



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA MECÁNICA

DISEÑO Y OPTIMIZACIÓN DE ESTRUCTURAS RETICULARES EN PANELES CELULARES PARA LA REDUCCIÓN DE VIBRACIONES

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL MECÁNICO

PABLO IGNACIO BETANZO CISTERNA

PROFESORA GUÍA:

Viviana Meruane Naranjo

MIEMBROS DE LA COMISIÓN:

Alejandro Ortiz Bernardin

Rubén Fernández Urrutia

Este trabajo ha sido parcialmente financiado por:
FONDECYT 1210442, Optimal Design of Ultralight Sandwich
Panels with Cellular Truss Cores and Large Phononic Band Gaps.

SANTIAGO DE CHILE

2024

RESUMEN DE: MEMORIA PARA OPTAR
AL TÍTULO DE INGENIERO CIVIL MECÁNICO
POR: PABLO IGNACIO BETANZO CISTERNA
FECHA: 2024
PROF. GUÍA: Viviana Meruane N.

DISEÑO Y OPTIMIZACIÓN DE ESTRUCTURAS RETICULARES EN PANELES CELULARES PARA LA REDUCCIÓN DE VIBRACIONES

A lo largo de los años, se ha investigado y profundizado en cómo mejorar las propiedades mecánicas de los materiales. En este contexto, ha surgido el concepto de metamateriales, que son materiales artificiales con propiedades inusuales dependientes de su estructura. Estos materiales han sido una solución en áreas como la aeronáutica, electromagnética, óptica y visual.

Entre las propiedades que se pueden lograr con los metamateriales se encuentra la capacidad de suprimir vibraciones mecánicas en ciertas bandas de frecuencia. Esta propiedad ha sido objeto de investigación durante mucho tiempo, ya que la propagación de vibraciones en un cuerpo puede generar problemas que afectan la estabilidad de una estructura, como en el caso de la construcción de puentes, donde la estabilidad depende directamente del tipo de estructura utilizada. La posibilidad de construir materiales con una buena relación entre rigidez y densidad, que además supriman la propagación de vibraciones en ciertas bandas de frecuencia, ha despertado un gran interés y ha sido respaldada por numerosas investigaciones.

Basándose en lo anterior, el objetivo de este trabajo de título es desarrollar una metodología que permita encontrar diseños óptimos de estructuras reticulares para paneles celulares que logren suprimir o reducir la propagación de vibraciones en ciertas bandas de frecuencias. Se busca desarrollar una metodología que permita encontrar diseños óptimos de estas estructuras para reducir la propagación de vibraciones en ciertas bandas de frecuencia. Para lograrlo, se proponen diseños numéricos de estructuras reticulares utilizando métodos de elementos finitos y se buscan configuraciones que puedan ser fabricadas mediante impresión 3D. Además, se emplea un algoritmo de optimización para encontrar mejores soluciones.

El trabajo se inicia replicando una estructura reticular utilizando la herramienta computacional MATLAB y métodos de elementos finitos, para luego comparar su respuesta en frecuencia con los resultados de la literatura y verificar la estrategia de diseño a seguir. A continuación, se proponen tres estructuras reticulares diferentes que se modelan mediante métodos de elementos finitos, obteniendo así la función de respuesta en frecuencia de cada una. Luego, se desarrolla un algoritmo de optimización para obtener configuraciones óptimas en distintos rangos de frecuencia y analizar la respuesta en frecuencia de cada una en estos rangos, con la expectativa de encontrar configuraciones que sean óptimas.

Se espera que este trabajo proporcione una metodología de trabajo capaz de generar diseños óptimos de estructuras reticulares en paneles celulares con una menor respuesta en ciertas bandas de frecuencia, lo que implica una reducción en la propagación de vibraciones. Además, al lograr la supresión de vibraciones en ciertos rangos de frecuencia, se podrían lograr avances en tecnologías de ingeniería tanto en aplicaciones mecánicas como estructurales.

*A mis padres que siempre velaron por darme las mejores oportunidades,
esto es por y para ustedes.*

Los amo con todo mi corazón

Agradecimientos

Las palabras son pocas para expresar la gratitud que siento hacia cada una de las personas que han estado a mi lado en este largo y complejo camino. En primer lugar, quiero agradecer a mi familia por su constante apoyo y confianza en mí, lo cual me ha motivado a seguir adelante incluso en los momentos más difíciles.

A mis padres, Oscar y Jacqueline, quienes siempre se han esforzado por brindarme lo mejor y han centrado sus esfuerzos en formarme como una persona íntegra. Sin ellos, no sería quien soy hoy en día. Su preocupación por mí ha sido tan grande que me ha impulsado a esforzarme aún más por alcanzar el futuro que anhelo. Espero poder corresponderles todo lo que me han dado algún día. Además, agradezco a mis hermanos, Oscar, Camila y Francisca, quienes siempre han sido un apoyo fundamental en mi vida y me han brindado todo el apoyo que un hermano menor puede necesitar.

A mis amigos que hice en la universidad, Joaquín, Mito, Pulentito, Bastián, Alexis y Toro, quienes han hecho de la vida universitaria un tiempo grato, lleno de recuerdos que atesoraré siempre. Los Ping-Pong, las salidas y las clases, son experiencias que valoro profundamente y por las cuales estoy agradecido de haber compartido con ustedes.

A mis amigos de mecánica, Guiselyn, Juanma y Gustavo, quienes fueron de gran importancia en momentos difíciles, brindándome su apoyo y preocupación de manera incondicional. Las clases, los tiempos muertos en el quinto, los paseos, me hicieron sentir muy cómodo en su compañía. Quiero hacer una mención especial a Guiselyn, quien fue mi compañera de toda la carrera, apoyándonos mutuamente en todo momento. Sin duda, has sido un pilar fundamental durante estos años.

A mis amigos del colegio, quienes siempre estuvieron ahí para mí, a pesar de lo difícil que era coincidir en los tiempos. Su apoyo durante la pandemia y a lo largo de la carrera fue invaluable. Cada consejo, risa y salida juntos me ayudaron a recargar energías y enfrentar cada semana con más energías.

A mi pareja, Fernanda, quien ha sido un apoyo constante durante esta última etapa de mi vida universitaria, siempre pendiente de mí, preocupada por mi bienestar y brindándome consejos para que pueda completar esta etapa de la mejor manera posible.

Por último, como mencioné al inicio, las palabras y las personas son insuficientes. Hay muchas otras personas a las que les debo gratitud. Cada compañero, profesor, conocido y persona que ha cruzado mi camino en estos 6 años merece mi agradecimiento, pues cada uno de ellos contribuyó de alguna manera en esta etapa de mi vida.

Tabla de Contenido

1. Introducción	1
1.1. Motivaciones	1
1.2. Objetivos	2
1.3. Alcances	2
2. Antecedentes	3
2.1. Materiales celulares	3
2.1.1. Metamateriales	4
2.2. Estructuras reticulares	5
2.2.1. Supresión de vibraciones en estructuras reticulares	5
2.2.2. Respuesta en frecuencia	6
2.3. Método de elementos finitos	7
2.4. Optimización de estructuras	7
2.5. Structural Dynamics Toolbox	8
2.6. Algoritmo de optimización	10
2.6.1. Función objetivo	10
2.6.2. Algoritmo <i>particleswarm</i>	11
3. Metodología	12
4. Verificación de estrategia de diseño	13
4.1. Verificación mediante caso de estudio	13
4.1.1. Bases del modelo	13
4.1.2. Resultados verificación	16
5. Formulación de estructuras reticulares	17
5.1. Formulación marco estructura	17
5.2. Celda Pratt	18
5.3. Celda Baltimore	19
5.4. Celda mixta	20
6. Optimización de estructuras	22
6.1. Función objetivo	22
6.1.1. Variables de decisión	22
6.1.2. Restricciones	23
6.1.3. Rangos de frecuencias	23
6.2. Configuración del algoritmo	23
6.3. Optimización estructura Pratt	24

6.4. Optimización estructura Baltimore	27
6.5. Optimización estructura Mixta	30
7. Análisis y discusión de resultados	33
8. Conclusiones	37
Bibliografía	38
Anexos	41
A. Código en Matlab para la verificación	41
B. Formulación estructuras	45
B.1. Formulación estructura Pratt	45
B.2. Formulación estructura Baltimore	49
B.3. Formulación estructura Mixta	55
C. Funciones objetivo	62
C.1. Estructura Pratt	62
C.2. Estructura Baltimore	62
C.3. Estructura Mixta	63
D. Gráficas de las estructuras	64
D.1. Estructura Pratt	64
D.2. Estructura Baltimore	64
D.3. Estructura Mixta	65
E. Optimización de estructuras	66
E.1. Estructura Pratt	66
E.2. Estructura Baltimore	66
E.3. Estructura Mixta	67
F. Resultado de iteraciones	68
F.1. Iteraciones estructura Pratt	68
F.1.1. Rango 1500 - 2000 Hz	68
F.1.2. Rango 2500 - 3000 Hz	69
F.1.3. Rango 3500 - 4000 Hz	70
F.2. Iteraciones estructura Baltimore	71
F.2.1. Rango 1500 - 2000 Hz	71
F.2.2. Rango 2500 - 3000 Hz	72
F.2.3. Rango 3500 - 4000 Hz	73
F.3. Iteraciones estructura Mixta	75
F.3.1. Rango 1500 - 2000 Hz	75
F.3.2. Rango 2500 - 3000 Hz	76
F.3.3. Rango 3500 - 4000 Hz	77

Índice de Tablas

4.1.	Parámetros iniciales del marco.	14
5.1.	Parámetros iniciales del marco.	17
6.1.	Número de variables de decisión para cada estructura.	22
6.2.	Valor RMS para estructuras bases antes de optimizar.	23
7.1.	Resumen de valores RMS para cada estructura.	33

Índice de Ilustraciones

2.1.	Materiales celulares naturales: (a) corcho, (b) madera de balsa, (c) esponja y (d) hueso esponjoso [1].	3
2.2.	Clasificación general de materiales celulares ligeros según la interconectividad estructural y la aleatoriedad de los poros internos [3].	4
2.3.	Propiedades en los materiales según su arquitectura [4].	4
2.4.	Tipos de estructuras reticulares [7].	5
2.5.	(a) Celda unitaria del metamaterial acústico y (b) diseño de la celda unitaria de celosía basada en AM. [9].	5
2.6.	Esquema de una estructura reticular unidimensional basada en AM sujeta a una excitación armónica de tiempo con una restricción fija en el lado izquierdo. [9].	6
2.7.	Respuesta de frecuencia de la esquina derecha en la dirección vertical calculada por el método de elementos espectrales (SEM) y el método de elementos finitos (FEM). [9].	6
2.8.	Tipos de familias de elementos [10].	7
2.9.	Diferentes tipos de optimización estructural [13].	8
4.1.	(a) La estructura de red periódica basada en AM que incluye un núcleo de tipo cruz que consta de dos segmentos con una discontinuidad de salto de radio en cada viga; (b) diseño de la celda unitaria de la red; (c) la viga en cada núcleo. [9].	14
4.2.	Marco base para el trabajo de la estructura a replicar.	15
4.3.	Estructura creada mediante el modelo de elementos finitos para el proceso de verificación.	15
4.4.	Respuesta en frecuencia en desplazamiento versus frecuencia en el modelo. . .	16
5.1.	Marco base (240 x 120 mm) en función a los parámetros de la tabla 5.1. . . .	18
5.2.	Celda unitaria y secuencia de celdas tipo Pratt.	18
5.3.	Estructura tipo Pratt (240 x 120 mm) creada mediante FEM.	19
5.4.	Respuesta en frecuencia para estructura Pratt base.	19
5.5.	Celda unitaria y secuencia de celdas tipo Baltimore.	19
5.6.	Estructura tipo Baltimore (240 x 120 mm) creada mediante FEM.	20
5.7.	Respuesta en frecuencia para estructura Baltimore base.	20
5.8.	Combinación de estructuras reticulares Pratt y armadura K.	21
5.9.	Estructura tipo mixta (240 x 120 mm) creada mediante FEM.	21
5.10.	Respuesta en frecuencia para estructura mixta base.	21
6.1.	Función respuesta en frecuencia de estructura Pratt optimizada en un rango de frecuencias de 1500-2000 Hz.	24
6.2.	Función respuesta en frecuencia de estructura Pratt optimizada en un rango de frecuencias de 2500-3000 Hz.	25
6.3.	Función respuesta en frecuencia de estructura Pratt optimizada en un rango de frecuencias de 3500-4000 Hz.	26

6.4.	Función respuesta en frecuencia de estructura Baltimore optimizada en un rango de frecuencias de 1500-2000 Hz.	27
6.5.	Función respuesta en frecuencia de estructura Baltimore optimizada en un rango de frecuencias de 2500-3000 Hz.	28
6.6.	Función respuesta en frecuencia de estructura Baltimore optimizada en un rango de frecuencias de 3500-4000 Hz.	29
6.7.	Función respuesta en frecuencia de estructura Mixta optimizada en un rango de frecuencias de 1500-2000 Hz.	30
6.8.	Función respuesta en frecuencia de estructura Mixta optimizada en un rango de frecuencias de 2500-3000 Hz.	31
6.9.	Función respuesta en frecuencia de estructura Mixta optimizada en un rango de frecuencias de 3500-4000 Hz.	32
7.1.	Zoom de la FRF de la estructura Pratt optimizada en cada rango de frecuencias.	34
7.2.	Zoom de la FRF de la estructura Baltimore optimizada en cada rango de frecuencias.	35
7.3.	Zoom de la FRF de la estructura Mixta optimizada en cada rango de frecuencias.	36

Capítulo 1

Introducción

En el marco del diseño ingenieril de sistemas mecánicos o componentes de estos, las propiedades mecánicas de los materiales utilizados son de suma importancia para garantizar un alto nivel de calidad, seguridad y rendimiento. En este sentido, a lo largo de los años se ha buscado mejorar las propiedades mecánicas de los materiales con el fin de mitigar efectos no deseados, como la propagación de vibraciones, que puede generar inestabilidad en un sistema y provocar daños al estar expuesto a vibraciones constantes.

En este contexto, surge la idea de los materiales celulares como solución a este problema. Estos materiales tienen topologías diseñadas para obtener propiedades mecánicas poco comunes. Si bien este concepto no es nuevo, ha sido objeto de investigación durante mucho tiempo con el objetivo de mejorar las propiedades mecánicas en la fabricación de componentes.

Una de las aplicaciones de los materiales celulares son los paneles celulares, los cuales presentan una topología particular para mejorar las propiedades mecánicas de la estructura. Un ejemplo de ello son los paneles celulares con estructura en forma de panal de abeja, que les otorga una gran rigidez y resistencia, además de un bajo peso.

Considerando lo anterior, este trabajo consiste en diseñar estructuras reticulares en paneles celulares con el propósito específico de reducir la propagación de vibraciones en el panel. Para lograrlo, se llevará a cabo un análisis exhaustivo de los diferentes aspectos relacionados con el comportamiento dinámico de estos sistemas, incluyendo la influencia de las propiedades del material, la configuración geométrica y los parámetros de diseño. Con esto, se propondrán algunas estructuras que serán evaluadas para encontrar un modelo óptimo que suprima o reduzca la propagación de vibraciones en la estructura.

1.1. Motivaciones

La motivación principal de este trabajo es contribuir mediante un estudio detallado sobre la supresión de vibraciones en paneles celulares con estructuras reticulares en la investigación principal titulada 'Optimal Design of Ultralight Sandwich Panels with Cellular Truss Cores and Large Phononic Band Gaps'. Además, estudiar los paneles celulares es crucial para mejorar la eficiencia, fiabilidad y seguridad de los equipos en ingeniería. El estudio de estos paneles impulsa la innovación y el desarrollo de nuevas tecnologías y materiales que pueden tener un impacto positivo en diversas áreas.

1.2. Objetivos

El objetivo general de este trabajo consiste en desarrollar una metodología que permita encontrar diseños óptimos de estructuras reticulares para paneles celulares que logren suprimir o reducir la propagación de vibraciones en ciertas bandas de frecuencias. Además, se establecen algunos objetivos específicos necesarios para cumplir con el objetivo general. Estos son mencionados a continuación:

- Realizar diseños numéricos de estructuras reticulares de paneles utilizando métodos de elementos finitos como herramienta, buscando configuraciones que sean factibles de fabricar mediante impresión 3D.
- Encontrar soluciones de diseño mediante algoritmos de optimización.

1.3. Alcances

El alcance de este trabajo abarca el diseño de tres tipos de estructuras reticuladas, dos de las cuales están basadas en la literatura existente y la última es una idea propia. Estas estructuras serán evaluadas y comparadas con el objetivo de encontrar la configuración óptima. La formulación del diseño y la optimización se llevarán a cabo utilizando elementos finitos parametrizados, utilizando exclusivamente la herramienta computacional MATLAB. Queda fuera del alcance de este trabajo la fabricación de los paneles celulares y la propuesta de estructuras diferentes a las estudiadas en esta investigación.

Capítulo 2

Antecedentes

En el siguiente capítulo se exponen los tópicos y la teoría necesaria para comprender en detalle el trabajo realizado. En primer lugar, se ofrece una descripción detallada de los materiales celulares con el fin de introducir el concepto de paneles celulares y, de este modo, comprender el material de estudio. A continuación, se abordan las estructuras reticulares y cómo ocurre la supresión de vibraciones en este tipo de estructuras. Por último, se presentan los tópicos requeridos para plantear el problema de diseño y optimización de las estructuras, basándose en el método de elementos finitos y la teoría de estructuras periódicas.

2.1. Materiales celulares

Los materiales celulares son estructuras compuestas por una red de celdas prismáticas o poliédricas con bordes y caras sólidas, las cuales están empaquetadas de manera eficiente para ocupar el espacio [1]. Estos materiales se pueden encontrar en la naturaleza, como el corcho, la madera o los huesos, como se muestra en la Figura 2.1. Estos tipos de materiales han evolucionado durante millones de años, generando una arquitectura optimizada y adquiriendo propiedades únicas que no suelen encontrarse en los materiales típicamente utilizados. Dependiendo de la estructura celular que posean, se pueden obtener propiedades que van desde una relación ejemplar entre rigidez, resistencia y densidad, hasta, por ejemplo, la supresión de ondas mecánicas [2], el cual es el tema de interés en este trabajo.

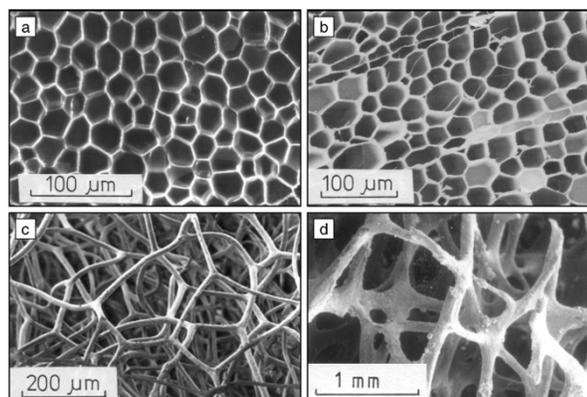


Figura 2.1: Materiales celulares naturales: (a) corcho, (b) madera de balsa, (c) esponja y (d) hueso esponjoso [1].

La investigación sobre este tipo de materiales ha ido aumentando a lo largo de los años, sirviendo de inspiración para el desarrollo de tecnologías en diversas aplicaciones. Se han creado estructuras celulares artificiales, siendo un excelente ejemplo de ello la espuma sintética. Estos materiales celulares se clasifican en estructuras celulares abiertas o cerradas, según la interconectividad estructural de los poros internos [3]. Además, se pueden dividir en estructuras periódicas o estocásticas, dependiendo de la aleatoriedad de los poros internos, como se muestra en la Figura 2.2.

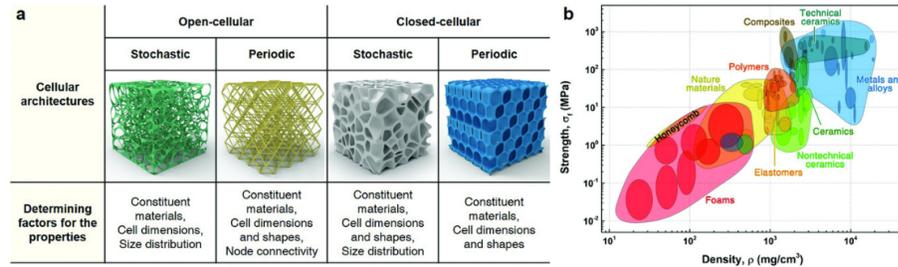


Figura 2.2: Clasificación general de materiales celulares ligeros según la interconectividad estructural y la aleatoriedad de los poros internos [3].

Con el paso del tiempo, se ha profundizado en el estudio de los materiales celulares, buscando modificar la estructura de las celdas y aplicarlos en diversas áreas con el fin de mejorar sus propiedades. De esta manera, al alterar la arquitectura celular, es decir, la configuración espacial de la estructura, se pueden obtener propiedades mecánicas poco comunes en los materiales [4]. Este concepto se refleja en la Figura 2.3, donde se muestran materiales con distintas arquitecturas y propiedades. A partir de esta idea de introducir la arquitectura en los materiales celulares, surge el concepto de metamaterial para adquirir propiedades deseadas.

Cellular architecture	None	Random	Ordered	Ordered and location specific
Properties	Continuous and homogeneous	Homogeneous at scales > cell	Homogeneous and highly anisotropic	Inhomogeneous and highly anisotropic
Design degrees of freedom	Solid constituent	Solid constituent, cell size	Solid constituent, cell size, and orientation	Cell size/shape, node topology, ligament shape, material...

Figura 2.3: Propiedades en los materiales según su arquitectura [4].

2.1.1. Metamateriales

Los metamateriales son materiales artificiales cuyas propiedades son dependientes del tipo de estructura que posean. Este tipo de materiales suelen ser diseñados en base a la estructura de una celda unitaria y cambian las propiedades del material para obtener ventajas con respecto a otro tipo de materiales [5]. Suelen estar diseñados para aplicaciones electromagnéticas, ópticas y acústicas [6]

2.2. Estructuras reticulares

Las estructuras reticulares, también conocidas como armaduras, son definidas en el campo de la ingeniería civil como conjuntos de elementos estructurales dispuestos en forma de uno o más triángulos [7]. La disposición triangular proporciona estabilidad a estas estructuras, ya que es imposible que cambien de forma bajo carga, a menos que uno o más lados se flexionen o se quiebren. Existen diversas configuraciones de estructuras reticulares según las aplicaciones para las que se utilicen. En la Figura 2.4 se pueden observar diferentes tipos de estructuras reticulares comúnmente utilizadas en ingeniería estructural.

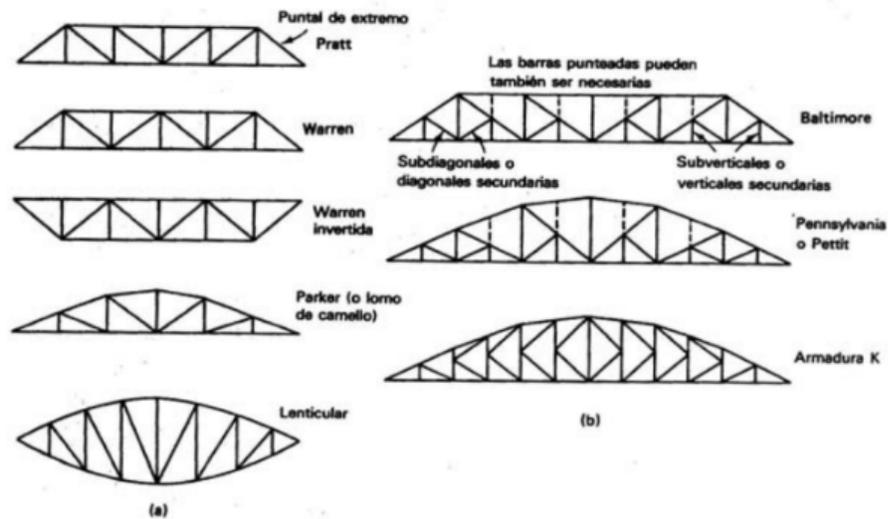


Figura 2.4: Tipos de estructuras reticulares [7].

Este tipo de configuraciones de estructuras reticulares aprovechan de manera eficiente el espacio, lo que resulta en un peso liviano en comparación con otras configuraciones, sin comprometer la rigidez deseada. Además, estas configuraciones suelen ofrecer buenos resultados en términos de supresión de vibraciones. Un ejemplo de ello se encuentra en Guo J. et Al. [8] donde se utilizan modelos de paneles sándwich con núcleos reticulares para lograr la supresión de vibraciones.

2.2.1. Supresión de vibraciones en estructuras reticulares

Basándose en An X. et Al. [9], donde se trabaja con una estructura periódica con configuración similar a la mostrada en la Figura 2.5. Esta configuración se inspira en los metamateriales acústicos (AM), los cuales exhiben propiedades de atenuación de ondas de baja frecuencia.

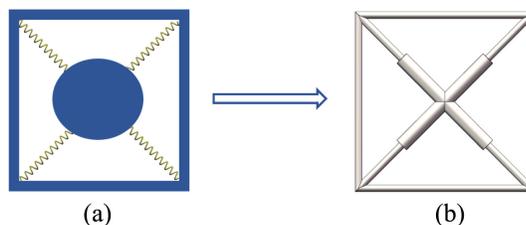


Figura 2.5: (a) Celda unitaria del metamaterial acústico y (b) diseño de la celda unitaria de celosía basada en AM. [9].

Este tipo de celda presenta la capacidad de debilitar la vibración externa de la estructura primaria cuando las ondas elásticas la atraviesan. Esto se logra gracias a la resonancia local de los subsistemas en rangos de frecuencia relativamente bajos, lo que impide la propagación de las ondas acústicas hacia adelante [9]. En este caso, los subsistemas se refieren a las barras con discontinuidades que se muestran en la Figura 2.5.

2.2.2. Respuesta en frecuencia

La función de respuesta en frecuencia (FRF) es una función de transferencia que se evalúa en el eje de las frecuencias [10]. Nos permite analizar la respuesta en frecuencia en uno o más nodos de una estructura cuando se aplica una fuerza unitaria en otro nodo. De esta manera, podemos comprender cómo se propagan las vibraciones en la estructura dentro de un rango determinado de frecuencias.

Siguiendo con el ejemplo de An X. et Al. [9], se realiza un análisis de la respuesta en frecuencia de una estructura similar a la mostrada en la Figura 2.6. Para esto, se utilizan tanto el método de elementos finitos (FEM) como el método de elementos espectrales (SEM). El gráfico de la FRF obtenido mediante ambos métodos se muestra en la Figura 2.7. En dicho gráfico se puede observar que en un rango específico de frecuencias, la respuesta es menor. Esto indica que, al utilizar esta configuración basada en estructuras reticulares, es posible reducir la propagación de vibraciones en la estructura.

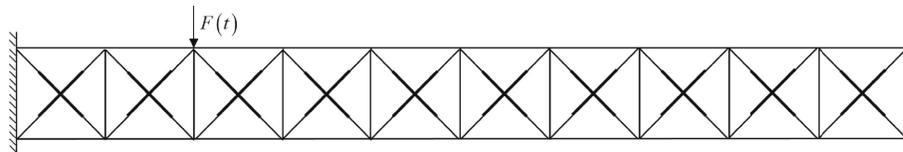


Figura 2.6: Esquema de una estructura reticular unidimensional basada en AM sujeta a una excitación armónica de tiempo con una restricción fija en el lado izquierdo. [9].

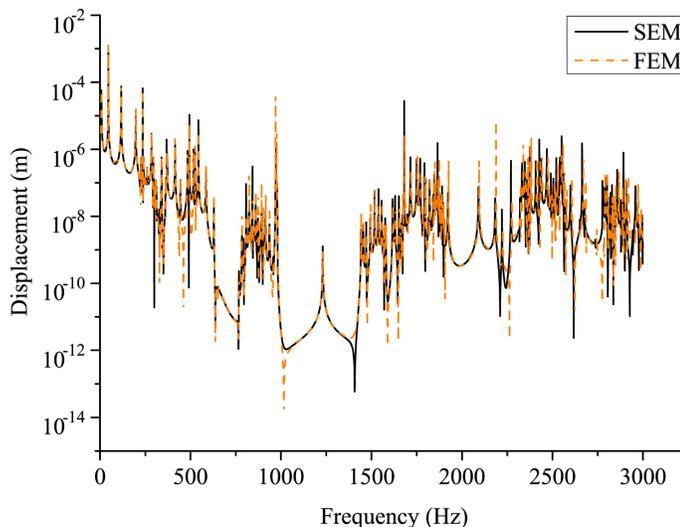


Figura 2.7: Respuesta de frecuencia de la esquina derecha en la dirección vertical calculada por el método de elementos espectrales (SEM) y el método de elementos finitos (FEM). [9].

2.3. Método de elementos finitos

El método de elementos finitos (FEM) es una herramienta numérica que permite resolver ecuaciones diferenciales parciales mediante aproximaciones definidas localmente en subdominios no traslapados (llamados elementos) que conjuntamente forman el dominio global donde se busca una solución al problema gobernado por la EDP [11]. Existen 3 tipos de elementos dependiendo de los grados de libertad que posea, en la Figura 2.8 se pueden ver las familias de elementos que existen.

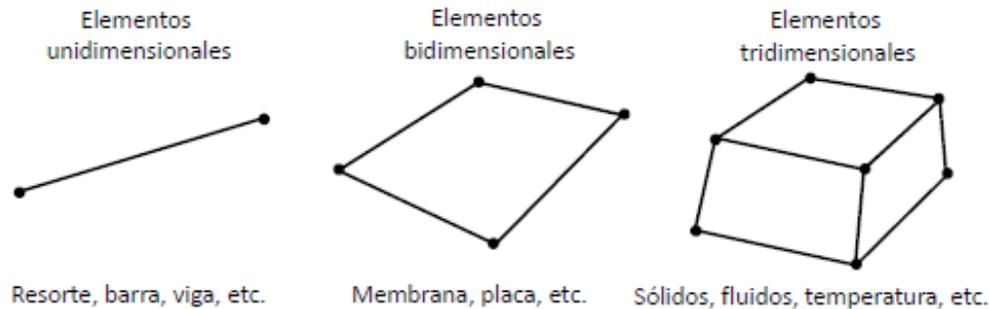


Figura 2.8: Tipos de familias de elementos [10].

Cabe destacar que para el funcionamiento de este método es de suma importancia entregarle varios parámetros de entrada, como lo son las propiedades del material que se utiliza, las características geométricas de la estructura y las condiciones de borde que gobiernan a la estructura. Además de esto, es de suma importancia realizar un mallado correcto de la estructura, para así obtener información relevante de la estructura a partir del cálculo de ecuaciones en los nodos de la malla [12].

2.4. Optimización de estructuras

Cuando se habla de optimización de estructuras hay que dejar claro que se puede abordar de distintos enfoques. En este caso, tenemos 3 problemas de optimización [13]:

- **Optimización de tamaño:** Se busca optimizar las medidas y dimensiones de los elementos que componen la estructura. No presenta muchas variables, por lo que su formulación es más sencilla.
- **Optimización de forma:** Como dice su nombre, busca la forma óptima partiendo desde una topología fija y así ir modificando la conectividad de los elementos o eliminando algunos innecesarios.
- **Optimización de topología:** Se busca que exista una distribución óptima en la estructura mediante la modificación o supresión tanto de los elementos como de la conectividad entre estos. Este problema termina siendo más complejo debido a que presenta una gran cantidad de variables.

En la Figura 2.9 se puede apreciar la diferencia sobre estos tres enfoques de optimización para una misma estructura reticular.

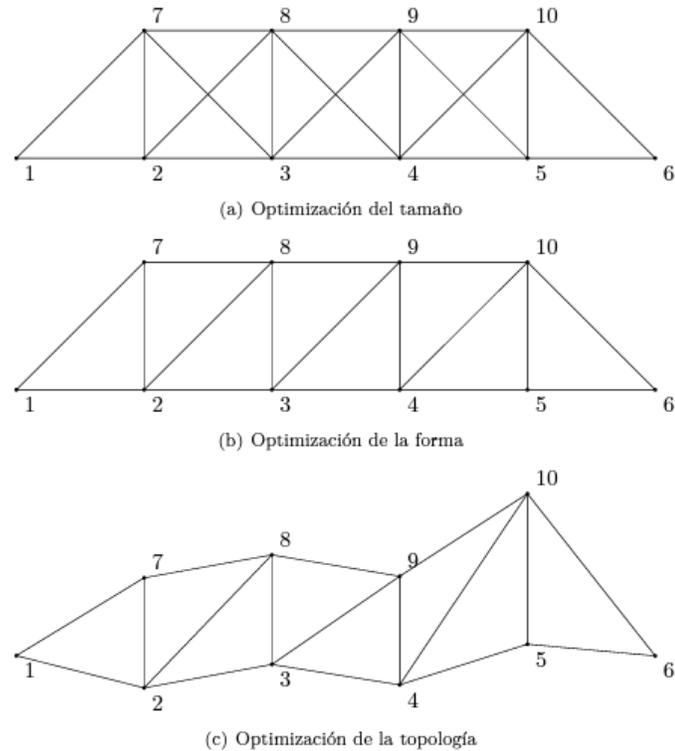


Figura 2.9: Diferentes tipos de optimización estructural [13].

2.5. Structural Dynamics Toolbox

Las bases del modelo de trabajo se fundamentan en el paquete SDT de MATLAB [14], el cual se utiliza principalmente para generar un modelo de las estructuras empleadas mediante elementos finitos. El paquete SDT utiliza el objeto 'model' como su sintaxis principal, el cual recibe la información de los nodos, elementos, propiedades y condiciones adicionales. Este modelo se utiliza para obtener información sobre los modos de vibración y la respuesta en frecuencia correspondiente.

Para crear nodos en este modelo estructural, se debe utilizar el comando '**model.Node**' de la siguiente manera:

```
1 model.Node = [];
2 model.Node = [N 0 0 0 x y z];
```

Donde se inicializa la matriz que contendrá los nodos. 'N' corresponde al número de nodo, seguido por una configuración del Toolbox SD. Por último, 'x', 'y' y 'z' representan las coordenadas del nodo en el plano cartesiano.

Para definir las propiedades de los materiales con los que se trabaja, se utiliza el comando '**model.pl**' de la siguiente forma:

```
1 model.pl = [];
2 model.pl=[model.pl; MatId type E nu rho];
```

Primero se crea la matriz donde se especificarán las propiedades de los materiales utilizados. Luego, se incorporan los materiales según un número de identificación (**'MatId'**), el tipo de elemento (**'type'**) y sus propiedades como el módulo de elasticidad (**'E'**), el coeficiente de Poisson (**'nu'**) y la densidad (**'rho'**).

Para definir las propiedades del elemento, se utiliza el comando **'model.il'** de la siguiente manera:

```
1 model.il = [];  
2 model.il=[mdl.il; ProId type Iz Ix Iy A]
```

Primero se crea la matriz donde se especificarán las propiedades de los elementos utilizados. Luego, se incorporan las propiedades según un número de identificación (**'ProId'**) y el tipo de propiedad (**'type'**). Además, se definen su rigidez torsional (**'Iz'**), los momentos de inercia (**'Ix'** e **'Iy'**) y el área de la sección (**'A'**).

Para crear un elemento en la estructura, se debe utilizar el comando **'model.Elt'** como se muestra a continuación:

```
1 model.Elt=[Inf abs('beam1') 0];  
2 model.Elt = [model.Elt; N_i N_f i j dir];
```

La primera línea define el tipo de elemento a utilizar, donde **'abs('beam1')** indica que el tipo de elemento se define como una viga Euler-Bernoulli. Luego, la segunda línea añade información al modelo, donde **'N_i'** y **'N_f'** corresponden a los nodos de inicio y fin del elemento, **'i'** proporciona el ID del material utilizado para este elemento (definido en **'model.pl'**), **'j'** proporciona el ID de las propiedades del elemento (definidas en **'model.il'**) y **'dir'** define la dirección de la normal al elemento.

Con la estructura modelada, lo siguiente que se debe realizar es colocar condiciones en algunos nodos según sus grados de libertad. Para ser más claros, tomemos el siguiente ejemplo:

```
1 model0 = fe_case(model, 'FixDof', 'AllDof', [1 4 5 .03 .04 .05],...  
2           'SensDof', 'Sensors', [4.02], ...  
3           'DofLoad', 'Force', [2.02]);
```

La primera línea define los grados de libertad y nodos que están fijos en la estructura. Según el ejemplo, los nodos 1, 4 y 5 están empotrados, mientras que los grados de libertad 3, 4 y 5 están fijos. En la segunda línea, se definen los nodos (con su grado de libertad) en los cuales se mide la respuesta. En el ejemplo, la respuesta se mide en el nodo 4 en su eje y. Por último, la tercera línea define la fuerza de carga aplicada en un cierto nodo con su respectivo grado de libertad. En el ejemplo, se define una fuerza de carga en el nodo 2 en dirección vertical.

Con el modelo generado, se construyen las matrices de rigidez y masa, se obtienen los modos de vibración y se calculan las funciones de respuesta en frecuencia según los siguientes comandos:

```

1 % Construir matrices de rigidez y masa
2 [m0,k0,mdof0] = fe_mk(model0,'Options',[]);
3
4 % Determinar modos de vibración
5 def0 = fe_eig({m0, k0, mdof0}, [5 length(m0) 1e3]);
6
7 % Calcular funciones de respuesta en frecuencia
8 IIXh0 = nor2xf(def0, 0, model0, 1:0.1:3000, ['hz', 'dis']);

```

Por último, si se desean visualizar los modos de vibración y las funciones de respuesta en frecuencia, es necesario utilizar los siguientes comandos:

```

1 % Graficar modos
2 cf = feplot(model0, def0);
3
4 % Graficar FRFs
5 figure;
6 semilogy(1:1:3000, mean(abs(IIXh0'))) % FRF en un rango de frecuencias de 1 a 3000 Hz

```

2.6. Algoritmo de optimización

Considerando todo lo mencionado anteriormente, el propósito del trabajo de titulación es desarrollar una metodología de trabajo destinada a optimizar estructuras con el objetivo de reducir la propagación de vibraciones. En este sentido, se emplea un algoritmo de optimización que consiste en realizar, para cada tipo de estructura propuesta, una optimización estructural utilizando el lenguaje de programación MATLAB.

2.6.1. Función objetivo

En un problema de optimización, se busca encontrar la mejor solución de una función objetivo (ya sea buscando un máximo o un mínimo). En nuestro caso, al buscar reducir la propagación de vibraciones, el objetivo es disminuir la función de respuesta en frecuencia (FRF) en un rango limitado. Sin embargo, trabajar con estas funciones no es sencillo, por lo que es necesario realizar ciertos ajustes antes de minimizarlas.

Como se explicó en la sección 2.5, las FRF se encuentran en la matriz 'IIXh0'. Para obtener una medida a minimizar de este valor complejo, se calcula el promedio del valor absoluto de las funciones de respuesta en frecuencia y se procede a calcular la raíz cuadrada media (RMS) [15] de estos promedios, utilizando la siguiente ecuación:

$$RMS = \sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2} = \sqrt{\frac{x_1^2 + x_2^2 + \dots + x_N^2}{N}} \quad (2.1)$$

Por ejemplo, si deseamos obtener los valores promedio de las funciones de respuesta en frecuencia en un rango de 1000 a 2000 Hz y calcular el valor de la RMS, debemos ejecutar los siguientes comandos:

```

1 xfp = mean(abs(xf(1000:2000,:)));
2 rm0 = sqrt(sum(abs(xfp).^2)/length(xfp));

```

2.6.2. Algoritmo *particleswarm*

En este trabajo, se utiliza la función **particleswarm** [16] del paquete **Global Optimization Toolbox** de MATLAB. Este algoritmo se encuentra basado en enjambres de partículas que se utilizan para encontrar respuesta a problemas de optimización continua, y también, no lineales. Funciona tomando como ejemplo el comportamiento de enjambres naturales, como las aves cuando realizan su vuelo o los peces cuando nadan. En específico, cada "partícula" (o solución candidata) en el espacio de búsqueda de la optimización representa una posible solución al problema de optimización que se busca resolver.

El funcionamiento de este algoritmo se podría detallar en las siguientes etapas:

1. **Inicialización:** Se generan aleatoriamente un conjunto de partículas en el espacio de búsqueda del problema. Cada partícula tiene una posición inicial y una velocidad inicial asociada.
2. **Evaluación de la función objetivo:** Para cada partícula, se evalúa la función objetivo en su posición actual. La función objetivo es la función que se busca minimizar o maximizar.
3. **Actualización:** Cada partícula mantiene un registro de su mejor posición y velocidad local, es decir, la posición y velocidad en la que ha obtenido el mejor valor de la función objetivo hasta el momento. Además, se mantiene un registro de la mejor posición y velocidad global entre todas las partículas del enjambre.
4. **Repetición del proceso:** Los pasos 2 y 3 se repiten durante un número predeterminado de iteraciones o hasta que se cumpla algún criterio de convergencia, como alcanzar un cierto valor de la función objetivo o que las partículas converjan a una región del espacio de búsqueda.

Por otro lado, el algoritmo de optimización se define como una función con la siguiente forma:

```
1 [x, fval] = particleswarm(fun, nvars, lb, ub);
```

Donde los parámetros de la función se describen a continuación:

- **fun:** Es la función objetivo que se está minimizando.
- **nvars:** Es el número de variables de decisión de la función objetivo.
- **lb:** Es un vector que contiene los límites inferiores de las variables de decisión.
- **ub:** Es un vector que contiene los límites superiores de las variables de decisión.

Además de estos parámetros, la función puede recibir una configuración especial que incluye un número máximo de iteraciones o una tolerancia máxima para lograr la convergencia. Como resultado, la función devuelve el valor óptimo '**x**' de las variables de decisión y el valor de la función objetivo en ese punto '**fval**'.

Capítulo 3

Metodología

El presente trabajo se fundamenta en una metodología sólida y estructurada que permitirá generar estructuras óptimas que reduzcan la propagación de vibraciones en ciertos rangos de frecuencias. A continuación, se presenta en detalle el paso a paso requerido para la realización de este trabajo:

1. **Verificación de estrategia de diseño:** En esta etapa se genera un modelo de elementos finitos que replica la estructura reticular de la Figura 2.6 y obtener la función respuesta en frecuencia (FRF) para así compararla con la obtenida en el artículo **Elastic wave and vibration bandgaps in planar square metamaterial-based lattice structures** [9].
2. **Formulación de estructuras de estudio:** En esta etapa se proponen estructuras reticulares (en este caso 3) y se modelan mediante el método de elementos finitos y se obtienen las funciones FRF de cada modelo.
3. **Optimización de estructuras:** Se construye el algoritmo de optimización mostrado en la sección 2.6 del capítulo 2 y se obtienen configuraciones óptimas y la respuesta en frecuencia de cada una.
4. **Análisis de resultados y conclusiones:** A partir de los resultados obtenidos, se comparan las funciones de respuesta en frecuencia, la cual entregará información sobre como propagan las vibraciones cada configuración, con lo que se puede obtener una configuración que tenga resultados bastante mejores que las otras.

El desarrollo de cada actividad se ve en detalle en cada uno de los capítulos siguientes.

Capítulo 4

Verificación de estrategia de diseño

En este capítulo se describe la creación del modelo de elementos finitos de una estructura reticular utilizando el toolbox SDT de MATLAB. Este modelo se construye tomando como referencia un ejemplo de la literatura que ya ha sido verificado. El propósito principal de este modelo es validar la estrategia de diseño que se utilizará a lo largo de la memoria. Por lo tanto, este modelo constituye el primer paso antes de proceder a generar las estructuras que serán objeto de estudio en este trabajo.

4.1. Verificación mediante caso de estudio

Para crear un modelo capaz de generar estructuras reticulares de manera precisa, es fundamental tener una base de comparación para corroborar la exactitud de los resultados obtenidos por el modelo de elementos finitos. Por tanto, tomaremos como referencia lo mencionado en la sección 2.2.2 y se tomarán los resultados obtenidos del trabajo del análisis de vibraciones mecánicas en estructuras reticulares de An X. et Al. [9] como punto de partida.

Dado que es necesario validar el modelo, es crucial replicar lo más fielmente posible el modelo de referencia. Este modelo de referencia describe una estructura reticular de gran tamaño que se analiza mediante el método de elementos finitos (FEM) y el método de elementos espectrales (SEM). Para el propósito de este trabajo, nos centraremos únicamente en el estudio y la respuesta obtenida a través del método de elementos finitos (indicada por la leyenda de color naranja punteada en la Figura 2.7).

4.1.1. Bases del modelo

El modelo generado consiste principalmente en la formulación de una estructura a partir de ciertos parámetros de entrada requeridos. Estos parámetros definen principalmente al esqueleto de la estructura, dejando a la configuración del enrejado dependiente de estos parámetros. A continuación, es que se definen cada uno de estos parámetros:

- a : Longitud de la celda unitaria.
- b : Ancho de la celda unitaria.
- n_{c_x} : Número de celdas unitarias en el eje x.
- n_{c_y} : Número de celdas unitarias en el eje y.

Además de estos parámetros que definen el marco de la estructura, se tienen otros dos parámetros para este caso en especial, principalmente debido al cambio de sección que ocurre en la cruceta de cada celda unitaria.

- r : Radio de sección menor.
- R : Radio de sección mayor.

De manera visual, la estructura que se busca recrear, junto con los parámetros que la definen, se pueden observar en la figura 4.1.

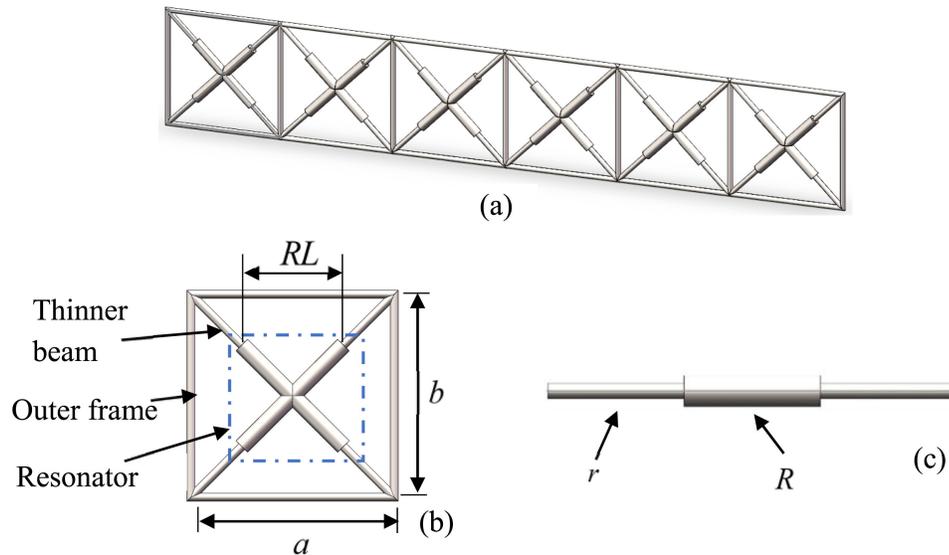


Figura 4.1: (a) La estructura de red periódica basada en AM que incluye un núcleo de tipo cruz que consta de dos segmentos con una discontinuidad de salto de radio en cada viga; (b) diseño de la celda unitaria de la red; (c) la viga en cada núcleo. [9].

La primera tarea del modelo es construir el marco base de la estructura. Como se mencionó antes, se requiere de unos parámetros de entrada, los cuales para el caso de estudio son los referidos en la Tabla 4.1:

Tabla 4.1: Parámetros iniciales del marco.

a [m]	b [m]	nc_x	nc_y
0.5	0.5	10	1

El siguiente paso consiste en crear los nodos y elementos de la estructura base. Utilizando el paquete 'Structural Dynamics' de MATLAB (sección 2.5) es que se puede crear el modelo estructural utilizando elementos finitos. Como resultado, en la Figura 4.2 se puede observar el marco que sirve como base para la configuración del enrejado.

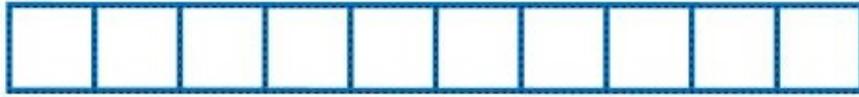


Figura 4.2: Marco base para el trabajo de la estructura a replicar.

Posteriormente, se procede a trabajar en el enrejado interno de la estructura, que consiste en la repetición de celdas unitarias con forma de cruceta con cambios de sección en sus barras. Este enrejado se formula simplemente con barras diagonales de un extremo a otro, generando las conexiones entre los extremos de estas barras con las esquinas de cada celda para así, darle rigidez a la estructura y que se encuentren conectados de manera correcta. Esto tiene como resultado en la estructura mostrada en la Figura 4.3, la cual es similar a la del estudio (Figura 2.6).



Figura 4.3: Estructura creada mediante el modelo de elementos finitos para el proceso de verificación.

Una vez definida la estructura, se deben de definir las propiedades del material y de los elementos. Como la estructura estudiada es completamente de acero, presenta las siguientes propiedades:

- **Modulo de elasticidad:** $E = 206 \text{ [GPa]}$.
- **Coefficiente de Poisson:** $\nu = 0.3$.
- **Densidad:** $\rho = 7850 \text{ [}\frac{\text{kg}}{\text{m}^3}\text{]}$.

Por otro lado, los elementos creados tendrán distintas propiedades dependiendo de la sección que presenten, en este caso, se trabaja con barras circulares macizas, por lo que se debe de considerar los momentos de inercia mostrados a continuación:

$$I_x = I_y = \frac{1}{4} \cdot \pi \cdot r^4 = \frac{I_z}{2} \quad (4.1)$$

De esta forma utilizando los radios correspondientes a cada sección, es que se definen las propiedades de los elementos de manera correcta.

4.1.2. Resultados verificación

Una vez definido el modelo de elementos finitos, se obtiene la función de respuesta en frecuencia (FRF) de la estructura para compararlos con los resultados obtenidos en [9]. Para obtener los resultados se trabaja la estructura de la Figura 4.3. La FRF se obtiene aplicando una excitación vertical en la esquina superior derecha de la segunda celda unitaria, y midiendo la respuesta de frecuencia en la esquina superior derecha de la estructura, tal como se aprecia en la Figura 2.6. La FRF obtenida del modelo se puede apreciar en la Figura 4.4.

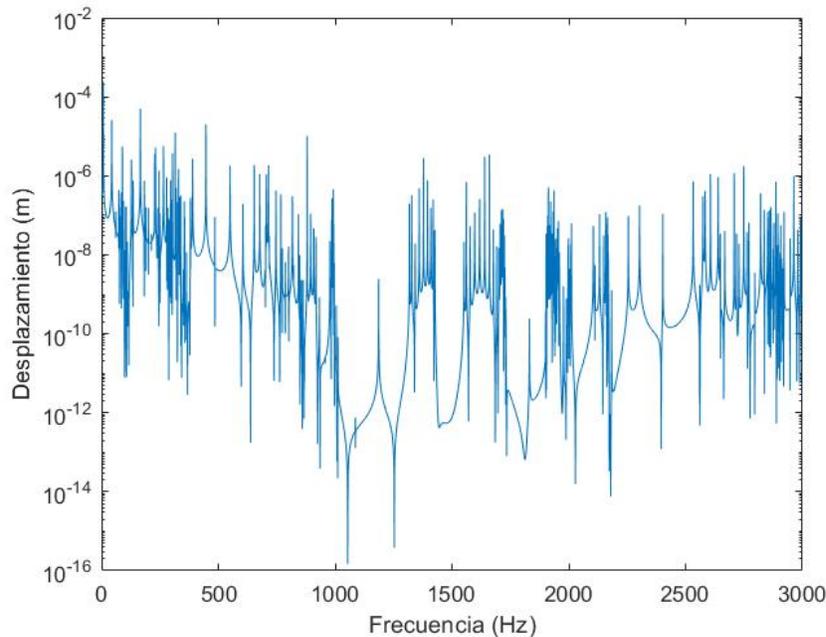


Figura 4.4: Respuesta en frecuencia en desplazamiento versus frecuencia en el modelo.

Es importante destacar que el estudio no detalla el mallado realizado mediante el método de elementos finitos, ni tampoco menciona el amortiguamiento utilizado, por lo que los resultados se basan en un mallado y amortiguamiento propios del modelo.

Considerando lo expuesto anteriormente, se observa que la respuesta obtenida es bastante similar a la del estudio de referencia (Figura 2.7), mostrando una notable disminución entre los 1000 y 1500 Hz. Aunque el problema planteado no sea exactamente el mismo, sino similar, los resultados obtenidos son bastante parecidos, lo que valida el funcionamiento de la estrategia de diseño.

Con esto, se cumple satisfactoriamente con el primer propósito del trabajo y se está preparado para avanzar a las siguientes etapas del trabajo.

Por último, el detalle del código MATLAB del modelo de elementos finitos para realizar la verificación se encuentra en el Anexo A.

Capítulo 5

Formulación de estructuras reticulares

En este capítulo se da el detalle sobre la formulación de las estructuras reticulares propuestas para el estudio de la supresión de vibraciones. En particular, se trabajara con celdas unitarias inspiradas en estructuras reticulares de la literatura. Estas celdas corresponden a estructuras reticulares del tipo **Pratt**, **Baltimore** y una celda mixta que combina conceptos de las estructuras **Pratt** y **Armadura K**. Todos estos tipos de estructura se hacen referencia en la Figura 2.4 de la sección 2.2.

5.1. Formulación marco estructura

Cada estructura estudiada consta principalmente de una configuración de celdas unitarias repetidas a lo largo de la estructura, por lo que en cada uno de los modelos de elementos finitos creados se requiere de una base estructural igual para todos, para luego avanzar con sus enrejados internos respectivos. De manera sencilla, la estructura base consta de una repetición de celdas unitarias sin enrejado interno en ellas.

De manera similar a la verificación, se construye un marco específico, el cual recibe forma en base a parámetros de entrada (mencionados en la sección 4.1.1 con respecto a la celda unitaria). Los valores de estos parámetros de entrada se muestran en la Tabla 5.1.

Tabla 5.1: Parámetros iniciales del marco.

a [mm]	b [mm]	nc_x	nc_y
30	30	8	4

Donde, si se recuerda lo mencionado en los capítulos anteriores, ' a ' y ' b ' corresponden a las dimensiones de la celda unitaria, mientras que ' nc_x ' y ' nc_y ' representan la cantidad de celdas por eje.

Cabe destacar que estos parámetros solo definen las dimensiones de la estructura base. Las cuales se adecuan a una estructura que pueda ser diseñada mediante impresión 3D y sus dimensiones no superen el área de impresión de la impresora que se quiere utilizar, la cual es una **Form 3L** de dimensiones $33,5 \times 20 \times 30$ cm. De esta forma es que se obtiene un marco base con un largo de 240 mm y un ancho de 120 mm, el cual se observa en la Figura 5.1.

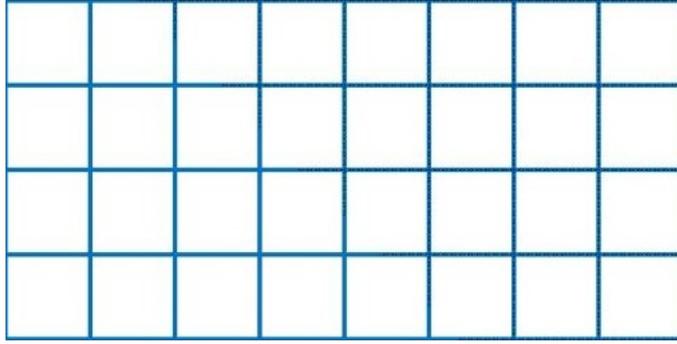


Figura 5.1: Marco base (240 x 120 mm) en función a los parámetros de la tabla 5.1.

Además, es necesario definir el material del cual estará hecha la estructura, el cual consiste en la resina de impresión 3D 'gray resin V4', la cual presenta las siguientes propiedades:

- **Modulo de elasticidad:** $E = 3.580 [GPa]$.
- **Coefficiente de Poisson:** $\nu = 0.3331$.
- **Densidad:** $\rho = 1333 [\frac{kg}{m^3}]$

5.2. Celda Pratt

Para este tipo de celdas, se necesitaba de una orientación específica al formular la estructura, ya que su geometría implica que las barras diagonales están conectadas entre sus extremos, de modo que el nodo en el que finaliza una barra, es el comienzo de la siguiente. De esta forma es que se esperan celdas y secuencias como las mostradas en la Figura 5.2.

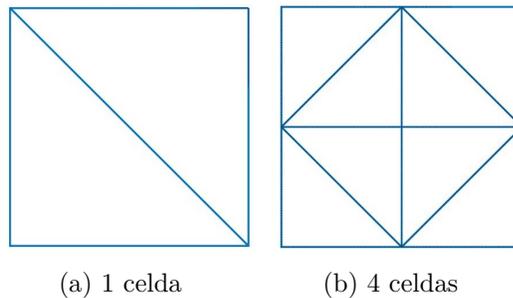


Figura 5.2: Celda unitaria y secuencia de celdas tipo Pratt.

Con el tipo de configuración mostrada en la figura anterior, es que se procede a formar la estructura reticular. Para la formación del enrejado, el único parámetro que varía es el radio de cada viga. De manera inicial se dispondrá de un radio uniforme en las secciones de las vigas, el cual toma un valor tal que $r_i = 1 [mm]$ (valor razonable para el tamaño de la estructura) para todo valor de i . De esta forma, es que se obtiene una estructura como la mostrada en la Figura 5.3, a la cual se le puede observar en la Figura 5.4 su función respuesta en frecuencia.

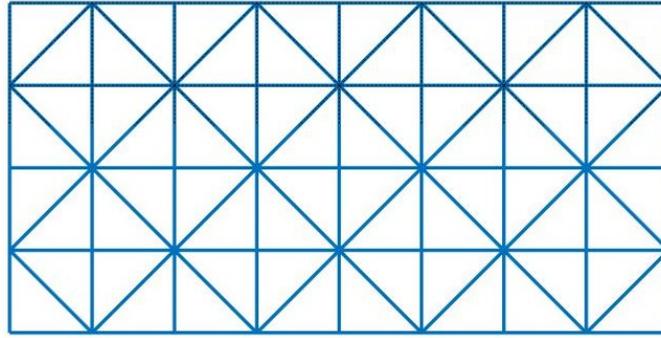


Figura 5.3: Estructura tipo Pratt (240 x 120 mm) creada mediante FEM.

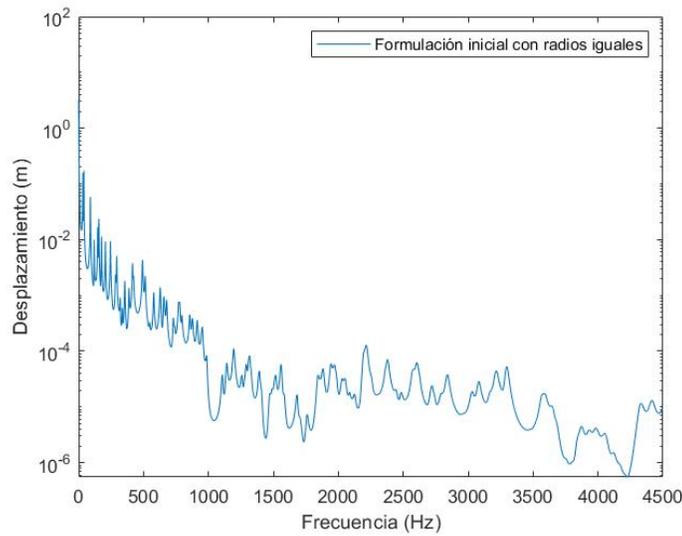


Figura 5.4: Respuesta en frecuencia para estructura Pratt base.

5.3. Celda Baltimore

Este tipo de celdas es bastante similar a las mencionadas anteriormente. La base de celdas tipo 'diamante' prevalece en este tipo de estructuras, pero la gran diferencia radica en que se agregan barras de soporte a estas barras diagonales de la estructura. Este tipo de celdas y la secuencia que toma se pueden ver representadas en la Figura 5.5.

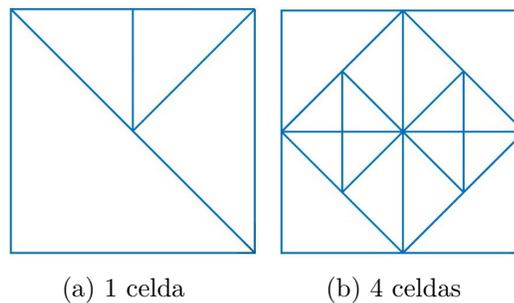


Figura 5.5: Celda unitaria y secuencia de celdas tipo Baltimore.

Con la configuración definida, es que se procede de la misma manera que la estructura anterior. Se escoge preliminarmente un radio uniforme $r_i = 1[mm]$ para todas las secciones de las vigas del enrejado. Con este planteamiento, es que se obtiene la estructura mostrada en la Figura 5.6, y con ella, su función respuesta en frecuencia en la Figura 5.7.

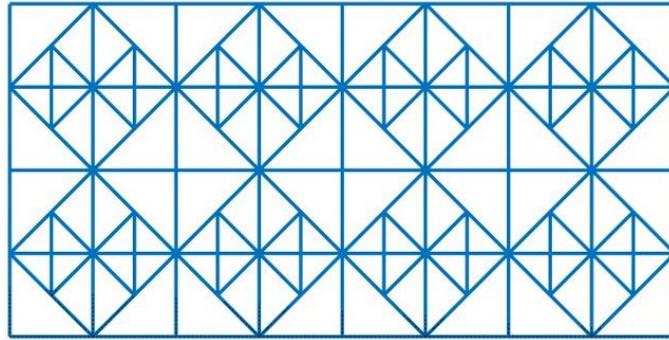


Figura 5.6: Estructura tipo Baltimore (240 x 120 mm) creada mediante FEM.

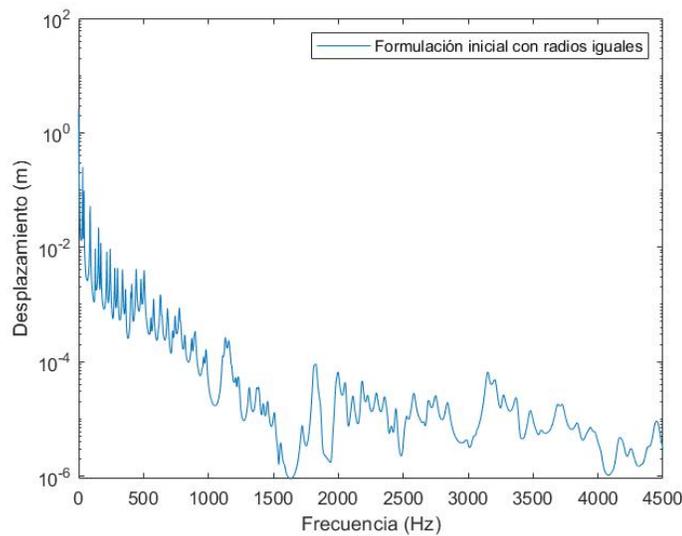


Figura 5.7: Respuesta en frecuencia para estructura Baltimore base.

5.4. Celda mixta

Para este tipo de configuración, se tiene una secuencia distinta a las anteriormente mencionadas. Esto pues, existen celdas diferentes en la estructura, siempre siguiendo un patrón. De manera simple, la estructura parte por sus extremos con una celda tipo Pratt, para así, mediante una celda de conexión, comenzar con una estructura de armadura K. Esto se ve mejor reflejado en la Figura 5.8, donde se puede observar como sería el patrón de este tipo de estructura, la cual requiere de 6 celdas como mínimo.

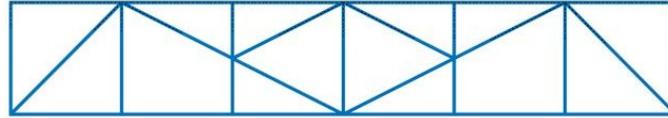


Figura 5.8: Combinación de estructuras reticulares Pratt y armadura K.

Ahora, repitiendo este patrón para adaptarlo al marco base y utilizando la misma condición anterior sobre los radios de sección de las vigas del enrejado, es que resulta la estructura mostrada en la Figura 5.9. Además, se muestra la función respuesta en frecuencia de este modelo base en la Figura 5.10.

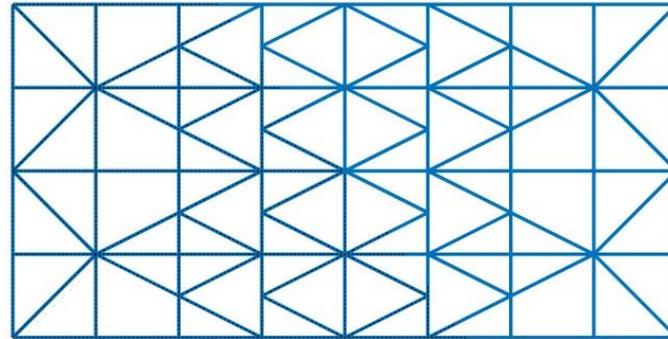


Figura 5.9: Estructura tipo mixta (240 x 120 mm) creada mediante FEM.

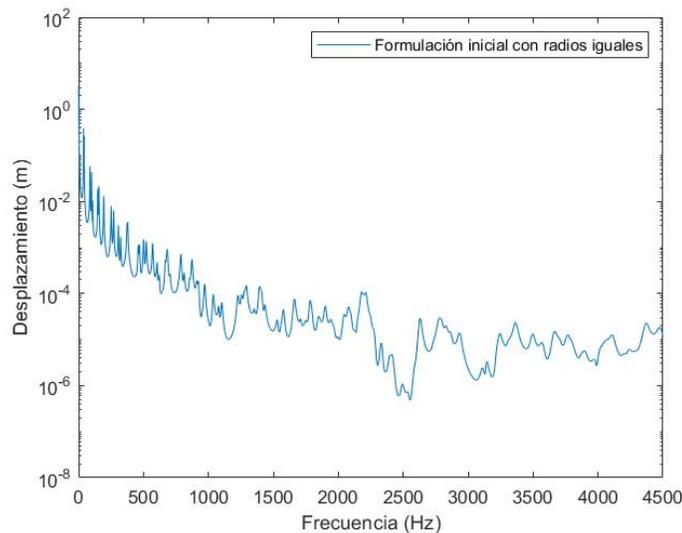


Figura 5.10: Respuesta en frecuencia para estructura mixta base.

Cada uno de los resultados mostrados en esta sección serán la base que se tendrá para el proceso de optimización de las estructuras mencionado en el siguiente capítulo. El detalle sobre la formulación de todas las estructuras y la obtención de las funciones de respuesta en frecuencia se encuentran en el Anexo B.

Capítulo 6

Optimización de estructuras

6.1. Función objetivo

Todo problema de optimización busca encontrar una solución óptima de una función objetivo. En nuestro caso buscamos que la función respuesta en frecuencia (FRF) tome valores mínimos en un rango limitado.

Como se vio en la sección 2.6.1, la función objetivo con la que se trabaja se basa en la RMS de los promedios de las FRF's de la estructura, la cual es un valor escalar que permite trabajar con un problema de optimización para encontrar los parámetros que minimicen este valor. De esta forma es que se consigue un valor escalar, el cual es posible optimizar para encontrar los parámetros que minimicen este valor.

Los detalles de como se construyeron las funciones objetivo para cada estructura se pueden observar en el Anexo C.

6.1.1. Variables de decisión

En este trabajo se presentan tres funciones objetivo (una para cada estructura), las cuales tienen una distinta cantidad de parámetros en función del número total de barras que tenga la estructura. En cada estructura, se optimizan los radios de las barras, teniendo dos parámetros principales: r_{marco} y R . Donde r_{marco} corresponde al radio de las barras de la estructura base (Figura 5.1), donde todas son iguales. Por otro lado, R corresponde a un vector que contiene todos los radios del enrejado interno de la estructura estudiada. De esta forma, considerando los parámetros con los que se construye cada marco (Tabla 5.1) y las Figuras 5.3, 5.6 y 5.9, es que la siguiente tabla muestra un resumen del número de variables de decisión que se tendrán para cada problema de optimización:

Tabla 6.1: Número de variables de decisión para cada estructura.

Estructura	Pratt	Baltimore	Mixta
Dimensión r_{marco}	1	1	1
Dimensión R	32	96	48
Número de variables de decisión	33	97	49

6.1.2. Restricciones

Como solo se trabajaron con parámetros de radios de barras, solo se tenían restricciones de su tamaño. Por lo tanto, de manera de no tener un problema con la relación **largo/diámetro** de las barras. Además, como las barras deben ser diseñadas para que puedan ser fabricadas mediante impresión 3D, los radios no deben ser tan pequeños para así evitar problemas con la impresión. Debido a esto, se definió un rango para los radios tal que:

$$0.5[mm] \leq r_i \leq 1.5[mm]; \forall i \quad (6.1)$$

6.1.3. Rangos de frecuencias

La función objetivo requiere ser evaluada en un rango de frecuencias no tan grande para tener un mejor resultado. De esta manera es que se escogen 3 rangos de frecuencia de un tamaño de 500 Hz (1500-2000, 2500-3000 y 3500-4000) para evaluar cada una de las funciones objetivo. A continuación se presentan los valores de la función objetivo en estos rangos para cada estructura base formulada en el capítulo 5 de formulación de estructuras:

Tabla 6.2: Valor RMS para estructuras bases antes de optimizar.

Rango	Pratt	Baltimore	Mixta
1500 - 2000 Hz	$2.679 \cdot 10^{-5}$	$2.611 \cdot 10^{-5}$	$3.098 \cdot 10^{-5}$
2500 - 3000 Hz	$2.429 \cdot 10^{-5}$	$1.329 \cdot 10^{-5}$	$1.340 \cdot 10^{-5}$
3500 - 4000 Hz	$7.192 \cdot 10^{-6}$	$8.929 \cdot 10^{-6}$	$7.973 \cdot 10^{-6}$

6.2. Configuración del algoritmo

Como se mencionó en la sección 2.6.2, además de los parámetros que recibe, se puede configurar la función para mejorar su desempeño. En nuestro problema, para cada estructura se utiliza el siguiente comando para configurar el algoritmo:

```
1 options = optimoptions('particleswarm', 'SwarmSize', 50, 'Display', 'iter', 'MaxIterations', 100, 'TolFun'  
    ↪ , 1e-3);
```

En estas configuraciones nos interesan los siguientes comandos:

- **SwarmSize:** Este parámetro especifica el tamaño que tendrá el enjambre, lo que quiere decir que se utilizarán 50 partículas en la optimización.
- **Display:** Este comando especifica la cantidad de información que se mostrará durante la optimización, en este caso se dice que se mostrará información en cada iteración del proceso.
- **MaxIterations:** Este comando indica el número máximo de iteraciones que el algoritmo realizará antes de detenerse, por lo que si el algoritmo alcanza 100 iteraciones se detendrá.
- **TolFun:** Este valor indica la tolerancia que tendrá la optimización, lo que significa que el algoritmo se detendrá si el cambio en la función objetivo es menor que 0.001.

Los códigos de optimización se pueden ver en más detalle en el Anexo E

6.3. Optimización estructura Pratt

- 1500 - 2000 Hz

En este rango de frecuencias la optimización arrojó que el radio de las barras del marco deben tener un valor de $r_{marco} = 1.3$ [mm], mientras que las otras 32 barras, correspondientes al enrejado se encuentran en el vector R tal que:

$$R = \begin{bmatrix} 1.1 & 0.7 & 1.3 & 0.6 & 0.8 & 0.7 & 1.5 & 1.5 \\ 0.8 & 0.6 & 0.8 & 1.3 & 1.5 & 1.4 & 0.6 & 0.9 \\ 1.0 & 0.5 & 1.4 & 1.4 & 1.5 & 0.5 & 0.5 & 0.5 \\ 0.8 & 0.6 & 1.5 & 0.8 & 0.5 & 1.5 & 0.5 & 1.3 \end{bmatrix} \text{ [mm]}$$

Con los parámetros óptimos encontrados, se puede evaluar la función objetivo para encontrar los valores y gráficas óptimas. De esta forma es que el valor óptimo de la RMS de los promedios de las funciones respuesta en frecuencia es $RMS = 9.7081 \cdot 10^{-7}$.

Por otro lado, en la Figura 6.1 se puede observar la FRF de la estructura Pratt optimizada en un rango de entre 1500 y 2000 Hz.

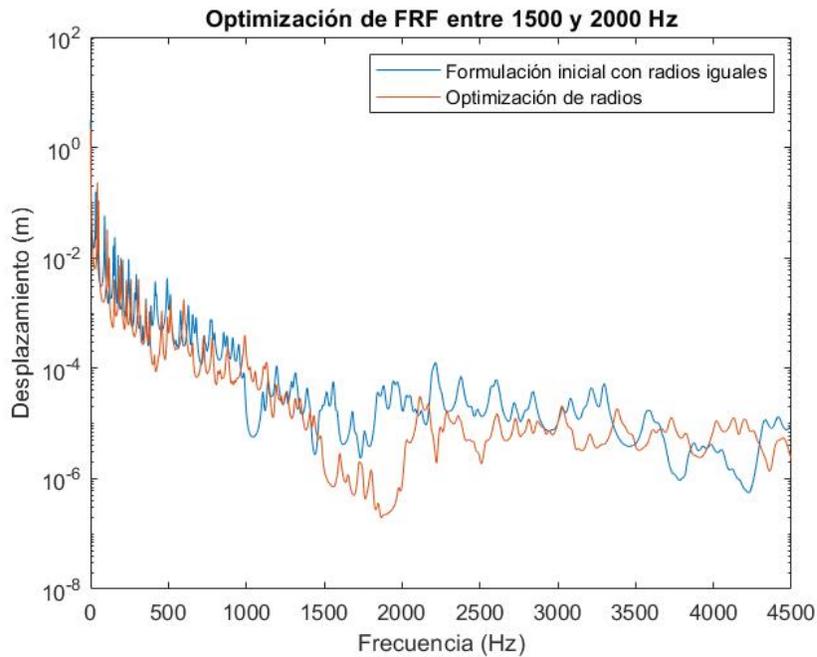


Figura 6.1: Función respuesta en frecuencia de estructura Pratt optimizada en un rango de frecuencias de 1500-2000 Hz.

- 2500 - 3000 Hz

En la optimización del segundo rango de frecuencias se obtuvo que el radio de las barras del marco deben tener un valor de $r_{marco} = 0.5$ [mm], mientras que las otras 32 barras del vector R asociado al enrejado, toma los valores siguientes:

$$R = \begin{bmatrix} 1.3 & 0.5 & 0.5 & 1.1 & 1.5 & 0.5 & 0.5 & 1.0 \\ 0.5 & 0.7 & 1.4 & 1.5 & 0.5 & 0.5 & 0.6 & 1.5 \\ 1.5 & 1.5 & 1.4 & 0.5 & 1.4 & 0.5 & 1.5 & 1.5 \\ 1.2 & 1.5 & 0.5 & 1.5 & 0.5 & 1.5 & 0.5 & 0.5 \end{bmatrix} \text{ [mm]}$$

Con estas nuevas variables de decisión encontradas, se evalúa la función objetivo para encontrar los valores y gráficas óptimas. De esta forma es que el valor óptimo de la RMS de los promedios de las funciones respuesta en frecuencia es $RMS = 3.6451 \cdot 10^{-8}$.

De la misma forma que antes, en la Figura 6.2 se puede observar la FRF de la estructura Pratt optimizada en un rango de entre 2500 y 3000 Hz.

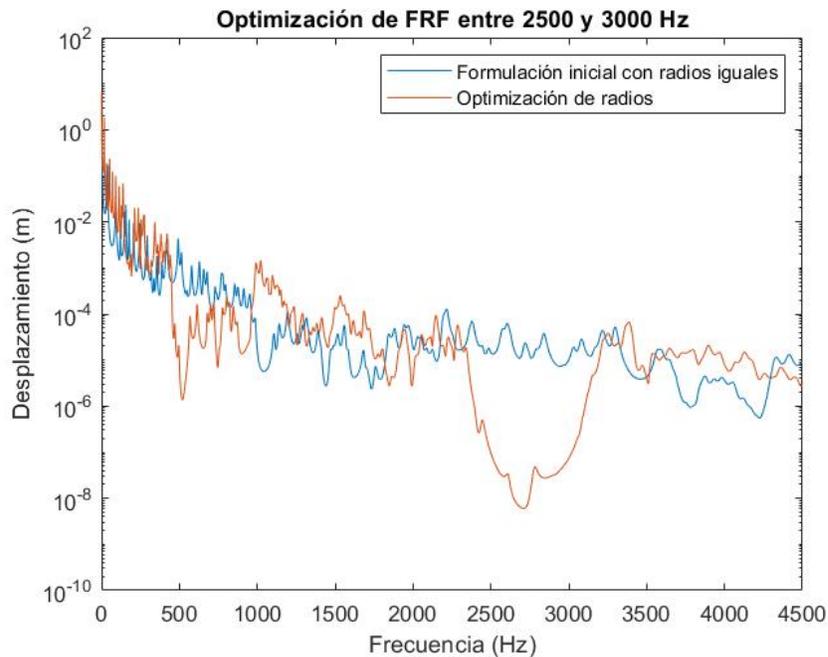


Figura 6.2: Función respuesta en frecuencia de estructura Pratt optimizada en un rango de frecuencias de 2500-3000 Hz.

- **3500 - 4000 Hz**

En el último rango la optimización arrojó que el radio de las barras del marco deben tener un valor de $r_{marco} = 0.7$ [mm], mientras que las otras 32 barras del enrejado toman los siguientes valores del vector R :

$$R = \begin{bmatrix} 1.5 & 1.4 & 1.3 & 0.5 & 1.2 & 0.7 & 1.5 & 1.3 \\ 0.5 & 1.4 & 0.5 & 1.5 & 1.3 & 1.0 & 0.6 & 0.7 \\ 1.4 & 0.6 & 1.0 & 0.7 & 1.2 & 0.5 & 1.1 & 0.6 \\ 0.7 & 1.2 & 0.7 & 1.0 & 0.5 & 1.3 & 1.5 & 1.4 \end{bmatrix} \text{ [mm]}$$

Con los valores de los radios encontrados, se puede obtener el valor mínimo de la función objetivo obtener su gráfica. De esta forma es que el valor óptimo de la RMS de los promedios de las funciones respuesta en frecuencia es $RMS = 9.4516 \cdot 10^{-8}$.

De manera similar a los otros rangos, en la Figura 6.3 se puede observar la FRF de la estructura Pratt optimizada en el último rango de frecuencias entre 3500 y 4000 Hz.

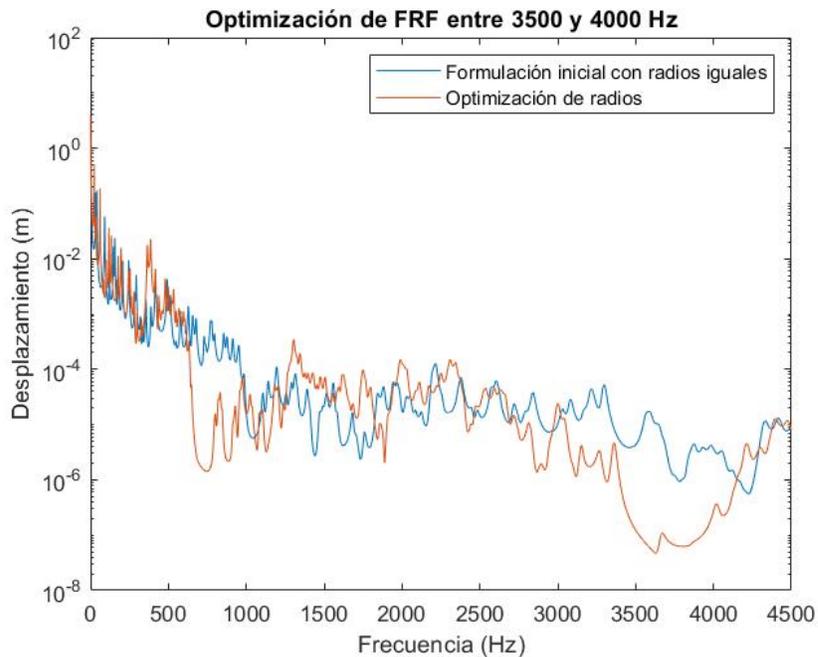


Figura 6.3: Función respuesta en frecuencia de estructura Pratt optimizada en un rango de frecuencias de 3500-4000 Hz.

6.4. Optimización estructura Baltimore

- 1500 - 2000 Hz

Para este primer rango de frecuencias, se encontró que la estructura Baltimore óptima cuenta con $r_{marco} = 1.2$ [mm], mientras que los radios asociados al enrejado interno toman los valores que se encuentran en el vector R :

$$R = \begin{bmatrix} 0.8 & 1.4 & 1.1 & 0.7 & 1.0 & 1.5 & 0.8 & 0.6 \\ 0.9 & 1.1 & 1.3 & 0.5 & 1.5 & 0.8 & 0.7 & 0.7 \\ 1.2 & 1.1 & 1.5 & 0.5 & 1.1 & 0.7 & 0.7 & 1.2 \\ 1.5 & 0.6 & 1.1 & 0.8 & 1.4 & 1.5 & 0.5 & 1.4 \\ 1.2 & 0.5 & 1.1 & 1.0 & 1.2 & 1.5 & 0.6 & 1.4 \\ 1.2 & 1.0 & 0.8 & 1.1 & 0.8 & 0.8 & 1.3 & 1.3 \\ 0.8 & 0.5 & 0.8 & 1.0 & 0.7 & 1.3 & 0.5 & 1.1 \\ 0.7 & 0.8 & 0.8 & 0.5 & 1.1 & 1.4 & 1.2 & 1.5 \\ 0.6 & 0.5 & 0.7 & 1.0 & 0.8 & 1.3 & 1.3 & 1.4 \\ 0.5 & 1.5 & 0.6 & 0.5 & 1.1 & 0.8 & 1.4 & 1.4 \\ 1.4 & 1.2 & 0.5 & 1.3 & 1.4 & 0.9 & 0.6 & 0.9 \\ 0.7 & 1.2 & 1.1 & 0.5 & 0.5 & 1.4 & 1.0 & 1.2 \end{bmatrix} \text{ [mm]}$$

Se evalúa la función objetivo y se encuentra que su valor mínimo es $RMS = 1.1559 \cdot 10^{-6}$.

Ahora, en la Figura 6.4 se aprecia la FRF de la estructura Baltimore optimizada en el rango de frecuencias de 1500 a 2000 Hz.

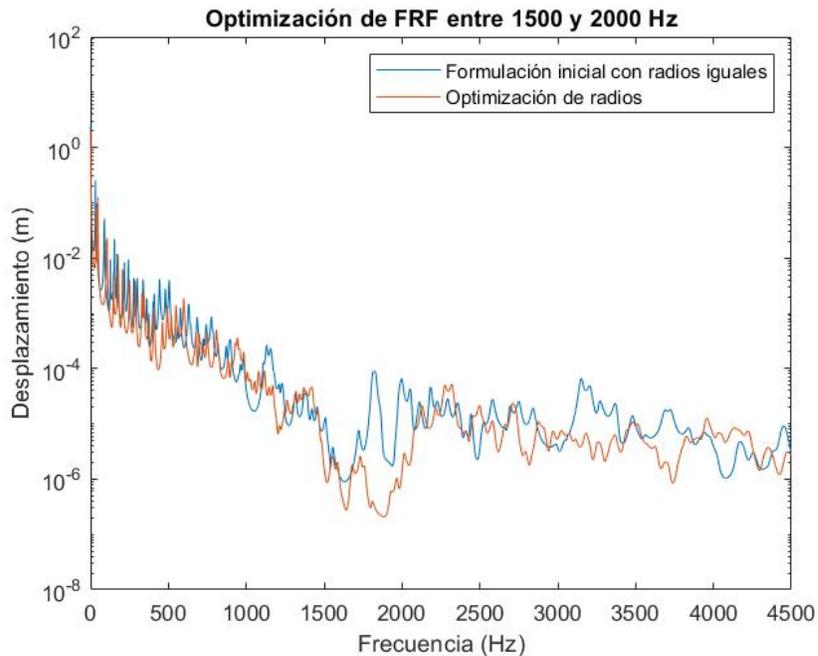


Figura 6.4: Función respuesta en frecuencia de estructura Baltimore optimizada en un rango de frecuencias de 1500-2000 Hz.

• 2500 - 3000 Hz

Para el segundo rango de frecuencias, se encontró que la estructura Baltimore óptima presenta un marco con radio $r_{marco} = 0.5$ [mm], mientras que los radios de las barras del enrejado interno toman los valores guardados en el vector R :

$$R = \begin{bmatrix} 1.0 & 0.9 & 1.1 & 1.4 & 1.3 & 0.9 & 0.7 & 0.5 \\ 1.2 & 0.7 & 0.5 & 1.0 & 0.5 & 0.7 & 0.6 & 0.7 \\ 1.0 & 0.5 & 1.0 & 0.9 & 0.8 & 0.8 & 1.4 & 1.3 \\ 1.4 & 1.1 & 1.2 & 1.4 & 1.2 & 1.2 & 0.9 & 1.3 \\ 1.5 & 0.7 & 0.6 & 1.4 & 1.0 & 1.4 & 1.2 & 0.9 \\ 1.0 & 0.5 & 1.5 & 0.8 & 0.5 & 0.6 & 1.2 & 1.3 \\ 1.5 & 0.8 & 0.9 & 1.4 & 1.2 & 1.2 & 1.2 & 1.0 \\ 1.4 & 1.5 & 1.3 & 0.7 & 0.8 & 0.8 & 0.8 & 0.7 \\ 1.5 & 0.6 & 1.3 & 1.1 & 1.2 & 0.6 & 0.6 & 1.2 \\ 0.6 & 1.1 & 1.0 & 1.0 & 0.5 & 1.4 & 1.2 & 0.8 \\ 1.5 & 0.6 & 1.0 & 0.8 & 0.9 & 1.0 & 1.0 & 1.5 \\ 1.5 & 1.5 & 1.1 & 0.6 & 1.0 & 1.3 & 1.4 & 1.5 \end{bmatrix} \text{ [mm]}$$

Conociendo los valores de r_{marco} y R , se evalúa la función objetivo y se encuentra su valor mínimo como $RMS = 3.1649 \cdot 10^{-8}$.

Por otro lado, en la Figura 6.5 se puede observar la FRF de la estructura Baltimore optimizada en un rango de frecuencias de 2500-3000 Hz.

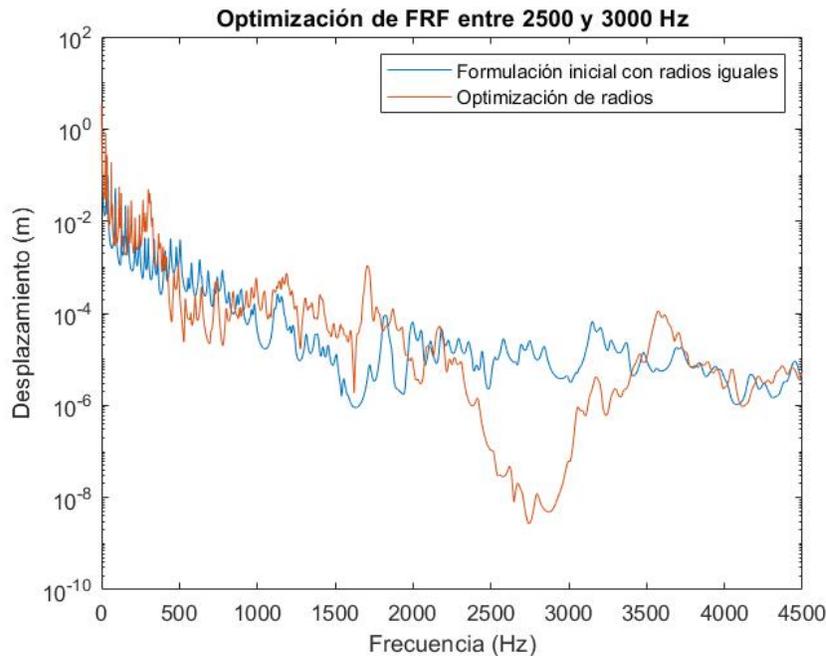


Figura 6.5: Función respuesta en frecuencia de estructura Baltimore optimizada en un rango de frecuencias de 2500-3000 Hz.

• 3500 - 4000 Hz

En este último rango se obtuvo que el marco debe presentar un radio $r_{marco} = 0.7$ [mm], mientras que los radios de las barras del enrejado interno deben tomar valores como los mostrados en el vector R :

$$R = \begin{bmatrix} 1.5 & 0.5 & 0.5 & 1.4 & 0.5 & 1.5 & 0.7 & 1.5 \\ 0.5 & 1.5 & 0.5 & 0.5 & 0.5 & 1.2 & 1.5 & 1.5 \\ 1.5 & 0.5 & 1.5 & 0.5 & 0.8 & 0.5 & 1.5 & 1.3 \\ 1.5 & 1.2 & 1.5 & 1.5 & 1.5 & 0.5 & 1.5 & 1.4 \\ 0.5 & 1.5 & 1.0 & 1.5 & 1.3 & 0.5 & 0.8 & 1.5 \\ 0.8 & 0.5 & 0.9 & 1.3 & 0.5 & 1.5 & 1.2 & 1.5 \\ 1.5 & 0.5 & 0.6 & 0.7 & 1.5 & 1.4 & 1.5 & 0.6 \\ 0.9 & 0.6 & 0.5 & 0.5 & 1.5 & 1.3 & 1.0 & 1.5 \\ 0.5 & 0.9 & 0.5 & 0.5 & 0.5 & 1.5 & 1.5 & 1.5 \\ 1.2 & 1.5 & 1.5 & 1.5 & 0.5 & 1.3 & 1.5 & 1.0 \\ 0.5 & 0.6 & 0.5 & 0.5 & 1.0 & 1.4 & 1.5 & 1.5 \\ 1.1 & 0.5 & 1.4 & 0.5 & 1.5 & 1.1 & 0.6 & 0.5 \end{bmatrix} \text{ [mm]}$$

Con los valores de r_{marco} y R obtenidos, se obtiene el mínimo de la función objetivo como $RMS = 3.3418 \cdot 10^{-8}$.

Por último, en la Figura 6.6 se observa la FRF de la estructura Mixta al ser optimizada en un rango de frecuencias de entre 3500 y 4000 Hz.

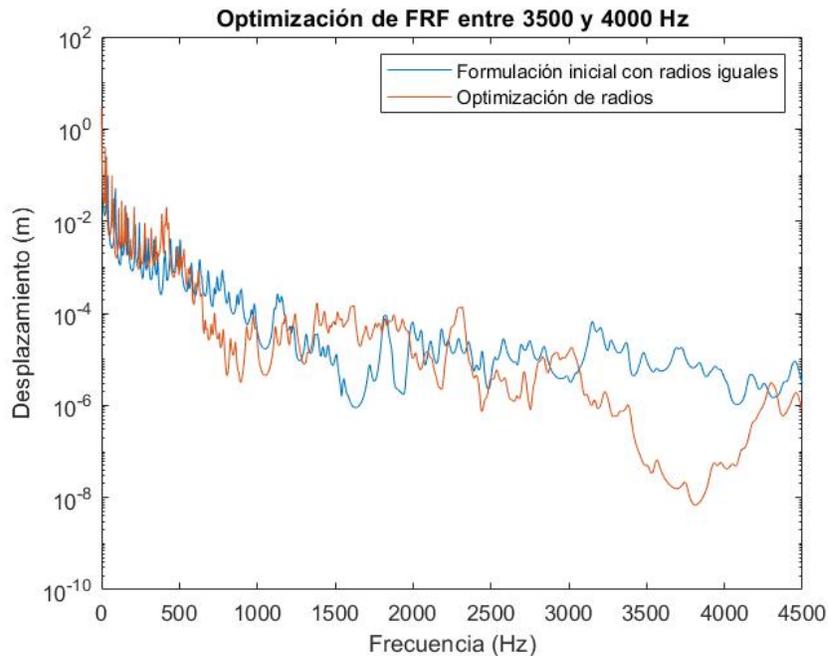


Figura 6.6: Función respuesta en frecuencia de estructura Baltimore optimizada en un rango de frecuencias de 3500-4000 Hz.

6.5. Optimización estructura Mixta

- 1500 - 2000 Hz

En el primer rango de frecuencias, se encontró una estructura óptima que cuenta con un radio de marco $r_{marco} = 1.5$ [mm]. Por otro lado, los radios asociados al enrejado interno toman los valores que se encuentran en el vector R :

$$R = \begin{bmatrix} 1.3 & 0.5 & 0.5 & 0.5 & 0.9 & 0.7 & 0.7 & 1.5 \\ 0.9 & 0.6 & 1.5 & 0.5 & 1.5 & 0.6 & 0.5 & 1.2 \\ 0.7 & 1.0 & 1.1 & 1.2 & 1.5 & 1.4 & 0.8 & 1.5 \\ 0.5 & 1.5 & 0.9 & 1.3 & 1.5 & 0.5 & 1.5 & 0.9 \\ 1.5 & 1.5 & 0.5 & 0.5 & 1.0 & 0.6 & 1.5 & 0.5 \\ 0.7 & 1.5 & 1.2 & 0.5 & 0.5 & 0.5 & 1.5 & 1.5 \end{bmatrix} \text{ [mm]}$$

Con los radios conocidos, se evalúa la función objetivo, dando como resultado que el mínimo es $RMS = 3.1689 \cdot 10^{-6}$.

En la Figura 6.7 se puede observar la FRF de la estructura Mixta optimizada en un rango de entre 1500 y 2000 Hz.

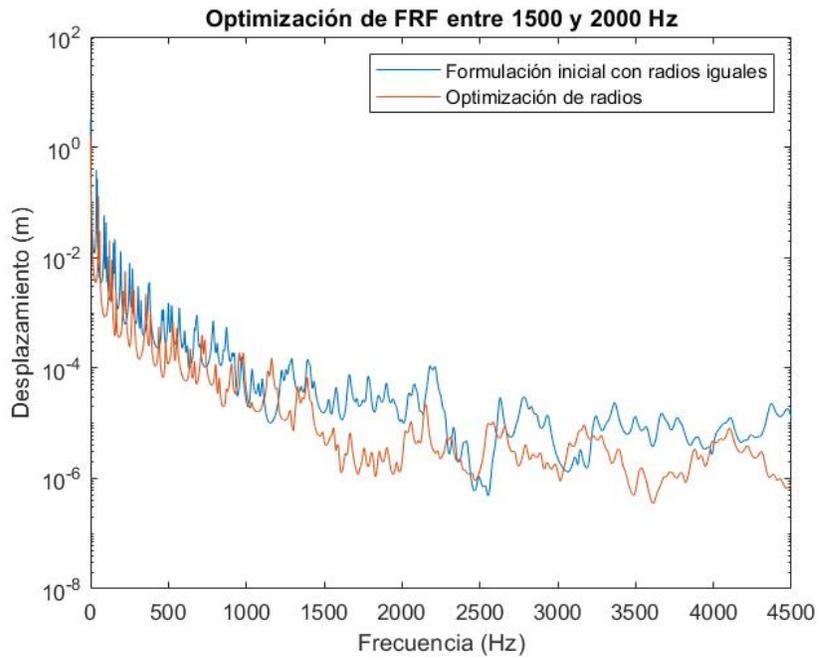


Figura 6.7: Función respuesta en frecuencia de estructura Mixta optimizada en un rango de frecuencias de 1500-2000 Hz.

• 2500 - 3000 Hz

Para el segundo rango de frecuencias, se encontró que la estructura óptima presenta un marco con radio $r_{marco} = 0.5$ [mm], mientras que los radios de las barras del enrejado interno toman los valores guardados en el vector R :

$$R = \begin{bmatrix} 1.5 & 0.6 & 1.5 & 1.5 & 1.5 & 1.2 & 1.5 & 0.5 \\ 0.9 & 0.5 & 1.5 & 1.5 & 1.3 & 0.5 & 0.8 & 1.5 \\ 1.5 & 1.5 & 1.2 & 0.7 & 1.5 & 0.7 & 1.5 & 1.5 \\ 0.5 & 0.5 & 1.0 & 0.7 & 1.5 & 0.5 & 1.4 & 1.5 \\ 0.6 & 1.1 & 0.5 & 0.5 & 0.5 & 0.5 & 1.5 & 1.1 \\ 1.5 & 1.3 & 1.4 & 1.5 & 1.5 & 0.5 & 0.5 & 0.5 \end{bmatrix} \text{ [mm]}$$

Con los valores de los radios conocidos, se puede evaluar la función objetivo, arrojando que el mínimo corresponde a $RMS = 2.9036 \cdot 10^{-8}$.

De manera similar al caso anterior, en la Figura 6.8 se observa la FRF de la estructura Mixta optimizada en un rango de entre 2500 y 3000 Hz.

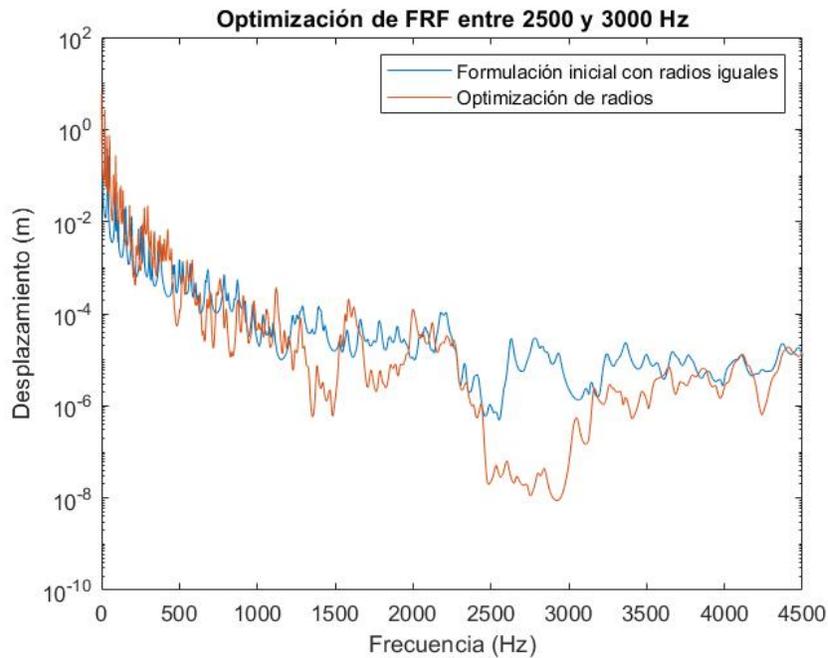


Figura 6.8: Función respuesta en frecuencia de estructura Mixta optimizada en un rango de frecuencias de 2500-3000 Hz.

• 3500 - 4000 Hz

Para el último rango de frecuencias, se obtuvo que la estructura óptima tiene un radio del marco $r_{marco} = 0.7$ [mm], mientras que las barras del enrejado interno tienen radios guardados en el vector R :

$$R = \begin{bmatrix} 0.6 & 1.5 & 0.6 & 1.3 & 1.4 & 0.6 & 1.5 & 0.5 \\ 0.5 & 1.4 & 0.5 & 1.0 & 1.3 & 0.9 & 1.1 & 1.2 \\ 0.5 & 1.5 & 1.4 & 1.1 & 0.9 & 1.1 & 1.5 & 0.6 \\ 1.5 & 1.2 & 1.3 & 0.5 & 1.3 & 1.5 & 1.3 & 1.5 \\ 1.1 & 0.9 & 0.7 & 1.0 & 1.2 & 1.5 & 1.5 & 1.4 \\ 1.1 & 1.0 & 0.8 & 0.8 & 1.0 & 1.5 & 0.7 & 1.5 \end{bmatrix} \text{ [mm]}$$

Con los valores de r_{marco} y R conocidos, se procede a evaluar la función objetivo, obteniendo que el mínimo es tal que, $RMS = 1.6113 \cdot 10^{-7}$.

Por último, en la Figura 6.9 se aprecia la FRF de la estructura Mixta optimizada en un rango de entre 3500 y 4000 Hz.

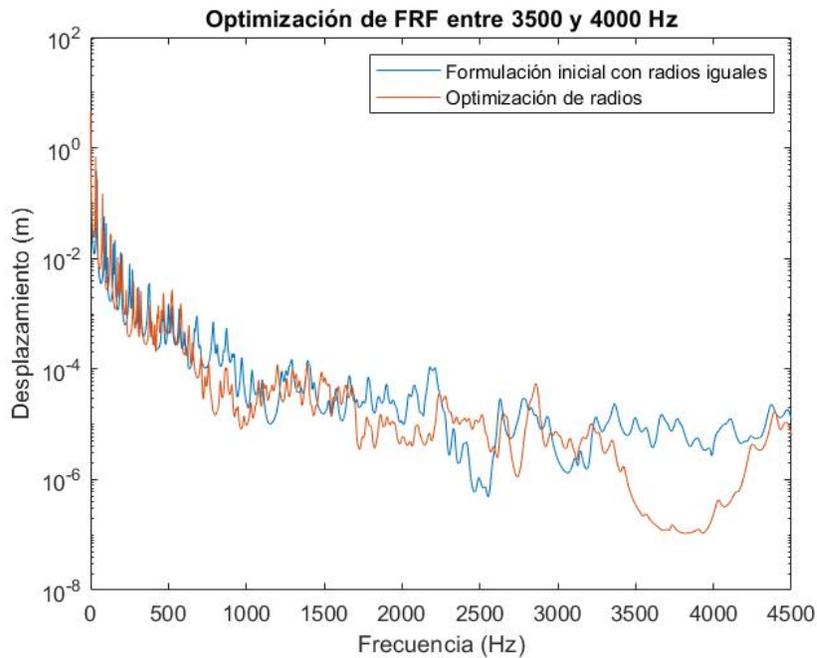


Figura 6.9: Función respuesta en frecuencia de estructura Mixta optimizada en un rango de frecuencias de 3500-4000 Hz.

Los detalles sobre los resultados de las iteraciones se pueden encontrar en el Anexo F

Capítulo 7

Análisis y discusión de resultados

Tras presentar los resultados de la optimización de cada estructura en el capítulo anterior, es fundamental llevar a cabo un análisis detallado para extraer conclusiones significativas sobre la reducción de vibraciones en las estructuras.

En primer lugar, es importante señalar que la optimización se realizó dentro de los límites impuestos por las restricciones de tamaño de los radios de las barras, que oscilan entre 0.5 y 1.5 mm. Es crucial tener en cuenta que, si bien se lograron mejoras significativas dentro de este rango, existen posibles soluciones aún más efectivas fuera de este límite.

En segundo lugar, tenemos los valores mínimos de la función objetivo que buscamos optimizar. Como este valor es un indicativo de la reducción en la propagación de vibraciones en cada estructura es de suma importancia analizarlo. Todos los valores de las RMS se encuentran resumidas en la siguiente tabla:

Tabla 7.1: Resumen de valores RMS para cada estructura.

Rango	Pratt	Baltimore	Mixta
1500 - 2000 Hz	$9.708 \cdot 10^{-7}$	$1.156 \cdot 10^{-6}$	$3.169 \cdot 10^{-6}$
2500 - 3000 Hz	$3.645 \cdot 10^{-8}$	$3.165 \cdot 10^{-8}$	$2.904 \cdot 10^{-8}$
3500 - 4000 Hz	$9.452 \cdot 10^{-8}$	$3.342 \cdot 10^{-8}$	$1.611 \cdot 10^{-7}$
Promedio	$3.673 \cdot 10^{-7}$	$4.070 \cdot 10^{-7}$	$1.120 \cdot 10^{-6}$

En la tabla 7.1 se puede observar que, con la optimización, se redujo de manera significativa el valor de las RMS con respecto a su valor original (aproximadamente 2-3 ordenes de magnitud, ver Tabla 6.2). Se aprecia que, para el primer rango de frecuencias (1500-2000 Hz), la estructura Pratt presenta mejores resultados, obteniendo una diferencia de un orden de magnitud con respecto a las otras opciones. Sin embargo, esto no se mantiene del todo en los siguientes rangos, ya que, en el segundo rango de frecuencias (2500-3000 Hz), si bien las estructuras presentan resultados bastante similares, la estructura Mixta es la que saca una pequeña ventaja con respecto a las demás opciones. Por último, en el rango de frecuencias final (3500 - 4000 Hz), la estructura Baltimore es la que presenta mejores resultados, logrando un valor considerablemente menor con respecto a las otras 2 configuraciones. Como conclusión general, se puede observar que la estructura Pratt es la que en promedio obtiene mejores resultados considerando los tres rangos estudiados.

Ahora, al analizar las FRF de cada estructura, se pueden obtener resultados interesantes. Para la estructura Pratt, se pueden observar en las Figuras 6.1, 6.2 y 6.3 que cada una de las curvas optimizadas presenta un decaimiento significativo en el rango de optimización correspondiente. En todos los resultados, la curva optimizada es bastante similar a la original; sin embargo, en el momento en que las frecuencias toman valores correspondientes al rango estudiado, la curva sufre un decaimiento que se mantiene a lo largo del rango, generando una disminución en la propagación de las vibraciones en la estructura en ese rango específico. Además, si hablamos de los tramos de optimización, en el primer rango, la curva es similar a la original solo que desplazada, mientras que en los otros rangos, la curva disminuye significativamente y se aplana. Para ser más precisos, se pueden observar las siguientes imágenes:

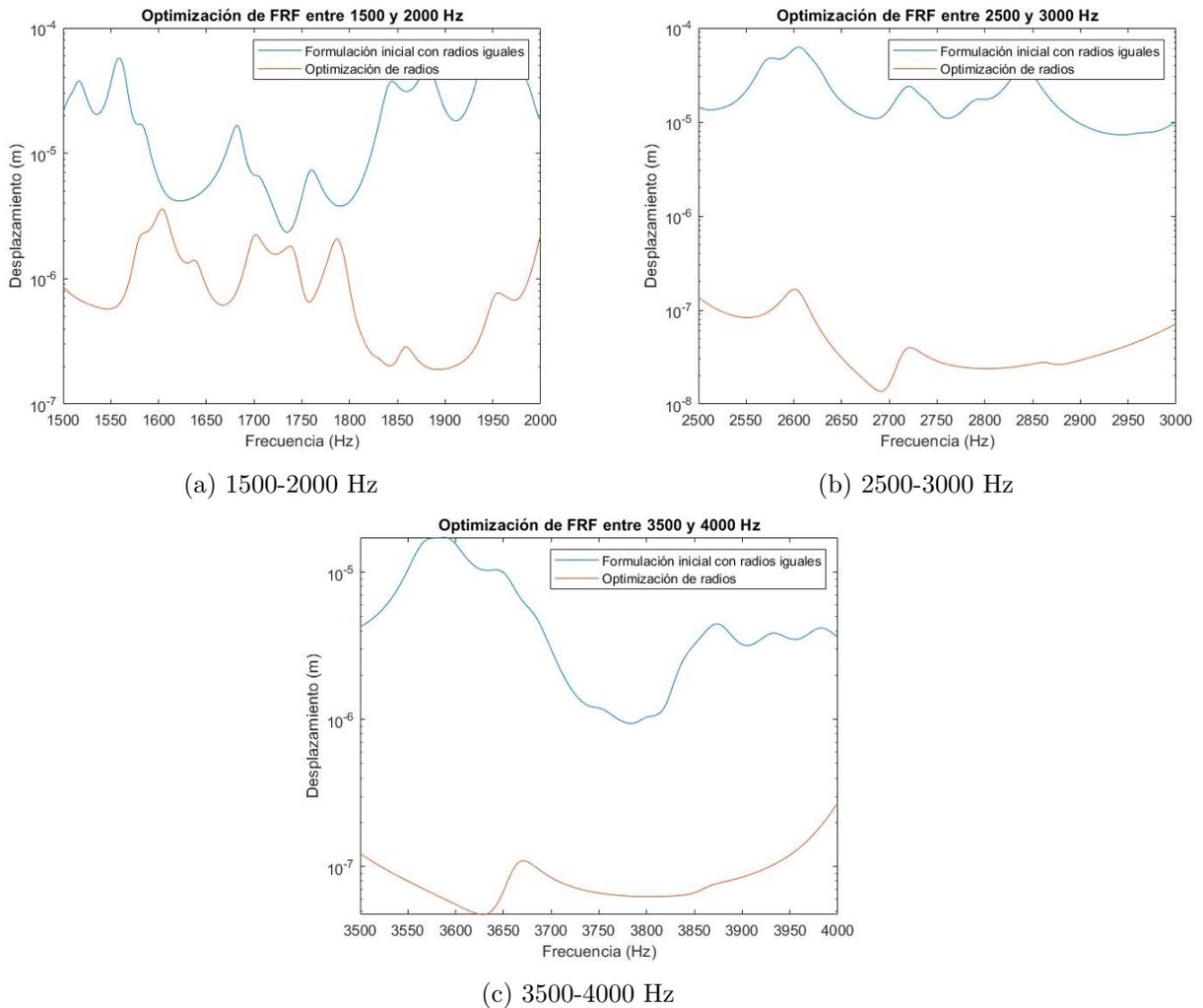


Figura 7.1: Zoom de la FRF de la estructura Pratt optimizada en cada rango de frecuencias.

En la Figura 7.1 se aprecia cómo, en la optimización de la estructura Pratt, la curva se desplaza hacia abajo y trata de mantenerse constante hasta el final del rango (a excepción de la Figura 7.1.a). Esto confirma que se logra una disminución de la propagación de vibraciones en la estructura Pratt en los rangos estudiados, logrando valores bastante bajos.

Continuando con un análisis similar, la estructura Baltimore presenta curvas algo parecidas a la estructura Pratt pero en una escala diferente. Si observamos las Figuras 6.4, 6.5 y 6.6, se puede apreciar que en el primer rango de frecuencias, la curva optimizada en momentos presenta mayores valores de respuesta que la curva original, lo cual genera algo de ruido. Este comportamiento de la curva ocurre también en el caso anterior, pero debido a que el desplazamiento inicial de la curva es mayor, no ocurre este problema en la estructura Pratt. Por otro lado, en los siguientes rangos ocurre algo similar a los casos anteriores, pues se presenta un decaimiento que se mantiene relativamente constante durante todo el tramo, para luego, al terminar el recorrido del tramo, la respuesta vuelve a subir. Para ver en detalle la respuesta de las curvas en estos rangos, es necesario estudiar las siguientes imágenes:

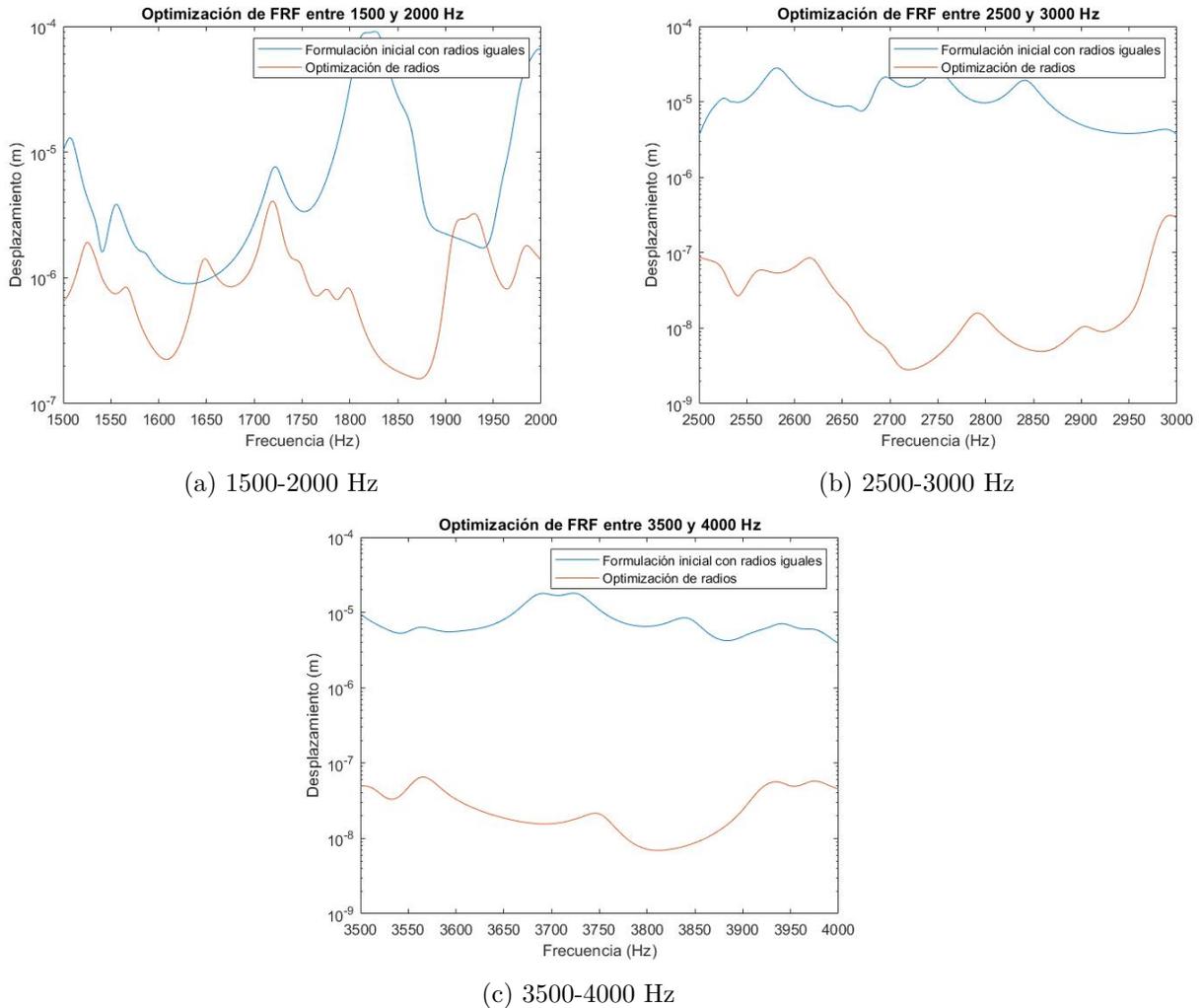


Figura 7.2: Zoom de la FRF de la estructura Baltimore optimizada en cada rango de frecuencias.

En la Figura 7.2, se aprecia que en el primer rango de frecuencias, las curvas presentan intersecciones debido a que no se logra una estabilización de la curva optimizada. Por otro lado, en los otros rangos, la respuesta se estabiliza, evitando el problema anterior. Cabe destacar que todas las curvas optimizadas presentan respuestas en promedio menor que la original, por lo que la optimización logra el objetivo de reducir las vibraciones en la estructura.

Finalmente, tenemos la estructura Mixta, la cual también presenta su propio comportamiento. Al observar las Figuras 6.7, 6.8 y 6.9, se aprecia que en los rangos estudiados ocurren decaimientos significativos que vuelven a subir una vez superado ese rango, lo que es bastante similar a lo ocurrido con las estructuras anteriores. Si analizamos el primer rango de estudio, se observa que presenta el mismo comportamiento que en los casos anteriores, con una sola diferencia: si bien presenta subidas y bajadas, el valor máximo de la respuesta en este rango no se encuentra cercano al valor mínimo de la respuesta en la curva original, lo cual es un buen resultado. Por otro lado, al analizar los otros rangos, se observa el mismo comportamiento que en las otras configuraciones, donde la curva decae significativamente y se mantiene relativamente constante para luego volver a subir al finalizar el tramo. Esto se puede ver a continuación, en el detalle de las curvas de los rangos estudiados:

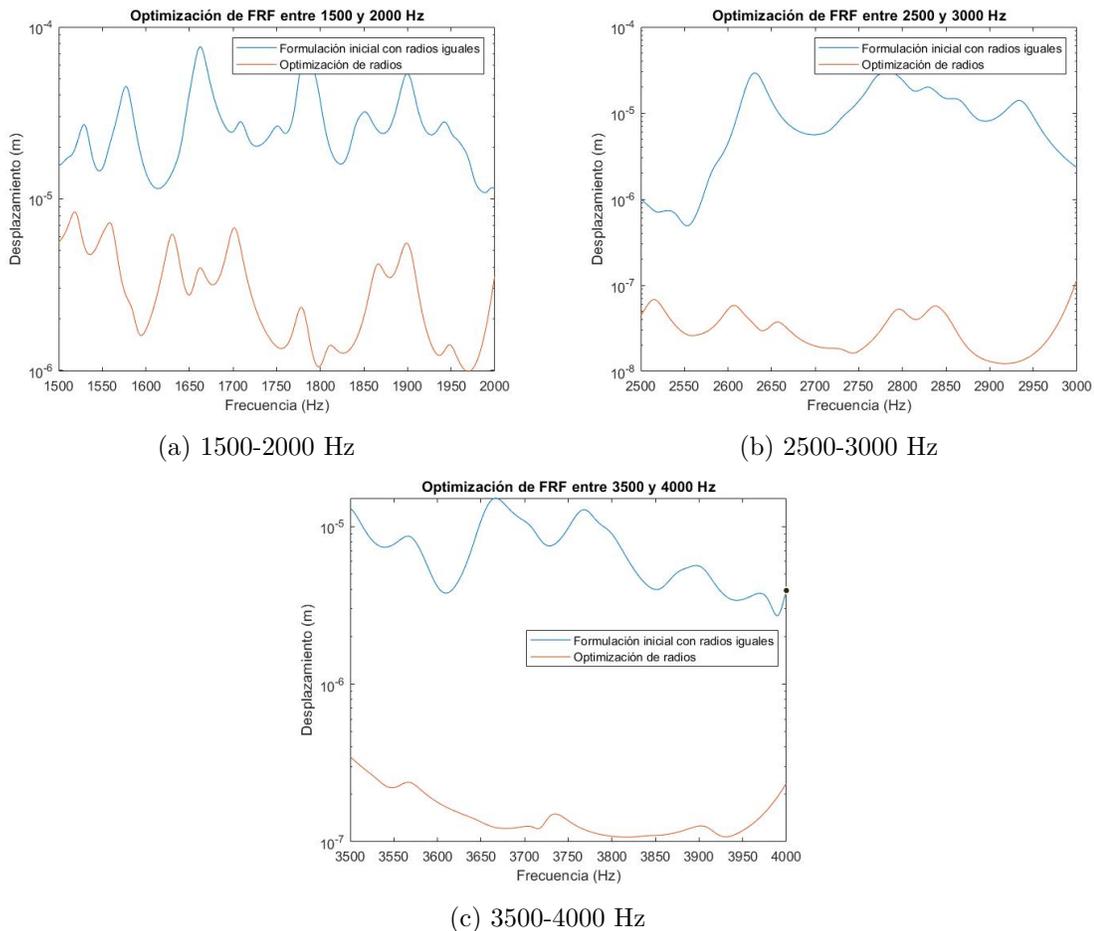


Figura 7.3: Zoom de la FRF de la estructura Mixta optimizada en cada rango de frecuencias.

En la Figura 7.3 se aprecia cómo la optimización de la estructura Mixta logró una baja considerable en la respuesta. En comparación con las demás configuraciones, se puede decir que la curva optimizada es menor en todo momento que la curva original, logrando un decaimiento al inicio del rango que se mantiene a lo largo del tramo, para luego volver a subir. Esto nos indica, al igual que en las otras estructuras, que la optimización logró su cometido de disminuir la propagación de las vibraciones en los rangos deseados.

Capítulo 8

Conclusiones

A modo de conclusión, se ha logrado alcanzar el objetivo principal de desarrollar una metodología para generar diseños óptimos de estructuras reticulares en paneles celulares que reduzcan la propagación de vibraciones en su estructura. Esta metodología tiene un potencial significativo en el ámbito de la investigación de supresión de vibraciones, ya que sirve como base para el diseño de estructuras más nuevas y complejas.

Además, se llevaron a cabo diversos diseños numéricos de estructuras reticulares utilizando la herramienta de elementos finitos, lo que permitió formular códigos para generar diferentes modelos de estructuras en función de la repetición de celdas unitarias (simples y complejas). Este proceso se apoyó en un estudio previo que validó la coherencia de los resultados con la realidad.

Se propusieron tres tipos diferentes de configuraciones de estructuras (Pratt, Baltimore y Mixta), las cuales se diseñaron con el fin de que sean factibles de fabricar mediante impresión 3D (cuidando el tamaño de impresión y las dimensiones de las estructuras para no tener problemas), lo que acercó el problema a la realidad y facilitó la generación de modelos que cumplieran con el objetivo principal de reducir las vibraciones y que fueran viables para su fabricación utilizando impresión 3D.

La aplicación del algoritmo de optimización (particleswarm) permitió encontrar soluciones óptimas para las configuraciones dentro de ciertos rangos de frecuencia, lo que generó diversas conclusiones sobre qué estructura puede ser mejor para diferentes escenarios. Además, esta aplicación permitió familiarizarse con algoritmos de optimización en Matlab, lo que abre la puerta a la posibilidad de probar diferentes algoritmos en busca de mejores resultados.

Con los resultados numéricos y gráficos, fue posible evaluar las configuraciones y compararlas entre sí y extraer conclusiones sobre qué tipo de estructuras responden mejor a diferentes rangos de frecuencias y cuál presenta mejores resultados en promedio. Sin embargo, es importante destacar que estas conclusiones solo son aplicables a los rangos estudiados.

Con respecto a este último punto, como se vio a lo largo del trabajo, cada una de las estructuras respondió de buena manera a lo solicitado, que era reducir la respuesta en un cierto rango de frecuencias, aunque algunas obtuvieron mejores resultados que otras en distintos rangos en los que fueron evaluadas.

Centrándonos en los datos arrojados por la optimización, si se quiere obtener una estructura que sea mejor que las demás, se deben evaluar varios rangos de frecuencias y encontrar una forma adecuada para decidir cuál responde mejor, ya que se observa que todas las estructuras responden de buena manera a distintos rangos de frecuencias. La estructura Pratt es la que mejor responde en un rango de frecuencia entre 1500 y 2000 Hz, mientras que la estructura Mixta es la que mejor responde en los rangos de 1500-2000, dejando a la estructura Baltimore como la mejor evaluada para un rango de 3500-4000 Hz.

Las gráficas demostraron que cada estructura se centra completamente en reducir las vibraciones en el rango correspondiente, generando resultados más consistentes a frecuencias más altas que en frecuencias bajas, donde la reducción no es tanta con respecto a la respuesta original.

Es importante mencionar que, aunque el trabajo cumple con los objetivos establecidos inicialmente, hay margen para mejoras en varios aspectos. Por ejemplo, se podrían probar distintos rangos de frecuencia, quizás más en cantidad y más extensos o acotados según los requisitos específicos, o experimentar con nuevos algoritmos de optimización para comparar sus resultados. También sería interesante explorar nuevas configuraciones, más complejas o más simples, variando su tamaño sin tantas restricciones como las impuestas en este trabajo, donde la factibilidad de fabricación mediante impresión 3D era un requisito. Todos estos aspectos son posibles de variar y buscar mejores resultados con la posibilidad de encontrar rangos donde la respuesta disminuya aún más y en consecuencia, encontrar soluciones mejores a las encontradas.

A pesar de estas posibles mejoras, el trabajo logra cumplir con cada uno de los objetivos planteados inicialmente, proporcionando una metodología de trabajo capaz de encontrar diseños óptimos para reducir vibraciones en estructuras reticulares. Gracias a este trabajo, se cuenta con una base sólida para futuras propuestas de estructuras reticulares diferentes y más complejas que requieran ser estudiadas.

Bibliografía

- [1] Gibson, L. J. (2003). Cellular Solids. *MRS Bulletin*, 28(4), 270–274. <https://doi.org/10.1557/mrs2003.79>
- [2] Quinteros, L., Meruane, V., Lenz Cardoso, E., & Ruiz, R. O. (2021). Phononic Bandgap Optimization in Sandwich Panels Using Cellular Truss Cores. *Materials*, 14(18), 5236. <https://doi.org/10.3390/ma14185236>
- [3] Yeo, S. J., Oh, M. J., & Yoo, P. J. (2018). Structurally Controlled Cellular Architectures for High-Performance Ultra-Lightweight Materials. *Advanced Materials*, 31(34), 1803670. <https://doi.org/10.1002/adma.201803670>
- [4] Schaedler, T. A., & Carter, W. B. (2016). Architected Cellular Materials. *Annual Review of Materials Research*, 46(1), 187–210. <https://doi.org/10.1146/annurev-matsci-070115-031624>
- [5] Ruopeng Liu, Chunlin Ji, Zhiya Zhao, Tian Zhou. (2015). Research: Advanced Materials and Materials Genome—Review, Metamaterials: Reshape and Rethink. core.ac.uk. <chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://core.ac.uk/download/pdf/82400065.pdf>
- [6] Sun, H., Du, X., & Pai, P. F. (2010). Theory of Metamaterial Beams for Broadband Vibration Absorption. *Journal of Intelligent Material Systems and Structures*, 21(11), 1085–1101. <https://doi.org/10.1177/1045389x10375637>
- [7] McCormac, J. C., & Elling, R. E. (2000). *Análisis de Estructuras - Métodos Clásicos y Matri*. Alfaomega Grupo Editor.
- [8] Guo, J., Li, Y., Xiao, Y., Fan, Y., Yu, D., & Wen, J. (2022). Multiscale modeling and design of lattice truss core sandwich metastructures for broadband low-frequency vibration reduction. *Composite Structures*, 289, 115463. <https://doi.org/10.1016/j.compstruct.2022.115463>
- [9] An, X., Fan, H., & Zhang, C. (2020). Elastic wave and vibration bandgaps in planar square metamaterial-based lattice structures. *Journal of Sound and Vibration*, 475, 115292. <https://doi.org/10.1016/j.jsv.2020.115292>

- [10] Meruane, V. (s.f.). Dinámica estructural. Universidad de Chile.
- [11] Ortiz, A. (2022). Método del Elemento Finito 1D. En Métodos Avanzados en Mecánica de Sólidos Computacional. CAMLAB. <https://camlab.cl/teaching/books/mcms/>
- [12] Gálvez, V. (2021), Desarrollo de modelos en elementos finitos parametrizados de paneles tipo sandwich con núcleos ultralivianos para predecir band gaps [Memoria para titulación Ingeniería Mecánica, Universidad de Chile]
- [13] Sánchez, C. S. (2012). Optimización estructural y topológica de estructuras morfológicamente no definidas mediante algoritmos genéticos [Doctoral thesis, Editorial Universitat Politècnica de València]. <http://hdl.handle.net/10251/15409>
- [14] Structural Dynamics Toolbox – Structural Dynamics Toolbox & FEMLink. (s. f.). <https://www.sdtools.com/sdt/sdt/>
- [15] Estadística, P. Y. (2023, 22 diciembre). Media cuadrática. Probabilidad y Estadística. <https://www.probabilidadyestadistica.net/media-cuadratica/>
- [16] Particle swarm optimization - MATLAB particleswarm - MathWorks América Latina. (s. f.). <https://la.mathworks.com/help/gads/particleswarm.html>
- [17] Quinteros, L. (2020), Optimización de band-gap en materiales tipo sandwich [Memoria para titulación Ingeniería Mecánica, Universidad de Chile]
- [18] Cubillos, J. G. (2020), Diseño y optimización de vigas de metamateriales con enrejado quiral para la supresión de vibraciones, utilizando modelo de elementos finitos y algoritmo genéticos [Memoria para titulación Ingeniería Mecánica, Universidad de Chile]

Anexos

Anexo A. Código en Matlab para la verificación

```
1     path(path,'C:\Program Files\MATLAB\R2018b\toolbox\sdt');
2     sdtcheck path
3     %ver primer
4
5     %% Parámetros iniciales
6
7     % Dimensiones de la celda unitaria y estructura
8     a = 0.5;           % Longitud de la celda unitaria en m
9     b = 0.5;           % Anchura de la celda unitaria en m
10    num_cells_x = 10;  % Número de celdas unitarias en el eje x
11    num_cells_y = 1;   % Número de celdas unitarias en el eje y
12
13    % Propiedades del material (acero)
14    E = 206e9;         % Módulo de Young en Pa
15    nu = 0.3;         % Coeficiente de Poisson
16    rho = 7850;       % Densidad en kg/m3
17
18    % Dimensiones de las vigas
19    r = 0.01; % Radio de sección menor en m
20    R = 0.04; % Radio de sección mayor en m
21
22    %% Creación de los nodos del marco exterior
23
24    mn = 10; %numero de nodos por viga de largo a (o b)
25
26    model.Node = [];
27    k=1;
28    NH1 = [];
29    NH2 = [];
30    NV = [];
31
32    % nodos horizontales inferiores
33    for i = 0:(a/mn):(num_cells_x*a)
34        model.Node = [model.Node;k 0 0 0 i 0 0];
35        NH1 = [NH1;k];
36        k=k+1;
37    end
38
39    % nodos horizontales superiores
40    for i = 0:(a/mn):(num_cells_x*a)
41        model.Node = [model.Node;k 0 0 0 i b 0];
42        NH2 = [NH2;k];
```

```

43     k=k+1;
44 end
45
46 % nodos verticales
47 for j=0:num_cells_x
48     for i=0:(b/nn):(num_cells_y*b)
49         model.Node = [model.Node;k 0 0 0 j*a i 0];
50         NV = [NV;k];
51         k=k+1;
52     end
53 end
54
55 %% Elementos del marco exterior
56
57 dir = [0 0 1]; % dirección de la normal al elemento
58 model.Elt=[Inf abs('beam1') 0]; % Elemento tipo viga Euler–Bernoulli
59
60 % Elementos marco inferior
61 for i = NH1(1):length(NH1)-1
62     model.Elt = [model.Elt; i i+1 1 1 dir];
63 end
64
65 % Elementos marco superior
66 for i = 1:length(NH2)-1
67     nodo = NH2(i);
68     model.Elt = [model.Elt; nodo nodo+1 1 1 dir];
69 end
70
71 % Elementos verticales
72 for j=0:num_cells_x
73     n_i = j*(nn+1) + 1 + length(NH1) + length(NH2);
74     n_f = n_i + nn;
75     % Unimos los extremos de las barras verticales con las barras horizontales
76     nh1 = j*nn + 1;
77     nh2 = j*nn + num_cells_x*nn + 2;
78     model.Elt = [model.Elt; n_i+1 nh1 1 1 dir];
79     model.Elt = [model.Elt; n_f-1 nh2 1 1 dir];
80     for i = n_i:n_f-1
81         model.Elt = [model.Elt; i i+1 1 1 dir];
82     end
83 end
84
85 %% Creación de los nodos del enrejado interno
86 ND1=[];
87 ND2=[];
88
89 % Barras diagonales (inferior izquierdo a superior derecho)
90 for j=1:num_cells_x
91     for i=0:(b/nn):(num_cells_y*b)
92         x = i + (j-1)*b;
93         y = i;
94         model.Node = [model.Node;k 0 0 0 x y 0];
95         ND1 = [ND1;k];
96         k = k+1;
97     end
98 end

```

```

99
100 % Barras diagonales (inferior derecho a superior izquierdo)
101 for j=1:num_cells_x
102     for i=0:(b/nn):(num_cells_y*b)
103         x = j*b - i;
104         y = i;
105         model.Node = [model.Node;k 0 0 0 x y 0];
106         ND2 = [ND2;k];
107         k = k+1;
108     end
109 end
110
111 %% Elementos enrejado interno
112
113 for j=1:num_cells_x
114     nd1_i = (j-1)*(nn+1) + 1 + length(NH1) + length(NH2) + length(NV);
115     nd1_f = nd1_i + nn;
116     nd2_i = (j-1)*(nn+1) + 1 + length(NH1) + length(NH2) + length(NV) + length(ND1);
117     nd2_f = nd2_i + nn;
118     % Unimos los extremos de las barras diagonales con las barras horizontales
119     nh1 = (j-1)*nn + 1;
120     nh2 = (j-1)*nn + num_cells_x*nn + 2;
121     model.Elt = [model.Elt; nd1_i+1 nh1 1 1 dir];
122     model.Elt = [model.Elt; nd1_f-1 nh2+nn 1 1 dir];
123     model.Elt = [model.Elt; nd2_i+1 nh1+nn 1 1 dir];
124     model.Elt = [model.Elt; nd2_f-1 nh2 1 1 dir];
125     for i = nd1_i:nd1_f-1
126         if i < (nd1_f - 3/4*(nd1_f-nd1_i))
127             model.Elt = [model.Elt; i i+1 1 1 dir];
128         elseif i > (nd1_f - 1/4*(nd1_f-nd1_i))
129             model.Elt = [model.Elt; i i+1 1 1 dir];
130         else
131             model.Elt = [model.Elt; i i+1 1 2 dir];
132         end
133     end
134     for l = nd2_i:nd2_f-1
135         if l < (nd2_f - 3/4*(nd2_f-nd2_i))
136             model.Elt = [model.Elt; l l+1 1 1 dir];
137         elseif l > (nd2_f - 1/4*(nd2_f-nd2_i))
138             model.Elt = [model.Elt; l l+1 1 1 dir];
139         else
140             model.Elt = [model.Elt; l l+1 1 2 dir];
141         end
142     end
143 end
144
145 %% Definicion del material
146 model.pl=[1 fe_mat('m_elastic','SI',1) E nu rho];
147
148 %% Definición de las propiedades de los elementos
149 model.il = [];
150 radios = [r R]; % dos secciones
151
152 for i = 1:length(radios)
153     Ix = 1/4 * pi * radios(i)^4;
154     Iy = Ix;

```

```

155     Iz = Ix + Iy;
156     A = pi * radios(i)^2;
157     model.il = [model.il; i fe_mat('p_beam','US',1) Iz Ix Iy A];
158 end
159
160 %% Empotrado izquierdo
161
162 % Nodos horizontales
163 empotradoH = [1 length(NH1)+1];
164 % Nodos verticales
165 empotradoV = [(length(NH1) + length(NH2)+1):1:(length(NH1) + length(NH2)+1+nn)];
166 % Nodos diagonales
167 empotradoD = [1+length(NH1)+length(NH2)+length(NV) 1+length(NH1)+length(NH2)+length
↪ (NV)+length(ND1)+nn];
168
169 %% Generar modelo, modos de vibración y FRF
170
171 % Grados de libertad fijos y cargas
172 model0 = fe_case(model, 'FixDof', 'AllDof', [empotradoH empotradoV empotradoD .03 .04
↪ .05],...
173             'SensDof', 'Sensors', [202.02], ...
174             'DofLoad', 'Force', [122.02]);
175
176 % Construir matrices de rigidez y masa
177 [m0,k0,mdof0] = fe_mk(model0,'Options',[]);
178
179 % Determinar modos de vibración
180 def0 = fe_eig({m0, k0, mdof0}, [5 length(m0) 1e3]);
181
182 % Calcular funciones de respuesta en frecuencia
183 IIXh0 = nor2xf(def0, 0, model0, 1:0.1:3000, ['hz', 'dis']);
184
185 %% Graficos
186
187 % Graficar modos
188 cf = feplot(model0, def0);
189
190 % Graficar FRF
191 figure;
192 semilogy(1:0.1:3000, abs(IIXh0(:,1)))
193 xlabel('Frecuencia (Hz)');
194 ylabel('Desplazamiento (m)');

```

Anexo B. Formulación estructuras

B.1. Formulación estructura Pratt

```
1 function [model0, def0] = Formulacion_Pratt(r, R)
2
3 path(path, 'C:\Program Files\MATLAB\R2018b\toolbox\sdt');
4 sdtcheck path
5 %% Parámetros que definen el marco
6
7 % a = Longitud de la celda unitaria en m
8 % b = Anchura de la celda unitaria en m
9 % num_cells_x = Número de celdas unitarias en el eje x
10 % num_cells_y = Número de celdas unitarias en el eje y
11 % r = radio de sección de las vigas del marco
12 % R = vector con radio de sección de las vigas
13
14 %% Parámetros para restricciones de optimización
15
16 % R_max = radio máximo de las secciones de las vigas
17 % R_min = radio mínimo de las secciones de las vigas
18
19 %% Parámetros para generar el marco
20 a = 0.03;
21 b = 0.03;
22 num_cells_x = 8;
23 num_cells_y = 4;
24
25 %% Parámetros que definen el enrejado
26 nc = num_cells_x*num_cells_y; % Número de celdas unitarias
27
28 %% Creación de nodos marco exterior
29
30 nn = 10; %numero de nodos por viga de largo a (o b)
31
32 model.Node = [];
33 k=1;
34 NH1 = [];
35 NH2 = [];
36 NV = [];
37
38 % nodos horizontales inferiores
39 for i = 0:(a/nn):(num_cells_x*a)
40     model.Node = [model.Node;k 0 0 0 i 0 0];
41     NH1 = [NH1;k];
42     k=k+1;
43 end
44
45 % nodos horizontales superiores
46 for j=1:num_cells_y
47     for i = 0:(a/nn):(num_cells_x*a)
48         model.Node = [model.Node;k 0 0 0 i j*b 0];
49         NH2 = [NH2;k];
50         k=k+1;
51     end
```

```

52     end
53
54     % nodos verticales
55     for j=0:num_cells_x
56         for i=1:num_cells_y
57             for l=b/nn:(b/nn):(b - b/nn)
58                 model.Node = [model.Node;k 0 0 0 j*a (i-1)*b+l 0];
59                 NV = [NV;k];
60                 k=k+1;
61             end
62         end
63     end
64
65     % Marcos laterales
66     MI = []; % izquierdo
67     MD = []; % derecho
68     for i=1:length(NV)/(num_cells_x+1)
69         izq = NV(i);
70         MI = [MI; izq];
71         der = NV(length(NV)-(i-1));
72         MD = [MD; der];
73     end
74
75     %% Elementos del marco
76     dir = [0 0 1]; % dirección a la normal del elemento
77     model.Elt=[Inf abs('beam1') 0]; % elemento tipo viga Euler-Bernoulli
78
79     % Elementos marco inferior
80     for i = NH1(1):length(NH1)-1
81         model.Elt = [model.Elt; i i+1 1 1 dir];
82     end
83
84     % Elementos horizontales superiores
85     for j=1:num_cells_y
86         m = length(NH2)/num_cells_y;
87         inicio = (j-1)*m +1;
88         final = j*m;
89         for i = inicio:final-1
90             nodo = NH2(i);
91             model.Elt = [model.Elt; nodo nodo+1 1 1 dir];
92         end
93     end
94
95     % Elementos verticales
96     for j=0:num_cells_x
97         for i = 1:num_cells_y
98             n_inicial = j*num_cells_y*(nn-1) + (i-1)*(nn-1) + 1 + length(NH1) + length(NH2);
99             n_final = n_inicial + (nn-2);
100             nh1 = j*nn + (i-1)*(nn*num_cells_x+1) + 1;
101             nh2 = nh1 + (num_cells_x*nn + 1);
102             model.Elt = [model.Elt; nh1 n_inicial 1 1 dir];
103             model.Elt = [model.Elt; n_final nh2 1 1 dir];
104             for l = n_inicial:n_final-1
105                 model.Elt = [model.Elt; l l+1 1 1 dir];
106             end
107         end

```

```

108     end
109
110     %% Creación de nodos del enrejado interno
111     ND = [];
112
113     %% Barras diagonales
114     for j=1:num_cells_y
115         for i=1:num_cells_x
116             for l=b/nn:(b/nn):b-b/nn
117                 if rem(j,2)==0
118                     if rem(i,2)==0
119                         x = (i-1)*a + l*(a/b);
120                         y = (j-1)*b + (b - l);
121                         model.Node = [model.Node;k 0 0 0 x y 0];
122                         ND = [ND;k];
123                         k = k+1;
124                     else
125                         x = (i-1)*a + l*(a/b);
126                         y = (j-1)*b + l;
127                         model.Node = [model.Node;k 0 0 0 x y 0];
128                         ND = [ND;k];
129                         k = k+1;
130                     end
131                 else
132                     if rem(i,2)==0
133                         x =(i-1)*a + l*(a/b);
134                         y = (j-1)*b + l;
135                         model.Node = [model.Node;k 0 0 0 x y 0];
136                         ND = [ND;k];
137                         k = k+1;
138                     else
139                         x = (i-1)*a + l*(a/b);
140                         y = (j-1)*b + (b - l);
141                         model.Node = [model.Node;k 0 0 0 x y 0];
142                         ND = [ND;k];
143                         k = k+1;
144                     end
145                 end
146             end
147         end
148     end
149
150     %% Elementos enrejado
151     for j=1:num_cells_y
152         for i=1:num_cells_x
153             nd_i = ND(1) + (j-1)*(nn-1)*num_cells_x + (i-1)*(nn-1);
154             nd_f = nd_i + (nn-2);
155             nh1 = (j-1)*(nn*num_cells_x + 1) + 1 + (i-1)*nn;
156             nh2 = nh1 + nn*num_cells_x + 1;
157             %% Unimos los extremos de las barras diagonales a las horizontales
158             if rem(j,2)==0
159                 if rem(i,2)==0
160                     model.Elt = [model.Elt; nd_i nh2 1 (j-1)*num_cells_x+i+1 dir];
161                     model.Elt = [model.Elt; nd_f nh1+nn 1 (j-1)*num_cells_x+i+1 dir];
162                 else
163                     model.Elt = [model.Elt; nd_i nh1 1 (j-1)*num_cells_x+i+1 dir];

```

```

164         model.Elt = [model.Elt; nd_f nh2+nn 1 (j-1)*num_cells_x+i+1 dir];
165     end
166 else
167     if rem(i,2)==0
168         model.Elt = [model.Elt; nd_i nh1 1 (j-1)*num_cells_x+i+1 dir];
169         model.Elt = [model.Elt; nd_f nh2+nn 1 (j-1)*num_cells_x+i+1 dir];
170     else
171         model.Elt = [model.Elt; nd_i nh2 1 (j-1)*num_cells_x+i+1 dir];
172         model.Elt = [model.Elt; nd_f nh1+nn 1 (j-1)*num_cells_x+i+1 dir];
173     end
174 end
175 % Unimos los nodos de las barras diagonales entre si
176 for l=nd_i:nd_f-1
177     model.Elt = [model.Elt; l l+1 1 (j-1)*num_cells_x+i+1 dir];
178 end
179 end
180 end
181
182 %% Definición del material (gray resin V4)
183 E = 3.580e9; % Módulo de Young en Pa
184 nu = 0.3331; % Coeficiente de Poisson
185 rho = 1333; % Densidad en kg/m^3
186
187 model.pl=[1 fe_mat('m_elastic','SI',1) E nu rho];
188
189 %% Definición de las propiedades de las vigas
190
191 % Marco
192 Ix = 1/4 * pi * r^4;
193 A = pi * r^2;
194 model.il = [1 fe_mat('p_beam','SI',1) 2*Ix Ix Ix A];
195
196 % Enrejado
197 for i=1:length(R)
198     R_i = R(i); % Radio de sección de vigas
199     Ix = 1/4 * pi * R_i^4;
200     A = pi * R_i^2;
201     model.il = [model.il; i+1 fe_mat('p_beam','SI',1) 2*Ix Ix Ix A];
202 end
203
204 %% Generar modelo y modos de vibración
205
206 % Grados de libertad fijos y cargas
207 model0 = fe_case(model, 'FixDof', 'AllDof', [.01 .02 .06],...
208     'SensDof', 'Sensors', MD(1)+.03, ...
209     'DofLoad', 'Force', MI(1:2:length(MI))+.03);
210
211 % Construir matrices de rigidez y masa
212 [m0,k0,mdof0] = fe_mk(model0,'Options',[]);
213
214 % Determinar modos de vibración
215 def0 = fe_eig({m0, k0, mdof0}, [5 length(m0) 1e3]);
216
217 end

```

B.2. Formulación estructura Baltimore

```
1 function [model0, def0] = Formulacion_Baltimore(r,R)
2
3 path(path,'C:\Program Files\MATLAB\R2018b\toolbox\sdt');
4 sdtcheck path
5 %% Parámetros que definen el marco
6
7 %% a = Longitud de la celda unitaria en m
8 %% b = Anchura de la celda unitaria en m
9 %% num_cells_x = Número de celdas unitarias en el eje x
10 %% num_cells_y = Número de celdas unitarias en el eje y
11 %% r = radio de sección de las vigas del marco
12 %% R = vector con radio de sección de las vigas
13
14 %% Parámetros para restricciones de optimización
15
16 %% R_max = radio máximo de las secciones de las vigas
17 %% R_min = radio mínimo de las secciones de las vigas
18
19 %% Parámetros para generar el marco
20 a = 0.03;
21 b = 0.03;
22 num_cells_x = 8;
23 num_cells_y = 4;
24
25 %% Parámetros que definen el enrejado
26 nc = num_cells_x*num_cells_y; % Número de celdas unitarias
27 nv = 3*nc; % Número de vigas en el enrejado
28
29 %% Creación de nodos del marco exterior
30
31 nn = 10; %numero de nodos por viga de largo a (o b)
32
33 model.Node = [];
34 k=1;
35 NH1 = [];
36 NH2 = [];
37 NV = [];
38
39 %% nodos horizontales inferiores
40 for i = 0:(a/nn):(num_cells_x*a)
41     model.Node = [model.Node;k 0 0 0 i 0 0];
42     NH1 = [NH1;k];
43     k=k+1;
44 end
45
46 %% nodos horizontales superiores
47 for j=1:num_cells_y
48     for i = 0:(a/nn):(num_cells_x*a)
49         model.Node = [model.Node;k 0 0 0 i j*b 0];
50         NH2 = [NH2;k];
51         k=k+1;
52     end
53 end
54
```

```

55 % nodos verticales
56 for j=0:num_cells_x
57     for i=1:num_cells_y
58         for l=b/nn:(b/nn):(b - b/nn)
59             model.Node = [model.Node;k 0 0 0 j*a (i-1)*b+1 0];
60             NV = [NV;k];
61             k=k+1;
62         end
63     end
64 end
65
66 % Marcos laterales
67 MI = []; % izquierdo
68 MD = []; % derecho
69 for i=1:length(NV)/(num_cells_x+1)
70     izq = NV(i);
71     MI = [MI; izq];
72     der = NV(length(NV)-(i-1));
73     MD = [MD; der];
74 end
75
76 %% Elementos del marco
77 dir = [0 0 1]; % dirección a la normal del elemento
78 model.Elt=[Inf abs('beam1') 0]; % elemento tipo viga Euler–Bernoulli
79
80 % Elementos marco inferior
81 for i = NH1(1):length(NH1)-1
82     model.Elt = [model.Elt; i i+1 1 1 dir];
83 end
84
85 % Elementos horizontales superiores
86 for j=1:num_cells_y
87     m = length(NH2)/num_cells_y;
88     inicio = (j-1)*m +1;
89     final = j*m;
90     for i = inicio:final-1
91         nodo = NH2(i);
92         model.Elt = [model.Elt; nodo nodo+1 1 1 dir];
93     end
94 end
95
96 % Elementos verticales
97 for j=0:num_cells_x
98     for i = 1:num_cells_y
99         n_inicial = j*num_cells_y*(nn-1) + (i-1)*(nn-1) + 1 + length(NH1) + length(NH2);
100        n_final = n_inicial + (nn-2);
101        nh1 = j*nn + (i-1)*(nn*num_cells_x+1) + 1;
102        nh2 = nh1 + (num_cells_x*nn + 1);
103        model.Elt = [model.Elt; nh1 n_inicial 1 1 dir];
104        model.Elt = [model.Elt; n_final nh2 1 1 dir];
105        for l = n_inicial:n_final-1
106            model.Elt = [model.Elt; l l+1 1 1 dir];
107        end
108    end
109 end
110

```

```

111 %% Creación de nodos del enrejado interno
112 ND = [];
113
114 %% Barras diagonales
115 for j=1:num_cells_y
116     for i=1:num_cells_x
117         for l=b/nn:(b/nn):b-b/nn
118             if rem(j,2)==0
119                 if rem(i,2)==0
120                     x = (i-1)*a + l*(a/b);
121                     y = (j-1)*b + (b - l);
122                     model.Node = [model.Node;k 0 0 0 x y 0];
123                     ND = [ND;k];
124                     k = k+1;
125                 else
126                     x = (i-1)*a + l*(a/b);
127                     y = (j-1)*b + l;
128                     model.Node = [model.Node;k 0 0 0 x y 0];
129                     ND = [ND;k];
130                     k = k+1;
131                 end
132             else
133                 if rem(i,2)==0
134                     x = (i-1)*a + l*(a/b);
135                     y = (j-1)*b + l;
136                     model.Node = [model.Node;k 0 0 0 x y 0];
137                     ND = [ND;k];
138                     k = k+1;
139                 else
140                     x = (i-1)*a + l*(a/b);
141                     y = (j-1)*b + (b - l);
142                     model.Node = [model.Node;k 0 0 0 x y 0];
143                     ND = [ND;k];
144                     k = k+1;
145                 end
146             end
147         end
148     end
149 end
150
151 %% Elementos enrejado
152 for j=1:num_cells_y
153     for i=1:num_cells_x
154         nd_i = ND(1) + (j-1)*(nn-1)*num_cells_x + (i-1)*(nn-1);
155         nd_f = nd_i + (nn-2);
156         nh1 = (j-1)*(nn*num_cells_x + 1) + 1 + (i-1)*nn;
157         nh2 = nh1 + nn*num_cells_x + 1;
158         %% Unimos los extremos de las barras diagonales a las horizontales
159         if rem(j,2)==0
160             if rem(i,2)==0
161                 model.Elt = [model.Elt; nd_i nh2 1 (j-1)*num_cells_x+i+1 dir];
162                 model.Elt = [model.Elt; nd_f nh1+nn 1 (j-1)*num_cells_x+i+1 dir];
163             else
164                 model.Elt = [model.Elt; nd_i nh1 1 (j-1)*num_cells_x+i+1 dir];
165                 model.Elt = [model.Elt; nd_f nh2+nn 1 (j-1)*num_cells_x+i+1 dir];
166             end
167         end
168     end
169 end

```

```

167         else
168             if rem(i,2)==0
169                 model.Elt = [model.Elt; nd_i nh1 1 (j-1)*num_cells_x+i+1 dir];
170                 model.Elt = [model.Elt; nd_f nh2+nn 1 (j-1)*num_cells_x+i+1 dir];
171             else
172                 model.Elt = [model.Elt; nd_i nh2 1 (j-1)*num_cells_x+i+1 dir];
173                 model.Elt = [model.Elt; nd_f nh1+nn 1 (j-1)*num_cells_x+i+1 dir];
174             end
175         end
176         % Unimos los nodos de las barras diagonales entre si
177         for l=nd_i:nd_f-1
178             model.Elt = [model.Elt; l l+1 1 (j-1)*num_cells_x+i+1 dir];
179         end
180     end
181 end
182
183 %% Nodos barras de conexion
184 NVI = []; % verticales
185 NDI = []; % diagonales
186
187 for j=1:num_cells_y
188     for i=1:num_cells_x
189         % nodos verticales de conexión
190         for l=(b/nn)/2:(b/nn)/2:(b-b/nn)/2
191             % creamos nodos con condición de paridad según la celda en
192             % que nos encontremos
193             if rem(j,2)==0
194                 x = (2*i-1)*a/2;
195                 y = (2*j-1)*b/2 - l;
196                 model.Node = [model.Node;k 0 0 0 x y 0];
197                 NVI = [NVI;k];
198                 k = k+1;
199             else
200                 x = (2*i-1)*a/2;
201                 y = (2*j-1)*b/2 + l;
202                 model.Node = [model.Node;k 0 0 0 x y 0];
203                 NVI = [NVI;k];
204                 k = k+1;
205             end
206         end
207         % nodos diagonales de conexión
208         for l=(b/nn)/2:(b/nn)/2:(b-b/nn)/2
209             % creamos nodos con condición de paridad según la celda en
210             % que nos encontremos
211             if rem(j,2)==0
212                 if rem(i,2)==0
213                     x = (2*i-1)*a/2 - l*(a/b);
214                     y = (2*j-1)*b/2 - l;
215                     model.Node = [model.Node;k 0 0 0 x y 0];
216                     NDI = [NDI;k];
217                     k = k+1;
218                 else
219                     x = (2*i-1)*a/2 + l*(a/b);
220                     y = (2*j-1)*b/2 - l;
221                     model.Node = [model.Node;k 0 0 0 x y 0];
222                     NDI = [NDI;k];

```

```

223         k = k+1;
224     end
225 else
226     if rem(i,2)==0
227         x = (2*i-1)*a/2 - l*(a/b);
228         y = (2*j-1)*b/2 + l;
229         model.Node = [model.Node;k 0 0 0 x y 0];
230         NDI = [NDI;k];
231         k = k+1;
232     else
233         x = (2*i-1)*a/2 + l*(a/b);
234         y = (2*j-1)*b/2 + l;
235         model.Node = [model.Node;k 0 0 0 x y 0];
236         NDI = [NDI;k];
237         k = k+1;
238     end
239 end
240 end
241 end
242 end
243
244 %% Elementos de barras de conexion enrejado
245
246 for j=1:num_cells_y
247     for i=1:num_cells_x
248         nvI_i = NVI(1) + 2*(i-1)*(nn - 1) + 2*(j-1)*(nn - 1)*num_cells_x;
249         nvI_f = nvI_i + nn - 2;
250         ndI_i = NDI(1) + 2*(i-1)*(nn- 1) + 2*(j-1)*(nn - 1)*num_cells_x;
251         ndI_f = ndI_i + nn - 2;
252         nh = (j-1)*(nn*num_cells_x + 1) + 1 + (i-1)*nn;
253         salto = nn*num_cells_x + 1; %salto entre celdas en el eje y
254         nd = ND(1) - 1 + (i-1)*(nn-1) + (j-1)*(nn-1)*num_cells_x;
255         % Unimos los extremos de las barras de conexion a las diagonales
256         if rem(j,2)==0
257             if rem(i,2)==0
258                 model.Elt = [model.Elt; nvI_i nd+nn/2 1 nc+(j-1)*2*num_cells_x+2*i dir];
259                 model.Elt = [model.Elt; nvI_f nh+nn/2 1 nc+(j-1)*2*num_cells_x+2*i dir];
260                 model.Elt = [model.Elt; ndI_i nd+nn/2 1 nc+(j-1)*2*num_cells_x+2*i+1 dir];
261                 model.Elt = [model.Elt; ndI_f nh 1 nc+(j-1)*2*num_cells_x+2*i+1 dir];
262             else
263                 model.Elt = [model.Elt; nvI_i nd+nn/2 1 nc+(j-1)*2*num_cells_x+2*i dir];
264                 model.Elt = [model.Elt; nvI_f nh+nn/2 1 nc+(j-1)*2*num_cells_x+2*i dir];
265                 model.Elt = [model.Elt; ndI_i nd+nn/2 1 nc+(j-1)*2*num_cells_x+2*i+1 dir];
266                 model.Elt = [model.Elt; ndI_f nh+nn 1 nc+(j-1)*2*num_cells_x+2*i+1 dir];
267             end
268         else
269             if rem(i,2)==0
270                 model.Elt = [model.Elt; nvI_i nd+nn/2 1 nc+(j-1)*2*num_cells_x+2*i dir];
271                 model.Elt = [model.Elt; nvI_f nh+salto+nn/2 1 nc+(j-1)*2*num_cells_x+2*i dir];
272                 model.Elt = [model.Elt; ndI_i nd+nn/2 1 nc+(j-1)*2*num_cells_x+2*i+1 dir];
273                 model.Elt = [model.Elt; ndI_f nh+salto 1 nc+(j-1)*2*num_cells_x+2*i+1 dir];
274             else
275                 model.Elt = [model.Elt; nvI_i nd+nn/2 1 nc+(j-1)*2*num_cells_x+2*i dir];
276                 model.Elt = [model.Elt; nvI_f nh+salto+nn/2 1 nc+(j-1)*2*num_cells_x+2*i dir];
277                 model.Elt = [model.Elt; ndI_i nd+nn/2 1 nc+(j-1)*2*num_cells_x+2*i+1 dir];
278                 model.Elt = [model.Elt; ndI_f nh+nn+salto 1 nc+(j-1)*2*num_cells_x+2*i+1 dir];

```

```

279         end
280     end
281     % Unimos los nodos de las barras de conexión entre sí
282     for l=ndI_i:ndI_f-1
283         model.Elt = [model.Elt; l l+1 1 nc+(j-1)*2*num_cells_x+2*i dir];
284     end
285     for l=ndI_i:ndI_f-1
286         model.Elt = [model.Elt; l l+1 1 nc+(j-1)*2*num_cells_x+2*i+1 dir];
287     end
288 end
289 end
290
291 %% Definición del material (gray resin V4)
292 E = 3.580e9; % Módulo de Young en Pa
293 nu = 0.3331; % Coeficiente de Poisson
294 rho = 1333; % Densidad en kg/m^3
295
296 model.pl=[1 fe_mat('m_elastic','SI',1) E nu rho];
297
298 %% Definición de las propiedades de las vigas
299
300 % Marco
301 Ix = 1/4 * pi * r^4;
302 A = pi * r^2;
303 model.il = [1 fe_mat('p_beam','SI',1) 2*Ix Ix Ix A];
304
305 % Enrejado
306 for i=1:length(R)
307     R_i = R(i); % Radio de sección de vigas
308     Ix = 1/4 * pi * R_i^4;
309     A = pi * R_i^2;
310     model.il = [model.il; i+1 fe_mat('p_beam','SI',1) 2*Ix Ix Ix A];
311 end
312
313 %% Generar modelo y modos de vibración
314
315 % Grados de libertad fijos y cargas
316 model0 = fe_case(model, 'FixDof', 'AllDof', [.01 .02 .06],...
317     'SensDof', 'Sensors', MD(1)+.03, ...
318     'DofLoad', 'Force', MI(1:2:length(MI))+.03);
319
320 % Construir matrices de rigidez y masa
321 [m0,k0,mdof0] = fe_mk(model0,'Options',[]);
322
323 % Determinar modos de vibración
324 def0 = fe_eig({m0, k0, mdof0}, [5 length(m0) 1e3]);
325
326 end

```

B.3. Formulación estructura Mixta

```
1 function [model0, def0] = Formulacion_Mixta(r,R)
2
3 path(path,'C:\Program Files\MATLAB\R2018b\toolbox\sdt');
4 sdtcheck path
5 %% Parámetros que definen el marco
6
7 %% a = Longitud de la celda unitaria en m
8 %% b = Anchura de la celda unitaria en m
9 %% num_cells_x = Número de celdas unitarias en el eje x
10 %% num_cells_y = Número de celdas unitarias en el eje y
11 %% r = radio de sección de las vigas del marco
12 %% R = vector con radio de sección de las vigas
13
14 %% Parámetros para restricciones de optimización
15
16 %% R_max = radio máximo de las secciones de las vigas
17 %% R_min = radio mínimo de las secciones de las vigas
18
19 %% Parámetros para generar el marco
20 a = 0.03;
21 b = 0.03;
22 num_cells_x = 8;
23 num_cells_y = 4;
24
25 %% Parámetros que definen el enrejado
26 nc = num_cells_x*num_cells_y; % Número de celdas unitarias
27 nc1 = 4*num_cells_y; % Número de celdas unitarias con 1 viga
28 nc2 = nc-nc1; % Número de celdas unitarias con 2 vigas
29 nv = nc1 + 2*nc2; % Número de vigas en el enrejado
30
31 %% Creación de nodos del marco exterior
32
33 nn = 10; %numero de nodos por viga de largo a (o b).
34
35 model.Node = [];
36 k=1;
37 NH1 = [];
38 NH2 = [];
39 NV = [];
40
41 %% nodos horizontales inferiores
42 for i = 0:(a/nn):(num_cells_x*a)
43     model.Node = [model.Node;k 0 0 0 i 0 0];
44     NH1 = [NH1;k];
45     k=k+1;
46 end
47
48 %% nodos horizontales superiores
49 for j=1:num_cells_y
50     for i = 0:(a/nn):(num_cells_x*a)
51         model.Node = [model.Node;k 0 0 0 i j*b 0];
52         NH2 = [NH2;k];
53         k=k+1;
54     end
55 end
```

```

55     end
56
57     % nodos verticales
58     for j=0:num_cells_x
59         for i=1:num_cells_y
60             for l=b/nn:(b/nn):(b - b/nn)
61                 model.Node = [model.Node;k 0 0 0 j*a (i-1)*b+l 0];
62                 NV = [NV;k];
63                 k=k+1;
64             end
65         end
66     end
67
68     % Marcos laterales
69     MI = []; % izquierdo
70     MD = []; % derecho
71     for i=1:length(NV)/(num_cells_x+1)
72         izq = NV(i);
73         MI = [MI; izq];
74         der = NV(length(NV)-(i-1));
75         MD = [MD; der];
76     end
77
78     %% Elementos del marco
79     dir = [0 0 1];
80     model.Elt=[Inf abs('beam1') 0];
81
82     % Elementos marco inferior
83     for i = NH1(1):length(NH1)-1
84         model.Elt = [model.Elt; i i+1 1 1 dir];
85     end
86
87     % Elementos horizontales superiores
88     for j=1:num_cells_y
89         m = length(NH2)/num_cells_y;
90         inicio = (j-1)*m +1;
91         final = j*m;
92         for i = inicio:final-1
93             nodo = NH2(i);
94             model.Elt = [model.Elt; nodo nodo+1 1 1 dir];
95         end
96     end
97
98     % Elementos verticales
99     for j=0:num_cells_x
100         for i = 1:num_cells_y
101             n_inicial = j*num_cells_y*(nn-1) + (i-1)*(nn-1) + 1 + length(NH1) + length(NH2);
102             n_final = n_inicial + (nn-2);
103             nh1 = j*nn + (i-1)*(nn*num_cells_x+1) + 1;
104             nh2 = nh1 + (num_cells_x*nn + 1);
105             model.Elt = [model.Elt; nh1 n_inicial 1 1 dir];
106             model.Elt = [model.Elt; n_final nh2 1 1 dir];
107             for l = n_inicial:n_final-1
108                 model.Elt = [model.Elt; l l+1 1 1 dir];
109             end
110         end

```

```

111     end
112
113     %% Creación de nodos del enrejado interno
114     N1 = [];
115     N2 = [];
116     N3 = [];
117     N4 = [];
118
119     for j=1:num_cells_y
120         for i=1:num_cells_x
121             for l=b/nn:(b/nn):b-b/nn
122                 if rem(j,2)==0
123                     if i==1
124                         x = (i-1)*a + l*(a/b);
125                         y = (j-1)*b + (b - l);
126                         model.Node = [model.Node;k 0 0 0 x y 0];
127                         N1 = [N1;k];
128                         k = k+1;
129                     elseif i==num_cells_x
130                         x = (i-1)*a + l*(a/b);
131                         y = (j-1)*b + l;
132                         model.Node = [model.Node;k 0 0 0 x y 0];
133                         N1 = [N1;k];
134                         k = k+1;
135                     elseif i==2
136                         x = (i-1)*a + l*(a/b);
137                         y = (j-1)*b + l/2;
138                         model.Node = [model.Node;k 0 0 0 x y 0];
139                         N2 = [N2;k];
140                         k = k+1;
141                     elseif i==num_cells_x-1
142                         x = (i-1)*a + l*(a/b);
143                         y = (j-1)*b + (b-1)/2;
144                         model.Node = [model.Node;k 0 0 0 x y 0];
145                         N2 = [N2;k];
146                         k = k+1;
147                     elseif i<=num_cells_x/2
148                         x = (i-1)*a + l*(a/b);
149                         y1 = (j-1)*b + (b+1)/2;
150                         model.Node = [model.Node;k 0 0 0 x y1 0];
151                         N3 = [N3;k];
152                         k = k+1;
153                         y2 = (j-1)*b + (b-1)/2;
154                         model.Node = [model.Node;k 0 0 0 x y2 0];
155                         N4 = [N4;k];
156                         k = k+1;
157                     else
158                         x = (i-1)*a + l*(a/b);
159                         y1 = (j-1)*b + b - l/2;
160                         model.Node = [model.Node;k 0 0 0 x y1 0];
161                         N3 = [N3;k];
162                         k = k+1;
163                         y2 = (j-1)*b + l/2;
164                         model.Node = [model.Node;k 0 0 0 x y2 0];
165                         N4 = [N4;k];
166                         k = k+1;

```

```

167         end
168     else
169         if i==1
170             x = (i-1)*a + l*(a/b);
171             y = (j-1)*b + l;
172             model.Node = [model.Node;k 0 0 0 x y 0];
173             N1 = [N1;k];
174             k = k+1;
175         elseif i==num_cells_x
176             x = (i-1)*a + l*(a/b);
177             y = (j-1)*b + (b - l);
178             model.Node = [model.Node;k 0 0 0 x y 0];
179             N1 = [N1;k];
180             k = k+1;
181         elseif i==2
182             x = (i-1)*a + l*(a/b);
183             y = (j-1)*b + b - l/2;
184             model.Node = [model.Node;k 0 0 0 x y 0];
185             N2 = [N2;k];
186             k = k+1;
187         elseif i==num_cells_x-1
188             x = (i-1)*a + l*(a/b);
189             y = (j-1)*b + (b+1)/2;
190             model.Node = [model.Node;k 0 0 0 x y 0];
191             N2 = [N2;k];
192             k = k+1;
193         elseif i<=num_cells_x/2
194             x = (i-1)*a + l*(a/b);
195             y1 = (j-1)*b + (b+1)/2;
196             model.Node = [model.Node;k 0 0 0 x y1 0];
197             N3 = [N3;k];
198             k = k+1;
199             y2 = (j-1)*b + (b-1)/2;
200             model.Node = [model.Node;k 0 0 0 x y2 0];
201             N4 = [N4;k];
202             k = k+1;
203         else
204             x = (i-1)*a + l*(a/b);
205             y1 = (j-1)*b + b - l/2;
206             model.Node = [model.Node;k 0 0 0 x y1 0];
207             N3 = [N3;k];
208             k = k+1;
209             y2 = (j-1)*b + l/2;
210             model.Node = [model.Node;k 0 0 0 x y2 0];
211             N4 = [N4;k];
212             k = k+1;
213         end
214     end
215 end
216 end
217 end
218
219 %% Elementos enrejado
220
221 for j=1:num_cells_y
222     for i=1:num_cells_x

```

```

223 rango = num_cells_x-4; % número de celdas con 2 vigas
224 if i<=2
225     n_i = N1(1) + (i-1)*(nn-1) + (j-1)*(nn-1)*(num_cells_x+rango);
226     n_f = n_i + nn-2;
227 elseif i>2 && i < num_cells_x-1
228     n_i = N3(1) + (i-3)*2*(nn-1) + (j-1)*(nn-1)*(num_cells_x+rango);
229     n_f = n_i + 2*(nn-2);
230 else
231     n_i = N1(1) + (i-1+rango)*(nn-1) + (j-1)*(nn-1)*(num_cells_x+rango);
232     n_f = n_i + nn-2;
233 end
234 nv1 = (j-1)*(nn-1) + (i-1)*(nn-1)*num_cells_y + NV(1);
235 nv2 = nv1 + (nn-1)*num_cells_y;
236 nh1 = (j-1)*(nn*num_cells_x + 1) + 1 + (i-1)*nn;
237 nh2 = nh1 + nn*num_cells_x + 1;
238 % Unimos los extremos de las barras internas al marco
239 if rem(j,2)==0
240     if i==1
241         model.Elt = [model.Elt; n_i nh2 1 (j-1)*(nv/num_cells_y)+i+1 dir];
242         model.Elt = [model.Elt; n_f nh1+nn 1 (j-1)*(nv/num_cells_y)+i+1 dir];
243     elseif i==2
244         model.Elt = [model.Elt; n_i nh1 1 (j-1)*(nv/num_cells_y)+i+1 dir];
245         model.Elt = [model.Elt; n_f nv2+nn/2-1 1 (j-1)*(nv/num_cells_y)+i+1 dir];
246     elseif i==num_cells_x
247         model.Elt = [model.Elt; n_i nh1 1 (j-1)*(nv/num_cells_y)+(nc2/num_cells_y)+i
248         ↪ +1 dir];
249         model.Elt = [model.Elt; n_f nh2+nn 1 (j-1)*(nv/num_cells_y)+(nc2/num_cells_y)
250         ↪ +i+1 dir];
251     elseif i==num_cells_x-1
252         model.Elt = [model.Elt; n_i nv1+nn/2-1 1 (j-1)*(nv/num_cells_y)+(nc2/
253         ↪ num_cells_y)+i+1 dir];
254         model.Elt = [model.Elt; n_f nh1+nn 1 (j-1)*(nv/num_cells_y)+(nc2/num_cells_y)
255         ↪ +i+1 dir];
256     elseif i<=num_cells_x/2
257         model.Elt = [model.Elt; n_i nv1+nn/2-1 1 (j-1)*(nv/num_cells_y)+2*(i-1) dir];
258         model.Elt = [model.Elt; n_i+1 nv1+nn/2-1 1 (j-1)*(nv/num_cells_y)+2*(i-1)+1
259         ↪ dir];
260         model.Elt = [model.Elt; n_f nh2+nn 1 (j-1)*(nv/num_cells_y)+2*(i-1) dir];
261         model.Elt = [model.Elt; n_f+1 nh1+nn 1 (j-1)*(nv/num_cells_y)+2*(i-1)+1
262         ↪ dir];
263     else
264         model.Elt = [model.Elt; n_i nh2 1 (j-1)*(nv/num_cells_y)+2*(i-1) dir];
265         model.Elt = [model.Elt; n_i+1 nh1 1 (j-1)*(nv/num_cells_y)+2*(i-1)+1 dir];
266         model.Elt = [model.Elt; n_f nv2+nn/2-1 1 (j-1)*(nv/num_cells_y)+2*(i-1) dir];
267         model.Elt = [model.Elt; n_f+1 nv2+nn/2-1 1 (j-1)*(nv/num_cells_y)+2*(i-1)+1
268         ↪ dir];
269     end
270 else
271     if i==1
272         model.Elt = [model.Elt; n_i nh1 1 (j-1)*(nv/num_cells_y)+i+1 dir];
273         model.Elt = [model.Elt; n_f nh2+nn 1 (j-1)*(nv/num_cells_y)+i+1 dir];
274     elseif i==2
275         model.Elt = [model.Elt; n_i nh2 1 (j-1)*(nv/num_cells_y)+i+1 dir];
276         model.Elt = [model.Elt; n_f nv2+nn/2-1 1 (j-1)*(nv/num_cells_y)+i+1 dir];
277     elseif i==num_cells_x
278         model.Elt = [model.Elt; n_i nh2 1 (j-1)*(nv/num_cells_y)+(nc2/num_cells_y)+i
279         ↪ +1 dir];

```

```

272         model.Elt = [model.Elt; n_f nh1+nn 1 (j-1)*(nv/num_cells_y)+(nc2/num_cells_y)
↪ +i+1 dir];
273     elseif i==num_cells_x-1
274         model.Elt = [model.Elt; n_i nv1+nn/2-1 1 (j-1)*(nv/num_cells_y)+(nc2/
↪ num_cells_y)+i+1 dir];
275         model.Elt = [model.Elt; n_f nh2+nn 1 (j-1)*(nv/num_cells_y)+(nc2/num_cells_y)
↪ +i+1 dir];
276     elseif i<=num_cells_x/2
277         model.Elt = [model.Elt; n_i nv1+nn/2-1 1 (j-1)*(nv/num_cells_y)+2*(i-1) dir];
278         model.Elt = [model.Elt; n_i+1 nv1+nn/2-1 1 (j-1)*(nv/num_cells_y)+2*(i-1)+1
↪ dir];
279         model.Elt = [model.Elt; n_f nh2+nn 1 (j-1)*(nv/num_cells_y)+2*(i-1) dir];
280         model.Elt = [model.Elt; n_f+1 nh1+nn 1 (j-1)*(nv/num_cells_y)+2*(i-1)+1 dir];
281     else
282         model.Elt = [model.Elt; n_i nh2 1 (j-1)*(nv/num_cells_y)+2*(i-1) dir];
283         model.Elt = [model.Elt; n_i+1 nh1 1 (j-1)*(nv/num_cells_y)+2*(i-1)+1 dir];
284         model.Elt = [model.Elt; n_f nv2+nn/2-1 1 (j-1)*(nv/num_cells_y)+2*(i-1) dir];
285         model.Elt = [model.Elt; n_f+1 nv2+nn/2-1 1 (j-1)*(nv/num_cells_y)+2*(i-1)+1
↪ dir];
286     end
287 end
288 % Unimos los nodos de las barras interiores entre si
289 if i<=2
290     for l=n_i:n_f-1
291         model.Elt = [model.Elt; l l+1 1 (j-1)*(nv/num_cells_y)+i+1 dir];
292     end
293 elseif i >= 3 && i <= num_cells_x-2
294     for l=n_i:2:n_f-2
295         model.Elt = [model.Elt; l l+2 1 (j-1)*(nv/num_cells_y)+2*(i-1) dir];
296         model.Elt = [model.Elt; l+1 l+3 1 (j-1)*(nv/num_cells_y)+2*(i-1)+1 dir];
297     end
298 else
299     for l=n_i:n_f-1
300         model.Elt = [model.Elt; l l+1 1 (j-1)*(nv/num_cells_y)+(nc2/num_cells_y)+i+1
↪ dir];
301     end
302 end
303 end
304 end
305
306 %% Definición del material (gray resin V4)
307 E = 3.580e9; % Módulo de Young en Pa
308 nu = 0.3331; % Coeficiente de Poisson
309 rho = 1333; % Densidad en kg/m^3
310
311 model.pl=[1 fe_mat('m_elastic','SI',1) E nu rho];
312
313 %% Definición de las propiedades de las vigas
314
315 % Marco
316 Ix = 1/4 * pi * r^4;
317 A = pi * r^2;
318 model.il = [1 fe_mat('p_beam','SI',1) 2*Ix Ix Ix A];
319
320 % Enrejado
321 for i=1:length(R)

```

```

322     R_i = R(i); % Radio de sección de vigas
323     Ix = 1/4 * pi * R_i^4;
324     A = pi * R_i^2;
325     model.il = [model.il; i+1 fe_mat('p_beam','SI',1) 2*Ix Ix Ix A];
326 end
327
328 %% Generar modelo y modos de vibración
329
330 % Grados de libertad fijos y cargas
331 model0 = fe_case(model, 'FixDof', 'AllDof', [.01 .02 .06],...
332     'SensDof', 'Sensors', MD(1)+.03, ...
333     'DofLoad', 'Force', MI(1:2:length(MI))+.03);
334
335 % Construir matrices de rigidez y masa
336 [m0,k0,mdof0] = fe_mk(model0,'Options',[]);
337
338 % Determinar modos de vibración
339 def0 = fe_eig({m0, k0, mdof0}, [5 length(m0) 1e3]);
340
341 end

```

Anexo C. Funciones objetivo

Se busca minimizar las funciones respuesta en frecuencia. Para esto, se calcula su promedio y se obtiene la RMS asociada a esa estructura.

C.1. Estructura Pratt

```
1 function val = obj_Pratt(r,R)
2 %% Generar modelo
3 [model0, def0] = Formulacion_Pratt(r,R);
4
5 %% Calcular funciones de respuesta en frecuencia
6 Iixh0 = nor2xf(def0, 0.005, model0, 1:1:4500, ['hz', 'dis']);
7 xf = Iixh0(:,:);
8
9 %% Función objetivo
10
11 %promedio de FRF para frecuencias en un rango de 500 Hz (en este caso 3500–4000 Hz)
12 xfp = mean(abs(xf(3500:4000,:)));
13
14 %rms calculada con promedio anterior
15 rm0 = sqrt(sum(abs(xfp).^2)/length(xfp));
16
17 val = rm0;
18 end
```

C.2. Estructura Baltimore

```
1 function val = obj_Baltimore(r,R)
2 %% Generar modelo
3 [model0, def0] = Formulacion_Baltimore(r,R);
4
5 %% Calcular funciones de respuesta en frecuencia
6 Iixh0 = nor2xf(def0, 0.005, model0, 1:1:4500, ['hz', 'dis']);
7 xf = Iixh0(:,:);
8
9 %% Función objetivo
10
11 %promedio de FRF para frecuencias en un rango de 500 Hz (en este caso 3500–4000 Hz)
12 xfp = mean(abs(xf(3500:4000,:)));
13
14 %rms calculada con promedio anterior
15 rm0 = sqrt(sum(abs(xfp).^2)/length(xfp));
16
17 val = rm0;
18 end
```

C.3. Estructura Mixta

```
1 function val = obj_Mixta(r,R)
2 %% Generar modelo
3 [model0, def0] = Formulacion_Mixta(r,R);
4
5 %% Calcular funciones de respuesta en frecuencia
6 IIXh0 = nor2xf(def0, 0.005, model0, 1:1:4500, ['hz', 'dis']);
7 xf = IIXh0(:,:);
8
9 %% Función objetivo
10
11 %promedio de FRF para frecuencias en un rango de 500 Hz (en este caso 3500–4000 Hz)
12 xfp = mean(abs(xf(3500:4000,:))');
13
14 %rms calculada con promedio anterior
15 rm0 = sqrt(sum(abs(xfp).^2)/length(xfp));
16
17 val = rm0;
18 end
```

Anexo D. Gráficas de las estructuras

Para la obtención de los gráficos, lo que se realiza es obtener los modos de vibración de la estructura nueva, y además obtener un gráfico en conjunto de la FRF original junto a la nueva. De esta forma, se tienen tres funciones para obtener las gráficas de las estructuras.

D.1. Estructura Pratt

```
1 function Graficos_Pratt(r,R)
2 %% Calcular funciones de respuesta en frecuencia
3 R0 = 0.001*ones(1,33);
4 [model0, def0] = Formulacion_Pratt(R0(1),R0(2:end));
5 IIXh0 = nor2xf(def0, 0.005, model0, 1:1:4500, ['hz', 'dis']);
6 xf0 = IIXh0(:,:);
7
8 [model1, def1] = Formulacion_Pratt(r,R);
9 IIXh1 = nor2xf(def1, 0.005, model1, 1:1:4500, ['hz', 'dis']);
10 xf1 = IIXh1(:,:);
11
12 %% Gráficos
13
14 % Graficar FRFs
15 figure;
16 semilogy(1:1:4500, mean(abs(xf0'))) % base
17 hold on
18
19 semilogy(1:1:4500, mean(abs(xf1'))) % optimizada
20 xlabel('Frecuencia (Hz)');
21 ylabel('Desplazamiento (m)');
22 title('Optimización de FRF entre 3500 y 4000 Hz'); %(en este caso)
23 legend('Formulación inicial con radios iguales','Optimización de radios')
24 hold off
25 end
```

D.2. Estructura Baltimore

```
1 function Graficos_Baltimore(r,R)
2 %% Calcular funciones de respuesta en frecuencia
3 R0 = 0.001*ones(1,97);
4 [model0, def0] = Formulacion_Baltimore(R0(1),R0(2:end));
5 IIXh0 = nor2xf(def0, 0.005, model0, 1:1:4500, ['hz', 'dis']);
6 xf0 = IIXh0(:,:);
7
8 [model1, def1] = Formulacion_Baltimore(r,R);
9 IIXh1 = nor2xf(def1, 0.005, model1, 1:1:4500, ['hz', 'dis']);
10 xf1 = IIXh1(:,:);
11
12 %% Gráficos
13
14 % Graficar FRFs
15 figure;
16 semilogy(1:1:4500, mean(abs(xf0'))) % base
17 hold on
```

```

18
19 semilogy(1:1:4500, mean(abs(xf1'))) % optimizada
20 xlabel('Frecuencia (Hz)');
21 ylabel('Desplazamiento (m)');
22 title('Optimización de FRF entre 3500 y 4000 Hz'); %(en este caso)
23 legend('Formulación inicial con radios iguales', 'Optimización de radios')
24 hold off
25 end

```

D.3. Estructura Mixta

```

1 function Graficos_Mixta(r,R)
2 %% Calcular funciones de respuesta en frecuencia
3 R0 = 0.001*ones(1,49);
4 [model0, def0] = Formulacion_Mixta(R0(1),R0(2:end));
5 Iixh0 = nor2xf(def0, 0.005, model0, 1:1:4500, ['hz', 'dis']);
6 xf0 = Iixh0(:,:);
7
8 [model1, def1] = Formulacion_Mixta(r,R);
9 Iixh1 = nor2xf(def1, 0.005, model1, 1:1:4500, ['hz', 'dis']);
10 xf1 = Iixh1(:,:);
11
12 %% Gráficos
13
14 % Graficar FRFs
15 figure;
16 semilogy(1:1:4500, mean(abs(xf0'))) % base
17 hold on
18
19 semilogy(1:1:4500, mean(abs(xf1'))) % optimizada
20 xlabel('Frecuencia (Hz)');
21 ylabel('Desplazamiento (m)');
22 title('Optimización de FRF entre 3500 y 4000 Hz'); %(en este caso)
23 legend('Formulación inicial con radios iguales', 'Optimización de radios')
24 hold off
25 end

```

Anexo E. Optimización de estructuras

E.1. Estructura Pratt

```
1 % Definir función objetivo y límites para sus parámetros
2 fun_P = @(x) obj_Pratt(x(1), x(2:end));
3
4 lb = 0.0005*ones(1,33); % Límite inferior
5 ub = 0.0015*ones(1,33); % Límite superior
6
7 % Configurar opciones de la optimización
8 options = optimoptions('particleswarm', 'SwarmSize', 50, 'Display', 'iter', 'MaxIterations', 100, '
↳ TolFun', 1e-3);
9
10 % Realizar la optimización con particleswarm
11 [u_P, fval, exitflag, output] = particleswarm(fun_P, length(lb), lb, ub, options);
12
13 % Mostrar los resultados
14 disp('Parámetros óptimos:');
15 disp(u_P);
16
17 % Calcular el valor de la función objetivo con los parámetros óptimos
18 val_P = obj_Pratt(u_P(1), u_P(2:end));
19 disp(['Valor de la función objetivo con parámetros óptimos: ', num2str(val_P)]);
20
21 % Gráfico de FRF
22 Graficos_Pratt(u_P(1), u_P(2:end));
```

E.2. Estructura Baltimore

```
1 % Definir función objetivo y límites para sus parámetros
2 fun_B = @(x) obj_Baltimore(x(1), x(2:end));
3
4 lb = 0.0005*ones(1,97); % Límite inferior
5 ub = 0.0015*ones(1,97); % Límite superior
6
7 % Configurar opciones de la optimización
8 options = optimoptions('particleswarm', 'SwarmSize', 50, 'Display', 'iter', 'MaxIterations', 100, '
↳ TolFun', 1e-3);
9
10 % Realizar la optimización con particleswarm
11 [u_B, fval, exitflag, output] = particleswarm(fun_B, length(lb), lb, ub, options);
12
13 % Mostrar los resultados
14 disp('Parámetros óptimos:');
15 disp(u_B);
16
17 % Calcular el valor de la función objetivo con los parámetros óptimos
18 val_B = obj_Baltimore(u_B(1), u_B(2:end));
19 disp(['Valor de la función objetivo con parámetros óptimos: ', num2str(val_B)]);
20
21 % Gráfico de FRF
22 Graficos_Baltimore(u_B(1), u_B(2:end));
```

E.3. Estructura Mixta

```
1     % Definir función objetivo y límites para sus parámetros
2     fun_M = @(x) obj_Mixta(x(1), x(2:end));
3
4     lb = 0.0005*ones(1,49); % Límite inferior
5     ub = 0.0015*ones(1,49); % Límite superior
6
7     % Configurar opciones de la optimización
8     options = optimoptions('particleswarm', 'SwarmSize', 50, 'Display', 'iter', 'MaxIterations', 100, '
↳ TolFun', 1e-3);
9
10    % Realizar la optimización con particleswarm
11    [u_M,fval,exitflag,output] = particleswarm(fun_M, length(lb), lb, ub, options);
12
13    % Mostrar los resultados
14    disp('Parámetros óptimos:');
15    disp(u_M);
16
17    % Calcular el valor de la función objetivo con los parámetros óptimos
18    val_M = obj_Mixta(u_M(1), u_M(2:end));
19    disp(['Valor de la función objetivo con parámetros óptimos: ', num2str(val_M)]);
20
21    % Gráfico de FRF
22    Graficos_Mixta(u_M(1), u_M(2:end));
```

Anexo F. Resultado de iteraciones

F.1. Iteraciones estructura Pratt

F.1.1. Rango 1500 - 2000 Hz

```
1 >> Optimizacion_Pratt
2
3 Iteration    f-count      Best          Mean          Stall
4             50      4.769e-06    3.051e-05     0
5             100     4.769e-06    3.916e-05     0
6             150     3.508e-06    1.614e-05     0
7             200     2.224e-06    1.006e-05     0
8             250     1.926e-06    7.542e-06     0
9             300     1.868e-06    7.287e-06     0
10            350     1.868e-06    7.203e-06     1
11            400     1.522e-06    6.725e-06     0
12            450     1.522e-06    6.584e-06     1
13            500     1.392e-06    7.065e-06     0
14            550     1.392e-06    8.941e-06     1
15            600     1.392e-06    8.427e-06     2
16            650     1.392e-06    9.352e-06     3
17            700     1.392e-06    7.152e-06     4
18            750     1.392e-06    1.132e-05     5
19            800     1.392e-06    8.487e-06     6
20            850     1.392e-06    5.707e-06     7
21            900     1.392e-06    4.769e-06     8
22            950     1.357e-06    4.407e-06     0
23           1000     1.091e-06    3.176e-06     0
24           1050     9.708e-07    2.846e-06     0
25           1100     9.708e-07    2.4e-06       1
26 Optimization ended: relative change in the objective value
27 over the last OPTIONS.MaxStallIterations iterations is less than OPTIONS.FunctionTolerance.
28 Parámetros óptimos:
29 Columns 1 through 10
30
31    0.0013  0.0011  0.0007  0.0013  0.0006  0.0008  0.0007  0.0015  0.0015  0.0008
32
33 Columns 11 through 20
34
35    0.0006  0.0008  0.0013  0.0015  0.0014  0.0006  0.0009  0.0010  0.0005  0.0014
36
37 Columns 21 through 30
38
39    0.0014  0.0015  0.0005  0.0005  0.0005  0.0008  0.0006  0.0015  0.0008  0.0005
40
41 Columns 31 through 33
42
43    0.0015  0.0005  0.0013
44
45 Valor de la función objetivo con parámetros óptimos: 9.7081e-07
```

F.1.2. Rango 2500 - 3000 Hz

```

1  >> Optimizacion_Pratt
2
3  Iteration    f-count      Best          Mean          Stall
4  0            50          3.581e-06     1.374e-05     0
5  1            100         1.541e-06     1.249e-05     0
6  2            150         3.026e-07     1.181e-05     0
7  3            200         3.026e-07     1.092e-05     1
8  4            250         1.221e-07     9.181e-06     0
9  5            300         1.221e-07     7.542e-06     1
10 6            350         1.221e-07     7.564e-06     2
11 7            400         1.177e-07     6.418e-06     0
12 8            450         1.11e-07      6.049e-06     0
13 9            500         9.931e-08     4.003e-06     0
14 10           550         7.96e-08      5.445e-06     0
15 11           600         7.96e-08      3.694e-06     1
16 12           650         7.96e-08      2.487e-06     2
17 13           700         7.96e-08      1.866e-06     3
18 14           750         7.96e-08      1.624e-06     4
19 15           800         5.907e-08     1.687e-06     0
20 16           850         5.907e-08     1.671e-06     1
21 17           900         5.643e-08     1.16e-06      0
22 18           950         4.831e-08     8.789e-07     0
23 19          1000         4.385e-08     5.926e-07     0
24 20          1050         3.701e-08     1.147e-06     0
25 21          1100         3.645e-08     7.844e-07     0
26 Optimization ended: relative change in the objective value
27 over the last OPTIONS.MaxStallIterations iterations is less than OPTIONS.FunctionTolerance.
28 Parámetros óptimos:
29 Columns 1 through 10
30
31 0.0005 0.0013 0.0005 0.0005 0.0011 0.0015 0.0005 0.0005 0.0010 0.0005
32
33 Columns 11 through 20
34
35 0.0007 0.0014 0.0015 0.0005 0.0005 0.0006 0.0015 0.0015 0.0015 0.0014
36
37 Columns 21 through 30
38
39 0.0005 0.0014 0.0005 0.0015 0.0015 0.0012 0.0015 0.0005 0.0015 0.0005
40
41 Columns 31 through 33
42
43 0.0015 0.0005 0.0005
44
45
46 Valor de la función objetivo con parámetros óptimos: 3.6451e-08

```

F.1.3. Rango 3500 - 4000 Hz

```
1 >> Optimizacion_Pratt
2
3 Iteration    f-count      Best          Mean          Stall
4             50      2.542e-06    8.879e-06
5             100     2.256e-06    1.755e-05
6             150     2.256e-06    1.153e-05
7             200     1.167e-06    1.352e-05
8             250     1.167e-06    1.63e-05
9             300     1.167e-06    1.474e-05
10            350     8.589e-07    1.391e-05
11            400     8.589e-07    1.134e-05
12            450     4.25e-07     1.84e-05
13            500     4.25e-07    1.716e-05
14            550     4.25e-07    1.403e-05
15            600     1.64e-07    1.359e-05
16            650     1.64e-07    1.337e-05
17            700     1.64e-07    1.628e-05
18            750     1.64e-07    1.204e-05
19            800     1.64e-07    1.374e-05
20            850     1.64e-07    1.255e-05
21            900     1.567e-07    7.29e-06
22            950     1.205e-07    6.518e-06
23           1000     1.172e-07    2.321e-06
24           1050     1.168e-07    1.876e-06
25           1100     9.452e-08    1.405e-06
26 Optimization ended: relative change in the objective value
27 over the last OPTIONS.MaxStallIterations iterations is less than OPTIONS.FunctionTolerance.
28 Parámetros óptimos:
29 Columns 1 through 10
30
31 0.0007 0.0015 0.0014 0.0013 0.0005 0.0012 0.0007 0.0015 0.0013 0.0005
32
33 Columns 11 through 20
34
35 0.0014 0.0005 0.0015 0.0013 0.0010 0.0006 0.0007 0.0014 0.0006 0.0010
36
37 Columns 21 through 30
38
39 0.0007 0.0012 0.0005 0.0011 0.0006 0.0007 0.0012 0.0007 0.0010 0.0005
40
41 Columns 31 through 33
42
43 0.0013 0.0015 0.0014
44
45
46 Valor de la función objetivo con parámetros óptimos: 9.4516e-08
```

F.2. Iteraciones estructura Baltimore

F.2.1. Rango 1500 - 2000 Hz

```
1 >> Optimizacion_Baltimore
2           Best           Mean   Stall
3 Iteration  f-count      f(x)      f(x)  Iterations
4     0         50  3.731e-06  3.666e-05    0
5     1        100  3.731e-06  2.894e-05    0
6     2        150  2.845e-06  1.528e-05    0
7     3        200  2.845e-06  1.754e-05    1
8     4        250  2.845e-06  1.784e-05    2
9     5        300  2.845e-06  1.683e-05    3
10    6        350  2.845e-06  1.404e-05    4
11    7        400  2.772e-06  1.581e-05    0
12    8        450  2.772e-06  2.095e-05    1
13    9        500  2.772e-06  1.854e-05    2
14   10        550  2.772e-06  1.682e-05    3
15   11        600  2.772e-06  1.098e-05    4
16   12        650  2.097e-06  7.787e-06    0
17   13        700  1.763e-06  5.492e-06    0
18   14        750  1.649e-06  5.169e-06    0
19   15        800  1.649e-06  4.59e-06     1
20   16        850  1.649e-06  3.978e-06    2
21   17        900  1.582e-06  4.549e-06    0
22   18        950  1.358e-06  3.954e-06    0
23   19       1000  1.275e-06  3.695e-06    0
24   20       1050  1.263e-06  3.21e-06     0
25   21       1100  1.156e-06  2.881e-06    0
26 Optimization ended: relative change in the objective value
27 over the last OPTIONS.MaxStallIterations iterations is less than OPTIONS.FunctionTolerance.
28 Parámetros óptimos:
29 Columns 1 through 10
30
31 0.0012 0.0008 0.0014 0.0011 0.0007 0.0010 0.0015 0.0008 0.0006 0.0009
32
33 Columns 11 through 20
34
35 0.0011 0.0013 0.0005 0.0015 0.0008 0.0007 0.0007 0.0012 0.0011 0.0015
36
37 Columns 21 through 30
38
39 0.0005 0.0011 0.0007 0.0007 0.0012 0.0015 0.0006 0.0011 0.0008 0.0014
40
41 Columns 31 through 40
42
43 0.0015 0.0005 0.0014 0.0012 0.0005 0.0011 0.0010 0.0012 0.0015 0.0006
44
45 Columns 41 through 50
46
47 0.0014 0.0012 0.0010 0.0008 0.0011 0.0008 0.0008 0.0013 0.0013 0.0008
48
49 Columns 51 through 60
50
51 0.0005 0.0008 0.0010 0.0007 0.0013 0.0005 0.0011 0.0007 0.0008 0.0008
52
```

```

53 Columns 61 through 70
54
55     0.0005  0.0011  0.0014  0.0012  0.0015  0.0006  0.0005  0.0007  0.0010  0.0008
56
57 Columns 71 through 80
58
59     0.0013  0.0013  0.0014  0.0005  0.0015  0.0006  0.0005  0.0011  0.0008  0.0014
60
61 Columns 81 through 90
62
63     0.0014  0.0014  0.0012  0.0005  0.0013  0.0014  0.0009  0.0006  0.0009  0.0007
64
65 Columns 91 through 97
66
67     0.0012  0.0011  0.0005  0.0005  0.0014  0.0010  0.0012
68
69
70 Valor de la función objetivo con parámetros óptimos: 1.1559e-06

```

F.2.2. Rango 2500 - 3000 Hz

```

1  >> Optimizacion_Baltimore
2
3  Iteration   f-count   Best          Mean          Stall
4             f(x)      f(x)          Iterations
5  0           50        3.194e-06     1.368e-05     0
6  1           100       5.518e-07     7.697e-06     0
7  2           150       5.518e-07     6.695e-06     1
8  3           200       2.813e-07     4.293e-06     0
9  4           250       2.641e-07     2.799e-06     0
10  5           300       1.93e-07      2.91e-06      0
11  6           350       1.6e-07       1.509e-06     0
12  7           400       1.217e-07     1.55e-06      0
13  8           450       9.89e-08      1.209e-06     0
14  9           500       9.89e-08      1.252e-06     1
15  10          550       4.306e-08     1.064e-06     0
16  11          600       4.306e-08     1.047e-06     1
17  12          650       4.148e-08     6.688e-07     0
18  13          700       4.148e-08     7.97e-07      1
19  14          750       4.148e-08     8.085e-07     2
20  15          800       4.148e-08     8.636e-07     3
21  16          850       4.148e-08     7.071e-07     4
22  17          900       4.148e-08     7.626e-07     5
23  18          950       3.165e-08     6.993e-07     0
24  19         1000       3.165e-08     5.554e-07     1
25  20         1050       3.165e-08     6.776e-07     2
26  21         1100       3.165e-08     3.112e-07     3
27 Optimization ended: relative change in the objective value
28 over the last OPTIONS.MaxStallIterations iterations is less than OPTIONS.FunctionTolerance.
29 Parámetros óptimos:
30 Columns 1 through 10
31
32     0.0005  0.0010  0.0009  0.0011  0.0014  0.0013  0.0009  0.0007  0.0005  0.0012
33
34 Columns 11 through 20
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70

```

```

35     0.0007  0.0005  0.0010  0.0005  0.0007  0.0006  0.0007  0.0010  0.0005  0.0010
36
37 Columns 21 through 30
38
39     0.0009  0.0008  0.0008  0.0014  0.0013  0.0014  0.0011  0.0012  0.0014  0.0012
40
41 Columns 31 through 40
42
43     0.0012  0.0009  0.0013  0.0015  0.0007  0.0006  0.0014  0.0010  0.0014  0.0012
44
45 Columns 41 through 50
46
47     0.0009  0.0010  0.0005  0.0015  0.0008  0.0005  0.0006  0.0012  0.0013  0.0015
48
49 Columns 51 through 60
50
51     0.0008  0.0009  0.0014  0.0012  0.0012  0.0012  0.0010  0.0014  0.0015  0.0013
52
53 Columns 61 through 70
54
55     0.0007  0.0008  0.0008  0.0008  0.0007  0.0015  0.0006  0.0013  0.0011  0.0012
56
57 Columns 71 through 80
58
59     0.0006  0.0006  0.0012  0.0006  0.0011  0.0010  0.0010  0.0005  0.0014  0.0012
60
61 Columns 81 through 90
62
63     0.0008  0.0015  0.0006  0.0010  0.0008  0.0009  0.0010  0.0010  0.0015  0.0015
64
65 Columns 91 through 97
66
67     0.0015  0.0011  0.0006  0.0010  0.0013  0.0014  0.0015
68
69
70 Valor de la función objetivo con parámetros óptimos: 3.1649e-08

```

F.2.3. Rango 3500 - 4000 Hz

```

1  >> Optimizacion_Baltimore
2
3  Iteration    f-count      Best          Mean          Stall
4  0            50          9.053e-07     8.141e-06     0
5  1            100         3.262e-07     1.407e-05     0
6  2            150         2.71e-07      7.261e-06     0
7  3            200         2.71e-07      1.158e-05     1
8  4            250         2.71e-07      1.053e-05     2
9  5            300         1.983e-07     8.046e-06     0
10 6            350         1.443e-07     1.241e-05     0
11 7            400         1.443e-07     1.109e-05     1
12 8            450         1.133e-07     1.136e-05     0
13 9            500         1.133e-07     1.048e-05     1
14 10           550         1.133e-07     1.138e-05     2
15 11           600         8.044e-08     1.075e-05     0
16 12           650         8.044e-08     9.487e-06     1

```

17	13	700	8.044e-08	1.015e-05	2
18	14	750	8.044e-08	1.138e-05	3
19	15	800	8.044e-08	9.095e-06	4
20	16	850	4.737e-08	1.149e-05	0
21	17	900	4.737e-08	1.228e-05	1
22	18	950	4.737e-08	1.423e-05	2
23	19	1000	4.737e-08	7.759e-06	3
24	20	1050	4.737e-08	4.361e-06	4
25	21	1100	3.342e-08	3.891e-06	0

26 Optimization ended: relative change in the objective value
27 over the last OPTIONS.MaxStallIterations iterations is less than OPTIONS.FunctionTolerance.
28 Parámetros óptimos:
29 Columns 1 through 10

30										
31	0.0007	0.0015	0.0005	0.0005	0.0014	0.0005	0.0015	0.0007	0.0015	0.0005
32										
33	Columns 11 through 20									
34										
35	0.0015	0.0005	0.0005	0.0005	0.0012	0.0015	0.0015	0.0015	0.0005	0.0015
36										
37	Columns 21 through 30									
38										
39	0.0005	0.0008	0.0005	0.0015	0.0013	0.0015	0.0012	0.0015	0.0015	0.0015
40										
41	Columns 31 through 40									
42										
43	0.0005	0.0015	0.0014	0.0005	0.0015	0.0010	0.0015	0.0013	0.0005	0.0008
44										
45	Columns 41 through 50									
46										
47	0.0015	0.0008	0.0005	0.0009	0.0013	0.0005	0.0015	0.0012	0.0015	0.0015
48										
49	Columns 51 through 60									
50										
51	0.0005	0.0006	0.0007	0.0015	0.0014	0.0015	0.0006	0.0009	0.0006	0.0005
52										
53	Columns 61 through 70									
54										
55	0.0005	0.0015	0.0013	0.0010	0.0015	0.0005	0.0009	0.0005	0.0005	0.0005
56										
57	Columns 71 through 80									
58										
59	0.0015	0.0015	0.0015	0.0012	0.0015	0.0015	0.0015	0.0005	0.0013	0.0015
60										
61	Columns 81 through 90									
62										
63	0.0010	0.0005	0.0006	0.0005	0.0005	0.0010	0.0014	0.0015	0.0015	0.0011
64										
65	Columns 91 through 97									
66										
67	0.0005	0.0014	0.0005	0.0015	0.0011	0.0006	0.0005			
68										
69										
70	Valor de la función objetivo con parámetros óptimos: 3.3418e-08									

F.3. Iteraciones estructura Mixta

F.3.1. Rango 1500 - 2000 Hz

```
1 >> Optimizacion_Mixta
2
3           Best      Mean      Stall
4 Iteration  f-count  f(x)    f(x)    Iterations
5 0          50      8.266e-06 2.892e-05 0
6 1          100     5.339e-06 2.387e-05 0
7 2          150     5.339e-06 1.502e-05 1
8 3          200     5.27e-06  9.394e-06 0
9 4          250     5.186e-06 8.086e-06 0
10 5         300     4.847e-06 7.406e-06 0
11 6         350     4.634e-06 7.72e-06  0
12 7         400     4.634e-06 7.025e-06 1
13 8         450     3.425e-06 6.723e-06 0
14 9         500     3.425e-06 7.324e-06 1
15 10        550     3.425e-06 6.75e-06  2
16 11        600     3.425e-06 5.99e-06  3
17 12        650     3.425e-06 5.735e-06 4
18 13        700     3.425e-06 5.467e-06 5
19 14        750     3.425e-06 5.278e-06 6
20 15        800     3.425e-06 5.609e-06 7
21 16        850     3.425e-06 5.087e-06 8
22 17        900     3.425e-06 4.775e-06 9
23 18        950     3.368e-06 4.597e-06 0
24 19       1000     3.328e-06 4.271e-06 0
25 20       1050     3.187e-06 4.133e-06 0
26 21       1100     3.169e-06 3.97e-06  0
27 Optimization ended: relative change in the objective value
28 over the last OPTIONS.MaxStallIterations iterations is less than OPTIONS.FunctionTolerance.
29 Parámetros óptimos:
30 Columns 1 through 10
31 0.0015 0.0013 0.0005 0.0005 0.0005 0.0009 0.0007 0.0007 0.0015 0.0009
32
33 Columns 11 through 20
34 0.0006 0.0015 0.0005 0.0015 0.0006 0.0005 0.0012 0.0007 0.0010 0.0011
35
36 Columns 21 through 30
37 0.0012 0.0015 0.0014 0.0008 0.0015 0.0005 0.0015 0.0009 0.0013 0.0015
38
39 Columns 31 through 40
40 0.0005 0.0015 0.0009 0.0015 0.0015 0.0005 0.0005 0.0010 0.0006 0.0015
41
42 Columns 41 through 49
43 0.0005 0.0007 0.0015 0.0012 0.0005 0.0005 0.0005 0.0015 0.0015
44
45 Valor de la función objetivo con parámetros óptimos: 3.1689e-06
```

F.3.2. Rango 2500 - 3000 Hz

```

1  >> Optimizacion_Mixta
2
3  Iteration    f-count      Best          Mean          Stall
4  0            50          1.685e-06     1.168e-05     0
5  1            100         9.645e-07     7.67e-06      0
6  2            150         3.492e-07     8.099e-06     0
7  3            200         3.492e-07     5.195e-06     1
8  4            250         2.961e-07     5.242e-06     0
9  5            300         2.961e-07     3.267e-06     1
10 6            350         2.354e-07     3.811e-06     0
11 7            400         1.505e-07     3.365e-06     0
12 8            450         1.289e-07     3.393e-06     0
13 9            500         1.289e-07     2.45e-06      1
14 10           550         1.289e-07     2.724e-06     2
15 11           600         1.286e-07     2.935e-06     0
16 12           650         7.482e-08     2.08e-06      0
17 13           700         4.537e-08     2.004e-06     0
18 14           750         4.537e-08     2.311e-06     1
19 15           800         4.537e-08     1.457e-06     2
20 16           850         4.537e-08     1.177e-06     3
21 17           900         4.537e-08     1.583e-06     4
22 18           950         4.537e-08     7.333e-07     5
23 19          1000         3.033e-08     6.674e-07     0
24 20          1050         2.904e-08     8.832e-07     0
25 21          1100         2.904e-08     7.532e-07     1
26 Optimization ended: relative change in the objective value
27 over the last OPTIONS.MaxStallIterations iterations is less than OPTIONS.FunctionTolerance.
28 Parámetros óptimos:
29 Columns 1 through 10
30
31 0.0005 0.0015 0.0006 0.0015 0.0015 0.0015 0.0012 0.0015 0.0005 0.0009
32
33 Columns 11 through 20
34
35 0.0005 0.0015 0.0015 0.0013 0.0005 0.0008 0.0015 0.0015 0.0015 0.0012
36
37 Columns 21 through 30
38
39 0.0007 0.0015 0.0007 0.0015 0.0015 0.0005 0.0005 0.0010 0.0007 0.0015
40
41 Columns 31 through 40
42
43 0.0005 0.0014 0.0015 0.0006 0.0011 0.0005 0.0005 0.0005 0.0005 0.0015
44
45 Columns 41 through 49
46
47 0.0011 0.0015 0.0013 0.0014 0.0015 0.0015 0.0005 0.0005 0.0005
48
49
50 Valor de la función objetivo con parámetros óptimos: 2.9036e-08

```

F.3.3. Rango 3500 - 4000 Hz

```

1  >> Optimizacion_Mixta
2
3  Iteration   f-count   Best           Mean           Stall
4             0         50   1.217e-06     7.001e-06      0
5             1         100  1.217e-06     6.757e-06      0
6             2         150  8.351e-07     6.399e-06      0
7             3         200  8.351e-07     4.514e-06      1
8             4         250  8.351e-07     4.782e-06      2
9             5         300  7.682e-07     6.653e-06      0
10            6         350  7.682e-07     5.297e-06      1
11            7         400  7.682e-07     5.22e-06       2
12            8         450  7.682e-07     5.209e-06      3
13            9         500  6.648e-07     5.622e-06      0
14           10         550  6.648e-07     6.126e-06      1
15           11         600  6.648e-07     5.586e-06      2
16           12         650  6.648e-07     6.29e-06       3
17           13         700  4.838e-07     4.478e-06      0
18           14         750  4.838e-07     5.086e-06      1
19           15         800  3.501e-07     4.315e-06      0
20           16         850  3.501e-07     4.307e-06      1
21           17         900  2.971e-07     3.936e-06      0
22           18         950  2.514e-07     3.478e-06      0
23           19        1000  2.499e-07     2.727e-06      0
24           20        1050  1.611e-07     2.581e-06      0
25           21        1100  1.611e-07     2.435e-06      1
26 Optimization ended: relative change in the objective value
27 over the last OPTIONS.MaxStallIterations iterations is less than OPTIONS.FunctionTolerance.
28 Parámetros óptimos:
29 Columns 1 through 10
30
31    0.0007  0.0006  0.0015  0.0006  0.0013  0.0014  0.0006  0.0015  0.0005  0.0005
32
33 Columns 11 through 20
34
35    0.0014  0.0005  0.0010  0.0013  0.0009  0.0011  0.0012  0.0005  0.0015  0.0014
36
37 Columns 21 through 30
38
39    0.0011  0.0009  0.0011  0.0015  0.0006  0.0015  0.0012  0.0013  0.0005  0.0013
40
41 Columns 31 through 40
42
43    0.0015  0.0013  0.0015  0.0011  0.0009  0.0007  0.0010  0.0012  0.0015  0.0015
44
45 Columns 41 through 49
46
47    0.0014  0.0011  0.0010  0.0008  0.0008  0.0010  0.0015  0.0007  0.0015
48
49
50 Valor de la función objetivo con parámetros óptimos: 1.6113e-07

```