



Universidad de Chile
Facultad de Ciencias Físicas y Matemáticas
Departamento de Ingeniería Mecánica

OPTIMIZACIÓN TOPOLÓGICA DE ENREJADOS PARA SUPRESIÓN DE
VIBRACIONES

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL MECÁNICO

IGNACIO ANDRÉS PUIGGROS TEJADA

PROFESORA GUÍA:
VIVIANA MERUANE NARANJO

PROFESOR CO-GUÍA:
RAFAEL RUIZ GARCÍA

COMISIÓN:
RUBÉN FERNÁNDEZ URRUTIA

Este trabajo ha sido parcialmente financiado mediante el Proyecto Fondecyt 1210442

SANTIAGO DE CHILE
2022

RESUMEN DE LA MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL
MECÁNICO
POR: IGNACIO ANDRÉS PUIGGROS TEJADA
FECHA: 2022
PROF. GUÍA: VIVIANA MERUANE NARANJO

OPTIMIZACIÓN TOPOLÓGICA DE ENREJADOS PARA SUPRESIÓN DE
VIBRACIONES

El desarrollo y modernización de los procesos de manufactura aditiva permiten hoy en día la elaboración de microestructuras sintéticas sumamente complejas, siendo de especial relevancia el diseño de estructuras con arquitectura celular con band-gap, es decir, una supresión total de vibraciones en un rango de frecuencias específico. En este contexto, el presente trabajo busca utilizar metodologías de optimización subrogada para diseñar estas estructuras de forma eficiente, en contraste con las usualmente costosas metodologías de alta fidelidad.

La hipótesis del trabajo es “Es posible elaborar metodologías de optimización subrogada para el diseño de enrejados bidimensionales y tridimensionales con band-gap que mejoren los costos computacionales respecto al equivalente analítico”. Por lo tanto, el objetivo del presente trabajo es el de “Diseñar metodologías de optimización subrogada que permitan la obtención de enrejados bidimensionales y tridimensionales con band-gaps que mejoren los costos computacionales respecto al equivalente analítico.”

La metodología de trabajo consistió en tres etapas principales; la obtención de una red neuronal que logre predecir band-gaps en base a los parámetros de un enrejado bidimensional, su posterior implementación en una metodología de optimización subrogada que entregue enrejados bidimensionales con band-gaps en torno a la frecuencia de diseño, y la replicación de las etapas anteriores para el caso tridimensional. Se considera algoritmos genéticos y enjambre de partículas como algoritmos de optimización global. Finalmente, se evalúan las metodologías elaboradas en su tiempo de ejecución, número de iteraciones, y características del enrejado obtenido.

Se concluye que de las combinaciones evaluadas, algoritmos genéticos con una red neuronal de predicción directa del band-gap presenta los mejores resultados. Resulta posible afirmar que la hipótesis planteada inicialmente es cierta, ya que efectivamente se logró, tanto para el caso bidimensional como tridimensional, diseñar metodologías de optimización subrogada que permiten la obtención de enrejados con band-gaps en torno a diferentes frecuencias de diseño, observándose en general mejoras significativas en los recursos computacionales requeridos respecto al caso de alta fidelidad.

Tabla de Contenido

1. Introducción	1
1.1. Hipótesis	1
1.2. Objetivos	1
1.2.1. Objetivo principal	2
1.2.2. Objetivos específicos	2
1.3. Alcances	2
2. Antecedentes	3
2.1. Materiales de arquitectura celular	3
2.2. Paneles tipo sándwich	4
2.3. Topología interna	5
2.4. Banda fonónica	8
2.5. Redes neuronales	9
2.6. Algoritmos genéticos	11
2.7. Optimización subrogada	13
3. Metodología	15
3.1. Hardware y software	15
3.2. Enrejado bidimensional	15
3.3. Enrejado tridimensional	16
3.4. Redes neuronales: Enrejado bidimensional	17
3.5. Red neuronal: Enrejado tridimensional	19
3.6. Optimización subrogada	20
3.7. Evaluación de la metodología	22
4. Resultados	26
4.1. Redes neuronales: Enrejado bidimensional	26
4.2. Redes neuronales: Enrejado tridimensional	30
4.3. Resultados metodología bidimensional: Algoritmo genético	33
4.4. Resultados metodología bidimensional: Enj. de partículas	39
4.5. Resultados metodología tridimensional	46
4.6. Resultados optimización analítica	52
4.7. Mejores enrejados: Conjunto de entrenamiento	52
4.8. Convergencia del modelo subrogado	53
5. Discusión y Análisis	58
6. Conclusión	62

7. Bibliografía	65
ANEXOS	67
ANEXO A: Gráficos	67
ANEXO B: Códigos de MATLAB	91

Lista de Tablas

1.	Propiedades de los posibles materiales de las barras en el enrejado bidimensional.	16
2.	Propiedades del material utilizado para el enrejado tridimensional.	17
3.	Rangos de variación para búsqueda aleatoria inicial (caso bidimensional). . .	18
4.	Rangos de variación para la búsqueda aleatoria refinada (caso bidimensional).	19
5.	Rangos de variación para la búsqueda aleatoria inicial.	20
4.1.	Mejor red directa.	26
4.2.	Mejor red de máximos y mínimos.	26
4.3.	Rendimiento mejores redes.	27
4.4.	Mejor red directa.	30
4.5.	Rendimiento de la red neuronal aplicada sobre enrejados de pirámide central.	31
4.6.	Resultados de la metodología con algoritmo genético al evaluar 2000 [Hz] y $\alpha = 0,5$	33
4.7.	Resultados de la metodología con algoritmo genético al evaluar 4000 [Hz] y $\alpha = 0,5$	33
4.8.	Resultados de la metodología con algoritmo genético al evaluar 6000 [Hz] y $\alpha = 0,5$	33
4.9.	Resultados de la metodología con algoritmo genético al evaluar 6000 [Hz] y $\alpha = 0,05$	34
4.10.	Resultados de la metodología con algoritmo genético al evaluar 6000 [Hz] y $\alpha = 0,25$	34
4.11.	Resultados de la metodología con algoritmo genético al evaluar 6000 [Hz] y $\alpha = 0,75$	34
4.12.	Resultados de la metodología con algoritmo genético al evaluar 6000 [Hz] y $\alpha = 0,95$	35
4.13.	Tiempos promedios hasta convergencia o máximo de iteraciones (algoritmo genético).	35
4.14.	Tiempos máximos, mínimos y promedio por iteración observados en la ejecución del algoritmo genético.	36
4.15.	Resultados de la metodología con enjambre de partículas al evaluar 2000 [Hz] y $\alpha = 0,5$	39
4.16.	Resultados de la metodología con enjambre de partículas al evaluar 4000 [Hz] y $\alpha = 0,5$	40
4.17.	Resultados de la metodología con enjambre de partículas al evaluar 6000 [Hz] y $\alpha = 0,5$	40
4.18.	Resultados de la metodología con enjambre de partículas al evaluar 6000 [Hz] y $\alpha = 0,05$	40
4.19.	Resultados de la metodología con enjambre de partículas al evaluar 6000 [Hz] y $\alpha = 0,25$	41

4.20. Resultados de la metodología con enjambre de partículas al evaluar 6000 [Hz] y $\alpha = 0,75$	41
4.21. Resultados de la metodología con enjambre de partículas al evaluar 6000 [Hz] y $\alpha = 0,95$	41
4.22. Tiempos promedios hasta convergencia o máximo de iteraciones (enjambre de partículas).	42
4.23. Tiempos máximos, mínimos y promedio por iteración observados en la ejecución del algoritmo de enjambre de partículas.	42
4.24. Resultados evaluación $5 \cdot 10^4$ [Hz] y $\alpha = 0,5$ (pirámide central).	46
4.25. Resultados evaluación $7 \cdot 10^4$ [Hz] y $\alpha = 0,5$ (pirámide central).	46
4.26. Resultados evaluación $1 \cdot 10^5$ [Hz] y $\alpha = 0,5$ (pirámide central).	46
4.27. Resultados evaluación $1,3 \cdot 10^5$ [Hz] y $\alpha = 0,5$ (pirámide central).	47
4.28. Resultados evaluación $1,5 \cdot 10^5$ [Hz] y $\alpha = 0,5$ (pirámide central).	47
4.29. Resultados evaluación $1,3 \cdot 10^5$ [Hz] y $\alpha = 0,05$ (pirámide central).	47
4.30. Resultados evaluación $1,3 \cdot 10^5$ [Hz] y $\alpha = 0,25$ (pirámide central).	48
4.31. Resultados evaluación $1,3 \cdot 10^5$ [Hz] y $\alpha = 0,75$ (pirámide central).	48
4.32. Resultados evaluación $1,3 \cdot 10^5$ [Hz] y $\alpha = 0,95$ (pirámide central).	48
4.33. Tiempos promedios hasta convergencia o máximo de iteraciones (pirámide central).	49
4.34. Tiempos máximos, mínimos y promedio por iteración observados en la ejecución del algoritmo genético (pirámide central).	49
4.35. Evaluación de la optimización analítica.	52
4.36. Funciones objetivo en los mejores elementos del conjunto de entrenamiento 2D.	52
4.37. Funciones objetivo en los mejores elementos del conjunto de entrenamiento 3D.	53

Lista de Figuras

2.1. Materiales de arquitectura celular con microestructura abierta (izquierda) y cerrada (derecha). [5]	4
2.2. Diferentes arquitecturas celulares al interior de paneles tipo sándwich.[7] . .	5
2.3. Posibles topologías y orientaciones de núcleos de paneles tipo sándwich. [8] .	6
2.4. Coordenadas de la IBZ en una celda cúbica simple (en rojo).[12]	8
2.5. Coordenadas de la IBZ en una celda cúbica centrada en las caras (en rojo). [12]	8
2.6. Salto en la banda fonónica de un material periódico, representada en las coordenadas propias del IBZ de una celda bidimensional. [14]	9
2.7. Ejemplo de la estructura de una red neuronal con n entradas, una capa oculta y una salida. [16]	10
3.8. Celda unitaria de los enrejados sobre los que se aplicará la metodología. . . .	15
3.9. (a) Esquema de la estructura 3. (b) Vista superior de la celda y su reducción por simetrías.	17
3.10. Funcionamiento general de la metodología de optimización subrogada aplicada.	22
3.11. Histograma de frecuencias de los band-gaps en el conjunto de entrenamiento bidimensional para anchos de banda superiores o iguales a 50 [Hz].	23
3.12. Histograma de frecuencias de los band-gaps en el conjunto de entrenamiento de pirámide central para anchos de banda superiores o iguales a 50 [Hz]. . .	24
4.1. Matriz de confusión de la red neuronal de predicción directa sobre enrejados bidimensionales.	27
4.2. Diagrama de dispersión de las frecuencias del band-gap con la red neuronal de predicción directa sobre enrejados bidimensionales.	28
4.3. Diagrama de dispersión del anchos de banda con la red neuronal de predicción directa sobre enrejados bidimensionales.	28
4.4. Matriz de confusión de la red neuronal de máximos y mínimos sobre enrejados bidimensionales.	29
4.5. Diagrama de dispersión de las frecuencias del band-gap con la red neuronal de máximos y mínimos sobre enrejados bidimensionales.	29
4.6. Diagrama de dispersión del anchos de banda con la red neuronal de máximos y mínimos sobre enrejados bidimensionales.	30
4.7. Matriz de confusión de la red neuronal aplicada sobre enrejados de pirámide central.	31
4.8. Diagrama de dispersión de las frecuencias del band gap obtenido con la red neuronal aplicada sobre enrejados de pirámide central.	32
4.9. Diagrama de dispersión del ancho de banda obtenido con la red neuronal aplicada sobre enrejados de pirámide central.	32
4.10. Ejemplo de diagrama de bandas con band-gap obtenido para 2000 [Hz] y $\alpha = 0,5$.	36

4.11. Ejemplo de diagrama de bandas con band-gap obtenido para 6000 [Hz] y $\alpha = 0,25$	37
4.12. Ejemplo de evolución en iteraciones de la frecuencia de alta fidelidad del óptimo (azul) y de la predicha por la red (rojo) para 2000 [Hz] y $\alpha = 0,5$	37
4.13. Ejemplo de evolución en iteraciones de la frecuencia de alta fidelidad del óptimo (azul) y de la predicha por la red (rojo) para 6000 [Hz] y $\alpha = 0,05$	38
4.14. Ejemplo de evolución del incumplimiento de la condición de término (azul) para 4000 [Hz] y $\alpha = 0,5$ (Algoritmo genético).	38
4.15. Ejemplo de evolución del incumplimiento de la condición de término (azul) para 6000 [Hz] y $\alpha = 0,05$ (Algoritmo genético).	39
4.16. Ejemplo de diagrama de bandas con band-gap obtenido para 4000 [Hz] y $\alpha = 0,5$ (Enjambre de partículas).	43
4.17. Ejemplo de diagrama de bandas con band-gap obtenido para 6000 [Hz] y $\alpha = 0,25$ (Enjambre de partículas).	43
4.18. Ejemplo de evolución en iteraciones de la frecuencia de alta fidelidad del óptimo (azul) y de la predicha por la red (rojo) para 4000 [Hz] y $\alpha = 0,5$ (Enjambre de partículas).	44
4.19. Ejemplo de evolución en iteraciones de la frecuencia de alta fidelidad del óptimo (azul) y de la predicha por la red (rojo) para 6000 [Hz] y $\alpha = 0,75$ (Enjambre de partículas).	44
4.20. Ejemplo de evolución del incumplimiento de la condición de término (azul) para 4000 [Hz] y $\alpha = 0,5$ (Enjambre de partículas).	45
4.21. Ejemplo de evolución del incumplimiento de la condición de término (azul) para 6000 [Hz] y $\alpha = 0,5$ (Enjambre de partículas).	45
4.22. Ejemplo de diagrama de bandas con band-gap obtenido para $1,3 \cdot 10^5$ [Hz] y $\alpha = 0,5$ (pirámide central).	50
4.23. Ejemplo de diagrama de bandas con band-gap obtenido para $1,5 \cdot 10^5$ [Hz] y $\alpha = 0,5$ (pirámide central).	50
4.24. Ejemplo de evolución en iteraciones de la frecuencia de alta fidelidad del óptimo (azul) y de la predicha por la red (rojo) para $7 \cdot 10^4$ [Hz] y $\alpha = 0,5$ (pirámide central).	51
4.25. Ejemplo de evolución del incumplimiento de la condición de término (azul) para $5 \cdot 10^4$ [Hz] y $\alpha = 0,5$ (Pirámide central).	51
4.26. Evolución de la condición de detención a lo largo de 500 iteraciones para la metodología 2D con algoritmo genético.	53
4.27. Evolución de la condición de detención a lo largo de 500 iteraciones para la metodología 3D.	54
4.28. Evolución del error normalizado de la frecuencia a lo largo de 500 iteraciones para la metodología 2D con algoritmo genético.	55
4.29. Evolución del error normalizado del ancho a lo largo de 500 iteraciones para la metodología 2D con algoritmo genético.	55

4.30. Evolución del error normalizado a lo largo de 500 iteraciones para la metodología 2D con algoritmo genético.	56
4.31. Evolución del error normalizado de la frecuencia a lo largo de 500 iteraciones para la metodología 3D.	56
4.32. Evolución del error normalizado del ancho a lo largo de 500 iteraciones para la metodología 3D.	57
4.33. Evolución del error normalizado a lo largo de 500 iteraciones para la metodología 3D.	57
7.34. Ejemplo de diagrama de bandas con band-gap obtenido para $5 \cdot 10^4$ [Hz] y $\alpha = 0,5$ (pirámide central).	67
7.35. Ejemplo de diagrama de bandas con band-gap obtenido para $7 \cdot 10^4$ [Hz] y $\alpha = 0,5$ (pirámide central).	68
7.36. Ejemplo de diagrama de bandas con band-gap obtenido para 10^5 [Hz] y $\alpha = 0,5$ (pirámide central).	68
7.37. Ejemplo de diagrama de bandas con band-gap obtenido para $1,3 \cdot 10^5$ [Hz] y $\alpha = 0,5$ (pirámide central).	69
7.38. Ejemplo de diagrama de bandas con band-gap obtenido para $1,3 \cdot 10^5$ [Hz] y $\alpha = 0,25$ (pirámide central).	69
7.39. Ejemplo de diagrama de bandas con band-gap obtenido para $1,3 \cdot 10^5$ [Hz] y $\alpha = 0,75$ (pirámide central).	70
7.40. Ejemplo de diagrama de bandas con band-gap obtenido para $1,5 \cdot 10^5$ [Hz] y $\alpha = 0,5$ (pirámide central).	70
7.41. Ejemplo de evolución en iteraciones de la frecuencia real del óptimo (azul) y de la predicha por la red (rojo) para $5 \cdot 10^4$ [Hz] y $\alpha = 0,5$ (pirámide central).	71
7.42. Ejemplo de evolución en iteraciones de la frecuencia real del óptimo (azul) y de la predicha por la red (rojo) para $7 \cdot 10^4$ [Hz] y $\alpha = 0,5$ (pirámide central).	71
7.43. Ejemplo de evolución en iteraciones de la frecuencia real del óptimo (azul) y de la predicha por la red (rojo) para 10^5 [Hz] y $\alpha = 0,5$ (pirámide central).	72
7.44. Ejemplo de evolución en iteraciones de la frecuencia real del óptimo (azul) y de la predicha por la red (rojo) para $1,3 \cdot 10^5$ [Hz] y $\alpha = 0,5$ (pirámide central).	72
7.45. Ejemplo de evolución en iteraciones de la frecuencia real del óptimo (azul) y de la predicha por la red (rojo) para $1,3 \cdot 10^5$ [Hz] y $\alpha = 0,75$ (pirámide central).	73
7.46. Ejemplo de evolución en iteraciones de la frecuencia real del óptimo (azul) y de la predicha por la red (rojo) para $1,5 \cdot 10^5$ [Hz] y $\alpha = 0,5$ (pirámide central).	73
7.47. Ejemplo de la evolución del valor alcanzado de la función objetivo para $5 \cdot 10^4$ [Hz] y $\alpha = 0,5$ (pirámide central).	74
7.48. Ejemplo de la evolución del valor alcanzado de la función objetivo para $7 \cdot 10^4$ [Hz] y $\alpha = 0,5$ (pirámide central).	74
7.49. Ejemplo de la evolución del valor alcanzado de la función objetivo para 10^5 [Hz] y $\alpha = 0,5$ (pirámide central).	75

7.50. Ejemplo de la evolución del valor alcanzado de la función objetivo para $1,3 \cdot 10^5$ [Hz] y $\alpha = 0,5$ (pirámide central).	75
7.51. Ejemplo de la evolución del valor alcanzado de la función objetivo para $1,3 \cdot 10^5$ [Hz] y $\alpha = 0,75$ (pirámide central).	76
7.52. Ejemplo de la evolución del valor alcanzado de la función objetivo para $1,5 \cdot 10^5$ [Hz] y $\alpha = 0,5$ (pirámide central).	76
7.53. Ejemplo de diagrama de bandas con band-gap obtenido para 2000 [Hz] y $\alpha = 0,5$.	77
7.54. Ejemplo de diagrama de bandas con band-gap obtenido para 4000 [Hz] y $\alpha = 0,5$.	77
7.55. Ejemplo de diagrama de bandas con band-gap obtenido para 6000 [Hz] y $\alpha = 0,5$.	78
7.56. Ejemplo de diagrama de bandas con band-gap obtenido para 6000 [Hz] y $\alpha = 0,25$	78
7.57. Ejemplo de diagrama de bandas con band-gap obtenido para 6000 [Hz] y $\alpha = 0,75$	79
7.58. Ejemplo de evolución en iteraciones de la frecuencia real del óptimo (azul) y de la predicha por la red (rojo) para 2000 [Hz] y $\alpha = 0,5$	79
7.59. Ejemplo de evolución en iteraciones de la frecuencia real del óptimo (azul) y de la predicha por la red (rojo) para 4000 [Hz] y $\alpha = 0,5$	80
7.60. Ejemplo de evolución en iteraciones de la frecuencia real del óptimo (azul) y de la predicha por la red (rojo) para 6000 [Hz] y $\alpha = 0,5$	80
7.61. Ejemplo de evolución en iteraciones de la frecuencia real del óptimo (azul) y de la predicha por la red (rojo) para 6000 [Hz] y $\alpha = 0,25$	81
7.62. Ejemplo de evolución en iteraciones de la frecuencia real del óptimo (azul) y de la predicha por la red (rojo) para 6000 [Hz] y $\alpha = 0,75$	81
7.63. Ejemplo de la evolución del valor alcanzado de la función objetivo para 2000 [Hz] y $\alpha = 0,5$	82
7.64. Ejemplo de la evolución del valor alcanzado de la función objetivo para 4000 [Hz] y $\alpha = 0,5$	82
7.65. Ejemplo de la evolución del valor alcanzado de la función objetivo para 6000 [Hz] y $\alpha = 0,5$	83
7.66. Ejemplo de la evolución del valor alcanzado de la función objetivo para 6000 [Hz] y $\alpha = 0,25$	83
7.67. Ejemplo de la evolución del valor alcanzado de la función objetivo para 6000 [Hz] y $\alpha = 0,75$	84
7.68. Ejemplo de diagrama de bandas con band-gap obtenido para 4000 [Hz] y $\alpha = 0,5$ (Enjambre de partículas).	85
7.69. Ejemplo de diagrama de bandas con band-gap obtenido para 6000 [Hz] y $\alpha = 0,5$ (Enjambre de partículas).	85
7.70. Ejemplo de diagrama de bandas con band-gap obtenido para 6000 [Hz] y $\alpha = 0,25$ (Enjambre de partículas).	86
7.71. Ejemplo de diagrama de bandas con band-gap obtenido para 6000 [Hz] y $\alpha = 0,75$ (Enjambre de partículas).	86

7.72. Ejemplo de evolución en iteraciones de la frecuencia real del óptimo (azul) y de la predicha por la red (rojo) para 4000 [Hz] y $\alpha = 0,5$ (Enjambre de partículas).	87
7.73. Ejemplo de evolución en iteraciones de la frecuencia real del óptimo (azul) y de la predicha por la red (rojo) para 6000 [Hz] y $\alpha = 0,5$ (Enjambre de partículas).	87
7.74. Ejemplo de evolución en iteraciones de la frecuencia real del óptimo (azul) y de la predicha por la red (rojo) para 6000 [Hz] y $\alpha = 0,25$ (Enjambre de partículas).	88
7.75. Ejemplo de evolución en iteraciones de la frecuencia real del óptimo (azul) y de la predicha por la red (rojo) para 6000 [Hz] y $\alpha = 0,75$ (Enjambre de partículas).	88
7.76. Ejemplo de la evolución del valor alcanzado de la función objetivo para 4000 [Hz] y $\alpha = 0,5$ (Enjambre de partículas).	89
7.77. Ejemplo de la evolución del valor alcanzado de la función objetivo para 6000 [Hz] y $\alpha = 0,5$ (Enjambre de partículas).	89
7.78. Ejemplo de la evolución del valor alcanzado de la función objetivo para 6000 [Hz] y $\alpha = 0,25$ (Enjambre de partículas).	90
7.79. Ejemplo de la evolución del valor alcanzado de la función objetivo para 6000 [Hz] y $\alpha = 0,75$ (Enjambre de partículas).	90

1. Introducción

El desarrollo y modernización de los procesos de manufactura aditiva permiten hoy en día la elaboración de microestructuras sintéticas sumamente complejas, lo que ha impulsado la investigación y desarrollo en torno a los materiales de arquitectura celular. Por materiales de arquitectura celular, se entiende aquellos cuya estructura interna, usualmente periódica, se compone por una con una fase sólida y una fase gaseosa o vacía. Estos pueden presentar excelentes propiedades mecánicas con un bajo peso, lo que los hace ideales para aplicaciones como componentes de satélites, aeronaves, transbordadores espaciales, navíos, automóviles, y sistemas de energía solar, entre otros.

La complejidad de las microestructuras que se puede alcanzar en estos materiales no solo resulta de suma utilidad para obtener excelentes propiedades mecánicas, sino que permite modificar incluso sus propiedades térmicas o de transmisión de vibraciones. En particular, resulta sumamente relevante la posibilidad de diseñar estructuras con arquitectura celular para obtener un salto en la banda fonónica o band-gap, es decir, una supresión total de vibraciones en un rango de frecuencias específico, lo que tiene incontables aplicaciones en sistemas donde se busca aislar ciertos componentes o subsistemas de perturbaciones mecánicas externas o internas.

En este contexto y dadas las potenciales aplicaciones, es de especial interés la utilización de metodologías o procedimientos para el diseño de estructuras de arquitectura celular que permitan la supresión de vibraciones en torno a una frecuencia de trabajo. Si bien existe la opción de implementar metodologías total o parcialmente analíticas, estas se asocian con altos costos computacionales y tiempos de ejecución, lo que frecuentemente las hace poco prácticas e ineficientes. En vista de lo anterior, el presente trabajo busca investigar la posibilidad de utilizar metodologías de optimización subrogada para el diseño de enrejados con band-gaps en torno a un valor específico, con especial énfasis en potencial ahorro de recursos y tiempo que podrían significar en comparación con la alternativa analítica.

1.1. Hipótesis

La hipótesis principal de la investigación es: “Es posible elaborar metodologías de optimización subrogada para el diseño de enrejados bidimensionales y tridimensionales con band-gap que mejoren los costos computacionales respecto al equivalente analítico.”

1.2. Objetivos

1.2.1. Objetivo principal

Diseñar metodologías de optimización subrogada que permitan la obtención de enrejados bidimensionales y tridimensionales con band-gaps que mejoren los costos computacionales respecto al equivalente analítico.

1.2.2. Objetivos específicos

- Desarrollar, entrenar y evaluar las redes neuronales a utilizar en las metodologías a evaluar.
- Elaborar metodologías de optimización subrogada para el caso bidimensional con diferentes algoritmos de optimización global.
- Elaborar una metodología de optimización subrogada para el caso tridimensional.
- Evaluar el desempeño de las metodologías construidas.

1.3. Alcances

El presente trabajo se realizará tomando en cuenta lo siguiente:

- Para los casos bidimensional y tridimensional, se trabajará sobre un tipo de enrejado específico, detallado en la sección de metodología.
- Se considerarán algoritmos genéticos y enjambre de partículas como los posibles algoritmos de optimización global.
- Como modelos subrogados, se considerará una red neuronal que prediga directamente la frecuencia y ancho de banda de los enrejados, y una red neuronal que prediga los máximos y mínimos de sus bandas de frecuencia.
- Se entrenará a los modelos subrogados con un conjunto de 20000 enrejados, cuya obtención es previa e independiente del presente trabajo.
- Los modelos analíticos con los que se trabajará se desarrollaron en trabajos anteriores por Vicente Galvez [1] y Leonel Quinteros [2], por lo que su elaboración no se incluye en el presente trabajo.

2. Antecedentes

2.1. Materiales de arquitectura celular

En el campo de estudio los materiales de arquitectura celular, se entiende por estos a los materiales cuya estructura interna se compone por una distribución usualmente porosa, con una fase sólida y una fase gaseosa o vacía. Uno de los usos más extendidos y relevantes de estos materiales es el de constituir estructuras altamente ligeras y resistentes, superando los resultados alcanzables con materiales tradicionales. Producto de sus únicas propiedades, estos materiales tienen potencial para aplicaciones en la industria automotriz, medicina, defensa, construcción, e industria aeroespacial, entre otros [3].

El estudio de los materiales de arquitectura celular ha sido altamente influenciado y potenciado por estructuras con estas características que se encuentran en sistemas biológicos [4], tales como huesos, esponjas, o madera, que presentan interesantes propiedades a nivel macroscópico principalmente por sus complejas estructuras internas evolucionadas a través de miles de generaciones.

Por otro lado, las técnicas modernas de manufactura aditiva igualmente impulsan la investigación y desarrollo en torno a los materiales de arquitectura celular, ya que permiten la elaboración de estructuras con topologías internas altamente complejas, diseñadas para obtener propiedades específicas a nivel macroscópico. Esto potencia su utilidad más allá de las excelentes propiedades mecánicas en relación con su densidad, ya que permitiría la obtención de comportamientos más complejos, como por ejemplo modificaciones en la forma en la que se transmiten las vibraciones a través de la estructura.

La microestructura de los materiales de arquitectura celular suele tener una geometría periódica bien definida, determinada por un limitado conjunto de parámetros de diseño, aunque existe la posibilidad de que esta tenga un carácter estocástico o aleatorio. Producto de sus complejas estructuras internas, estos materiales se comportan como estructuras a nivel microscópico, y como materiales homogéneos a nivel macroscópico (si la misma microestructura se repite a lo largo del material). Cuando la microestructura de estos materiales se compone principalmente por barras unidas por nodos, también se les denomina materiales celulares basados en enrejados (“lattice-based cellular materials” en inglés).

La unidad geométrica de la microestructura de los materiales de arquitectura celular periódicos se denomina celda unitaria o simplemente celda. La posibilidad de modificar la microestructura del material mediante cambios específicos a los parámetros geométricos que definen la celda permite diseñarlos con un amplio rango de propiedades, razón por la que resultan sumamente versátiles y aplicables a múltiples rubros. Inclusive, se pueden obtener

materiales de arquitectura celular cuyas propiedades emergen puramente de su estructura, no siendo predecibles únicamente por sus constituyentes, por lo que en ciertos casos resulta posible incluirlos en la categoría de metamateriales.

Las celdas se pueden clasificar en abiertas o cerradas; si la microestructura es tal que los espacios vacíos o gaseosos de diferentes celdas se encuentran interconectados, entonces la celda es abierta, y si por otro lado la celda está delimitada por material sólido, impidiendo así la conexión entre espacios vacíos, entonces esta es cerrada.

A continuación, en la figura 2.2, se muestran materiales celulares con ambos tipos de celdas:

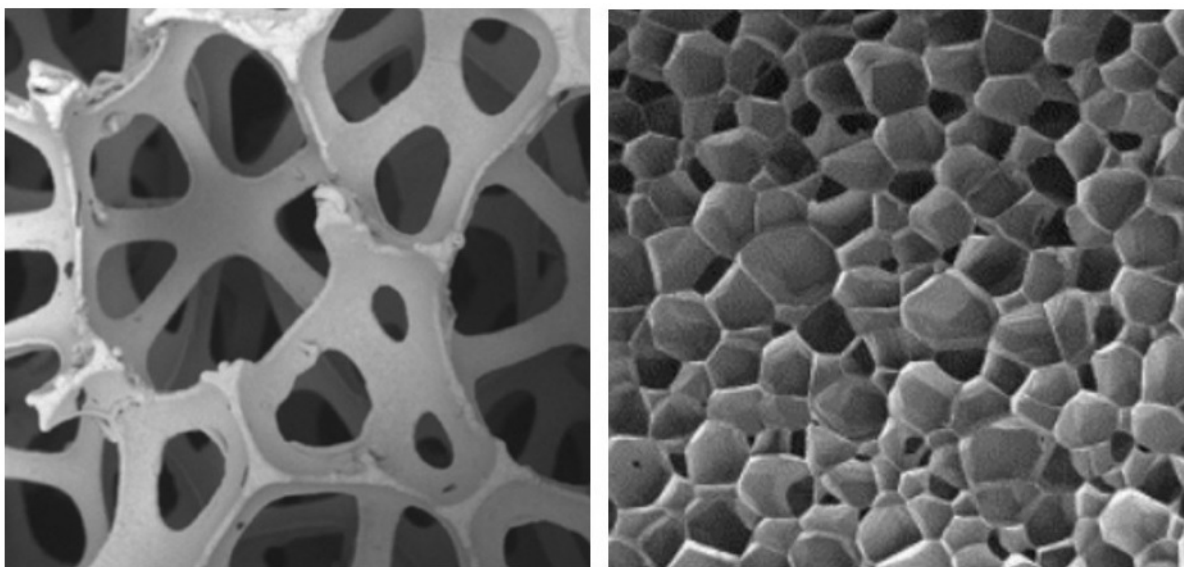


Figura 2.1: Materiales de arquitectura celular con microestructura abierta (izquierda) y cerrada (derecha). [5]

2.2. Paneles tipo sándwich

Una de las aplicaciones más importantes de los materiales celulares basados en enrejados es la de constituir los núcleos de paneles tipo sándwich. Las estructuras tipo sándwich consisten en láminas o capas delgadas y paralelas entre sí, en cuya separación se encuentra un núcleo ligero que mejora y define en gran medida sus propiedades mecánicas. Gracias a la alta relación rigidez/densidad, los paneles y estructuras tipo sándwich resultan sumamente útiles en aplicaciones donde es fundamental cumplir con condiciones estructurales con limitaciones de peso significativas, como en satélites, aeronaves, transbordadores espaciales, navíos, automóviles, sistemas de energía solar, entre otros [6].

A continuación, en la figura 2.2, se muestran ejemplos de paneles tipo sándwich con núcleos de arquitectura celular:

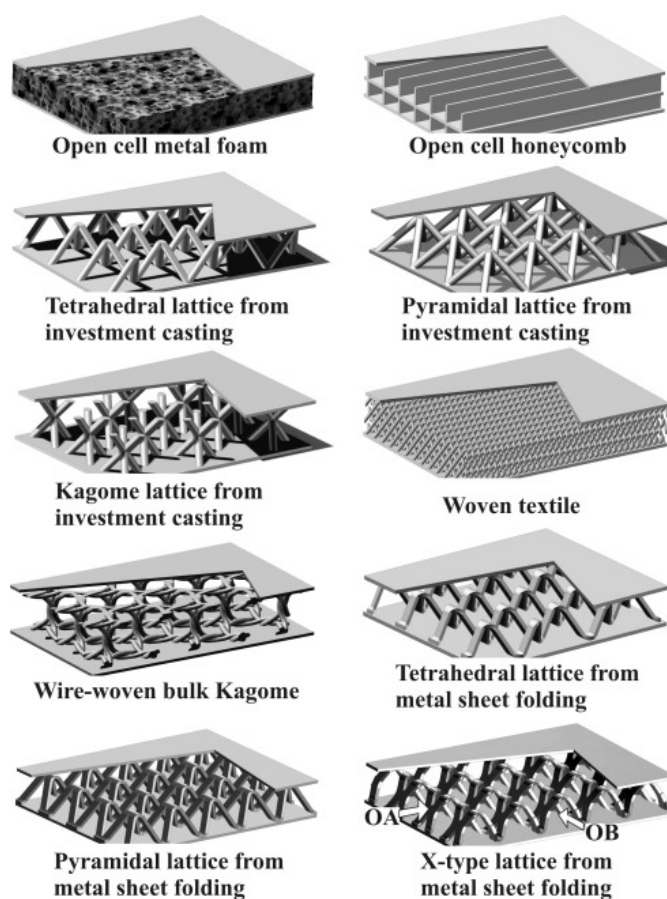


Figura 2.2: Diferentes arquitecturas celulares al interior de paneles tipo sándwich.[7]

2.3. Topología interna

La topología interna de los materiales de arquitectura celular, es decir la geometría que define su microestructura, puede ser periódica o estocástica. En el primer caso, como ya se detalló, la topología interna está definida por la celda unitaria y sus parámetros geométricos, la que a su vez, puede estar basada en diferentes estructuras y ordenamientos internos. Algunos ejemplos de esto son celdas con estructuras tipo panel hexagonal, celdas con prismas huecos, estructuras augéticas (i.e. módulo de Poisson negativo), celdas con superficies constituidas por curvas, celdas basadas en enrejados, entre otras.

A continuación, en la Figura 2.3, se muestran ejemplos de posibles topologías de núcleos de paneles tipo sándwich y diferentes posibles orientaciones:

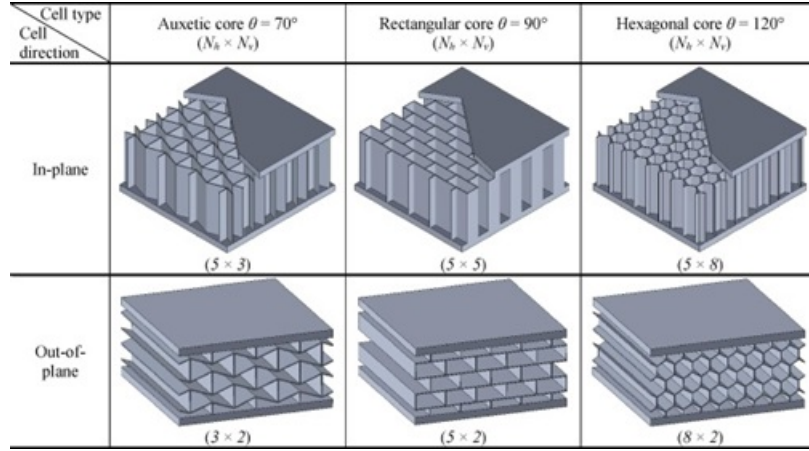


Figura 2.3: Posibles topologías y orientaciones de núcleos de paneles tipo sándwich. [8]

En particular, cuando se tiene estructuras con arquitectura celular basadas en enrejados, es posible aplicar herramientas teóricas de geometría y cristalografía para describir su topología interna y diferentes fenómenos físicos, tales como la transferencia de calor o transmisión de vibraciones.

Para la descripción de enrejados resulta útil la formulación del enrejado de Bravais [9][10], que corresponde a su definición mediante la determinación de la posición de los nodos. En estricto rigor, el enrejado de Bravais consiste en el conjunto infinito de puntos que corresponden a los nodos de un enrejado periódico, y se genera mediante traslaciones discretas específicas. En particular, el conjunto de estos puntos se define como:

$$R = n_1 a_1 + n_2 a_2 + n_3 a_3 \quad (2.1)$$

Donde:

- R : Conjunto de puntos pertenecientes al enrejado infinito teórico.
- n_i : Números enteros que parametrizan y definen los puntos del enrejado teórico.
- a_i : Vectores que determinan las direcciones de las traslaciones que generan el enrejado. Linealmente independientes pero no necesariamente ortogonales.

Se puede definir la celda de Wigner-Seitz en torno a un punto del enrejado como el conjunto de puntos que están a una menor distancia de dicho punto que de cualquier otro que se genere por las traslaciones periódicas del enrejado.

En particular, para el análisis de fenómenos físicos sobre el enrejado resulta útil definir el enrejado recíproco. El enrejado recíproco se corresponde con un enrejado en el espacio de las frecuencias (también llamado recíproco), constituido por vectores de onda que generan ondas

planas con la misma periodicidad que el enrejado original. Este enrejado se puede definir de la siguiente forma:

$$G = m_1 b_1 + m_2 b_2 + m_3 b_3 \quad (2.2)$$

Donde:

- G : Conjunto de puntos pertenecientes al enrejado recíproco en el espacio de frecuencias.
- m_i : Números enteros que parametrizan y definen los puntos del enrejado teórico.
- b_i : Vectores de onda que generan el enrejado en el espacio de frecuencias.

Se tiene que:

$$b_i = k_i e_i \quad (2.3)$$

Donde:

- k_i : Número de onda angular del vector de onda i .
- e_i : Vector unitario perpendicular al plano formado por el origen y los puntos a_j con $i \neq j$.

A partir del concepto de enrejado recíproco, resulta útil definir la primera zona de Brillouin [11][12], llamada también simplemente zona de Brillouin. La zona de Brillouin es una celda única bien definida en el espacio de frecuencias, correspondiente a la celda de Wigner-Seitz del espacio recíproco. Cuando la primera zona de Brillouin se reduce por todas las simetrías aplicables del grupo puntual (reflexiones, rotaciones, y rotaciones impropias), se obtiene la zona de Brillouin irreducible, o IBZ en inglés.

La IBZ resulta de utilidad, entre otras aplicaciones, para analizar y describir la transmisión de vibraciones a través del enrejado. Para esto, dependiendo de la celda resultante que corresponda a la IBZ, se suele asignar un set de coordenadas único para este propósito. A continuación, en las figuras 2.4 y 2.5, se muestran ejemplos de estas coordenadas para diferentes tipos de celdas:

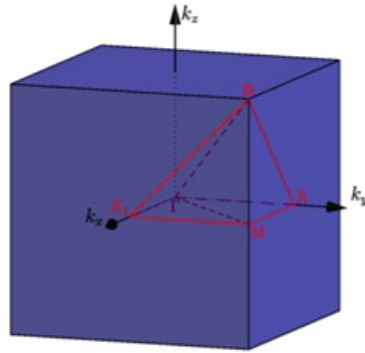


Figura 2.4: Coordenadas de la IBZ en una celda cúbica simple (en rojo).[12]

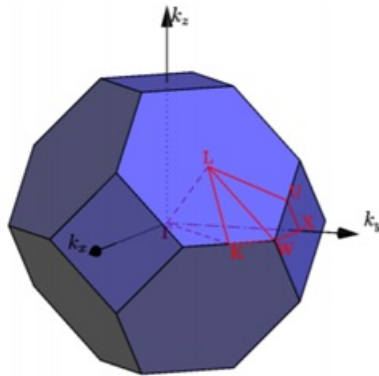


Figura 2.5: Coordenadas de la IBZ en una celda cúbica centrada en las caras (en rojo). [12]

2.4. Banda fonónica

Una aplicación sumamente relevante de los paneles tipo sándwich con núcleos de arquitectura celular es la de darles uso como materiales fonónicos. Por materiales fonónicos, se entiende aquellos con una microestructura periódica que les otorga propiedades que controlan y manipulan la propagación de ondas elásticas o acústicas.

En particular, se ha demostrado que, en ciertas condiciones, materiales periódicos basados en enrejados pueden presentar saltos en la banda fonónica (phononic band-gaps en inglés). Estos saltos de banda consisten en la supresión de la transmisión de ondas elásticas a través del material en un rango de frecuencias específico [13], por lo que resultan de suma utilidad en aplicaciones que requieran aislar partes de un sistema de vibraciones externas o internas, en particular si se busca aislar un intervalo específico de frecuencias.

Para representar las propiedades de transmisión de ondas elásticas a través de un material periódico basado en enrejado, se suele utilizar el sistema de coordenadas propio de la IBZ. A

continuación, en la Figura 2.6, se muestra un ejemplo del salto en la banda fonónica que se genera en un material periódico:

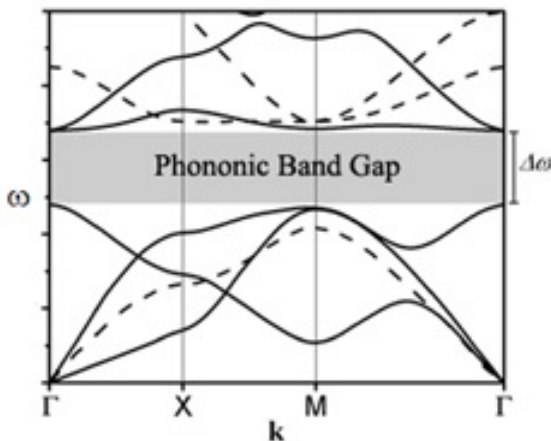


Figura 2.6: Salto en la banda fonónica de un material periódico, representada en las coordenadas propias del IBZ de una celda bidimensional. [14]

2.5. Redes neuronales

Las redes neuronales artificiales son modelos computacionales inspirados en las redes biológicas de neuronas interconectadas que existen en los sistemas nerviosos de diferentes organismos. Al igual que en el caso biológico, las redes neuronales artificiales tienen como unidad básica a un equivalente computacional de la neurona, capaz de conectarse unas con otras y de contener un valor numérico análogo a la activación biológica [15].

La estructura de una red neuronal artificial suele estructurarse en capas de neuronas, la primera de las cuales es la capa de entrada, conformada por una neurona por cada parámetro que recibe el modelo que se desea construir. La última capa de la red neuronal es la capa de salida, que puede estar conformada por una o múltiples neuronas y corresponde a los valores de salida que entrega el modelo. El resto de capas se ubica entre las capas de entrada y salida, y se les conoce como capas ocultas. Estas capas ocultas conectan los valores de entrada del modelo con la salida, e idealmente logran mediante estas conexiones que el modelo se comporte de la forma deseada. A continuación, en la Figura 2.7 se muestra un ejemplo de estructura de red neuronal:

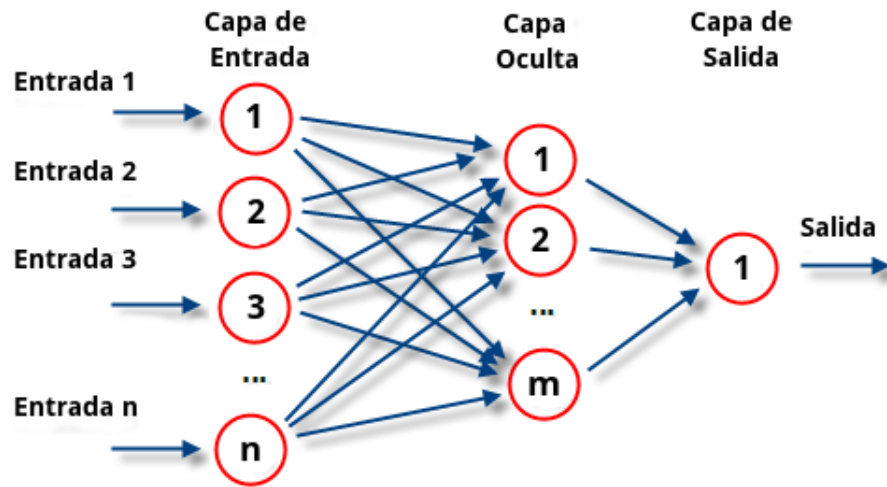


Figura 2.7: Ejemplo de la estructura de una red neuronal con n entradas, una capa oculta y una salida. [16]

Como se muestra en la figura 2.7, cada neurona se conecta con todas las neuronas de las capas adyacentes y solo con estas neuronas. Estas conexiones representan la influencia de cada neurona sobre la activación de las de la siguiente capa. Esta relación entre neuronas es unidireccional, es decir, las neuronas más cercanas a la capa de entrada influyen la activación de la capa adyacente más cercana a la capa de salida.

Matemáticamente, para determinar la activación o valor de una neurona, se realiza una suma ponderada sobre los valores de las neuronas de la capa anterior y se añade un valor constante llamado sesgo o “bias”. Esquemáticamente, el valor que se le asigna a la conexión entre neuronas corresponde al peso que la neurona emisora (capa anterior) tendrá en la suma ponderada en la neurona receptora (capa siguiente). Una vez realizada esta suma, se introduce en una función de activación que entrega un valor entre 0 y 1, siendo este valor la salida de la neurona que influye en las siguientes capas. De esta forma, mediante las conexiones ponderadas, la capa de entrada transmite la información a las siguientes capas, e idealmente, si los pesos y estructura general de la red neuronal están bien ajustadas, en la capa final se obtendrán los resultados esperados por el modelo.

Para lograr que una red neuronal artificial ajuste los pesos de sus conexiones de forma presentar el comportamiento deseado para el modelo, se entrena exponiéndola a un conjunto de datos de salida y entrada correctamente relacionados. Este entrenamiento consiste en calcular el error en la salida de la red neuronal y aplicar retropropagación o “backpropagation” en inglés, algoritmo que modifica los pesos de las conexiones en función de la magnitud en la que influyen en el error observado. Este proceso se repite sobre el conjunto de datos disponible para entrenar la red y ajustar su comportamiento.

Más específicamente, para entrenar una red neuronal se suele dividir el conjunto de datos

disponibles en tres sub-conjuntos; Conjunto de entrenamiento, conjunto de validación, y conjunto de evaluación. El conjunto de entrenamiento es el que se utiliza de forma reiterada para ajustar los pesos de la red neuronal, llamando usualmente época a un pase completo por este conjunto. El conjunto de validación se utiliza durante el entrenamiento para periódicamente exponer a la red a datos con los que no se ha entrenado previamente, de forma de evitar un sobreajuste a los datos de entrenamiento o “overfitting”. Por último, una vez se concluye el entrenamiento de la red, se utiliza el conjunto de evaluación para medir el rendimiento de la red sobre elementos a los que no ha sido expuesta.

Este mecanismo general de entrenamiento se puede aplicar con múltiples variaciones y adiciones que se ajusten a aplicaciones específicas o a conjuntos de datos de diferente tamaño. Dentro de estas variaciones se incluye utilizar diferentes números de capas, funciones de activación, número de neuronas por capa o factor de aprendizaje inicial (relacionado con el proceso de ajuste de pesos). También se puede incluir capas de “drop-out” (desactivando neuronas durante el entrenamiento para evitar dependencia de un número limitado de neuronas), mini-lotes o subdivisiones del conjunto de entrenamiento asociados a más de un ajuste de pesos por época, o diferentes programas de entrenamiento, entre otros. En general, los parámetros asociados a estas posibles variaciones se conocen como hyperparámetros de la red neuronal.

2.6. Algoritmos genéticos

Los algoritmos genéticos son una herramienta del área de la inteligencia artificial inspirada en el proceso de selección natural, y son comúnmente utilizados en problemas de optimización. En su funcionamiento, incluyen entidades y operaciones que emulan a las biológicas tales como población, individuo, función fitness, cromosomas y mutación, entre otros [17].

En primer lugar, estos algoritmos requieren codificar los candidatos a solución en base a los parámetros que los definen. Esto se conoce como la representación genética de la solución, y cada representación corresponde a un individuo en el algoritmo genético. El algoritmo se inicia mediante la construcción de una población inicial, usualmente conformada por un número lo suficientemente grande de individuos generados de forma aleatoria o semi-aleatoria. El tamaño necesario de la población inicial depende de la naturaleza del problema a optimizar.

Posteriormente, se procede a generar los individuos que conforma a la segunda generación. Para esto, se seleccionan pares de individuos de la primera generación (progenitores) que mediante las operaciones de mutación y recombinación (también llamada crossover), generan los individuos de la segunda (hijos). Para cada individuo de la nueva generación, los padres se generan de forma independiente. El mecanismo de selección varía dependiendo de la naturaleza del problema, pero en general tienden favorecer que las características (genes o cromosomas) de las soluciones con mejor desempeño se presenten en la siguiente generación

por sobre las características asociadas a menores rendimientos. En particular, este desempeño se mide usando una “función fitness” que depende de los parámetros del individuo, y que biológicamente corresponde a la capacidad de supervivencia y reproducción.

La operación de crossover o recombinación consiste en combinar los parámetros (genes) de los progenitores seleccionados para obtener hijos con una combinación genética diferente. Esto se puede realizar de múltiples formas, ya sea intercambiando conjuntos de parámetros (segmentos de genes) o escogiendo genes de forma aleatoria entre ambos progenitores, entre otras. Por otro lado, la operación de mutación busca mantener la diversidad genética en la población y la aparición de nuevas soluciones. Esto se logra mediante la inclusión de una probabilidad baja de que un hijo, después de haber sido construido por crossover, cambie de forma aleatoria alguno de sus parámetros antes de ser introducido en la siguiente generación.

Los mecanismos de selección de progenitores pueden tomar múltiples formas, siendo una de las más utilizadas la selección de rueda de ruleta, en la que la probabilidad de que un individuo sea seleccionado para reproducirse es proporcional a su función de fitness. También se puede utilizar la selección por rangos, en la cual la probabilidad de selección se relaciona con la posición relativa en comparación con el resto de individuos, en lugar de la función fitness. Igualmente, existe la alternativa de aplicar una selección por torneo, en la cual se escogen múltiples subconjuntos de la generación progenitora, a partir de los cuales se selecciona el individuo con mejor fitness, y se utiliza en conjunto con los ganadores de los otros subconjuntos (torneos) para construir la siguiente generación. Otras alternativas de selección son posibles, ya sea incorporando modificaciones o mecanismos completamente diferentes.

Una vez se genera completamente la segunda generación, se aplican los mismos procedimientos sobre esta para generar una tercera generación, y así sucesivamente. Este proceso de construcción de nuevas generaciones continúa hasta que el algoritmo cumpla alguna condición de término definida inicialmente, que puede tomar múltiples formas y su implementación depende de la naturaleza del problema. Algunos ejemplos son la identificación de un individuo con fitness por sobre un valor específico, un máximo número de generaciones alcanzado, o la presencia de un individuo que se mantiene como el fitness máximo por un número predeterminado de generaciones, entre otros. Una vez culmina el algoritmo genético, se tendrá como solución de la optimización el individuo con el mayor fitness.

Si bien se suelen usar los procedimientos ya descritos en la mayoría de los casos, es posible utilizar variaciones tales como reproducción con más de dos padres, mecanismos diferentes de reproducción, y otros operadores genéticos tales como reagrupación, clonación, colonización-extinción o migración, entre otros.

2.7. Optimización subrogada

En diversos tipos de proyectos de ingeniería se utilizan simulaciones para evaluar diseños y posteriormente optimizarlos, como por ejemplo al diseñar la geometría de una aeronave, la geometría de estructuras o sistemas térmicos. Sin embargo, cuando se desea evaluar sistemas de una complejidad mayor, los requerimientos de la simulación crecen rápidamente, llegando a generar conflicto con las limitaciones de software o de tiempo disponible.

Para estos casos, resulta útil la implementación de modelos subrogados, que permiten evaluar diseños y soluciones con un costo computacional significativamente menor. Los modelos subrogados buscan aproximar con alta precisión el comportamiento del sistema solo recibiendo como input su descripción, lo que se logra mediante un entrenamiento del modelo con una base de datos seleccionada estratégicamente. Estos modelos no requieren conocer los mecanismos internos del sistema, sino que se centran únicamente en la relación input-output.

Para construir estos modelos subrogados se pueden utilizar diversas herramientas de inteligencia artificial, tales como redes neuronales, regresión de vector soporte (SVR en inglés), estrategias de vecino más cercano o kriging, entre otros. Los modelos de redes neuronales predicen el comportamiento del sistema en base a los inputs relevantes del mismo, mientras que SVR usa un principio de funcionamiento similar al de las máquinas de vector soporte, incorporando un margen de tolerancia en la regresión (épsilon), y la posibilidad del mapeo de los datos a un espacio de kernel de más dimensiones. Por otro lado, las estrategias de vecinos más cercanos buscan aproximar el comportamiento del sistema comparándolo con el comportamiento de sistemas similares (o cercanos) suministrados al modelo, variando el número de modelos más similares considerados, y los coeficientes de pesos asignados. Por último, kriging busca predecir el valor de la función (o comportamiento del sistema) en un punto específico computando un promedio ponderado de los resultados obtenidos en otros puntos cercanos.

La optimización subrogada consiste en la aplicación de un método de optimización (P.E. Algoritmos genéticos) que, en lugar de utilizar una simulación costosa o un análisis complejo para evaluar el comportamiento y rendimiento de las soluciones, utiliza modelos subrogados que aproximen el sistema con una alta precisión [18]. Esto permite incrementar significativamente la eficiencia del proceso ya que, en la mayoría de los casos, los modelos subrogados requieren una cantidad de recursos computacionales mucho menor que una simulación precisa o un análisis determinístico del sistema.

Si bien el funcionamiento general de la optimización subrogada es simple, es fundamental que los modelos subrogados sean lo suficientemente precisos para que el proceso de optimización sea efectivo. En el caso de que los modelos no sean lo suficientemente precisos, resulta necesario refinarlos mediante la incorporación de nuevos puntos de muestreo al set de datos utilizado para su construcción. Este proceso de añadir nuevos elementos al set de datos para

refinar el modelo se llama “sample-point refinement”, y se pueden utilizar diferentes criterios de selección para asegurar que el elemento añadido aumente la precisión del modelo.

Un posible criterio de selección de elementos del set de datos es “searching surrogate models”, que consiste en aplicar optimización subrogada y evaluar la precisión del modelo sobre el valor obtenido; si la precisión es insuficiente, se incorpora el valor al set de datos y se repite el proceso, si la precisión es suficiente, entonces el valor obtenido es el óptimo. También se puede aplicar “expected improvement”, método que define una función que representa la mejora en precisión esperada al incorporar un dato al modelo, y posteriormente aplica algún método de optimización global para encontrar el valor que maximiza esta función, para posteriormente añadirlo al set de datos. De igual forma, se puede aplicar “lower-boundig confidence”, que incorpora al set de datos el valor resultante al optimizar una función (LBC) que combina la función original a optimizar con el error estándar de la predicción del modelo mediante una ponderación simple.

3. Metodología

El trabajo realizado para elaborar las metodologías de diseño de enrejados bidimensionales y tridimensionales con supresión de vibraciones consistió en tres etapas principales; la obtención de una red neuronal que logre predecir band-gaps en base a los parámetros de un enrejado bidimensional, su posterior implementación en un algoritmo de optimización subrogada que entregue enrejados bidimensionales con band-gaps en torno a la frecuencia de diseño, y la replicación de las etapas anteriores para el caso tridimensional. A continuación se detallará la metodología llevada a cabo, así como el software y hardware utilizados.

3.1. Hardware y software

El presente trabajo se llevó a cabo en su totalidad en una laptop Dell modelo Alienware m15 R4, con un procesador Intel(R) Core(TM) i7-10870H de 2.2 GHz, 32 GB de RAM, SSD de 1048 GB modelo PC SN730 NVMe WDC, tarjeta gráfica interna al procesador Intel(R) UHD Graphics, y tarjeta gráfica externa NVIDIA GeForce RTX 3080 Laptop GPU.

En cuanto a software, el trabajo se realizó en su totalidad en MATLAB 2018, en Windows 10. Se utilizaron los toolboxes de MATLAB “Deep Learning Toolbox” y “Global Optimization Toolbox”.

3.2. Enrejado bidimensional

Las dos primeras etapas del trabajo se aplicaron sobre un mismo tipo de enrejado bidimensional, definido completamente mediante 15 parámetros. A continuación se muestra la celda unitaria del enrejado bidimensional sobre el que se aplicará la metodología:

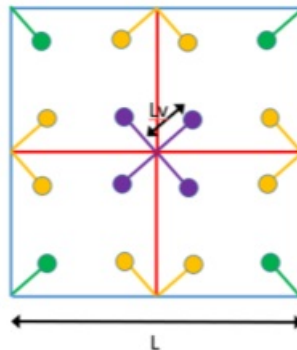


Figura 3.8: Celda unitaria de los enrejados sobre los que se aplicará la metodología.

Los parámetros a variar en la celda unitaria son:

- Longitud del lado de la celda L . Variable continua con valores entre 0.3 y 1.4, correspondiendo a un rango entre 30 [mm] y 140 [mm].
- Longitud de las varas internas L_v . Variable continua con valores entre 0.3 y 1.4, correspondiendo a un rango entre 3,5 [mm] y 16,5 [mm].
- Radio de cada uno de los 5 grupos de barras. Variables continuas con valores entre 0.3 y 1.4, correspondiendo a un rango entre 0,6 [mm] y 2,8 [mm].
- Material de cada uno de los 5 grupos de barras. Variable binaria correspondiente a aluminio o titanio.
- Presencia o ausencia de masas puntuales en cada uno de los 3 grupos de resonadores. Variable binaria, correspondiente a masa de 0,01 [kg] o 0 [kg].

A continuación, se muestran los materiales y propiedades consideradas para el enrejado bidimensional:

Tabla 1: Propiedades de los posibles materiales de las barras en el enrejado bidimensional.

Material	M. de Young [GPa]	Densidad [kg/m ³]	C. de Poisson []
Aluminio	68,9	2700	0,33
Titanio	411,0	19300	0,28

3.3. Enrejado tridimensional

La metodología para enrejados tridimensionales se aplicó sobre una estructura de pirámide central, cuya topología depende de 7 variables a optimizar. La geometría estructura también depende de un factor de altura f_l , valor que en el presente trabajo de mantiene constante e igual a 0,5, lo que se asocia con una separación entre placas definida por:

$$L_z = f_l \cdot \frac{L_c}{3} \tag{3.4}$$

Donde:

- L es el largo de la celda unitaria [mm].
- L_z es el largo de la celda unitaria [mm].
- f_l es el factor de altura [mm].

A continuación se muestra la celda unitaria del enrejado bidimensional sobre el que se aplicará la metodología:

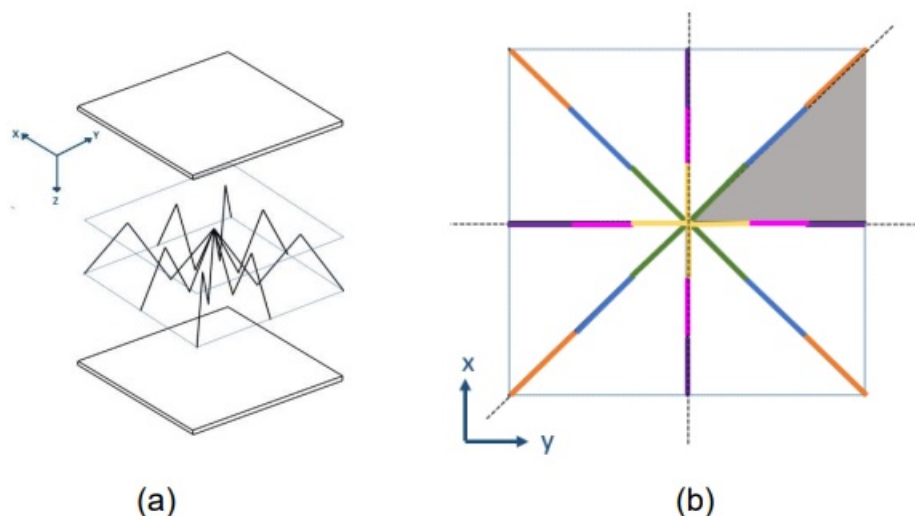


Figura 3.9: (a) Esquema de la estructura 3. (b) Vista superior de la celda y su reducción por simetrías.

Los 7 parámetros que definen la celda son variables continuas que varían entre 0 y 1, cada una de las cuales se utiliza para definir una de las siguientes dimensiones de la estructura:

- Largo de la celda unitaria L . Variable continua con valores entre 0,015 y 0,030, correspondiendo a un rango entre 15 [mm] y 30 [mm].
- 6 factores relacionados con los radios de los elementos definidos en la IBZ. Variables continuas entre 0,5 y 2,0, correspondiendo a un rango entre 0,4 [mm] y 1,6 [mm].

A continuación, se muestra el material y propiedades consideradas para el enrejado tridimensional:

Tabla 2: Propiedades del material utilizado para el enrejado tridimensional.

Material	M. de Young [GPa]	Densidad [kg/m ³]	C. de Poisson []
Acero	210,0	7800	0,30

3.4. Redes neuronales: Enrejado bidimensional

En la etapa de obtención de una red neuronal predictora de band-gaps para enrejados bidimensionales se consideraron dos posibles modelos. Para su entrenamiento se cuenta con un conjunto de 20000 enrejados, de los cuales 2264 presentan band-gap, y sus 8 primeras bandas de frecuencias obtenidas de forma analítica. Se toma el 60% de los datos como conjunto de entrenamiento, 30% como conjunto de validación, y 10% como conjunto de evaluación.

Ambas redes reciben en su entrada los 15 parámetros que definen la celda unitaria del enrejado, diferenciándose en que una predice directamente la frecuencia media y ancho del band-gap, y la otra los máximos y mínimos de las bandas necesarios para su obtención. Por lo tanto, mientras la primera red tiene solo 2 salidas, la segunda tiene 14 salidas; los 6 máximos y mínimos de las bandas centrales, el máximo de la primera banda, y el mínimo de la octava. En particular, para la red de predicción directa, los elementos sin band-gaps del conjunto de entrenamiento se asocian con un valor de ancho negativo correspondiente a la menor superposición entre bandas, y a una frecuencia correspondiendo al centro de esta superposición.

Para obtener la configuración óptima de hiperparámetros en cada caso, se utilizó un procedimiento de búsqueda aleatoria con distribución uniforme, incluyendo validación cruzada de 5 divisiones para disminuir el efecto probabilístico de la evaluación de las redes. Para cada conjunto de hiperparámetros generado, se evaluó la red tanto para la función de activación Relu como con LeakyRelu.

La evaluación de las redes al aplicar la búsqueda aleatoria consistió en la obtención del promedio de coeficiente de determinación (R^2) para el ancho y frecuencia predicha del band-gap. Al evaluar la mejor red obtenida, se obtiene también el coeficiente de determinación para el ancho y frecuencia evaluado exclusivamente sobre los enrejados con band-gap positivo (R^2+), y el tiempo de entrenamiento de la red.

La búsqueda aleatoria se aplicó en dos etapas, una búsqueda inicial con mayor libertad de variación en los parámetros, y posteriormente una búsqueda refinada. A continuación se muestran los parámetros variados y rangos utilizados en la búsqueda inicial:

Tabla 3: Rangos de variación para búsqueda aleatoria inicial (caso bidimensional).

Hyperparámetro	Rango de valores
Número de capas ocultas	2-8
Número de neuronas por capa	200-600
Factor de aprendizaje inicial	0.01-0.12
Tamaño de mini-lote	300-600
Factor de drop-out	0-0.15
Función de activación	Relu-LeakyRelu

Una vez realizada la búsqueda aleatoria inicial, se repitió el proceso restringiendo los rangos de variación de los hiperparámetros en función de los resultados de la primera búsqueda. Dado el comportamiento general similar en cuanto al rendimiento observado en ambas redes al variar los hiperparámetros, se utilizan los siguientes rangos para ambas redes en la búsqueda refinada:

Tabla 4: Rangos de variación para la búsqueda aleatoria refinada (caso bidimensional).

Hyperparámetro	Rango de valores
Número de capas ocultas	2-5
Número de neuronas por capa	200-600
Factor de aprendizaje inicial	0.01-0.12
Tamaño de mini-lote	300-600
Factor de drop-out	0-0.15
Función de activación	Relu

En cuanto a las opciones de entrenamiento de la red neuronal, para “solver” se utiliza “adam”, para “LearnRateSchedule” se utiliza “Piecewise”, y para “ExecutionEnvironment” se utiliza “cpu”. El resto de las opciones toman el valor estándar según lo especificado por el toolbox utilizado.

3.5. Red neuronal: Enrejado tridimensional

Para la obtención de una red neuronal predictora de band-gaps para enrejados tridimensionales se opta por implementar una red neuronal de predicción directa, dados los mejores resultados del caso bidimensional. Para su entrenamiento, nuevamente se cuenta con un conjunto de 20000 enrejados, de los cuales 401 presentan band-gap, y sus 8 primeras bandas de frecuencias obtenidas de forma analítica. Se toma el 60 % de los datos como conjunto de entrenamiento, 30 % como conjunto de validación, y 10 % como conjunto de evaluación.

Esta red recibe 7 parámetros de entrada, correspondientes a los parámetros que determinan completamente al enrejado de pirámide central, y tiene 2 salidas, correspondientes a la frecuencia del band-gap y a su ancho. Al igual que en el caso bidimensional, los elementos sin band-gaps del conjunto de entrenamiento se asocian con un valor de ancho negativo correspondiente a la menor superposición entre bandas, y a una frecuencia correspondiendo al centro de esta superposición.

Nuevamente, se utilizó un procedimiento de búsqueda aleatoria con distribución uniforme, incluyendo validación cruzada de 5 divisiones para disminuir el efecto probabilístico de la evaluación de las redes.

La evaluación de las redes al aplicar la búsqueda aleatoria consistió en la obtención del promedio de coeficiente de determinación (R^2) para el ancho y frecuencia predicha del band-gap. Al evaluar la mejor red obtenida, se obtiene también el coeficiente de determinación para el ancho y frecuencia evaluado exclusivamente sobre los enrejados con band-gap positivo (R^2+), y el tiempo de entrenamiento de la red.

A diferencia del caso bidimensional, se utiliza solo una etapa de búsqueda aleatoria, considerando los resultados de obtenidos en la etapa anterior al definir los rangos de los hiperparámetros. A continuación se muestran los parámetros variados y rangos utilizados:

Tabla 5: Rangos de variación para la búsqueda aleatoria inicial.

Hyperparámetro	Rango de valores
Número de capas ocultas	2-6
Número de neuronas por capa	200-500
Factor de aprendizaje inicial	0.01-0.12
Tamaño de mini-lote	200-500
Factor de drop-out	0-0.15
Función de activación	Relu

En cuanto a las opciones de entrenamiento de la red neuronal para el caso tridimensional, se utiliza “adam” para “solver”, para “LearnRateSchedule” se utiliza “Piecewise”, y para “ExecutionEnvironment” se utiliza “cpu”. El resto de las opciones toman el valor estándar según lo especificado por el toolbox utilizado.

3.6. Optimización subrogada

La optimización subrogada se aplica de forma recursiva, asegurando así la fidelidad de la solución final obtenida. Para esto, se comienza por el entrenamiento de la red neuronal utilizando el conjunto de datos iniciales de 20000 enrejados e incorporándola en un algoritmo de optimización global. Posteriormente, se evalúan los 5 mejores individuos obtenidos de dicha optimización y se comparan con el comportamiento físico real mediante un modelo analítico. En caso de que estos 5 mejores individuos no presenten un comportamiento lo suficientemente cercano al predicho por la red, se añaden al conjunto de datos de entrenamiento, se vuelve a entrenar la red neuronal, y se incorporan a la población inicial de la siguiente iteración del algoritmo genético/enjambre de partículas. Este proceso se aplica de forma recursiva, permitiendo que la red neuronal alcance un nivel de precisión que permita la obtención de soluciones satisfactorias, y agilizando el proceso de optimización global.

En cuanto a los hiperparámetros y estructura de la red neuronal a utilizar, se selecciona la mejor red obtenida en base a lo detallado anteriormente, presentándose dichos resultados en la siguiente sección.

Respecto al algoritmo de optimización global, para el caso bidimensional se consideró y evaluó la aplicación de algoritmos genéticos y de enjambre de partículas. Para el caso tridimensional, solo se implementó la metodología con algoritmo genético como método de optimización global, ya que, como se expone en la sección de resultados, esta alternativa presentó un mejor rendimiento en el caso bidimensional.

Tanto en el caso bidimensional como tridimensional, los individuos del algoritmo de optimización global se codifican incluyendo los parámetros ya detallados que describen la geometría del enrejado en cada caso, es decir, 15 en el caso bidimensional y 7 en el caso tridimensional. En todos los casos, se utilizó la siguiente como función objetivo a minimizar:

$$f(x) = \alpha \cdot |f_{diseño} - f_{predicha}| - (1 - \alpha) \cdot w_{predicho} \quad (3.5)$$

Donde:

- f : Función a minimizar.
- x : Vector de parámetros correspondientes a las características de la celda unitaria del enrejado.
- $f_{diseño}$: Frecuencia en torno a la cual se busca suprimir las vibraciones del enrejado.
- $f_{predicha}$: Frecuencia del band-gap predicho por la red neuronal.
- $w_{predicho}$: Ancho del band-gap predicho por la red neuronal.
- α : Ponderador entre 0 y 1 que permite otorgar prioridad a la precisión en la frecuencia o al ancho de banda.

La metodología se aplica de forma recursiva hasta que se alcanza el número máximo de iteraciones o bien, cuando se logre la convergencia. Para determinar la convergencia de las soluciones, se usa la siguiente condición:

$$|f_{diseño} - f_{analítica}| \leq \frac{w_{analítico}}{2} \quad (3.6)$$

Donde:

- $f_{analítica}$: Frecuencia del band-gap calculada por el modelo analítico.
- $w_{analítico}$: Ancho del band-gap calculado por el modelo analítico.

Esta condición de termino asegura que el enrejado resultante de la optimización contenga a la frecuencia de diseño dentro del band-gap obtenido.

A continuación, se muestra un esquema explicativo de la estructura general de la metodología de optimización subrogada aplicada:

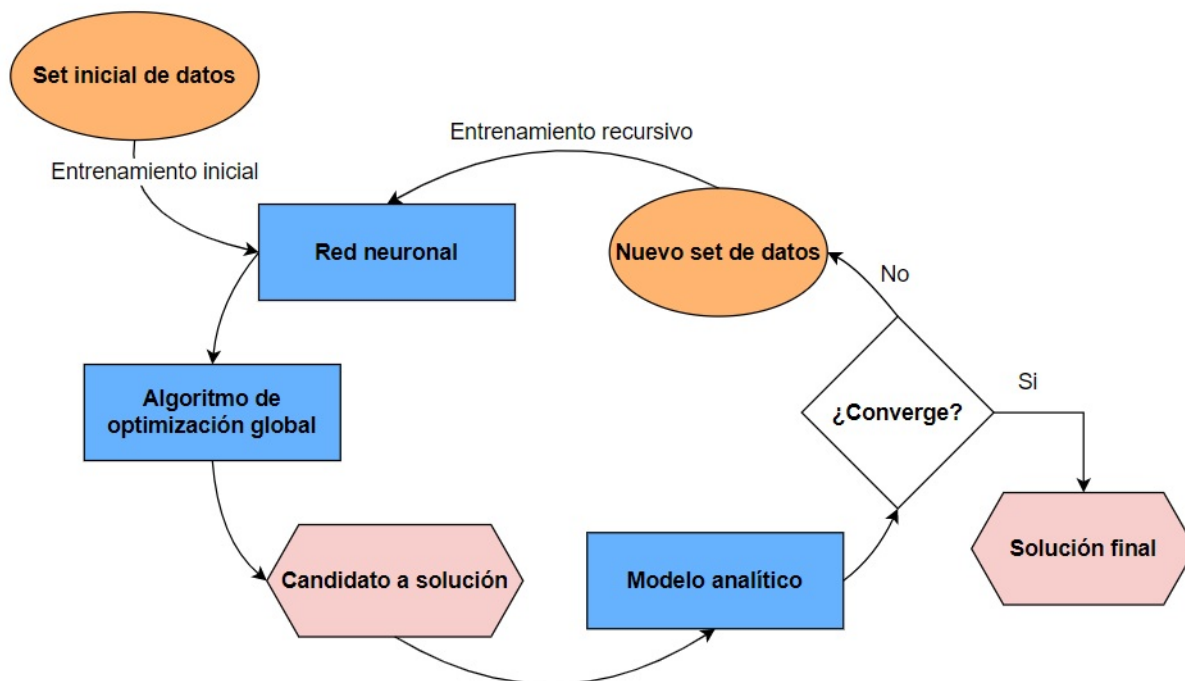


Figura 3.10: Funcionamiento general de la metodología de optimización subrogada aplicada.

En cuanto a las opciones utilizadas para el algoritmo genético en el caso bidimensional y tridimensional, se utilizan los valores estándar en la primera iteración de la metodología, mientras que para el resto de las iteraciones se utiliza un valor de 0 tanto para “FunctionTolerance” como “ConstraintTolerance”. En ambos casos se utiliza “InitialPopulationMatrix” para incorporar los 5 mejores elementos de la iteración anterior en la población inicial. El resto de parámetros toman los valores estándar según lo especificado por el toolbox de optimización global.

Para el algoritmo de enjambre de partículas, se utiliza 1 para “FunctionTolerance” en la iteración inicial y el valor estándar para el resto de iteraciones. De forma análoga a la metodología de algoritmo genético, se utiliza “InitialSwarmMatrix” para incorporar los 5 mejores elementos de la iteración anterior en la matriz inicial. El resto de parámetros toman los valores estándar según lo especificado por el toolbox de optimización global.

3.7. Evaluación de la metodología

Se evaluarán las metodologías bidimensional con algoritmos genéticos, bidimensional con enjambre de partículas, y tridimensional de igual forma y por separado. En cada caso, se evaluarán combinaciones de funciones objetivo y frecuencias de diseño, registrando tiempo de ejecución, el promedio del número de iteraciones hasta la convergencia, la frecuencia y ancho real del enrejado resultante, y el factor de centro, definido como:

$$f_c = 2 \cdot \frac{|f_{diseño} - f_{analítica}|}{w_{analítico}} \quad (3.7)$$

Donde:

- $f_{diseño}$: Frecuencia en torno a la cual se busca suprimir las vibraciones del enrejado.
- $f_{analítica}$: Frecuencia del band-gap calculada por el modelo analítico.
- $w_{analítico}$: Ancho del band-gap calculado por el modelo analítico.

El factor de centro definido en la ec. 3.7 resulta de utilidad para evaluar la precisión con la que la frecuencia de diseño se ubica dentro del band-gap; si la frecuencia de diseño se encuentra en el centro del band-gap, el factor de centro tomará un valor de 0, mientras que si la frecuencia de diseño se ubica en los extremos, el factor de centro será cercano a 1.

La selección de las frecuencias de diseño a evaluar se realiza basándose en la distribución de frecuencias de los band-gaps en el conjunto de entrenamiento de la red neuronal. A continuación se muestran los histogramas correspondientes a la ubicación de los band-gaps para el caso bidimensional y tridimensional:

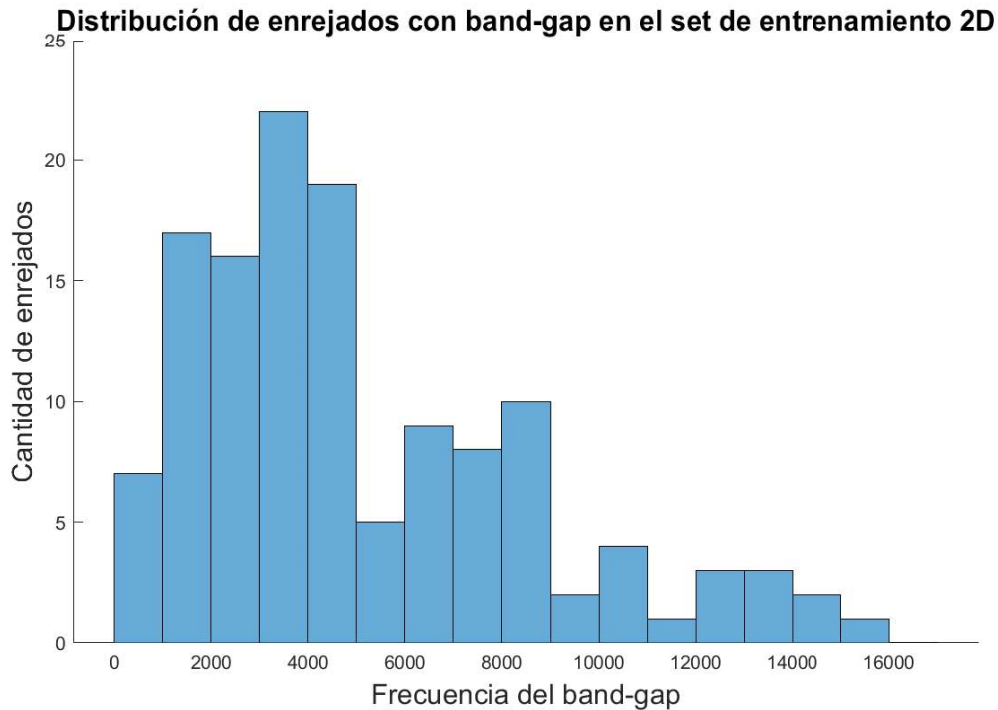


Figura 3.11: Histograma de frecuencias de los band-gaps en el conjunto de entrenamiento bidimensional para anchos de banda superiores o iguales a 50 [Hz].

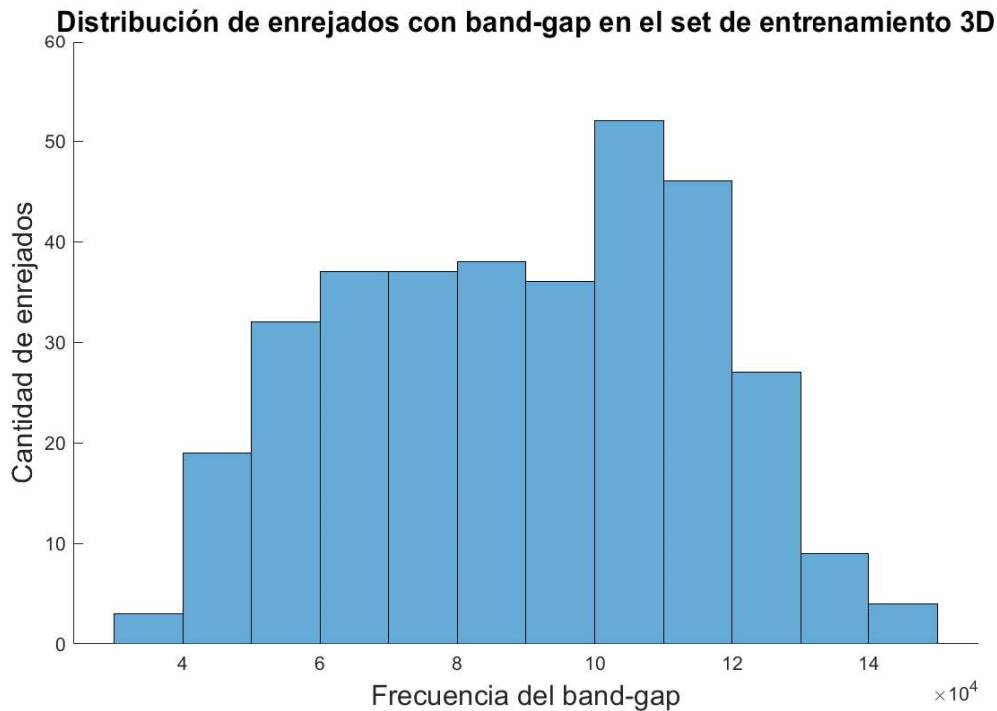


Figura 3.12: Histograma de frecuencias de los band-gaps en el conjunto de entrenamiento de pirámide central para anchos de banda superiores o iguales a 50 [Hz].

En el caso bidimensional, dado que el máximo observado en torno a 4000 [Hz], se opta por evaluar las frecuencias de diseño de 2000 [Hz], 4000 [Hz] y 6000 [Hz] para ambas metodologías. En el caso tridimensional, dado el máximo observado en 10^4 [Hz], se opta por evaluar las frecuencias $5 \cdot 10^5$ [Hz], $7 \cdot 10^4$ [Hz], 10^4 [Hz], $13 \cdot 10^4$ [Hz] y $15 \cdot 10^4$ [Hz]. Una vez evaluadas estas frecuencias de diseño con $\alpha = 0,5$, se escogerá aquella con mejor rendimiento para evaluar la metodología con valores de $\alpha = 0,05$ (alta prioridad al ancho del band-gap), $\alpha = 0,25$, $\alpha = 0,75$ y $\alpha = 0,95$ (alta prioridad a la precisión de la frecuencia) en la función objetivo.

Para la obtención de los promedios de las variables a evaluar, se consideran 5 evaluaciones de la metodología para cada combinación. Junto a estos promedios, en caso de haber convergencia satisfactoria, se graficará la evolución de la frecuencia predicha por la red a medida progresan las iteraciones, el diagrama de bandas del enrejado, la evolución del incumplimiento de la condición de término, y la evolución de la función objetivo.

Se establece un límite de 75 iteraciones para la convergencia de la metodología. En caso de no cumplirse la condición de salida (3.6) antes de este límite, se considerará la medición como convergencia no satisfactoria. Estos casos se registrarán pero no se tomarán en cuenta para el cálculo de valores promedios al evaluar, indicando esto con “N.A.” en la sección de resultados.

Para comparar el rendimiento de las metodologías de optimización subrogada con su equivalente analítico, se procederá aplicar la optimización con algoritmos genéticos y función objetivo analítica sobre las mismas frecuencias de diseño. Se calcularán los valores promedio de las frecuencias, anchos, factores de centro y tiempo de ejecución, considerando 3 ejecuciones de la metodología analítica para cada frecuencia.

De forma similar, se comparará la calidad de las soluciones entregadas por la metodología con los mejores enrejados del conjunto de entrenamiento para cada combinación frecuencia- α . Esto se realizará localizando los elementos con los mejores valores de cada función objetivo el conjunto de entrenamiento, y corroborando si cumplen o no la condición de detención.

Por último, se analizará la convergencia entre el modelo subrogado y el modelo de alta fidelidad, así como la convergencia de la condición de detención. Para esto, se ejecutará la metodología bidimensional con algoritmo genético y tridimensional por 500 iteraciones, y se elaborarán gráficos que representen la evolución del error entre ambos modelos y de la condición de detención.

Para representar la convergencia entre ambos modelos se define el siguiente parámetro:

$$e_n = \frac{\frac{|f_{analitica} - f_{predicha}|}{|f_{analitica}| + |f_{predicha}|} + \frac{|\omega_{analitico} - \omega_{predicho}|}{|\omega_{analitico}| + |\omega_{predicho}|}}{2} \quad (3.8)$$

Donde e_n representa el error normalizado entre el modelo subrogado y el modelo de alta fidelidad. Corresponde al promedio de los errores normalizados de la frecuencia y el ancho, que se definen de forma tal de tomar valores entre 0 y 1.

Por otro lado, para representar la evolución de la condición de convergencia, se reconoce que en el caso límite se tiene:

$$2 \cdot \left| \frac{f_{diseño} - f_{analitica}}{w_{analitico}} \right| = 1 \quad (3.9)$$

Por lo anterior, se escoge la cantidad del lado izquierdo de la ecuación 3.9 para evaluar la convergencia de la condición de detención, utilizando 1 como el valor de referencia.

4. Resultados

4.1. Redes neuronales: Enrejado bidimensional

A continuación, se muestran los hiperparámetros correspondientes a las mejores redes neuronales obtenidas mediante el proceso de búsqueda aleatoria para el caso de enrejado bidimensional:

Tabla 4.1: Mejor red directa.

Hyperparámetro	Valores
Número de capas ocultas	3
Número de neuronas por capa	267
Factor de aprendizaje inicial	0.0381
Tamaño de mini-lote	421
Factor de drop-out	0.1415
Frecuencia de drop-out	50
Épocas máximas	1000
Función de activación	Relu

Tabla 4.2: Mejor red de máximos y mínimos.

Hyperparámetro	Valores
Número de capas ocultas	3
Número de neuronas por capa	543
Factor de aprendizaje inicial	0.0108
Tamaño de mini-lote	355
Factor de drop-out	0.1302
Frecuencia de drop-out	50
Épocas máximas	1000
Función de activación	Relu

Como ya se detalló anteriormente, la evaluación de ambas redes neuronales consistió en la obtención del promedio de coeficiente de determinación (R^2) para el ancho y frecuencia predicha del band-gap, el coeficiente de determinación para el ancho y frecuencia evaluado exclusivamente sobre los enrejados con band-gap positivo (R^2+), y el tiempo de entrenamiento de la red. A continuación, se exponen los resultados obtenidos de esta evaluación para ambas redes:

Tabla 4.3: Rendimiento mejores redes.

Red Neuronal	Directa	Max. y min.
R^2 Frecuencia	0.83370	0.87367
R^2 Ancho	0.59058	0.62794
$R^2 +$ Frecuencia	0.86668	0.89443
$R^2 +$ Ancho	-0.90575	-1.38217
Tiempo [s]	432	1576

Se aprecia como la red de máximos y mínimos presenta coeficientes de determinación mayores a los de la red de predicción directa, con la excepción del caso de ancho de banda con band-gaps positivos. En este último caso ambas redes presentan valores negativos, lo que se traduce en una mala capacidad de predicción del ancho de banda cuando su valor es positivo. En cuanto al tiempo de entrenamiento, se observa que la red de predicción directa se logra entrenar en tiempos significativamente menores, lo que tiene una relación directa con el tiempo de cómputo que requiere la metodología completa. A partir de estos resultados, se decide utilizar la red de predicción directa en la metodología completa por sobre la red de máximos y mínimos.

A continuación, se muestran ejemplos de diagramas de dispersión y matriz de confusión obtenidas con las mejores redes neuronales para cada caso. En la matriz de confusión, la clase “0” corresponde a enrejados sin band-gap (asociados a un band-gap negativo), mientras que la clase “1” corresponde a los enrejados con band-gap. Esto permite visualizar la capacidad de predicción de la red analizando ambos casos por separado:

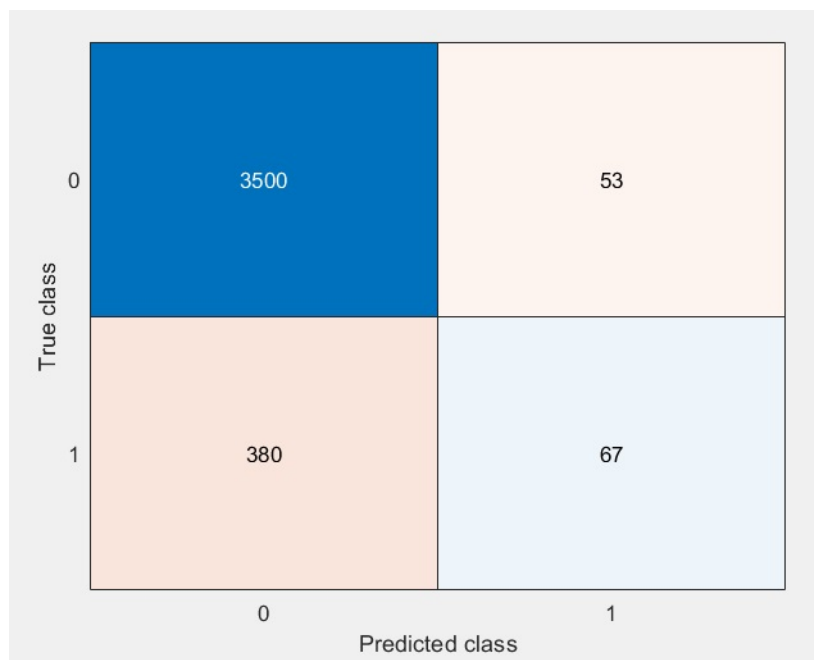


Figura 4.1: Matriz de confusión de la red neuronal de predicción directa sobre enrejados bidimensionales.

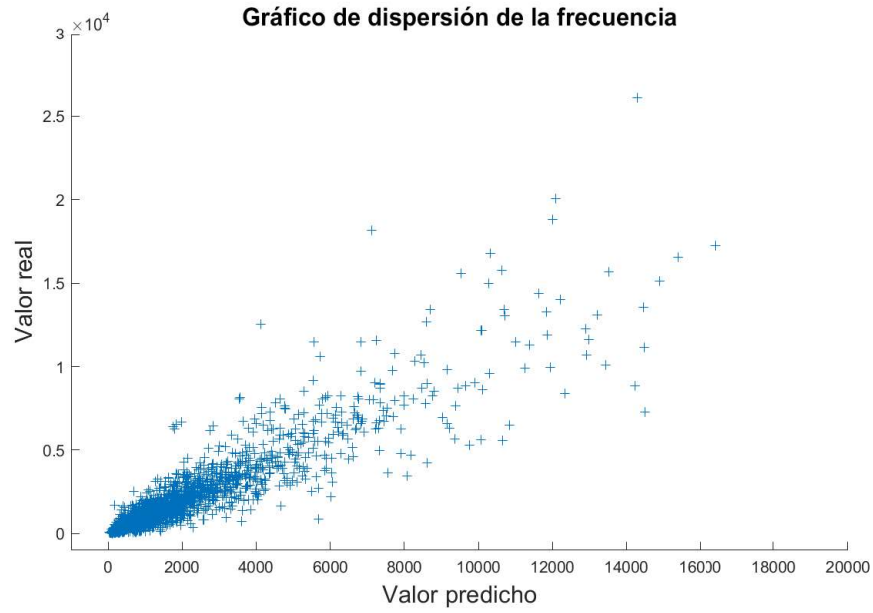


Figura 4.2: Diagrama de dispersión de las frecuencias del band-gap con la red neuronal de predicción directa sobre enrejados bidimensionales.

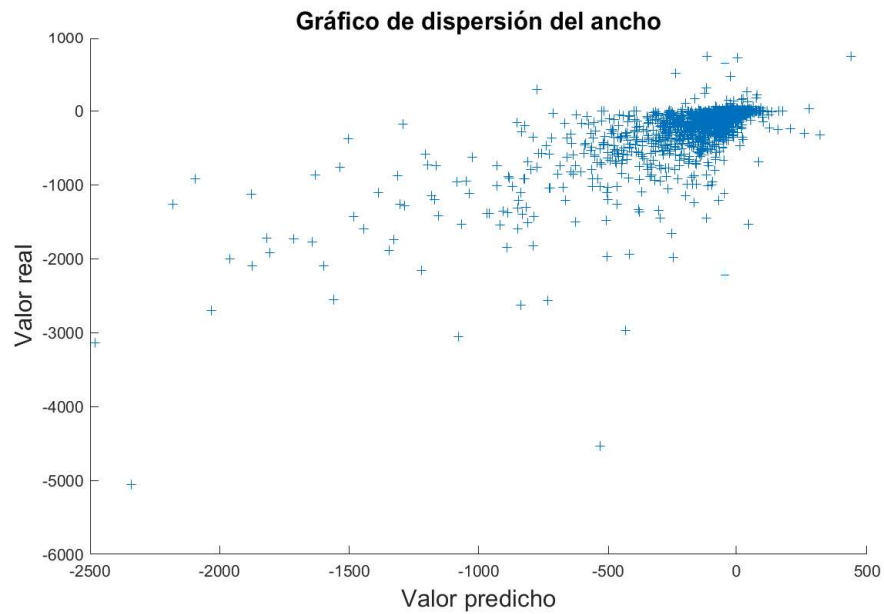


Figura 4.3: Diagrama de dispersión del anchos de banda con la red neuronal de predicción directa sobre enrejados bidimensionales.

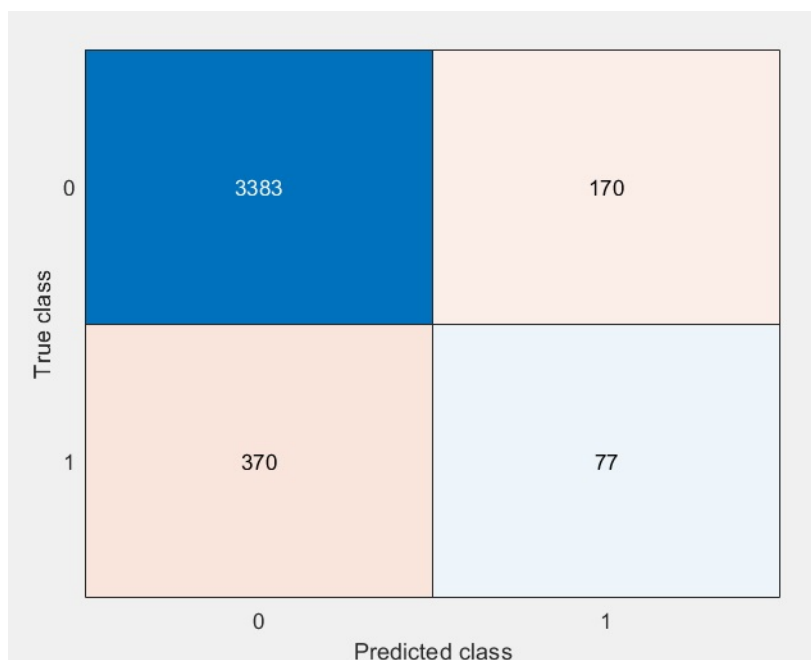


Figura 4.4: Matriz de confusión de la red neuronal de máximos y mínimos sobre enrejados bidimensionales.

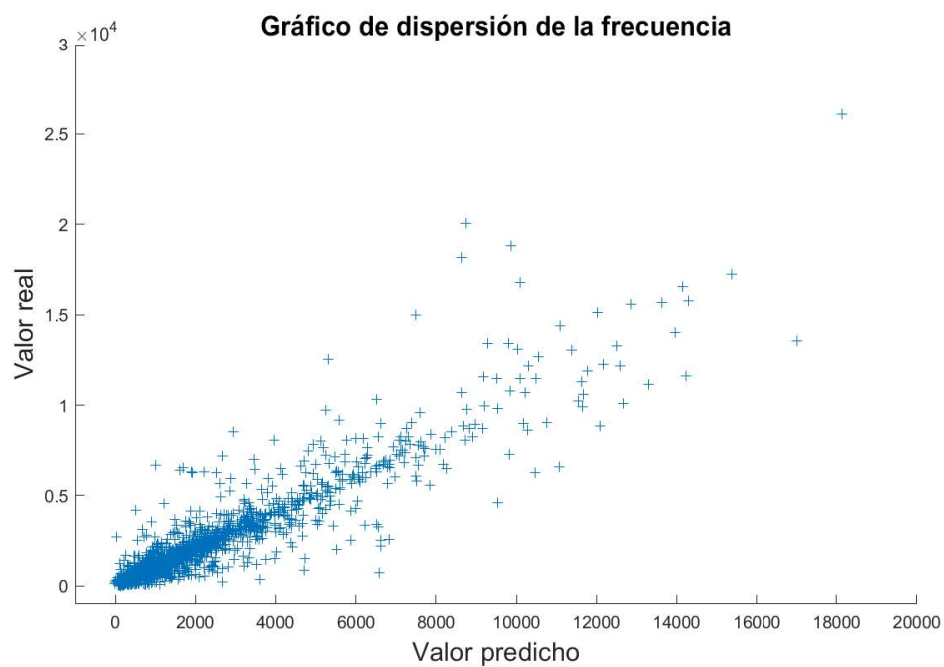


Figura 4.5: Diagrama de dispersión de las frecuencias del band-gap con la red neuronal de máximos y mínimos sobre enrejados bidimensionales.

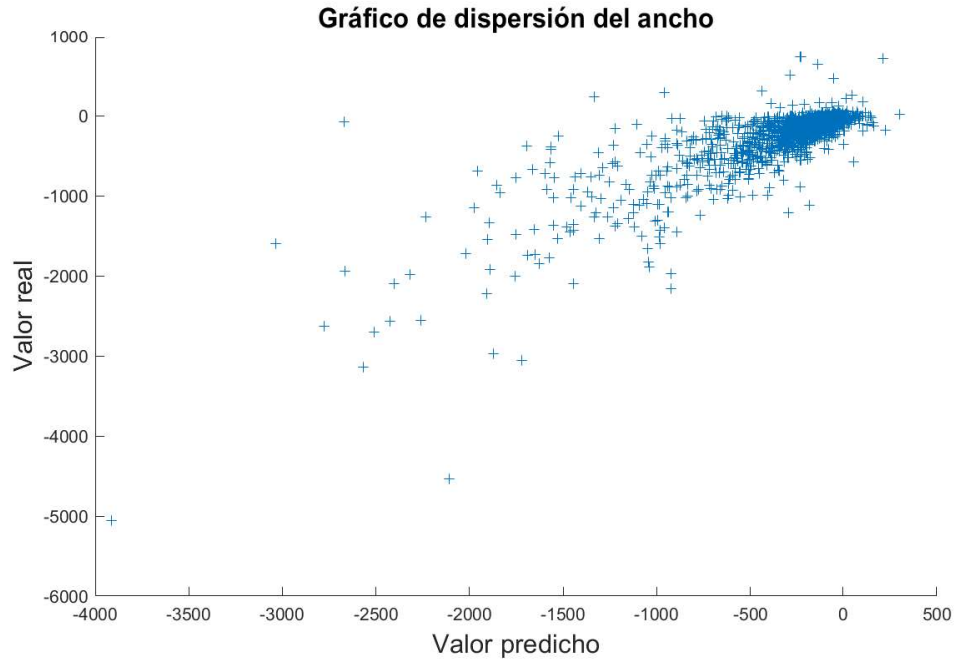


Figura 4.6: Diagrama de dispersión del anchos de banda con la red neuronal de máximos y mínimos sobre enrejados bidimensionales.

4.2. Redes neuronales: Enrejado tridimensional

A continuación, se muestran los hiperparámetros correspondientes a la mejor red neuronal obtenida mediante el proceso de búsqueda aleatoria para el caso de enrejado tridimensional:

Tabla 4.4: Mejor red directa.

Hyperparámetro	Rango de valores
Número de capas ocultas	3
Número de neuronas por capa	488
Factor de aprendizaje inicial	0.0991
Tamaño de mini-lote	476
Factor de drop-out	0.1419
Frecuencia de drop-out	50
Épocas máximas	1000
Función de activación	Relu

Al igual que en el caso bidimensional, la evaluación de la red neuronal consistió en la obtención del promedio de coeficiente de determinación (R^2) para el ancho y frecuencia predicha del band-gap, el coeficiente de determinación para el ancho y frecuencia evaluado exclusivamente sobre los enrejados con band-gap positivo (R^2+), y el tiempo de entrenamiento de la red.

A continuación, se exponen los resultados obtenidos de esta evaluación para el caso tridimensional:

Tabla 4.5: Rendimiento de la red neuronal aplicada sobre enrejados de pirámide central.

Red Neuronal 3D	
R^2 Frecuencia	0,89035
R^2 Ancho	0,85471
$R^2 +$ Frecuencia	0,99114
$R^2 +$ Ancho	-0,7301
Tiempo [s]	1075

A continuación, se muestran ejemplos de diagramas de dispersión y matriz de confusión obtenidas con la red neuronal seleccionada, donde las clases de la matriz de confusión nuevamente representan los enrejados con (clase 1) y sin (clase 0) band-gap:

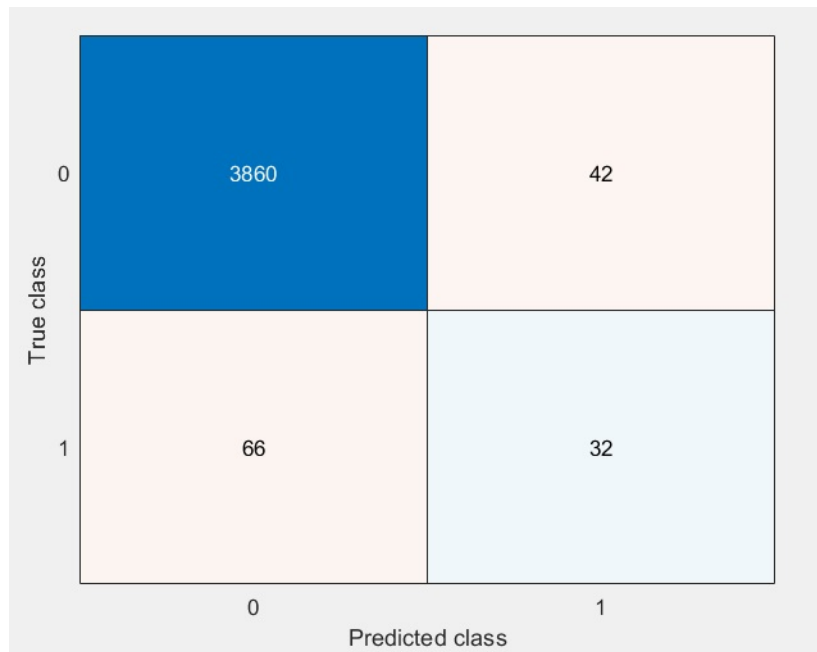


Figura 4.7: Matriz de confusión de la red neuronal aplicada sobre enrejados de pirámide central.

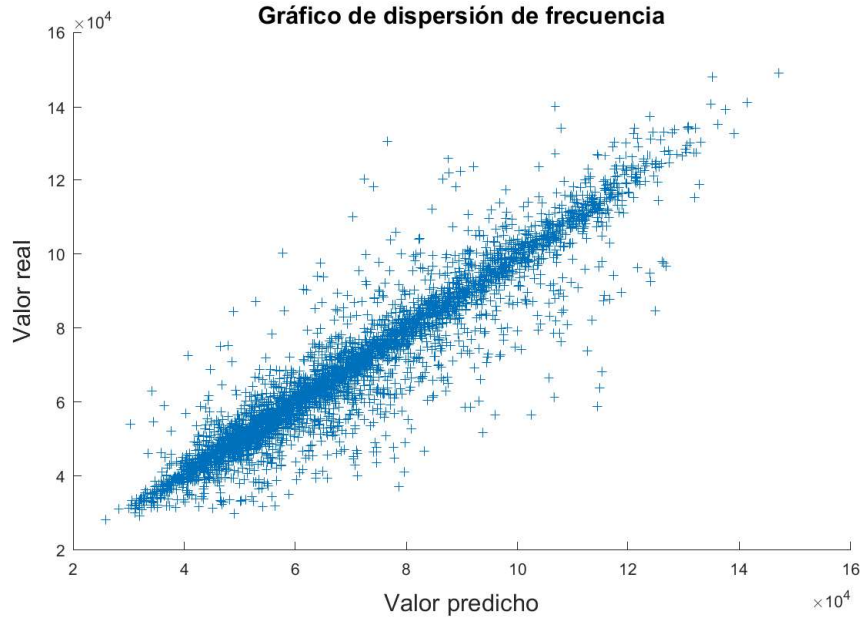


Figura 4.8: Diagrama de dispersión de las frecuencias del band gap obtenido con la red neuronal aplicada sobre enrejados de pirámide central.

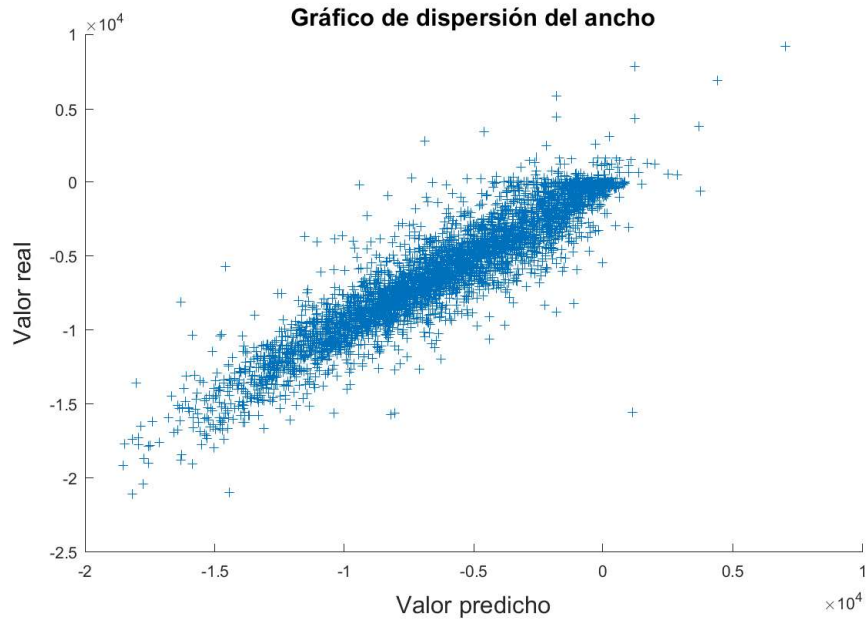


Figura 4.9: Diagrama de dispersión del ancho de banda obtenido con la red neuronal aplicada sobre enrejados de pirámide central.

4.3. Resultados metodología bidimensional: Algoritmo genético

Como se detalló anteriormente, para evaluar la metodología se procede a medir el promedio del número de iteraciones hasta la convergencia, la frecuencia y ancho real del enrejado resultante, y el factor de centro (ec. 3.7). Esta evaluación se realizó variando en primer lugar la frecuencia de diseño, utilizando $\alpha = 0,5$ (igual peso para frecuencia y ancho de banda) en la función objetivo de la optimización (ec. 3.5). Nótese el valor “N.A.”, que toma la frecuencia, ancho y factor de centro al no haber convergencia satisfactoria. A continuación, se muestran los resultados obtenidos para frecuencias de diseño de 2000 [Hz], 4000 [Hz], y 6000 [Hz] con algoritmo genético como método de optimización global:

Tabla 4.6: Resultados de la metodología con algoritmo genético al evaluar 2000 [Hz] y $\alpha = 0,5$.

Muestra	Iteraciones	Δ Frecuencia [Hz]	Ancho [Hz]	Factor de centro
1	53	3,5	89,4485	0,0788
2	61	139,6	349,0988	0,8
3	75+	N.A.	N.A.	N.A.
4	75+	N.A.	N.A.	N.A.
5	75+	N.A.	N.A.	N.A.
Prom. ≤ 75	57	71,6	219,2737	0,4394

Tabla 4.7: Resultados de la metodología con algoritmo genético al evaluar 4000 [Hz] y $\alpha = 0,5$.

Muestra	Iteraciones	Δ Frecuencia [Hz]	Ancho [Hz]	Factor de centro
1	7	5,9	70,5032	0,1679
2	65	41,1	147,7367	0,5563
3	75+	N.A.	N.A.	N.A.
4	49	3,4	18,0794	0,3722
5	75+	N.A.	N.A.	N.A.
Prom. ≤ 75	40	14,5	78,7731	0,3655

Tabla 4.8: Resultados de la metodología con algoritmo genético al evaluar 6000 [Hz] y $\alpha = 0,5$.

Muestra	Iteraciones	Δ Frecuencia [Hz]	Ancho [Hz]	Factor de centro
1	39	17,9	61,0842	0,5877
2	8	77,3	162,6391	0,9503
3	28	34,7	406,3134	0,1709
4	33	25,8	76,9895	0,6701
5	4	78,7	269,0420	0,5849
Prom. ≤ 75	22	46,9	195,2136	0,5927

Se observa como la frecuencia de diseño con los mejores resultados es 6000 [Hz], registrándose convergencia en todas las mediciones realizadas en menos de 75 iteraciones. Por esto, al evaluar el rendimiento de la metodología con diferentes valores de α , se utilizará una frecuencia de diseño de 6000 [Hz].

A continuación, se muestran los resultados obtenidos para $\alpha = 0,05$ (alta prioridad al ancho de banda), $\alpha = 0,25$, $\alpha = 0,75$ y $\alpha = 0,95$ (alta prioridad a la frecuencia):

Tabla 4.9: Resultados de la metodología con algoritmo genético al evaluar 6000 [Hz] y $\alpha = 0,05$.

Muestra	Iteraciones	Δ Frecuencia [Hz]	Ancho [Hz]	Factor de centro
1	75+	N.A.	N.A.	N.A.
2	75+	N.A.	N.A.	N.A.
3	75+	N.A.	N.A.	N.A.
4	75+	N.A.	N.A.	N.A.
5	75+	N.A.	N.A.	N.A.
Prom. ≤ 75	N.A.	N.A.	N.A.	N.A.

Tabla 4.10: Resultados de la metodología con algoritmo genético al evaluar 6000 [Hz] y $\alpha = 0,25$.

Muestra	Iteraciones	Δ Frecuencia [Hz]	Ancho [Hz]	Factor de centro
1	23	45,4	349,6774	0,2597
2	32	33,8	115,329	0,5868
3	9	140,7	608,2834	0,4626
4	61	113,9	271,1712	8403
5	44	57,7	322,3012	0,3578
Prom. ≤ 75	34	78,3	333,3524	0,50144

Tabla 4.11: Resultados de la metodología con algoritmo genético al evaluar 6000 [Hz] y $\alpha = 0,75$.

Muestra	Iteraciones	Δ Frecuencia [Hz]	Ancho [Hz]	Factor de centro
1	69	127,0	333,0796	0,7625
2	11	89,1	261,4073	0,6815
3	49	23,6	84,6722	0,5584
4	75+	N.A.	N.A.	N.A.
5	75+	N.A.	N.A.	N.A.
Prom. ≤ 75	43	79,9	226,3864	0,6675

Tabla 4.12: Resultados de la metodología con algoritmo genético al evaluar 6000 [Hz] y $\alpha = 0,95$.

Muestra	Iteraciones	Δ Frecuencia [hz]	Ancho [hz]	Factor de centro
1	32	46,5	644,1152	0,1444
2	75+	N.A.	N.A.	N.A.
3	75+	N.A.	N.A.	N.A.
4	20	16,9	68,431	0,4944
5	75+	N.A.	N.A.	N.A.
Prom. ≤ 75	26	31,7	356,2731	0,3194

Si bien el caso con $\alpha = 0,5$ logra convergencia en menos iteraciones, se observa que en promedio de los anchos de banda alcanzados es superior para $\alpha = 0,25$, $\alpha = 0,75$ y $\alpha = 0,95$. Para $\alpha = 0,05$ no se observa convergencia satisfactoria en ninguna de las muestras. Se observa también una mejora en el factor de centro en las muestras con $\alpha = 0,25$ y $\alpha = 0,95$ con respecto al caso base $\alpha = 0,5$, aunque en el último caso los resultados no son completamente comparables ya que solo 2 de las 5 muestras convergen satisfactoriamente.

A continuación, se muestran los tiempos promedios calculados en alcanzar la convergencia / máximo de iteraciones para las diferentes combinaciones evaluadas, con un tiempo promedio por iteración de 426 [s]:

Tabla 4.13: Tiempos promedios hasta convergencia o máximo de iteraciones (algoritmo genético).

Frecuencia [Hz]	α	t. [s]	t. [hr]
2000	0.5	28898	8,03
4000	0.5	23102	6,42
6000	0.05	31044	8,62
6000	0.25	14407	4,00
6000	0.5	9548	2,65
6000	0.75	23784	6,61
6000	0.95	25379	7,05

Se presenta también la información relativa a los tiempos mínimos, máximos y promedios observados para la ejecución del algoritmo genético en una iteración:

Tabla 4.14: Tiempos máximos, mínimos y promedio por iteración observados en la ejecución del algoritmo genético.

Frecuencia [Hz]	α	t. Min. [s]	t. Max. [s]	t. Promedio [s]
2000	0.5	33	178	109
4000	0.5	22	171	110
6000	0.05	12	178	43
6000	0.25	12	170	84
6000	0.5	11	171	109
6000	0.75	15	170	99
6000	0.95	14	184	96

A continuación, se exponen ejemplos de gráficos obtenidos al aplicar la metodología bi-dimensional con algoritmo genético. En particular, se muestran diagramas de banda con band-gap en torno a la frecuencia de diseño deseada, y diagramas correspondientes a la evolución de la frecuencia del enrejado con mejor función objetivo en cada iteración:

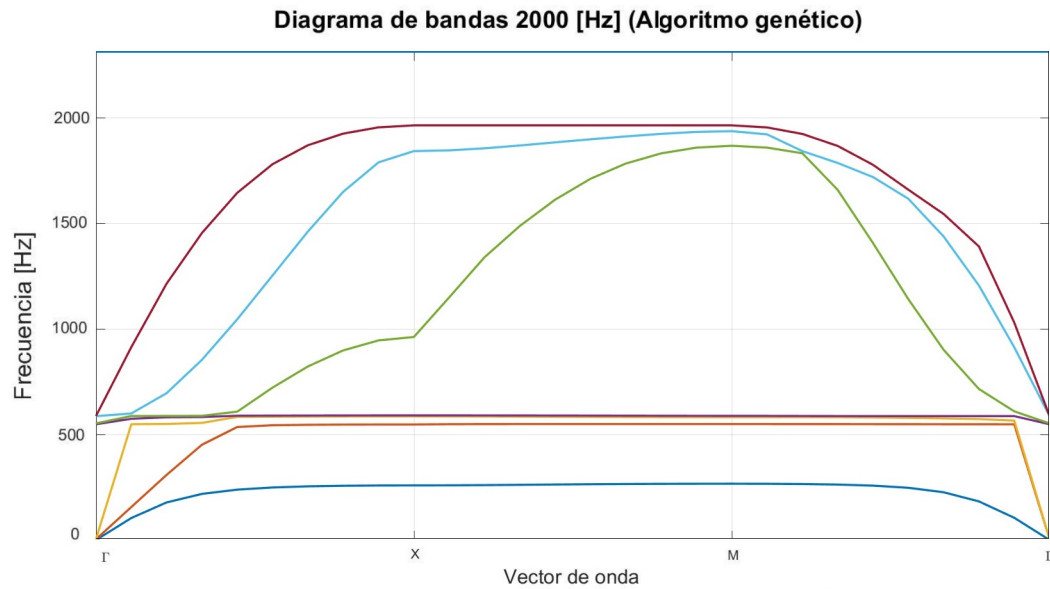


Figura 4.10: Ejemplo de diagrama de bandas con band-gap obtenido para 2000 [Hz] y $\alpha = 0,5$.

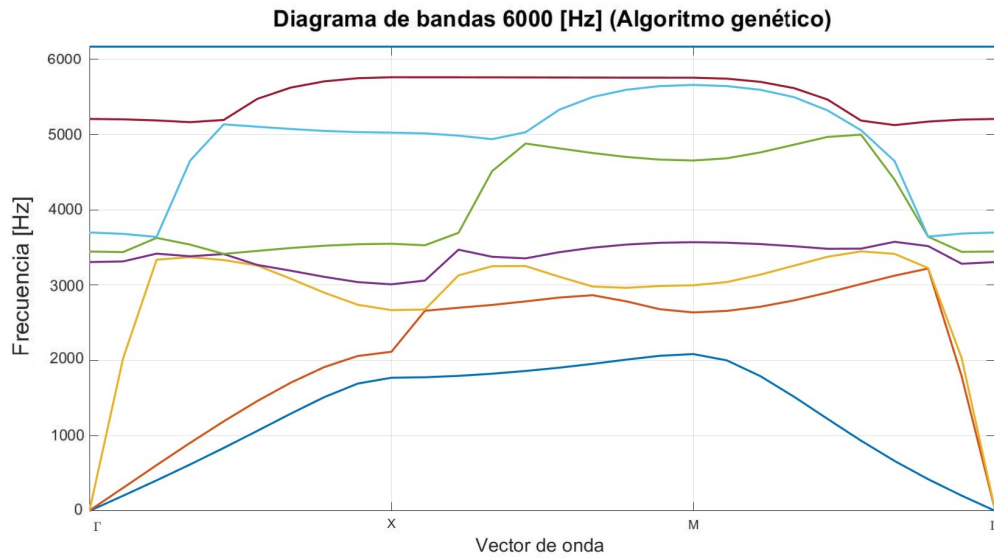


Figura 4.11: Ejemplo de diagrama de bandas con band-gap obtenido para 6000 [Hz] y $\alpha = 0,25$.

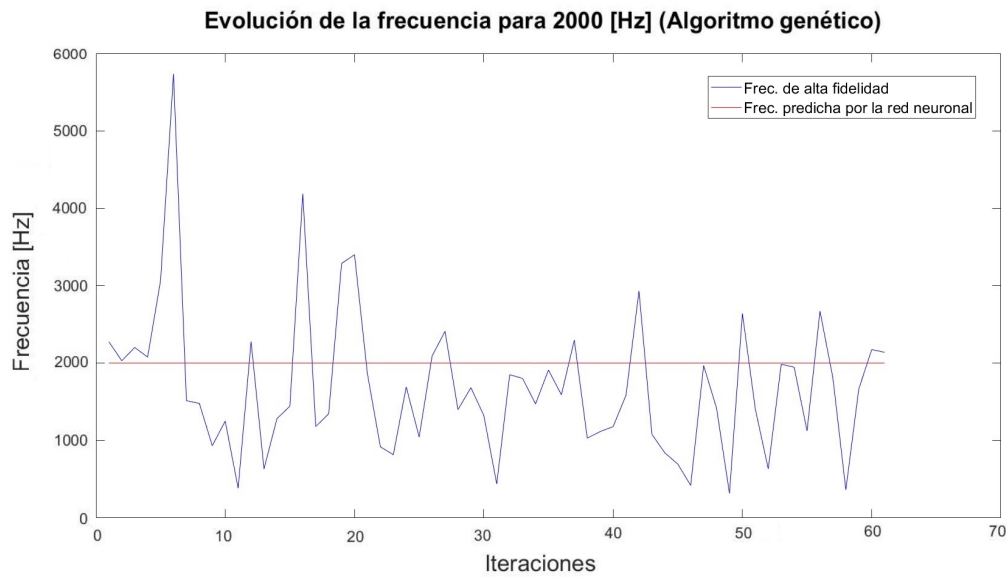


Figura 4.12: Ejemplo de evolución en iteraciones de la frecuencia de alta fidelidad del óptimo (azul) y de la predicha por la red (rojo) para 2000 [Hz] y $\alpha = 0,5$.

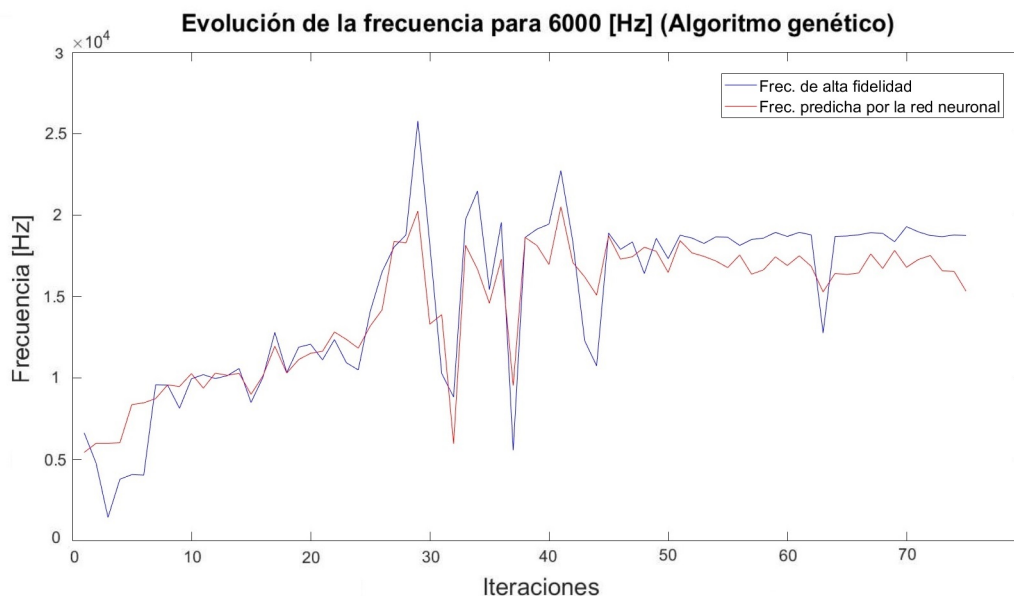


Figura 4.13: Ejemplo de evolución en iteraciones de la frecuencia de alta fidelidad del óptimo (azul) y de la predicha por la red (rojo) para 6000 [Hz] y $\alpha = 0,05$.

A continuación, se muestran ejemplos de diagrama describiendo la evolución de la magnitud por la que se incumple la condición de término 3.6 en el transcurso de la metodología, donde la línea roja corresponde al cumplimiento mínimo de dicha condición:

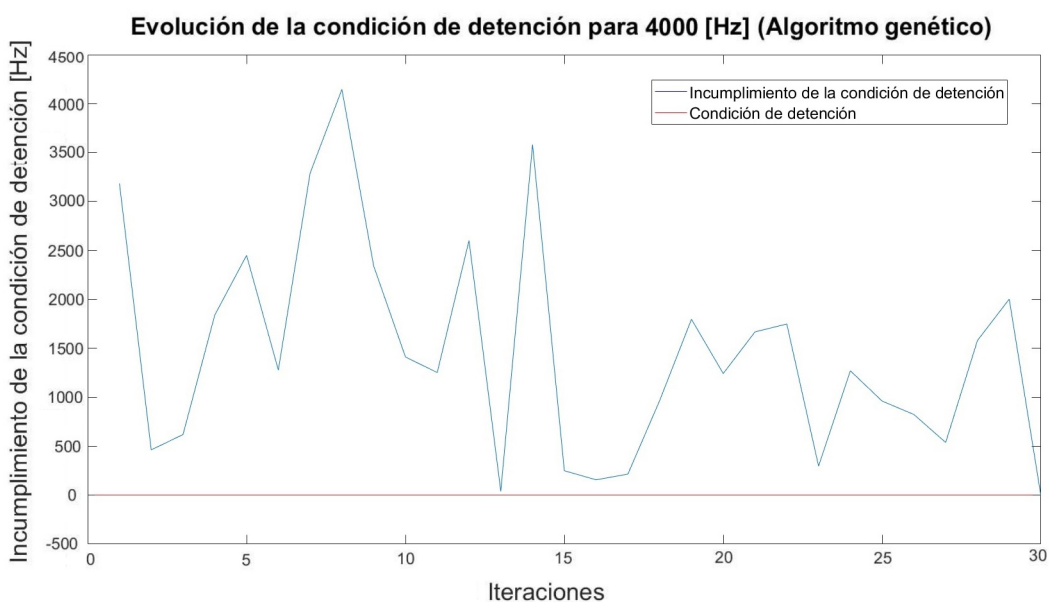


Figura 4.14: Ejemplo de evolución del incumplimiento de la condición de término (azul) para 4000 [Hz] y $\alpha = 0,5$ (Algoritmo genético).

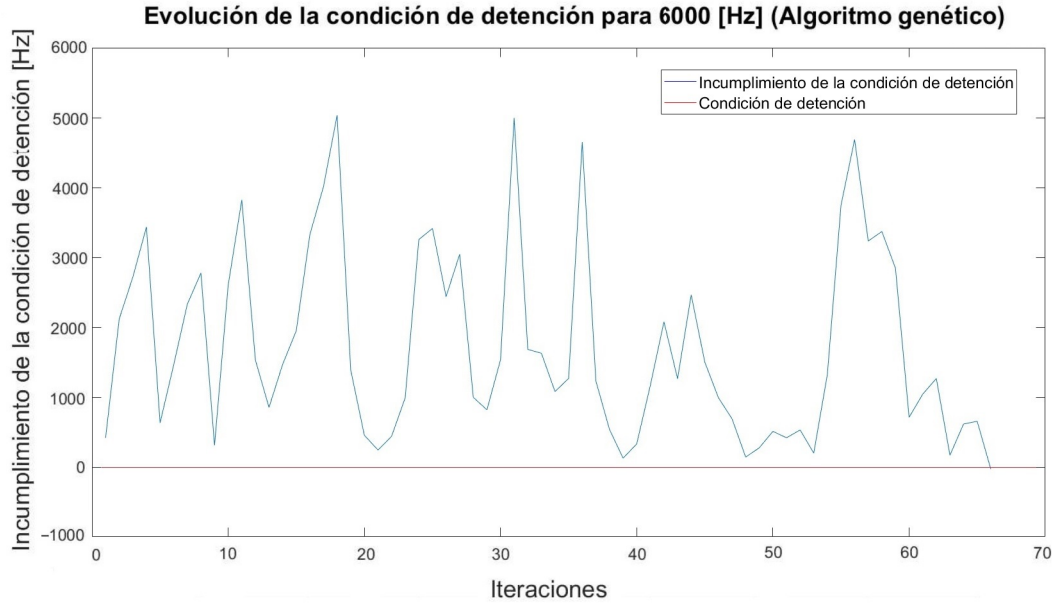


Figura 4.15: Ejemplo de evolución del incumplimiento de la condición de término (azul) para 6000 [Hz] y $\alpha = 0,05$ (Algoritmo genético).

4.4. Resultados metodología bidimensional: Enj. de partículas

De igual forma al caso anterior, para evaluar la metodología con enjambre de partículas se procedió a medir el promedio del número de iteraciones hasta la convergencia, la frecuencia y ancho real del enrejado resultante, y el factor de centro (ec. 3.7). Idénticamente, se varió en primer lugar la frecuencia de diseño, utilizando $\alpha = 0,5$ (igual peso para frecuencia y ancho de banda) en la función objetivo de la optimización (ec. 3.5). Nótese el valor “N.A.”, que toma la frecuencia, ancho y factor de centro al no haber convergencia satisfactoria. A continuación, se muestran los resultados obtenidos para frecuencias de diseño de 2000 [Hz], 4000 [Hz], y 6000 [Hz] con enjambre de partículas como método de optimización global:

Tabla 4.15: Resultados de la metodología con enjambre de partículas al evaluar 2000 [Hz] y $\alpha = 0,5$.

Muestra	Iteraciones	Δ Frecuencia [Hz]	Ancho [Hz]	Factor de centro
1	75+	N.A.	N.A.	N.A.
2	75+	N.A.	N.A.	N.A.
3	75+	N.A.	N.A.	N.A.
4	75+	N.A.	N.A.	N.A.
5	75+	N.A.	N.A.	N.A.
Prom. ≤ 75	N.A.	N.A.	N.A.	N.A.

Tabla 4.16: Resultados de la metodología con enjambre de partículas al evaluar 4000 [Hz] y $\alpha = 0,5$.

Muestra	Iteraciones	Δ Frecuencia [Hz]	Ancho [Hz]	Factor de centro
1	75+	N.A.	N.A.	N.A.
2	75+	N.A.	N.A.	N.A.
3	75+	N.A.	N.A.	N.A.
4	75+	N.A.	N.A.	N.A.
5	54	111,4	547,3578	0,4069
Prom. ≤ 75	54	111,4	547,3578	0,4069

Tabla 4.17: Resultados de la metodología con enjambre de partículas al evaluar 6000 [Hz] y $\alpha = 0,5$.

Muestra	Iteraciones	Δ Frecuencia [Hz]	Ancho [Hz]	Factor de centro
1	35	9,5	19,4954	0,9788
2	75+	N.A.	N.A.	N.A.
3	3	102,7	289,635	0,7095
4	33	69,5	142,5437	0,9758
5	43	73,8	290,3441	0,5083
Prom. ≤ 75	29	63,9	185,5046	0,7931

Se observa como la frecuencia de diseño con los mejores resultados es, al igual que en el caso anterior, 6000 [Hz]. Por esto, al evaluar el rendimiento de la metodología con diferentes valores de α , se utilizará una frecuencia de diseño de 6000 [Hz].

A continuación, se muestran los resultados obtenidos para $\alpha = 0,05$ (alta prioridad al ancho de banda), $\alpha = 0,25$, $\alpha = 0,75$ y $\alpha = 0,95$ (alta prioridad a la frecuencia):

Tabla 4.18: Resultados de la metodología con enjambre de partículas al evaluar 6000 [Hz] y $\alpha = 0,05$.

Muestra	Iteraciones	Δ Frecuencia [hz]	Ancho [hz]	Factor de centro
1	75+	N.A.	N.A.	N.A.
2	23	173,8	427,8058	0,8126
3	75+	N.A.	N.A.	N.A.
4	75+	N.A.	N.A.	N.A.
5	31	370,1	906,9940	0,8161
Prom. ≤ 75	27	272,0	667,3999	0,8144

Tabla 4.19: Resultados de la metodología con enjambre de partículas al evaluar 6000 [Hz] y $\alpha = 0,25$.

Muestra	Iteraciones	Δ Frecuencia [Hz]	Ancho [Hz]	Factor de centro
1	61	22,6	605,3246	0,0748
2	5	288,7	616,2035	0,9371
3	8	30,9	134,8863	0,4583
4	3	0,6	5,2064	0,2178
5	25	83,4	204,8087	0,8145
Prom. ≤ 75	20	85,2	313,2859	0,5005

Tabla 4.20: Resultados de la metodología con enjambre de partículas al evaluar 6000 [Hz] y $\alpha = 0,75$.

Muestra	Iteraciones	Δ Frecuencia [Hz]	Ancho [Hz]	Factor de centro
1	26	17,4	217,4679	0,1604
2	1	42,7	163,685	0,5218
3	20	135,1	325,694	0,8296
4	28	10,7	134,9248	0,1593
5	54	5,6	266,4373	0,0422
Prom. ≤ 75	26	42,3	221,6418	0,34266

Tabla 4.21: Resultados de la metodología con enjambre de partículas al evaluar 6000 [Hz] y $\alpha = 0,95$.

Muestra	Iteraciones	Δ Frecuencia [hz]	Ancho [hz]	Factor de centro
1	41	36,4	92,6416	0,7859
2	4	48,4	107,2366	0,9018
3	39	141,7	353,5324	0,8018
4	72	177,0	408,7254	0,8663
5	75+	N.A.	N.A.	N.A.
Prom. ≤ 75	39	100,9	240,534	0,8390

Se observa que los resultados obtenidos para $\alpha = 0,75$ y $\alpha = 0,25$ logran convergencia en todas las muestras y en un menor número de iteraciones que $\alpha = 0,5$. Se observa también una mejora en el factor de centro y de ancho con respecto al caso base de $\alpha = 0,75$. Se tiene que el caso $\alpha = 0,25$ presenta un mayor el mayor ancho de banda promedio, mientras que el caso $\alpha = 0,75$ presenta la mayor precisión en la frecuencia del band-gap, resultados coherentes con las diferentes prioridades de las funciones objetivo evaluadas. Las muestras asociadas a $\alpha = 0,95$ presentan un mejor ancho de banda en relación al caso base, pero un peor desempeño en el resto de métricas, incluyendo el porcentaje de convergencias satisfactorias. En cuanto a $\alpha = 0,05$, solo se observan 2 convergencias satisfactorias, presentando anchos significativamente mayores que el resto de muestras.

A continuación, se muestran los tiempos promedios calculados en alcanzar la convergencia / máximo de iteraciones para las diferentes combinaciones evaluadas, con un tiempo promedio por iteración de 373 [s]:

Tabla 4.22: Tiempos promedios hasta convergencia o máximo de iteraciones (enjambre de partículas).

Frecuencia [Hz]	α	Tiempo [s]	Tiempo [hr]
2000	0.5	28830	8,01
4000	0.5	26716	7,42
6000	0.05	21261	5,91
6000	0.25	7427	2,06
6000	0.5	14050	3,90
6000	0.75	9569	2,66
6000	0.95	17332	4,81

Se presenta también la información relativa a los tiempos mínimos, máximos y promedios observados para la ejecución del algoritmo de enjambre de partículas en una iteración:

Tabla 4.23: Tiempos máximos, mínimos y promedio por iteración observados en la ejecución del algoritmo de enjambre de partículas.

Frecuencia [Hz]	α	T Min. [s]	T Max. [s]	T Promedio [s]
2000	0.5	11	90	34
4000	0.5	7	110	36
6000	0.05	8	36	15
6000	0.25	14	55	31
6000	0.5	13	81	38
6000	0.75	9	90	33
6000	0.95	6	68	18

A continuación, se exponen ejemplos de gráficos obtenidos en casos convergentes al aplicar la metodología bidimensional con enjambre de partículas. En particular, se muestran diagramas de banda con band-gap en torno a la frecuencia de diseño deseada, y diagramas correspondientes a la evolución de la frecuencia del enrejado con mejor función objetivo en cada iteración:

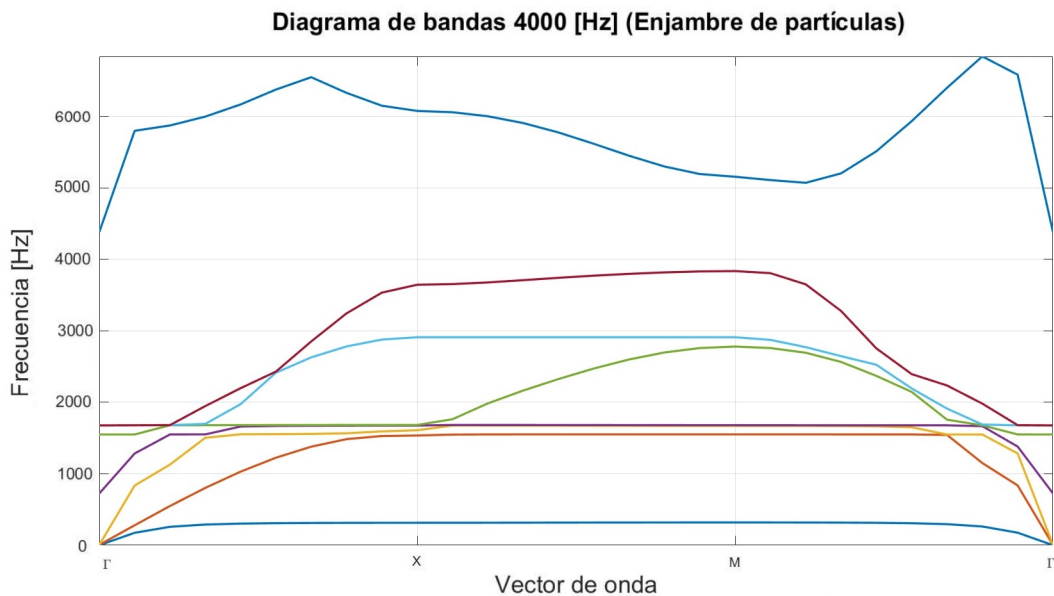


Figura 4.16: Ejemplo de diagrama de bandas con band-gap obtenido para 4000 [Hz] y $\alpha = 0,5$ (Enjambre de partículas).

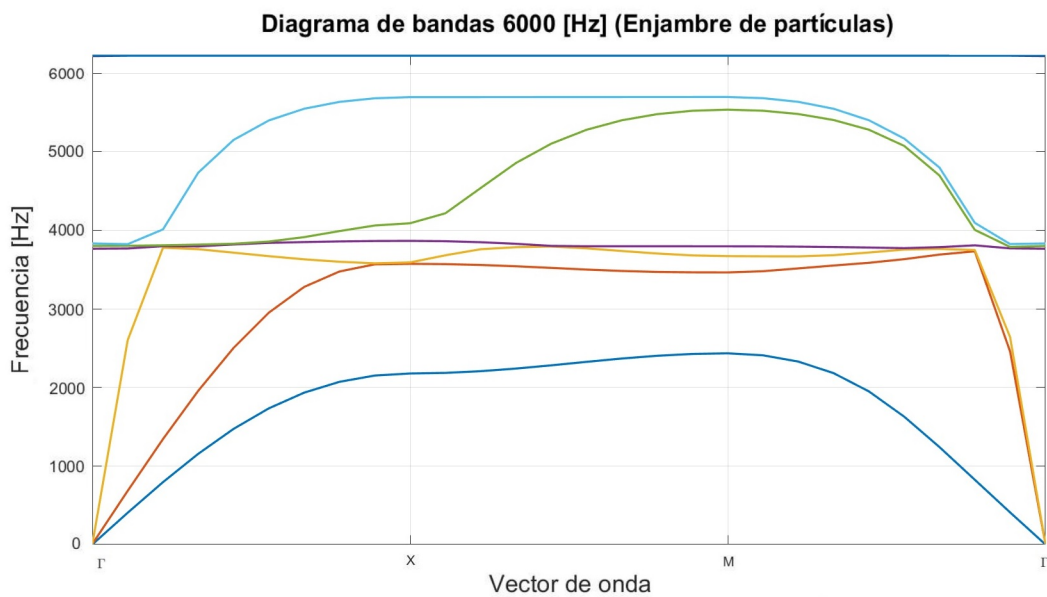


Figura 4.17: Ejemplo de diagrama de bandas con band-gap obtenido para 6000 [Hz] y $\alpha = 0,25$ (Enjambre de partículas).

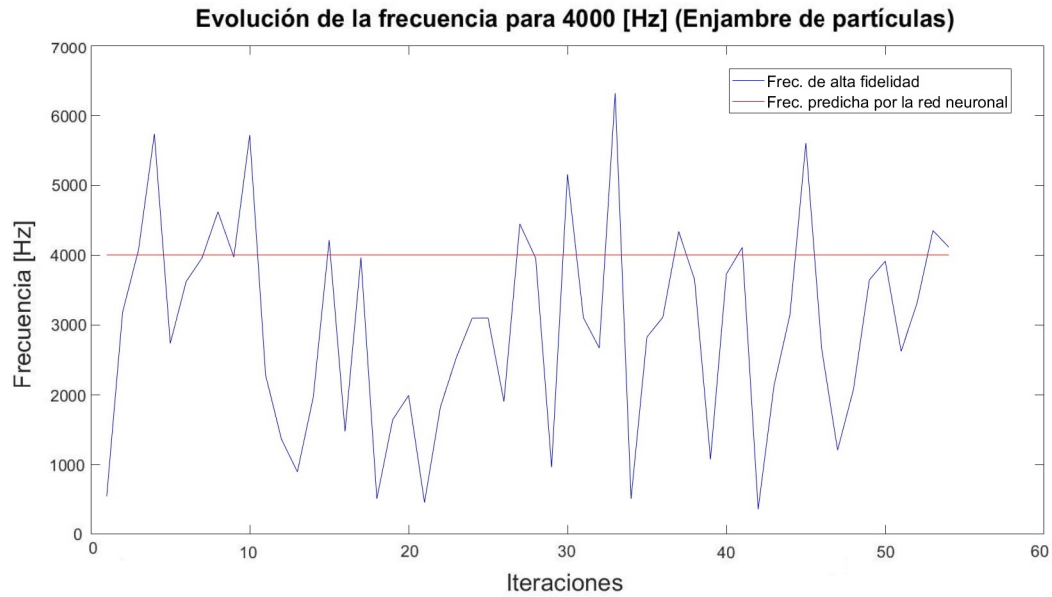


Figura 4.18: Ejemplo de evolución en iteraciones de la frecuencia de alta fidelidad del óptimo (azul) y de la predicha por la red (rojo) para 4000 [Hz] y $\alpha = 0,5$ (Enjambre de partículas).

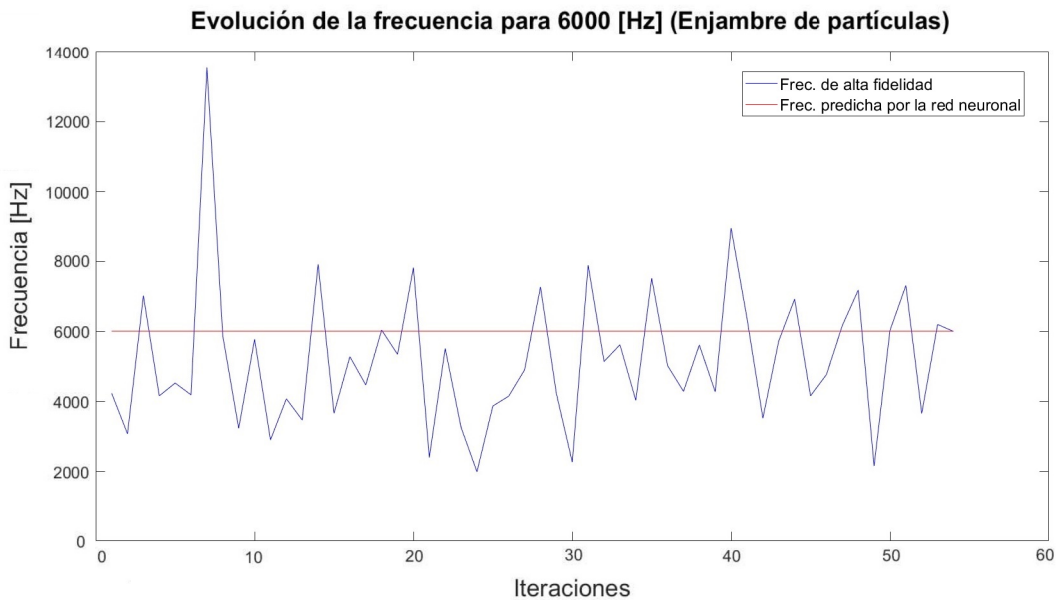


Figura 4.19: Ejemplo de evolución en iteraciones de la frecuencia de alta fidelidad del óptimo (azul) y de la predicha por la red (rojo) para 6000 [Hz] y $\alpha = 0,75$ (Enjambre de partículas).

A continuación, se muestran 2 ejemplos de diagrama describiendo la evolución de la magnitud por la que se incumple la condición de término 3.6 en el transcurso de la metodología,

donde la línea roja corresponde al cumplimiento mínimo de dicha condición:

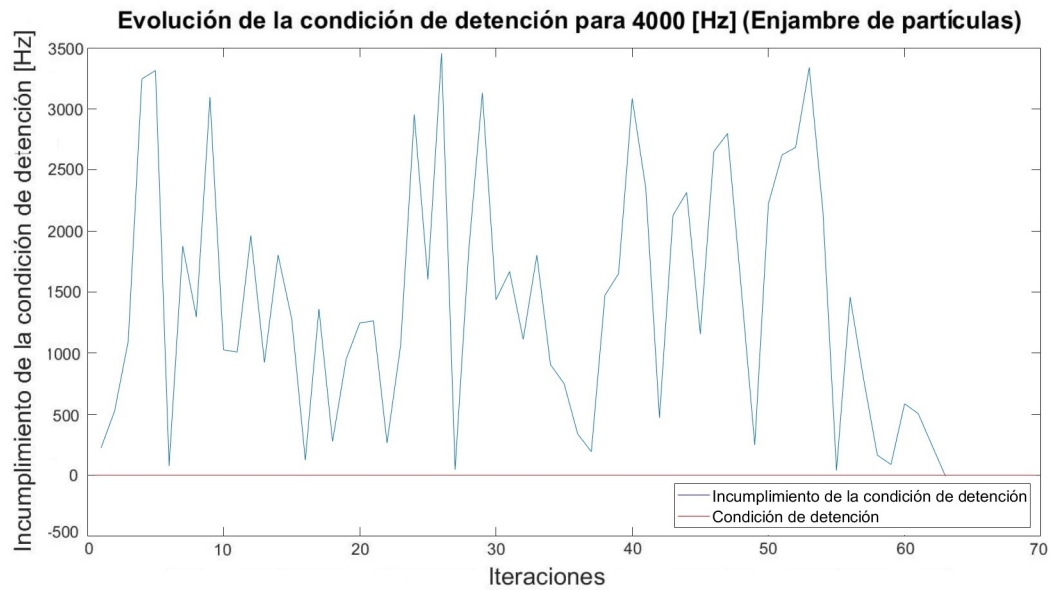


Figura 4.20: Ejemplo de evolución del incumplimiento de la condición de término (azul) para 4000 [Hz] y $\alpha = 0,5$ (Enjambre de partículas).

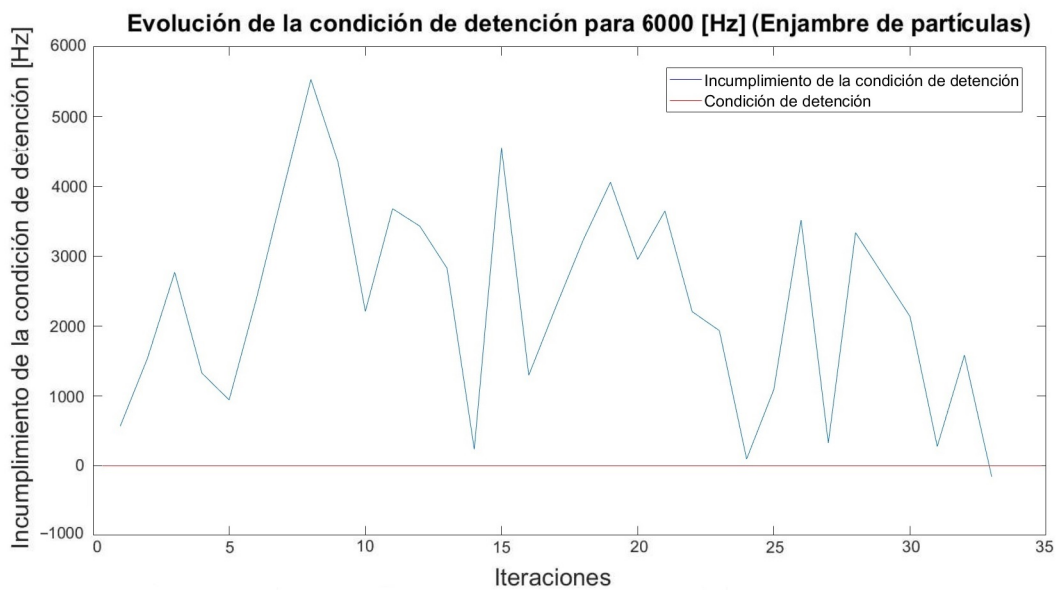


Figura 4.21: Ejemplo de evolución del incumplimiento de la condición de término (azul) para 6000 [Hz] y $\alpha = 0,5$ (Enjambre de partículas).

4.5. Resultados metodología tridimensional

Para evaluar la metodología tridimensional aplicada sobre enrejados de pirámide central se procedió a medir el promedio del número de iteraciones hasta la convergencia, la frecuencia y ancho real del enrejado resultante, y el factor de centro (ec. 3.7). Como se mencionó anteriormente, la metodología tridimensional se implementa solo con algoritmos genéticos, dados los mejores resultados observados en el caso bidimensional.

Al igual que en ambas metodologías en el caso bidimensional, se varió en primer lugar la frecuencia de diseño, utilizando $\alpha = 0,5$ en la función objetivo de la optimización (ec. 3.5). Nótese el valor “N.A.”, que toma la frecuencia, ancho y factor de centro al no haber convergencia satisfactoria. A continuación, se muestran los resultados obtenidos para frecuencias de diseño de 50000 [Hz], 70000 [Hz], 10000 [Hz], 130000 [Hz], y 150000 [Hz]:

Tabla 4.24: Resultados evaluación $5 \cdot 10^4$ [Hz] y $\alpha = 0,5$ (pirámide central).

Muestra	Iteraciones	Δ Frecuencia [Hz]	Ancho [Hz]	Factor de centro
1	30	965,6	2044	0,9450
2	17	503,4	1817	0,5542
3	9	280,0	1962	0,2855
4	4	1176,7	2392	0,9839
5	5	410,9	2096	0,3921
Prom. ≤ 75	24	667,3	1930	0,7496

Tabla 4.25: Resultados evaluación $7 \cdot 10^4$ [Hz] y $\alpha = 0,5$ (pirámide central).

Muestra	Iteraciones	Δ Frecuencia [Hz]	Ancho [Hz]	Factor de centro
1	4	1737,8	5906	0,5885
2	13	2294,1	5187	0,8846
3	18	2651,2	7002	0,7573
4	14	1225,5	5240	0,4677
5	15	882,6	4936	0,3576
Prom. ≤ 75	13	1758,2	5654	0,6111

Tabla 4.26: Resultados evaluación $1 \cdot 10^5$ [Hz] y $\alpha = 0,5$ (pirámide central).

Muestra	Iteraciones	Δ Frecuencia [Hz]	Ancho [Hz]	Factor de centro
1	3	2101,8	7424	0,5662
2	1	1571,5	3667	0,8571
3	6	969,8	7123	0,2723
4	4	1004,5	7332	0,2740
5	1	4873,9	12923	0,7543
Prom. ≤ 75	3	2104,3	7694	0,5448

Tabla 4.27: Resultados evaluación $1,3 \cdot 10^5$ [Hz] y $\alpha = 0,5$ (pirámide central).

Muestra	Iteraciones	Δ Frecuencia [Hz]	Ancho [Hz]	Factor de centro
1	1	3754,6	18953	0,3962
2	1	1264,3	19617	0,1289
3	2	2738,6	11681	0,4689
4	3	464,4	26538	0,0350
5	2	6630,0	18870	0,7027
Prom. ≤ 75	2	2970,4	19132	0,3463

Tabla 4.28: Resultados evaluación $1,5 \cdot 10^5$ [Hz] y $\alpha = 0,5$ (pirámide central).

Muestra	Iteraciones	Δ Frecuencia [Hz]	Ancho [Hz]	Factor de centro
1	2	1993,6	4704	0,8476
2	1	8649,3	19178	0,9020
3	2	11562,1	23299	0,9925
4	1	1491,2	5211	0,5723
5	6	7690,3	15475	0,9939
Prom. ≤ 75	2	6277,3	13573	0,8617

Se observa que la frecuencia con mejores resultados es $1,3 \cdot 10^5$ [Hz]. Se procede de igual forma que en ambas metodologías bidimensionales, utilizando la frecuencia con los mejores resultados para evaluar el rendimiento al variar α .

A continuación, se muestran los resultados obtenidos para $\alpha = 0,05$ (alta prioridad al ancho de banda), $\alpha = 0,25$, $\alpha = 0,75$ y $\alpha = 0,95$ (alta prioridad a la frecuencia):

Tabla 4.29: Resultados evaluación $1,3 \cdot 10^5$ [Hz] y $\alpha = 0,05$ (pirámide central).

Muestra	Iteraciones	Δ Frecuencia [hz]	Ancho [hz]	Factor de centro
1	2	3120,5	17665	0,3533
2	1	1758,1	21532	0,1633
3	3	6065,0	22878	0,5302
4	1	5542,2	27022	0,4102
5	2	9098,6	19159	0,9498
Prom. ≤ 75	2	5116,9	21651	0,4813

Tabla 4.30: Resultados evaluación $1,3 \cdot 10^5$ [Hz] y $\alpha = 0,25$ (pirámide central).

Muestra	Iteraciones	Δ Frecuencia [Hz]	Ancho [Hz]	Factor de centro
1	1	5355,9	22386	0,4785
2	1	1495,1	20665	0,1447
3	1	7536,8	24277	0,6209
4	1	3583,9	20317	0,3528
5	1	4926,9	15924	0,6188
Prom. ≤ 75	1	4579,7	20714	0,4431

Tabla 4.31: Resultados evaluación $1,3 \cdot 10^5$ [Hz] y $\alpha = 0,75$ (pirámide central).

Muestra	Iteraciones	Δ Frecuencia [Hz]	Ancho [Hz]	Factor de centro
1	4	2086,6	9552	0,4369
2	1	3255,8	15653	0,416
3	1	1387,0	22069	0,1257
4	4	580,2	8732	0,1329
5	1	8729,5	21422	0,815
Prom. ≤ 75	2	3207,8	15486	0,3853

Tabla 4.32: Resultados evaluación $1,3 \cdot 10^5$ [Hz] y $\alpha = 0,95$ (pirámide central).

Muestra	Iteraciones	Δ Frecuencia [hz]	Ancho [hz]	Factor de centro
1	7	190,3	504	0,7548
2	1	903,5	13720	0,1317
3	1	740,9	19473	0,0761
4	3	2342,3	22631	0,207
5	1	5787,5	14044	0,8242
Prom. ≤ 75	3	1992,9	14074	0,39876

Se observa que el caso base es el que tiene los mejores factores de centro, presentando los casos de $\alpha = 0,05$ y $\alpha = 0,25$ band-gaps más anchos, mientras que $\alpha = 0,75$ y $\alpha = 0,95$ presentan saltos de banda más estrechos. Se logra 100 % de convergencias satisfactorias en todos los casos con un número similar de iteraciones, destacando $\alpha = 0,25$ con convergencia en una iteración en todas las muestras.

A continuación, se muestran los tiempos promedios calculados en alcanzar la convergencia / máximo de iteraciones para las diferentes combinaciones evaluadas, con un tiempo promedio por iteración de 1254 [s]:

Tabla 4.33: Tiempos promedios hasta convergencia o máximo de iteraciones (pirámide central).

Frecuencia [Hz]	α	Tiempo [s]	Tiempo [hr]
$5 \cdot 10^4$	0.5	16709	4,64
$7 \cdot 10^4$	0.5	16184	4,50
$1 \cdot 10^5$	0.5	3834	1,07
$1,3 \cdot 10^5$	0.05	2321	0,64
$1,3 \cdot 10^5$	0.25	1225	0,34
$1,3 \cdot 10^5$	0.5	2418	0,67
$1,3 \cdot 10^5$	0.75	2730	0,76
$1,3 \cdot 10^5$	0.95	3391	0,94
$1,5 \cdot 10^5$	0.5	2912	0,81

Se presenta también la información relativa a los tiempos mínimos, máximos y promedios observados para la ejecución del algoritmo de enjambre de partículas en una iteración:

Tabla 4.34: Tiempos máximos, mínimos y promedio por iteración observados en la ejecución del algoritmo genético (pirámide central).

Frecuencia [Hz]	α	T Min. [s]	T Max. [s]	T Promedio [s]
$5 \cdot 10^4$	0.5	48	157	122
$7 \cdot 10^4$	0.5	48	157	129
$1 \cdot 10^5$	0.5	96	130	118
$1,3 \cdot 10^5$	0.05	49	143	96
$1,3 \cdot 10^5$	0.25	60	60	60
$1,3 \cdot 10^5$	0.5	80	109	95
$1,3 \cdot 10^5$	0.75	97	130	121
$1,3 \cdot 10^5$	0.95	136	169	157
$1,5 \cdot 10^5$	0.5	62	88	78

A continuación, se exponen ejemplos de gráficos obtenidos en casos convergentes al aplicar la metodología tridimensional. En particular, se muestran diagramas de banda con band-gap en torno a la frecuencia de diseño deseada, y diagramas correspondientes a la evolución de la frecuencia del enrejado con mejor función objetivo en cada iteración:

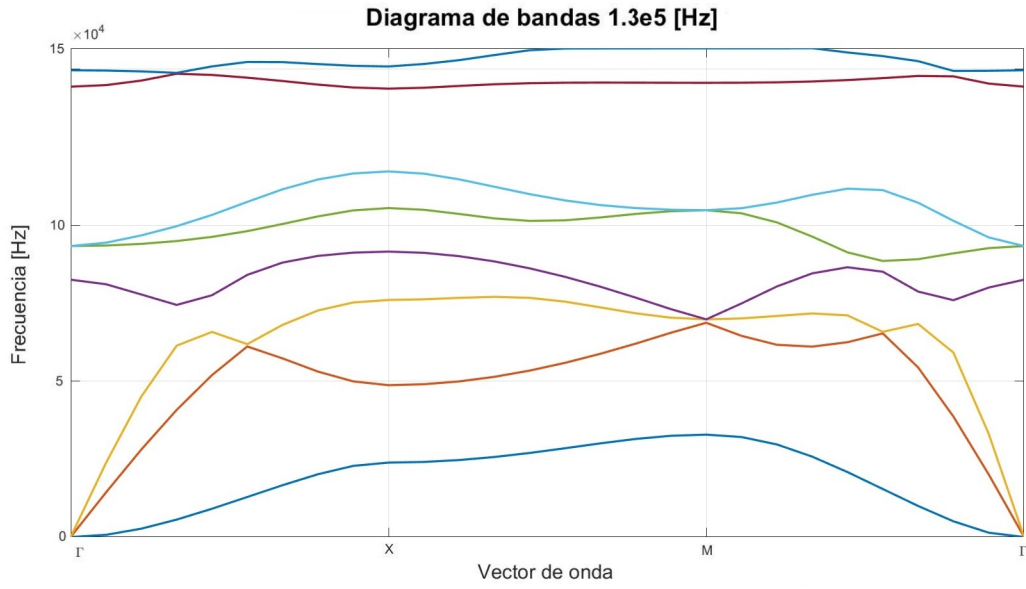


Figura 4.22: Ejemplo de diagrama de bandas con band-gap obtenido para $1,3 \cdot 10^5$ [Hz] y $\alpha = 0,5$ (pirámide central).

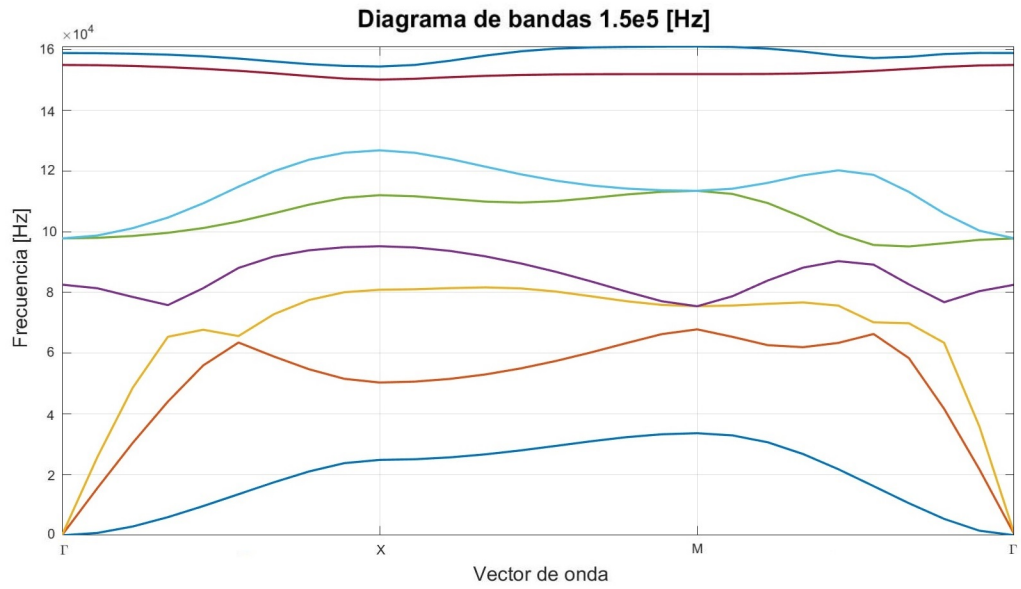


Figura 4.23: Ejemplo de diagrama de bandas con band-gap obtenido para $1,5 \cdot 10^5$ [Hz] y $\alpha = 0,5$ (pirámide central).

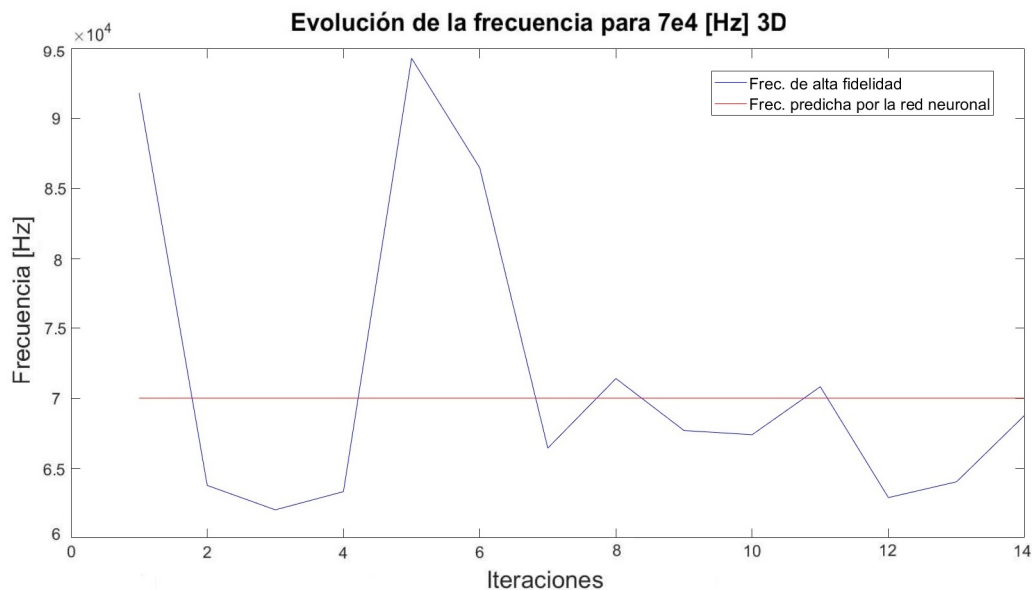


Figura 4.24: Ejemplo de evolución en iteraciones de la frecuencia de alta fidelidad del óptimo (azul) y de la predicha por la red (rojo) para $7 \cdot 10^4$ [Hz] y $\alpha = 0,5$ (pirámide central).

A continuación, se muestra un ejemplo de diagrama describiendo la evolución de la magnitud por la que se incumple la condición de término 3.6 en el transcurso de la metodología, donde la línea roja corresponde al cumplimiento mínimo de dicha condición:

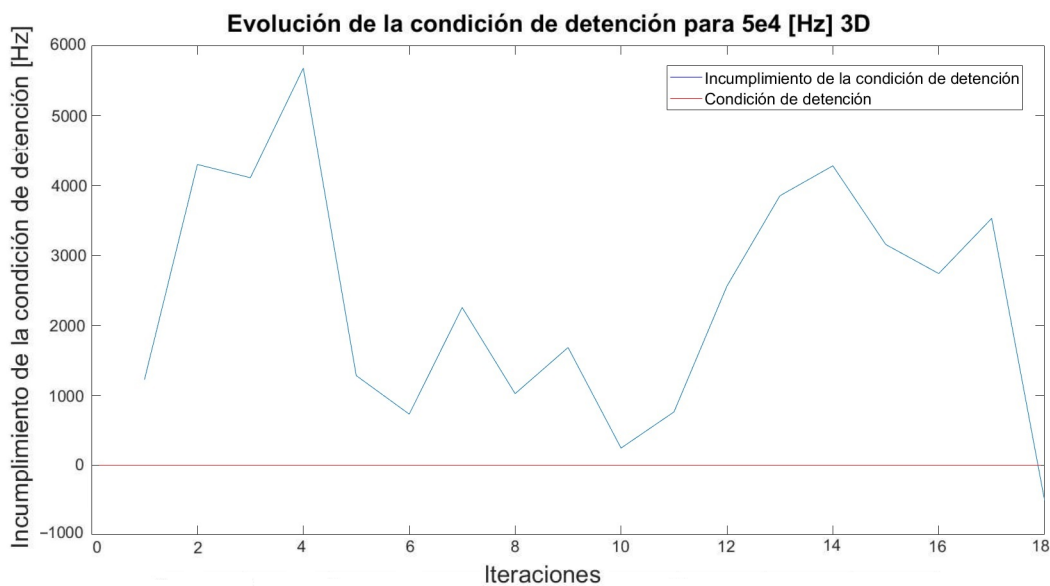


Figura 4.25: Ejemplo de evolución del incumplimiento de la condición de término (azul) para $5 \cdot 10^4$ [Hz] y $\alpha = 0,5$ (Pirámide central).

4.6. Resultados optimización analítica

Para evaluar la utilidad de las metodologías construidas, se midió el tiempo requerido para lograr optimización con algoritmos genéticos y una función objetivo analítica, así como el factor de centro, frecuencia, y ancho del band-gap. A continuación se muestran los resultados obtenidos para el caso bidimensional:

Tabla 4.35: Evaluación de la optimización analítica.

F. Diseño [Hz]	Tiempo [s]	Tiempo [h]	F. obtenida [Hz]	Ancho [Hz]	F. de centro
2000	54762	15,2	2000,0	75,9	0,00000
4000	79091	22,0	4000,1	493,9	0,00027
6000	85296	23,7	5999,9	1048,5	0,00019

En el caso tridimensional, no se logró convergencia mediante la metodología analítica en tiempos razonables. Esto probablemente se debe al mayor tiempo que requiere el cálculo analítico de un diagrama de bandas de enrejado tridimensional en comparación con el caso bidimensional. En el caso de pirámide central, este cálculo toma aproximadamente 1 [min] por enrejado. Considerando las poblaciones de 200 individuos por generación en el algoritmo genético (valor estándar, equivalente al utilizado en el resto del presente trabajo) y el hecho de que una optimización satisfactoria con redes neuronales para el caso tridimensional toma del orden de 150 generaciones, se puede estimar un cálculo necesario de 30000 enrejados, tomando un tiempo aproximado de 500 [h] o 21 [días].

4.7. Mejores enrejados: Conjunto de entrenamiento

A continuación, en las tablas 4.36 y 4.37, se muestra la comparación entre los valores promedio de las funciones objetivo obtenidas para las mejores metodologías en los casos 2D y 3D, y los valores obtenidos localizando los mejores elementos en el conjunto de entrenamiento. Dado que se busca minimizar la función objetivo, se consideran mejores los valores más bajos. Los resultados destacados en rojo corresponden a enrejados cuyo band-gap no incluye la frecuencia de diseño:

Tabla 4.36: Funciones objetivo en los mejores elementos del conjunto de entrenamiento 2D.

F. Diseño [Hz]	α	F.O. Metodología A.G.	F.O. Entrenamiento
2000	0,5	-73,86	-105,20
4000	0,5	-30,99	-72,49
6000	0,05	N.A.	-890,48
6000	0,25	-63,76	-357,33
6000	0,5	-74,17	16,98
6000	0,75	-109,86	32,09
6000	0,95	-2,09	13,75

Si bien para $7 \cdot 10^4$ [Hz] el enrejado con la mejor función objetivo no cumple con la condición de detención, sí existen enrejados que la cumplen en el conjunto de entrenamiento

Tabla 4.37: Funciones objetivo en los mejores elementos del conjunto de entrenamiento 3D.

F. Diseño [Hz]	α	F.O. Metodología	F.O. Entrenamiento
$5 \cdot 10^4$	0,5	-697,31	-598,10
$7 \cdot 10^4$	0,5	-1947,89	-1573,44
$1 \cdot 10^5$	0,5	-2794,80	-3208,90
$1,3 \cdot 10^5$	0,05	-826,72	-14499,08
$1,3 \cdot 10^5$	0,25	-4033,52	-11263,32
$1,3 \cdot 10^5$	0,5	-8080,71	-7218,62
$1,3 \cdot 10^5$	0,75	-9208,32	-3173,93
$1,3 \cdot 10^5$	0,95	-11477,47	61,83
$1,5 \cdot 10^5$	0,5	-3648,09	-457,73

(aunque con peor función objetivo que el resultado obtenido para la metodología). En el resto de casos destacados con rojo, no existen enrejados en el conjunto de entrenamiento que cumplan la condición de término, es decir, que incluyan a la frecuencia de diseño en su band-gap.

4.8. Convergencia del modelo subrogado

A continuación, en las figuras 4.26 y 4.27, se muestran los gráficos de la evolución de la condición de detención para el caso 2D y 3D según lo detallado en la sección anterior (ec. 3.9):

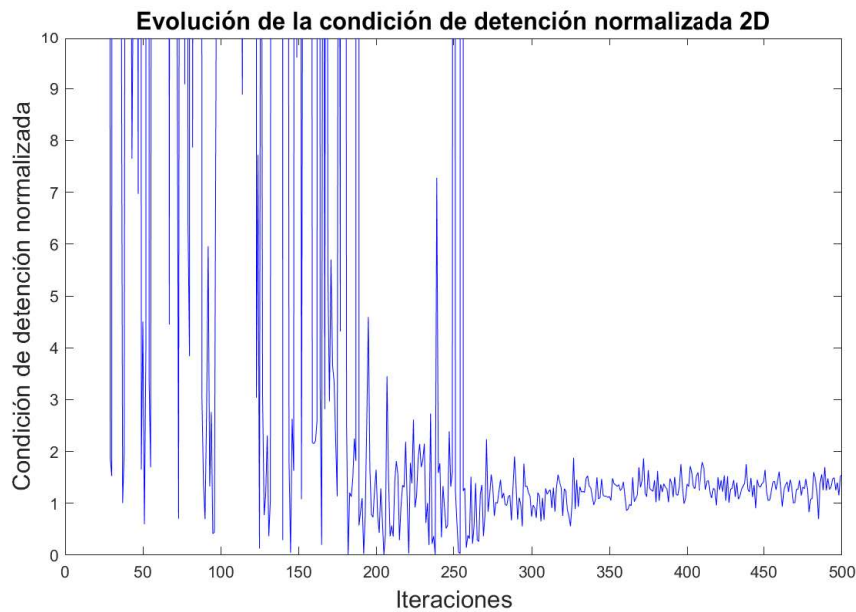


Figura 4.26: Evolución de la condición de detención a lo largo de 500 iteraciones para la metodología 2D con algoritmo genético.

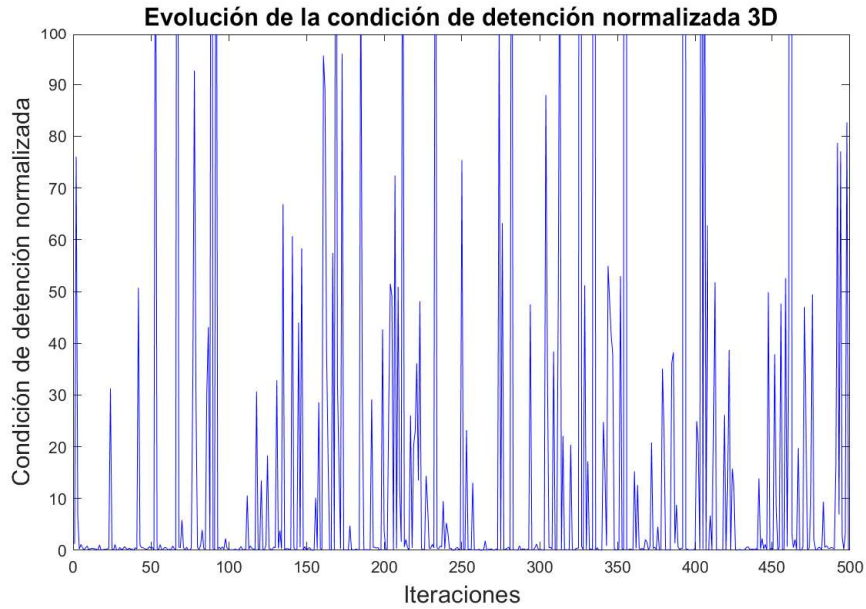


Figura 4.27: Evolución de la condición de detención a lo largo de 500 iteraciones para la metodología 3D.

En las figuras 4.28, 4.29, 4.31 y 4.32 se muestran los errores normalizados de la frecuencia y ancho en los casos bidimensional (con algoritmo genético) y tridimensional, respectivamente. Al calcular los promedios de estos valores, se obtiene el error normalizado medio (ec. 3.8), cuya evolución se muestra en las figuras 4.30 y 4.33 para los casos 2D y 3D. Se señala con una línea vertical roja el punto correspondiente a la primera iteración donde se alcanza la condición de detención:

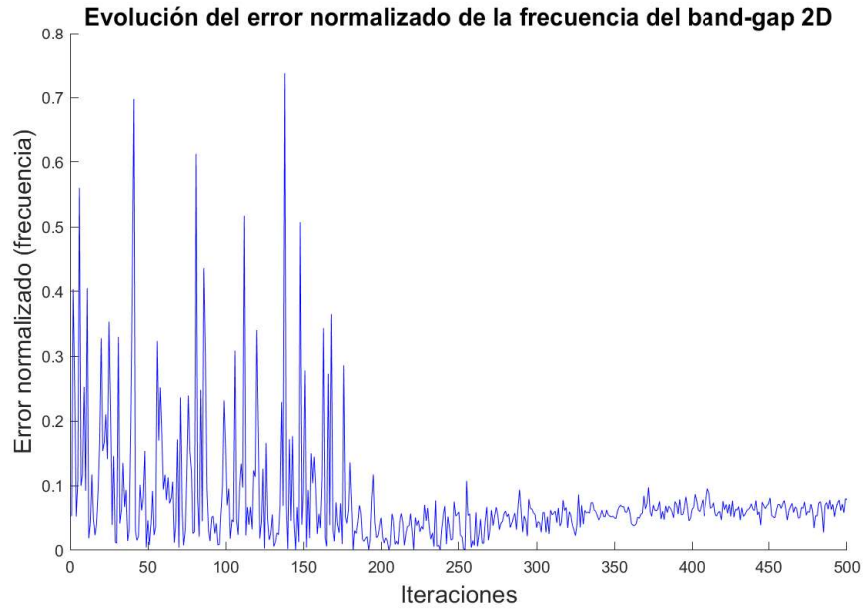


Figura 4.28: Evolución del error normalizado de la frecuencia a lo largo de 500 iteraciones para la metodología 2D con algoritmo genético.

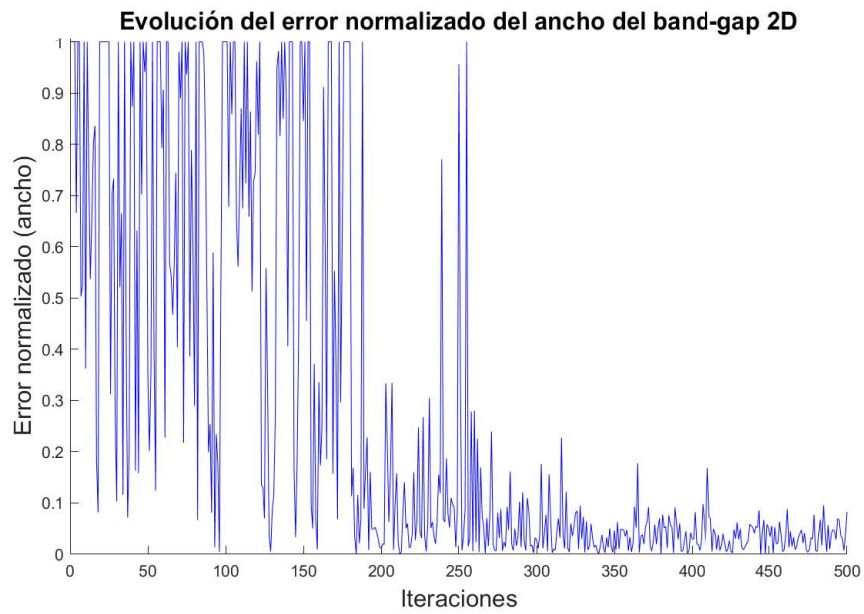


Figura 4.29: Evolución del error normalizado del ancho a lo largo de 500 iteraciones para la metodología 2D con algoritmo genético.

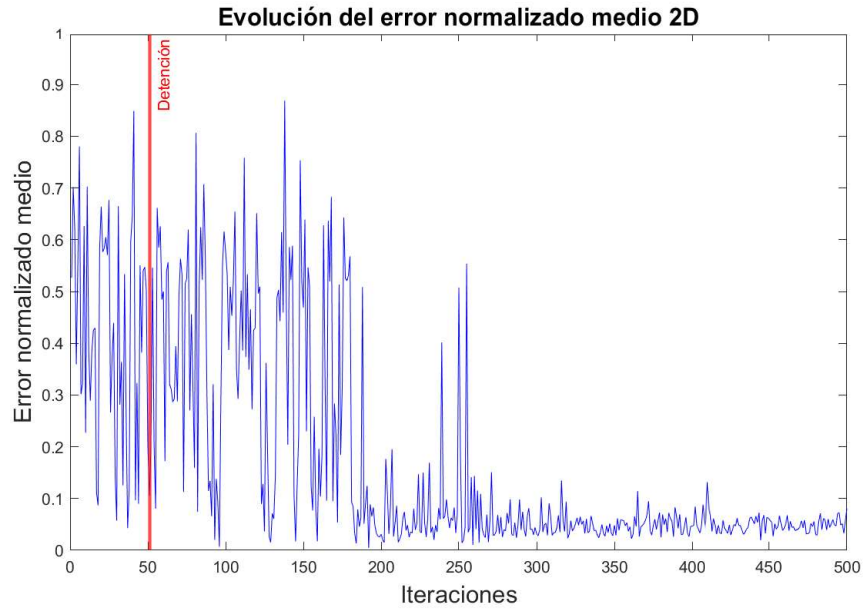


Figura 4.30: Evolución del error normalizado a lo largo de 500 iteraciones para la metodología 2D con algoritmo genético.

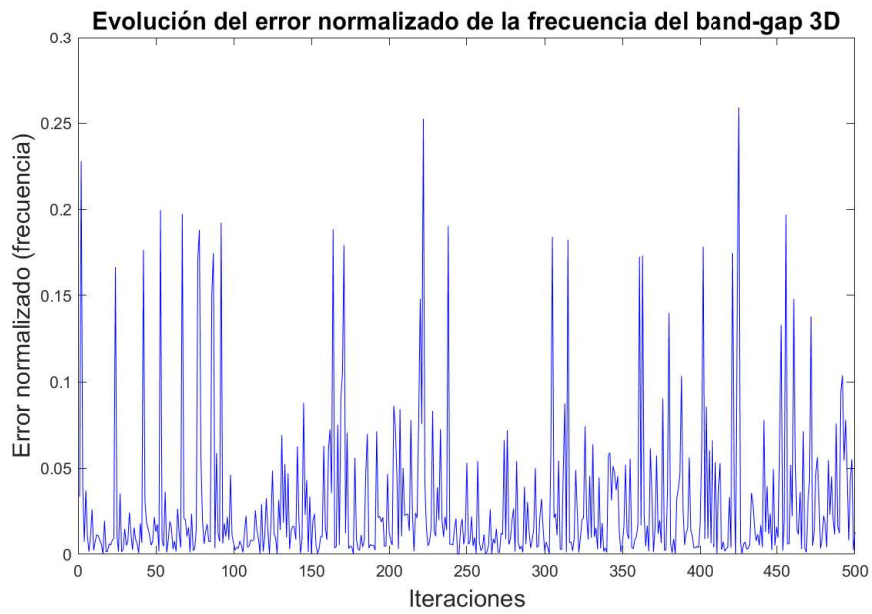


Figura 4.31: Evolución del error normalizado de la frecuencia a lo largo de 500 iteraciones para la metodología 3D.

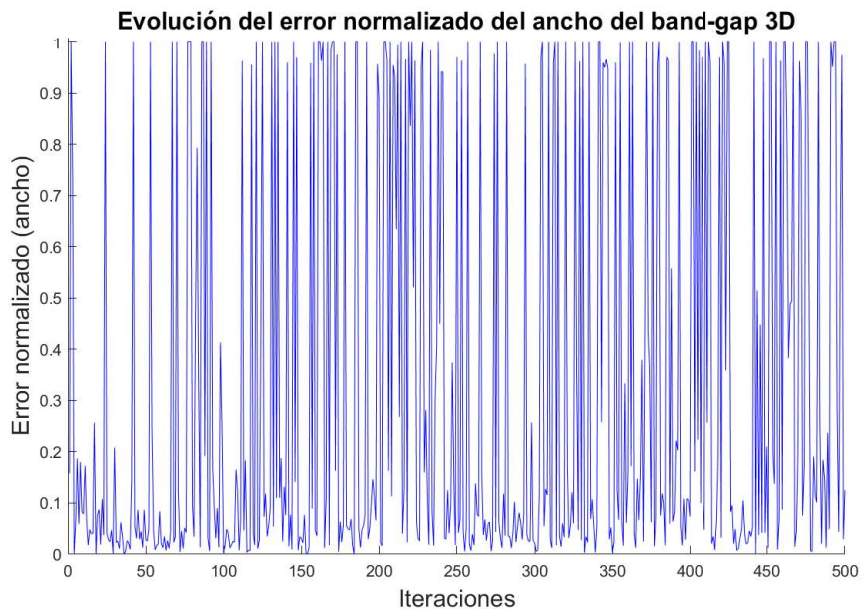


Figura 4.32: Evolución del error normalizado del ancho a lo largo de 500 iteraciones para la metodología 3D.

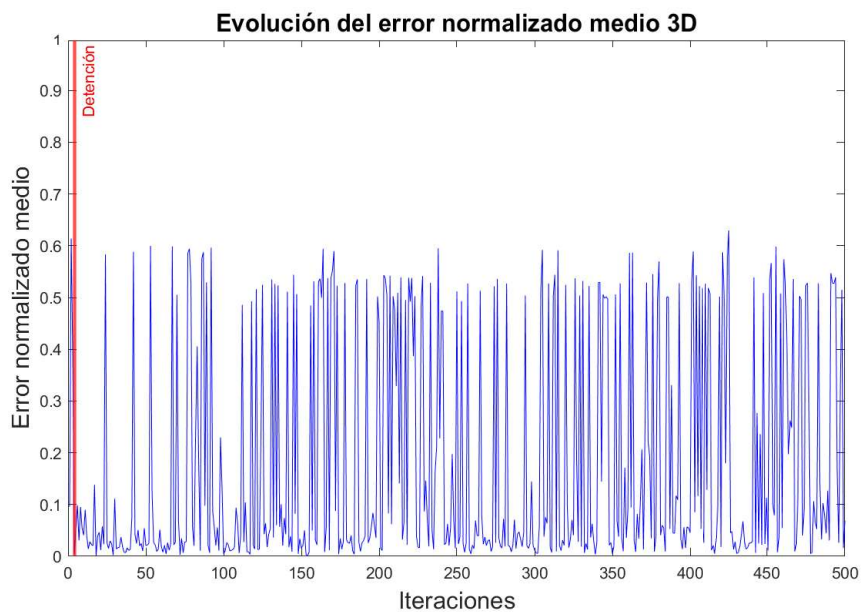


Figura 4.33: Evolución del error normalizado a lo largo de 500 iteraciones para la metodología 3D.

5. Discusión y Análisis

Respecto a los resultados obtenidos al evaluar las redes neuronales del caso bidimensional, en la tabla 4.3 se observa que los coeficientes de determinación asociados a la frecuencia de los band-gap, su ancho, y la frecuencia en el caso de ancho positivo, son mejores en la red neuronal de máximos y mínimos que en la de predicción directa. En cuanto al R^2 asociado al ancho de banda en caso de band-gap positivo, ambas redes presentan valores negativos, lo que se corresponde con una mala capacidad de predicción de esta variable. Esta deficiencia en la capacidad predictiva de las redes podría estar relacionada con la desproporción que existe en el conjunto de entrenamiento entre enrejados con y sin band-gap, ya que al predominar significativamente estos últimos, favorecen la predicción de anchos negativos al ajustar la red durante su entrenamiento. Por otro lado, la red de predicción directa logra tiempos de entrenamiento sustancialmente mejores que los de la red de máximos y mínimos, lo que justifica su uso en las metodologías dado el impacto directo que esto tiene sobre el tiempo de ejecución total.

En cuanto a la red neuronal de predicción directa obtenida para el caso tridimensional, como se detalla en la tabla 4.5, se observa un mejor rendimiento en comparación con ambas redes del caso bidimensional. Se repite la obtención de un valor negativo para el R^2 asociado al ancho de banda en caso de band-gap positivo, lo que se asocia a una deficiencia en la capacidad predictiva de la red, y podría estar nuevamente relacionado con la desproporción entre elementos con y sin band-gap en el conjunto de entrenamiento, que se acentúa aún más en el caso tridimensional. El tiempo logrado por esta red se encuentra por sobre el tiempo de la red de predicción directa en el caso bidimensional, lo que tendería a generar iteraciones más largas en la metodología tridimensional.

En relación a la metodología bidimensional con algoritmo genético, se observa en las tablas 4.6, 4.7 y 4.8 como al variar la frecuencia de diseño, también varía el rendimiento en las diferentes métricas. En primer lugar, se observa como el número de iteraciones promedio, así como la cantidad de convergencias satisfactorias alcanzadas (75 iteraciones o menos) mejora al incrementar la frecuencia, presentando el caso de 6000 [Hz] la mayor efectividad incluso al haber menos elementos con band-gap significativos en el conjunto de entrenamiento al comparar con 2000 [Hz] y 4000 [Hz] (Ver figura 3.11). En cuanto al ancho de banda, se observan resultados similares entre los casos con 2000 [Hz] y 6000 [Hz], y peores resultados en 4000 [Hz], y respecto al factor de centro, se observan los mejores resultados en 4000 [Hz], seguido por 2000 [Hz] y 6000 [Hz]. Sin embargo, los resultados de ancho de banda y factor de centro no son completamente comparables entre frecuencias, ya que solo se calculan considerando las convergencias satisfactorias, que especialmente en 2000 [Hz] y 4000 [Hz] solo corresponde a una fracción de las muestras, por lo que podrían no ser lo suficientemente representativas.

Como se observa en las tablas 4.9, 4.10, 4.11, 4.12, al variar α , y por lo tanto la prioridad que da la función objetivo del algoritmo de optimización a la frecuencia y ancho del band-gap, las métricas de evaluación de la metodología nuevamente varían. Si bien el caso con $\alpha = 0,5$ logra convergencia en menos iteraciones, se observa que el promedio de los anchos de banda alcanzados es superior para $\alpha = 0,25$, $\alpha = 0,75$ y $\alpha = 0,95$. Para $\alpha = 0,05$ no se observa convergencia satisfactoria en ninguna de las muestras. Se observa también una mejora en el factor de centro en las muestras con $\alpha = 0,25$ y $\alpha = 0,95$ con respecto al caso base $\alpha = 0,5$, aunque en el último caso los resultados no son completamente comparables ya que solo 2 de las 5 muestras convergen satisfactoriamente.

En relación a la metodología bidimensional con enjambre de partículas, se observa en las tablas 4.15, 4.16 y 4.17 como al variar la frecuencia de diseño, también varía el rendimiento en las diferentes métricas. En primer lugar, se observa como, nuevamente, el número de iteraciones promedio, así como la cantidad de convergencias satisfactorias alcanzadas (75 iteraciones o menos) mejora al incrementar la frecuencia, destacando el que la totalidad de las muestras en 2000 [Hz] resulta en convergencia no satisfactoria. En cuanto al ancho de banda y factor de centro, se observan resultados significativamente mejores para 4000 [Hz] que para 6000 [Hz], sin embargo, nuevamente estos resultados no son comparables por la predominancia de las convergencias no satisfactorias en 4000 [Hz].

Como se observa en las tablas 4.18, 4.19, 4.20 y 4.21, al variar α se tiene que el caso base es aquel con los band-gaps más estrechos, observándose los mayores anchos en $\alpha = 0,05$ y $\alpha = 0,25$. Se observa también un número de iteraciones similar con la excepción de $\alpha = 0,95$, que requiere más iteraciones hasta la convergencia. En cuanto a la precisión en la frecuencia, no se observa un comportamiento claro en función de α . En general los mejores resultados se observan en $\alpha = 0,25$.

Las tablas 4.23 y 4.14 muestran como el algoritmo genético toma tiempos significativamente mayores por iteración al comparar con enjambre de partículas, sin embargo esto no se traduce en una mejora real en la metodología, ya que con algoritmo genético se logra convergencia de forma más robusta y en un mayor rango de frecuencias. Se destaca que en ambos casos, el entrenamiento de la red neuronal toma un tiempo sustancialmente mayor a la ejecución del algoritmo de optimización global.

En cuanto a la metodología tridimensional, se observa en las tablas 4.24, 4.25, 4.26, 4.27 y 4.28 como al variar la frecuencia de diseño, también varía el rendimiento en las diferentes métricas. Se tiene que tanto el número de iteraciones promedio como la cantidad de convergencias satisfactorias alcanzadas (75 iteraciones o menos) mejora al incrementar la frecuencia hasta $1,3 \cdot 10^5$ [Hz], manteniéndose aproximadamente constante en $1,5 \cdot 10^5$ [Hz]. Se tiene que los mayores anchos de banda promedio alcanzados se obtienen al evaluar en $1,3 \cdot 10^5$ [Hz], seguido de $1,5 \cdot 10^5$ [Hz], y que los mejores factores de centro se obtienen nuevamente al

evaluar $1,3 \cdot 10^5$ [Hz], seguido de 10^5 [Hz].

Como se evidencia en las tablas 4.29, 4.30, 4.31 y 4.32, el caso base es el que tiene los mejores factores de centro. Respecto a los anchos de band-gap, los casos de $\alpha = 0,05$ y $\alpha = 0,25$ presentan mejores resultados al comparar con $\alpha = 0,5$, mientras que $\alpha = 0,75$ y $\alpha = 0,95$ presentan saltos de banda más estrechos. Se logra 100 % de convergencias satisfactorias en todos los casos con un número similar de iteraciones, destacando $\alpha = 0,25$ con convergencia en una iteración en todas las muestras.

En las tablas 4.13, 4.22 y 4.33 se observan los tiempos promedios requeridos para alcanzar la convergencia o el máximo de iteraciones en las tres metodologías. Para la metodología bidimensional con algoritmos genéticos, se observa como el mejor rendimiento se alcanza para la frecuencia de 6000 [Hz] con $\alpha = 0,5$, seguido de la misma frecuencia con $\alpha = 0,25$. Por otro lado, para la metodología con enjambre de partículas se tiene que el mejor desempeño se logra para la frecuencia de 6000 [Hz] con $\alpha = 0,25$, seguido de la misma frecuencia con $\alpha = 0,75$. Por último, en el caso de la metodología tridimensional, los mejores resultados se observan para la frecuencia de $1,3 \cdot 10^5$ [Hz] con $\alpha = 0,25$, seguido de la misma frecuencia con $\alpha = 0,05$, $\alpha = 0,5$, y de $\alpha = 0,75$. Las siguientes mejores frecuencias evaluadas son $1,5 \cdot 10^5$ [Hz] seguido de $1 \cdot 10^5$ [Hz], teniendo $5 \cdot 10^4$ [Hz] los peores rendimientos.

En las múltiples figuras (4.12 a 4.13, 4.18 a 4.19, y 4.24 asociadas a la evolución de la frecuencia predicha por la red para el mejor elemento en cada iteración, se observa un comportamiento complejo que no presenta características convergentes. Algo similar ocurre al observar las figuras 4.14, 4.15, 4.20, 4.21 y 4.25, donde no se aprecia una tendencia clara.

En las figuras 4.12 a 4.13, 4.18 a 4.19, y 4.24, asociadas a la evolución de la frecuencia predicha, se observa como en la mayoría de los casos es constante e igual a la frecuencia de diseño. En estos casos, se evidencia que el valor de α es demasiado alto para permitir variaciones en la frecuencia de los enrejados óptimos en el funcionamiento interno del conjunto red neuronal-algoritmo de optimización global. Este comportamiento se deja de observar cuando se utilizan valores significativamente bajos para α , como 0,05, donde se obtienen gráficos como los de la Figura 4.13. La tendencia a presentar frecuencias predichas constantes e iguales a la de diseño puede explicar la baja correlación entre la precisión en la frecuencia y α , ya que dado que el resultado del algoritmo genético es el mismo para todos los $\alpha > 0,05$, la variación observada tiene mayor relación con la diferencia de precisión entre el modelo subrogado y el de alta fidelidad que con el valor de α .

Al comparar las tablas 4.13 y 4.35 se observa como para todas las frecuencias evaluadas en el caso bidimensional con algoritmos genéticos, la metodología de optimización subrogada presenta tiempos de ejecución sustancialmente menores que la metodología analítica. Sin embargo, estos resultados no son completamente comparables ya que, especialmente en los

casos de 2000 [hz] y 6000 [Hz] con $\alpha = 0,05$, no en todas las ejecuciones se logra convergencia, lo que sí ocurre en el caso analítico.

En cuanto a la metodología subrogada tridimensional, se observan tiempos de ejecución significativamente menores al comparar con la metodología analítica, donde no se logra convergencia alguna en los tiempos disponibles. En particular, en la tabla 4.33 se aprecia como los resultados son incluso mejores que en el caso bidimensional, mientras que los tiempos de ejecución del caso analítico rondarían 500 [h] o 21 [días] según la estimación realizada en la sección anterior, lo que corresponde a tiempos entre 10^2 y 10^3 veces lo requerido por la metodología de optimización subrogada.

En las tablas 4.36 y 4.37 se aprecia como incluso un conjunto de 20000 enrejados no es suficiente para contener enrejados con band-gap de forma consistente en torno a las frecuencias de diseño evaluadas. En cambio, para todas las frecuencias evaluadas la metodología logra diseñar enrejados que en general tienen funciones objetivo comparables o mejores a las de los elementos del conjunto de entrenamiento.

En cuanto a la convergencia entre los modelos subrogados y de alta fidelidad, en la Figura 4.26 claramente se observa un comportamiento convergente acercándose a valores cercanos a 1, valor que representa la condición de detención. En la Figura 4.27 nuevamente se ve un comportamiento general decreciente de la condición de detención, sin embargo en este caso es menos evidente que en el caso bidimensional.

Por último, respecto a la evolución del error normalizado, en las figuras 4.28, 4.29 y 4.30 se observa nuevamente un claro comportamiento convergente, lo que evidencia el refinamiento del modelo subrogado. Se evidencia también que la condición de detención se alcanza en un número significativamente menor de iteraciones que la convergencia entre ambos modelos. Por otro lado, en las figuras 4.31, 4.32 y 4.33, si bien existe una alta densidad de valores cercanos a 0, no se observa un comportamiento convergente claro.

6. Conclusión

Una vez desarrolladas y evaluadas las metodologías tratadas en el presente trabajo, resulta posible afirmar que la hipótesis planteada inicialmente es cierta, ya que efectivamente se logró, tanto para el caso bidimensional como tridimensional, diseñar metodologías de optimización subrogada que permiten la obtención de enrejados con band-gaps en torno a diferentes frecuencias de diseño, observándose en general mejoras significativas en los recursos computacionales requeridos respecto al caso analítico equivalente.

En cuanto a los modelos subrogados es posible afirmar que, en el caso bidimensional, la mejor red neuronal para la predicción de los band-gaps entre las alternativas consideradas es la de predicción directa, por sobre la red de máximos y mínimos. Aunque ambas redes presentan una mala capacidad de predicción de ancho de banda cuando su valor es positivo y existe un amplio margen de mejora, el desempeño de la red directa es suficiente para lograr elaborar las metodologías de optimización subrogada propuestas. Por otro lado, la evolución de los valores predichos por las redes neuronales presentan un comportamiento complejo sin características convergentes, careciendo de relaciones claras con la cercanía al fin de la optimización.

De los resultados obtenidos para el caso bidimensional, se puede concluir que en general, aunque requiera mayores tiempos por iteración, algoritmos genéticos resulta una mejor alternativa como algoritmo de optimización global que enjambre de partículas, ya que se obtienen menores tiempos de ejecución, convergencia de forma más robusta, y se logra optimizar en torno a un mayor rango de frecuencias.

Es posible afirmar que la robustez y rapidez con las que las metodologías convergen están altamente relacionadas con las frecuencias de diseño, con tendencia a obtener mejores resultados con frecuencias más altas, incluso al comparar con valores de frecuencia con abundancia de elementos con band-gap en el conjunto de entrenamiento del modelo subrogado. Por otro lado, en cuanto a las diferentes funciones objetivo evaluadas se observa que, en general, los mejores resultados se obtienen para valores de α entre 0,25 y 0,5, presentando los menores tiempos de convergencia y número de iteraciones, así como excelentes factores de centro y anchos de band-gap. En general, se observa el comportamiento esperado entre α y el ancho de banda, lo que no ocurre en el caso de la precisión en la frecuencia. Esto probablemente se debe a la tendencia de la metodología a presentar una frecuencia predicha para el mejor enrejado constante a lo largo de su ejecución.

Al analizar los elementos en el conjunto de entrenamiento, se observa que esta no es una forma robusta de obtener enrejados con band-gap en torno a las frecuencias de diseño, y que los valores de las funciones objetivo alcanzadas por los enrejados de las metodologías son, en

la mayoría de los casos, comparables o superiores a los del conjunto de entrenamiento. Por lo anterior, se justifica el uso de redes neuronales por sobre el diseño enrejados con band-gap mediante la generación aleatoria de enrejados y su posterior evaluación con el modelo de alta fidelidad.

Resulta posible afirmar que, en el caso bidimensional, la metodología subrogada permite la convergencia entre los modelos subrogados y el modelo de alta fidelidad si se permite que su ejecución continúe por sobre la condición de detención. En el caso tridimensional, el comportamiento convergente no es totalmente claro, lo que se podría deber a una cantidad de iteraciones insuficiente para alcanzar la convergencia, a limitaciones de la arquitectura de la red neuronal utilizada, o a un comportamiento más complejo y difícil de predecir en el caso tridimensional. En ambos casos, la condición de detención propuesta permite la obtención de enrejados con band-gap en un número de iteraciones significativamente menor al requerido para alcanzar la convergencia, por lo que se evidencia su utilidad para fines de diseño.

En cuanto a la comparación entre los tiempos de ejecución de las metodologías desarrolladas y las metodologías analíticas, es posible afirmar que en ambos casos, para las frecuencias evaluadas, las metodologías de optimización subrogada presentan una tendencia a requerir tiempos de cómputo significativamente menores. En el caso bidimensional, esto se hace evidente al observar las combinaciones con convergencia consistente (como lo es 6000 [hz]), donde se logran tiempos hasta 10 veces menores que lo requerido por la metodología analítica. Aunque el resto de frecuencias en el caso bidimensional también presenta tiempos de ejecución menores que la alternativa analítica, los resultados no son completamente comparables ya que en estos casos no todas las ejecuciones de la metodología culminan en convergencia.

En el caso de la metodología tridimensional la tendencia a requerir menores tiempos de cómputo es aún más clara, ya que al culminar todas las ejecuciones en convergencia satisfactoria, se simplifica la comparación con respecto al caso analítico. Como se mencionó anteriormente, la metodología de optimización subrogada tridimensional presenta, según las estimaciones ya detalladas, tiempos entre 10^2 y 10^3 veces menores que lo requerido por la metodología analítica, lo que se traduce en ventajas evidentes en costo computacional y aplicabilidad.

Existe potencial valor en futuros estudios que consideren estrategias para obtener mejores modelos subrogados, tanto en precisión como en rapidez de ejecución y entrenamiento, ya que esto tendría un efecto significativo sobre el rendimiento de las metodologías. De igual forma, existe potencial valor en trabajos futuros que indaguen en el impacto que podrían tener diferentes estrategias de refinamiento de las redes neuronales en el rendimiento de la metodología, ya que este aspecto puede haber sido fundamental para compensar las deficiencias de los modelos subrogados. Existe también potencial valor en la posibilidad de explorar diferentes algoritmos de optimización global con variaciones en su incorporación a la meto-

Conclusión

dología subrogada, tales como distintas funciones objetivo o enfoques diferentes al incorporar elementos en una nueva iteración.

Es posible afirmar entonces que el uso de una metodología de optimización subrogada podría ser de suma utilidad en un contexto de fabricación reiterada de estructuras con band-gap en torno a una frecuencia de diseño, ya que en este caso se incurriría en el costo asociado a la elaboración del conjunto de entrenamiento una sola vez, lo que se compensaría a largo plazo con el significativo ahorro de tiempo y recursos asociado a la metodología subrogada al momento de diseñar cada estructura específica. Esto, por supuesto, implica múltiples aplicaciones en industrias como la aeronáutica, espacial, transporte, y generación de energía, entre otros.

7. Bibliografía

- [1] V. Gálvez, «Desarrollo de modelos en elementos finitos parametrizados de paneles tipo sándwich con núcleos ultralivianos para predecir band gaps» Universidad de Chile, 2021.
- [2] L. Quinteros, «OPTIMIZACIÓN DE BAND-GAP EN MATERIALES TIPO SÁNDWICH» Universidad de Chile, 2020.
- [3] M. Benedetti, A. du Plessis, R.O. Ritchie, M. Dallago, S.M.J. Razavi, F. Berto. 2021. Architected cellular materials: A review on their mechanical properties towards fatigue-tolerant design and fabrication [En línea] <<https://www.sciencedirect.com.uchile.idm.oclc.org/science/article/pii/S0927796X21000012>> [Consulta: 26 junio 2021]
- [4] O. Al-Ketan, A.H. Alami. Biomimetic Materials for Engineering Applications. [En línea] <<https://www.sciencedirect.com/science/article/pii/B978012815732900019X>> [Consulta: 26 Junio 2021]
- [5] Open-Cell vs. Closed-Cell Spray Foam. [En línea] <<https://www.jm.com/en/blog/2020/october/open-cell-vs--closed-cell-spray-foam/>> [Consulta: 20 Septiembre 2022].
- [6] How a Composite Sandwich Panel Works. [En línea] <http://www.sandwichpanels.org/articles/article_whatmakessandwich.html> [Consulta: 27 junio 2021].
- [7] Hongbin Yan, Xiaohu Yang, Tianjian Lu, Gongnan Xie. 2017. Convective heat transfer in a lightweight multifunctional sandwich panel with X-type metallic lattice core. [En línea] <<https://www.sciencedirect.com/science/article/abs/pii/S135943116338649>> [Consulta: 20 Septiembre 2022].
- [8] H. Yazdani Sarvestani, A. H. Akbarzadeh, H. Niknam, K. Hermenean. 2018. Open-Cell vs. Closed-Cell Spray Foam. [En línea] <<https://www.sciencedirect.com/science/article/abs/pii/S0263822317342423>> [Consulta: 20 Septiembre 2022].
- [9] B.J. Van Zeghbroeck. 1997. Bravais lattices. [En línea] <<https://ecee.colorado.edu/~bart/book/bravais.html>> [Consulta: 27 junio 2021].
- [10] I. Shvets. 2014. Introduction to Solid State Physics. [En línea] <https://www.tcd.ie/Physics/study/current/undergraduate/lecture-notes/py3p03/Lecture4_2014.pdf> [Consulta: 27 junio 2021].
- [11] P.K. Misra. 2012. Brillouin Zones. [En línea] <<https://www.sciencedirect.com/topics/physics-and-astronomy/brillouin-zones>> [Consulta: 28 junio 2021].

- [12] A. Dal Corso. Points inside the Brillouin zone. [En línea] <http://web.mit.edu/espresso_v6.1/amd64_ubuntu1404/qe-6.1/Doc/brillouin_zones.pdf> [Consulta: 28 junio 2021].
- [13] C.Guillén-Gallegos, F.Villa-Villa. Marzo de 2019. Phononic band structure of an acoustic waveguide that behaves as a phononic crystal. [En línea] <<https://www.sciencedirect.com/science/article/pii/S2211379718323829>> [Consulta: 28 junio 2021].
- [14] Maldovan's Research Group. 2016. Periodic Materials And Interference Lithography For Photonics, Phononics, And Mechanics. [En línea] <<https://maldovan.gatech.edu/past-research>> [Consulta: 20 Septiembre 2022].
- [15] H. Singh 2021. Deep Learning 101: Beginners Guide to Neural Network [En línea] <<https://www.analyticsvidhya.com/blog/2021/03/basics-of-neural-network/>> [Consulta: 27 junio 2022]
- [16] Artificial Neural Networks. [En línea] <<https://www.slideshare.net/GermnRamosGarca/artificial-neural-networks-71956998>> [Consulta: 20 Septiembre 2022].
- [17] 15 de marzo de 2018. Parallel and distributed genetic algorithms. [En línea] <<https://towardsdatascience.com/parallel-and-distributed-genetic-algorithms-1ed2e76866e3#:~:text=Parallel%20genetic%20algorithm%20is%20such,a%20single%20task%20>> [Consulta: 29 junio 2021].
- [18] Z. Han, K Zhang. Marzo de 2012. Surrogate-Based Optimization. [En línea] <https://www.researchgate.net/profile/Zhong-Hua-Han/publication/221927230_Surrogate-Based_Optimization/links/57284dde08ae586b21e2a232/Surrogate-Based-Optimization.pdf> [Consulta: 2 julio 2021].

ANEXOS

ANEXO A: Gráficos

Metodología tridimensional

A continuación se muestran los gráficos extraídos al evaluar la metodología tridimensional. Los gráficos muestran los diagramas de bandas obtenidos en caso de convergencia, evolución de la frecuencia, y evolución de la función objetivo (ec. 3.5).

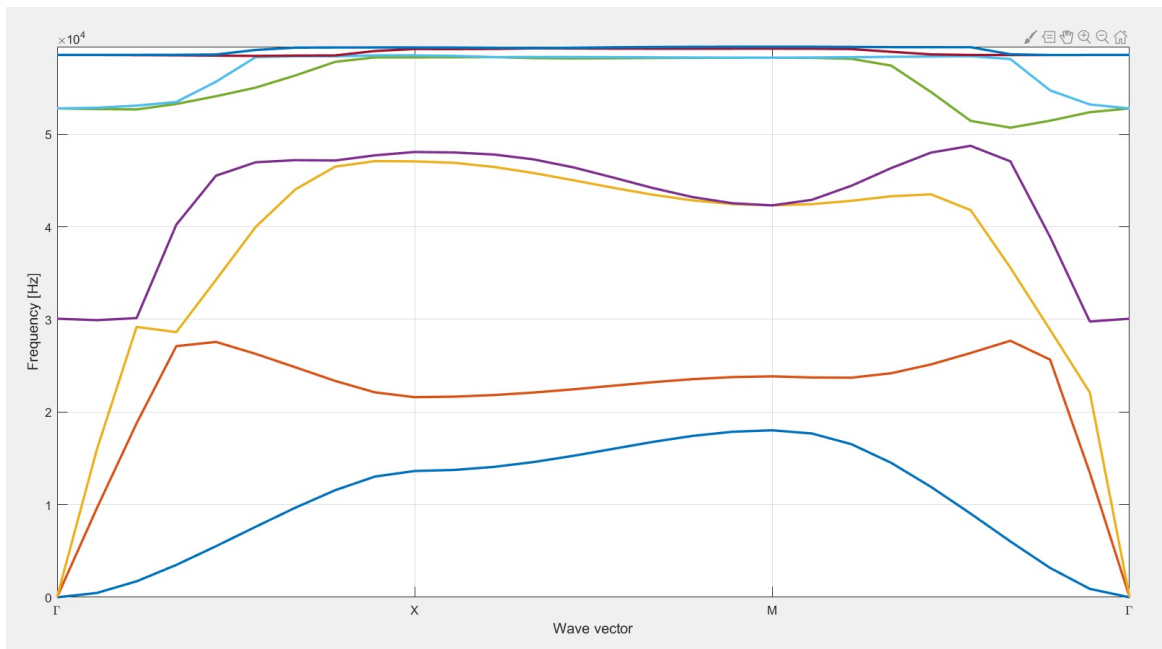


Figura 7.34: Ejemplo de diagrama de bandas con band-gap obtenido para $5 \cdot 10^4$ [Hz] y $\alpha = 0,5$ (pirámide central).

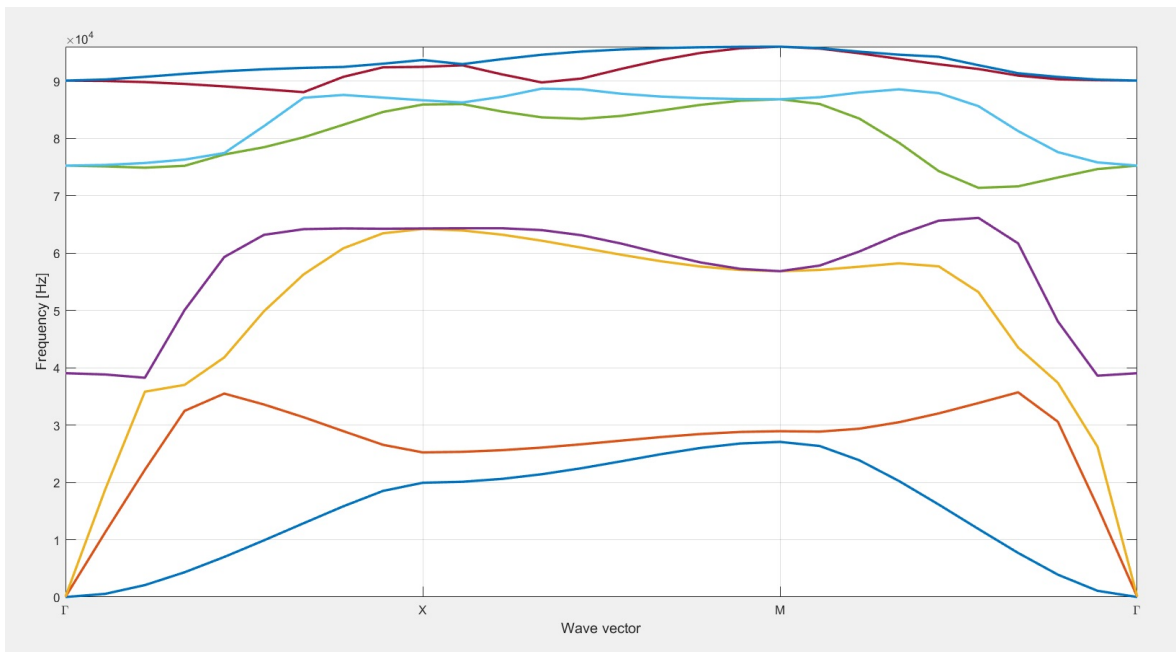


Figura 7.35: Ejemplo de diagrama de bandas con band-gap obtenido para $7 \cdot 10^4$ [Hz] y $\alpha = 0,5$ (pirámide central).

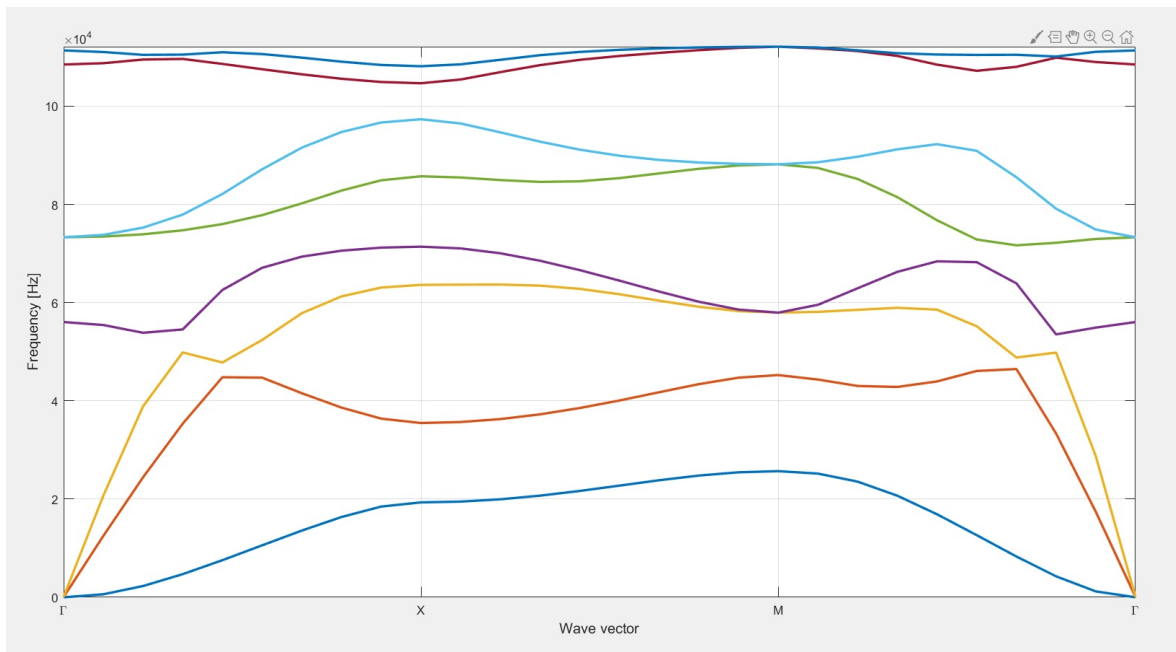


Figura 7.36: Ejemplo de diagrama de bandas con band-gap obtenido para 10^5 [Hz] y $\alpha = 0,5$ (pirámide central).

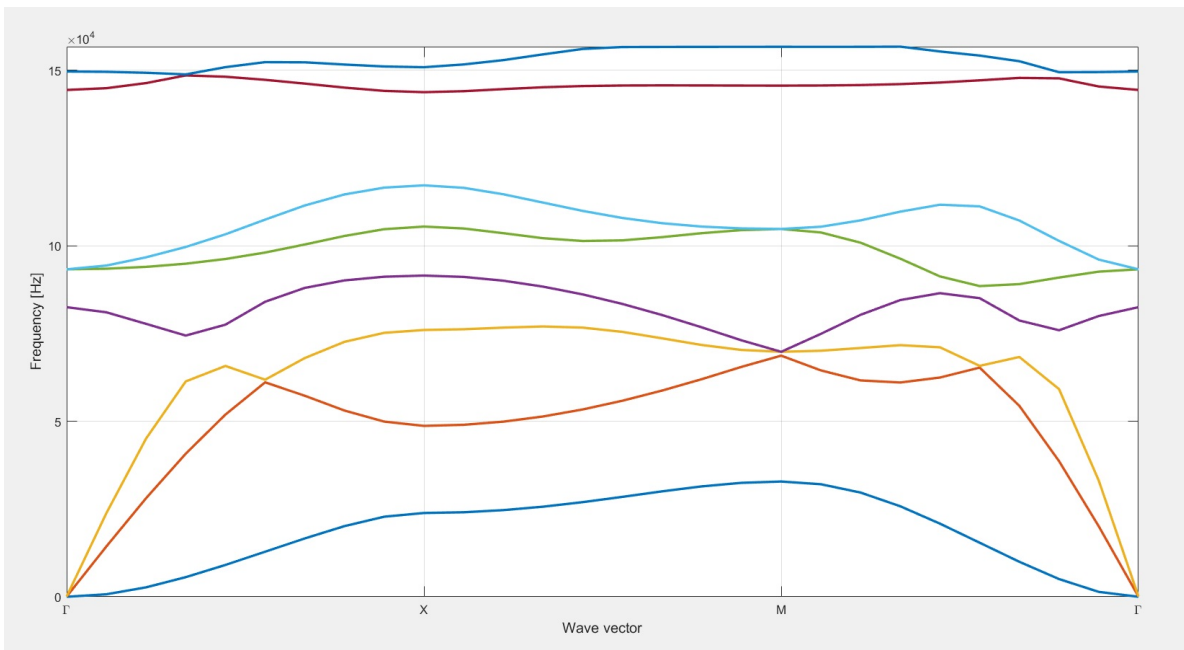


Figura 7.37: Ejemplo de diagrama de bandas con band-gap obtenido para $1,3 \cdot 10^5$ [Hz] y $\alpha = 0,5$ (pirámide central).

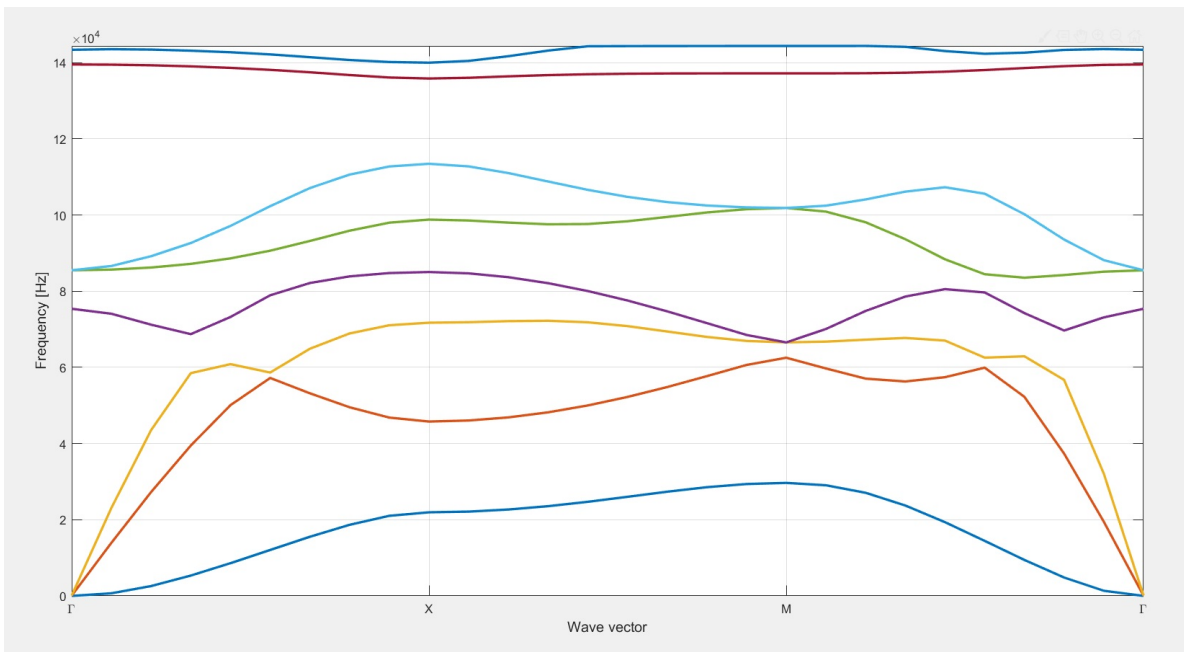


Figura 7.38: Ejemplo de diagrama de bandas con band-gap obtenido para $1,3 \cdot 10^5$ [Hz] y $\alpha = 0,25$ (pirámide central).

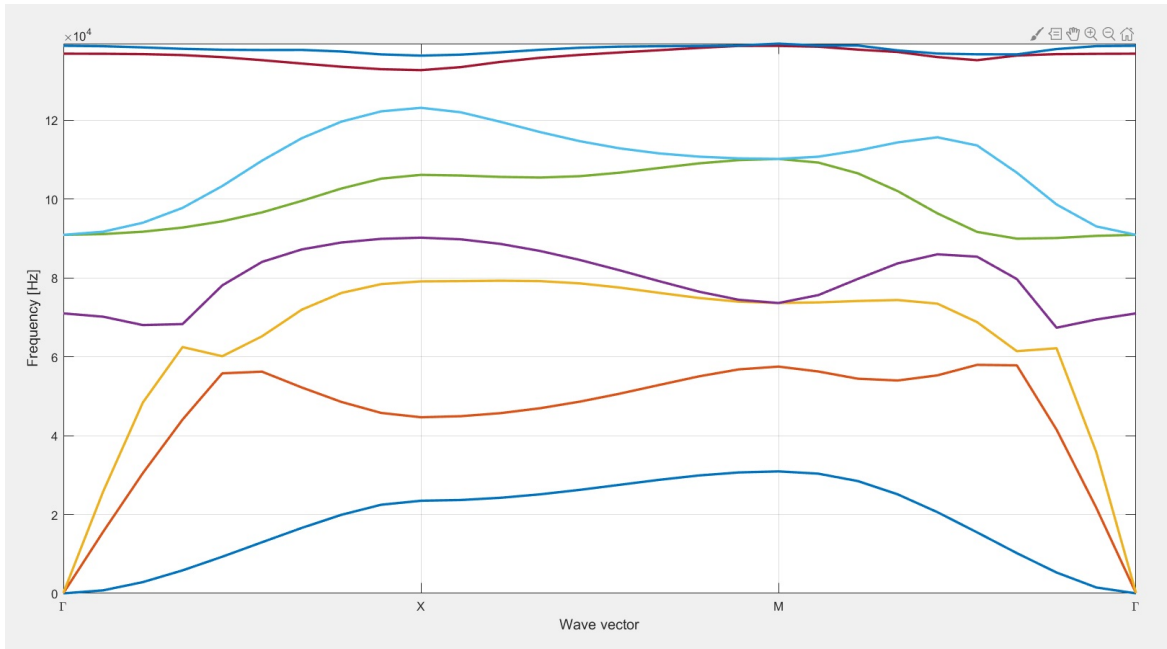


Figura 7.39: Ejemplo de diagrama de bandas con band-gap obtenido para $1,3 \cdot 10^5$ [Hz] y $\alpha = 0,75$ (pirámide central).

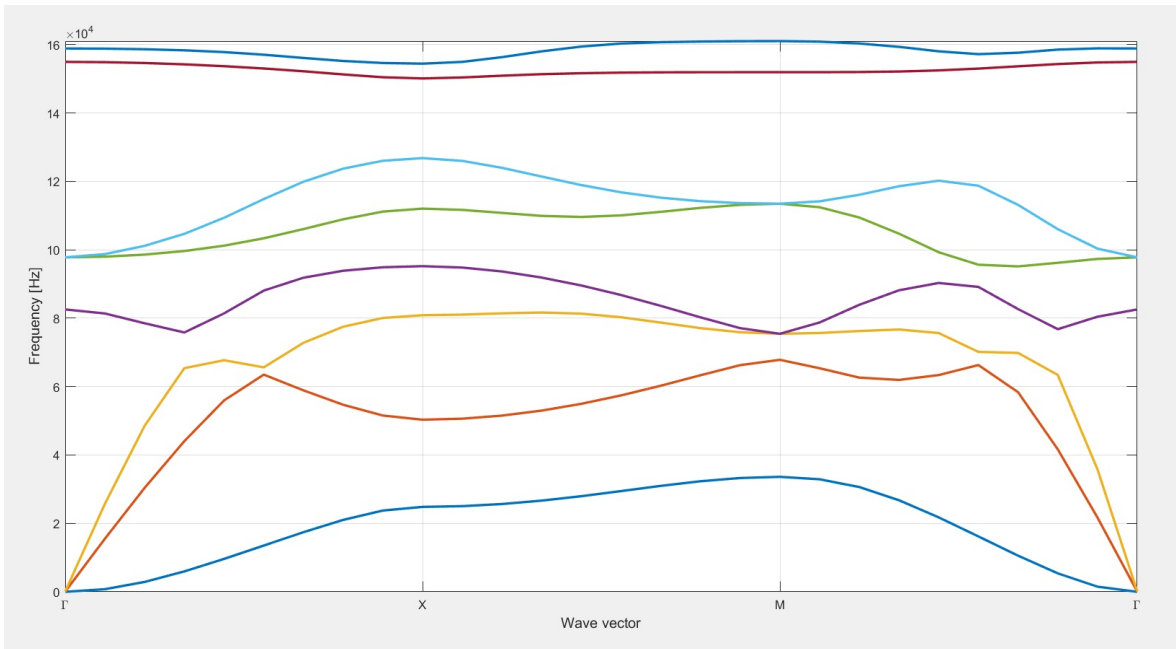


Figura 7.40: Ejemplo de diagrama de bandas con band-gap obtenido para $1,5 \cdot 10^5$ [Hz] y $\alpha = 0,5$ (pirámide central).

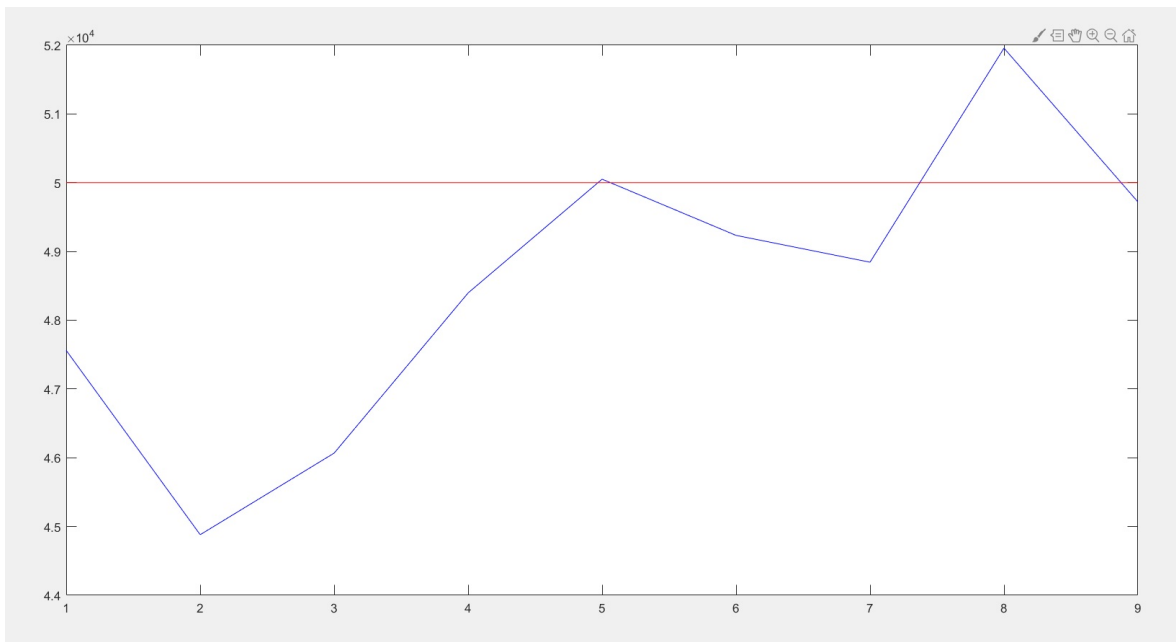


Figura 7.41: Ejemplo de evolución en iteraciones de la frecuencia real del óptimo (azul) y de la predicha por la red (rojo) para $5 \cdot 10^4$ [Hz] y $\alpha = 0,5$ (pirámide central).

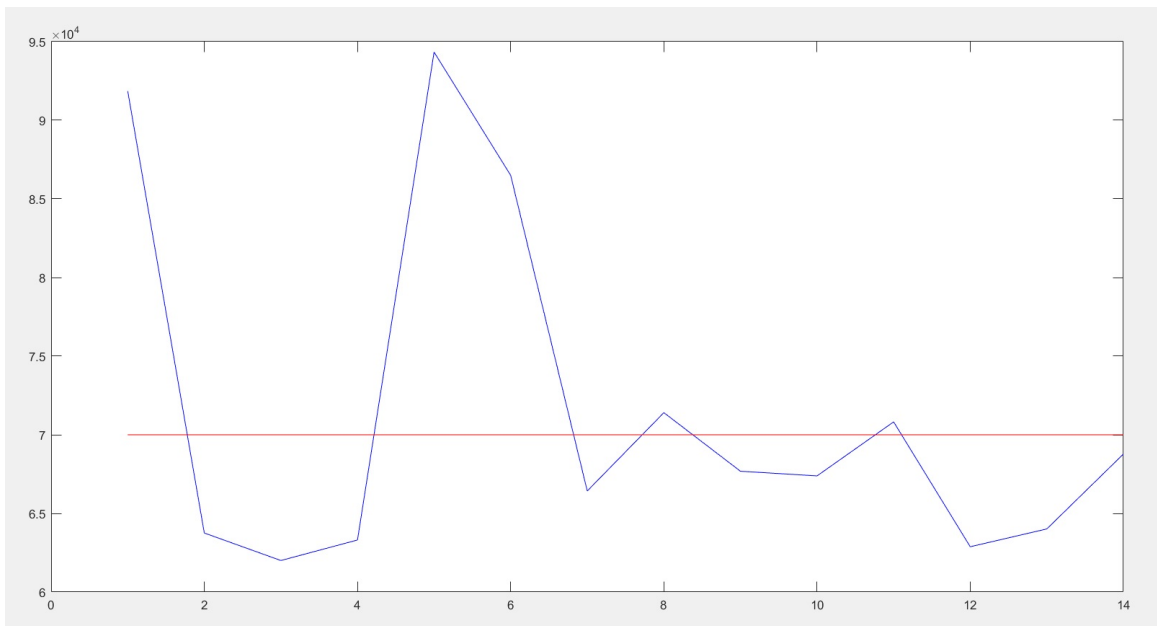


Figura 7.42: Ejemplo de evolución en iteraciones de la frecuencia real del óptimo (azul) y de la predicha por la red (rojo) para $7 \cdot 10^4$ [Hz] y $\alpha = 0,5$ (pirámide central).

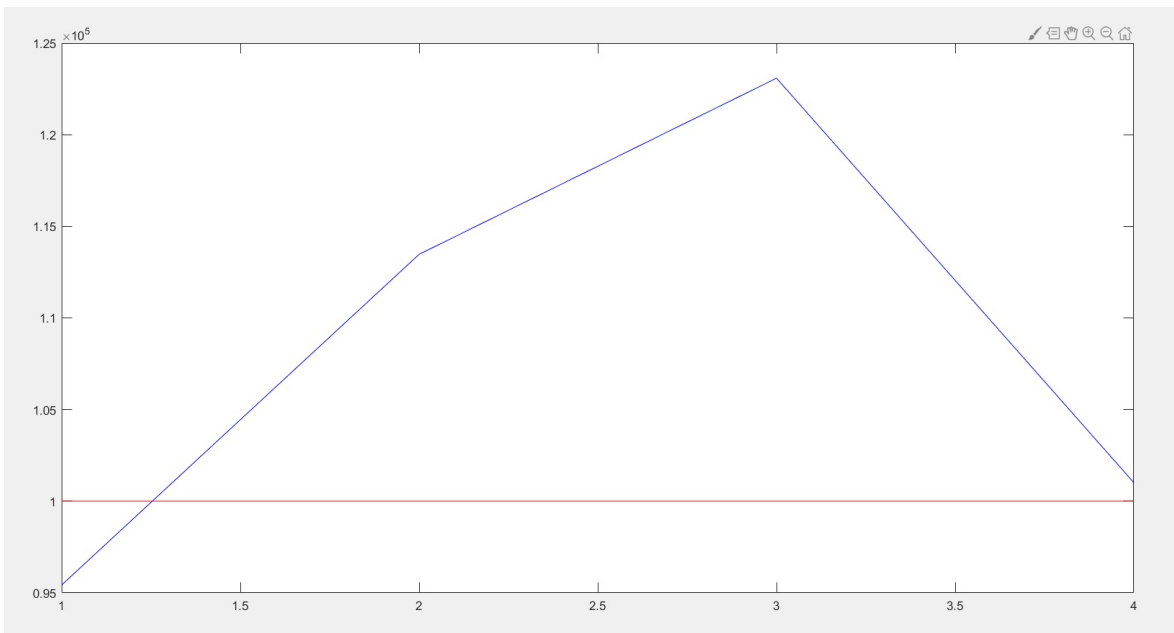


Figura 7.43: Ejemplo de evolución en iteraciones de la frecuencia real del óptimo (azul) y de la predicha por la red (rojo) para 10^5 [Hz] y $\alpha = 0,5$ (pirámide central).

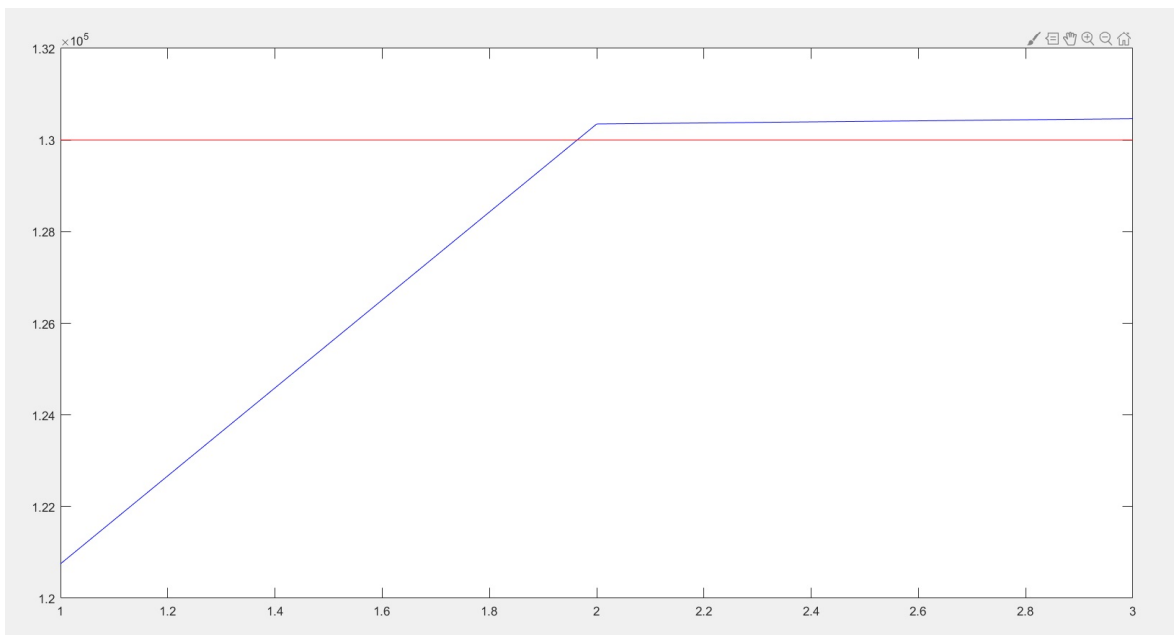


Figura 7.44: Ejemplo de evolución en iteraciones de la frecuencia real del óptimo (azul) y de la predicha por la red (rojo) para $1,3 \cdot 10^5$ [Hz] y $\alpha = 0,5$ (pirámide central).

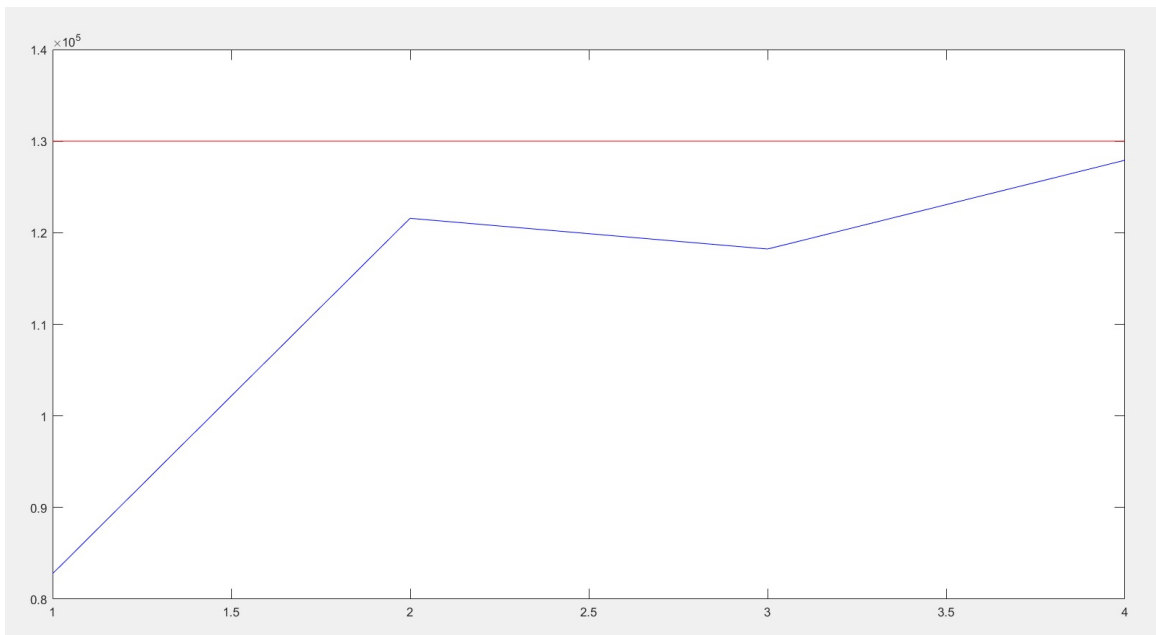


Figura 7.45: Ejemplo de evolución en iteraciones de la frecuencia real del óptimo (azul) y de la predicha por la red (rojo) para $1,3 \cdot 10^5$ [Hz] y $\alpha = 0,75$ (pirámide central).

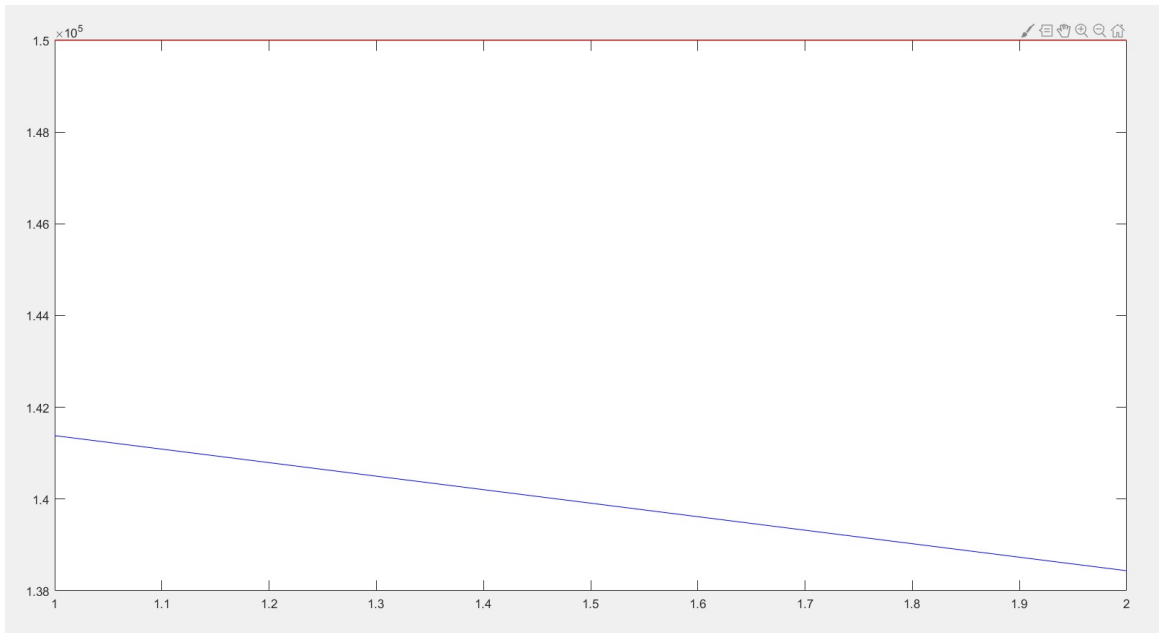


Figura 7.46: Ejemplo de evolución en iteraciones de la frecuencia real del óptimo (azul) y de la predicha por la red (rojo) para $1,5 \cdot 10^5$ [Hz] y $\alpha = 0,5$ (pirámide central).

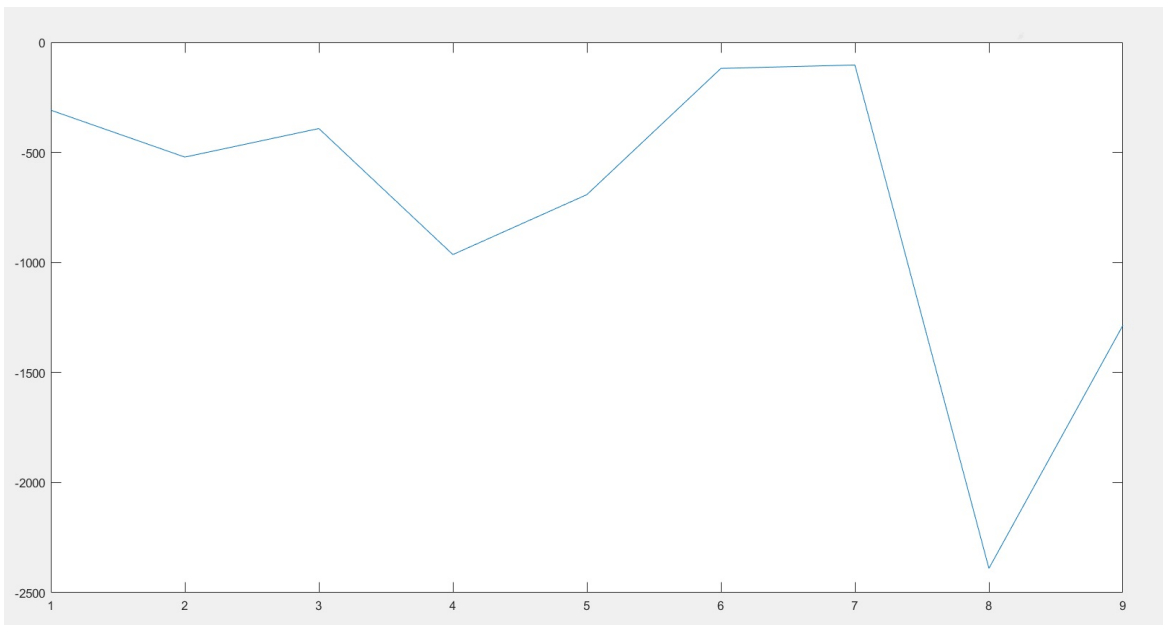


Figura 7.47: Ejemplo de la evolución del valor alcanzado de la función objetivo para $5 \cdot 10^4$ [Hz] y $\alpha = 0,5$ (pirámide central).

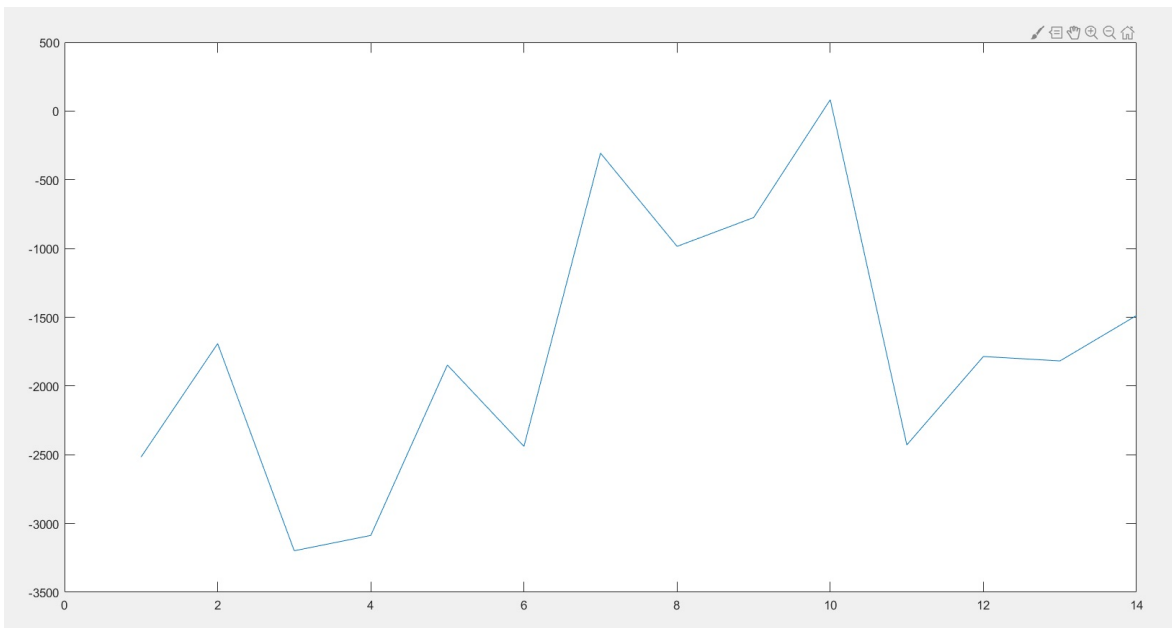


Figura 7.48: Ejemplo de la evolución del valor alcanzado de la función objetivo para $7 \cdot 10^4$ [Hz] y $\alpha = 0,5$ (pirámide central).

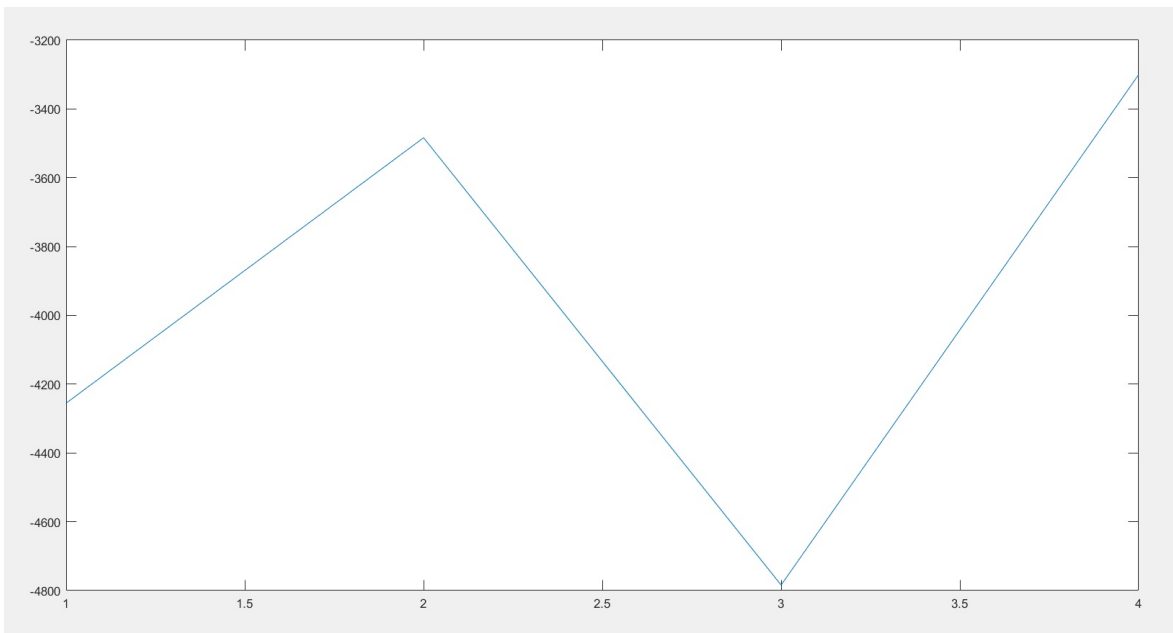


Figura 7.49: Ejemplo de la evolución del valor alcanzado de la función objetivo para 10^5 [Hz] y $\alpha = 0,5$ (pirámide central).

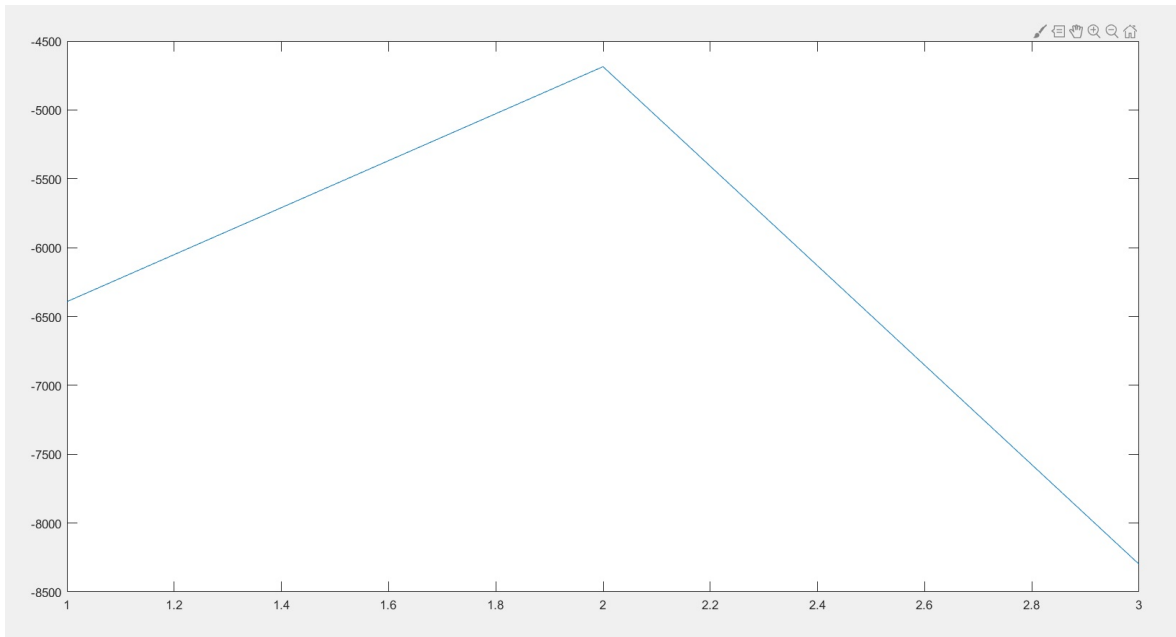


Figura 7.50: Ejemplo de la evolución del valor alcanzado de la función objetivo para $1,3 \cdot 10^5$ [Hz] y $\alpha = 0,5$ (pirámide central).

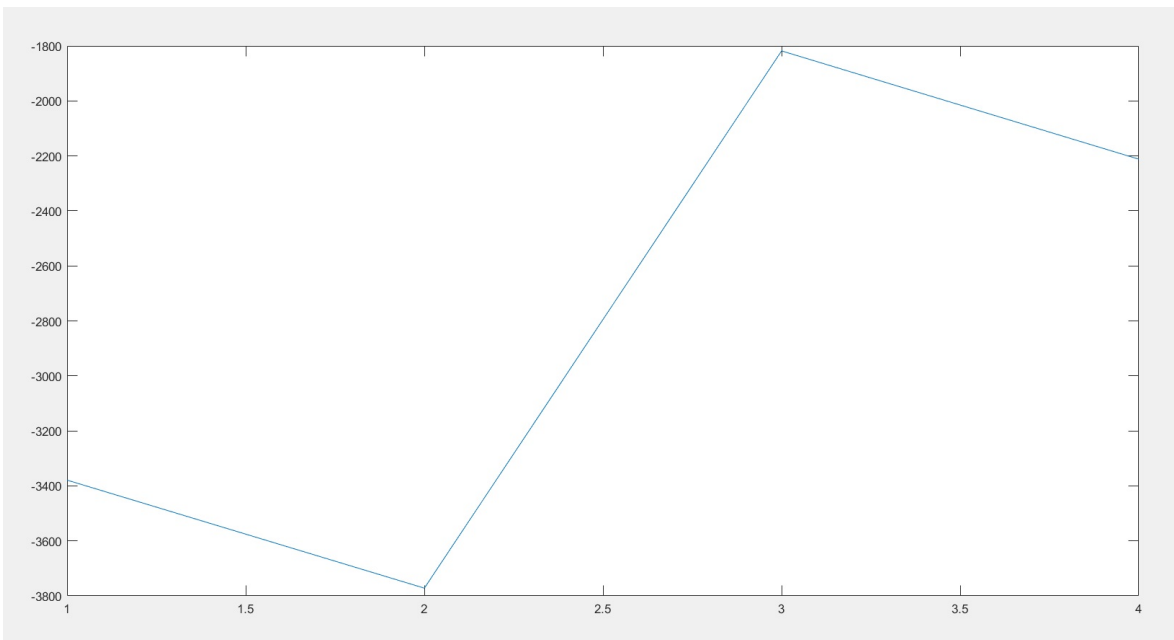


Figura 7.51: Ejemplo de la evolución del valor alcanzado de la función objetivo para $1,3 \cdot 10^5$ [Hz] y $\alpha = 0,75$ (pirámide central).

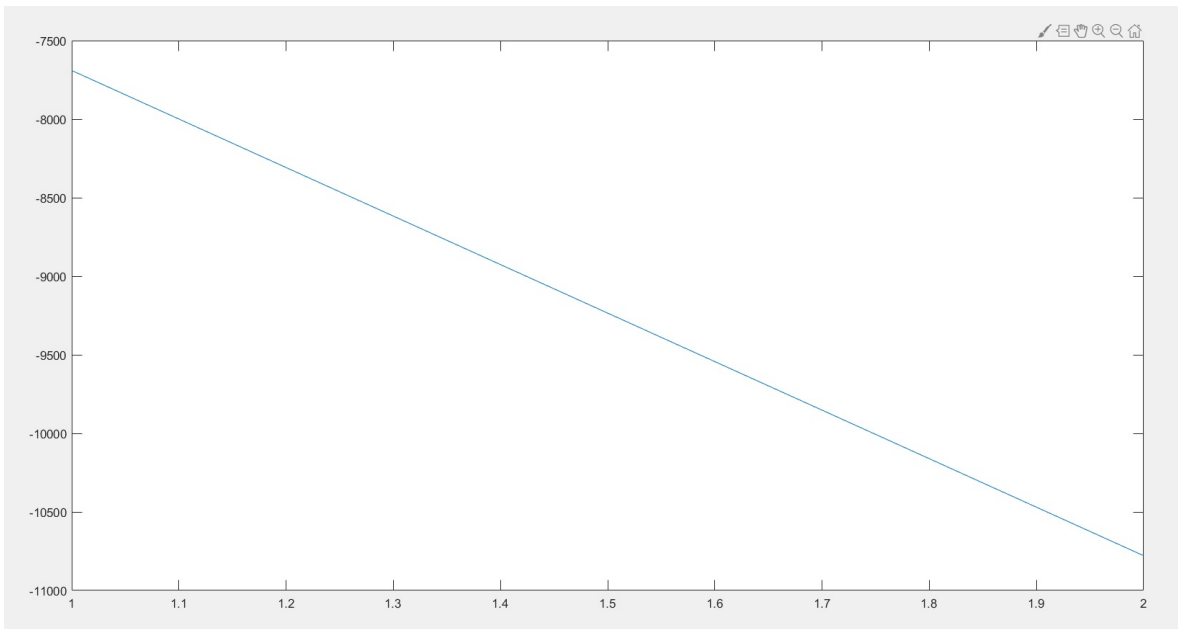


Figura 7.52: Ejemplo de la evolución del valor alcanzado de la función objetivo para $1,5 \cdot 10^5$ [Hz] y $\alpha = 0,5$ (pirámide central).

Metodología bidimensional: Algoritmo genético

A continuación se muestran los gráficos extraídos al evaluar la metodología bidimensional con algoritmo genético. Los gráficos muestran los diagramas de bandas obtenidos en caso de convergencia, evolución de la frecuencia, y evolución de la función objetivo (ec. 3.5).

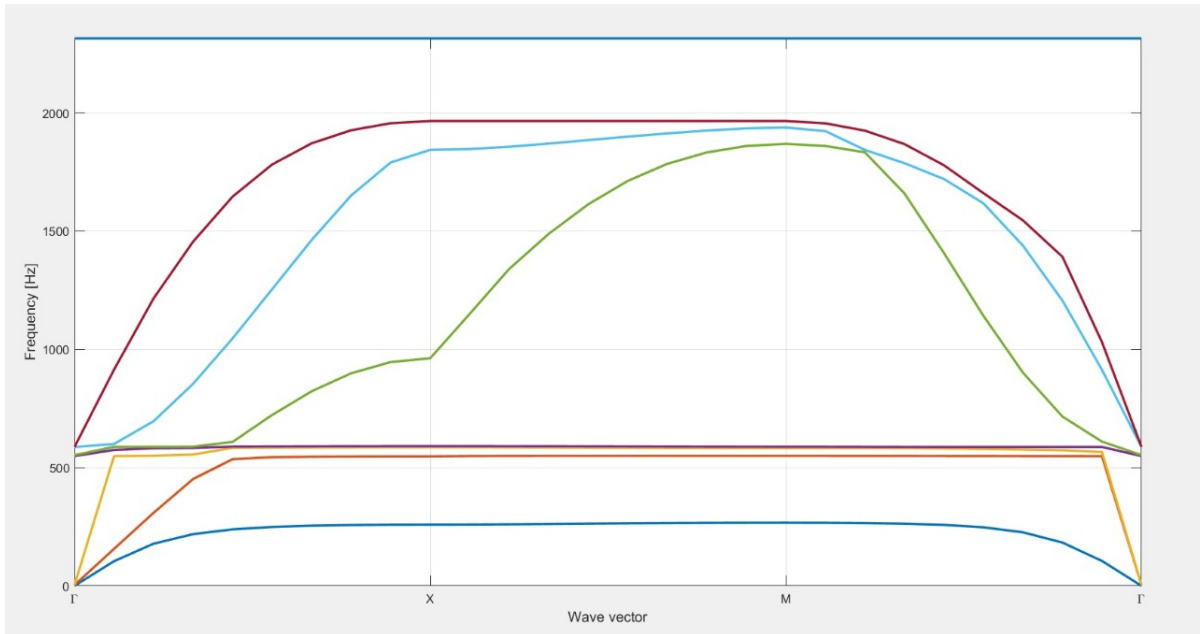


Figura 7.53: Ejemplo de diagrama de bandas con band-gap obtenido para 2000 [Hz] y $\alpha = 0,5$.

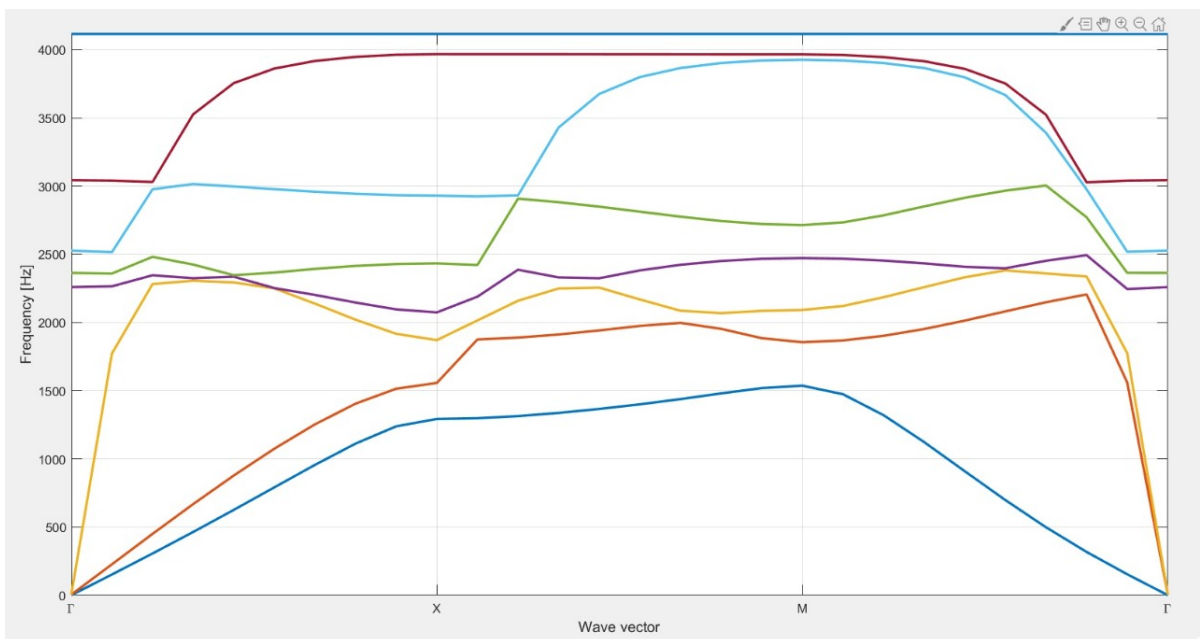


Figura 7.54: Ejemplo de diagrama de bandas con band-gap obtenido para 4000 [Hz] y $\alpha = 0,5$.

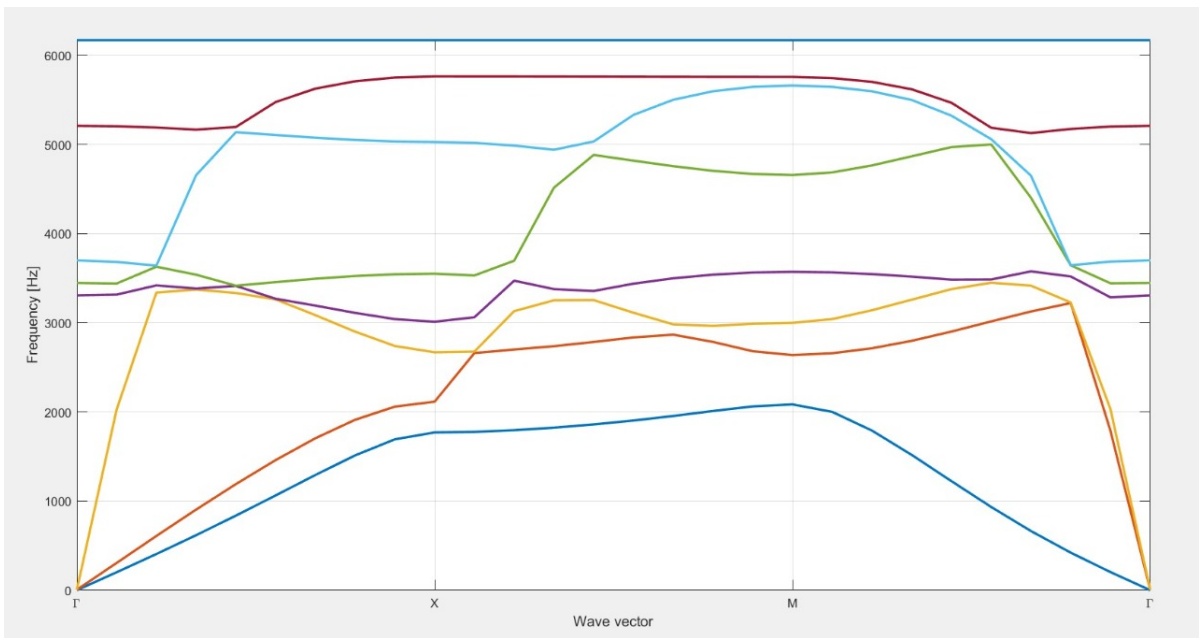


Figura 7.55: Ejemplo de diagrama de bandas con band-gap obtenido para 6000 [Hz] y $\alpha = 0,5$.

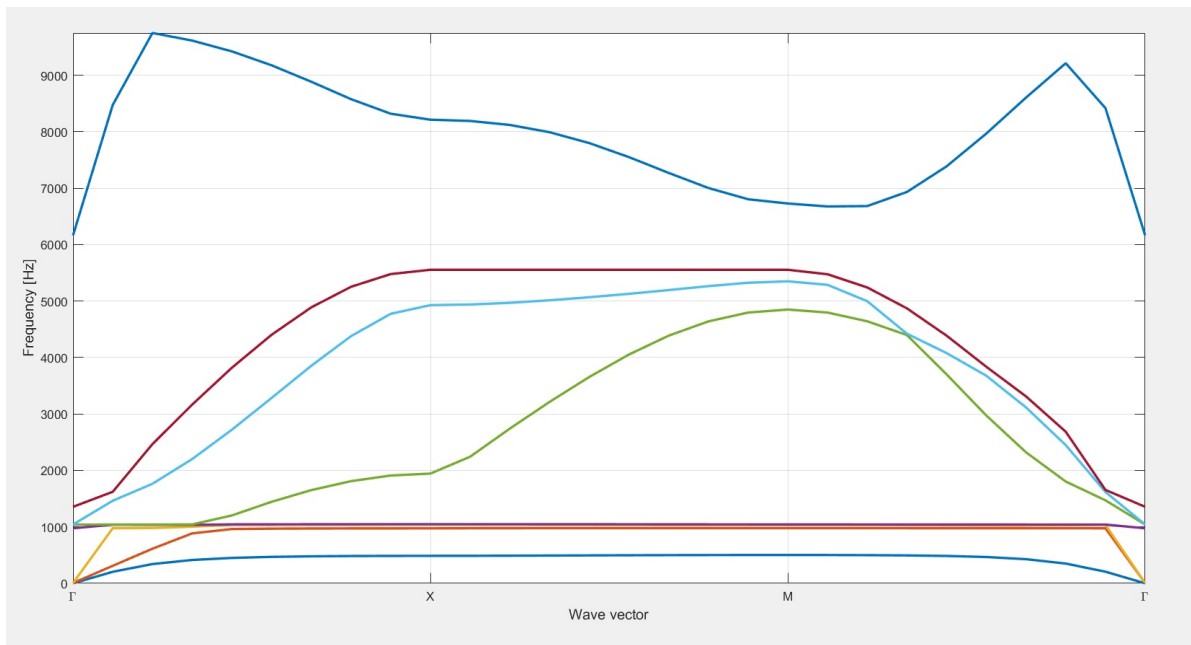


Figura 7.56: Ejemplo de diagrama de bandas con band-gap obtenido para 6000 [Hz] y $\alpha = 0,25$.

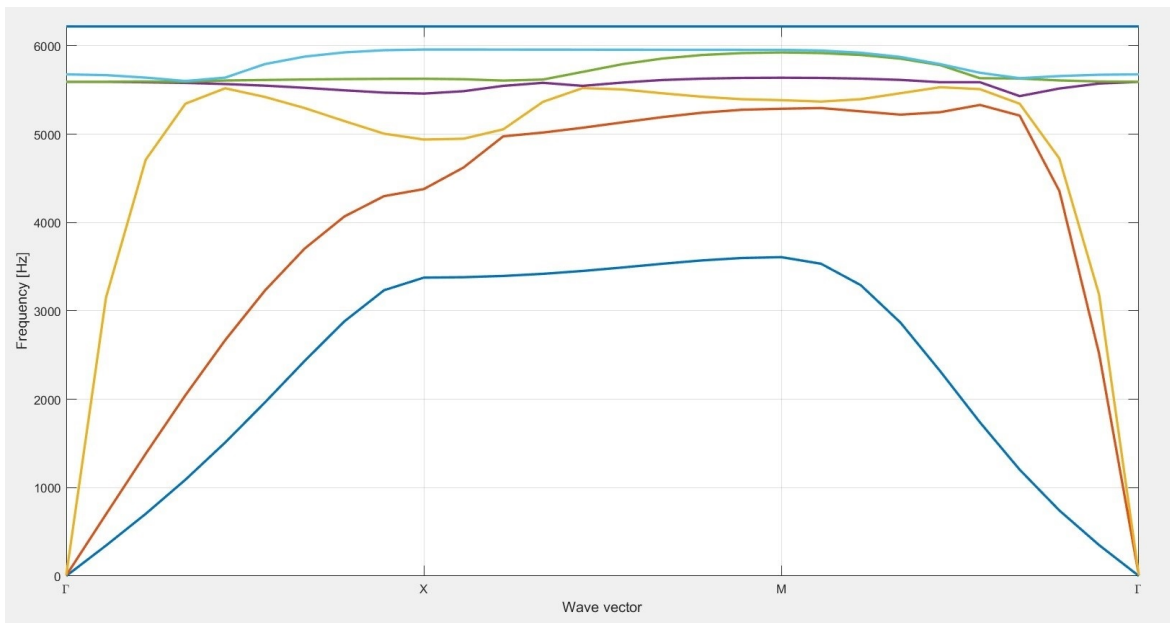


Figura 7.57: Ejemplo de diagrama de bandas con band-gap obtenido para 6000 [Hz] y $\alpha = 0,75$.

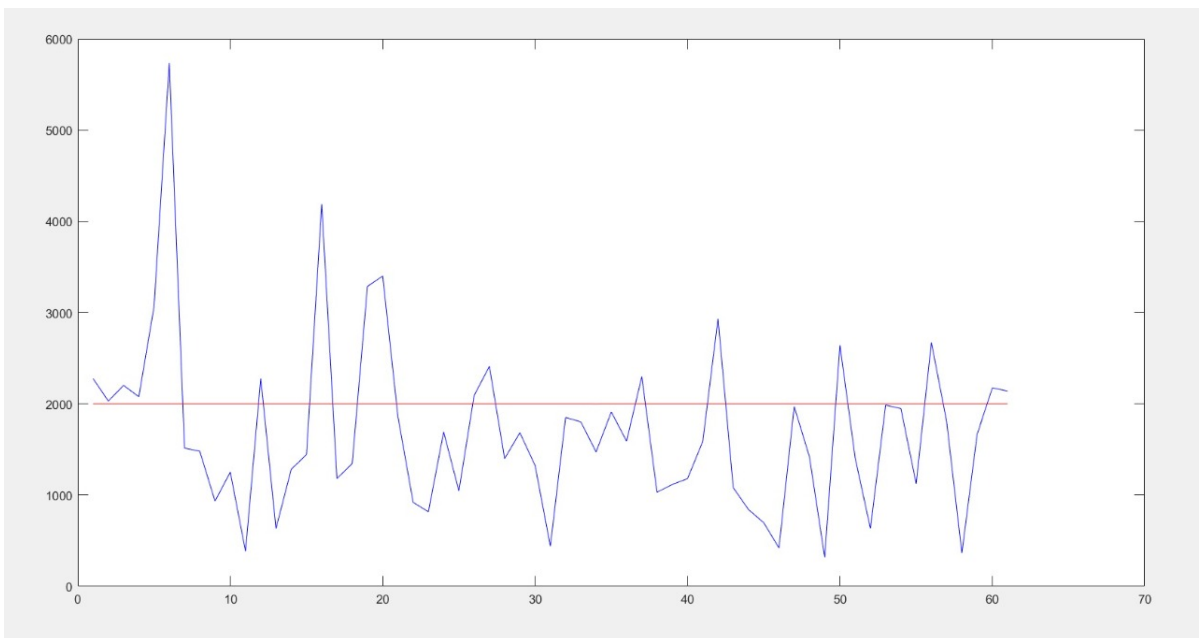


Figura 7.58: Ejemplo de evolución en iteraciones de la frecuencia real del óptimo (azul) y de la predicha por la red (rojo) para 2000 [Hz] y $\alpha = 0,5$.

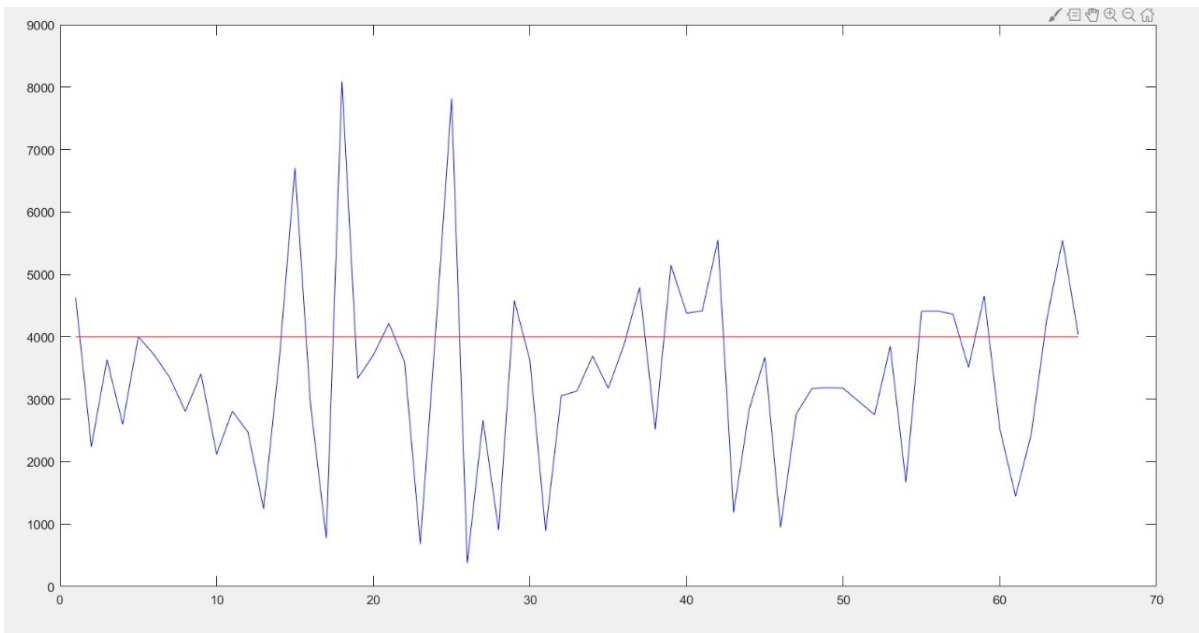


Figura 7.59: Ejemplo de evolución en iteraciones de la frecuencia real del óptimo (azul) y de la predicha por la red (rojo) para 4000 [Hz] y $\alpha = 0,5$.

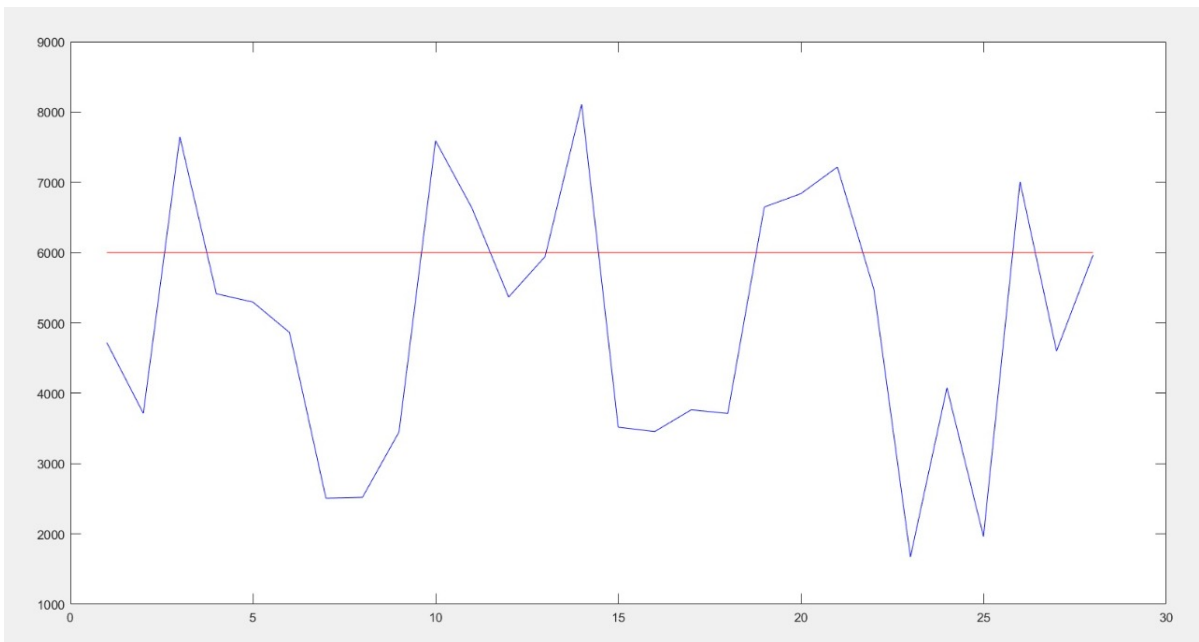


Figura 7.60: Ejemplo de evolución en iteraciones de la frecuencia real del óptimo (azul) y de la predicha por la red (rojo) para 6000 [Hz] y $\alpha = 0,5$.

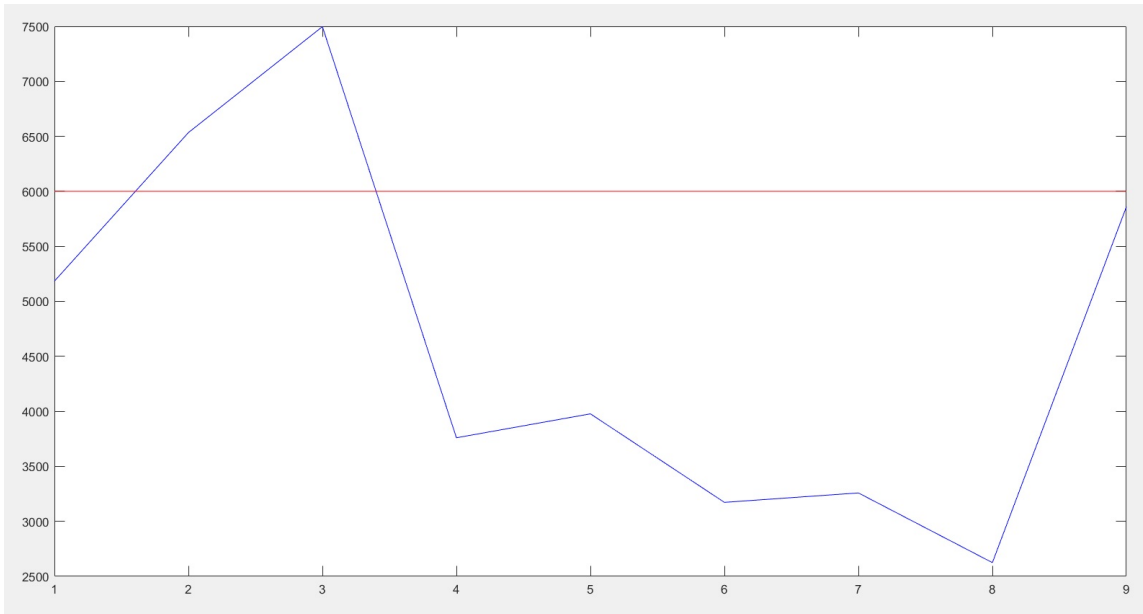


Figura 7.61: Ejemplo de evolución en iteraciones de la frecuencia real del óptimo (azul) y de la predicha por la red (rojo) para 6000 [Hz] y $\alpha = 0,25$.

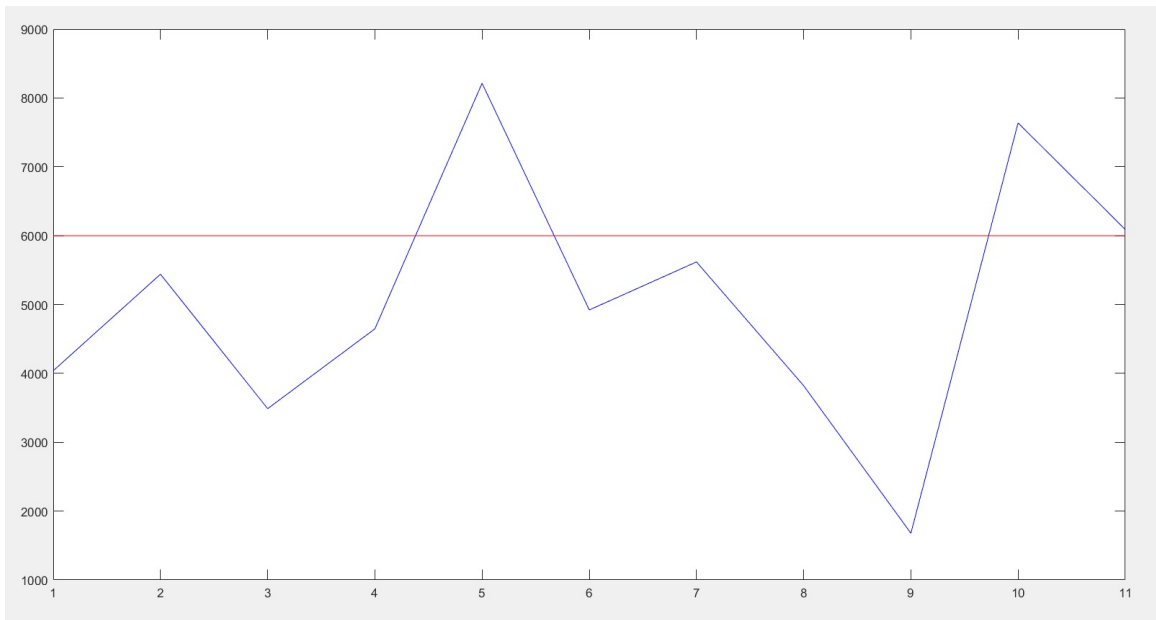


Figura 7.62: Ejemplo de evolución en iteraciones de la frecuencia real del óptimo (azul) y de la predicha por la red (rojo) para 6000 [Hz] y $\alpha = 0,75$.

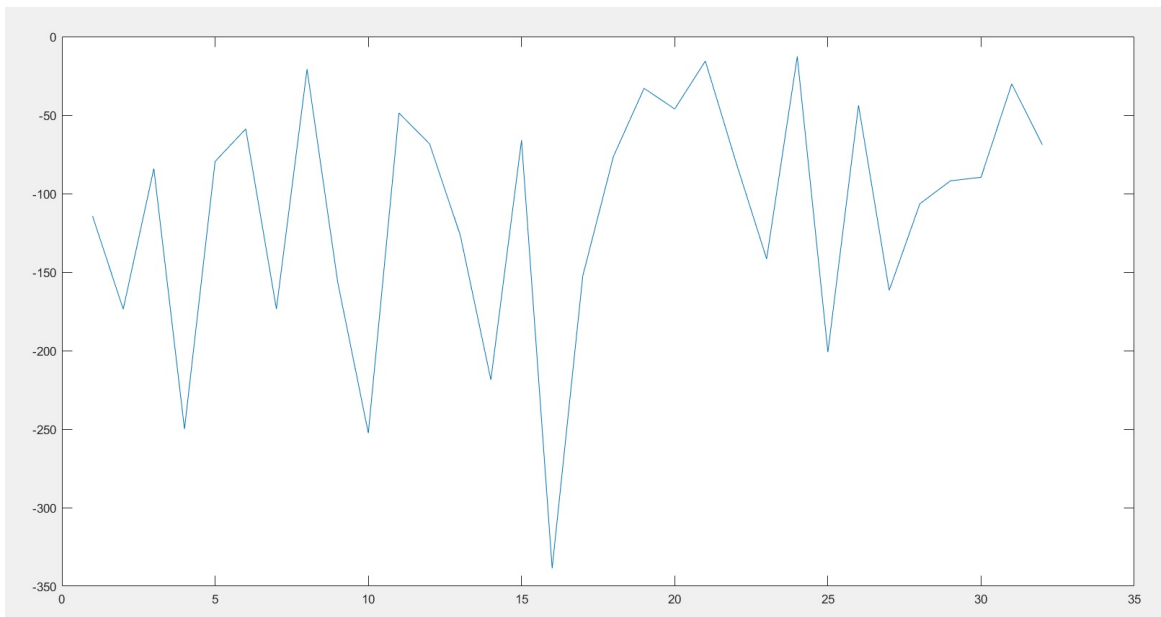


Figura 7.63: Ejemplo de la evolución del valor alcanzado de la función objetivo para 2000 [Hz] y $\alpha = 0,5$.

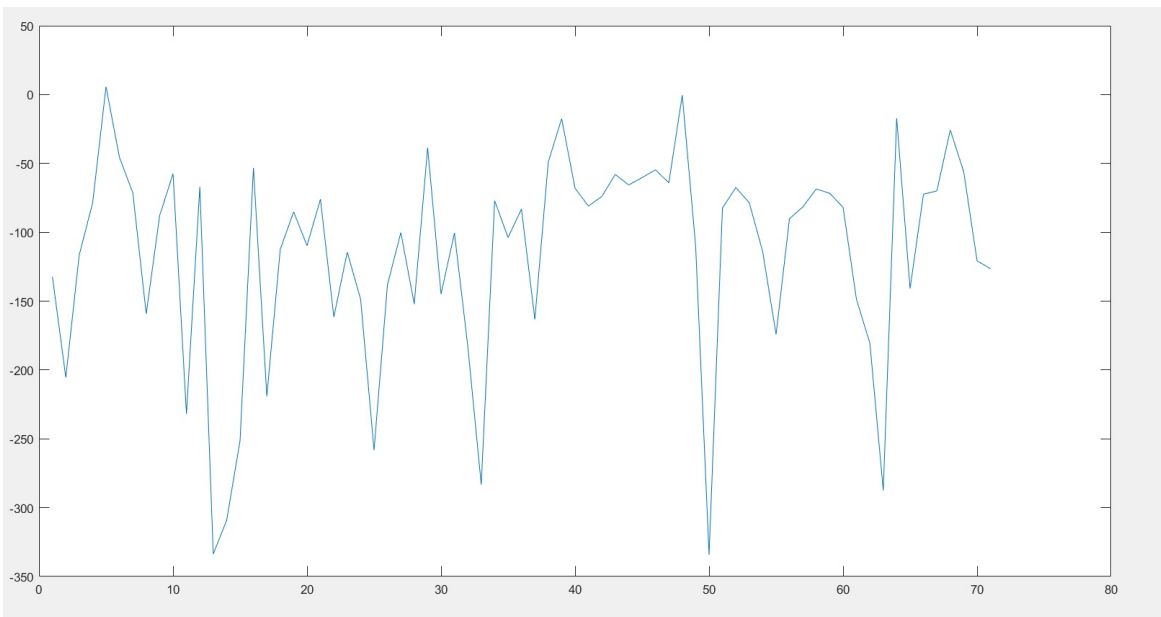


Figura 7.64: Ejemplo de la evolución del valor alcanzado de la función objetivo para 4000 [Hz] y $\alpha = 0,5$.

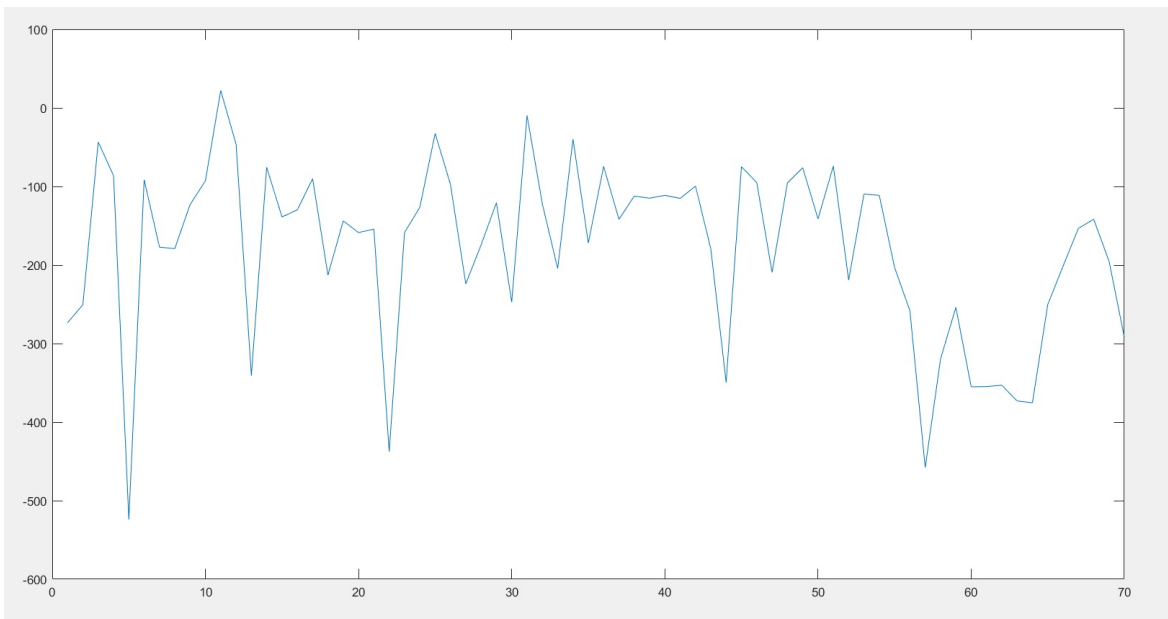


Figura 7.65: Ejemplo de la evolución del valor alcanzado de la función objetivo para 6000 [Hz] y $\alpha = 0,5$.

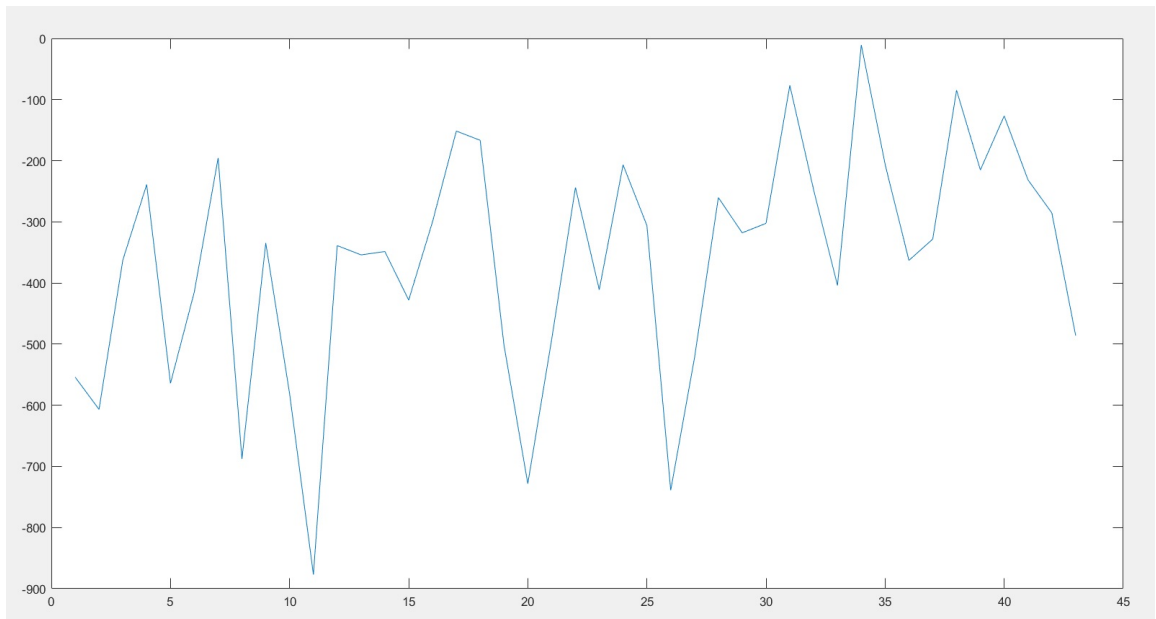


Figura 7.66: Ejemplo de la evolución del valor alcanzado de la función objetivo para 6000 [Hz] y $\alpha = 0,25$.

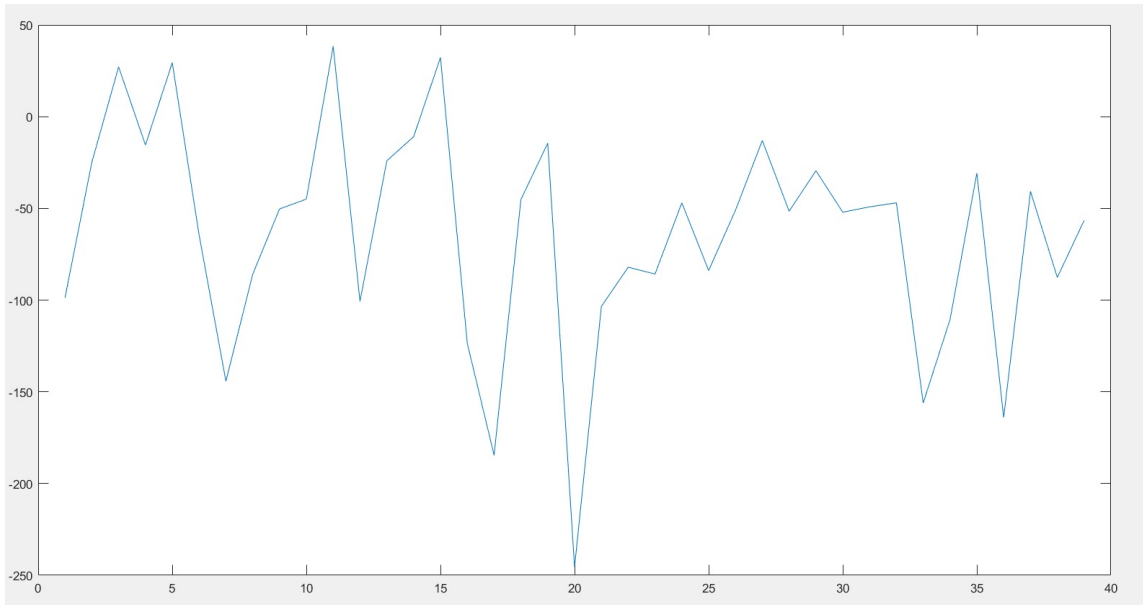


Figura 7.67: Ejemplo de la evolución del valor alcanzado de la función objetivo para 6000 [Hz] y $\alpha = 0,75$.

Metodología bidimensional: Enjambre de partículas

A continuación se muestran los gráficos extraídos al evaluar la metodología bidimensional con enjambre de partículas. Los gráficos muestran los diagramas de bandas obtenidos en caso de convergencia, evolución de la frecuencia, y evolución de la función objetivo (ec. 3.5).

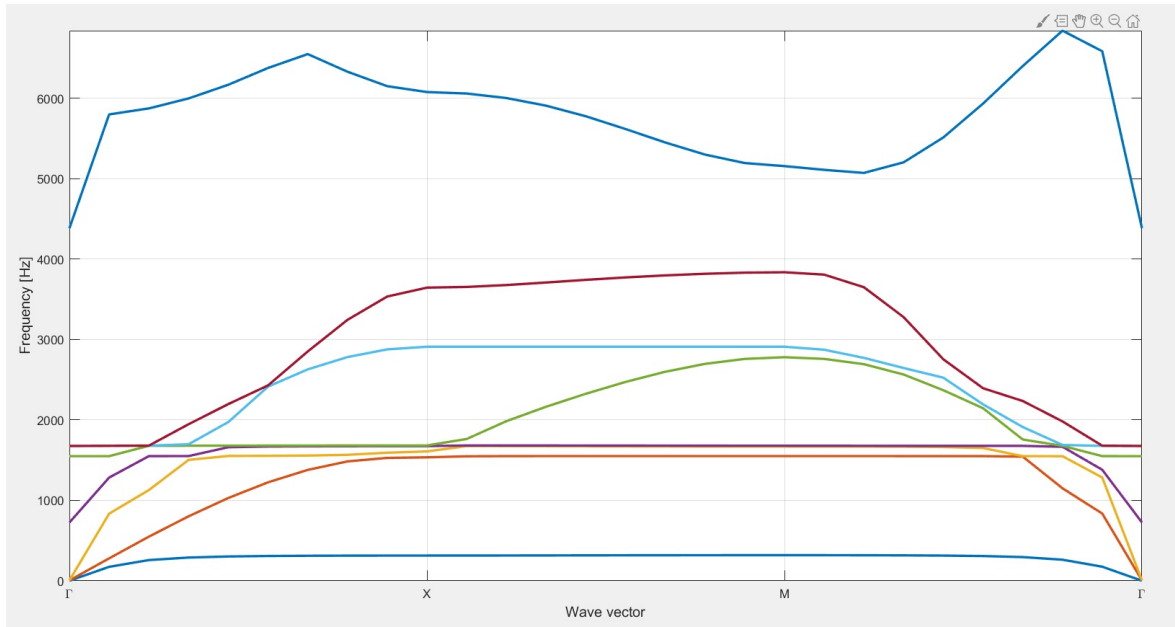


Figura 7.68: Ejemplo de diagrama de bandas con band-gap obtenido para 4000 [Hz] y $\alpha = 0,5$ (Enjambre de partículas).

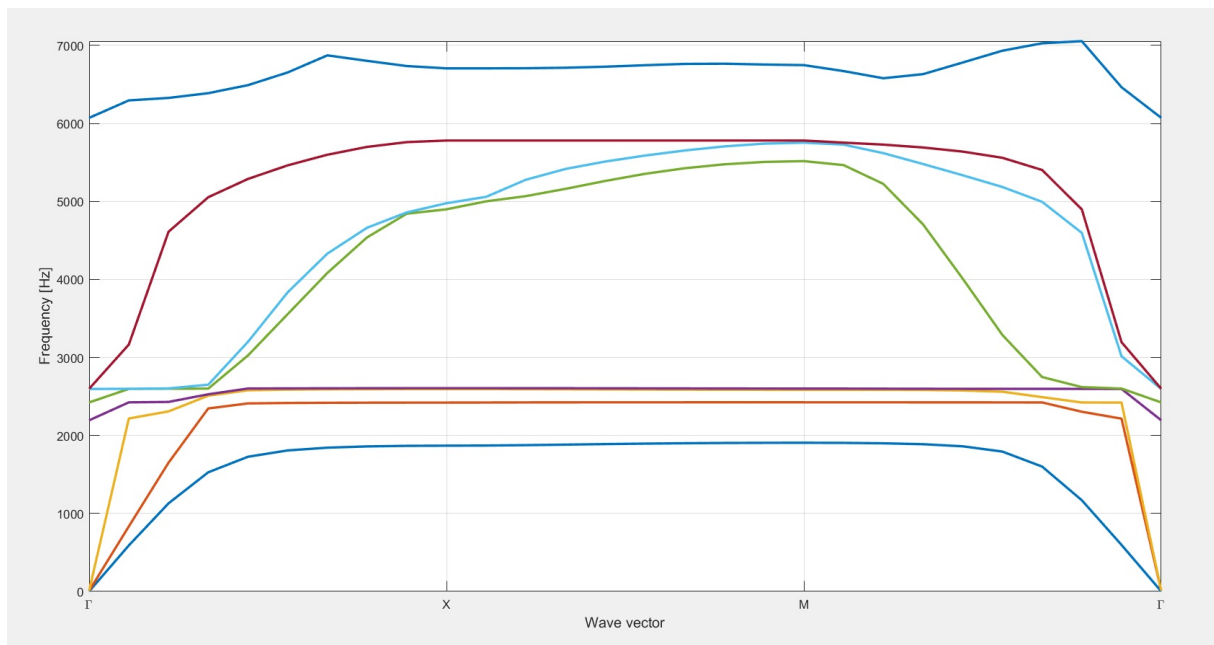


Figura 7.69: Ejemplo de diagrama de bandas con band-gap obtenido para 6000 [Hz] y $\alpha = 0,5$ (Enjambre de partículas).

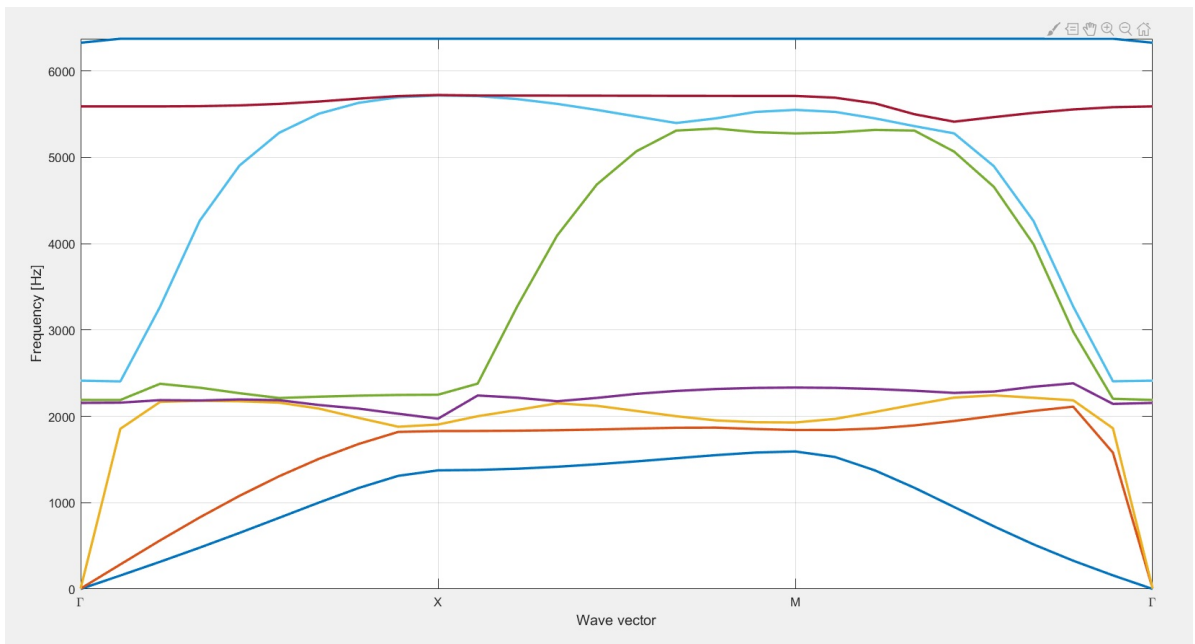


Figura 7.70: Ejemplo de diagrama de bandas con band-gap obtenido para 6000 [Hz] y $\alpha = 0,25$ (Enjambre de partículas).

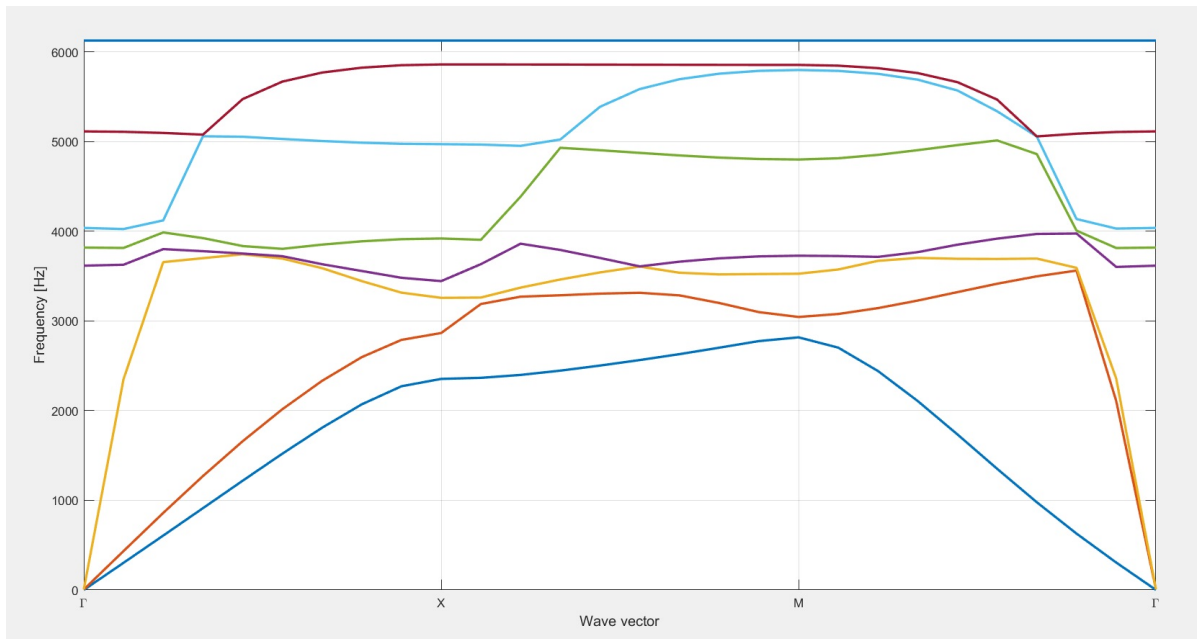


Figura 7.71: Ejemplo de diagrama de bandas con band-gap obtenido para 6000 [Hz] y $\alpha = 0,75$ (Enjambre de partículas).

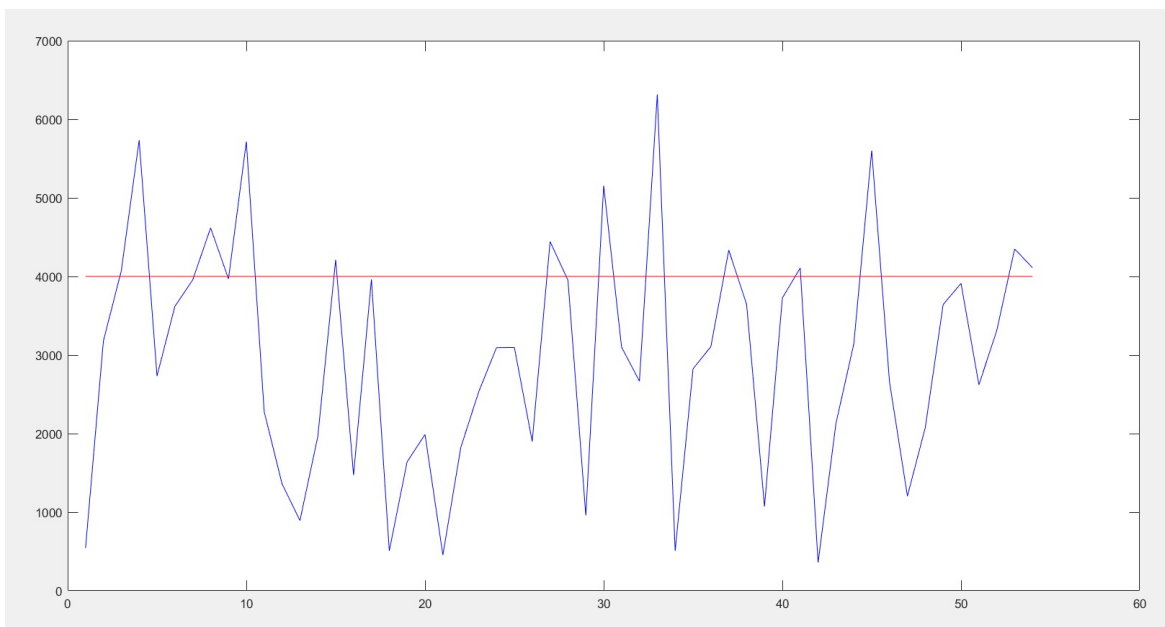


Figura 7.72: Ejemplo de evolución en iteraciones de la frecuencia real del óptimo (azul) y de la predicha por la red (rojo) para 4000 [Hz] y $\alpha = 0,5$ (Enjambre de partículas).

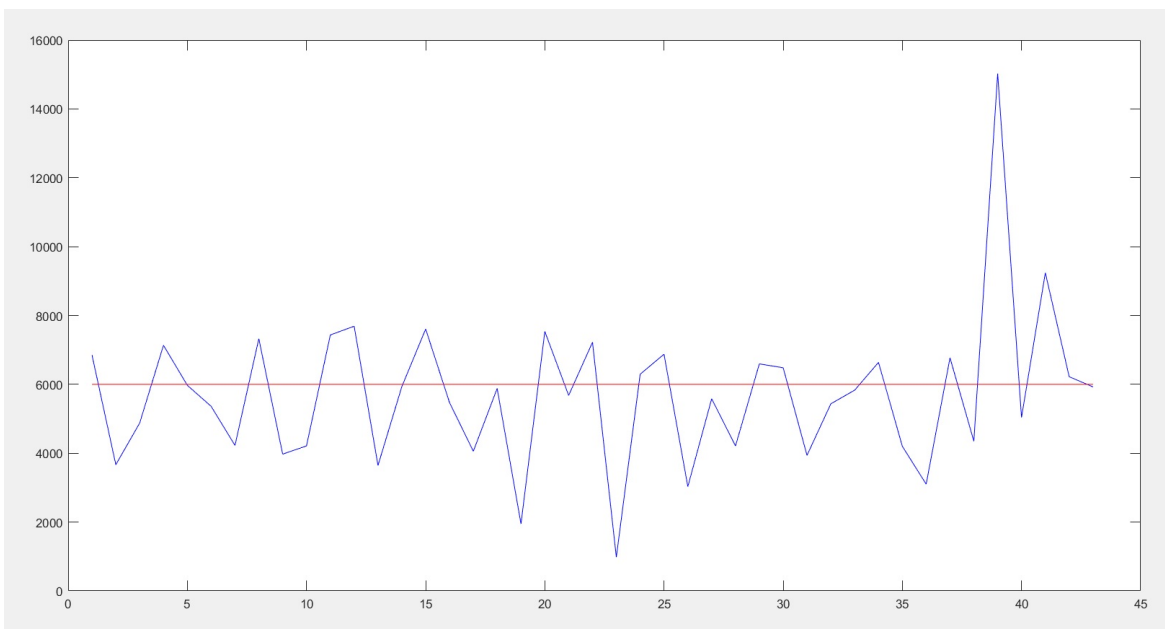


Figura 7.73: Ejemplo de evolución en iteraciones de la frecuencia real del óptimo (azul) y de la predicha por la red (rojo) para 6000 [Hz] y $\alpha = 0,5$ (Enjambre de partículas).

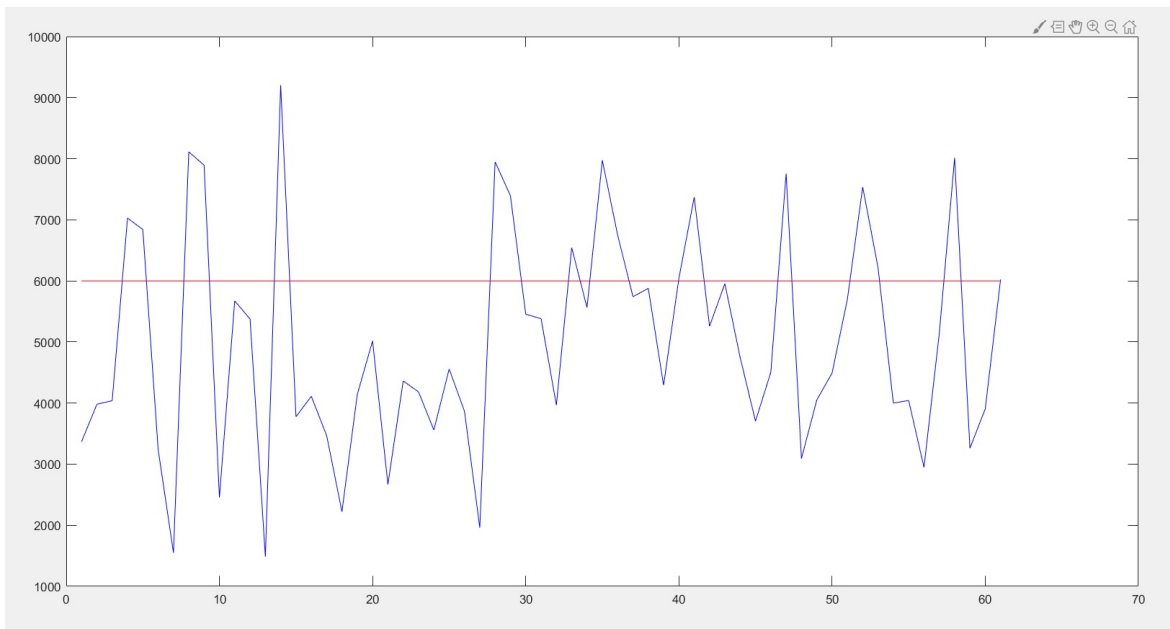


Figura 7.74: Ejemplo de evolución en iteraciones de la frecuencia real del óptimo (azul) y de la predicha por la red (rojo) para 6000 [Hz] y $\alpha = 0,25$ (Enjambre de partículas).

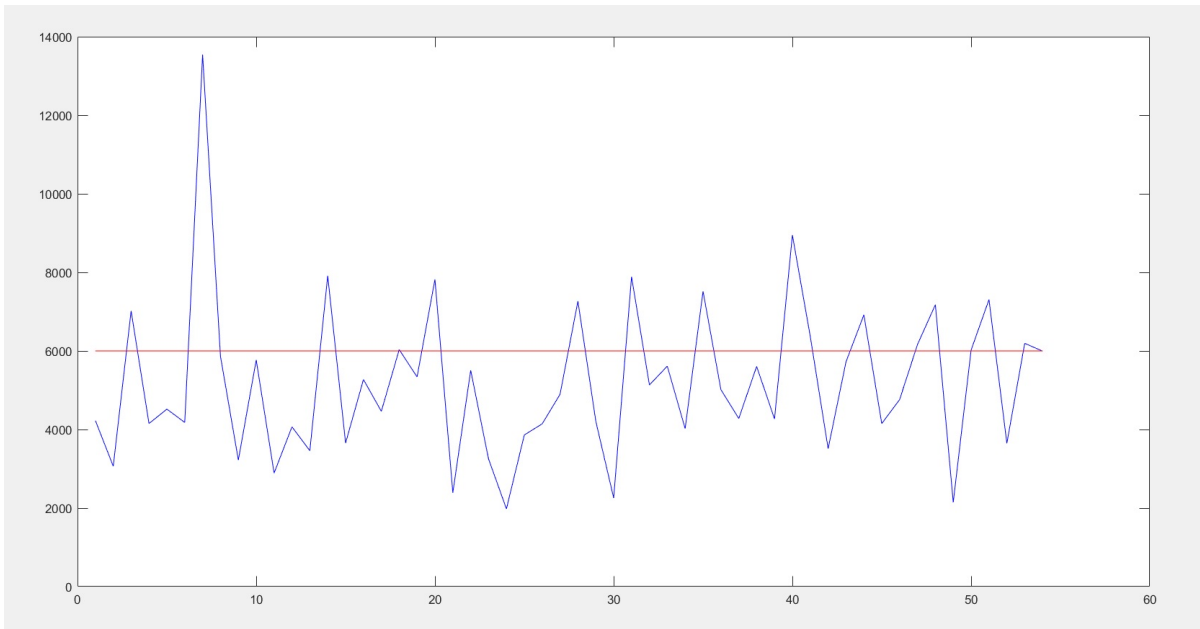


Figura 7.75: Ejemplo de evolución en iteraciones de la frecuencia real del óptimo (azul) y de la predicha por la red (rojo) para 6000 [Hz] y $\alpha = 0,75$ (Enjambre de partículas).

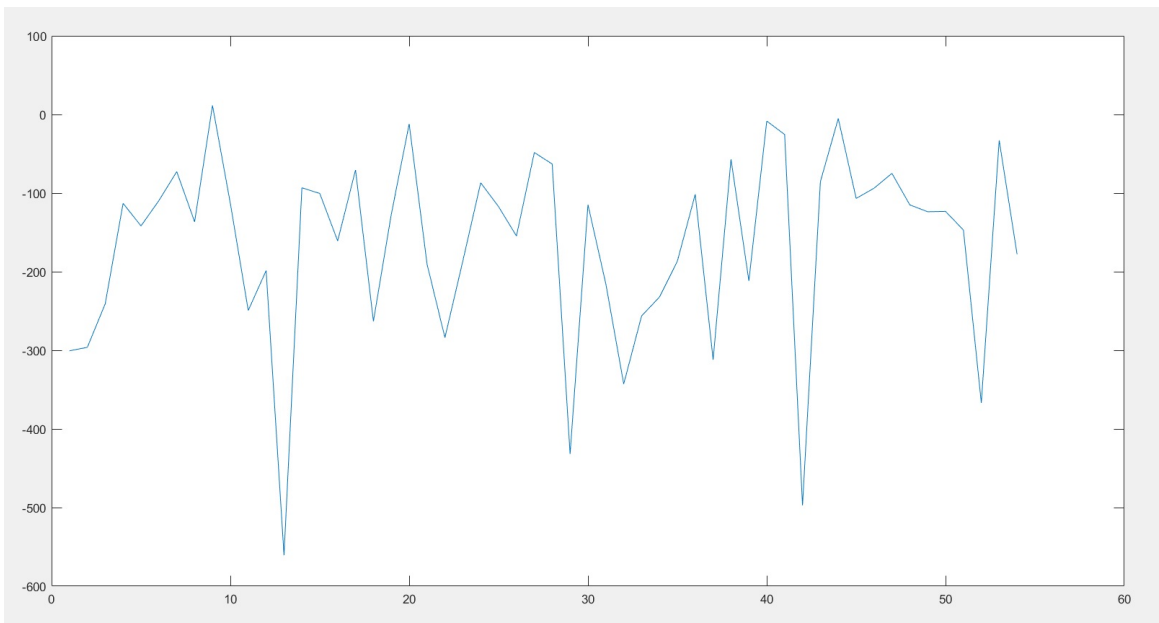


Figura 7.76: Ejemplo de la evolución del valor alcanzado de la función objetivo para 4000 [Hz] y $\alpha = 0,5$ (Enjambre de partículas).

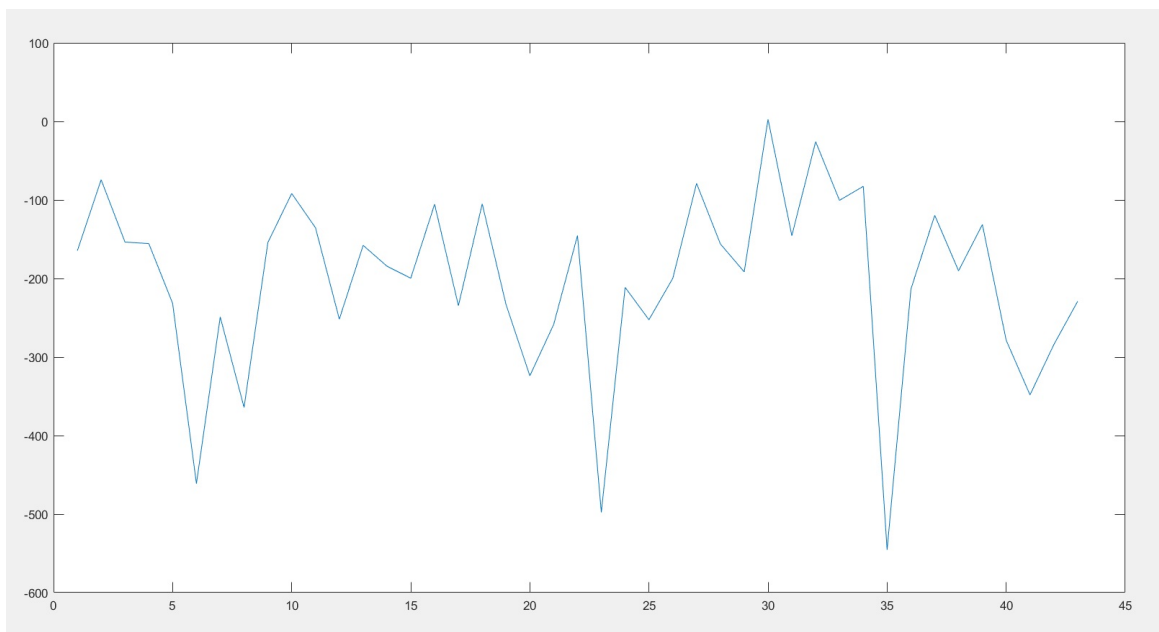


Figura 7.77: Ejemplo de la evolución del valor alcanzado de la función objetivo para 6000 [Hz] y $\alpha = 0,5$ (Enjambre de partículas).

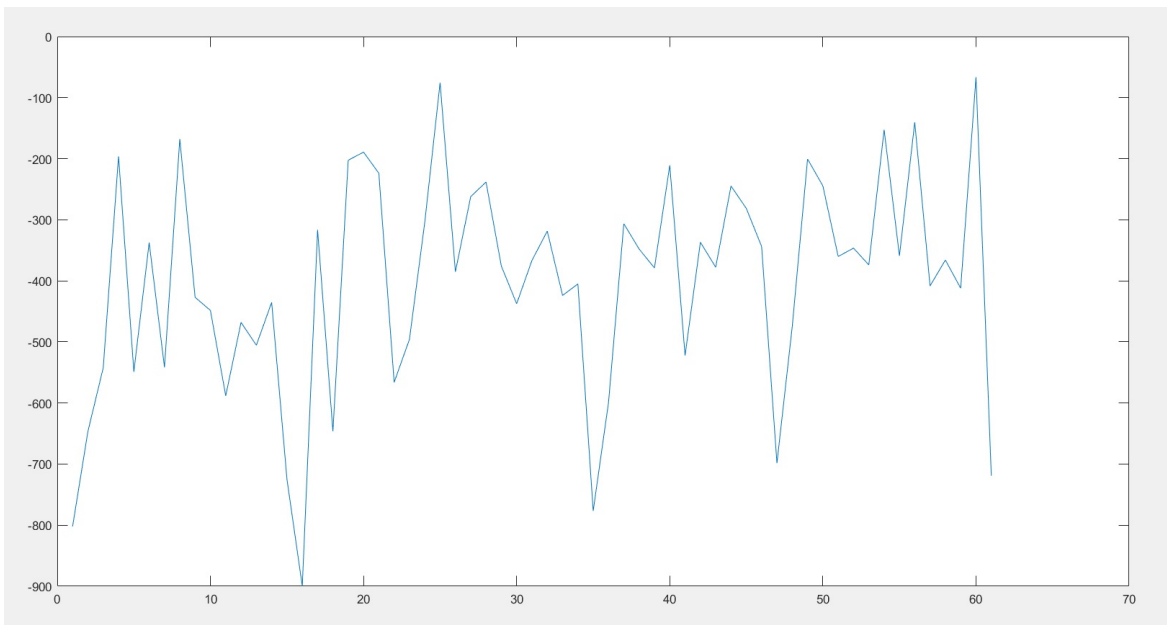


Figura 7.78: Ejemplo de la evolución del valor alcanzado de la función objetivo para 6000 [Hz] y $\alpha = 0,25$ (Enjambre de partículas).

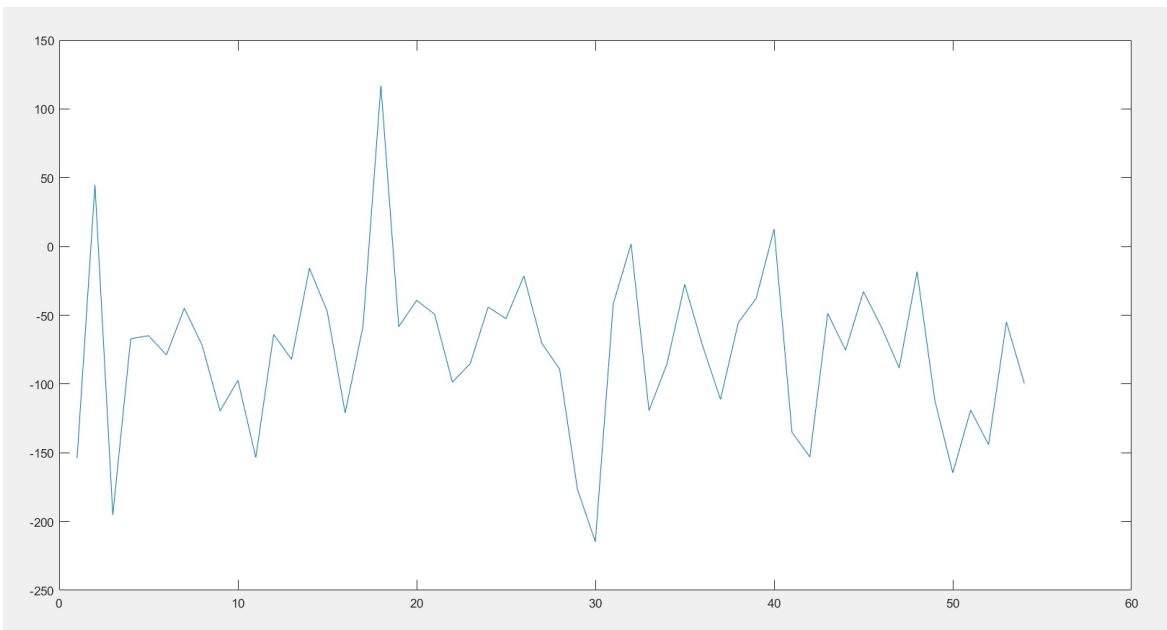


Figura 7.79: Ejemplo de la evolución del valor alcanzado de la función objetivo para 6000 [Hz] y $\alpha = 0,75$ (Enjambre de partículas).

ANEXO B: Códigos de MATLAB

Metodología tridimensional

```
1 clear all
2 warning off all;
3 cd 'C:\Users\vmerru\Desktop\Memoria Ignacio\Completo3D';
4 addpath('C:\Users\vmerru\Desktop\Modelos memoria\prediccion bandas',
        'C:\Users\vmerru\Desktop\Memoria Ignacio');estructuras=load('
        Base_3D.mat');
5 X=estructuras.Base;
6 archivo2 = matfile('Ybg3D.mat');
7 Y = archivo2.BG;
8 N=length(X);
9 path(path, 'C:\Program Files\MATLAB\R2018b\toolbox\sdt');
10 sdtcheck path
11
12 TimeStart=tic;
13
14 %% Construir datos
15
16 disp('Preprocesando Datos')
17
18 %construccion de los sets de datos
19 train=0.5;
20 val=0.3;
21 test=1-train - val;
22
23 a=round(train*N);
24 b=round((1 - test)*N);
25
26 Xtr = reshape(X(1:a,:), [1,1, size(X(1:a,:), 2), size(X(1:a,:), 1)]);
        %entrenamiento
27 Xvl = reshape(X(a+1:b,:), [1,1, size(X(a+1:b,:), 2), size(X(a+1:b,:),
        ,1)]); %validacion
28 Xts = reshape(X(b+1:end,:), [1,1, size(X(b+1:end,:), 2), size(X(b+1:
        end,:), 1)]); %testeo
29
30 Ytr=Y(1:a,:); %entrenamiento
31 Yvl=Y(a+1:b,:); %validacion
32 Yts=Y(b+1:end,:); %testeo
```

```

33
34 %% Arquitectura y opciones de la red neuronal
35
36 disp('Construyendo red')
37
38 A=size(Xtr);
39 B=size(Y);
40
41 nFeatures = size(Xtr,3);
42 nExamples = size(Xtr,4);
43 nOutputs = size(Y,2);
44
45
46 solver='adam';
47 ilr=0.0991;
48 miniBatchSize=476;
49 maxEpochs=1000;
50 fa=reluLayer;
51
52 %Arquitectura de la red
53
54 layers=[imageInputLayer([1 1 nFeatures])
55         fullyConnectedLayer(488)
56         fa
57         fullyConnectedLayer(488)
58         fa
59         dropoutLayer(0.1419)
60         fullyConnectedLayer(488)
61         fa
62         fullyConnectedLayer(nOutputs)
63         regressionLayer];
64
65 % Opciones del entrenamiento de la red neuronal
66
67 options = trainingOptions('adam', ...
68     'MaxEpochs',maxEpochs, ...
69     'MiniBatchSize',miniBatchSize, ...
70     'Plots','none','LearnRateSchedule','piecewise',...
71     'LearnRateDropPeriod',50,'LearnRateDropFactor',0.97,'Verbose'
72     ,0,...
73     'ExecutionEnvironment','cpu','ValidationData',...
74     {Xv1,Yv1},'InitialLearnRate',0.0991);

```

```

74 %% Set Up
75 disp('Set Up');
76
77 %Frecuencia de diseño
78 fobj=13e4;
79
80 nb=8;
81 iteracion=0;
82 maxit=75;
83 optionsag=optimoptions('ga');
84 best=5;
85
86 %% Optimización subrogada
87 disp('Iterando Optimización Subrogada')
88
89 while iteracion < maxit
90     iteracion=iteracion+1;
91     iteracion
92
93     %Entrenar red
94     net = trainNetwork(Xtr,Ytr, layers , options);
95
96     %AG
97     FitnessFunction = @(x) Fag3D(x,net ,fobj);
98     numberOfVariables = 7;
99     lb = [0,0,0,0,0,0,0];
100    ub = [1,1,1,1,1,1,1];
101    continuas = [1,2,3,4,5,6,7];
102
103    tic
104    [x,fval , exitflag , output , population , scores] = ga(FitnessFunction ,
numberOfVariables ,[],[],[],[],lb,ub,[], optionsag);
105    Tag(iteracion)=toc;
106    evolfit(iteracion)=fval;
107
108    %Mejores elementos
109
110    sortedscores=sort(scores);
111    bestind=zeros(best , size(population ,2));
112    for i= 1:5
113        for j = 1:size(sortedscores)
114            if sortedscores(i)==scores(j)
115                bestind(i ,:)=population(j ,:);

```

```

116         end
117     end
118 end
119
120     opt{1}= reshape(bestind(1,:) ', [1,1, size(bestind(1,:),2), size(
bestind(1,:),1)]);
121     alpha{1}=predict(net, opt{1}(:,:,:));
122     opt{2}= reshape(bestind(2,:) ', [1,1, size(bestind(2,:),2), size(
bestind(2,:),1)]);
123     alpha{2}=predict(net, opt{2}(:,:,:));
124     opt{3}= reshape(bestind(3,:) ', [1,1, size(bestind(3,:),2), size(
bestind(3,:),1)]);
125     alpha{3}=predict(net, opt{3}(:,:,:));
126     opt{4}= reshape(bestind(4,:) ', [1,1, size(bestind(4,:),2), size(
bestind(4,:),1)]);
127     alpha{4}=predict(net, opt{4}(:,:,:));
128     opt{5}= reshape(bestind(5,:) ', [1,1, size(bestind(5,:),2), size(
bestind(5,:),1)]);
129     alpha{5}=predict(net, opt{5}(:,:,:));
130
131     %Bandas reales
132     frec=zeros(best,1);
133     ancho=zeros(best,1);
134     for j=1:best
135
136         u=zeros(1,7);
137         for i=1:7
138             u(1,i)=opt{j}(:,: ,i);
139         end
140
141         h=1e-3;
142         r=8e-4;
143         L=0.015+0.015*u(7);
144         f1=0.5+1.5*u(1);
145         f2=0.5+1.5*u(2);
146         f3=0.5+1.5*u(3);
147         f4=0.5+1.5*u(4);
148         f5=0.5+1.5*u(5);
149         f6=0.5+1.5*u(6);
150         fL=0.5;
151         [k0,m0,model,mdof0]=estructura3(L,h,r,f1,f2,f3,f4,f5,f6,fL);
152         fecom('textnode','GroupAll','FontSize',12);
153         NINT=10;

```

```

154     [kx,ky]=wave_vector(NINT,L);
155     [k,m,nul , nub , nub1 , nui]=preparar_matrices(k0,m0,model ,mdof0,L
);
156
157     NINT=10; %numero de divisiones en el diagrama de bandas (por
trayecto)
158     [kx,ky]=wave_vector(NINT,L);
159
160     %ordena las matrices para implementar las condiciones de
borde periodicas
161     [k,m,nul , nub , nub1 , nui]=preparar_matrices(k0,m0,model ,mdof0,L
);
162
163     for i=[1 length(kx)]
164         [kr,mr]=periodic(k,m,L,kx(i),ky(i),nul ,nub ,nub1 ,nui);
165         [lambda] = eig(kr,mr,'chol');
166         lambda=round(real(lambda),5);
167         lambda=lambda(lambda>=0);
168         [lambda,idx]=sort(lambda);
169         banda(:,i) = sqrt((lambda(1:nb)))/2/pi;
170     end
171
172     for i=2:length(kx)-1
173         [kr,mr]=periodic(k,m,L,kx(i),ky(i),nul ,nub ,nub1 ,nui);
174         [lambda] = eigs(kr,mr,15,'smallestabs');
175         lambda=round(real(lambda),5);
176         lambda=lambda(lambda>=0);
177         [lambda,idx]=sort(lambda);
178         banda(:,i) = sqrt((lambda(1:nb)))/2/pi;
179     end
180     bands{j}=banda;
181
182     %Nuevas frec y ancho
183     anchomax=-10^100000;
184     for i=1:7
185         delta=(min(min(bands{j}(i+1:nb,:)))-max(max(bands{j}(1:i
,:),:))));
186         if delta > anchomax
187             ancho(j)=delta;
188             frec(j)=max(max(bands{j}(1:i,:),:))+delta/2;
189             anchomax=ancho(j);
190         end
191     end

```

```

192     end
193
194     %Redefinir sets de entrenamiento
195     for i = 1:best
196         Xtr(:, :, :, size(Xtr,4)+1)=opt{i};
197         Ytr(size(Xtr,4),1)=frec(i);
198         Ytr(size(Xtr,4),2)=ancho(i);
199     end
200
201     %Incorporar los mejores en la población inicial
202     optionsag=optimoptions('ga','FunctionTolerance',0,'
ConstraintTolerance',0,'InitialPopulationMatrix',bestind,'Display
','off');
203
204     %Encontrar índice del mejor
205     mejor=zeros(best,1);
206     for i=1:best
207         mejor(i)=Fag3D(bestind(i,:),net,fobj);
208     end
209
210     for i=1:best
211         if mejor(i)==max(mejor)
212             omega=i;
213         end
214     end
215
216     %Evolucion de la metodología
217
218     hf(iteracion)=frec(omega);
219     ha(iteracion)=ancho(omega);
220     hfpr(iteracion)=alpha{1}(1);
221     hapr(iteracion)=alpha{1}(2);
222
223     %Condición de término
224     if abs(fobj - frec(omega)) < ancho(omega)/2
225         disp('Enrejado logrado')
226         break
227     else
228         if iteracion==maxit
229             disp('Enrejado no logrado')
230         end
231         continue
232     end

```

```

233 end
234 % Fin optimización subrogada
235
236 %Mejor elemento
237
238 plot_band(bands{omega},NINT)
239
240 for i = 1:7
241     enrejado(i)=opt{omega}(:, :, i);
242 end
243
244 %Factor de centro
245
246 fc=(abs(fobj - frec(omega)))/(ancho(omega)/2);
247
248 %% Resultados
249
250 disp('Resultados')
251 disp('Iteración')
252 iteracion
253 disp('Frecuencia real')
254 frec(omega)
255 disp('Ancho real')
256 ancho(omega)
257 disp('Factor de centro')
258 fc
259
260 % Gráfico evolución
261
262 for i=1:size(hf,2)
263     nit(i)=i;
264 end
265
266 plot(nit , hf , 'b' , nit , hfpr , 'r')
267
268 TimeEnd = toc(TimeStart)
269
270 plot(nit , evolfit)
271
272 promedioTag=sum(Tag)/size(nit ,2);
273 maxTag=max(Tag);
274 minTag=min(Tag);

```


Metodología bidimensional: Algoritmo genético

```
1 clear all
2 warning off all;
3 cd 'C:\Users\vmuru\Desktop\Memoria Ignacio\Completo';
4 addpath('C:\Users\vmuru\Desktop\Modelos memoria\prediccion bandas',
        'C:\Users\vmuru\Desktop\Memoria Ignacio');
5 path(path, 'C:\Program Files\MATLAB\R2018b\toolbox\sdt');
6 sdtcheck path
7
8 TimeStart=tic;
9 %% Construir datos
10
11 disp('Preprocesando Datos')
12
13 estructuras=load('Base.mat');
14 X=estructuras.Base;
15 archivo2 = matfile('Ybg.mat');
16 Y = archivo2.BG;
17 N=length(X);
18
19 %%construccion de los sets de datos
20 train=0.5;
21 val=0.3;
22 test=1-train - val;
23
24 a=round(train*N);
25 b=round((1-test)*N);
26
27 Xtr = reshape(X(1:a,:), [1,1, size(X(1:a,:),2), size(X(1:a,:),1)]);
        %entrenamiento
28 Xvl = reshape(X(a+1:b,:), [1,1, size(X(a+1:b,:),2), size(X(a+1:b,:),
        ,1)]); %validacion
29 Xts = reshape(X(b+1:end,:), [1,1, size(X(b+1:end,:),2), size(X(b+1:
        end,:),1)]); %testeo
30
31 Ytr=Y(1:a,:); %entrenamiento
32 Yvl=Y(a+1:b,:); %validacion
33 Yts=Y(b+1:end,:); %testeo
34
35 %% Arquitectura y opciones de la red neuronal
```

```

36 disp('Construyendo red')
37
38 A=size(Xtr);
39 B=size(Y);
40
41 nFeatures = size(Xtr,3);
42 nExamples = size(Xtr,4);
43 nOutputs = size(Y,2);
44
45 npc=267;
46 dropperiod=50;
47 dropfactor=0.97;
48 ilr=0.0381;
49 maxEpochs = 1000;
50 miniBatchSize = 421;
51 dof=0.1415;
52
53 layersP = importKerasLayers("modeloPaula.h5");
54 layersP2 = importKerasLayers("modeloPaula2.h5");
55 fa=reluLayer;
56
57 layers=[imageInputLayer([1 1 nFeatures])
58         fullyConnectedLayer(npc)
59         fa
60         fullyConnectedLayer(npc)
61         fa
62         dropoutLayer(dof)
63         fullyConnectedLayer(npc)
64         fa
65         fullyConnectedLayer(nOutputs)
66         regressionLayer];
67
68 % Opciones del entrenamiento de la red neuronal
69
70 options = trainingOptions('adam', ...
71     'MaxEpochs',maxEpochs, ...
72     'MiniBatchSize',miniBatchSize, ...
73     'Plots','none','LearnRateSchedule','piecewise',... %,'training-
74     progress'
75     'LearnRateDropPeriod',dropperiod,'LearnRateDropFactor',
76     dropfactor,'Verbose',0,...
77     'ExecutionEnvironment','cpu','ValidationData',...
78     {Xv1,Yv1},'InitialLearnRate',ilr);

```

```

77
78 %% Set Up
79 disp('Set Up');
80 fobj=6000;
81 nb=8;
82 iteracion=0;
83 maxit=3;
84 optionsag=optimoptions('ga','FunctionTolerance',0,'
    ConstraintTolerance',0);
85 best=5;
86
87 %% Optimización subrogada
88 disp('Iterando Optimización Subrogada')
89
90 while iteracion < maxit
91     iteracion=iteracion+1;
92     iteracion
93     %Entrenar;
94     net = trainNetwork(Xtr,Ytr, layers , options);
95
96     %AG
97     FitnessFunction = @(x) Fagcomp(x,net,fobj);
98     numberOfVariables = 15;
99     lb = [0,0,0,0,0,0,0,0,0,0.3,0.3,0.3,0.3,0.3,0.3,0.3];
100    ub = [1,1,1,1,1,1,1,1,1,1.4,1.4,1.4,1.4,1.4,1.4,1.4];
101    binarias = [1,2,3,4,5,6,7,8];
102    continuas = [9,10,11,12,13,14,15];
103
104    tic
105    [x,fval,exitflag,output,population,scores] = ga(FitnessFunction,
    numberOfVariables,[],[],[],[],lb,ub,[],binarias,optionsag);
106    Tag(iteracion)=toc;
107    evolfit(iteracion)=fval;
108
109    sortedscores=sort(scores);
110    bestind=zeros(best,size(population,2));
111    for i= 1:5
112        for j = 1:size(sortedscores)
113            if sortedscores(i)==scores(j)
114                bestind(i,:)=population(j,:);
115            end
116        end
117    end

```

```

118     opt{1}= reshape(bestind(1,:) ', [1,1,size(bestind(1,:) ,2),size(
bestind(1,:) ,1)]);
119     alpha{1}=predict(net,opt{1}(:,:,:,1));
120     opt{2}= reshape(bestind(2,:) ', [1,1,size(bestind(2,:) ,2),size(
bestind(2,:) ,1)]);
121     alpha{2}=predict(net,opt{2}(:,:,:,1));
122     opt{3}= reshape(bestind(3,:) ', [1,1,size(bestind(3,:) ,2),size(
bestind(3,:) ,1)]);
123     alpha{3}=predict(net,opt{3}(:,:,:,1));
124     opt{4}= reshape(bestind(4,:) ', [1,1,size(bestind(4,:) ,2),size(
bestind(4,:) ,1)]);
125     alpha{4}=predict(net,opt{4}(:,:,:,1));
126     opt{5}= reshape(bestind(5,:) ', [1,1,size(bestind(5,:) ,2),size(
bestind(5,:) ,1)]);
127     alpha{5}=predict(net,opt{5}(:,:,:,1));
128
129     %Bandas reales
130     frec=zeros(best,1);
131     ancho=zeros(best,1);
132     for j=1:best
133         u=zeros(1,15);
134         for i=1:15
135             u(1,i)=opt{j}(:,: ,i);
136         end
137         L=100e-3*u(9);
138         [k0,m0,model,mdof0]=barras(u);
139         fecom('textnode','GroupAll','FontSize',12);
140         NINT=10;
141         [kx,ky]=wave_vector(NINT,L);
142         [k,m,nul,nub,nubl,nui]=preparar_matrices(k0,m0,model,mdof0,L
);
143
144         %-----Arreglo error start
145
146         if or((nub*2+nul*2+nubl*4+nui)~=size(k0,1),(nub*2+nul*2+nubl
*4+nui)~=size(k0,2))
147             break
148         end
149
150         %-----Arreglo error end
151
152         for i=1:length(kx)
153             [kr,mr]=periodic(k,m,L,kx(i),ky(i),nul,nub,nubl,nui);

```

```

154         [lambda] = eigs(kr,mr,25,'smallestabs');
155         lambda=round(real(lambda),5);
156         lambda=lambda(lambda>=0);
157         [lambda,idx]=sort(lambda);
158         bands{j}(:,i) = sqrt((lambda(1:nb)))/2/pi;
159     end
160
161     %Nuevas frec y ancho
162     anchomax=-10^100000;
163     for i=1:7
164         delta=(min(min(bands{j}(i+1:nb,:)))-max(max(bands{j}(1:i
165         ,:)))));
166         if delta>anchomax
167             ancho(j)=delta;
168             frec(j)=max(max(bands{j}(1:i,:)))+delta/2;
169             anchomax=ancho(j);
170         end
171     end
172
173     %-----Arreglo error start
174
175     if or((nub*2+nul*2+nubl*4+nui)~=size(k0,1),(nub*2+nul*2+nubl*4+
176     nui)~=size(k0,2))
177         disp('Falla dimensional en T')
178         vars = {'k','m','L','kx','ky','nul','nub','nubl','nui'};
179         clear(vars{:})
180         iteracion=iteracion-1;
181         continue
182     end
183
184     %-----Arreglo error fin
185
186     %Redefinir sets de entrenamiento
187     for i = 1:best
188         Xtr(:, :, :, size(Xtr,4)+1)=opt{i};
189         Ytr(size(Xtr,4),1)=frec(i);
190         Ytr(size(Xtr,4),2)=ancho(i);
191     end
192
193     %Incorporar los mejores en la población inicial
194     optionsag=optimoptions('ga','FunctionTolerance',0,'
195     ConstraintTolerance',0,'InitialPopulationMatrix',bestind,'Display

```

```

    ', 'off');

194     %Encontrar índice del mejor
195     mejor=zeros(best,1);
196     for i=1:best
197         mejor(i)=Fagcomp(bestind(i,:),net,fobj);
198     end
199
200     for i=1:best
201         if mejor(i)==max(mejor)
202             omega=i;
203         end
204     end
205
206     %Evolucion de la metodología
207
208     hf(iteracion)=frec(omega);
209     ha(iteracion)=ancho(omega);
210     hfpr(iteracion)=alpha{1}(1);
211     hapr(iteracion)=alpha{1}(2);
212
213     %Condición de término
214     if abs(fobj - frec(omega)) < ancho(omega)/2
215         disp('Enrejado logrado')
216         break
217     else
218         if iteracion==maxit
219             disp('Enrejado no logrado')
220         end
221         continue
222     end
223
224 end
225
226 %Mejor
227 plot_band(bands{omega},NINT)
228
229 for i = 1:15
230     enrejado(i)=opt{omega}(:, :, i);
231 end
232
233 %Factor de centro

```

```

234 fc=(abs(fobj - frec(omega)))/(ancho(omega)/2);
235
236 %% Resultados
237
238 disp('Resultados')
239 disp('Iteración')
240 iteracion
241 disp('Frecuencia real')
242 frec(omega)
243 disp('Ancho real')
244 ancho(omega)
245 disp('Factor de centro')
246 fc
247
248 % Gráfico evolución
249
250 for i=1:size(hf,2)
251     nit(i)=i;
252 end
253
254 plot(nit , hf , 'b' , nit , hfpr , 'r')
255
256 TimeEnd = toc(TimeStart)
257
258 plot(nit , evolfit)
259
260 promedioTag=sum(Tag)/size(nit ,2)
261 max(Tag)
262 min(Tag)

```

Metodología bidimensional: Enjambre de partículas

```
1 clear all
2 warning off all;
3 cd 'C:\Users\vmuru\Desktop\Memoria Ignacio\Completo';
4 addpath('C:\Users\vmuru\Desktop\Modelos memoria\prediccion bandas',
        'C:\Users\vmuru\Desktop\Memoria Ignacio');
5 path(path, 'C:\Program Files\MATLAB\R2018b\toolbox\sdt');
6 sdtcheck path
7
8 TimeStart=tic;
9 %% Construir datos
10
11 disp('Preprocesando Datos')
12
13 estructuras=load('Base.mat');
14 X=estructuras.Base;
15 archivo2 = matfile('Ybg.mat');
16 Y = archivo2.BG;
17 N=length(X);
18
19 %%construccion de los sets de datos
20 train=0.5;
21 val=0.3;
22 test=1-train - val;
23
24 a=round(train*N);
25 b=round((1 - test)*N);
26
27 Xtr = reshape(X(1:a,:), [1,1, size(X(1:a,:), 2), size(X(1:a,:), 1)]);
        %entrenamiento
28 Xvl = reshape(X(a+1:b,:), [1,1, size(X(a+1:b,:), 2), size(X(a+1:b,:),
        1)]); %validacion
29 Xts = reshape(X(b+1:end,:), [1,1, size(X(b+1:end,:), 2), size(X(b+1:
        end,:), 1)]); %testeo
30
31 Ytr=Y(1:a,:); %entrenamiento
32 Yvl=Y(a+1:b,:); %validacion
33 Yts=Y(b+1:end,:); %testeo
34
35 %% Arquitectura y opciones de la red neuronal
```



```

36 disp('Construyendo red')
37
38 A=size(Xtr);
39 B=size(Y);
40
41 nFeatures = size(Xtr,3);
42 nExamples = size(Xtr,4);
43 nOutputs = size(Y,2);
44
45 npc=267;
46 dropperiod=50;
47 dropfactor=0.97;
48 ilr=0.0381;
49 maxEpochs = 1000;
50 miniBatchSize = 421;
51 dof=0.1415;
52
53 fa=reluLayer;
54
55 layers=[imageInputLayer([1 1 nFeatures]) %6 capas inicialmente
56         fullyConnectedLayer(npc)
57         fa
58         fullyConnectedLayer(npc)
59         fa
60         dropoutLayer(dof)
61         fullyConnectedLayer(npc)
62         fa
63         fullyConnectedLayer(nOutputs)
64         regressionLayer];
65
66 % Opciones del entrenamiento de la red neuronal
67
68 options = trainingOptions('adam', ...
69     'MaxEpochs',maxEpochs, ...
70     'MiniBatchSize',miniBatchSize, ...
71     'Plots','none','LearnRateSchedule','piecewise',... %,'training-
72     progress'
73     'LearnRateDropPeriod',dropperiod,'LearnRateDropFactor',
74     dropfactor,'Verbose',0,...
75     'ExecutionEnvironment','cpu','ValidationData',...
76     {Xv1,Yv1},'InitialLearnRate',ilr);

```

```

75
76 %% Set Up
77 disp('Set Up');
78 fobj=2000;
79 nb=8;
80 iteracion=0;
81 maxit=75;
82 optionsps=optimoptions('particleswarm','FunctionTolerance',1);
83 best=5;
84
85 %% Optimización subrogada
86 disp('Iterando Optimización Subrogada')
87
88 while iteracion < maxit
89     iteracion=iteracion+1;
90     iteracion
91     %Entrenar
92     net = trainNetwork(Xtr,Ytr, layers , options);
93
94     %AG
95     FitnessFunction = @(x) Fpscomp(x, net , fobj);
96     nvars = 15;
97     lb = [0,0,0,0,0,0,0,0,0.3,0.3,0.3,0.3,0.3,0.3,0.3];
98     ub = [1,1,1,1,1,1,1,1,1.4,1.4,1.4,1.4,1.4,1.4,1.4];
99     binarias = [1,2,3,4,5,6,7,8];
100    continuas = [9,10,11,12,13,14,15];
101
102    tic
103    [x,fval,exitflag,output]= particleswarm(FitnessFunction,nvars,lb
,ub);
104    evolfit(iteracion)=fval;
105    Tps(iteracion)=toc;
106
107    for i=1:8
108        if round(x(i))>=0.5
109            x(i)=1;
110        else
111            x(i)=0;
112        end
113    end
114
115    for i= 1:5
116        bestind(i,:)=x;

```

```

117     end
118     opt{1}= reshape(bestind(1,:) ', [1,1, size(bestind(1,:) ,2), size(
bestind(1,:) ,1)]);
119     alpha{1}=predict(net ,opt {1}(:,:,:,1));
120     opt{2}= reshape(bestind(2,:) ', [1,1, size(bestind(2,:) ,2), size(
bestind(2,:) ,1)]);
121     alpha{2}=predict(net ,opt {2}(:,:,:,1));
122     opt{3}= reshape(bestind(3,:) ', [1,1, size(bestind(3,:) ,2), size(
bestind(3,:) ,1)]);
123     alpha{3}=predict(net ,opt {3}(:,:,:,1));
124     opt{4}= reshape(bestind(4,:) ', [1,1, size(bestind(4,:) ,2), size(
bestind(4,:) ,1)]);
125     alpha{4}=predict(net ,opt {4}(:,:,:,1));
126     opt{5}= reshape(bestind(5,:) ', [1,1, size(bestind(5,:) ,2), size(
bestind(5,:) ,1)]);
127     alpha{5}=predict(net ,opt {5}(:,:,:,1));
128
129     %Bandas reales
130     frec=zeros(best ,1);
131     ancho=zeros(best ,1);
132     for j=1:best
133         u=zeros(1,15);
134         for i=1:15
135             u(1,i)=opt {j}(:,: ,i);
136         end
137         L=100e-3*u(9);
138         [k0,m0,model ,mdof0]=barras(u);
139         fecom('textnode' , 'GroupAll' , 'FontSize' ,12);
140         NINT=10;
141         [kx ,ky]=wave_vector(NINT,L);
142         [k,m,nul ,nub ,nubl ,nui]=preparar_matrices(k0,m0,model ,mdof0,L
);
143
144         %-----Arreglo error start
145
146         if or((nub*2+nul*2+nubl*4+nui)~=size(k0,1) ,(nub*2+nul*2+nubl
*4+nui)~=size(k0,2))
147             break
148         end
149
150         %-----Arreglo error end
151
152         for i=1:length(kx)

```

```

153         [kr, mr]=periodic(k, m, L, kx(i), ky(i), nul, nub, nub1, nui);
154         [lambda] = eigs(kr, mr, 25, 'smallestabs');
155         lambda=round(real(lambda), 5);
156         lambda=lambda(lambda>=0);
157         [lambda, idx]=sort(lambda);
158         bands{j}(:, i) = sqrt((lambda(1:nb)))/2/pi;
159     end
160
161     %Nuevas frec y ancho
162     anchomax=-10^100000;
163     for i=1:7
164         delta=(min(min(bands{j}(i+1:nb, :))) - max(max(bands{j}(1:i
165 , :)))));
166         if delta > anchomax
167             ancho(j)=delta;
168             frec(j)=max(max(bands{j}(1:i, :))) + delta/2;
169             anchomax=ancho(j);
170         end
171     end
172
173     %------Arreglo error start
174
175     if or((nub*2+nul*2+nubl*4+nui)~=size(k0, 1), (nub*2+nul*2+nubl*4+
176 nui)~=size(k0, 2))
177         disp('Falla dimensional en T')
178         vars = {'k', 'm', 'L', 'kx', 'ky', 'nul', 'nub', 'nubl', 'nui'};
179         clear(vars{:})
180         iteracion=iteracion - 1;
181         continue
182     end
183
184     %------Arreglo error fin
185
186     %Redefinir sets de entrenamiento
187     for i = 1:best
188         Xtr(:, :, :, size(Xtr, 4)+1)=opt{i};
189         Ytr(size(Xtr, 4), 1)=frec(i);
190         Ytr(size(Xtr, 4), 2)=ancho(i);
191     end
192
193     %Incorporar los mejores en la población inicial

```

```

193     optionsps=optimoptions('particleswarm','InitialSwarmMatrix',
bestind,'Display','off','FunctionTolerance',0.000001);
194
195     %Encontrar índice del mejor
196     mejor=zeros(best,1);
197     for i=1:best
198         mejor(i)=Fpscomp(bestind(i,:),net,fobj);
199     end
200
201     for i=1:best
202         if mejor(i)==max(mejor)
203             omega=i;
204         end
205     end
206
207     %Evolucion de la metodología
208
209     hf(iteracion)=frec(omega);
210     ha(iteracion)=ancho(omega);
211     hfpr(iteracion)=alpha{1}(1);
212     hapr(iteracion)=alpha{1}(2);
213
214     %Condición de término
215     if abs(fobj - frec(omega)) < ancho(omega)/2
216         disp('Enrejado logrado')
217         break
218     else
219         if iteracion==maxit
220             disp('Enrejado no logrado')
221         end
222         continue
223     end
224
225 end
226
227 %Mejor
228 plot_band(bands{omega},NINT)
229
230 for i = 1:15
231     enrejado(i)=opt{omega}(:, :, i);
232 end
233
234 %Factor de centro

```

```

235 fc=(abs(fobj - frec(omega)))/(ancho(omega)/2);
236
237
238
239 %% Resultados
240
241 disp('Resultados')
242 disp('Iteración')
243 iteracion
244 disp('Frecuencia real')
245 frec(omega)
246 disp('Ancho real')
247 ancho(omega)
248 disp('Factor de centro')
249 fc
250
251 % Gráfico evolución
252
253 for i=1:size(hf,2)
254     nit(i)=i;
255 end
256
257 TimeEnd = toc(TimeStart)
258
259 plot(nit , evolfit)
260 plot(nit , hf , 'b' , nit , hfpr , 'r')
261
262 promedioTps=sum(Tps)/size(nit ,2);
263 maxtps=max(Tps);
264 mintps=min(Tps);

```