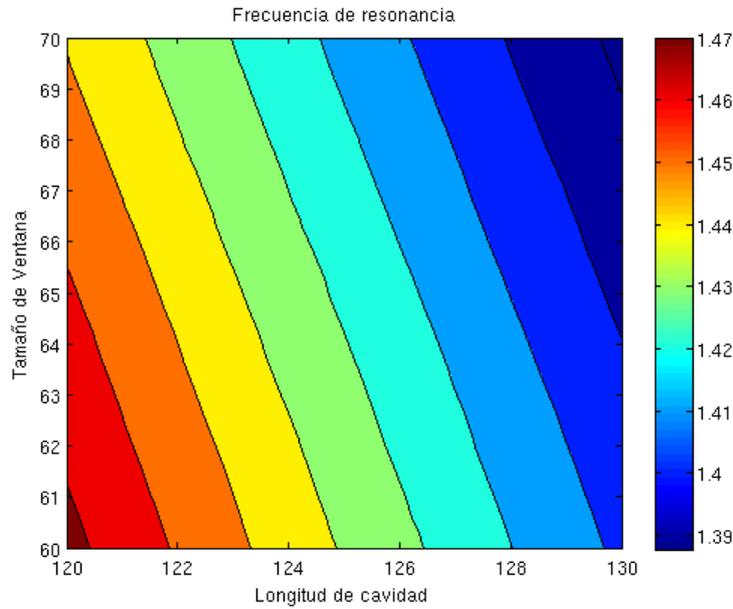


Ilustración 3.18 Evolución de la frecuencia de resonancia con las dimensiones

El resultado en este caso es coherente con las medidas obtenidas con una cavidad. Se aprecia que al aumentar la longitud de la cavidad la frecuencia de resonancia disminuye y al aumentar el tamaño de la ventana interna la frecuencia de resonancia también disminuye.

En cuanto al coeficiente de acoplo podemos ver:

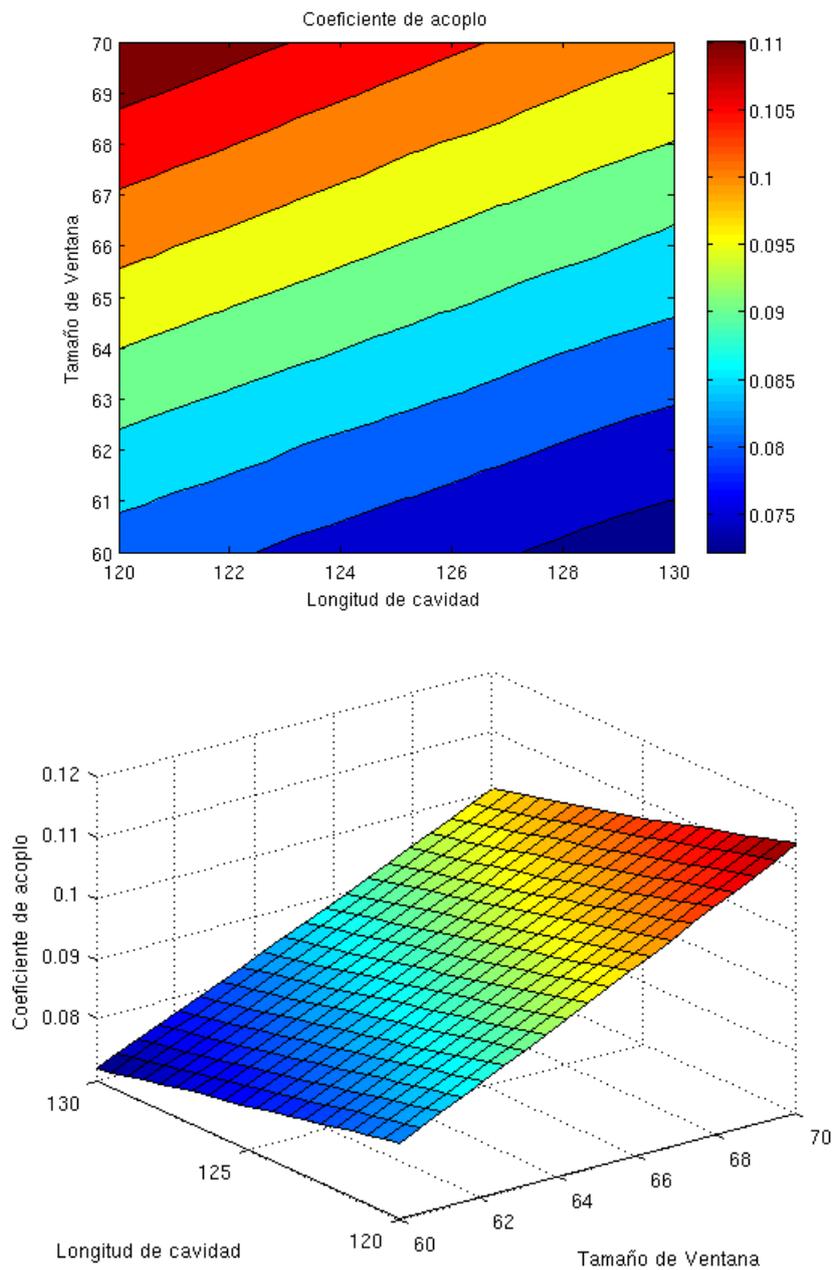


Ilustración 3.19 Evolución del acoplo entre-resonadores con las dimensiones

Podemos ver en la gráfica anterior como el coeficiente de acoplo disminuye conforme crece la longitud de la cavidad y aumenta al hacer mayor el tamaño de la ventana.

4. RED NEURONAL

4.1 ¿Qué es una red neuronal?

Las redes neuronales surgen del estudio de nuestro sistema nervioso tratando de imitar el sistema de computación más complejo de los que se conocen, nuestro cerebro. El cerebro está formado por millones de células llamadas neuronas. Estas neuronas son unos procesadores de información muy sencillos con un canal de entrada de información (dendritas), un órgano de cómputo (soma) y un canal de salida de información (axón).

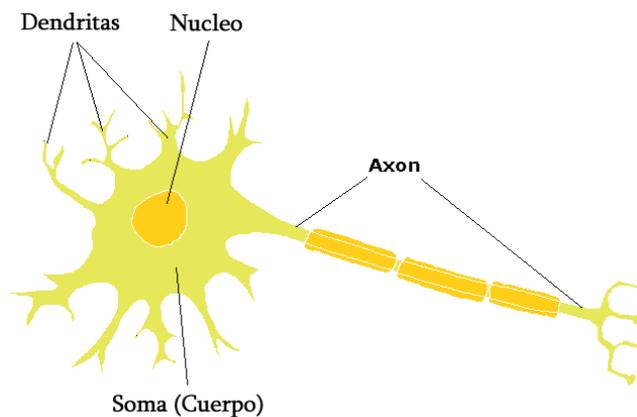


Ilustración 4.1 Estructura de una neurona

Generalmente la estructura de una red neuronal está formada por distintas capas: de entrada, oculta (pueden existir más de una) y de salida. La capa oculta está formada por una serie de unidades denominadas neuronas que proporcionan diferentes valores de salida dependiendo del valor a su entrada. Las neuronas están conectadas a la entrada y a la salida de la red y pueden estar interconectadas entre sí al igual que las neuronas del sistema nervioso. Cada conexión puede tener un peso de forma que el valor en el extremo final de la conexión es fruto de la multiplicación de dicho peso por el extremo final.

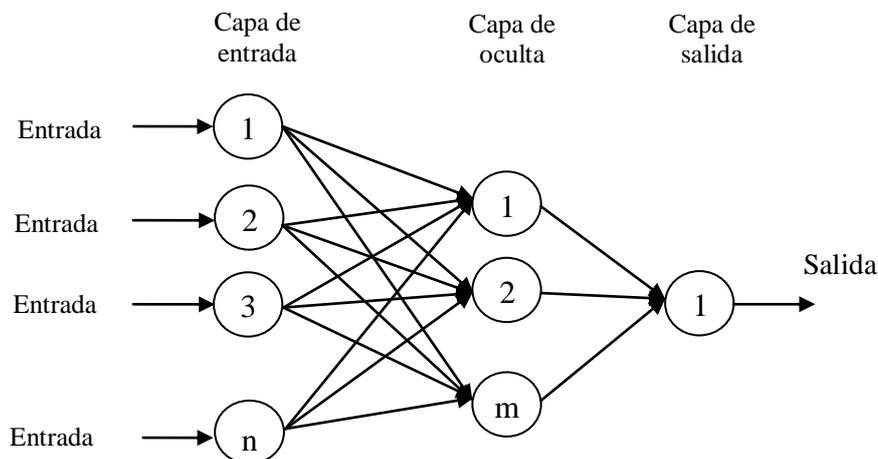


Ilustración 4.2 Estructura de red neuronal artificial

Las redes neuronales han sido aplicadas a un creciente número de problemas prácticos de considerable complejidad como por ejemplo reconocimiento de patrones, clasificación de datos, predicciones, etc.

En el presente trabajo las redes neuronales se han utilizado en tareas de regresión, es decir, en la aproximación de una función objetivo. Las redes neuronales suelen utilizarse en este tipo de problemas por dos motivos diferentes. Bien cuando la generación de la función objetivo es muy costosa o bien cuando la función no se puede calcular analíticamente y sólo se puede conocer a través de su evaluación en una serie de puntos. Las redes neuronales son capaces de interpolar funciones no-lineales en espacios multidimensionales. Además las salidas de la red neuronal se calculan rápidamente y su cálculo es fácilmente paralelizable. Por estas razones se han elegido en el presente trabajo para capturar las relaciones existentes entre las dimensiones de un circuito y sus parámetros eléctricos.

4.2 Tipos de redes neuronales

Los dos tipos más importantes de redes neuronales son los siguientes:

- **“Multilayer Perceptron” (MLP):** en este tipo de redes, cada neurona calcula en su entrada la suma de las entradas ponderadas por los pesos. La salida de cada neurona es el valor de una función sigmoide:

$$y = \varphi \left(\sum_{i=1}^n w_i x_i + b \right) = \varphi(w^T x + b) \quad (4.1)$$

Donde w representa el vector de pesos, x el vector de entradas, b la desviación y φ la función de activación.

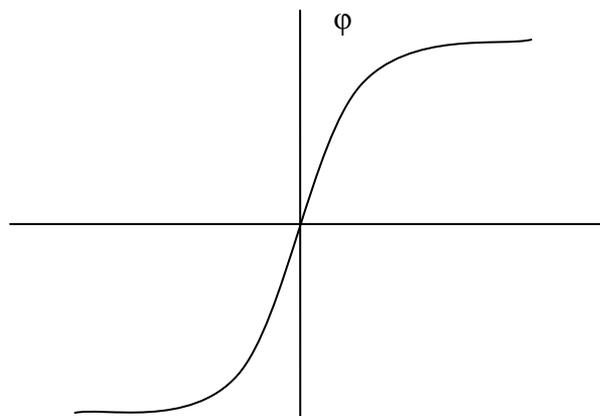


Ilustración 4.3 Función sigmoide

- **“Radial Basis Function Networks” (RBFN):** En este caso cada neurona calcula la distancia euclídea entre un vector que caracteriza dicha neurona y que recibe el nombre de centro o centroide y el vector entrada. Sobre esa distancia se aplica una función de tipo radial con forma gaussiana. Podemos caracterizar la salida de este tipo de red mediante la expresión (4.2):

$$F(x) = \sum_{i=1}^N \omega_i \varphi(\|x - c_i\|) \quad (4.2)$$

Donde φ es la función de base radial, $\|\cdot\|$ es la distancia o norma euclídea, N es el número de pares de entrenamiento, y c_i es el centro de la i ésima función de base radial.

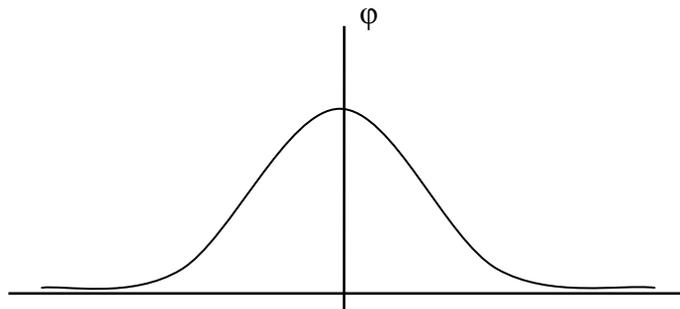


Ilustración 4.4 Función de base radial

En cuanto a las diferencias entre ambos tipos de redes, las RBF se caracterizan porque sólo tienen una capa oculta y por utilizar funciones diferentes. La función de base radial o gaussiana tiende a cero para valores alejados del centro. Debido a este hecho la aproximación de la función objetivo en una RBFN se realiza de forma local, cada grupo de neuronas aproxima una región de la función objetivo. La función sigmoide de las MLP no tiende a cero en ninguna región del espacio de salida de forma que la aproximación en este tipo de redes es global. Por ello la eliminación de una neurona afecta de forma apreciable en el resultado final.

La tarea consistente en el cálculo de los parámetros que definen la red neuronal se denomina entrenamiento. Para calcular dichos parámetros (centros, pesos, varianzas) es necesario contar con una serie de datos que reciben el nombre de set o conjunto de entrenamiento. Estos datos deben estar desprovistos a ser posible de ruido y deben ser un número suficiente para que la red neuronal aprenda las relaciones no-lineales entre la entrada y salida. El entrenamiento debe poseer dos características deseables, por un lado debe ser rápido y por otro debe proporcionar el menor número de neuronas a fin de conseguir el nivel de error en la aproximación deseado. El entrenamiento de una RBF es más sencillo que el de un MLP (**poner referencia**). Además también necesitan un set de entrenamiento mucho menor.

En tecnología de microondas las redes neuronales pueden utilizarse para dos propósitos principalmente:

- A partir de dimensiones físicas obtener parámetros eléctricos. Por ejemplo, podríamos caracterizar los parámetros S de un transistor a partir sus dimensiones físicas.
- A partir de parámetros eléctricos obtener dimensiones físicas. Este es el caso que vamos a utilizar en nuestro trabajo ya que a partir de parámetros como frecuencia de resonancia, factor de calidad o coeficiente de acoplo obtendremos dimensiones físicas. Dichos parámetros eléctricos son obtenidos a partir de las especificaciones del filtro. Generalmente las dimensiones físicas que permiten la obtención de los anteriores parámetros se obtienen mediante curvas o mediante el empleo de fórmulas empíricas. Estos métodos presentan ciertos inconvenientes. Por un lado las fórmulas empíricas no consiguen el grado de precisión requerido por todos los diseños. Por otro lado el empleo de curvas requiere del cálculo de muchos valores para alcanzar cierta precisión. El uso de curvas es complicado en herramientas de CAD automático. Las redes neuronales, una vez entrenadas, solucionan los problemas mencionados ya que ofrecen precisión y una fácil integración con herramientas de CAD. Recordemos además que la red neuronal elegida necesita un conjunto de datos de entrenamiento reducido para aproximar la función objetivo.

Podemos ver en las siguientes imágenes cómo, en nuestro caso, a partir de parámetros eléctricos la red proporciona las correspondientes dimensiones físicas. En el caso de la Ilustración 4.5 tenemos como entradas frecuencia de resonancia y factor de calidad (parámetros eléctricos) y como salidas longitud de cavidad y tamaño de ventana (dimensiones físicas).

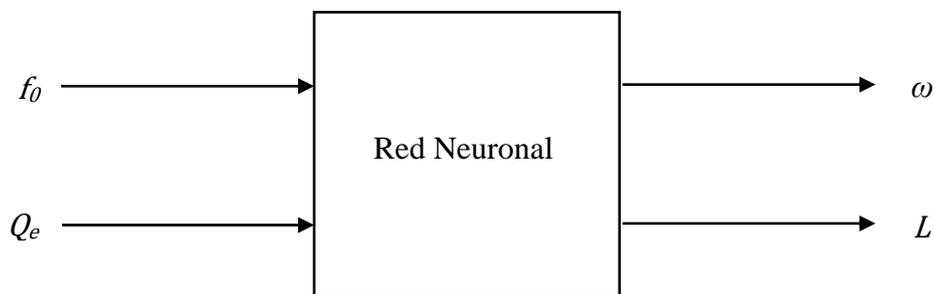


Ilustración 4.5 Estructura de la red neuronal

En el caso de la Ilustración 4.6 tenemos como entradas a la red neuronal frecuencia de resonancia y coeficiente de acoplo (parámetros eléctricos) y como salidas longitud de cavidad y tamaño de ventana externa (dimensiones físicas).

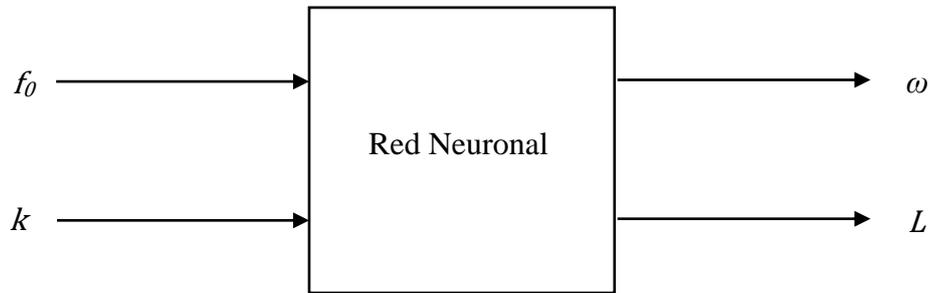


Ilustración 4.6 Estructura de la red neuronal

4.3 Red neuronal utilizada

Debido a las ventajas anteriormente mencionadas elegiremos para nuestro trabajo una RBF, cuya estructura será la siguiente:

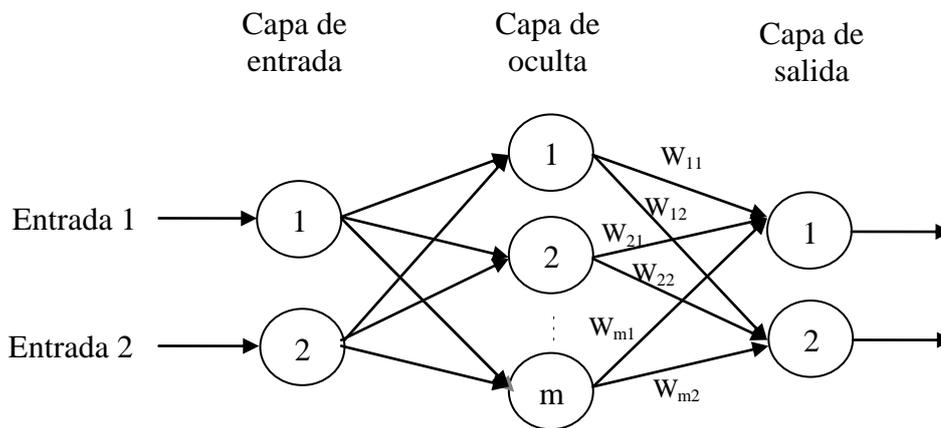


Ilustración 4.7 Estructura de nuestra RBFN

En nuestro caso utilizaremos la red para aproximar las curvas que definen la variación de frecuencia de resonancia, factor de calidad y coeficiente de acoplo con las dimensiones físicas.

4.4 Entrenamiento de la red

El propósito del algoritmo de entrenamiento es elegir los centros, varianzas y los pesos de la capa de salida de la red, aunque la cuestión más importante es la selección de centros. Tenemos dos tipos de estrategias de entrenamiento:

- Métodos supervisados: Tienen en cuenta las salidas deseadas para calcular los pesos. Ciertos métodos también calculan los centros de forma supervisada.
- Métodos no supervisados: Sólo usan las entradas para calcular los centros y las varianzas.

Normalmente, las estrategias de entrenamiento combinan métodos supervisados y no supervisados.

El diseño de una red RBF se compone de dos fases: el proceso de entrenamiento y la fase de test. Durante la primera, la red aprende de un set bien conocido de pares entrada-salida llamado set de entrenamiento. Una vez finalizado el entrenamiento, se comprueba la capacidad de generalización de la red en la fase de test. En dicha fase se le entrega a la red un set de pares entrada-salida distinto al usado en la fase de entrenamiento evaluando el funcionamiento de la red. Generalmente se fija antes del inicio de la fase de entrenamiento un nivel de error que debe rebajar la red tanto para el set de entrenamiento como para el set de test. Conforme avanza el entrenamiento se incrementa el número de neuronas hasta que ambos errores quedan por debajo del nivel de error fijado.

Existen dos situaciones no deseables y que indican un mal entrenamiento de la red neuronal: el sobreentrenamiento y el subentrenamiento. El sobreentrenamiento consiste en que la red no generalice bien para los valores que no hayamos entrenado. De esta forma el error entrenamiento es pequeño y queda por debajo del nivel prefijado de error mientras que el error de test es elevado. Esta situación suele darse cuando el número de neuronas respecto del número de datos de entrenamiento es alto. El subentrenamiento consiste en que la red no funcione bien ni siquiera para valores presentes en el conjunto de entrenamiento. Generalmente es debido a que el número de neuronas es insuficiente y por tanto es necesario aumentarlo hasta alcanzar el nivel de error fijado.

Para explicarlo detenidamente, mostraremos en la Ilustración 4.8 una evolución del coste cuadrático medio con el entrenamiento. En negro podemos ver el coste cuadrático medio asociado al conjunto de entrenamiento y en rojo dicho coste asociado al conjunto de test.

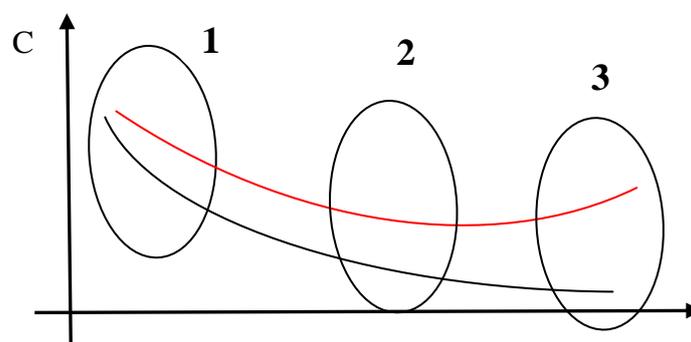


Ilustración 4.8 Evolución del error con el entrenamiento

- En la *zona 1* nos encontraríamos con el problema del subentrenamiento ya que, como hemos dicho, la red cometería mucho error incluso para valores presentes en el conjunto de entrenamiento.

- En la zona 2 tenemos valores similares de error para ambos conjuntos. En esta zona estableceremos un umbral de error para parar el algoritmo de entrenamiento.
- En la zona 3 sufriríamos del problema conocido como sobrentrenamiento. En este caso la red funcionaría bien para los valores que hemos entrenado pero no generalizaría bien para el resto.

Hay numerosos algoritmos de entrenamiento de RBF pero en nuestro caso hemos utilizado los siguientes:

- “Orthogonal Least Squares” (OLS). El método OLS se aplica a la selección de centros de una red RBF. Su objetivo es seleccionar un set adecuado de centros de una manera racional y maximizar la contribución de los centros seleccionados a la respuesta deseada. Este algoritmo iterativo completa el proceso de entrenamiento evaluando los pesos lineales de salida.
- K-means: Es un algoritmo no supervisado de clustering, donde k es el número de grupos que se desea encontrar, y se corresponde con el número de neuronas de la capa oculta. Éste parámetro se debe decidir de antemano.

4.5 Ejemplos de entrenamiento

Mostraremos a continuación distintas configuraciones de la red neuronal para comparar los resultados obtenidos.

Para medir el error utilizaremos el estimador NMSE (Normalised Mean Square Error), que cuantifica la desviación total entre los valores predichos y medidos. Se define como:

$$NMSE = \frac{1}{N} \sum_i \frac{(P_i - M_i)^2}{\bar{P}\bar{M}} \quad (4.3)$$

Siendo P_i el valor predicho, M_i el valor medido y N el número total de puntos. Podemos calcular \bar{P} y \bar{M} con las expresiones (4.3) y (4.4):

$$\bar{P} = \frac{1}{N} \sum_i P_i \quad (4.4)$$

$$\bar{M} = \frac{1}{N} \sum_i M_i \quad (4.5)$$

En el NMSE las desviaciones (valores absolutos) se suman en lugar de las diferencias. Por esta razón, el NMSE generalmente muestra las diferencias más significativas entre los modelos. Si un modelo tiene una muy baja NMSE, entonces está bien modelado en el espacio y el tiempo. Por otra parte, una alta NMSE no significa necesariamente que el modelo es completamente erróneo.

Contrary to the bias, in the NMSE the deviations (absolute values) are summed instead of the differences. For this reason, the NMSE generally shows the most striking differences among models. If a model has a very low NMSE, then it is well performing both in space and time. On the other hand, high NMSE values do not necessarily mean that a model is completely wrong. That case could be due to time and/or space shifting. Moreover, it must be pointed out that differences on peaks have a higher weight on NMSE than differences on other values.

Para este caso hemos entrenado la red utilizando el método OLS. Los datos que hemos utilizado han sido los mismos que los del apartado 3.3 para el caso double ended:

Varianzas	Tiempo de entrenamiento (segundos)	Neuronas	Error entrenamiento	Error validación
$\sigma_1^2=10, \sigma_2^2=10$	0.204013	61	0.0070594689	0.0802911731
$\sigma_1^2=30, \sigma_2^2=30$	0.124008	51	0.0035712656	0.0922374787
$\sigma_1^2=20, \sigma_2^2=50$	0.276017	91	0.0016351404	0.0287548584
$\sigma_1^2=10, \sigma_2^2=80$	0.300019	91	0.0011100598	0.0340757036

Tabla 4.1 Resultados del entrenamiento para distintas configuraciones

A la vista de los resultados podemos decir que es mejor utilizar una varianza distinta para cada variable, ya que aunque el tiempo de entrenamiento y el número de neuronas necesarias sea mayor, los resultados obtenidos son mejores.

Veamos a continuación como las curvas generadas se aproximan a las reales:

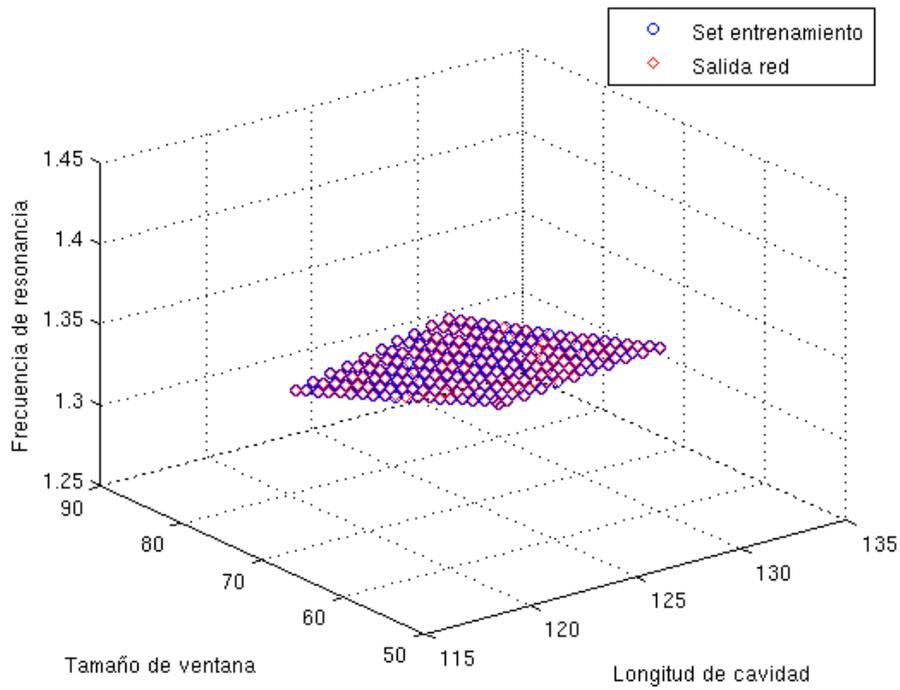


Ilustración 4.9 Evolución de la frecuencia de resonancia con las dimensiones

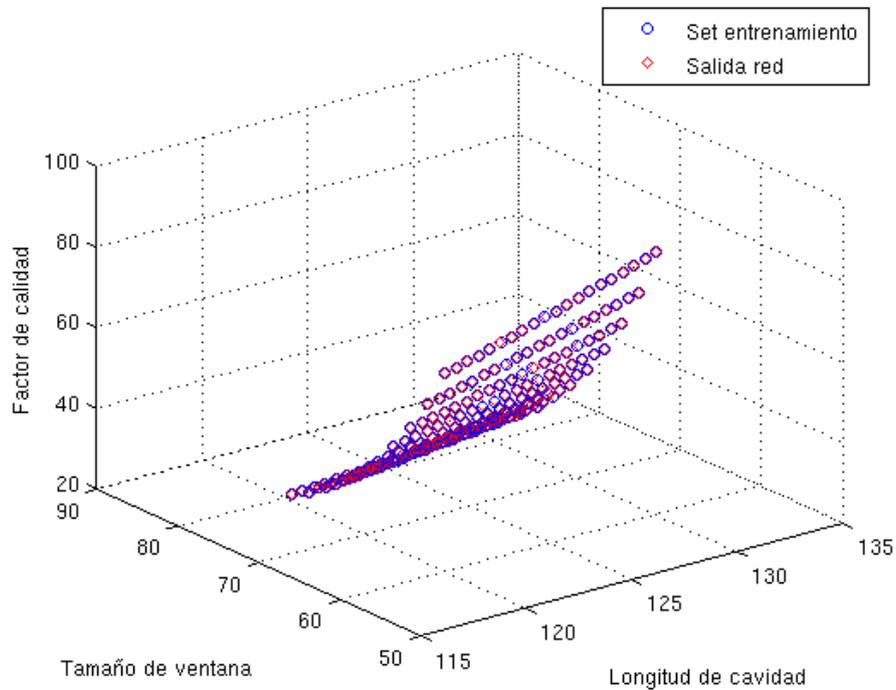


Ilustración 4.10 Evolución del factor de calidad con las dimensiones

Podemos ver en las gráficas 4.9 y 4.10 la evolución de la frecuencia de resonancia y el factor de calidad con las dimensiones. En ambas gráficas se representa el set de entrenamiento de la red y la salida de la red tras el entrenamiento, donde podemos ver que la red aproxima muy bien el set de entrenamiento.

A continuación podemos ver otros ejemplos de entrenamiento, en este caso de coeficientes de acoplo:

Varianzas	Tiempo de entrenamiento (segundos)	Neuronas	Error entrenamiento	Error validación
$\sigma_1^2=10, \sigma_2^2=10$	$5.6004 \cdot 10^{-2}$	11	0.0050998508	0.0051902661
$\sigma_1^2=30, \sigma_2^2=30$	$2.0001 \cdot 10^{-2}$	11	0.0052285851	0.0053978818
$\sigma_1^2=20, \sigma_2^2=60$	$4.8003 \cdot 10^{-2}$	11	0.0052637898	0.0054490250
$\sigma_1^2=10, \sigma_2^2=80$	$4.8003 \cdot 10^{-2}$	11	0.0051551552	0.0053114752

Tabla 4.2 Resultados del entrenamiento para distintas configuraciones

En el caso de los coeficientes de acoplo podemos observar que tanto los tiempos, como el número de neuronas y los errores son muy similares independientemente de los valores de configuración de la red.

A continuación representamos, tanto para la frecuencia de resonancia como el coeficiente de acoplo, los set de entrenamiento y la salida generada por la red.

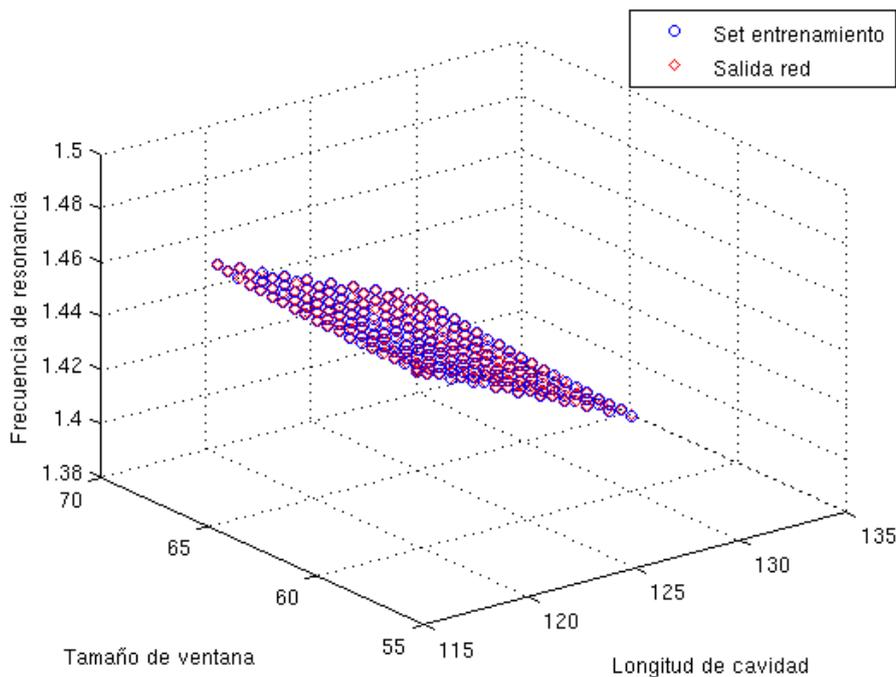


Ilustración 4.11 Evolución de la frecuencia de resonancia con las dimensiones

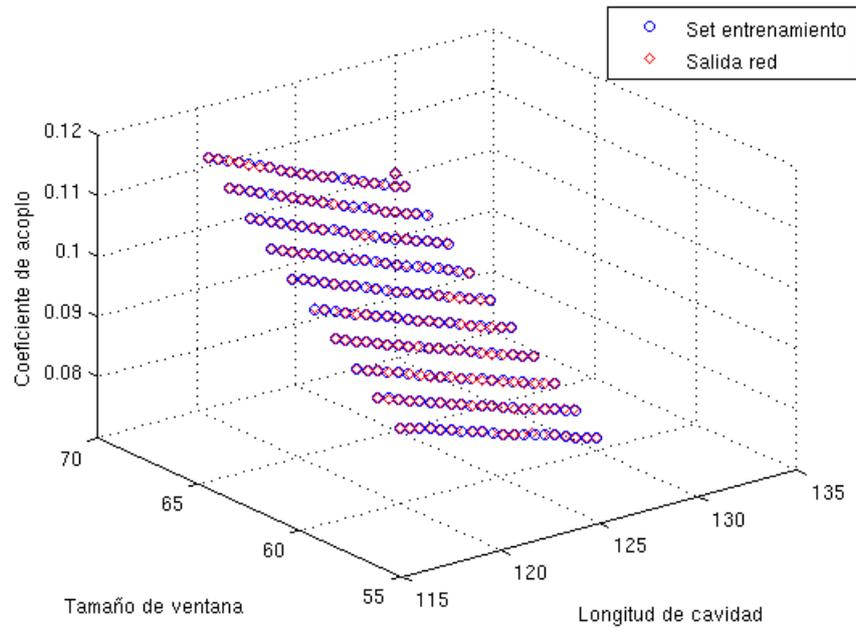


Ilustración 4.12 Evolución del coeficiente de acoplo con las dimensiones

En ambos casos la red aproxima muy bien los datos de entrenamiento de la red.

5. PROCESO DE DISEÑO

Como primer paso en este proceso de diseño debemos decidir el estándar de guía que utilizaremos, dependiendo de la banda de frecuencia del servicio que utilizemos. En el ANEXO I situado en la página 99 mostramos una tabla con los estándares de guía más utilizados. En nuestro caso, los filtros diseñados deberán estar centrados a una frecuencia de 1.4 GHz y por tanto utilizaremos el estándar WR-650 con dimensiones 165.1 x 82.55 mm.

5.1 Interfaz para la generación de datos

Para simplificar la comunicación con las funciones anteriormente explicadas hemos programado una interfaz gráfica con ayuda del GUI de MATLAB, que puede verse en la Ilustración 5.1:

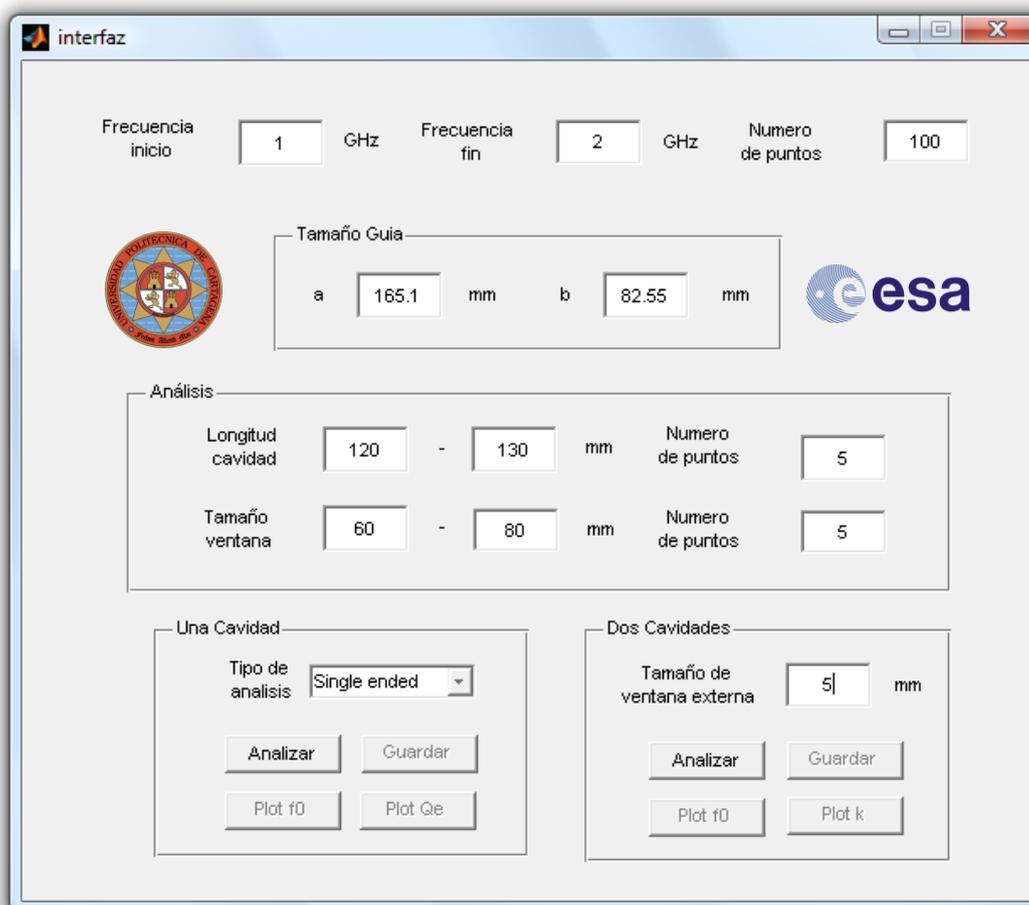


Ilustración 5.1 Interfaz utilizada para la generación de datos

Esta interfaz guardará como variables los datos introducidos en ella y llamará a la función correspondiente dependiendo del tipo de análisis que deseemos.

A continuación explicaremos cada uno de los paneles de forma más detenida:

1. **Margen frecuencias.** Se corresponde con los parámetros de configuración de FEST vistos en el capítulo correspondiente. Tendremos que proporcionar la frecuencia de inicio y fin que se utilizará para todos los análisis. Para facilitar el uso de la precisión introduciremos en la interfaz el número de puntos a simular entre ambos límites. Con estos valores se calculará el *Freq. Step* que deberemos proporcionar a FEST para el análisis.
2. **Tamaño guía.** Se trata de las dimensiones físicas de la guía con la que trabajaremos, donde *a* será la dimensión mayor y *b* la menor. Ambas vendrán expresadas en milímetros.
3. **Análisis.** En este panel observamos las características que tendrá el barrido que vamos a realizar en las dos dimensiones. En el caso de la figura tomará 5 puntos linealmente espaciados entre 120 y 130 mm. como longitudes de cavidad y 5 puntos linealmente espaciados entre 60 y 80 mm. para tamaños de ventana.
4. **Una cavidad.** Será la opción seleccionada si deseamos realizar un análisis de una cavidad. Podremos elegir en *Tipo de Análisis* si deseamos que sea *Single Ended* o *Double Ended*. El botón *Análisis* llamará a la función *calcula_se.m* o *calcula_de.m*, dependiendo de lo seleccionado en el *Pop-up menu*. Una vez finalizado el análisis, los datos podrán representarse gráficamente o guardarse en un archivo de texto, llamado *factores_calidad.out*, que tomará la red neuronal para realizar el proceso de entrenamiento.
5. **Dos cavidades.** Será la opción seleccionada si deseamos realizar un análisis de un filtro de orden $N=2$. En este caso podremos introducir el tamaño de la ventana externa. El botón *Análisis* llamará a la función *calcula.m*, para realizar el estudio y una vez finalizado los datos podrán representarse gráficamente o guardarse en un archivo de texto, llamado *coeficientes_acoplo.out*, utilizado para entrenar la red neuronal.

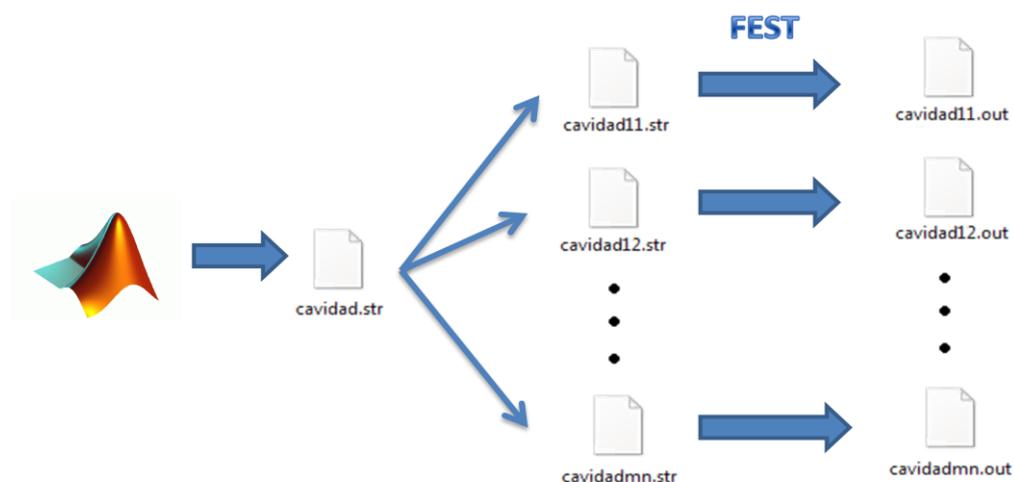


Ilustración 5.2 Estructura de generación de archivos

Dependiendo del tipo de análisis seleccionado la interfaz se comunicará con una de las funciones explicadas anteriormente, pudiendo ver en la Ilustración 5.2 un gráfico en el que se resume la creación de los archivos de configuración y la obtención de los archivos de resultados.

Una vez analizados los archivos .out, las funciones *calcula_se.m* y *calcula_de.m* nos devolverán dos matrices como las mostradas a continuación. Cada posición representa la frecuencia de resonancia y el factor de calidad de una cavidad con las dimensiones correspondientes a esos índices.

$$\begin{array}{cccccc}
 f_{0,11} & f_{0,12} & f_{0,13} & \dots & f_{0,1n} & Q_{e,11} & Q_{e,12} & Q_{e,13} & \dots & Q_{e,1n} \\
 f_{0,21} & f_{0,22} & f_{0,23} & \dots & f_{0,2n} & Q_{e,21} & Q_{e,22} & Q_{e,23} & \dots & Q_{e,2n} \\
 f_{0,31} & f_{0,32} & f_{0,33} & \dots & f_{0,3n} & Q_{e,31} & Q_{e,32} & Q_{e,33} & \dots & Q_{e,3n} \\
 \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
 f_{0,m1} & f_{0,m2} & f_{0,m3} & \dots & f_{0,mn} & Q_{e,m1} & Q_{e,m2} & Q_{e,m3} & \dots & Q_{e,mn}
 \end{array}$$

Los parámetros que nos devuelve la función *calcula.m* tendrán la siguiente forma, donde cada posición representa la frecuencia de resonancia y el acoplo entre-resonadores de un filtro de segundo orden con las dimensiones correspondientes a esos índices.

$$\begin{array}{cccccc}
 f_{0,11} & f_{0,12} & f_{0,13} & \dots & f_{0,1n} & k_{11} & k_{12} & k_{13} & \dots & k_{1n} \\
 f_{0,21} & f_{0,22} & f_{0,23} & \dots & f_{0,2n} & k_{21} & k_{22} & k_{23} & \dots & k_{2n} \\
 f_{0,31} & f_{0,32} & f_{0,33} & \dots & f_{0,3n} & k_{31} & k_{32} & k_{33} & \dots & k_{3n} \\
 \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
 f_{0,m1} & f_{0,m2} & f_{0,m3} & \dots & f_{0,mn} & k_{m1} & k_{m2} & k_{m3} & \dots & k_{mn}
 \end{array}$$

Con ayuda de los botones *Plot f0*, *Plot Qe* y *Plot k* podemos representar gráficamente los resultados obtenidos. En el caso de una cavidad podremos dibujar la evolución de la frecuencia de resonancia con el botón *Plot f0* y la del factor de calidad con el botón *Plot Qe*. En el caso de dos cavidades podremos dibujar la evolución de la frecuencia de resonancia con el botón *Plot f0* y la del coeficiente de acoplo con el botón *Plot k*. Hemos podido ver ejemplos de dichas gráficas en la sección 3.3.

En ambos casos dispondremos del botón *Guardar* para almacenar los resultados obtenidos en un archivo de texto que utilizaremos para entrenar la red neuronal.

5.2 Obtención del prototipo paso bajo del filtro

Como es habitual en el diseño de filtros de microondas, tendremos que calcular el prototipo paso bajo del filtro en primera instancia. A partir de este prototipo paso bajo obtendremos la matriz de acoplos y el factor de calidad necesarios para el diseño del filtro. Para realizar esta conversión utilizaremos las siguientes expresiones:

$$g_0 = 1.0 \quad (5.1)$$

$$g_1 = \frac{2}{\gamma} \sin\left(\frac{\pi}{2n}\right) \quad (5.2)$$

$$g_i = \frac{1}{g_{i-1}} \frac{4 \sin\left[\frac{(2i-1)\pi}{n}\right] \sin\left[\frac{(2i-3)\pi}{n}\right]}{\gamma^2 + \sin^2\left[\frac{(i-1)\pi}{n}\right]} \text{ para } i = 2, 3, \dots, n \quad (5.3)$$

$$g_{n+1} = \begin{cases} 1.0 & \text{para } n \text{ impar} \\ \coth^2\left(\frac{\beta}{4}\right) & \text{para } n \text{ par} \end{cases} \quad (5.4)$$

, donde β y γ son:

$$\beta = \ln\left[\coth\left(\frac{L_{Ar}}{17.37}\right)\right] \quad (5.5)$$

$$\gamma = \sinh\left(\frac{\beta}{2\pi}\right) \quad (5.6)$$

, y L_{Ar} es el rizado de la banda pasante.

Para calcular el orden del filtro utilizamos la siguiente expresión:

$$n \geq \frac{\cosh^{-1} \sqrt{\frac{10^{0.1L_{As}} - 1}{10^{0.1L_{Ar}} - 1}}}{\cosh^{-1} \Omega_s} \quad (5.7)$$

, siendo L_{As} la atenuación mínima en dB para la banda eliminada en $\Omega = \Omega_s$.

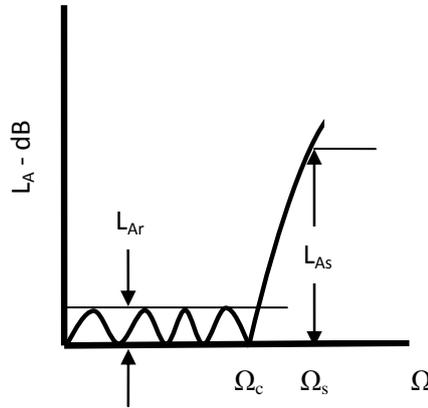


Ilustración 5.3 Prototipo paso bajo del filtro

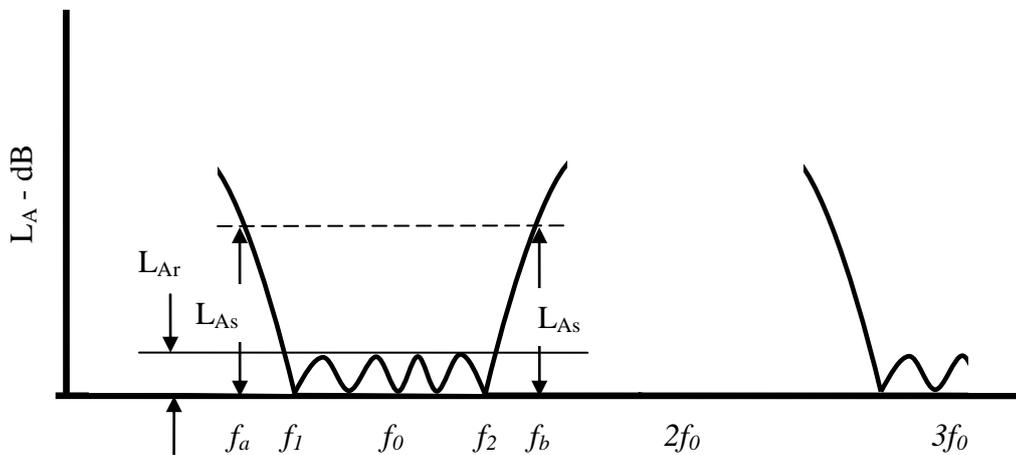


Ilustración 5.4 Respuesta filtro pasa-banda correspondiente

Para calcular Ω_s debemos realizar la conversión de f_a y f_b a Ω_a y Ω_b con la siguiente expresión:

$$\Omega = \frac{1}{FBW} \left(\frac{f}{f_0} - \frac{f_0}{f} \right) \quad (5.8)$$

Una vez calculados Ω_a y Ω_b podremos hallar Ω_s :

$$\Omega_s = \min(\Omega_a, \Omega_b) \quad (5.9)$$

5.3 Obtención de los parámetros a sintetizar

Con los coeficientes del filtro paso bajo equivalente y las ecuaciones generalizadas para el diseño de filtros paso banda a partir del prototipo paso bajo calculamos el factor de calidad y la matriz de acoplos del filtro. El factor de calidad lo podemos calcular aplicando la siguiente expresión:

$$Q_e = \frac{g_0 g_1 \Omega_c}{FBW} \quad (5.10)$$

En cuanto a los coeficientes de acoplo los calculamos utilizando:

$$k_{j,j+1} \Big|_{j=1}^{n-1} = \frac{FBW}{\Omega_c \sqrt{g_j g_{j+1}}} \quad (5.11)$$

, siendo Ω_c la frecuencia de corte ($\Omega_c=1$) y ω el ancho de banda relativo y cuya expresión mostramos a continuación:

$$FBW = \frac{f_2 - f_1}{f_0} \quad (5.12)$$

Para que nos resulte más cómodo trabajar con estos parámetros implementamos una función llamada *Chebyshev.m* que tiene como parámetros de entrada la frecuencia central, el ancho de banda, el rizado de la banda pasante y la atenuación de la banda eliminada. Esta función calculará el prototipo paso bajo del filtro y nos dará como salida el factor de calidad y la matriz de acoplos que necesitamos sintetizar.

5.4 Interfaz red neuronal

Para comunicarnos con la red neuronal hemos creado una interfaz que a partir de los datos generados realice el diseño del filtro. Dicha interfaz, tiene dos partes:

- **Especificaciones del filtro:** Debemos suministrar parámetros como frecuencia de resonancia, ancho de banda, rizado de la banda pasante y atenuación en la banda rechazada. Con estos datos se calculará el orden del filtro y obtendremos el prototipo paso bajo del filtro. A partir de dicho prototipo podremos calcular el factor de calidad y matriz de acoplos que necesitamos sintetizar con la red neuronal.

- **Parámetros de la Red Neuronal:** Aquí introduciremos el nivel de error de entrenamiento y validación, además de dos valores de varianza, uno para cada entrada. Como métodos de entrenamiento podremos escoger entre el método OLS y el K-means, ya explicados en el capítulo 4.

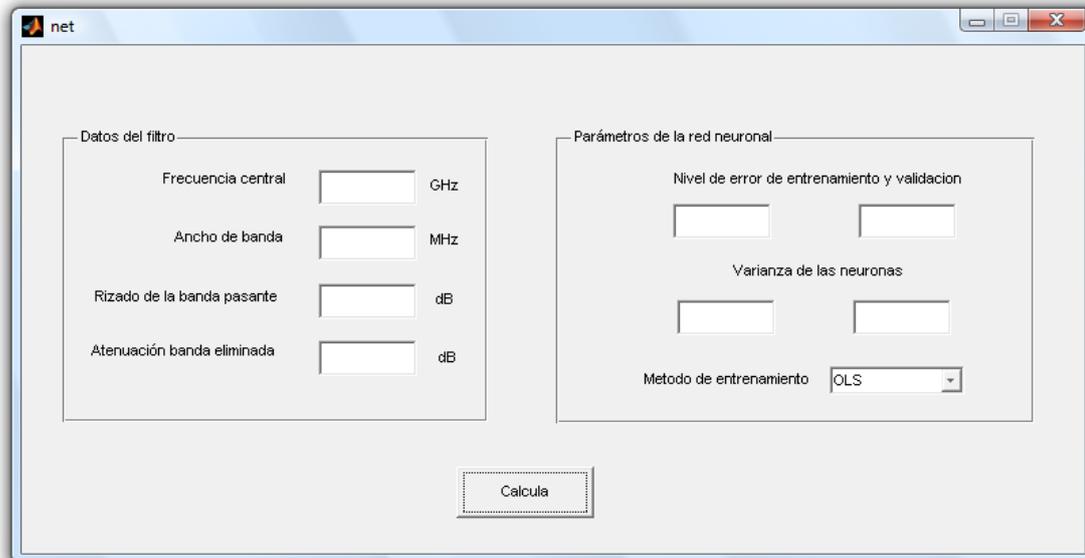


Ilustración 5.5 Interfaz utilizada para la red neuronal

Cuando se ejecuta la interfaz, ésta obtiene la matriz de acoplos y el factor de calidad a sintetizar a través de la función *Chebyshev.m*, que ha sido explicada anteriormente. Con estos datos y los parámetros de configuración de la red especificados, ejecutamos la función *disena.m*. Dicha función modifica un archivo llamado *Configuracion.in*, utilizado para comunicarnos con la red neuronal, donde se especifican los parámetros de configuración de la red: varianza de las neuronas, errores de entrenamiento y validación y método de entrenamiento. En este archivo también se especificarán los pares frecuencia de resonancia - factor de calidad y frecuencia de resonancia - acoplo que necesitamos calcular. Una vez modificado este archivo de configuración se ejecuta la red y se obtienen las dimensiones deseadas.

Es importante saber que estas dimensiones no pueden emplearse directamente ya que las ventanas influyen en la longitud de la cavidad y por tanto en la frecuencia a la que resuena dicha cavidad. Debemos hallar la longitud que cada ventana añade (a las que llamaremos $\Delta\omega_i$) y con ellas obtener las longitudes finales de los resonadores. La función *disena.m* está programada para calcular estas dimensiones finales automáticamente. A continuación explicaremos las expresiones utilizadas en dichos cálculos:

5.5 Obtención de las dimensiones finales

Como hemos dicho, necesitaremos obtener la longitud que añade cada ventana para obtener las dimensiones finales del filtro. Como hemos venido haciendo durante todo el trabajo explicaremos el proceso para cada uno de los tipos de análisis:

- **Análisis de una cavidad:**

Llamaremos L_1 a la longitud del resonador asociada al cálculo del factor de calidad. Como ya sabemos, tendremos dos métodos para obtener medidas en el caso de una cavidad y debido a las diferencias entre ambas estructuras la forma de calcular lo que añade la ventana será distinta, como veremos a continuación. En ambos casos necesitaremos la longitud L_g llamada longitud del resonador sin cargar, y que se explica en el apartado 5.6.

- Single ended

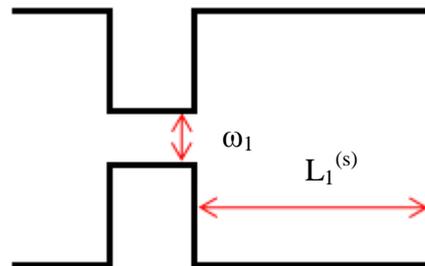


Ilustración 5.6 Cavidad terminada simplemente

$$L_1^{(s)} + \Delta\omega_1 = L_g \quad (5.13)$$

- Double ended

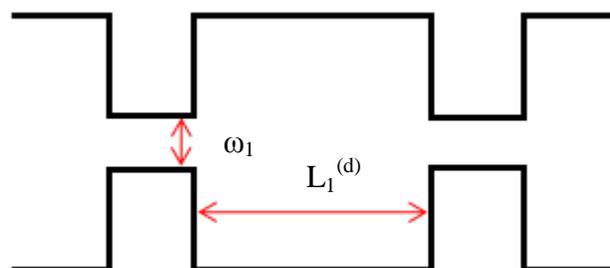


Ilustración 5.7 Cavidad terminada doblemente

$$L_1^{(d)} + 2\Delta\omega_1 = L_g \quad (5.14)$$

En ambos casos las ventanas hacen que la longitud efectiva de la cavidad sea mayor que L_1 y por tanto la frecuencia de resonancia disminuye. En el caso double ended, al tener dos ventanas, la distancia añadida será mayor. Por tanto, para que una cavidad con los mismos tamaños de ventana resuene a la misma frecuencia en ambos casos tendremos que $L_1^{(d)} > L_1^{(s)}$.

- **Análisis de dos cavidades**

A la hora de diseñar un filtro de orden N sólo necesitaremos sintetizar un factor de calidad. En cambio, tendremos que obtener N-1 coeficientes de acoplo. Para cada par frecuencia de resonancia-acoplo la red neuronal nos devolverá unas dimensiones físicas ($L_i - \omega_i$). De acuerdo con la notación empleada reservaremos $L_i - \omega_i$ a las dimensiones asociadas al cálculo del factor de calidad. Para el cálculo de los acoplos continuaremos desde $L_2 - \omega_2$ en adelante.

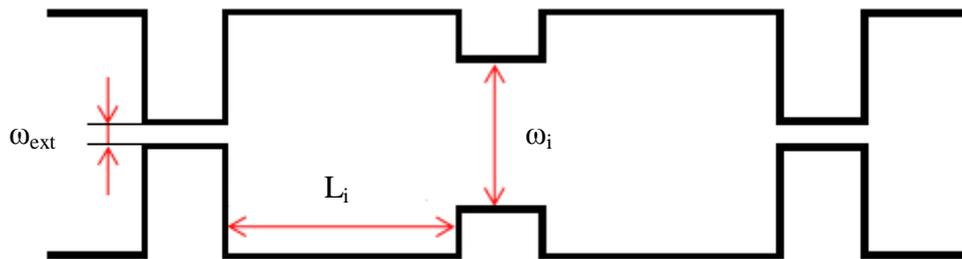


Ilustración 5.8 Filtro de orden 2

$$L_i + \Delta\omega_i + \Delta\omega_{ext} = L_g \quad (5.15)$$

Como ω_{ext} está muy cerrada podremos despreciar su efecto, tal y como se explica en el ANEXO II. La expresión anterior se reduce a:

$$L_i + \Delta\omega_i = L_g \quad (5.16)$$

Las dimensiones finales del filtro pueden calcularse con las fórmulas que siguen, dependiendo de si el orden del filtro es par o impar:

Si N es par:

$$d_i + \Delta\omega_i + \Delta\omega_{i+1} = L_g \quad \text{para } i = 1 \dots N/2 \quad (5.17)$$

Si N es impar:

$$d_i + \Delta\omega_i + \Delta\omega_{i+1} = L_g \text{ para } i = 1 \dots (N-1)/2 \quad (5.18)$$

$$d_i + 2\Delta\omega_i = L_g \text{ para } i = 1 \dots (N+1)/2$$

5.5.1 Ejemplo de filtro orden 3:

Realizaremos a continuación el proceso genérico para obtener un filtro de orden 3. Tendremos que obtener un factor de calidad y la siguiente matriz de acoplos:

$$Q_e \quad k = \begin{pmatrix} 0 & \kappa_1 & 0 \\ 0 & 0 & \kappa_1 \end{pmatrix}$$

Para cada par frecuencia de resonancia - factor de calidad o frecuencia de resonancia - coeficiente de acoplo la red nos proporcionará unas dimensiones físicas.

$$f_0 - Q_e \left\{ \begin{matrix} L_1 \\ \omega_1 \end{matrix} \right.$$

$$f_0 - \kappa_1 \left\{ \begin{matrix} L_2 \\ \omega_2 \end{matrix} \right.$$

A continuación mostramos la representación del filtro:

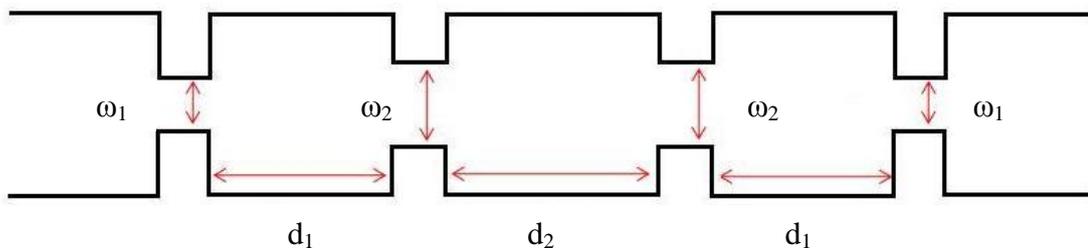


Ilustración 5.9 Filtro de orden 3 y dimensiones correspondientes

A partir de las expresiones (5.18) (caso impar) obtenemos:

$$d_1 + \Delta\omega_1 + \Delta\omega_2 = L_g \quad (5.19)$$

$$d_2 + 2\Delta\omega_2 = L_g$$

5.5.2 Ejemplo de filtro orden 4:

Realizaremos a continuación el proceso genérico para obtener un filtro de orden 4. Tendremos que obtener un factor de calidad y la siguiente matriz de acoplos:

$$Q_e \quad k = \begin{pmatrix} 0 & \kappa_1 & 0 & 0 \\ 0 & 0 & \kappa_2 & 0 \\ 0 & 0 & 0 & \kappa_1 \end{pmatrix}$$

Para cada par frecuencia de resonancia - factor de calidad o frecuencia de resonancia - coeficiente de acoplo la red nos proporcionará unas dimensiones físicas.

$$f_0 - Q_e \left\{ \begin{matrix} L_1 \\ \omega_1 \end{matrix} \right.$$

$$f_0 - \kappa_1 \left\{ \begin{matrix} L_2 \\ \omega_2 \end{matrix} \right.$$

$$f_0 - \kappa_2 \left\{ \begin{matrix} L_3 \\ \omega_3 \end{matrix} \right.$$

A partir de la fórmula (5.17) (caso par) obtenemos:

$$d_1 + \Delta\omega_1 + \Delta\omega_2 = L_g \tag{5.20}$$

$$d_2 + \Delta\omega_2 + \Delta\omega_3 = L_g$$

A continuación mostramos la representación del filtro con las dimensiones finales:

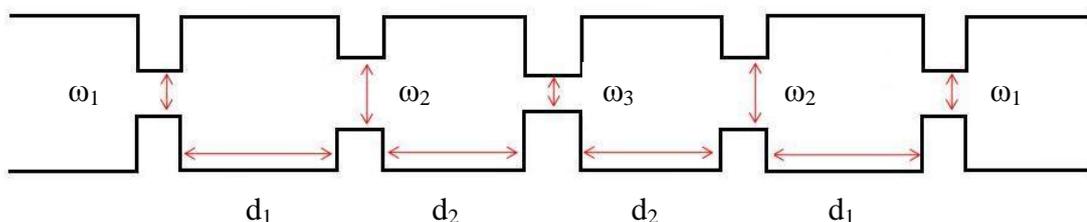


Ilustración 5.10 Filtro de orden 4 y dimensiones correspondientes

5.6 Obtención de la longitud del resonador sin cargar

Para conseguir las dimensiones finales necesitaremos emplear la longitud del resonador sin cargar (L_g). Ya que estamos trabajando con resonadores en $\lambda_g/2$ esta longitud será:

$$L_g = \lambda_g / 2 \quad (5.21)$$

, siendo λ_g la longitud de onda de la guía, que podremos calcular con la expresión (5.22).

$$\lambda_g = \frac{\lambda}{\sqrt{1 - \left(\frac{\lambda}{2a}\right)^2}} \text{ con } f > f_c \quad (5.22)$$

, donde λ es la longitud de onda a la que queremos que resuene la cavidad y a la dimensión mayor de la guía.

También puede verse como la frecuencia de resonancia del modo TE_{101}^z representada en la expresión (5.24). Esta expresión es una simplificación de la de la fórmula (5.23) en la que se expresa la frecuencia de resonancia para cualquier modo TE_{mnp}^z .

$$(f_r)_{mnp}^{TE} = \frac{1}{2\pi\sqrt{\mu\varepsilon}} \sqrt{\left(\frac{m\pi}{a}\right)^2 + \left(\frac{n\pi}{b}\right)^2 + \left(\frac{p\pi}{c}\right)^2} \quad \begin{array}{l} m = 0,1,2 \dots \\ n = 0,1,2 \dots \\ p = 0,1,2 \dots \end{array} \quad (5.23)$$

Si $c > a > b$, el modo de orden más bajo es el TE_{101}^z cuya frecuencia de resonancia es:

$$(f_r)_{101}^{TE} = \frac{1}{2\sqrt{\mu\varepsilon}} \sqrt{\left(\frac{1}{a}\right)^2 + \left(\frac{1}{c}\right)^2} \quad (5.24)$$

Por tanto este método también puede ser usado para obtener la longitud a la que resuena la cavidad, ya que conocemos la dimensión a de la guía, conocemos μ y ε ya que son los del aire y la frecuencia de resonancia será a la que queremos que resuene la cavidad. Tan sólo tendremos que despejar la longitud c que será nuestra L_g .

En la Ilustración 5.11 se muestra la estructura de la cavidad rectangular y las dimensiones mencionadas.

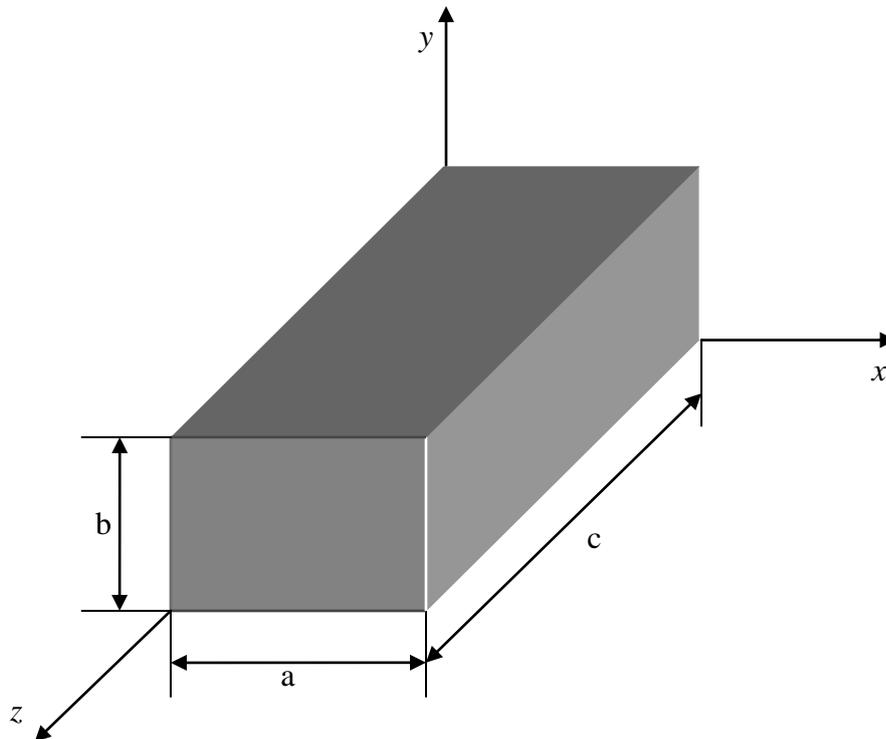


Ilustración 5.11 Geometría de una cavidad rectangular

5.7 Optimización automática

Como podremos ver en el capítulo siguiente, dedicado a ejemplos de diseño, a veces la respuesta del filtro obtenida se aleja de la deseada. Por ello, como último paso en el método de diseño propuesto necesitaremos optimizar las dimensiones que nos proporciona la red. Para la optimización hemos utilizado dos métodos:

- Optimización mediante gradiente: para ello hemos utilizado la función *fminimax*. Esta función reformula internamente en un equivalente _____ problema añadiendo restricciones adicionales de la forma: $F_i(x) \leq \gamma$ a las restricciones dadas en la ecuación, y entonces minimiza γ sobre x . Dicha función usa un método de programación cuadrática (SQP) para resolver este problema. Las modificaciones se hacen en la búsqueda en línea y en el Hessiano. En la búsqueda en línea se utiliza una función de mérito exacta y una propuesta por [3] y [5]. La búsqueda en línea termina cuando la función de mérito muestra mejora.

Debido a que el gradiente utilizado por defecto es 10^{-8} , muy pequeño para nuestro caso, tenemos que modificar las opciones para proporcionar un gradiente manual.

- Optimización mediante algoritmo genético: Son llamados así porque se inspiran en la evolución biológica, haciendo evolucionar una población de individuos sometiéndola a acciones aleatorias semejantes a las que actúan en la evolución biológica (mutaciones y recombinaciones genéticas), así como también a una selección de acuerdo con algún criterio, en función del cual se decide cuáles son los individuos más adaptados, que sobreviven, y cuáles los menos aptos, que son descartados.

Un algoritmo genético es un método de búsqueda dirigida basada en probabilidad. Se puede demostrar que el algoritmo converge en probabilidad al óptimo. En otras palabras, al aumentar el número de iteraciones, la probabilidad de tener el óptimo en la población tiende a 1 (uno).

Realizaremos una comparación para el ejemplo del apartado 6.2 mostrando los resultados obtenidos con distintos gradientes para el caso de la función *fminimax*:

Gradiente	Iteraciones	funcont	Error	Tiempo empleado
10^{-1}	36	83	0	573 seg
10^{-2}	26	57	0	392 seg.
10^{-3}	21	48	0	341 seg.

Tabla 5.1 Comparación de resultados para distintos gradientes

Con gradientes más pequeños a los mostrados en la tabla 5.1 el algoritmo no llega a una solución.

Para el mismo ejemplo del apartado 6.2 hemos realizado la optimización utilizando el algoritmo genético que nos proporciona la función *ga* de MATLAB. En este caso compararemos los datos obtenidos para distinto número de generaciones e individuos por generación.

Generaciones	Individuos	Error	Funccount	Tiempo empleado
10	25	119,3864	275	168 seg
20	25	20,6354	525	322 seg
20	50	9,0555	1050	653 seg
30	25	22,2888	775	484 seg
30	50	1,7037	1550	962 seg

Tabla 5.2 Comparación de resultados para distintas configuraciones del algoritmo genético

Como podemos ver el algoritmo de gradiente llega a una solución sin error en menor tiempo que el algoritmo genético. Por ello realizaremos la optimización de los filtros que sea necesarios utilizando el algoritmo de gradiente, utilizando un gradiente de 10^{-3} .

6. EJEMPLOS DE DISEÑO

6.1 Filtro pasa banda de orden 3 y 30 MHz de ancho de banda

Especificaciones del filtro

Frecuencia central: $f_0=1.4$ GHz

Rizado de la banda pasante: $L_{Ar} = 0.01$ dB

Ancho de Banda: $BW= 30$ MHz

Atenuación de la banda rechazada: $L_{AS} = -25$ dB

Para un filtro de las especificaciones mencionadas tendremos que sintetizar el factor de calidad y matriz de acoplos siguientes:

$$\kappa = \begin{bmatrix} 0 & 0.0274 & 0 \\ 0 & 0 & 0.0274 \end{bmatrix} \quad Q_e = 29.3626$$

Para cada uno de los valores de frecuencia de resonancia – factor de calidad y frecuencia de resonancia – coeficiente de acoplo la red nos proporciona las dimensiones físicas siguientes:

$$f_0 = 1.4 - Q_e = 29.3626 \begin{cases} L_1 = 111.9836 \text{ mm} \\ \omega_1 = 70.5323 \text{ mm} \end{cases}$$

$$f_0 = 1.4 - \kappa_1 = 0.0274 \begin{cases} L_2 = 136.5497 \text{ mm} \\ \omega_2 = 41.0949 \text{ mm} \end{cases}$$

Tras esto calculamos lo que “añade” cada resonador con las formulas para el caso impar (5.18) y las dimensiones finales de los resonadores. Podemos verlas en la siguiente tabla:

$\Delta\omega$
$\Delta\omega_1 = 14.4205 \text{ mm}$
$\Delta\omega_2 = 4.2749 \text{ mm}$

Dimensiones
$d_1 = 122.192 \text{ mm}$
$d_2 = 132.2748 \text{ mm}$

Podemos ver en la Ilustración 6.1 la respuesta del filtro obtenido:

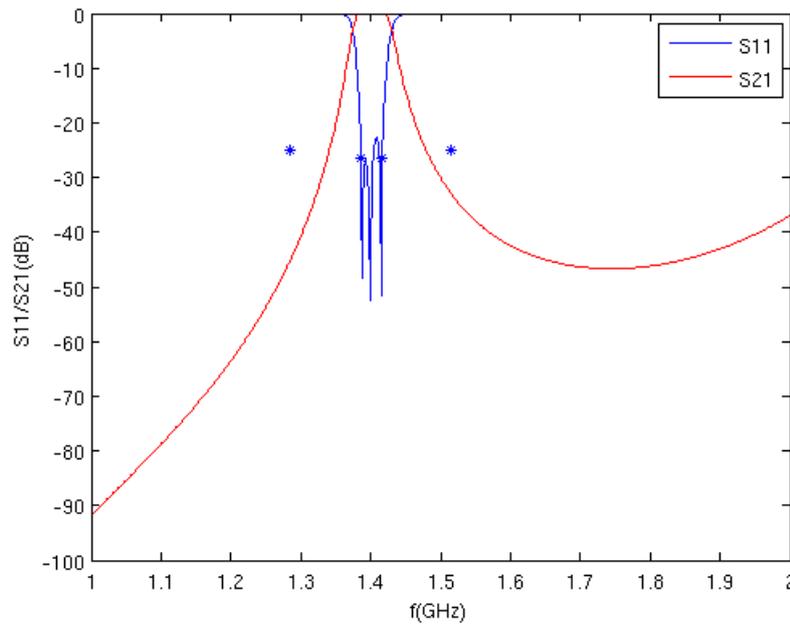


Ilustración 6.1 Respuesta del filtro obtenido

Aunque la respuesta del filtro es muy próxima a la deseada intentaremos optimizarla utilizando un algoritmo de gradiente (función *fminimax*). En la Tabla 6.1 mostramos las dimensiones obtenidas tras los cálculos anteriores y las obtenidas tras la optimización, así como los errores relativos cometidos.

Valor obtenido	Valor optimizado	Error relativo
$\omega_1 = 70.5323$ mm	$\omega_1 = 70.5855$ mm	0.0753 %
$\omega_2 = 41.0949$ mm	$\omega_2 = 40.8154$ mm	0.6848 %
$d_1 = 122.192$ mm	$d_1 = 122.2745$ mm	0.0675 %
$d_2 = 132.2748$ mm	$d_2 = 132.5173$ mm	0.1830 %

Tabla 6.1 Dimensiones iniciales, optimizadas y errores para el ejemplo 6.1

Podemos ver que los errores cometidos han sido muy pequeños. En las ilustraciones 6.2 y 6.3 podremos ver la respuesta del filtro tras el proceso de optimización:

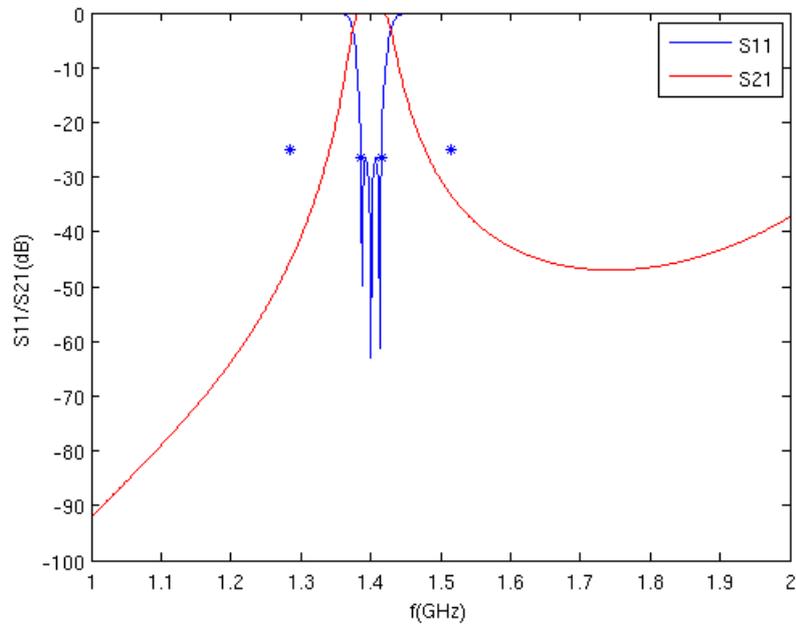


Ilustración 6.2 Respuesta optimizada del filtro

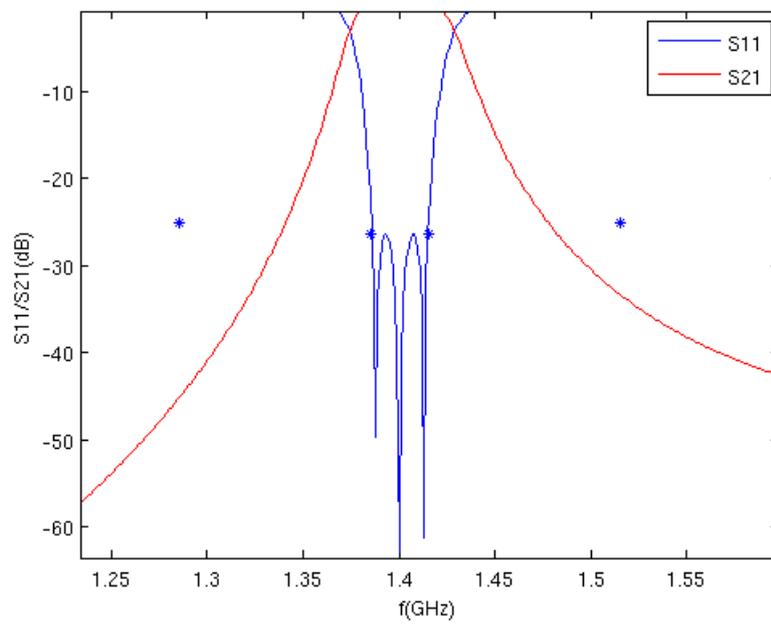


Ilustración 6.3 Respuesta optimizada del filtro en detalle

6.2 Filtro pasa banda de orden 4 y 80 MHz de ancho de banda

Especificaciones del filtro

Frecuencia central: $f_0=1.4$ GHz

Rizado de la Banda pasante: $L_{Ar} = 0.01$ dB

Ancho de Banda: $BW= 80$ MHz

Atenuación de la banda rechazada: $L_{AS} = -25$ dB

En primer lugar la interfaz llamará a la función *Chebyshev.m* y obtendrá el factor de calidad y la matriz de acoplos del filtro deseado:

$$\kappa = \begin{bmatrix} 0 & 0.0618 & 0 & 0 \\ 0 & 0 & 0.0454 & 0 \\ 0 & 0 & 0 & 0.0618 \end{bmatrix} \quad Q_e = 12.4754$$

Para cada par frecuencia de resonancia – factor de calidad y frecuencia de resonancia – coeficiente de acoplo, la red nos dará unas dimensiones físicas.

$$f_0 = 1.4 - Q_e = 12.4754 \begin{cases} L_1 = 101.9428 \text{ mm} \\ \omega_1 = 81.3954 \text{ mm} \end{cases}$$

$$f_0 = 1.4 - \kappa_1 = 0.0618 \begin{cases} L_2 = 133.4429 \text{ mm} \\ \omega_2 = 56.8599 \text{ mm} \end{cases}$$

$$f_0 = 1.4 - \kappa_2 = 0.0454 \begin{cases} L_3 = 134.3427 \text{ mm} \\ \omega_3 = 50.3362 \text{ mm} \end{cases}$$

Estas dimensiones no serán las que utilicemos directamente ya que necesitaremos hacer una serie de cálculos. Con ayuda de las fórmulas del caso par (5.15) obtendremos lo que descuenta cada ventana y con ellas las dimensiones finales representadas en la siguiente tabla:

Valor obtenido
$\Delta\omega_1 = 20.5726$ mm
$\Delta\omega_2 = 10.0983$ mm
$\Delta\omega_3 = 7.7957$ mm

Dimensiones
$d_1 = 113.0143$ mm
$d_2 = 125.9856$ mm

Una vez tenemos estas dimensiones, las introducimos en el programa FEST y obtenemos la siguiente respuesta en frecuencia:

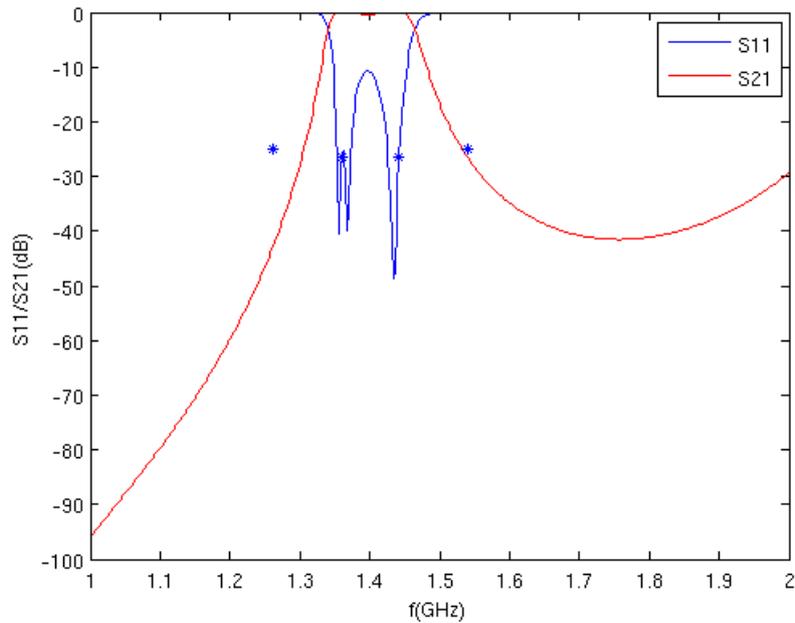


Ilustración 6.4 Respuesta del filtro obtenido

Para mejorar la respuesta del filtro realizamos un optimizado del filtro utilizando la función *fminimax* y un gradiente de 10^{-3} . Podemos ver en la siguiente tabla los valores obtenidos, los optimizados y el error relativo cometido.

Valor obtenido	Valor optimizado	Error relativo
$\omega_1 = 81.3954$ mm	$\omega_1 = 84.0879$ mm	3.2020 %
$\omega_2 = 56.8599$ mm	$\omega_2 = 55.6048$ mm	2.2573 %
$\omega_3 = 50.3362$ mm	$\omega_3 = 50.1155$ mm.	0.44038 %
$d_1 = 113.0143$ mm	$d_1 = 111.9174$ mm	0.9800 %
$d_2 = 125.9856$ mm	$d_2 = 126.0215$ mm.	0.02848 %

Tabla 6.2 Dimensiones iniciales, optimizadas y errores para el ejemplo 6.2

En la Ilustración 6.5 podemos ver la respuesta optimizada del filtro, en la que se puede apreciar que se cumplen las especificaciones. Además podemos ver los cuatro polos en la respuesta.

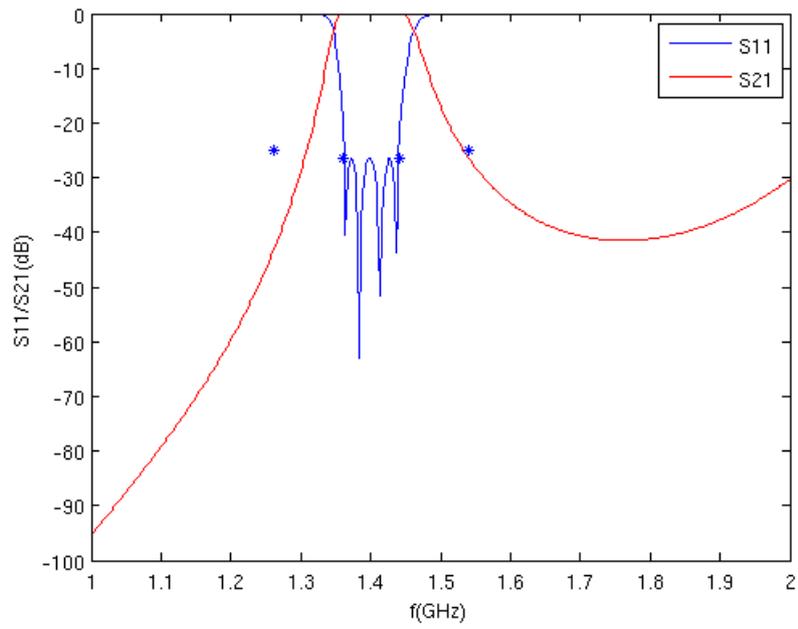


Ilustración 6.5 Respuesta optimizada del filtro

6.3 Filtro pasa banda de orden 4 y 100 MHz de ancho de banda

Especificaciones del filtro

Frecuencia central: $f_0=1.4$ GHz

Rizado de la Banda pasante: $L_{Ar} = 0.01$ dB

Ancho de Banda: $BW= 100$ MHz

Atenuación de la banda rechazada: $L_{AS} = -20$ dB

Con estas especificaciones, tendremos que sintetizar el siguiente factor de calidad y matriz de acoplos:

$$\kappa = \begin{bmatrix} 0 & 0.0772 & 0 & 0 \\ 0 & 0 & 0.0567 & 0 \\ 0 & 0 & 0 & 0.0772 \end{bmatrix} \quad Q_e = 9.9803$$

Para cada par frecuencia de resonancia-factor de calidad y frecuencia de resonancia-acoplo obtendremos unas dimensiones físicas de longitud de cavidad y tamaño de ventana:

$$f_0 = 1.4 - Q_e = 9.9803 \begin{cases} L_1 = 99.6548 \text{ mm} \\ \omega_1 = 83.9253 \text{ mm} \end{cases}$$

$$f_0 = 1.4 - \kappa_1 = 0.0772 \begin{cases} L_2 = 130.7017 \text{ mm} \\ \omega_2 = 62.1229 \text{ mm} \end{cases}$$

$$f_0 = 1.4 - \kappa_2 = 0.0567 \begin{cases} L_3 = 133.0043 \text{ mm} \\ \omega_3 = 54.9967 \text{ mm} \end{cases}$$

Con ayuda de las fórmula (5.17) obtendremos lo que descuenta cada ventana y con ellas las dimensiones finales representadas en la siguiente tabla:

Valor obtenido
$\Delta\omega_1 = 20.5726$ mm
$\Delta\omega_2 = 10.0983$ mm
$\Delta\omega_3 = 7.7957$ mm

Dimensiones
$d_1 = 110.129$ mm
$d_2 = 122.906$ mm

Con las dimensiones obtenidas obtenemos la siguiente respuesta:

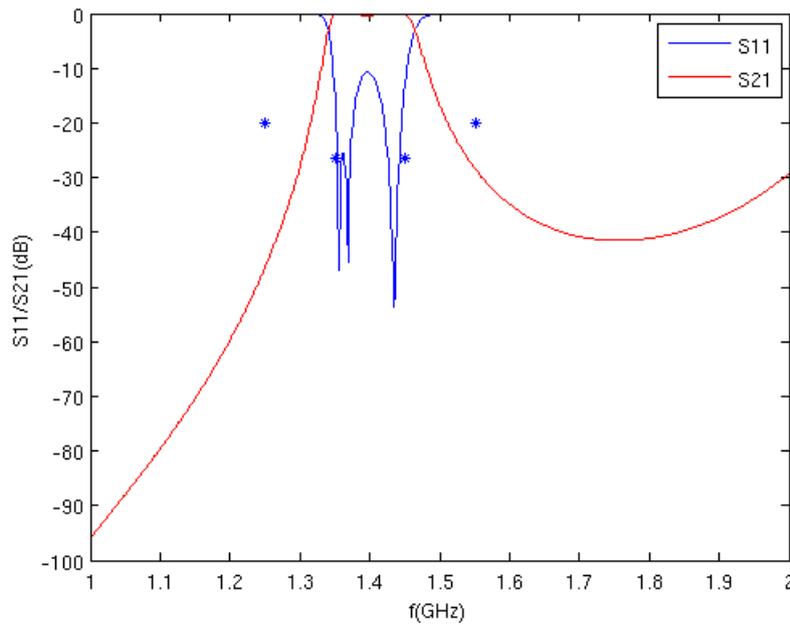


Ilustración 6.6 Respuesta obtenida del filtro

Para mejorar la respuesta del filtro realizamos un optimizado del filtro utilizando la función *fminimax* y un gradiente de 10^{-3} . Podemos ver en la siguiente tabla los valores obtenidos, los optimizados y el error relativo cometido.

Valor obtenido	Valor optimizado	Error relativo
$\omega_1 = 83.9253$ mm	$\omega_1 = 91.3396$ mm.	8.1173 %
$\omega_2 = 62.1229$ mm	$\omega_2 = 66.2711$ mm	6.2594 %
$\omega_3 = 50.3362$ mm	$\omega_3 = 55.2317$ mm	8.8635 %
$d_1 = 110.129$ mm	$d_1 = 105.1682$ mm	4.7223 %
$d_2 = 122.906$ mm	$d_2 = 119.4877$ mm.	2.8608 %

Tabla 6.3 Dimensiones iniciales, optimizadas y errores para el ejemplo 6.3

A continuación podemos ver la respuesta del filtro las tras el proceso de optimización:

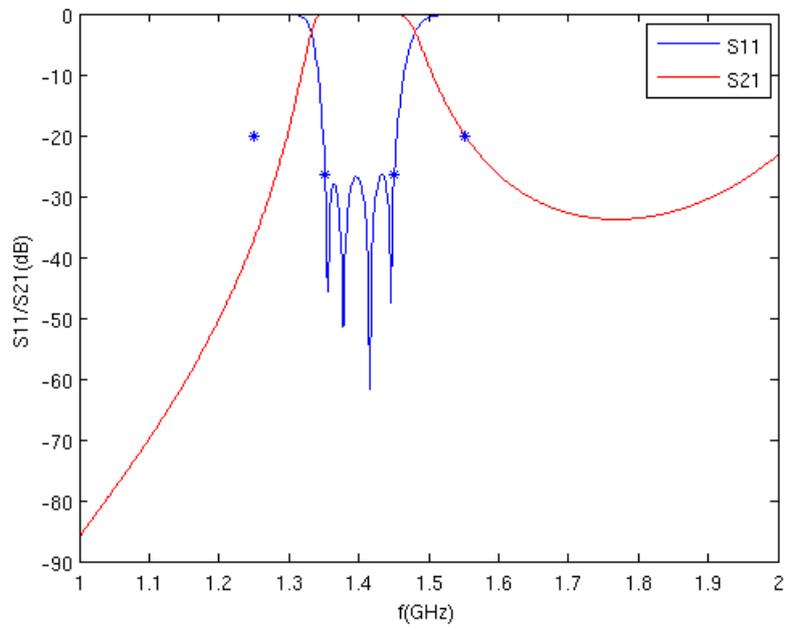


Ilustración 6.7 Respuesta optimizada del filtro

Podemos ver que se cumplen las especificaciones de diseño.

6.4 Filtro pasa banda de orden 5 y 30 MHz de ancho de banda

Especificaciones del filtro:

Frecuencia central: $f_0=1.4$ GHz Rizado de la Banda pasante: $L_{Ar} = 0.01$ dB
 Ancho de Banda: $BW= 30$ MHz Atenuación de la banda rechazada: $L_{AS} = -65$ dB

Con estas especificaciones, tendremos que sintetizar el siguiente factor de calidad y matriz de acoplos:

$$\kappa = \begin{bmatrix} 0 & 0.0216 & 0 & 0 & 0 \\ 0 & 0 & 0.0149 & 0 & 0 \\ 0 & 0 & 0 & 0.0149 & 0 \\ 0 & 0 & 0 & 0 & 0.0216 \end{bmatrix} \quad Q_e = 35.2960$$

Para cada par frecuencia de resonancia – factor de calidad y frecuencia de resonancia – acoplo la red nos devuelve unas dimensiones físicas.

$$f_0 = 1.4 - Q_e = 35.2960 \begin{cases} L_1 = 114.0214 \text{ mm} \\ \omega_1 = 68.2351 \text{ mm} \end{cases}$$

$$f_0 = 1.4 - \kappa_1 = 0.0216 \begin{cases} L_2 = 137.3100 \text{ mm} \\ \omega_2 = 37.7833 \text{ mm} \end{cases}$$

$$f_0 = 1.4 - \kappa_2 = 0.0149 \begin{cases} L_3 = 138.1933 \text{ mm} \\ \omega_3 = 32.7510 \text{ mm} \end{cases}$$

Con ayuda de las fórmulas para el caso impar (5.18) obtendremos lo que descuenta cada ventana y con esos valores seremos capaces de obtener las dimensiones finales, mostradas a continuación:

$\Delta\omega$
$\Delta\omega_1 = 13.4016 \text{ mm}$
$\Delta\omega_2 = 2.9746 \text{ mm}$
$\Delta\omega_3 = 2.0913 \text{ mm}$

Valores obtenidos
$d_1 = 123.9084 \text{ mm}$
$d_2 = 135.2187 \text{ mm}$
$d_3 = 136.1020 \text{ mm}$

La respuesta obtenida con esas dimensiones será la siguiente:

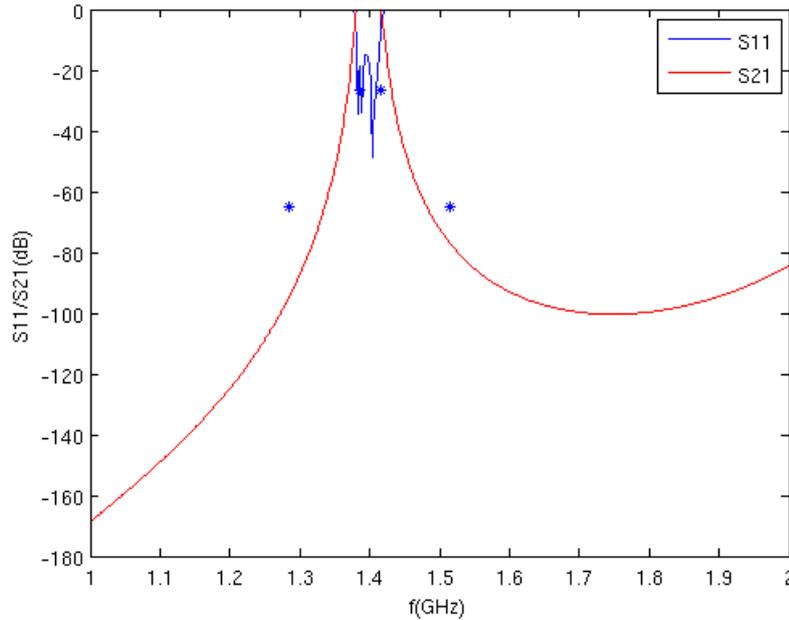


Ilustración 6.8 Respuesta obtenida del filtro

A continuación mostramos una tabla con las dimensiones obtenidas y los valores tras optimizar. Para el proceso de optimización utilizaremos la función *fminimax* con un gradiente igual a 10^{-4} , ya que al tratarse de un filtro de orden 5 la respuesta es más sensible al cambio de dimensiones. Así mismo mostraremos los errores relativos cometidos.

Valor obtenido	Valor optimizado	Error relativo
$\omega_1 = 68.2351$ mm	$\omega_1 = 68.7612$ mm	0.765 %
$\omega_2 = 37.7833$ mm	$\omega_2 = 37.7375$ mm	0.1213 %
$\omega_3 = 32.7510$ mm	$\omega_3 = 32.7493$ mm	0.0052 %
$d_1 = 123.9084$ mm	$d_1 = 123.6678$ mm	0.1945 %
$d_2 = 135.2187$ mm	$d_2 = 134.7727$ mm	0.3309 %
$d_3 = 136.1020$ mm	$d_3 = 135.6976$ mm	0.2980 %

Tabla 6.4 Dimensiones iniciales, optimizadas y errores para el ejemplo 6.4

Mostraremos la respuesta tras optimizar a continuación:

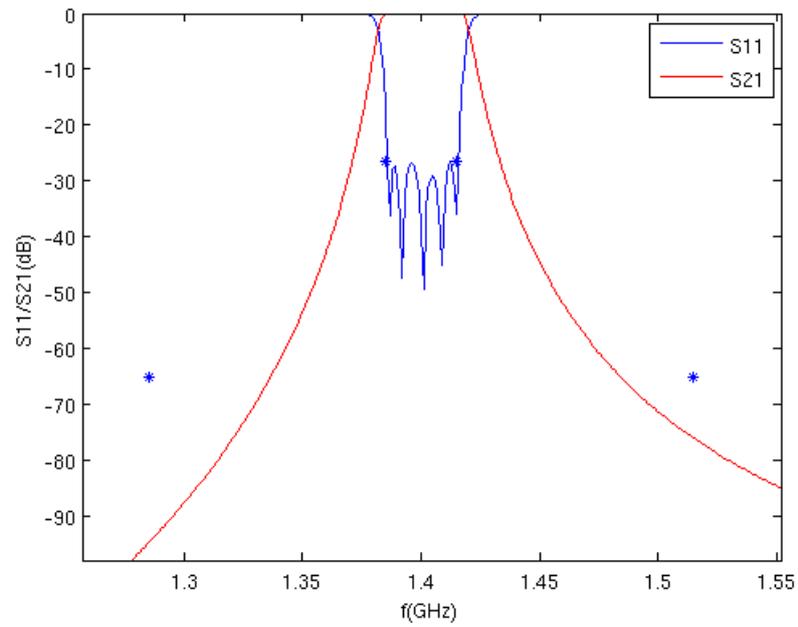


Ilustración 6.9 Respuesta del filtro optimizado en detalle

Podemos ver como tras el optimizado se cumplen las especificaciones de diseño del filtro.

6.5 Filtro pasa banda de orden 5 y 75 MHz de ancho de banda

Especificaciones del filtro:

Frecuencia central: $f_0=1.4$ GHz Rizado de la Banda pasante: $L_{Ar} = 0.01$ dB
 Ancho de Banda: $BW= 75$ MHz Atenuación de la banda rechazada: $L_{AS} = -40$ dB

$$\kappa = \begin{bmatrix} 0 & 0.0539 & 0 & 0 & 0 \\ 0 & 0 & 0.0373 & 0 & 0 \\ 0 & 0 & 0 & 0.0373 & 0 \\ 0 & 0 & 0 & 0 & 0.0539 \end{bmatrix} \quad Q_e = 14.1184$$

Para cada par frecuencia de resonancia – factor de calidad y frecuencia de resonancia – acoplo la red nos devuelve unas dimensiones físicas.

$$f_0 = 1.4 - Q_e = 14.1184 \begin{cases} L_1 = 103.3249 \text{ mm} \\ \omega_1 = 79.8753 \text{ mm} \end{cases}$$

$$f_0 = 1.4 - \kappa_1 = 0.0539 \begin{cases} L_2 = 133.3253 \text{ mm} \\ \omega_2 = 54.0017 \text{ mm} \end{cases}$$

$$f_0 = 1.4 - \kappa_2 = 0.0373 \begin{cases} L_3 = 135.3107 \text{ mm} \\ \omega_3 = 46.6512 \text{ mm} \end{cases}$$

Con ayuda de las fórmulas (5.17) obtendremos lo que descuenta cada ventana y con ellas las dimensiones finales representadas en la siguiente tabla:

$\Delta\omega$
$\Delta\omega_1 = 18.74985 \text{ mm}$
$\Delta\omega_2 = 7.4993 \text{ mm}$
$\Delta\omega_3 = 5.5139 \text{ mm}$

Valores obtenidos
$d_1 = 114.57545 \text{ mm}$
$d_2 = 127.8114 \text{ mm}$
$d_3 = 129.7968 \text{ mm}$

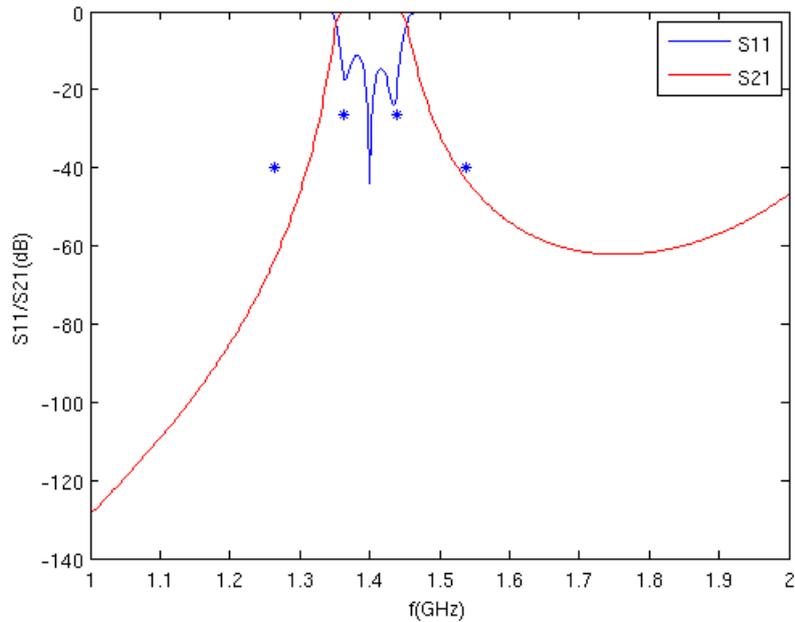


Ilustración 6.10 Respuesta del filtro obtenido

Como podemos ver en la Ilustración 6.10 la respuesta del filtro no se termina de ajustar a las especificaciones. Por ello optimizamos el filtro utilizando el algoritmo *fminimax* utilizando un gradiente de 10^{-3} . A continuación mostramos las dimensiones iniciales y las obtenidas tras el proceso de optimizado.

Valor obtenido	Valor optimizado	Error relativo
$\omega_1 = 78.8753$ mm	$\omega_1 = 81.2945$ mm	2.9758 %
$\omega_2 = 54.0017$ mm	$\omega_2 = 53.2113$ mm	1.4854%
$\omega_3 = 46.6512$ mm	$\omega_3 = 46.7166$ mm	0.1400 %
$d_1 = 114.57545$ mm	$d_1 = 113.5731$ mm	0.8825 %
$d_2 = 127.8114$ mm	$d_2 = 127.8475$ mm	0.0282 %
$d_3 = 129.7968$ mm	$d_3 = 129.8727$ mm	0.0584%

Tabla 6.5 Dimensiones iniciales, optimizadas y errores para el ejemplo 6.5

Los errores relativos obtenidos son muy pequeños y podemos ver la respuesta final del filtro a continuación, en la que se cumplen las especificaciones de diseño del filtro.

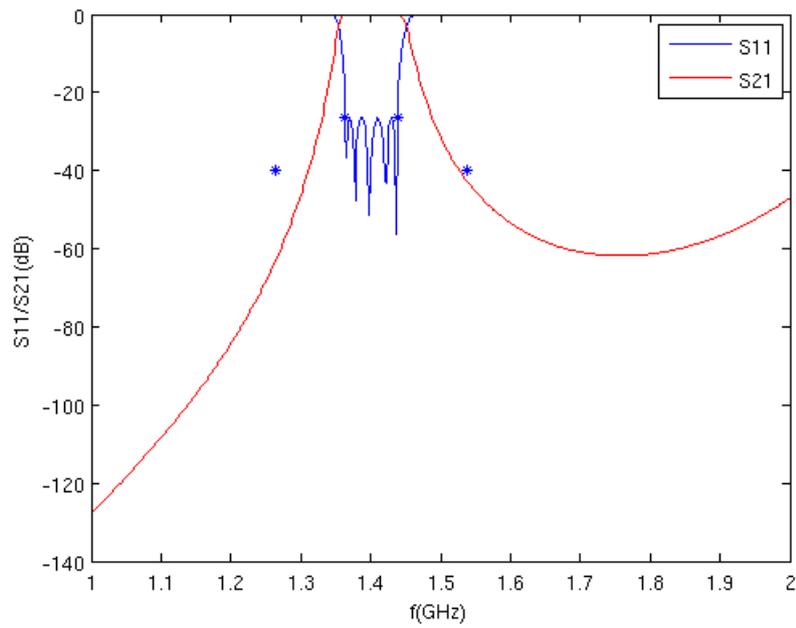


Ilustración 6.11 Respuesta del filtro optimizado

7. CONCLUSIONES Y LÍNEAS FUTURAS

7.1 Conclusiones

Como hemos visto a lo largo de este trabajo hemos presentado una herramienta que permite el diseño asistido de filtros en guía de onda de una forma muy rápida. Dicha herramienta se basa en relacionar medias de parámetros eléctricos con dimensiones físicas. Estas medidas se han realizado utilizando el programa FEST de la Agencia Espacial Europea.

Para proporcionar más precisión a la herramienta hemos propuesto el uso de redes neuronales. Dichas redes van a aproximar muy bien las funciones de datos generados. De hecho, los diseños obtenidos se acercan mucho a los deseados pero realizaremos una optimización mediante algoritmos genéticos y algoritmos de gradiente para obtener las características deseadas. Esta optimización es necesaria debido a que FEST nos proporciona una precisión de 3 dígitos, lo que a veces puede resultar escaso.

Por último, como ejemplos hemos realizado el diseño de varios filtros en la banda de 1400 MHz, mostrando la efectividad de dicha herramienta.

7.2. Líneas Futuras

En cuanto a la guía de onda utilizada podríamos plantear las siguientes líneas:

- Variar la anchura de la cavidad respecto a la de la guía de entrada/salida para situarnos en un punto de la carta de modos más adecuado, de forma que los espurios no asimetricen la respuesta como ocurre en la mayoría de los casos.
- Utilizar una guía con postes dieléctricos, ya que estos filtros tienen menor masa y volumen y una mayor estabilidad térmica en aplicaciones de alta potencia. Para este caso no podríamos utilizar FEST ya que no permite el uso de estos postes, pero sí HFSS (High Frequency Structure Simulator).

En cuanto a las técnicas de diseño:

- Podríamos utilizar FEST 3D (actualmente en desarrollo) en puesto del programa FEST, el cual obtiene una mayor precisión a costa de un mayor tiempo de cómputo.
- Aplicar la técnica de diseño Space Mapping con el fin de diseñar filtros con resonadores dieléctricos (3D), a partir de la respuesta de filtros con postes dieléctricos inductivos (2D).

En cuanto a la red neuronal se podrían plantear distintos métodos de entrenamiento.

ANEXO I

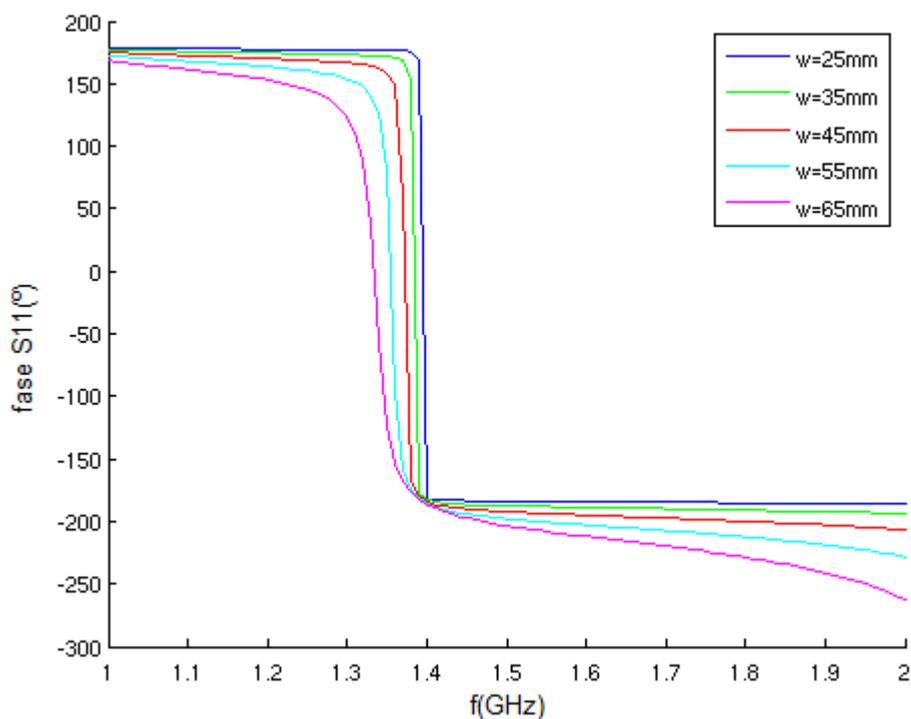
Bandas de frecuencia de la guía y límites interiores			
Banda de frec.	Standard	Límites en frecuencia (GHz)	Dimensiones (milímetros)
	WR-2300	0.32 - 0.49	584.2 x 292.1
	WR-2100	0.35 - 0.53	533.4 x 266.7
	WR-1800	0.43 - 0.62	457.2 x 228.6
	WR-1500	0.49 - 0.74	381 x 190.5
	WR-1150	0.64 - 0.96	292.1 x 146.05
	WR-1000	0.75 - 1.1	253.365 x 123.825
	WR-770	0.96 - 1.5	195.58 x 85.979
	WR-650	1.12 - 1.70	165.1 x 82.55
R band	WR-430	1.70 - 2.60	109.22 x 54.61
D band	WR-340	2.20 - 3.30	86.36 x 43.18
S band	WR-284	2.60 - 3.95	72.136 x 34.036
E band	WR-229	3.30 - 4.90	58.166 x 29.21
G band	WR-187	3.95 - 5.85	47.5488 x 22.1488
F band	WR-159	4.90 - 7.05	40.386 x 19.2786
C band	WR-137	5.85 - 8.20	34.8488 x 15.7988
H band	WR-112	7.05 - 10.00	28.4988 x 12.6238
X band	WR-90	8.2 - 12.4	22.86 x 10.16
Ku band	WR-62	12.4 - 18.0	15.7988 x 7.8994
K band	WR-51	15.0 - 22.0	12.954 x 6.477
K band	WR-42	18.0 - 26.5	10.668 x 4.318
Ka band	WR-28	26.5 - 40.0	7.112 x 3.556
Q band	WR-22	33 - 50	5.6896 x 2.8448
U band	WR-19	40 - 60	4.7752 x 2.3876
V band	WR-15	50 - 75	3.7592 x 1.8796
E band	WR-12	60 - 90	3.0988 x 1.5494
W band	WR-10	75 - 110	2.54 x 1.27
F band	WR-8	90 - 140	2.032 x 1.016
D band	WR-6	110 - 170	1.651 x 0.8255
G band	WR-5	140 - 220	1.2954 x 0.6477
	WR-4	170 - 260	1.0922 x 0.5461
	WR-3	220 - 325	0.8636 x 0.4318
Y-band	WR-2	325 - 500	0.508 x 0.254
	WR-1.5	500 - 750	0.381 x 0.1905
	WR-1	750 - 1100	0.254 x 0.127

ANEXO II

Variación de la frecuencia de resonancia con el tamaño de ventana

Como ya pudimos ver en los ejemplos mostrados en el capítulo 3.3, la frecuencia de resonancia disminuía conforme aumentábamos el tamaño de la ventana. Para comprobar que en el caso de la obtención de los acoplos la ventana externa no influía en la frecuencia a la que resonaba la cavidad decidimos realizar el siguiente ejemplo:

Utilizamos como longitud de cavidad la longitud del resonador sin cargar para una guía del estándar WR-650 (165.1 x 82.55 mm.) que resuena a 1.4 GHz. Esta longitud es $L_g=140.8246$ mm. Representaremos gráficamente la evolución de la fase en torno a la frecuencia de resonancia para distintos tamaños de la ventana.



Puede observarse que en el caso de la ventana de 25 mm. la fase tiene una variación abrupta en 1.4 GHz. Conforme vamos aumentando el tamaño de la ventana, la frecuencia de resonancia disminuye. Así mismo la variación en torno a la frecuencia de resonancia va haciéndose menos abrupta.

Para el caso del double ended realizamos otra prueba de este estilo. Utilizando unas ventanas de 5 mm. realizamos un barrido de longitudes de cavidad entre 135 y 145 mm. y tras analizar buscamos la longitud de cavidad que dé como resultado una frecuencia de resonancia de 1.4 GHz. Obtenemos un valor de 140.7895 mm, y aplicando la expresión (5.14) logramos el valor que añade la ventana:

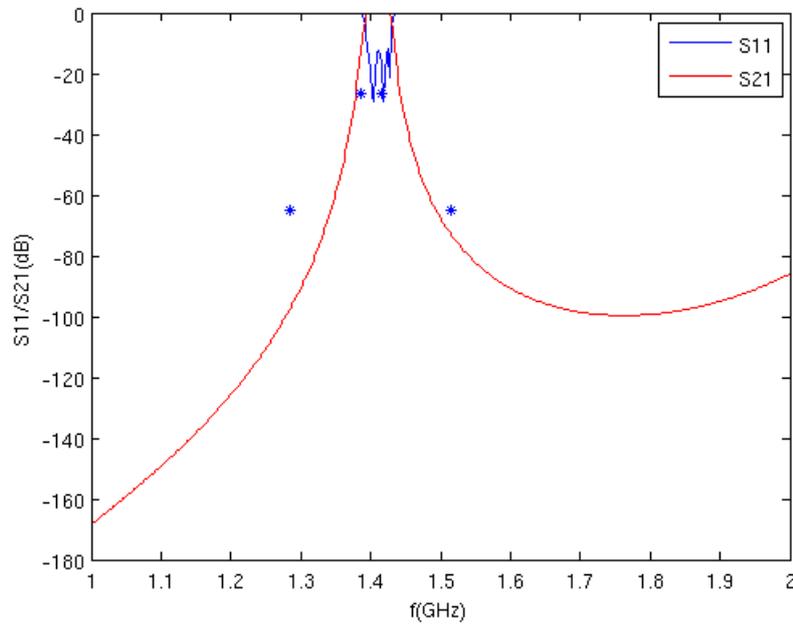
$$\Delta\omega_{\text{ext}} = 0.017563 \text{ mm.}$$

Este valor es muy pequeño, y por tanto ha sido despreciado en los cálculos realizados.

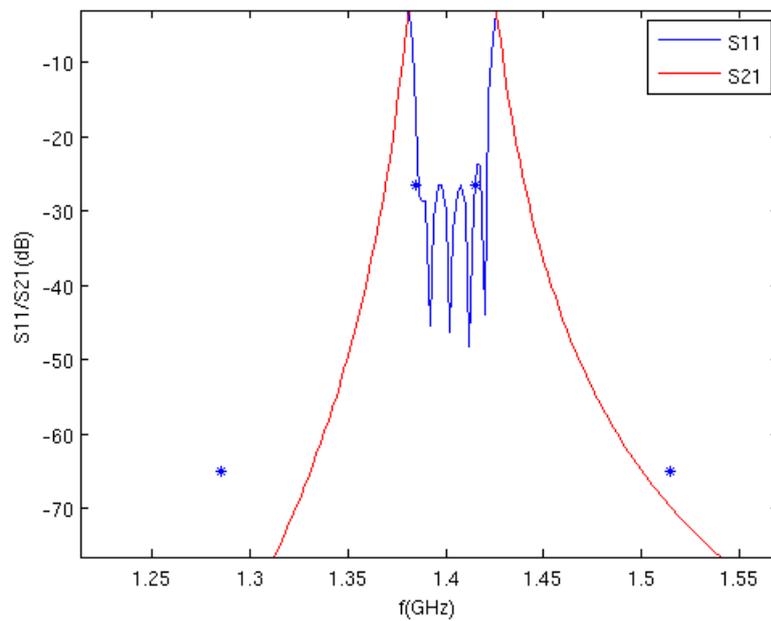
ANEXO III

Importancia de la red neuronal

Debido a la efectividad de la optimización automática podríamos pensar que la red neuronal es un paso innecesario. Por ellos realizaremos la optimización de un filtro con valores incorrectos, descentrado de la frecuencia central. Podemos ver su respuesta a continuación:



Tras el proceso de optimización obtenemos la siguiente respuesta:



Podemos ver como no se terminan de cumplir las especificaciones y no se observan todos los polos del filtro. Además la respuesta no termina de estar centrada en torno a 1.4 GHz y el tiempo en optimizar es muy superior al resto de casos.

Por tanto, podemos concluir que la red neuronal es un paso clave, ya que la obtención de un punto base suficientemente bueno es imprescindible para que la optimización funcione bien y obtengamos la respuesta deseada del filtro.

BIBLIOGRAFÍA

[1] Jia-Sheng Hong, M. J. Lancaster. Microstrip Filters for RF/Microwave Applications. 2001. John Wiley & Sons, Inc.

[2] G.L. Matthaei L. Young E.M.T. Jones, “Microwave Filters Impedance-matching Networks and Coupling Structures”, Artech House.

[3] Balanis, C.A. (1989). “Advanced Engineering Electromagnetics”. John Wiley and Sons.

[4] Gallardo Antolín, Asunción. Pascual García, Juan. Sancho Gómez, José Luis. Radial basis function networks and their application on communication systems.

[5] Juan Pascual García, et. Al. “A Neural Network Method for the Analysis of Multilayered Shielded Microwave Circuits”, IEEE Transactions on Microwave Theory and Techniques, Vol. 54, No.1, January 2006, pp. 309-320 (ISSN 0018-9480).