



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

DETECCIÓN DE DAÑOS EN UNA VIGA SIMPLE MEDIANTE ANTIRESONANCIA Y REDES NEURONALES

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL MECANICO

JAVIER ANTONIO MAHU SINCLAIR

PROFESORA GUÍA:
VIVIANA MERUANE NARANJO

MIEMBROS DE LA COMISIÓN:
JUAN CRISTÓBAL ZAGAL MONTEALEGRE
EDUARDO SALAMANCA HENRÍQUEZ

Este trabajo ha sido parcialmente financiado por proyecto Fondecyt

SANTIAGO DE CHILE
MAYO 2012

Resumen

El objetivo de este trabajo es desarrollar un algoritmo capaz de detectar, localizar y cuantificar daños en una estructura simple en tiempo real, considerando como daño la disminución en la rigidez en uno o más elementos. Para ello se propuso utilizar Redes Neuronales Artificiales (RNA), que son métodos no explícitos utilizados en problemas de aproximación de curvas, clasificación e identificación de parámetros. Se plantea utilizar como información de entrada del algoritmo las frecuencias de antiresonancia, obteniendo como resultado la disminución de rigidez efectiva para cada elemento de la estructura.

El entrenamiento y validación de la red neuronal se realiza en cuatro etapas, las que se señalan a continuación:

- La determinación de las frecuencias características, resonantes y antiresonantes.
- Generación de patrones de entrenamiento.
- El diseño y entrenamiento de la RNA.
- La validación analítica y experimental, del algoritmo de detección de daños.

En la realización de este trabajo se ha conseguido entrenar redes neuronales capaces de detectar hasta 2 daños simultáneos entre 20 posibles ubicaciones a lo largo de una viga. Los resultados fueron validados con datos de vigas experimentales, a las que se les introdujo daño de forma artificial. La red utilizada es una red neuronal multicapa de una capa oculta, que presenta un 32,64% de falsos negativos, 8,36% de falsos positivos y un 7.96% de daños mal cuantificados a nivel global.

Los posibles usos de redes neuronales en aplicaciones de detección de daños en tiempo real y los resultados obtenidos, motivan futuros estudios que se pueden centrar en la búsqueda de mejores resultados haciendo uso de mejores diseños de redes neuronales o mejores formas de modelar el ruido experimental. También es necesario explorar los modos en que estos algoritmos se puedan aplicar a estructuras más grandes y complejas.

Agradecimientos

Primero que todo, gracias a Dios y a todos en los que me ha acompañado durante estos seis años de estudio y formación, tanto personal como académica. También le doy gracias a Dios porque la lista no es corta, profesores/as, funcionarios/as, compañeros/as, familiares, amigos y amigas de la vida, a quienes no alcanzo a nombrar en estas palabras, pero les estoy profundamente agradecido y lo saben. De forma especial, quiero agradecer a la profesora Viviana Meruane por generar las condiciones para desarrollar de manera adecuada un tema tan relevante y vigente, por su guía y respaldo en estos meses de trabajo e investigación. También agradecer a los profesores de la comisión, J. Zagal y E. Salamanca, y los profesores encargados del proceso de titulación, M. Béjar y A. Sepúlveda, por su disposición y contribuciones en las instancias correspondientes para mejorar este trabajo.

Índice

Resumen.....	ii
Agradecimientos.....	iii
Índice.....	iv
1. Introducción	1
2. Antecedentes	4
2.1 Vibraciones Mecánicas	4
2.2 Identificación de Daños	6
2.3 Nociones de Elementos Finitos para Vibraciones Mecánicas	7
2.4 Medición Experimental	9
2.5 Ajuste de Modelos.....	10
2.6 Red Neuronal Artificial (RNA).....	11
2.7 Diseño y Entrenamiento de RNA	11
2.8 Retro-propagación y otros Métodos de Entrenamiento	15
2.9 Aplicaciones de RNA en ingeniería.....	18
2.10 Funciones de Error e Indicadores de Desempeño	20
3. Metodología	23
3.1 Metodología de Entrenamiento de RNA.....	24
4. Resultados Problema 1	25
4.1 Ajuste de Modelo	26
4.2 Selección RNA	26
4.3 Red Neuronal.....	30
4.4 Prueba Experimental.....	35
5. Resultados Problema 2.....	37
5.1 Ajuste de Modelo	38

5.2	Red Neuronal (Vertical).....	40
5.3	Prueba Experimental (Vertical)	46
5.4	Red Neuronal (Horizontal).....	49
5.5	Prueba Experimental (Horizontal)	53
5.6	Resultado Finales	56
6.	Discusión	59
7.	Conclusiones	63
8.	Trabajos Futuros.....	64
9.	Referencias.....	65
10.	Anexo A	67
11.	Anexo B	68
12.	Anexo C	70
13.	Anexo D	72
14.	Anexo E	74
15.	Anexo F.....	75
16.	Anexo G: Indicadores de desempeño para patrones sin ruido	76
17.	Anexo H: Frecuencias antiresonantes	82

1. Introducción

La motivación de este trabajo surge de la importancia de una detección oportuna de daños estructurales, ya que esta permite tomar medidas apropiadas para prolongar la vida de la estructura y/o evitar daños catastróficos. El propósito de la evaluación de daños es detectar y caracterizar un daño estructural en la etapa más temprana posible, y estimar cuánto tiempo se dispone antes que se requiera mantenimiento o se produzca una falla, lo que conlleva importantes beneficios económicos y tiene un alto potencial para la seguridad de las personas. Este procedimiento se puede aplicar a estructuras de ingeniería civil, como edificios o puentes; a vehículos de transporte, como aviones, helicópteros, trenes o barcos y a equipos industriales, como molinos, turbinas, bombas, calderas, entre otros.

Dentro de las posibles técnicas para detectar daños (radiografías, ultrasonido, entre otras) se requiere encontrar una cuya aplicación sea confiable y eficiente, de tal manera de obtener datos concisos que lleven a buenas conclusiones en un tiempo que permita tomar medidas apropiadas. Dentro de las técnicas actuales, predominan las técnicas locales que permiten obtener información de zonas particulares. Esto puede tener dos inconvenientes; primero, se debe tener algún grado de sospecha de la ubicación de daño y segundo, existen zonas difíciles de acceder para realizar mediciones. Por lo tanto, se requieren métodos que permitan tener una mirada global de la estructura y/o permitan realizar mediciones remotas con gran precisión.

Una técnica emergente en los últimos años es la detección de daño a partir de cambios en la respuesta dinámica de una estructura. El concepto básico es que las vibraciones características (frecuencias naturales, formas modales, amortiguación, función de respuesta de frecuencia, entre otras) [1,2,3] son funciones de las propiedades físicas de la estructura. Por lo tanto, cambios en el material o en las propiedades geométricas de una estructura debido a un daño, aparecen reflejados en cambios detectables en sus características de vibración. Las vibraciones características de una estructura se pueden predecir mediante modelos analíticos o numéricos, como el método de elementos finitos. Además, este tipo de respuestas se pueden obtener en terreno con un proceso de medición bastante directo y confiable, lo que permite tener una visión global de la estructura y la implementación de sensores remotos.

La principal dificultad en la evaluación de daños es determinar la presencia, localización y severidad del daño estructural, dada la respuesta dinámica de la estructura. Los algoritmos más exitosos de evaluación de daño, son algoritmos basados en ajustes de modelos que utilizan métodos de optimización global [1]. Sin embargo, estos algoritmos son muy lentos, y el proceso de evaluación de daños se logra a través de un proceso inverso que requiere mucho tiempo. Esto es un problema para aplicaciones de monitoreo en tiempo real. Las Redes Neuronales Artificiales (RNA) se

han introducido recientemente como una alternativa a los métodos de ajuste de modelos [2,3]. Una vez que una red neuronal ha sido debidamente entrenada, puede potencialmente detectar, localizar y cuantificar daños estructurales con precisión en tiempo real. Una red neuronal es un algoritmo de procesamiento de datos, que trata de emular el sistema de procesamiento del cerebro humano [2]. Está formado por "neuronas" interconectadas entre si para construir una red compleja. El conocimiento es adquirido por un proceso de aprendizaje y se almacena en las interconexiones neuronales.

Un segundo desafío en la evaluación de daños estructurales es la selección de un indicador, a partir de la respuesta del sistema que sea lo suficientemente sensible a daños pequeños. El hecho de que se han estudiado muchos indicadores, y siguen siendo investigados, sin consenso en cuanto al óptimo, pone en evidencia la dificultad del problema. Las frecuencias naturales y los modos de vibración son las propiedades más frecuentemente utilizadas [2,3]. En general, las frecuencias naturales detectan y cuantifican un daño, mientras que los modos de vibración lo localizan. Recientemente, se ha propuesto el uso de frecuencias de antiresonancia como una alternativa a los modos de vibración [4,5]. Las antiresonancias son una buena alternativa a los modos de vibración como valores de entrada a una red neuronal, ya que el número de variables de entrada es menor, están menos contaminadas por el ruido y siguen ofreciendo la misma información que los modos de vibración. Sin embargo, el uso de las antiresonancias todavía está en fase de desarrollo y su aplicación a la detección de daño estructural aun no ha sido investigada a fondo.

El objetivo de este trabajo es desarrollar un algoritmo capaz de detectar, localizar y cuantificar daño en una viga simple en tiempo real. Para ello se propone utilizar redes neuronales artificiales, donde las frecuencias de antiresonancia son las entradas a la red y las salidas son los factores de rigidez efectiva para cada elemento de la estructura. Para esto se tendrán los siguientes objetivos específicos:

- Determinar las frecuencias resonantes y antiresonantes, para una viga simple usando Elementos Finitos.
- Analizar los elementos que constituyen una Red Neuronal y seleccionar la configuración que mejor se adecue a la resolución del problema de estudio.
- Determinar el número de datos necesarios tanto para entrenar la Red Neuronal como para localizar y cuantificar los daños.
- Validar analítica y experimentalmente los resultados.

Para el logro de estos objetivos se estudian dos problemas, un sistema masa-resorte y una viga libre. Se los ha elegido ya que estos son la primera aproximación a una estructura de mayor complejidad y son problemas ampliamente estudiados en el área.

En este trabajo se presentan inicialmente los antecedentes necesarios para contextualizar y realizar el trabajo propuesto. Primero se aborda el problema de las vibraciones mecánicas en varias secciones que incluyen: introducción analítica y conceptual al problema, representación de daños en modelos analíticos, utilización de elementos finitos y mediciones experimentales asociadas a este problema, junto con ajustes de modelos que permiten mejorar la correlación de los modelos analíticos con la estructura real. Luego, se abordan las Redes Neuronales, incluyendo la aplicación de estas a problemas de ingeniería. También se introducen distintos indicadores que permiten calificar y comparar la capacidad de detectar daños en los problemas estudiados. A continuación, se presenta la metodología a utilizar en la resolución de los distintos problemas, junto con algunas consideraciones que se deben tener para llevarla a cabo. Finalmente, se presenta los resultados y su discusión para los dos problemas planteados, donde se revisando el método de selección de las redes, el desempeño de estas en la detección de daños, la influencia de los daños en la respuesta dinámica y la validación experimental de estos métodos.

Este trabajo es financiado en parte por el fondo nacional de desarrollo científico y tecnológico del Gobierno de Chile, siendo parte del proyecto Fondecyt de iniciación N° 11110046.

2. Antecedentes

En esta sección se contextualizan y discuten los antecedentes relevantes para la comprensión y elaboración del presente trabajo.

2.1 Vibraciones Mecánicas

En una estructura, se define como coordenadas los parámetros necesarios para determinar la posición y la velocidad de ésta en cada momento. Mientras que, se definen como grados de libertad a las coordenadas que resultan independientes tras aplicar las condiciones de vínculos entre ellas. A partir de los grados de libertad (gl) y las propiedades de las estructura (tales como la masa y la rigidez), se pueden encontrar sistemas equivalentes como el de la Figura 1, cuyo movimiento queda descrito por la ecuación (1).

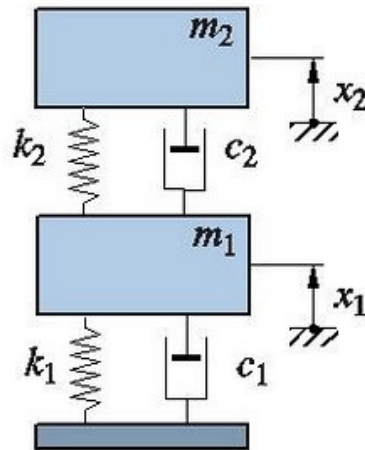


Figura 1: Modelo de sistema de 2 grados de libertad.

$$[M]\{\ddot{x}\} + [C]\{\dot{x}\} + [K]\{x\} = \{F\} \quad (1)$$

Donde K , M y C son las matrices de rigidez, masa y amortiguamiento respectivamente, F es un vector de fuerzas externas y $\{\ddot{x}\}$, $\{\dot{x}\}$, $\{x\}$ son vectores con las aceleraciones, velocidades y desplazamientos de los grados de libertad. En las estructuras estáticas, por lo general, no existe amortiguamiento o es muy pequeño, por lo que se puede despreciar su influencia en la respuesta dinámica de la estructura, con lo que la ecuación (1) queda reducida a:

$$[M]\{\ddot{x}\} + [K]\{x\} = \{F\} \quad (2)$$

Las soluciones de la ecuación diferencial (2), homogénea y particular, representan la respuesta vibratoria transiente y estacionaria del sistema, respectivamente. Para la solución homogénea de la ecuación (2) se puede suponer una solución periódica.

$$\{x\} = \{A\} \sin(\omega t) \Rightarrow \{\ddot{x}\} = -\omega^2 \{A\} \sin(\omega t) \quad (3)$$

Donde las componentes del vector $\{A\}$ representan la amplitud vibratoria de cada grado de libertad y ω representa las frecuencia naturales o resonantes del sistema. Remplazando (3) en la solución homogénea de (2) y reordenado se obtiene la siguiente relación.

$$(-\omega^2[M] + [K])\{x\} = 0 \quad (4)$$

Se puede observar que la ecuación (4) corresponde a un problema de valores y vectores propios de la matriz $[M]^{-1}[K]$, en el cual la matriz que acompaña al vector desplazamiento es conocida como rigidez dinámica.

Para la solución del problema estacionario de la ecuación (2) se define la Función Respuesta en Frecuencia (FRF) como el cociente entre los vectores desplazamiento y fuerza en el dominio de frecuencias. Calculando la transformada de Laplace de la ecuación (2) y llevándola al dominio de frecuencias ($p=j\omega$), se obtiene:

$$[FRF(\omega)] = \frac{\{x(\omega)\}}{\{F(\omega)\}} = ([K] - \omega^2[M])^{-1} = \frac{adj([K] - \omega^2[M])}{det([K] - \omega^2[M])} \quad (5)$$

Donde $adj(\cdot)$ y $det(\cdot)$ definen a los operadores matriciales adjunto y determinante. De la ecuación (5) se puede obtener la amplitud de la respuesta normalizada según la fuerza en función de ω , en una matriz de $N \times N$ (N = numero de grados de libertad del sistema), donde $FRF(\omega)_{ij}$ corresponde a la función de respuesta medida en i al excitar la estructura en j , cabe notar que la matriz es simétrica. Las funciones de respuesta en frecuencia obtenidas cuando i y j son iguales, es decir, el punto de medición de la respuesta es el mismo que el punto de excitación, se conocen como FRF directas (Figura 2), mientras que si estos puntos difieren se les denomina FRF indirectas (Figura 3). En ambas figuras aparecen las frecuencias resonantes, que son los puntos en que la amplitud de la respuesta vibratoria se maximiza y tiende a infinito para el caso sin amortiguamiento, se puede observar que estas frecuencias se mantiene fijas en ambos tipos de FRF. Por otro lado, se pueden definir las frecuencias antiresonantes como los puntos en que la respuesta vibratoria tiende a cero, a diferencia del caso anterior estas cambian su cantidad y valor en las distintas FRF según la posición y distancia que separan al punto de medición del punto de excitación de la estructura. Analíticamente, las frecuencias resonantes hacen cero el denominador de la ecuación 5, lo que maximiza su valor, lo que se determina resolviendo que el determinante de la matriz de rigidez dinámica sea igual a cero. Por otro lado, las frecuencias antiresonantes se pueden determinar resolviendo que el operador $adj(\cdot)$ de la matriz de rigidez dinámica sea cero, el operador $adj(\cdot)$ representa la matriz adjunta. Experimentalmente, se puede obtener la FRF de una estructura para determinar ambas frecuencias características, buscando los mínimos y máximos locales de la función. Además, estos máximos y

mínimos están acompañados por un cambio abrupto en la fase de la FRF, en dirección de 180° para las frecuencias antirresonantes y de -180° para las resonantes.

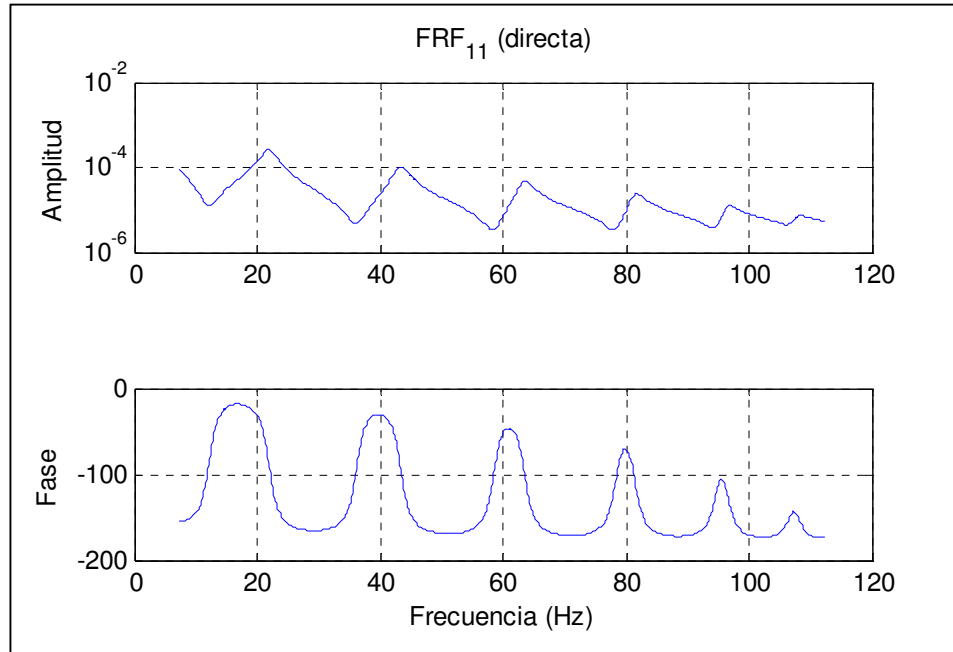


Figura 2: Grafico tipo de FRF directa (8 gl).

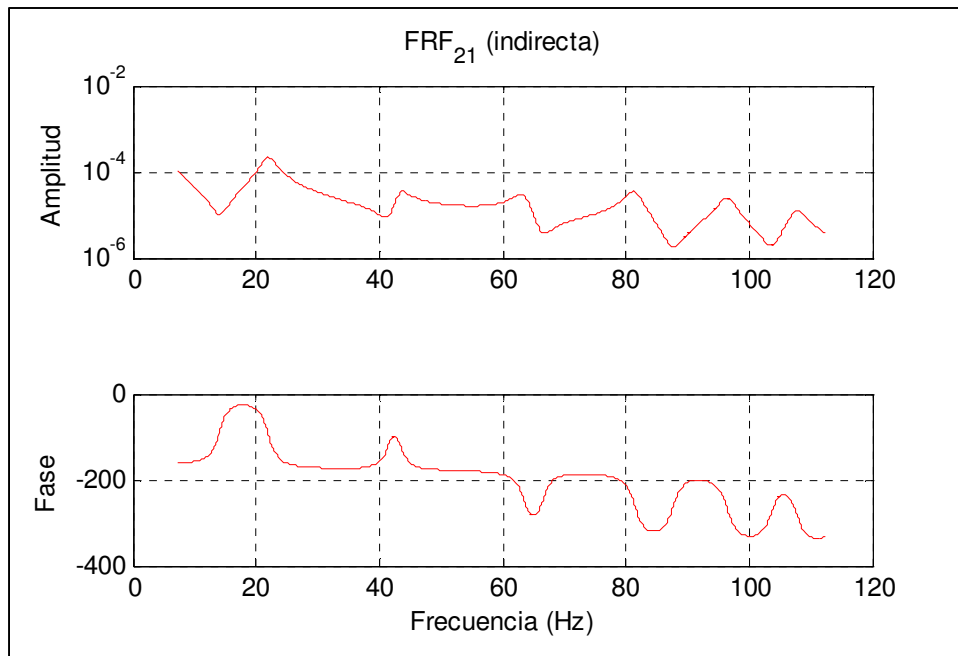


Figura 3: Grafico tipo de FRF indirecta (8 gl).

2.2 Identificación de Daños

La influencia del daño en la estructura, generalmente causado por una grieta, esto se puede definir analíticamente como una reducción local de rigidez o mediante el análisis de un modelo en 2 o 3 dimensiones [1].

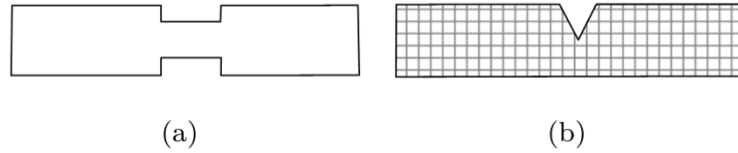


Figura 4: Esquema de a) Reducción local de rigidez. b) Modelo de grieta.

La reducción de rigidez es la manera más fácil de modelar una estructura con daños, mediante la introducción del factor de rigidez efectiva (β). Este representa la proporción efectiva de la rigidez en comparación a la rigidez original, afectando la contribución de las rigideces locales en la matriz de rigidez global (6).

$$[K] = \sum \beta_i [K_i] \quad (6)$$

Donde $[K_i]$ es la matriz local de rigidez, en un sistema de masa-resorte es la constante del i -ésimo resorte. También, se define una constante complementaria a la rigidez efectiva, el factor de reducción de rigidez (definido como $1-\beta_i$) que, en este trabajo solo se usa para presentar de manera más clara algunos resultados. Por otro lado, se pueden obtener mejores resultados al modelar las zonas cercanas a la grieta (Figura 4b). Sin embargo, estos modelos son difíciles de aplicar en detección de daño, ya que requieren muchos grados de libertad y la malla debe ser modificada cada vez que se cambia la ubicación o el tamaño de la grieta.

2.3 Nociones de Elementos Finitos para Vibraciones Mecánicas

El método de elementos finitos aplicado a problemas de vibraciones se puede entender como una extensión del método matricial para varios grados de libertad, mostrado en la sección 2.1. Con este método, un sistema continuo puede ser discretizado en elementos de menor tamaño, de una forma que sea representativa de la geometría del sistema original o sirva para analizar los fenómenos de estudio (Figura 5).

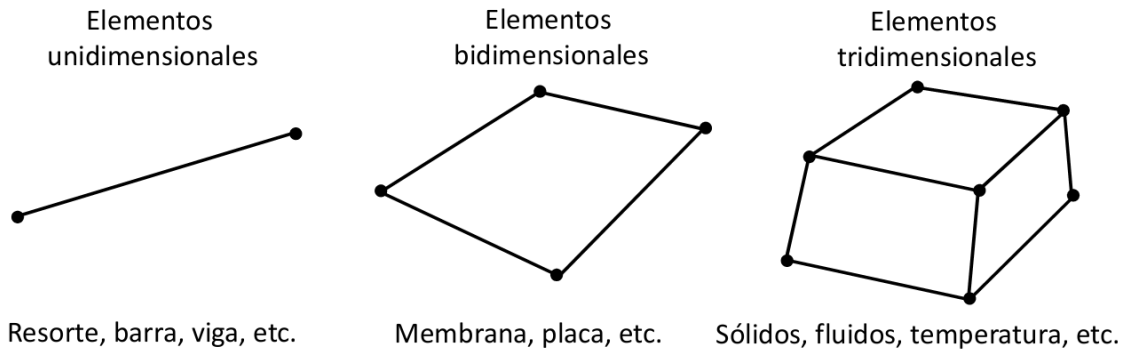


Figura 5: Tipo de elementos finitos.

Para poder estimar la respuesta dinámica de una viga en flexión, esta puede ser discretizada en elementos unidimensionales de viga (Figura 6), cuyas coordenadas impares corresponden al desplazamiento vertical y sus coordenadas pares corresponden a la rotación de cada nodo, respectivamente. Cada elemento de viga cumple que su ecuación de movimiento estacionaria satisface la relación 7, con las condiciones de borde de sus coordenadas nodales (8), con esto se logra una descripción aproximada del movimiento de la viga [6].



Figura 6: Elemento tipo viga (en flexión).

$$EI \frac{\partial^4 y(x, t)}{\partial x^4} = 0 \quad (7)$$

$$y(0, t) = v_i \quad y(l, t) = v_j \quad \frac{\partial y}{\partial x}(0, t) = \theta_i \quad \frac{\partial y}{\partial x}(l, t) = \theta_j \quad (8)$$

Para calcular la matriz de rigidez del elemento de viga, se calcula su energía potencial elástica a partir de su movimiento aproximado y se reordena matricialmente en función de sus coordenadas nodales (9 y 10).

$$V(t) = \frac{1}{2} EI \int_0^l \left(\frac{\partial^2 y(x, t)}{\partial x^2} \right)^2 dx = \frac{1}{2} y^T K y \quad (9)$$

$$[K_i] = \frac{EI}{L^3} \begin{bmatrix} 12 & 6L & -12 & 6L \\ 6L & 4L^2 & -6L & 2L^2 \\ -12 & -6L & 12 & -6L \\ 6L & 2L^2 & -6L & 4L^2 \end{bmatrix} \quad (10)$$

Para calcular la matriz de masa del elemento se procede de igual manera con la energía cinética de la viga (11 y 12).

$$T(t) = \frac{1}{2} \rho A \int_0^l \left(\frac{\partial y(x, t)}{\partial x} \right)^2 dx = \frac{1}{2} \dot{y}^T M \dot{y} \quad (11)$$

$$[M_i] = \frac{\rho A}{420} \begin{bmatrix} 156 & 22L & 54 & -13L \\ 22L & 4L^2 & 13L & -3L^2 \\ 54 & 13L & 156 & -22L \\ -13L & -3L^2 & -22L & 4L^2 \end{bmatrix} \quad (12)$$

Dónde:

- E = Módulo de Young
- I = Momento de Inercia del eje en flexión
- L = Largo del elemento
- ρ = Densidad
- A = Área de la sección transversal

Para caracterizar el comportamiento de todo el sistema (viga), se aplican las relaciones existentes entre las coordenadas nodales de cada elemento y las coordenadas generales de viga. Con esto se ensamblan las matrices de rigidez y masa de cada elemento en una matriz global superponiendo las coordenadas que comparten cada elemento según el sistema de coordenadas globales del sistema (Figura 6) [7].

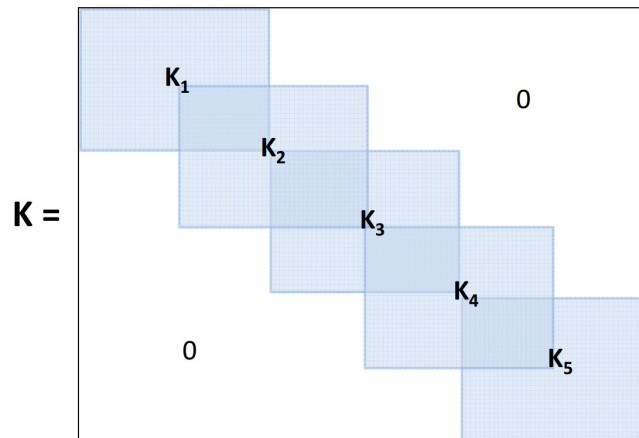


Figura 7: Ensamble matriz de rigidez para una viga de 5 elementos.

2.4 Medición Experimental

Para determinar la Función Respuesta en Frecuencia de una estructura real, se miden las señales de excitación y respuesta en el tiempo [7]. Sean $X(f)$ y $F(f)$ los espectros en frecuencia de las señales respuesta y excitación, respectivamente, calculados de las señales medidas. Con esta información, se puede definir $H(f)$ (FRF) como la función de respuesta entre ambas señales (12).

$$H(f) = \frac{X(f)}{F(f)} \quad (12)$$

Esta definición puede llevar errores, dada la posible existencia de valores cero en el espectro de la señal de excitación. Para evitar esto se utilizan dos alternativas para calcular $H(f)$, utilizando los valores conjugados de las señales (13 y 14).

$$H_1(f) = \frac{X(f)F^*(f)}{F(f)F^*(f)} = \frac{G_{XF}}{G_{FF}} \quad (13)$$

$$H_2(f) = \frac{X(f)X^*(f)}{F(f)X^*(f)} = \frac{G_{XX}}{G_{FX}} \quad (14)$$

Normalmente, los valores de G se obtienen del promedio de varias mediciones de las señales de excitación y respuesta de la estructura. Además, se define la coherencia entre las aproximaciones de H (f) (15), que tiene valores entre 0 y 1. Se obtiene 1 cuando existe una correlación perfectamente lineal entre la entrada y la salida de la función de transferencia.

$$\gamma^2 = \frac{\|G_{FX}\|^2}{G_{FF}G_{XX}} = \frac{H_1(f)}{H_2(f)} \quad (15)$$

2.5 Ajuste de Modelos

El ajuste de modelos es un método que permite verificar y ajustar los modelos analíticos con las estructuras o modelos experimentales [7]. Para asegurar que el ajuste del modelo analítico sea representativo del sistema mecánico que se está estudiando se requieren parámetros para correlacionar ambas respuestas dinámicas. En problemas de vibraciones mecánicas es normal que se ocupen las diferencias entre las frecuencias características o la comparación (visual o analítica) de los modos de vibrar.

Como se señaló en la sección 2.1, las frecuencias características del sistema dependen globalmente de su masa y rigidez (en particular, de su densidad, módulo de Young y otras propiedades geométricas). Se define λ_j (con j entre 1 y el total de variables a ajustar) como los parámetros de ajuste del modelo. Estos parámetros son ponderadores de las propiedades originales y se ocupan como variable de optimización, para plantear un problema de optimización no lineal que minimice (16) la diferencia entre las frecuencias características, analíticas y experimentales, en este caso las frecuencias antiresonantes del sistema [8].

$$\min \sum_n \sum_i \left\| \frac{\omega_{r,i,n}^A(\lambda_1, \lambda_2, \dots, \lambda_j)^2}{\omega_{r,i,n}^E} - 1 \right\| \text{ sujeto a } l_j \leq \lambda_j \leq s_j \quad (16)$$

Donde $\omega_{r,i,n}$ es la *i-esima* frecuencia antiresonante para el *n-esimo* grado de libertad, mientras que el superíndice A y E hace la diferencia entre los datos analíticos y los experimentales. Para acotar el rango de soluciones se tiene los límites inferiores (l_j) y superiores (s_j) para cada parámetro de ajuste. Con esto, se evita tener múltiples soluciones que podrían corresponder a otros problemas.

Cabe señalar que, para minimizar la diferencia entre las antiresonancias analíticas y experimentales se utiliza un método ajuste de modelos basado en algoritmos de optimización no lineal, pero su implementación no es tema de discusión en este trabajo. Para problemas de mayor complejidad se ha propuesto implementar algoritmos genéticos como método de optimización [8].

2.6 Red Neuronal Artificial (RNA)

A pesar de que, actualmente, los avances en la computación permiten procesar datos de forma fluida y/o en grandes cantidades, aún existen problemas que el cerebro humano puede resolver de manera mucho más rápida y eficaz que un computador, por ejemplo, el reconocimiento de patrones. Esto se debe a que el cerebro es capaz de filtrar una gran capacidad de información al mismo tiempo, analizarla de acuerdo a la experiencia y aprender de ella, mientras que el computador funciona de una manera secuencial. Las Redes Neuronales Artificiales (RNA) intentan imitar esta capacidad, emulando los elementos constitutivos de una red neuronal: las neuronas y la sinapsis. Las neuronas son el elemento básico de una red neuronal, son células altamente interconectadas, que al ser excitadas por uno o más impulsos sobre cierto nivel transmiten un impulso nervioso “aguas abajo”. Mientras que la sinapsis es la unión entre una neurona y otra, que determina el tipo de comunicación que hay entre ellas, tanto en velocidad, niveles de saturación y cantidad de información comunicada.

Las Redes Neuronales tienen el potencial de implementar funciones complejas. Dependiendo de si el tamaño de la red (número de interconexiones), la estructura y los pesos de la red son adecuados, esta debería ser capaz de resolver problemas [9] en distintas disciplinas con el nivel de precisión que se desea. En este trabajo son de particular interés las RNA utilizadas para reconocer patrones o para ajustar curvas, como puede ser la configuración de rigideces de un sistema.

2.7 Diseño y Entrenamiento de RNA

En una RNA se tiene la neurona como unidad básica de procesamiento, que está representada por un nodo en una estructura de grafo (Figura 8), y las sinapsis, que se representan como el peso de las conexiones. Cada nodo j cuenta con R entradas p_i , cada una de ellas ponderadas por un peso w_{ij} antes de entrar al nodo. En el nodo, se suman todas las entradas ponderadas junto a un factor de corrección denominado bias (b) (17). Al activarse la neurona, la señal n pasa por una función de transferencia ($f(n)$) antes de convertirse en la salida de la neurona o_j . Cuando se tiene más de una capa de neuronas, se considera un superíndice l a cada variable, que representa la capa al que corresponde dicho valor.

$$n = \sum_{i=1}^R (w_i p_i + b_j) \quad (17)$$

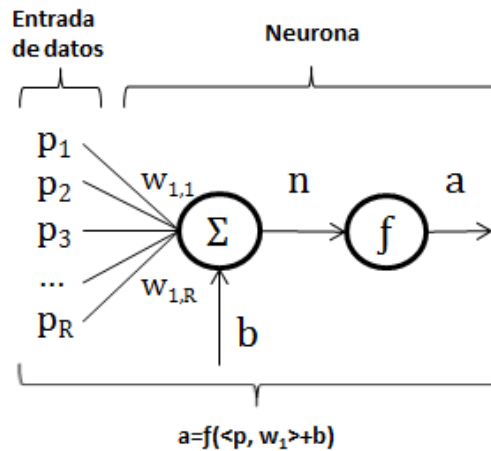


Figura 8: Esquema básico de una neurona.

$$O_j = f(n) \quad (18)$$

Dado que el comportamiento de la RNA está muy influenciado por el resultado de la función de propagación, entradas y pesos (17), el objetivo de la función de transferencia (18) es acotar los valores de salida de la neurona, lo que da un mayor control sobre la RNA. El tipo de función ocupada depende del tipo de red a entrenar. En la figura 9 se presentan algunos tipos de funciones, siendo la más utilizada la función sigmoidea tangente (Figura 9b).

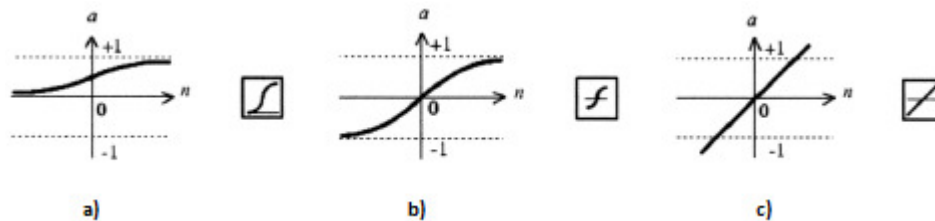


Figura 9: Funciones de salida. a) Sigmoidea logarítmica. b) Sigmoidea tangente. c) Identidad.

Para definir una RNA como un conjunto organizado de neuronas, ésta y su funcionamiento quedan caracterizados por su arquitectura o topología y por su modo de aprendizaje o entrenamiento.

La arquitectura o topología de la red, es la forma en que están dispuestas y conectadas las neuronas en la red. Las principales variables de la arquitectura de una red son:

- Número de capas: Se puede distinguir entre redes mono capa (Figura 11) donde todas las neuronas se encuentran en un mismo nivel y redes multicapas (Figura 10). En general, las redes multicapa permiten mas conexiones y de mayor complejidad.
- Número de neuronas por capa: entre mayor número de neuronas haya en una capa, mayor cantidad de conexiones pueden existir.

- Tipo de conexiones: En general, las neuronas están conectadas y envían información hacia la capa siguiente o posterior (Figura 10), pero también pueden haber conexiones laterales en la misma capa o conexiones recursivas a la misma neurona (Figura 11) o a capas anteriores, como es el caso de la redes recursivas [9].

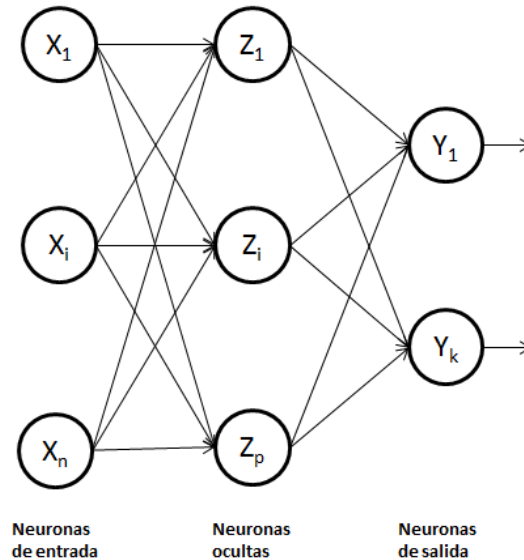


Figura 10: Esquema red Perceptrón multicapa.

- Nivel de interconexión: La red puede estar totalmente interconectada (figura 10 y 11) o pueden haber neuronas o capas que no se conectan, principalmente para realizar distintos procesos con los datos. También hay redes en que existen conexiones entre capas no vecinas, como las de tipo cascada en que la tercera capa no solo recibe datos de la segunda capa sino que también lo hace de la primera.

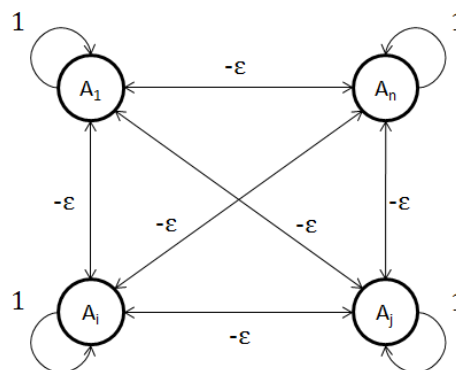


Figura 11: Esquema red competitiva Maxnet.

Los métodos de entrenamiento para RNA pueden ser clasificados en dos grandes grupos: aprendizaje supervisado y aprendizaje no supervisado (Tabla 1). Por un lado, los métodos de aprendizaje supervisado implican un entrenamiento controlado,

donde se conoce la salida deseada para cada grupo de entradas y esta es comparada con la salida de la red, realizando un proceso iterativo para modificar los pesos de la red, si es que existen diferencias significativas. Mientras que en el aprendizaje no supervisado la red no recibe ninguna información respecto a si la salida generada es o no correcta para modificar los pesos durante el entrenamiento, sino que, se auto organiza para agrupar y ajustar los datos.

Tabla 1: Clasificación de métodos de entrenamiento.

Métodos supervisados	
Entrenamiento por corrección de error	Se ajustan los pesos de las conexiones en función de las diferencias entre los valores deseados y los obtenidos en la salida de la red.
Entrenamiento por refuerzo	A diferencia del mecanismo anterior, no se cuenta con una referencia exacta respecto de la salida deseada, sino que, solamente con una señal de refuerzo que indica éxito o fracaso.
Entrenamiento estocástico	Se realizan cambios aleatorios en los valores de los pesos de las conexiones, evaluando su efecto a partir del resultado deseado y de distribuciones de probabilidad.
Métodos no supervisados	
Entrenamiento hebbiano	Se ajustan los pesos de acuerdo con la correlación de los valores de las salidas de las neuronas conectadas.
Entrenamiento competitivo o cooperativo	Se ajustan los pesos de tal modo que, cuando la red reciba cierta información de entrada sólo algunas neuronas se activen.

En ambas categorías se puede reconocer la utilización de métodos de entrenamiento fuera de línea, donde todos los ajustes de la red se realizan antes que ésta entre en funcionamiento. Mientras que existen métodos en línea donde los pesos de las conexiones de la red se van ajustando cada vez que la red se enfrenta a información nueva.

La duración del entrenamiento de la red es muy importante, ya que si es demasiado extenso se produce un sobre ajuste de la red y esta pierde la capacidad de generalizar nuevos datos. Como se desconoce a priori la duración del entrenamiento, normalmente se ocupan tres maneras para acortarla y mejorar el desempeño de la red. La primera de ellas se basa en el error objetivo, es decir, la red finaliza su entrenamiento cuando el error total de la red alcanza un valor o la diferencia entre una iteración y otra es menor a cierto valor objetivo. También se puede limitar el número de iteraciones del entrenamiento, considerando como una iteración cada vez que la red

revisa todos los patrones de entrenamiento. Finalmente, se ocupa una técnica llamada early stopping (parada prematura) en que además del set de datos de entrenamiento, existe un set de datos de validación que no se ocupa para actualizar los pesos durante el entrenamiento, sino que, se utiliza como condición para terminar el entrenamiento, cuando el error en este set alcanza el mínimo.

Algunos autores [10,11] han verificado que al agregar una tasa de ruido a los valores de entrada de la red, se consigue mejorar la capacidad de generalización de la red al enfrentar información nueva. Esto se conoce como aprendizaje con introducción de ruido (Noise Injection Learning, en inglés). Este método también es muy útil para casos en que los datos de entrada de la red tienen ruido experimental, debido a su mayor capacidad de generalización. Este efecto se puede lograr sumando un vector de valores aleatorio acotados, por la tasa de ruido deseada, a las entradas de la red antes de empezar el entrenamiento de esta.

2.8 Retro-propagación y otros Métodos de Entrenamiento

Para reconocimiento de patrones y ajuste de curvas son ampliamente usadas las RNA del tipo Perceptrón Multicapa (Multilayer Perceptron, MLP) (Figura 10), que son redes de una o más capas ocultas, totalmente conectadas hacia adelante y sin conexiones entre las capas no vecinas. Estas pueden ser entrenadas mediante retropropagación (backpropagation), a partir de un conjunto de datos representativos del patrón o curva a identificar. Se trata de un método de aprendizaje supervisado por corrección de error, en que los pesos de la red son totalmente ajustados antes de poner en funcionamiento la red. Este método se puede ilustrar de la siguiente manera [12], considerando el error total de una RNA de J salidas sin capas ocultas, con G patrones de entrenamiento.

$$E = \sum_{g=1}^G E_g = \sum_{g=1}^G \left(\frac{1}{2} \sum_{j=1}^J [t_j(g) - o_j(g)]^2 \right) \quad (19)$$

Donde t_j y o_j , son, respectivamente, el j -ésimo valor objetivo y de salida de la red para un patrón de entrada g dado. El objetivo del método será ajustar las ponderaciones de cada conexión neuronal para minimizar el error. Para esto, se define la dependencia del error (E_g) con el peso de acuerdo a la siguiente relación:

$$\Delta w_{ij} \approx -\eta \frac{\partial E_g}{\partial w_{ij}} \quad (20)$$

$$-\frac{\partial E_g}{\partial w_{ij}} = \frac{\partial E_g}{\partial O_j(g)} \frac{\partial O_j}{\partial w_{ij}} = \sum_{j=1}^J [t_j(g) - O_j(g)] \frac{\partial O_j(g)}{\partial w_{ij}} \quad (21)$$

De este modo, se genera una relación entre la ponderación (w_{ij}) y el ajuste de sí misma (21), que depende de tasa de aprendizaje (η) y de la función de transferencia de

la RNA. Así, se genera un método iterativo para reducir el error de la RNA. Para una red con capas ocultas se procede de la misma manera aplicando la regla de la cadena al calcular la derivada. Por lo tanto entre más capas tenga la red mayor complejidad va a tener este método y, por consiguiente, se va a necesitar un mayor tiempo de entrenamiento.

El método de backpropagation corresponde a uno de los métodos de descenso de gradiente, los que a pesar de su convergencia suelen ser muy lentos. Debido a ello, se han planteado mejoras como el método de gradientes conjugados escalonados (Scaled Conjugate Gradient, SCG) o alternativas como el método de Levenberg-Marquardt.

El método de gradientes conjugados escalonados (SCG) [11] es una mejora al método tradicional de backpropagation, donde la regla de actualización para el vector de los ponderadores de la red queda dada por la ecuación 22.

$$w_{t+1} = w_t + \alpha_t d_t \quad (22)$$

Donde α_t (23) es el largo óptimo de avance y d_t es la dirección de avance que debe cumplir con la relación 24. En ambas relaciones, g_t es el gradiente y H_t es el hessiano de la función error con respecto a los ponderadores y bias de la red, evaluados en la iteración t .

$$\alpha_t = -\frac{d_t^T g_t}{d_t^T H_t d_t} \quad (23)$$

$$d_{t+1}^T H_{t+1} d_t = 0 \quad \text{con } d_1 = -g_1 \quad (24)$$

Dada la relación 24, se eliminan las componentes de la dirección que sean paralelas a la dirección de avance anterior, con lo que se evita que haya “serpenteos” en el camino de optimización. Dado que no todas las funciones de error son cuadráticas, se puede dar que el hessiano no sea siempre definido positivo, por lo que el sentido de avance se invierte en la relación 23, lo que a su vez, causa un aumento en el error. Para evitar esto se introduce un factor de escalonamiento, λ_t (25), que asegura que el denominador de la nueva ecuación para α_t (26) sea positivo y se mantenga el sentido de avance del algoritmo.

$$H^* = H + \lambda_t I \quad (25)$$

$$\alpha_j = -\frac{d_t^T g_t}{d_t^T H_t d_t + \lambda_t \|d_t\|^2} \quad (26)$$

Con esto se esta suponiendo cierto comportamiento cuadrático del error; si esta suposición es correcta el valor de λ_t debe ser pequeña, mientras que si es una

suposición débil el valor de λ_t debe ser mayor. Para controlar esto, se introduce un parámetro de comparación (27), donde E_c es la aproximación cuadrática del error local.

$$\Delta_t = \frac{E(w_t) - E(w_t + \alpha_t d_t)}{E(w_t) - E_c(w_t + \alpha_t d_t)} \quad (27)$$

Si Δ_t es cercano a 1 la suposición es buena, por lo que se ocupa este valor para ir ajustando λ . Si $\Delta_t > 0.75$ se ajusta $\lambda_{t+1} = \lambda_t/2$ y si $\Delta_t < 0.25$ se ajusta $\lambda_{t+1} = \lambda_t * 4$, Δ_t tiene un valor intermedio se mantiene el valor de λ para la siguiente iteración. En el caso en que Δ_t sea menor que cero, se actualiza el valor de λ pero no se actualizan los ponderadores de la red ya que su nuevo valor aumentaba el error de la red.

Por otro lado el método de Levenberg-Marquardt [11] ha sido pensado para optimizar específicamente la suma de los error cuadráticos (19), la que puede ser re escrita de la siguiente manera (28).

$$E = \sum_{g=1}^G \left(\frac{1}{2} \sum_{j=1}^J [T_j(g) - O_j(g)]^2 \right) = \frac{1}{2} \|\varepsilon_G\|^2 \quad (28)$$

Donde ε_G es el vector que contiene el error de cada uno de los G patrones de entrenamiento. Si se considera un paso pequeño al actualizar las ponderaciones de la red se puede aproximar el error usando una serie de Taylor de primer orden.

$$\varepsilon(w_{t+1}) = \varepsilon(w_t) + Z(w_{t+1} - w_t) \quad (29)$$

En la ecuación 29 el termino Z es una matriz que contiene la derivada de cada uno de los G patrones con respecto a cada ponderador y bia de la red, mientras que w es el vector que contiene los ponderadores y bias de la red. Con esto se puede re- escribir la ecuación 28 y minimizarla con respecto a w_{t+1} , con lo que se consigue la siguiente relación (30).

$$w_{t+1} = w_t - (Z^T Z)^{-1} Z^T \varepsilon(w_t) \quad (30)$$

Con esto se tiene un proceso iterativo para la optimización de los ponderadores de la red. Sin embargo, en algunos casos el cambio el ponderadores puede lo suficientemente grande para que no se cumpla la validez de la relación 29 (Aproximación de primer orden). Para poder controlar esto se agrega un termino λ por la matriz identidad, de manera muy similar al método de SCG, por lo que la regla de actualización queda finalmente como se muestra en la ecuación 31.

$$w_{t+1} = w_t - (Z^T Z + \lambda I)^{-1} Z^T \varepsilon(w_t) \quad (31)$$

En este caso el valor de λ se reduce de manera constante cada vez que se realiza una iteración exitosa, o sea en que reduzca el error de la red. Por ejemplo, se reduce a la mitad en cada iteración exitosa ($\lambda_{new} = \lambda_{old}/2$). Hay que notar que el gradiente

del error de la red se puede aproximar como $Z^T \varepsilon(w_t)$, por lo que para valores muy grandes de λ se tiene una regla de actualización muy similar a un método descenso de gradiente (20). Esto es una ventaja que tiene el método de Levenberg-Marquardt, aprovecha la buena convergencia de los métodos de descenso de gradiente, pero mejora su eficiencia al aumentar la velocidad y variar la dirección de algoritmo. También se puede notar que en cada iteración del método hay que invertir una matriz de $W \times W$ celdas y multiplicarla con otra de $G \times W$ ($W = N^e$ de ponderadores de la red), esto se puede volver muy costoso computacionalmente para redes de gran tamaño o números patrones de entrenamiento.

Esta sección es una muestra referencial de algunos métodos de entrenamiento que se consideraron para este trabajo, para mas detalle sobre el tema se pueden consultar las referencias de este trabajo ya citadas.

2.9 Aplicaciones de RNA en ingeniería

En ingeniería se está acostumbrado al uso de métodos basados en modelos explícitos o simbólicos, como el presentado en este trabajo para determinar analíticamente la respuesta vibratoria de una estructura o problemas de transferencia de calor, solidos, fluidos, entre otros. Estos modelos requieren de una sólida base conceptual para garantizar sus buenos resultados, e incluso, están influenciados por el criterio profesional. Es el caso de los modelos que tienen comportamientos discretos como la resistencia a la fatiga en diseño mecánico, donde se ajustan o eligen ciertas constantes según las condiciones del problema. En otros casos, los modelos abarcan una gran cantidad de datos, como es el caso de los modelos de elementos finitos y se requiere un gran gasto computacional para llegar a resultados exactos.

Frente a esto, las RNA presentan 2 ventajas importantes:

- El desarrollo de la RNA es la etapa que concentra mayoritariamente la necesidad de un conocimiento avanzado del problema de estudio, para definir qué parámetros están correlacionados con otros y evaluar si la RNA entrega resultados válidos. Hace que se liberen recursos de la obtención de resultados y los lleva al análisis de los resultados obtenidos.
- Debido a su construcción, requieren muy pocos recursos computacionales para su utilización final, por lo que una RNA bien diseñada podría trabajar en situación de análisis de datos en tiempo real.

Dado lo anterior, las RNA tienen un gran atractivo para su aplicación en problemas de ingeniería, en especial, en procesos de modelación inversa o monitoreo de condiciones, donde a pesar de tener grandes conocimiento del fenómeno, aún no se tienen modelos confiables o estos requieren una gran cantidad de tiempo lo que evita que se puedan tomar medidas oportunas. En el caso particular de las RNA de detección de daños se tiene una estructura a la cual se estudia su comportamiento vibratorio u otro característico. Luego, se seleccionan los parámetros relevantes (tales

como amplitudes, modos de vibrar o frecuencias) que sirvan como entrada para la red y ésta entrega información, como por ejemplo, los factores de rigidez efectiva para cada elemento en la estructura, tal como se ilustra en la figura 12.

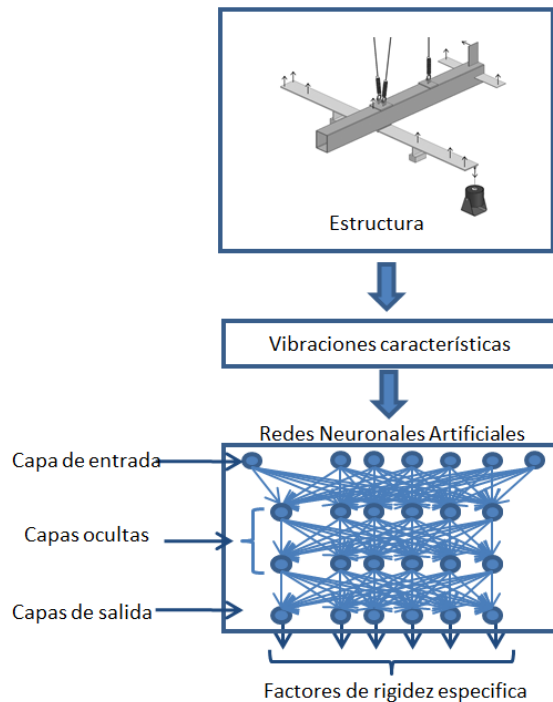


Figura 12: Esquema del problema de detección de daños con RNA.

Fang, Luo y Tang plantean [3] la posibilidad de detectar daños en una estructura sencilla (viga empotrada) a partir de sus FRFs. Ellos utilizaron la amplitud de las FRFs alrededor de las zonas resonantes (78 puntos) como datos de entrada a una red neuronal, la que se entrenó con el método de retro-propagación (backpropagation). Para esto, los autores generaron una base de datos de 30 casos de daños estructural, variando la localización, severidad y multiplicidad de los daños. Los datos de entrenamiento se obtuvieron de un modelo sencillo de elemento finitos de 20 elementos (Figura 13). Con este modelo se simuló distintas situaciones de daños en los puntos marcados en gris, donde los daños fueron representados por la pérdida de rigidez local en uno o más elementos. Se consiguió detectar daños con una precisión mayor a un 80% en casos distintos a los utilizados para entrenar la RNA.

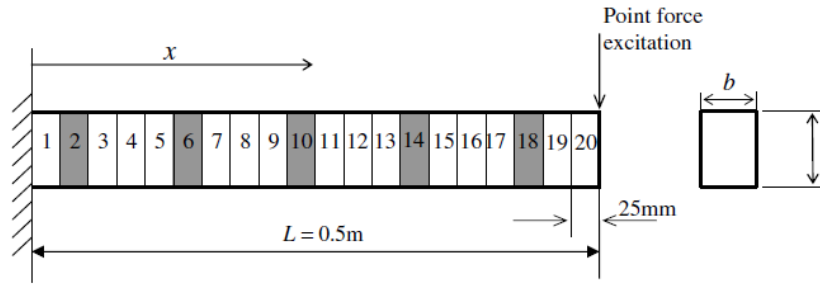


Figura 13: Viga empotrada utilizada por Fang [3].

En este trabajo, se plantea que utilizando exclusivamente frecuencias de antirresonancia, se puede entrenar una RNA para detección de daños. Esto debido a la fuerte dependencia que existe entre estas frecuencias y la rigidez de los problemas a estudiar. Además, son puntos de fácil y precisa identificación dentro de la FRF, lo que minimizaría los errores de arrastre en la entrada de datos a RNA. Con este método, se estudian 2 problemas, un sistema masa-resorte y una viga libre.

- Problema 1: este problema consiste en un sistema de 8 masas alineadas de manera horizontal, unidas por resortes de constantes elásticas iguales entre ellas. Por lo tanto, en este problema la matriz de rigidez $[K]$ va a estar exclusivamente en función de las constantes de cada resorte. El objetivo de estudiar este problema primero es limitar las variables del problema físico, para analizar las dificultades y características que hay que tener en cuenta para el entrenamiento de la RNA. Además, de este problema ya se tienen datos experimentales para la reducción de rigidez en uno de los resortes.
- Problema 2: este problema consiste en una viga libre similar a la utilizada por Fang [3] y se estudian las vibraciones transversales de cada uno de sus elementos. En este problema la matriz de rigidez $[K]$ está expresada en función del módulo de Young y del momento de inercia, propios del material y de la geometría de barra, respectivamente. El objetivo de incluir este caso es hacer el primer acercamiento al estudio de problemas de mayor complejidad. Además, se pueden comparar los resultados obtenidos con los resultados de otros estudios como el de Fang [3].

2.10 Funciones de Error e Indicadores de Desempeño

Para comparar las redes entre si es necesario la utilización de funciones de error e indicadores de desempeño que nos indiquen en que tareas una red es superior a otra [13]. En primera instancias para comparar las distintas redes se ocupa el error cuadrático medio (Mean Square Error, MSE) (32).

$$MSE = \frac{1}{NT} \sum_{N_p} (T_n - Y_n)^2 \quad (32)$$

Donde T_n es el vector objetivo para un patrón, Y_n es el vector de salida de la red para el mismo patrón y NT el número de casos analizados (N° de patrones* N° de salidas). Se ocupa este error ya que es independiente del número de patrones que se ocupe para entrenar o testear la red. Este permite tener una mirada global de la calidad de la regresión o ajuste de curva que esta realizando la red.

De forma adicional, se definen tres indicadores de la calidad con la que detecta daños la red.

- Falsos positivos (33): Cantidad de veces que la red detecta daños en elementos sanos.

$$FP = \frac{1}{NS} \sum_1^{NS} \varepsilon_n^I, \text{ con } \varepsilon_n^I = 1 \text{ si } \|1 - y_n\| \geq c \quad (33)$$

- Falsos negativos (34): El indicador inverso al anterior, cantidad de veces que la red no detecta daños en elementos con perdida de rigidez.

$$FN = \frac{1}{ND} \sum_1^{ND} \varepsilon_n^{II}, \text{ con } \varepsilon_n^{II} = 1 \text{ si } \|1 - y_n\| \leq c \quad (34)$$

- Daños mal cuantificados (35): Cantidad de daños que a pesar de ser detectados, la magnitud entregada por la red difiere con la pérdida de rigidez del elemento por sobre una cota establecida.

$$DMC = \frac{1}{NT} \sum_1^{NT} \varepsilon_n^{III}, \text{ con } \varepsilon_n^{III} = 1 \text{ si } \|t_n - y_n\| \geq c \quad (35)$$

Para todos los indicadores y_n corresponde a una de las salidas de la red y t_n su valor objetivo. Además todos estos indicadores quedan expresados en el porcentaje de errores encontrados dentro de total de detecciones correspondientes, N° de elementos dañados (ND), sanos (NS) o el total analizado (NT), esto para que los resultados sean comparables entre una estructura y otra. Además, todos ellos van a depender de la cota (c) que uno defina para considerar si un elemento tiene un daño real o que diferencia de magnitudes sean importantes para poder tomar decisiones en base a los resultados dados por la red. Por ejemplo si uno pone como cota un 5% de error, todas las detecciones que resulten con 5% de daño o menos se consideran sin daño. Por lo que si es un elemento sin daños se considerara como una “detección” exitosa, mientras que si es un elemento con un daño se considera como un falso negativo. Por esta razón, las redes van a ser probadas con daños mayores a esta cota. Para los indicadores de falsos positivos y negativos se considera un cota del 2.5%, ya que se considera que estos errores son mas relevantes porque ocultan o generan falsas alarmas sobre daños que se están generando. Mientras que se considera una cota del 5% para la

cuantificación de los daños, ya que una vez detectado y localizado el daño (en un caso real) la magnitud de este es una referencia de la severidad más que una medición exacta debido al ruido existente en las mediciones reales.

3. Metodología

Para la realización de este trabajo, se utiliza la siguiente metodología, descrita a continuación (Figura 14).

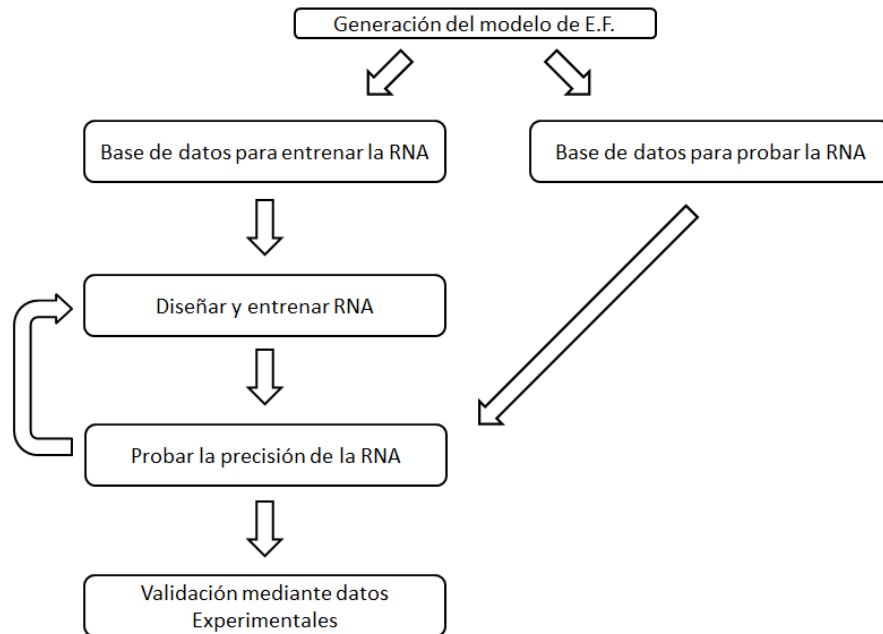


Figura 14: Diagrama de Flujo de la Metodología.

- Generación del modelo de Elemento Finitos: Para cada problema, se van a determinar las frecuencias resonantes y antiresonantes, mediante el uso del método de elementos finitos. Con este modelo se van a generar 2 grupos de resultados analíticos, uno de mayor tamaño para entrenar la RNA y otro con casos específicos para probar la capacidad de la red para resolver el problema planteado.
- Diseñar y entrenar RNA: En esta etapa, se define el tipo de RNA y su método de aprendizaje o entrenamiento, además de su estructura (Nº Neuronas y capas). Esto influye en la precisión de los resultados y en el tiempo de entrenamiento de la RNA. Posteriormente, se procede a entrenar la red con los datos preparados en la etapa anterior.
- Probar la precisión de la RNA: luego de entrenar la RNA esta se prueba con el segundo grupo de datos que son nuevos para la red, de esta manera se evalúa la precisión que tiene la red para resolver el problema. Dependiendo de los resultados, se considera re-entrenar la RNA modificando el número de casos a entrenar, el número de datos de entrada u otra característica que pueda afectar la precisión del resultado.
- Validación mediante datos experimentales: Luego de alcanzar resultados aceptables con datos analíticos, se procede a probar la red con datos

experimentales, los que están sujetos al margen de error propio del equipo de medición y al ruido correspondiente a la cadena de medición. Así, se establece la precisión de la RNA para resolver un problema real, siendo posible comparar los resultados con aquellos obtenidos con otros modelos, para su posterior discusión.

Para uno de los casos de estudio (sistema masa resortes) ya se cuenta con datos experimentales obtenidos por Duffey et al. [14]. Mientras que para el segundo problema (barra libre), se realizaron el montaje y las mediciones experimentales en el laboratorio de sólidos de la universidad.

3.1 Metodología de Entrenamiento de RNA

Para el entrenamiento de la RNA utilizando el toolbox de Redes Neuronales hay que tener presente los siguientes puntos:

- Aleatoriedad: debido a que la inicialización de las redes se realiza de manera aleatoria, el entrenamiento de estas puede estar condicionado por su estructura inicial. Para evitar esto todas las redes son entrenadas 10 veces y se conserva la red que muestre el menor MSE de validación.
- Utilización de patrones de entrenamiento: El toolbox para ciertas operaciones requiere que se divida el total de patrones en tres conjuntos: entrenamiento, validación y testeo. El conjunto de entrenamiento es el que se ocupa directamente en los algoritmos de entrenamiento para optimizar la red. El conjunto de validación es necesario para criterios de terminación como el early stopping y participa de manera indirecta en el entrenamiento de la red. Finalmente el conjunto de testeo se utiliza para probar la performance de la red al enfrentar patrones que no hayan participado ni directa ni indirectamente del entrenamiento. Para este trabajo se define:
 1. Conjunto de entrenamiento: 85%.
 2. Conjunto de validación: 10%.
 3. Conjunto de testeo: 5%.
- Criterio de terminación: las redes se entrenan hasta cumplir con uno de los siguientes criterios:
 1. Early stopping: La red deja de entrenar si el error de validación aumenta 6 veces de manera consecutiva. Se debe notar que se conservan los valores de los ponderadores antes de que el error de validación empezara a aumentar.
 2. N° máximo de iteraciones: 10000 iteraciones

4. Resultados Problema 1

El primer problema de estudio consiste en un sistema masa-resorte con 8 grados de libertad, que permite que las masas se muevan libremente en el eje axial como se ilustra en la figura 15. La primera masa del sistema es ligeramente mayor ya que tiene adherido el dispositivo vibrador utilizado para excitar al sistema en las pruebas.

Datos:

- Masa 1: 559.3 [gr]
- Masas 2 a 8: 419.4 [gr]
- Constantes de los resortes 1 a 7: 56.7 [kN/m]

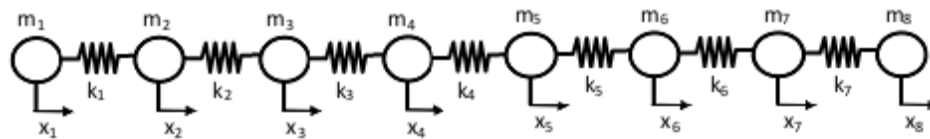


Figura 15: Sistema masa-resorte (8 grados de libertad).

En la figura 16 se puede ver el montaje experimental que se usó en este problema. Este consiste de las 8 masas conectadas por resortes de igual constante elástica, este conjunto está soportado por una barra de acero pulido que permite al conjunto moverse libremente. Además en el extremo derecho del montaje se ubica el vibrador utilizado para excitar la estructura, este está suspendido usando cuerdas elásticas para evitar interferencias en la prueba.

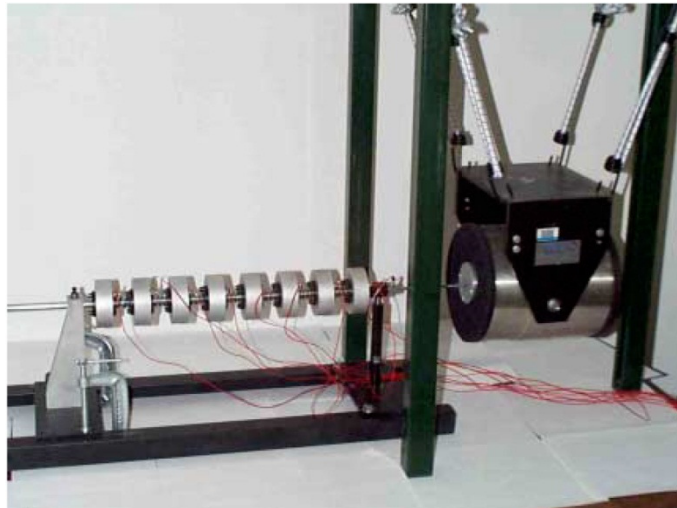


Figura 16: Montaje experimental problema 1 [14].

La resolución de este problema se realiza programando una función en Matlab que recibe los factores de rigidez efectiva para cada resorte y devuelve las frecuencias características del sistema, resonantes y antiresonantes (Anexo A).

4.1 Ajuste de Modelo

El ajuste de modelos del problema se realiza en base a los datos experimentales obtenidos por otros investigadores [14], se cuenta con los datos de FRF directa en la masa 1 ($FRF_{1,1}$), frecuencias resonantes y frecuencias antiresonantes. Para el ajuste de modelo se agregan 15 variables de optimización λ ; 8 para las masas y 7 para las constantes de rigidez de los resortes. Se utiliza la función objetivo presentada en la ecuación (16). Se realiza el ajuste de modelo con la restricción de que los valores de ajuste no varíen más del 10% con respecto de las propiedades iniciales. Los resultados se presentan en la tabla 2.

Tabla 2: Factores de optimización del ajuste de modelo.

Optimización de k_i							Optimización de m_i							
1	2	3	4	5	6	7	1	2	3	4	5	6	7	8
1,001	1,022	1,020	0,971	1,037	1,091	1,055	0,930	0,974	1,005	0,997	1,021	1,015	1,010	1,004

En la tabla 3 se puede ver como las frecuencias antiresonantes se vuelven más ajustadas en todo el recorrido de frecuencia luego de aplicar el algoritmo de ajuste.

Tabla 3: Ajuste de frecuencia antiresonantes.

Frecuencias Antiresonantes (ω_r)	Experimental [Hz]	Analítica inicial [Hz]	Error [%]	Analítica corregida [Hz]	Error [%]
1	12,3	12,2	-0,59%	12,3	-0,43%
2	36,9	36,2	-2,08%	36,7	-0,54%
3	59,6	58,5	-1,83%	59,2	-0,70%
4	80	78,3	-2,06%	79,5	-0,55%
5	96,1	94,7	-1,49%	95,6	-0,52%
6	109,4	106,9	-2,31%	109	-0,40%
7	115,7	114,5	-1,08%	115,5	-0,19%

4.2 Selección RNA

Las redes neuronales tienen muchas variables de diseño; en esta sección se presenta una línea de búsqueda de la configuración de la red que ha mostrado una mayor precisión y los resultados que han marcado el camino a seguir.

Primeramente se debe definir los patrones de entrenamiento, estos deben ser inclusivos en cuanto al tipo de situaciones que se desea detectar, ya que la no linealidad del problema impide que una red entrenada para encontrar un daño pueda generalizar esa información para encontrar daños múltiples. Debido a que en una situación real es posible la aparición de múltiples daños en una estructura, incluso si esta es monitoreada regularmente, es necesario generar patrones que incluyan 2, 3 o 4 daños simultáneos. En este trabajo se generan patrones con presencia de hasta 2 daños simultáneos. Para que los patrones sean representativos de todos los daños

posibles, se generan patrones que consideran a todos los resortes como posibles ubicaciones para el o los daños y estos se distribuyen de manera uniforme en el rango de detección deseada. Es decir si se quieren detectar daños entre 0 y 80%, se tendrán patrones la forma [0 ... 0.8] para cada resorte y la combinación de daños simultáneos deseada. El número de patrones a entrenar va a estar determinado según la regla de la combinatoria a partir de la cantidad de casos de severidad estudiados y la cantidad de daños simultáneos a detectar (36).

$$N^{\circ} \text{ de patrones} = \sum_{i=0}^{N^{\circ} \text{ de daños}} (N^{\circ} \text{ de casos de severidad})^i * \binom{N^{\circ} \text{ de elementos}}{N^{\circ} \text{ de daños simultáneos}} \quad (36)$$

Se generan patrones con 4 tipos de entradas posibles:

- Entrada directa: un conjunto con las antiresonancias del sistema.
- Entrada normalizada (37): un conjunto con las diferencias porcentuales con respecto al caso sin daño. Usar la diferencia porcentual podría limitar las diferencias entre los datos simulados y los datos experimentales, lo que mejoraría el desempeño de la red al enfrentarse a daños en casos reales.

$$x_i = \frac{\omega_{r,i}^D - \omega_{r,i}}{\omega_{r,i}^D} \quad (37)$$

- Sin y con ruido: Además para estos dos casos se entrenan conjuntos de patrones con las frecuencias obtenidas directamente al resolver el problema analíticamente con el modelo ajustado y otro conjunto en que se le agrega una tasa de ruido aleatoria. Esta tasa de ruido aleatoria intenta simular el ruido presente en las mediciones experimentales.

A cada patrón se le asocia una salida objetivo de la red. Esta salida corresponde a los coeficientes de rigidez efectiva para cada resorte o elemento con se obtuvieron las frecuencias antiresonantes usando el modelo analítico del problema. Para realizar las pruebas iniciales, se elige usar patrones de entrada directa con 8 casos de severidad de daño entre 20% y 100% de rigidez efectiva, incluyendo hasta 2 daños simultáneos.

La topología de la red que se elige, corresponde a una red multicapa con una capa oculta, la que se pueda adaptar a la no linealidad del problema. La naturaleza del problema impone que la red debe tener 7 salidas para poder detectar la variación de rigidez en cualquiera de los resortes del sistema. Para las entradas de la red se cuenta con 7 frecuencias antiresonantes que pueden ser identificadas experimentalmente de la función FRF_{11} , por lo que se inicia con 7 entradas de tal manera de determinar si la antiresonancias contienen suficiente información para la detección de daños.

Para la función de transferencia se opta por funciones que incluyan en su recorrido los valores objetivos de la red, los que van entre 0 y 1. En las neuronas de la capa oculta se trabaja con una función sigmoidea tangente, dado que es la que tiene el

mayor dominio y permite acotar de manera suave la salida de las neuronas. Mientras que en las neuronas de la capa de salida se debe elegir la función que permita modelar mejor el comportamiento de los patrones, para esto se puede elegir entre funciones sigmoideas y lineales, las que son ampliamente usadas. Para elegir la función que mejor se adecua al problema, se entrenaron redes con 30 neuronas en la capa oculta, usando el método de Levenberg-Marquardt, ya que debería ser el mas rápido según la documentación del toolbox [15]. Las redes son entrenadas diez veces para cada caso y se conserva la que presenta el menor error de validación, acorde a la metodología del trabajo. En la tabla 4 se puede observar la variación en el error de entrenamiento y validación al utilizar diferentes funciones en la capa de salida.

Tabla 4: MSE v/s Función de transferencia en la capa de salida.

Función de Transferencia	Mean Square Error	
	Error de Entrenamiento	Error de Validación
Tangente Sigmoidea	2,83E-04	3,87E-04
Logarítmica Sigmoidea	1,10E-02	9,14E-03
Identidad	1,20E-03	1,54E-03
Identidad Positiva	1,20E-02	1,01E-02
Identidad Saturada Simétrica	9,72E-05	1,37E-04
Identidad Saturada	1,07E-02	9,73E-03

Se puede ver que la función que mejor minimiza el error es la función identidad saturada simétrica, la que está descrita por la ecuación (38). Contrario a lo que se pudiera pensar, las funciones que tienen el recorrido exacto de la función a aproximar (entre 0 y 1) presentan los peores resultados.

$$f(x) = \begin{cases} -1 & x \leq -1 \\ x & -1 \leq x \leq 1 \\ 1 & 1 \leq x \end{cases} \quad (38)$$

Con este resultado se procede a seleccionar el algoritmo de entrenamiento a utilizar, para esto se compara cinco métodos implementados en Matlab. Estos son; descenso de gradiente, gradiente conjugado, gradiente conjugado escalonado, Quasi-Newton y Levenberg-Marquardt. Para comparar los resultados se entrenan las distintas redes utilizando la misma configuración utilizada en el caso anterior, y ahora se varía el método en cada caso. A continuación (tabla 5), se puede ver como evoluciona el error y el tiempo que toma cada método en entrenar la red, ésto en función de la red que presente menor error de validación, al igual que en caso anterior.

Tabla 5: MSE v/s Algoritmo de entrenamiento.

Algoritmo de Entrenamiento	Mean Square Error		Tiempo de entrenamiento [s]
	Error de Entrenamiento	Error de Validación	
Descenso de Gradiente	2,08E-02	2,06E-02	1745,7*
Gradiente Conjugado	3,68E-02	3,43E-02	1,78
Gradiente Conjugado Escalonado	2,88E-03	3,90E-03	7,92
Quasi-Newton	3,14E-02	3,61E-02	11,09
Levenberg-Marquardt	1,13E-04	1,54E-04	18,17

Se puede notar que el método de Levenberg-Marquardt es el que presenta mejores resultados en cuanto a la minimización del error, incluso un orden de magnitud menos que el resto de los métodos, aunque es un poco más lento que los demás. También hay que notar que el método de descenso de gradiente o backpropagation tradicional es el único que termina por alcanzar el número máximo de iteraciones (100000, utilizado solo en este caso), mientras los demás métodos terminaron por early stopping.

Con estos resultados, se procede a validar la selección del número de casos de severidad a utilizar para generar los patrones de entrenamiento. Los patrones de entrenamiento tienen como rango de salida entre 20% y 100% de rigidez. No se consideraron los casos con rigidez efectiva entre 0% a 20%, ya que en un monitoreo regular de la estructura no deberían aparecer daños de tal magnitud. Además con esto se reduce el número total de patrones, variable que será más relevante en el problema 2. Teniendo esto en cuenta se estudian distintos números de casos de severidad, de tal manera de que todos los grupos de patrones estén uniformemente distribuidos en todo el rango de estudio, lo que permite compararlos objetivamente. Se entrenan redes con la misma configuración que en los casos anteriores, variando el conjunto de patrones utilizados para su entrenamiento. A continuación, se presentan los resultados del número de patrones utilizados y el error resultante para cada caso (tabla 6).

Tabla 6: MSE v/s Nº de Casos de severidad.

Nº de Casos de Severidad	Nº de Patrones	Mean Square Error	
		Error de Entrenamiento	Error de Validación
4	365	1,08E-04	1,82E-03
8	1401	8,14E-05	1,87E-04
16	5489	7,84E-05	9,94E-05

Se puede ver que tanto el error de entrenamiento como el de validación se reduce al aumentar el número de patrones a utilizar, pero al pasar de 8 a 16 casos de severidad la ganancia no es tan grande siendo que se requieren más del triple de patrones de entrenamiento. También se puede notar que al aumentar el número de

patrones disminuye la diferencia entre el error de entrenamiento y de validación, por lo que puede ser una alternativa si se requiriera aumentar la capacidad de generalización de la red.

Al observar los resultados destacados en las tres tablas anteriores, se puede notar que a pesar de que no se consigue el mismo error en los tres casos, estos valores son muy cercanos y estas redes deberían presentar capacidad de detección similares. Por lo que no se considera aumentar el número de veces que cada red es entrenada, como se menciona en la sección 3.1.

4.3 Red Neuronal

Ya determinada la mayor parte de la configuración de la red y los patrones de entrenamiento a utilizar, resta determinar el número de neuronas en la capa oculta. Para esto se realiza un trabajo de inspección aumentando sucesivamente el número de neuronas hasta encontrar el MSE que asegure la detección de daños, cuya rutina en Matlab se incluye en el anexo B. Los resultados se presentan a continuación en la figura 17.

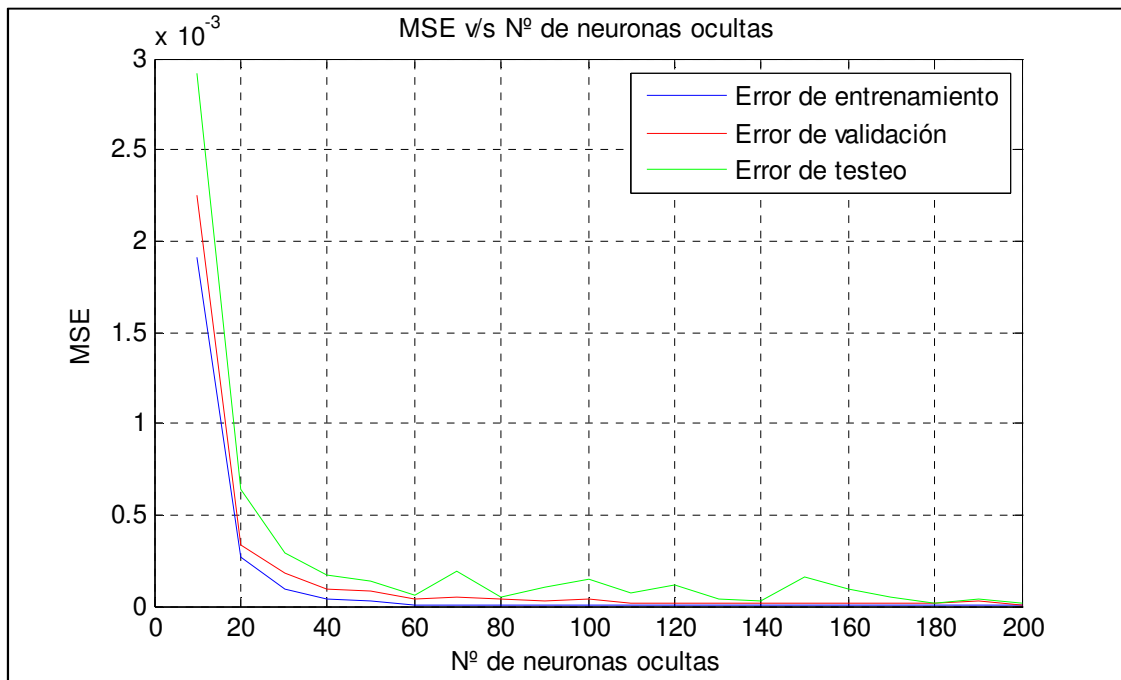


Figura 17: Evolución del error para redes de entrada directa sin ruido.

Se puede notar que el error total de la red en el conjunto de entrenamiento se comporta de manera muy similar al conjunto de validación. Para una red entrenada con patrones de entrada directa se alcanza un mínimo aceptable en el error de la red ($\text{Error}_e = 8.71 \times 10^{-6}$ - $\text{Error}_v = 4.12 \times 10^{-5}$ - $\text{Error}_t = 5.58 \times 10^{-5}$), al utilizar una red con 60 neuronas en la capa oculta. Al entrenar la red con los datos de entrada normalizados se obtiene una evolución de error muy similar a la mostrada en la figura 17 y se alcanzan un error similar a las redes de entrada directa ($\text{Error}_e = 7.89 \times 10^{-6}$ - $\text{Error}_v = 5.24 \times 10^{-5}$ -

$\text{Error}_t=2.96 \times 10^{-5}$), también al utilizar una red con 60 neuronas en la capa oculta. Esto niveles de error de consideran aceptables, ya que para MSE menores a 2×10^{-4} el error promedio por elemento es cercano a 1.0%.

A continuación en la figura 16 se muestra el mismo proceso de entrenamiento pero para patrones que tiene una tasa de ruido aleatorio, se defino esta tasa en $\pm 2\%$ ya que es superior al error aun existente entre los datos reales y los datos simulados (Tabla 3).

En este caso se puede notar que el error de entrenamiento en general es mayor que en los casos sin ruido y que hay una mayor diferencia entre el error de entrenamiento y error de validación. Esto se explica porque el ruido no es constante, por lo que hay un componente de los datos que no puede ser predicha y esto le quita suavidad a la “curva” que se esta ajustando. Se puede ver que para las redes de entrada directa se encuentra un error de validación minino al utilizar 80 neuronas en la capa oculta ($\text{Error}_e = 0.0019$ - $\text{Error}_v=0.0034$ - $\text{Error}_t=0.0028$). Al igual que en el caso anterior, las redes que utilizan datos normalizados se comportan de manera muy similar y alcanzan nivel similares de ajuste ($\text{Error}_e = 0.0018$ - $\text{Error}_v=0.0037$ - $\text{Error}_t=0.0038$). A pesar de que no se alcanzan los niveles de ajuste de la redes sin ruido, esta redes deberían poder detectar daños con una precisión cercana al **5.0%**.

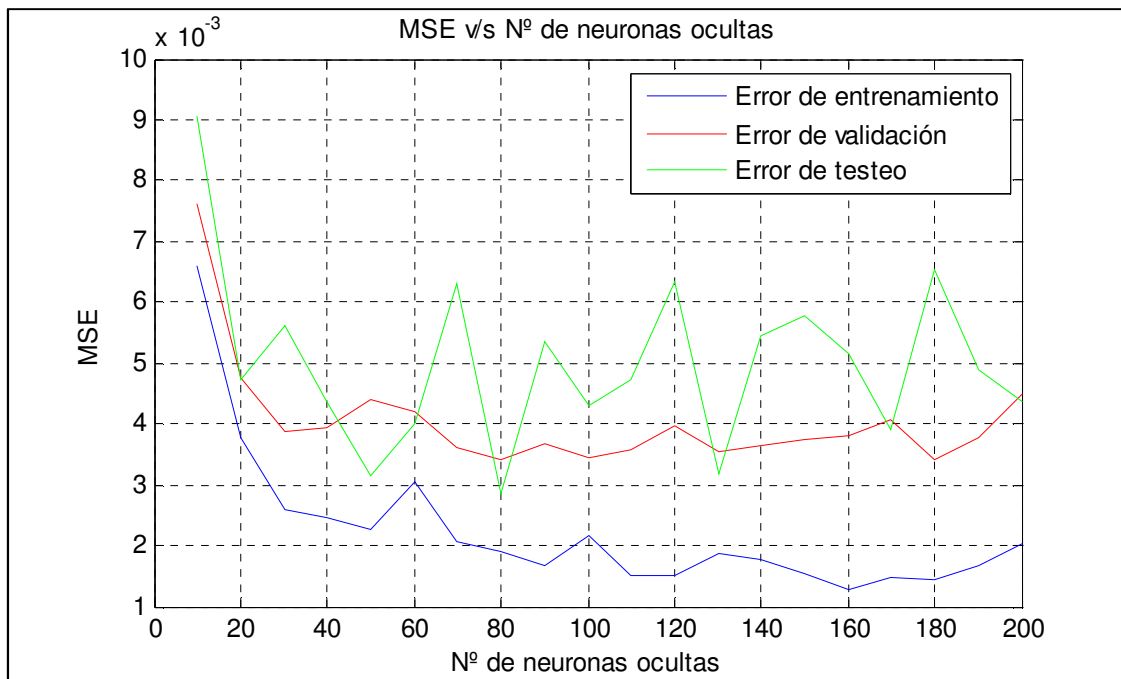


Figura 18: Evolución del error para redes de entrada directa con ruido.

Con esto se determino que las mejores configuraciones encontradas, corresponden a las siguientes redes:

- Tipo: MLP

- Nº de entradas: 7
- Nº de capas ocultas: 1
- Nº de neuronas capa oculta: 60 (red sin ruido) y 80 (red con ruido)
- Función de transferencia en la capa oculta: Sigmoidea tangente
- Nº de neuronas de salida: 7
- Función de transferencia en la capa de salida: Identidad saturada simétrica
- Método de entrenamiento: Levenberg-Marquardt

Teniendo la configuración óptima para las redes según su error de entrenamiento y validación. Se procede a analizar en profundidad su desempeño al detectar daños, para esto se genera un nuevo conjunto de validación. Este conjunto consiste en daños semi-aleatorio (generados con latin hypercube¹) de 1 y 2 daños simultáneos, con número equivalente al 10% de los patrones utilizados en el entrenamiento (6 de un daño y 134 de 2 daños simultáneos, 140 en total). Esto con el fin de que analizar como se comporta la red antes un conjunto de datos nuevos, que sea consistente con los patrones de entrenamiento pero que no siga una distribución uniforme como ellos. Para simular un caso real, todas las redes son probadas con patrones con ruidos y entradas consistes a los patrones de entrenamiento, directas o normalizadas según corresponda.

Para las redes seleccionadas en esta sección se calcularon los indicadores de desempeño presentados en la sección 2.7 haciendo uso de los nuevos patrones de validación (figura 19 a la 26). Se puede observar que en todos lo casos las redes de entrada directa tiene indicadores de desempeño ligeramente mejores a sus pares de entrada normalizada. Al comparar las redes que fueron entrenadas sin ruido y a las que se les introdujo ruido antes de entrenar, esta ultimas tiene mejores indicadores de desempeño. Esto se cumple para todos los indicadores y ambos tipos de entrada, directa y normalizada. Salvo para los falsos negativos y las redes de entrada directa, en que el porcentaje se mantiene constante, pero cambia el lugar donde ocurre el error.

¹ Latin Hypercube: Método de generación de números quasi-aleatorios, en este caso consiste en numero aleatorios por intervalos que están ordenados aleatoriamente [16]

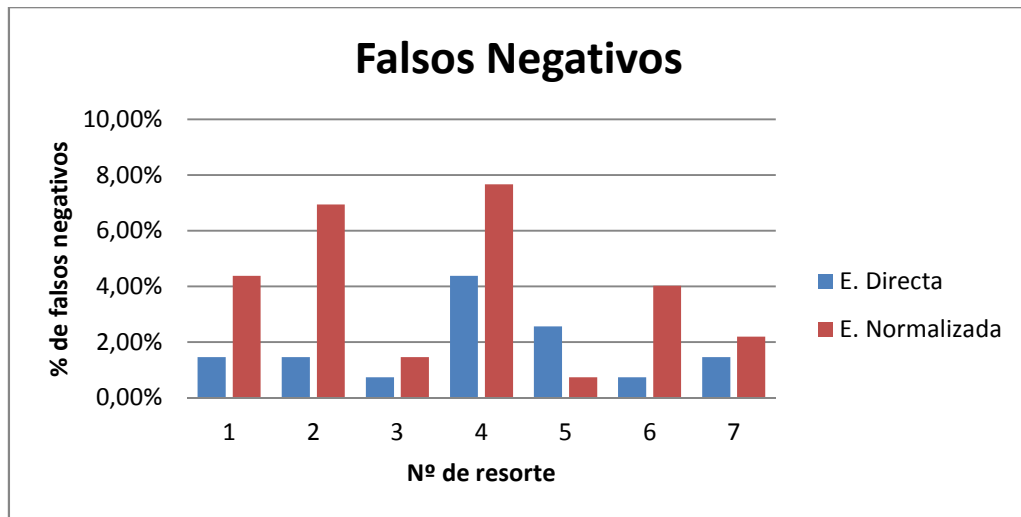


Figura 19: Número de Falsos Negativos al 2.5% (Redes sin ruido).

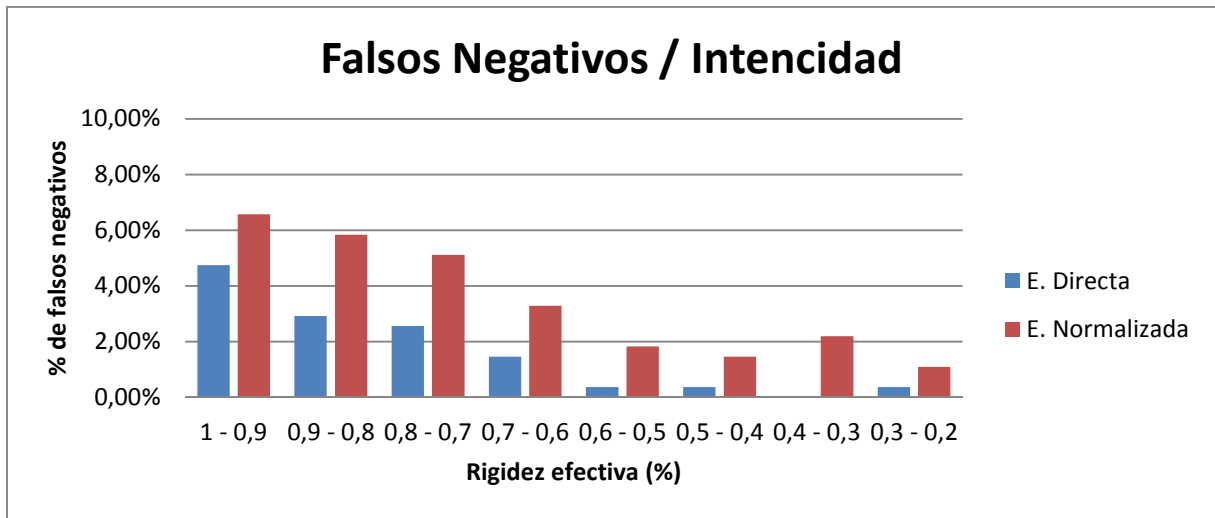


Figura 20: Número de Falsos Negativos, según rigidez efectiva al 2.5% (Redes sin ruido).

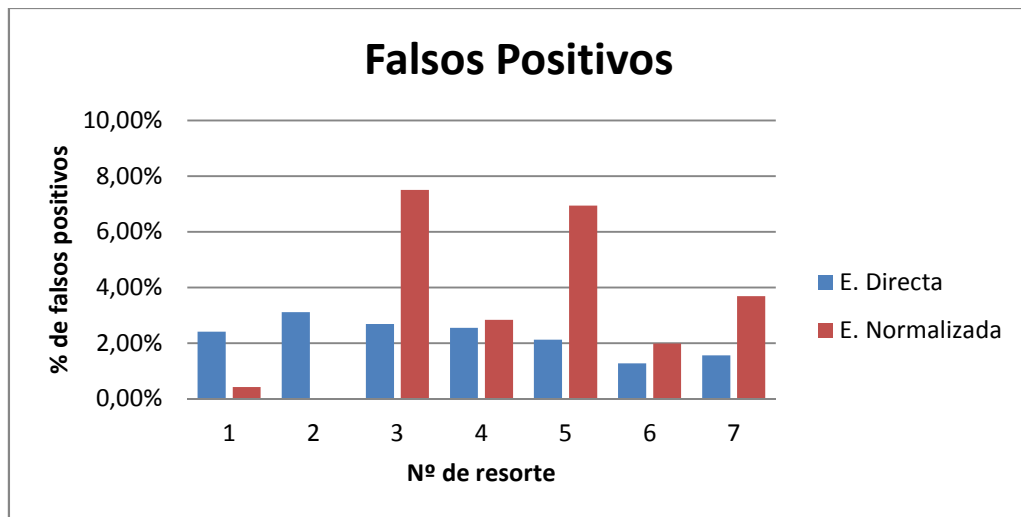


Figura 21: Número de Falsos Positivos al 2.5% (Redes sin ruido).

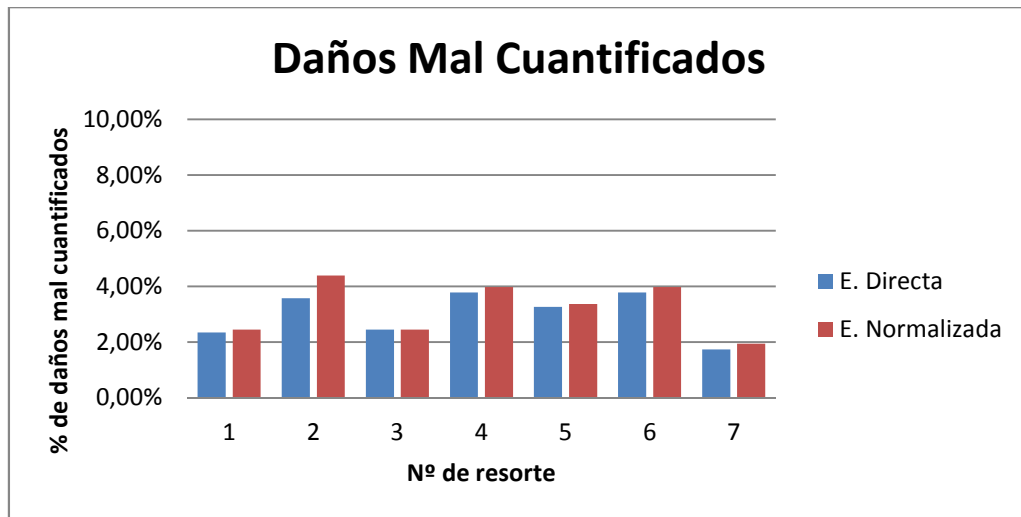


Figura 22: Daños mal cuantificado, según nº de resorte al 5% (Redes sin ruido).

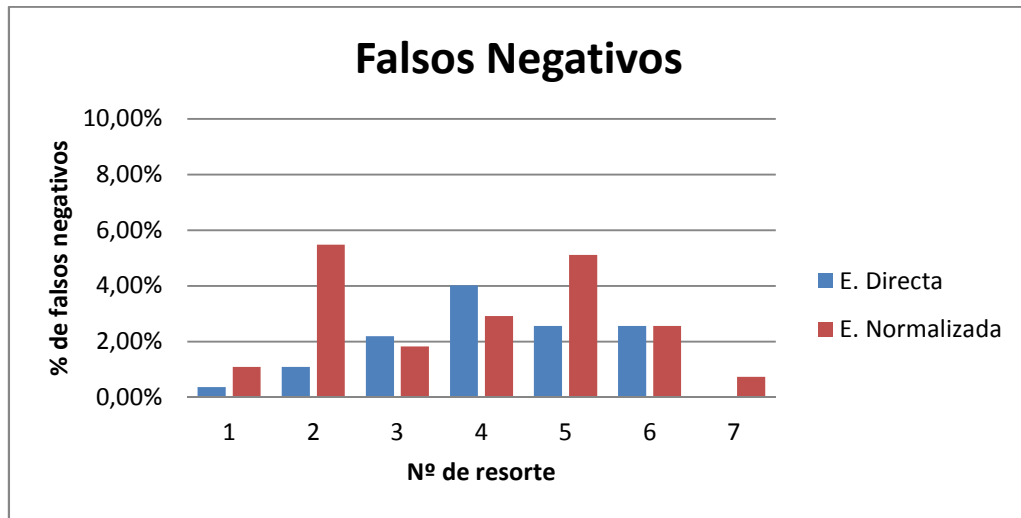


Figura 23: Número de Falsos Negativos al 2.5% (Redes con ruido).

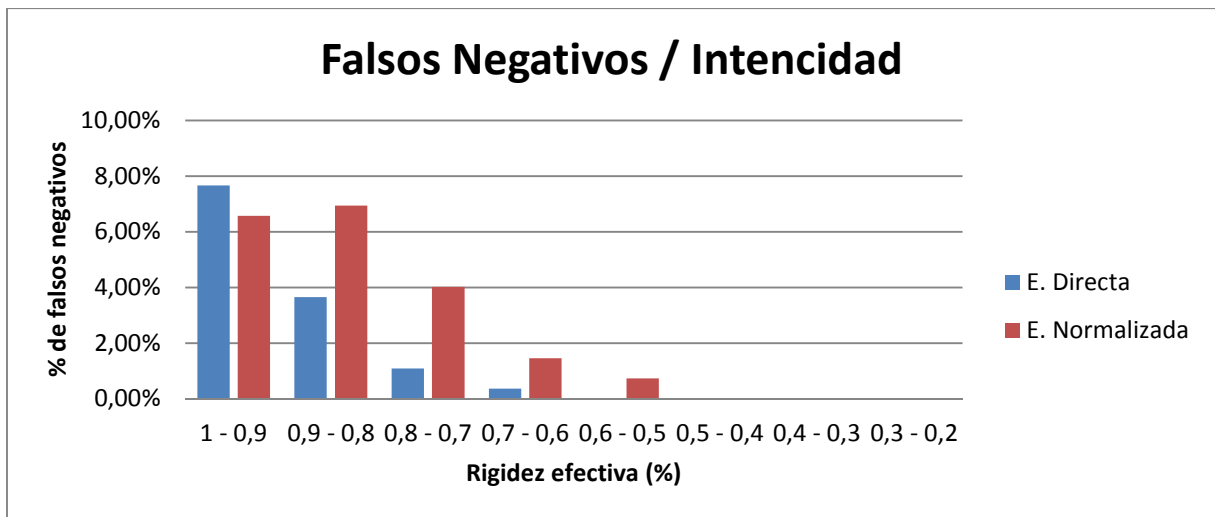


Figura 24: Número de Falsos Negativos, según rigidez efectiva al 2.5% (Redes con ruido).

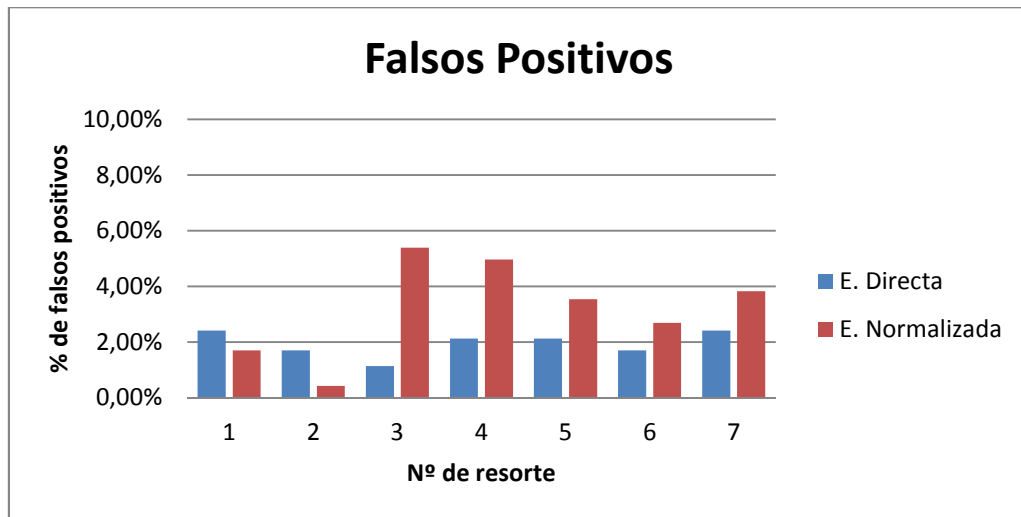


Figura 25: Número de Falsos Positivos al 2.5% (Redes con ruido).

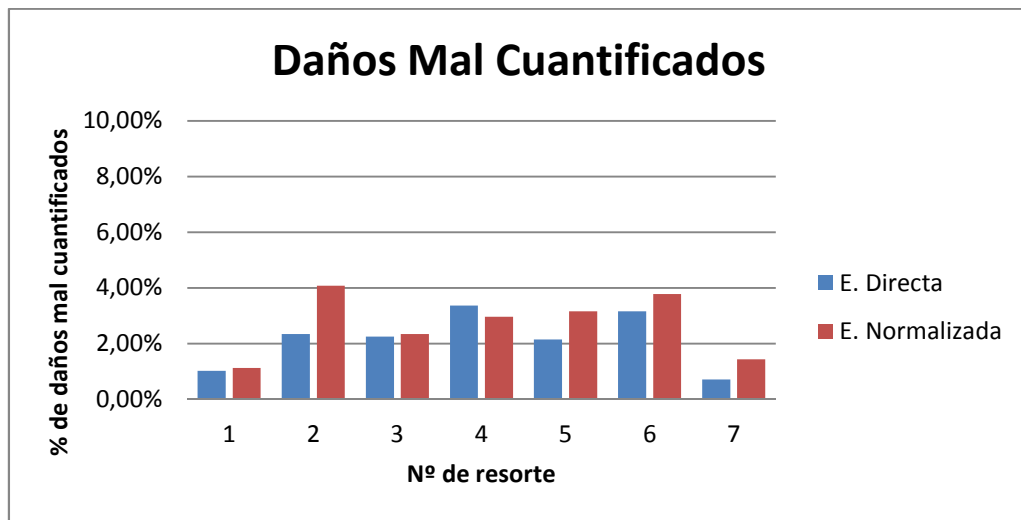


Figura 26: Daños mal cuantificado, según nº de resorte al 5% (Redes con ruido).

4.4 Prueba Experimental

Además de los datos utilizados para validar el modelo, se cuenta con datos de un caso real de daño para este sistema, este consiste en la disminución de rigidez del 55% en el quinto resorte. La variación en las frecuencias antiresonantes se puede revisar en anexo H.

Con estos datos se prueba la red y se presenta los resultados en la figura 27 y 28. En esta figura primero se muestra el daño objetivo (azul) y la detección obtenida al ingresar los datos experimentales en las redes. El caso marcado con una R hace referencia a las redes entrenadas con patrones que incluían ruido. Los resultados se muestran como factores de pérdida de rigidez, por lo que 0% corresponde a que el resorte esta sano y 100% corresponde a que el resorte esta totalmente dañado.

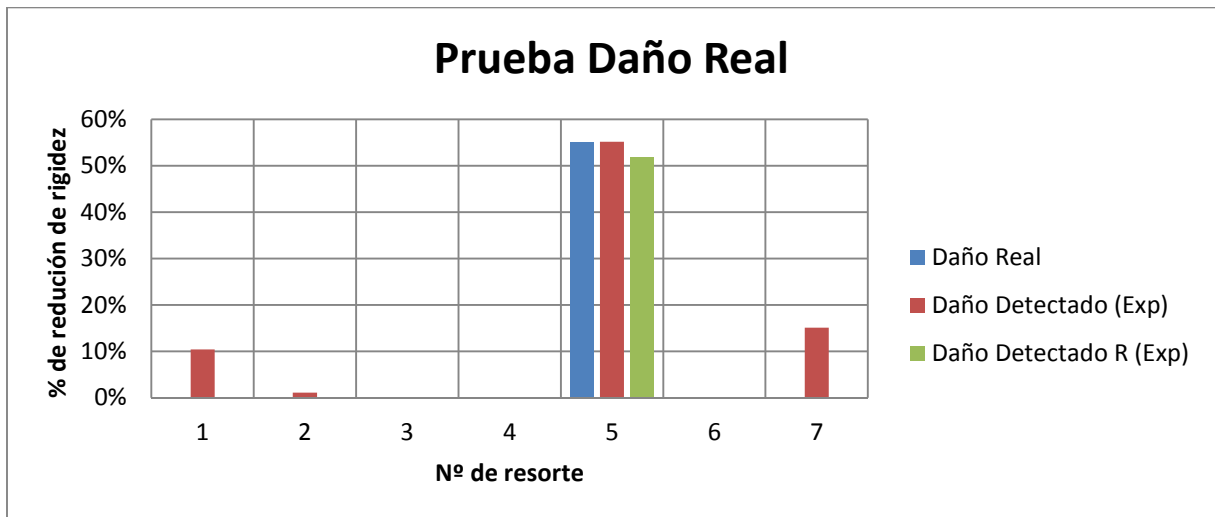


Figura 27: Resultados detección caso experimental (Entrada directa).

Al analizar los resultados de las redes con entrada directa, se puede detectar correctamente el daño en el resorte 5, pero además se detectan 2 casos de falsos positivos. Luego, se puede ver que al agregar ruido a los patrones de entrenamiento, aun se puede lograr una detección muy cercana al daño en el resorte 5 (52%) y además se elimina la aparición de falsos positivos.

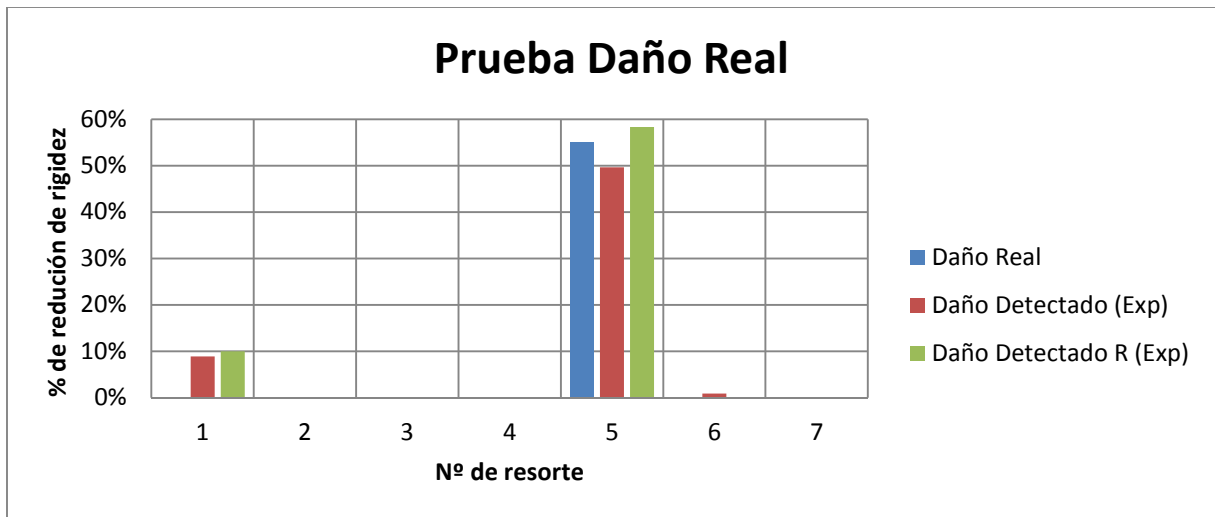


Figura 28: Resultados detección caso experimental (Entrada normalizada).

Por otro lado al analizar los resultados de la redes con entradas normalizadas, se observa que hay un error de más del 5% en la magnitud del daño del quinto resorte y hay existencia de un falso positivo en el resorte número 1. Al probar la red entrenada con ruido existen mejoras en la detección (3% de diferencia), pero el falso positivo persiste en los resultados. Al realizar pruebas en otras redes entrenadas de igual manera el falso positivo varia su posición e intensidad en cada caso, por lo que se descarta que pueda ser un problema del ajuste del modelo analítico.

5. Resultados Problema 2

El segundo problema de estudio consiste en una viga libre que se puede mover en todas direcciones, pero solo se estudia los desplazamientos verticales u horizontales que sean transversales a ella. Para el método de elementos finitos la viga es discretizada en 20 secciones de igual tamaño, lo que da un total de 42 grados de libertad contando desplazamientos transversales y rotaciones en el mismo sentido.

Datos:

- $E = 2 \times 10^{11}$ [Pa]
- $L = 1$ [m]
- $\rho = 7800$ [Kg/m³]
- Pretina solida de 25x10 [mm]

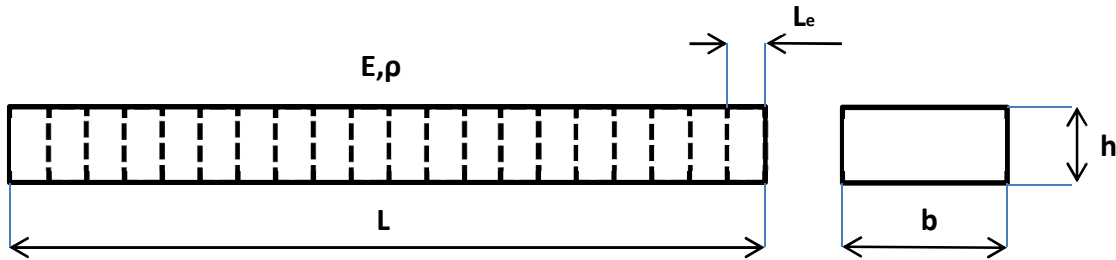


Figura 29: Viga libre.



Figura 30: Montaje experimental problema 2.

En la figura 30 se puede observar el montaje experimental que se utiliza para la toma de datos. Este consiste en la viga de acero montada por su lado más ancho en

dos cuerdas elásticas, que permiten al sistema moverse libremente sin agregar rigidez al sistema. En este montaje se utiliza un sensor móvil que es fijado con cera, él que registra la respuesta en el tiempo de la viga al ser excita con un martillo modal. Con esta configuración se obtienen las FRFs directas entre 0 y 5120 [Hz] en distintos punto de viga, promediando los resultados de varias pruebas.

La resolución de este problema se realiza programando una función en Matlab que recibe los factores de rigidez efectiva para cada resorte y devuelve las frecuencias características del sistema, resonantes y antiresonantes (Anexo C). A diferencia del problema 1, en este caso hay 2 ejes transversales que tienen su propia respuesta dinámica, por lo que se trabaja el problema en los 2 ejes para comparar el desempeño de la detección de daños. Los siguientes resultados se clasificaran como horizontales o verticales, según la disposición de la barra en la figura 29.

5.1 Ajuste de Modelo

El ajuste de modelos del problema se realiza en base a los datos experimentales obtenidos durante la realización de este trabajo, al igual que en el problema 1 se cuenta con los datos de FRF, frecuencias resonantes y frecuencias antiresonantes. En este caso no tiene sentido realizar un ajuste para cada elemento de la viga como si fueran resortes y masas independientes, sino que se realiza un ajuste de las propiedades de la viga suponiendo su uniformidad. Para el ajuste de modelo se agregan 2 variables de optimización λ ; una para la densidad y otra para el modulo de Young. Se realiza el ajuste de modelo con las mismas restricciones que el problema 1. El ajuste de modelo se realiza en las 2 direcciones de manera independiente, por si la viga no fuera absolutamente politrópica debido a su manufactura. Los resultados se presentan en la tabla 7.

Tabla 7: Factores de optimización del ajuste de modelo.

Optimización vertical		Optimización horizontal	
ρ	E	ρ	E
0.953	0.923	1.052	0.952

Como es de esperar en ambos casos las propiedades de la barra son muy similares en cada dirección y a pesar de que hay diferencias en los módulos de Young obtenidos, la diferencia no es suficiente para sospechar de problemas en los datos. Hay que notar que al ajustar el modulo de Young indirectamente se esta ajustando la rigidez de la viga, la cual también depende de la inercia de esta (7), por lo que estas diferencias se podrían deber a pequeñas diferencias en las dimensiones de la viga. A continuación se presentan los efectos del ajuste en las frecuencias antiresonantes de la viga (Tabla 8 y 9).

Tabla 8: Ajuste de frecuencia antiresonantes verticales

Frecuencias Antiresonantes (ω_r)	Experimental [Hz]	Analítica inicial [Hz]	Error [%]	Analítica corregida [Hz]	Error [%]
1	112,8	119,1	5,60%	117,2	3,92%
2	242,2	248,5	2,61%	244,6	0,99%
3	415,0	425,0	2,42%	418,3	0,79%
4	638,8	648,7	1,55%	638,4	-0,06%
5	904,7	919,6	1,65%	905,1	0,04%
6	103,4	106,3	2,80%	104,6	1,17%
7	239,7	241,3	0,68%	237,5	-0,92%
8	460,0	463,7	0,81%	456,4	-0,79%
9	659,4	678,8	2,95%	668,1	1,31%
10	832,2	845,9	1,64%	832,4	0,03%
11	104,4	106,3	1,82%	104,6	0,20%
12	237,8	241,3	1,48%	237,5	-0,13%
13	456,6	463,7	1,56%	456,4	-0,05%
14	671,6	678,8	1,08%	668,1	-0,53%
15	829,1	845,9	2,02%	832,4	0,40%
16	118,1	119,1	0,86%	117,2	-0,74%
17	246,9	248,5	0,66%	244,6	-0,94%
18	422,5	425,0	0,60%	418,3	-1,00%
19	643,1	648,7	0,87%	638,4	-0,73%
20	910,0	919,6	1,06%	905,1	-0,54%

Tabla 9: Ajuste de frecuencia antiresonantes horizontales

Frecuencias Antiresonantes (ω_r)	Experimental [Hz]	Análítica inicial [Hz]	Error [%]	Análítica corregida [Hz]	Error [%]
1	284,1	297,8	4,82%	283,3	-0,29%
2	595,9	621,3	4,27%	591,1	-0,81%
3	1024,1	1062,6	3,76%	1010,8	-1,30%
4	1543,1	1621,8	5,10%	1542,8	-0,02%
5	2168,1	2299,1	6,04%	2187,1	0,88%
6	260,9	265,7	1,86%	252,8	-3,11%
7	583,1	603,3	3,46%	573,9	-1,58%
8	1112,8	1159,3	4,18%	1102,8	-0,90%
9	1621,9	1697,1	4,64%	1614,4	-0,46%
10	1995,0	2114,6	6,00%	2011,6	0,83%
11	260,0	265,7	2,21%	252,8	-2,77%
12	584,4	603,3	3,23%	573,9	-1,79%
13	1113,8	1159,3	4,09%	1102,8	-0,98%
14	1617,5	1697,1	4,92%	1614,4	-0,19%
15	1994,4	2114,6	6,03%	2011,6	0,86%
16	293,4	297,8	1,49%	283,3	-3,45%
17	606,3	621,3	2,48%	591,1	-2,51%
18	1032,8	1062,6	2,88%	1010,8	-2,13%
19	1552,8	1621,8	4,44%	1542,8	-0,65%
20	2181,3	2299,1	5,40%	2187,1	0,27%

5.2 Red Neuronal (Vertical)

Para encontrar la configuración óptima de este problema se va a seguir la misma metodología que en el problema 1, de tal manera de comprobar que se mantiene el comportamiento de las redes con problemas más complejos. Se mantiene la configuración de red salvo el número de neuronas a utilizar en cada capa. Para este problema se cuenta con 41 frecuencias antiresonantes en cada punto de medición, pero estas no se pueden utilizar en su totalidad ya que alcanzan magnitudes muy altas y no todas son medibles experimentalmente. Para que el problema tenga solución debe existir más entradas que salidas en la red, por lo que se utilizan las 5 antiresonancias (3° a la 7°) de cuatro puntos de la viga. Estos puntos corresponden a los 2 extremos y a 2 puntos ubicados a 25 de cada extremo. En un comienzo se iban a utilizar más frecuencias de solo 3 puntos, pero esta opción se descartó, ya que no se pudieron determinar todas las frecuencias antiresonantes en el punto medio de la viga. La rutina utilizada en Matlab se incluye en el anexo D.

Al seguir este procedimiento se obtiene la evolución del error para redes con entradas directas y patrones sin y con ruido al aumentar el número de neuronas oculta, estos resultados se pueden observar en la figuras 31 y 32.

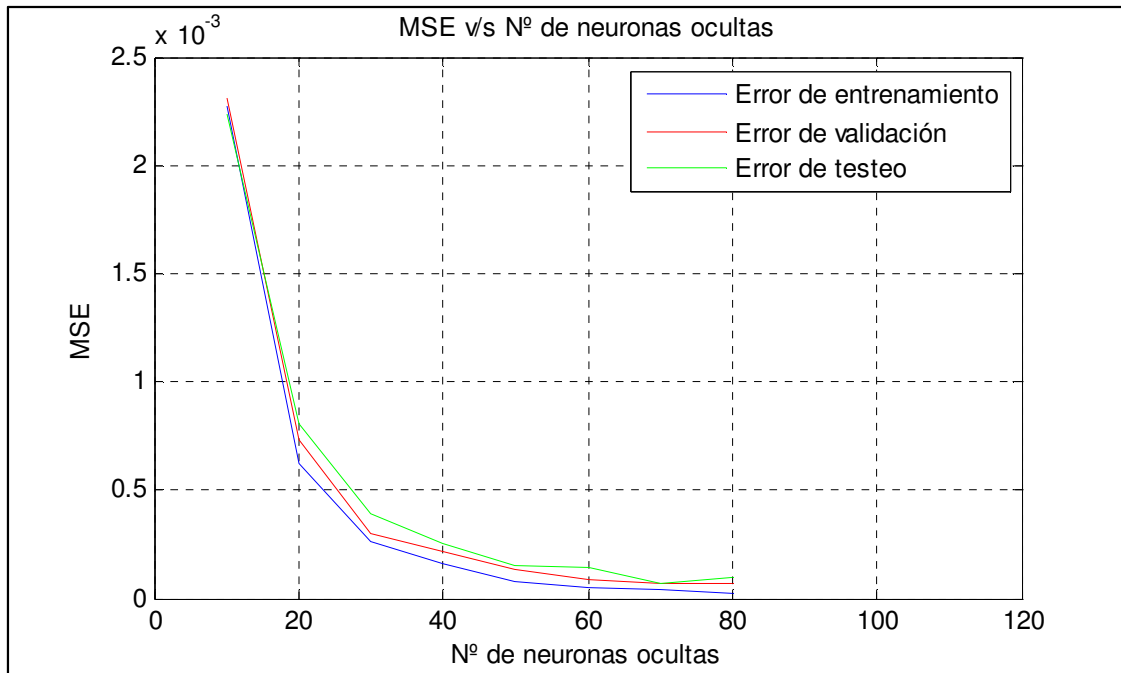


Figura 31: Evolución del error para redes de entrada directa sin ruido

Debido a la mayor cantidad de datos para entrenar la red (12321 patrones), la naturaleza e implementación del método de entrenamiento, tanto los tiempos de entrenamiento como los recursos necesarios (memoria RAM) crecen de manera geométrica por lo que no se pueden entrenar redes tan grandes como en el problema 1. Igualmente al entrenar las redes hasta 80 neuronas ocultas, se encuentran niveles de error comparables al del problema 1. También se observan tendencias muy similares, al aumentar el número neuronas en la capa oculta, tanto para el caso sin ruido como para el caso con ruido. Para las redes entrenadas sin ruido los errores de los tres conjuntos permanecen muy cercanos, mientras que en las redes con ruido tienen una mayor diferencia. En la tabla 10 se presenta el error obtenido al entrenar las redes con los 4 tipos de redes definidos en la sección 4.2. Se puede ver que tal como en el problema 1, existe un comportamiento muy similar en cuanto al error obtenido sin importar si la entrada es directa o normalizada.

Tabla 10: Error obtenido en las RNAs.

Patrones	Sin ruido		Con ruido	
	Directa	Normalizada	Directa	Normalizada
Error _e	2,76E-05	3,00E-05	2,89E-03	2,91E-03
Error _v	6,51E-05	6,08E-05	3,51E-03	3,48E-03
Error _t	9,70E-05	6,41E-05	3,67E-03	3,29E-03

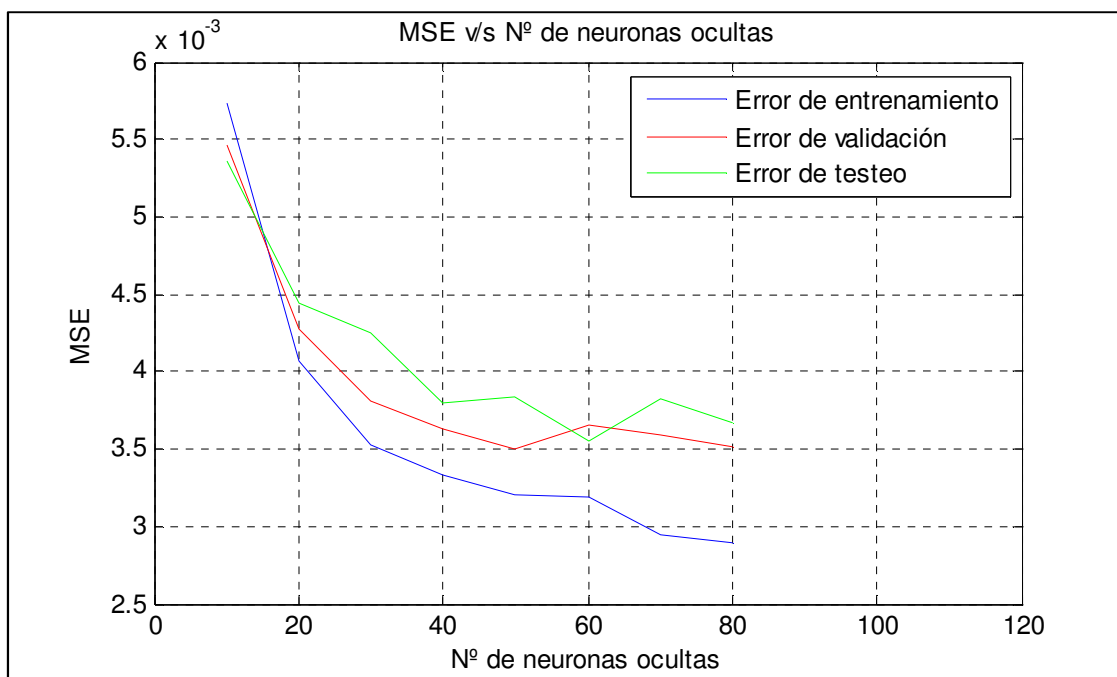


Figura 32 Evolución del error para redes de entrada directa con ruido

Con esto se determino que la configuración óptima para este problema, consiste en una red de las siguientes características:

- Tipo: MLP
- Nº de entradas: 20
- Nº de capas ocultas: 1
- Nº de neuronas capa oculta: 80 (redes sin y con ruido)
- Función de transferencia en la capa oculta: Sigmoidea tangente
- Nº de neuronas de salida: 20
- Función de transferencia en la capa de salida: Identidad saturada simétrica
- Método de entrenamiento: Levenberg-Marquardt

Ya determinada la configuración óptimas de las RNAs, se procede al igual que en el problema 1 a crear un nuevo conjunto de validación que sea representativo de los patrones de entrenamiento pero sin seguir la misma uniformidad que estos. Con esto se crean patrones con 16 casos de un daño y 1216 casos con daños 2 semi-aleatorios. En los anexos E y F se adjunta las rutinas utilizadas para generar los distintos patrones de validación y utilizarlos para calcular los indicadores de las redes presentadas.

Con estos patrones se determinan los indicadores de desempeño para las distintas redes seleccionadas (figura 33 a la 40). Lo primero que se puede notar es que no existe ninguna tendencia marcada para ningún indicador, de que este pudiera depender de su ubicación o de la severidad del daño a detectar. También se puede ver que al igual que en el problema 1 el hecho de agregar ruido a los patrones de entrenamiento mejora todos los indicadores, aunque siguen siendo bastante altos. Este

efecto es ligeramente mayor en la redes de entrada directa, salvo en los falsos negativos, que para la redes normalizadas pasan de un 56% aun 31% gracias a la adición de ruido.

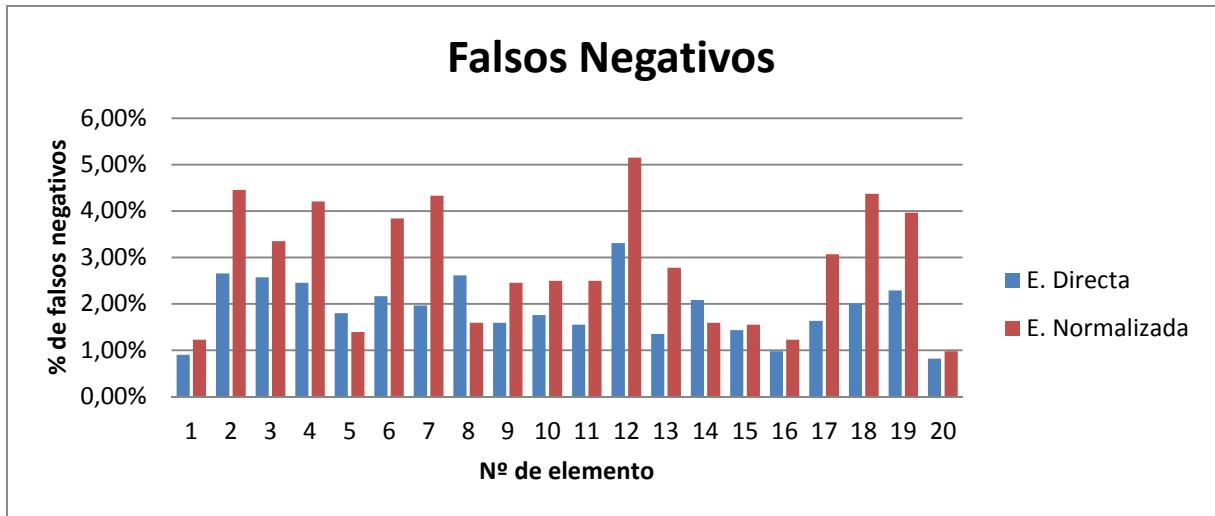


Figura 33: Número de Falsos Negativos al 2.5% (Redes sin ruido).

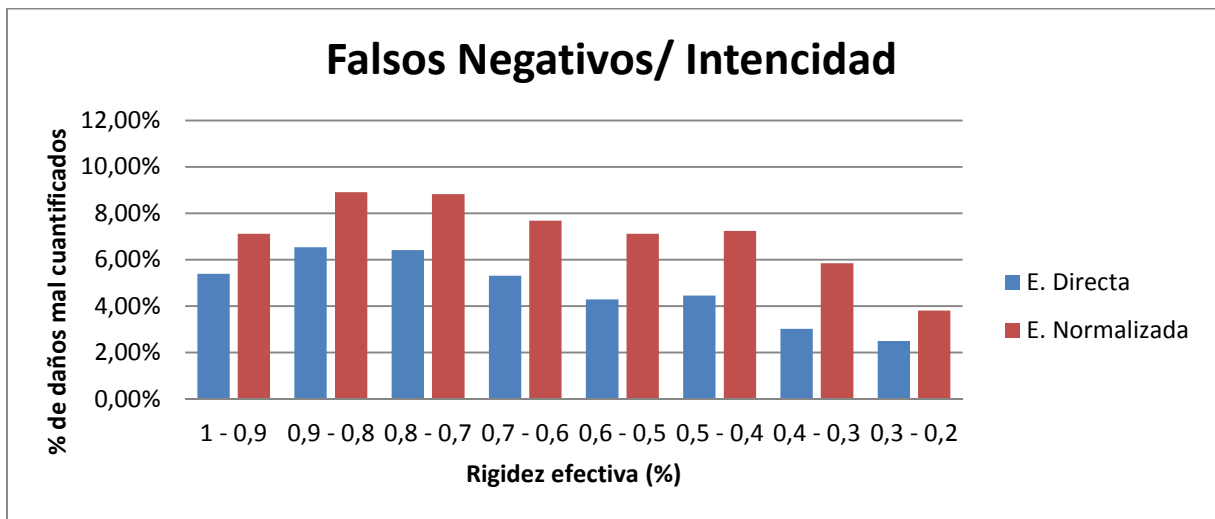


Figura 34: Número de Falsos Negativos, según rigidez efectiva al 2.5% (Redes sin ruido).

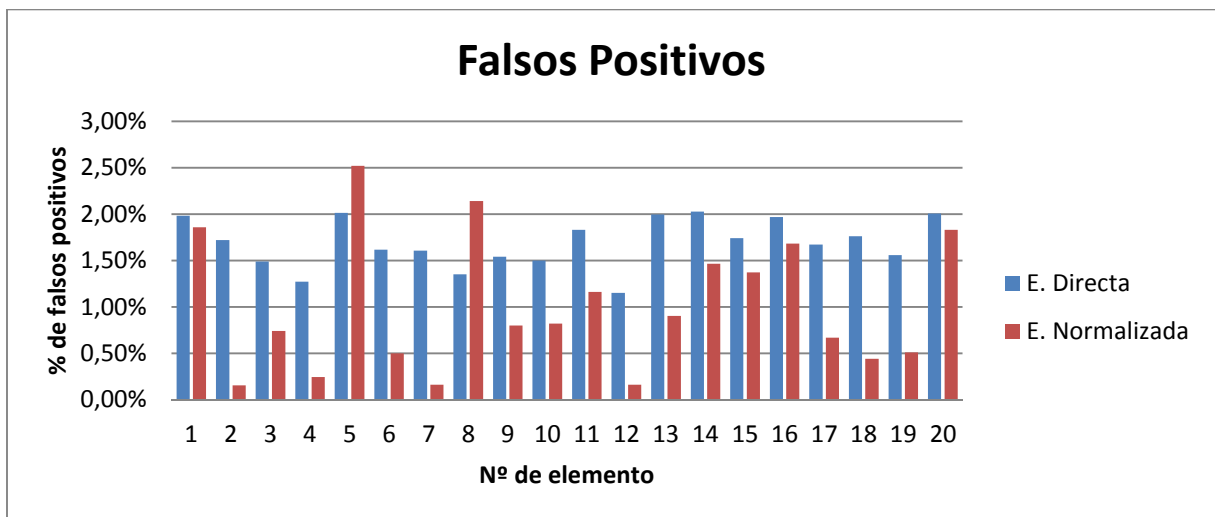


Figura 35: Número de Falsos Positivos al 2.5% (Redes sin ruido).

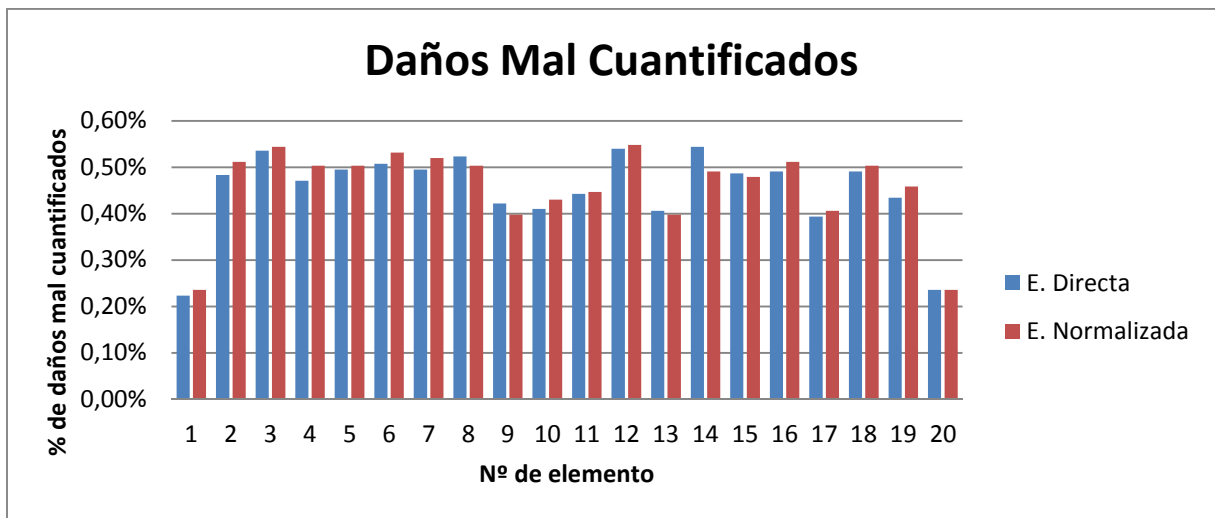


Figura 36: Daños mal cuantificado, según Nº de elemento al 5% (Redes sin ruido).

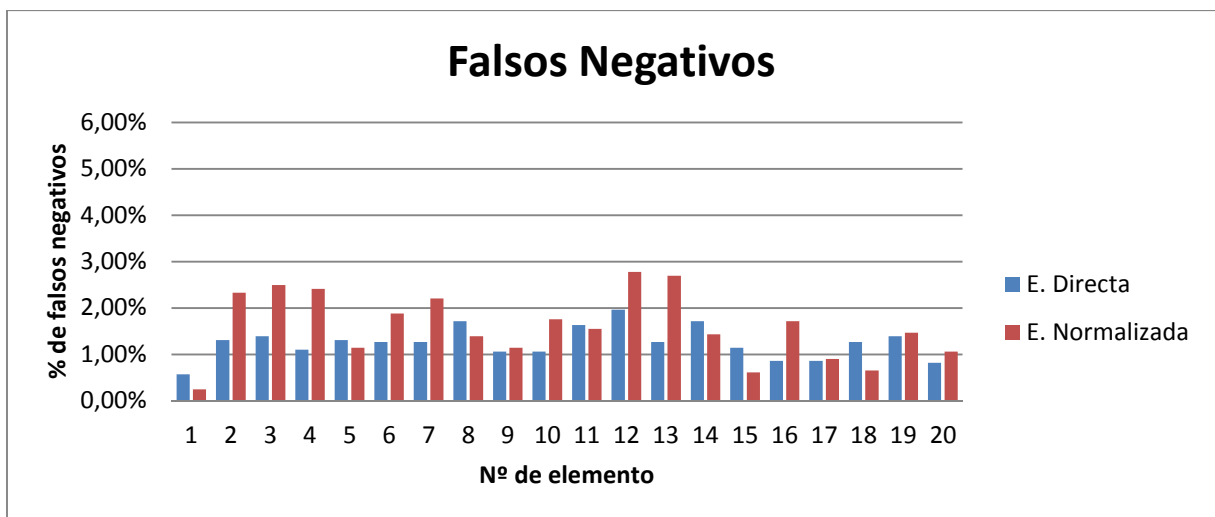


Figura 37: Número de Falsos Negativos al 2.5% (Redes con ruido).

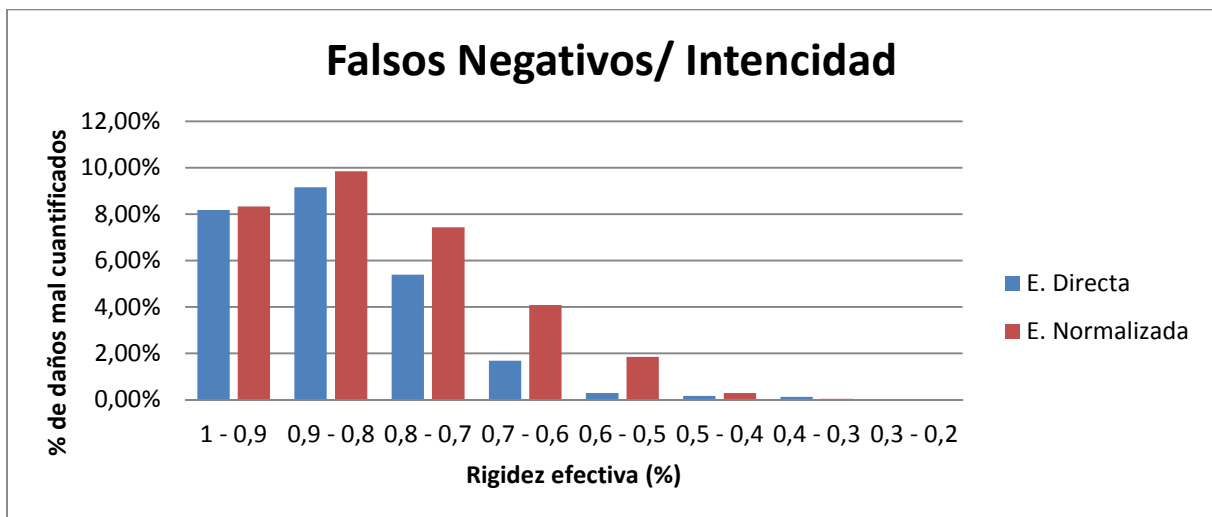


Figura 38: Número de Falsos Negativos, según rigidez efectiva al 2.5% (Redes con ruido).

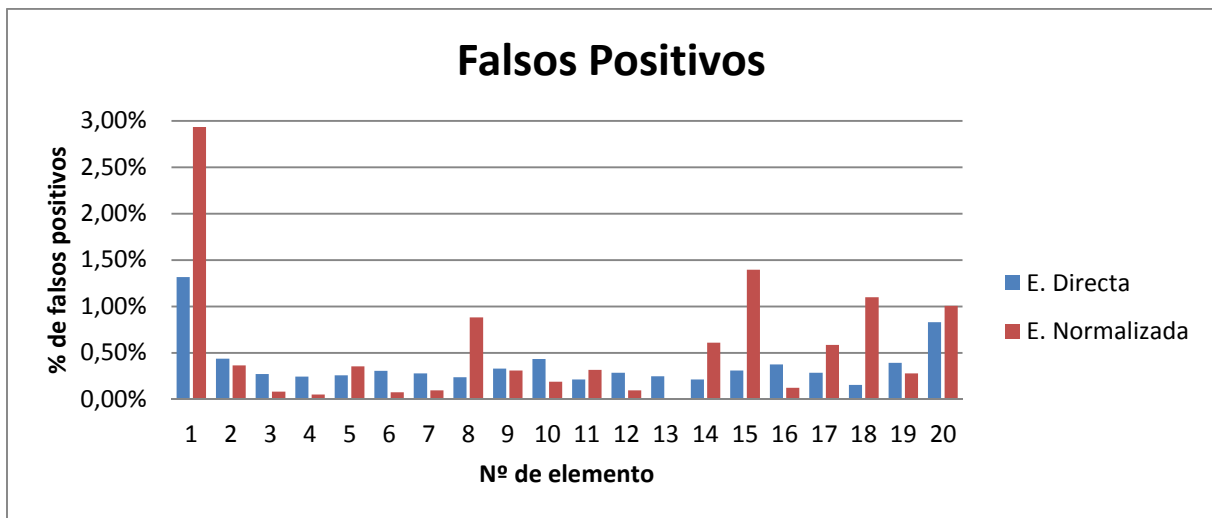


Figura 39: Número de Falsos Positivos al 2.5% (Redes con ruido).

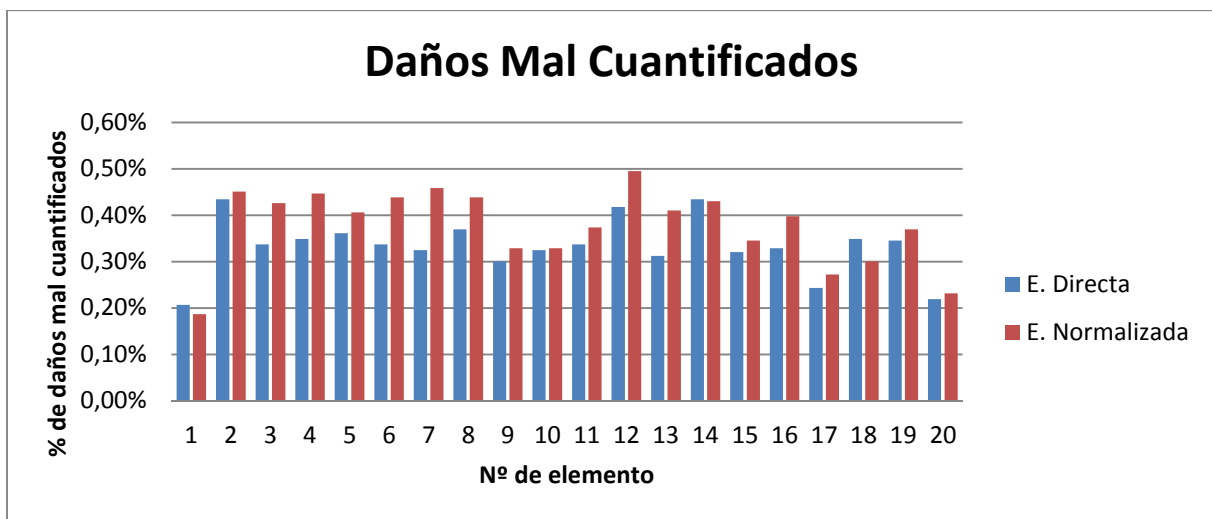


Figura 40: Daños mal cuantificado, según Nº de elemento al 5% (Redes con ruido).

5.3 Prueba Experimental (Vertical)

Para realizar la validación experimental se definió un grupo de daños para realizarlos en cinco vigas (6 incluyendo el caso sin daño), a las que se le realizaron cortes de sierra en distintos elemento (Figura 41) y distintas profundidades para simular grietas. Estos casos se presentan en la tabla 11.

Tabla 11: Distribución de cortes para representar daños reales

Viga N°	Elemento N°	Profundidad [mm]	Elemento N°	Profundidad [mm]	Dirección
0	-	-	-	-	Sin daño
1	7	5	-	-	Horizontal
2	13	10	-	-	Horizontal
3	7	2	-	-	Vertical
4	7	5	13	10	Horizontal
5	8	5	17	15	Horizontal



Figura 41: Muestra de cortes en la viga 3 y 5.

Ya realizados los cortes descritos en la tabla se obtienen las frecuencias antiresonantes verticales para los seis casos y se utilizan para dar la información de entrada a las distintas redes seleccionadas (Anexo H).

En este problema no existe una relación directa entre los cortes existentes en los elementos de las barras y la pérdida rigidez de estos. Por esto no tiene sentido hablar de daño objetivo, en este caso se busca que haya una localización clara de los cortes realizados y una aproximación adecuada del daño. Si una viga tiene dos cortes de distinta profundidad, se espera que la red pueda detectar dos daños de magnitudes distintas respetando la severidad de los cortes. Para facilitar la observación de los datos se agrega una flecha roja indicando la posición esperada del daño. En las figura 42 a la 47, se presentan los resultados obtenidos para cada viga al ingresar los datos experimentales en las 4 redes seleccionadas en la sección 5.2.

Al observar las distintas figuras, se puede notar que es difícil hablar de que haya una detección clara de daños y menos una detección que sea representativa de todos los daños buscados. En los resultados es difícil discernir que resultados podrían ser causados por los cortes en las vigas, dado a que se puede ver múltiples daños (mas de los que debería detectar la red) con magnitudes importantes por lo que no se les puede descartar como ruido en la mayoría de los casos. Se puede ver que la red de entrada directa entrenada sin ruido, detecta siempre daños de gran magnitud en distintos elementos, esto se puede deber a las diferencias entre el modelo y la estructura real. Incluso considerando esto, debido a la no linealidad del problema no se puede usar el caso conocido sin daños para filtrar los resultados y poder identificar daños en otros elementos. En cambio la red de entrada directa entrenada con ruido, no presenta esta gran cantidad de falsos positivos, incluso hay una detección clara de un daño en caso 3 y una detección mas difusa en el caso 2 y 4. En el caso 5 (figura 47), se podría apreciar una detección del corte mas profundo en el elemento 17, al utilizar la red de entrada normalizada entrenada con ruido, pero no habría detección del otro corte presente en la viga.

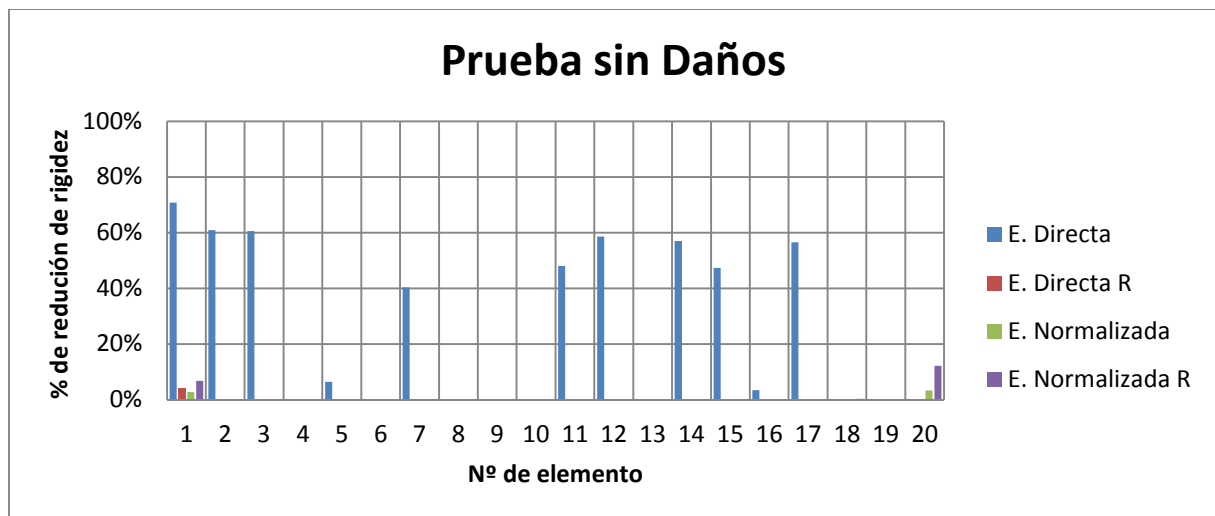


Figura 42: Resultados detección caso experimental 0 (Sin daños).

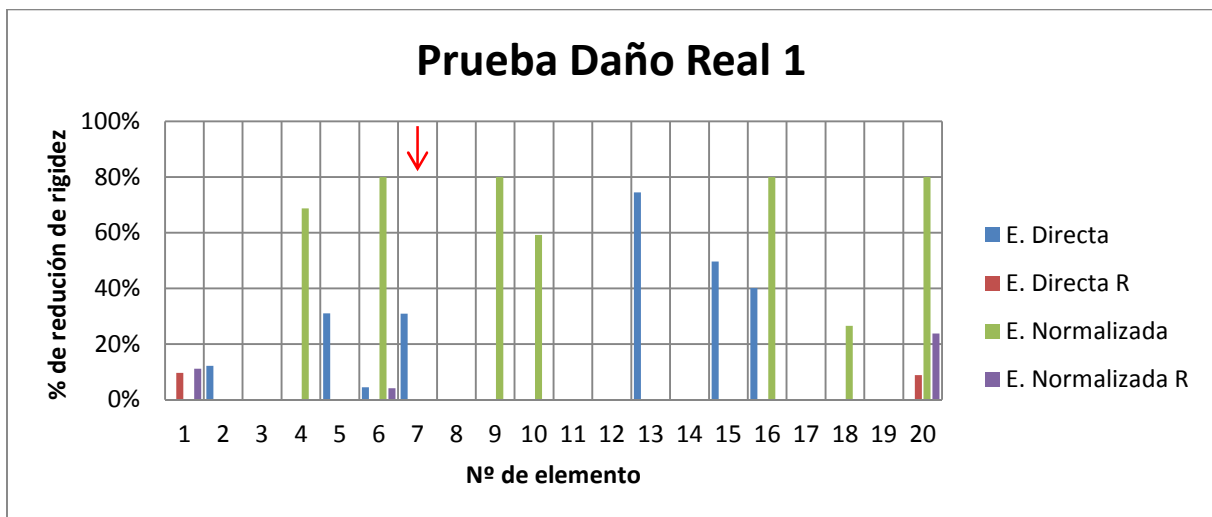


Figura 43: Resultados detección caso experimental 1 (Daño elemento 7).

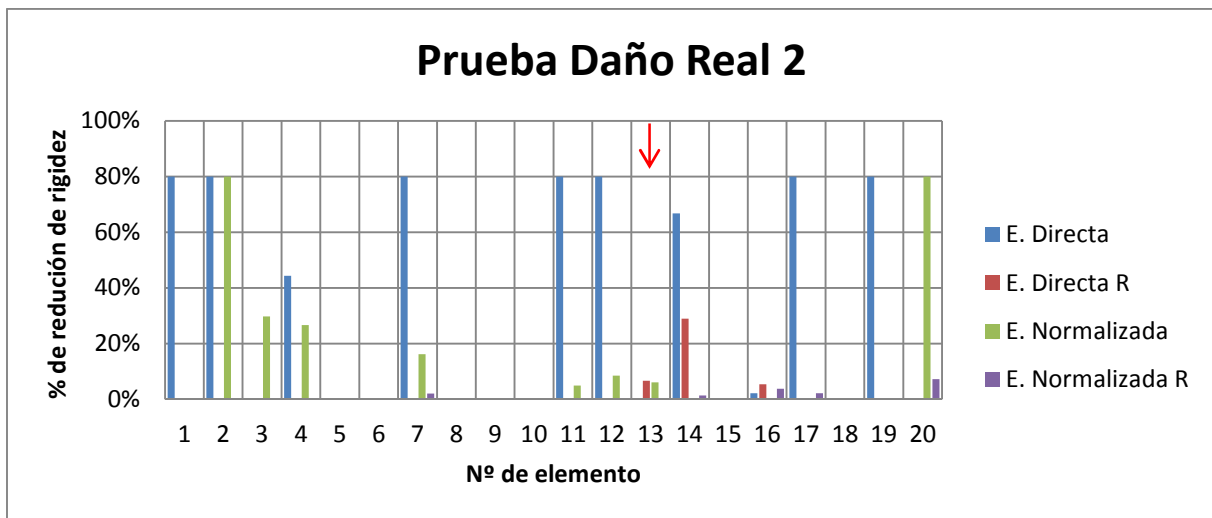


Figura 44: Resultados detección caso experimental 2 (Daño elemento 13).

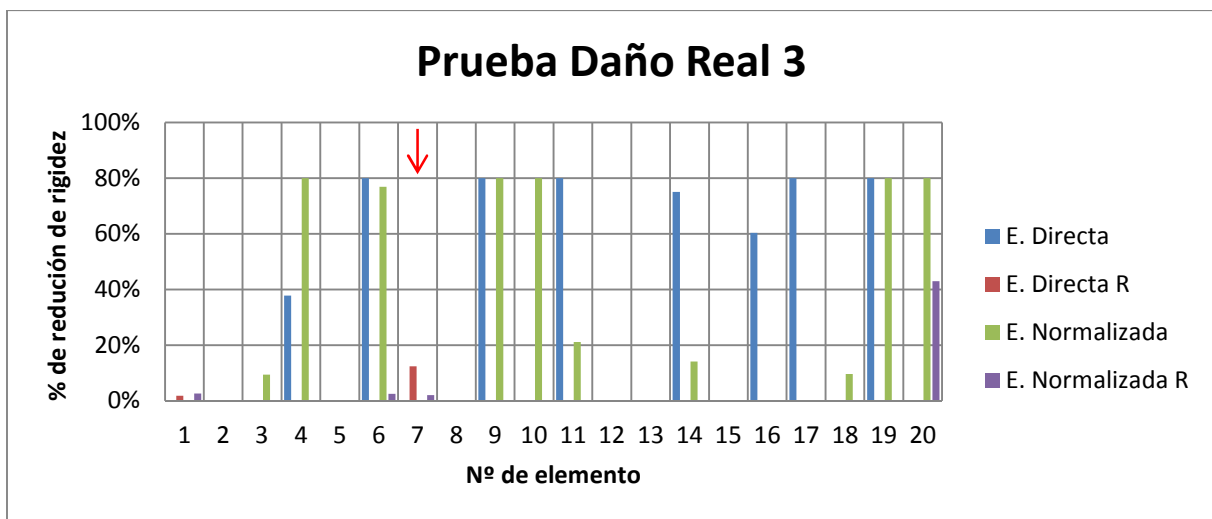


Figura 45: Resultados detección caso experimental 3 (Daño elemento 7).

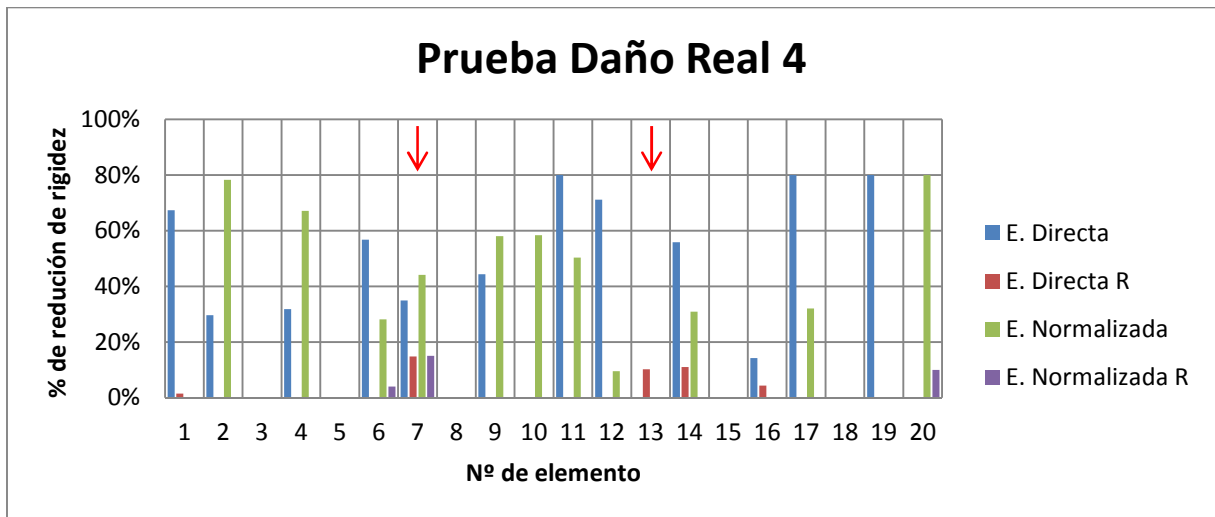


Figura 46: Resultados detección caso experimental 4 (Múltiples daños elementos 7 y 13).

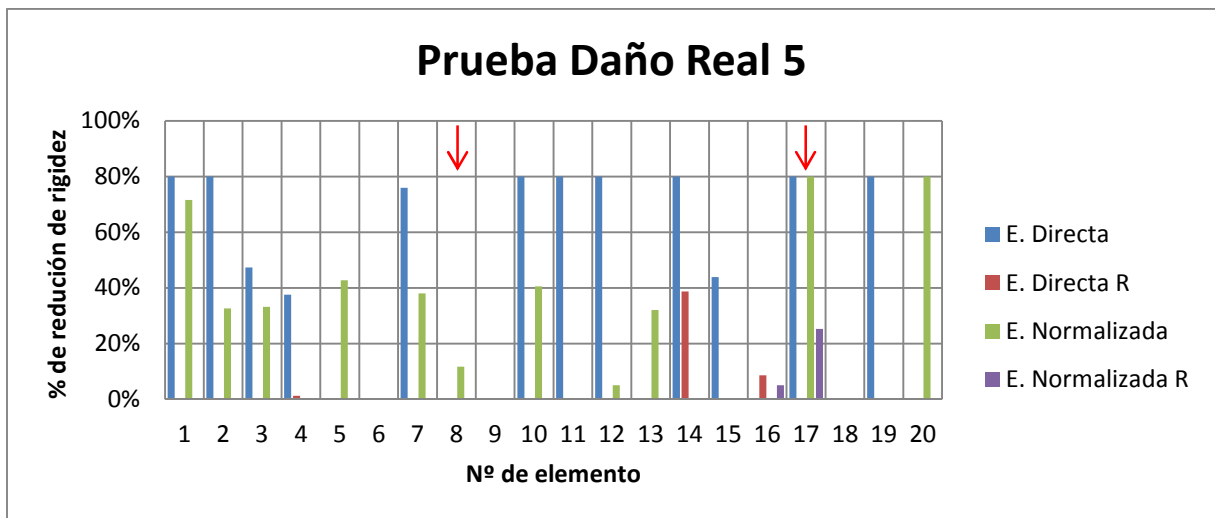


Figura 47: Resultados detección caso experimental 5 (Múltiples daños elementos 8 y 17).

Esta poca capacidad de la red para procesar y generalizar la información entregada de los casos experimentales, se puede deber a que exista una poca influencia de los cortes realizado en la respuesta dinámica en esta dirección. Esto es consistente, ya que los casos en que se podría apreciar detección corresponden a los cortes más profundos y al corte realizado en la dirección vertical. En consecuencia, es de esperar que se obtengan mejores resultado en la dirección horizontal.

5.4 Red Neuronal (Horizontal)

Dado que el problema en dirección horizontal es muy similar problema en dirección vertical visto en la sección 5.2 (mismo numero de entradas, salidas y patrones), incluso se puede mostrar que los patrones normalizados sin ruido son iguales, por lo que es esperar resultados muy similares a los ya mostrados. De esta

manera se puede determinar que la misma configuración de NRA se adapta al problema, con similares niveles de error (tabla 12) y las siguientes características:

- Tipo: MLP
- N° de entradas: 20
- N° de capas ocultas: 1
- N° de neuronas capa oculta: 80 (redes sin y con ruido)*
- Función de transferencia en la capa oculta: Sigmoidea tangente
- N° de neuronas de salida: 20
- Función de transferencia en la capa de salida: Identidad saturada simétrica
- Método de entrenamiento: Levenberg-Marquardt

Tabla 12: Error obtenido en las RNAs.

Patrones	Sin ruido		Con ruido	
	Directa	Normalizada	Directa*	Normalizada
Error _e	2,95E-05	2,96E-05	3,19E-03	2,93E-03
Error _v	5,39E-05	6,10E-05	3,86E-03	3,46E-03
Error _t	5,94E-05	5,75E-05	4,14E-03	3,53E-03

Al observar los resultados se puede observar una anomalía en los resultados, la que es más evidente al comparar estos resultados con lo presentados en la tabla 10. Estos resultados dispares se deben a que para la red de entrada directa entrenada con ruido no se alcanza el error mínimo, este es incluso mayor que el error obtenido con 70 neuronas en la capa oculta, Error_e =0.0044 y Error_v=0.0050. Por esto, se considera este caso *, donde la configuración óptima se caracteriza por tener 70 neuronas en la capa oculta. Esta discrepancia se puede deber que al tener mas ponderadores (por ser un red mas grande), hay mas variables aleatorias que pueden causar que el algoritmo se estanque en un mínimo local para el error de la red. Igualmente se considera este tipo de redes y su mejor configuración alcanzada para los análisis posteriores.

Al igual que en la sección 5.2, se determinan los indicadores de desempeños de las distintas redes para el conjunto de validación correspondiente (figura 48 a la 55). Incluso usando una red más pequeña, se puede observar que los indicadores de performance de ambas redes directas siguen siendo (aunque ligeramente) mejores que para las redes normalizadas. Como se visto en ambos caso anteriores la introducción de ruido a los patrones de entrenamiento genera mejoras en los indicadores de desempeño de las redes. En este caso en particular la mejora mas importante de para ambos tipos de redes, en cuanto a la cantidad de falsos positivos detectados por la red.

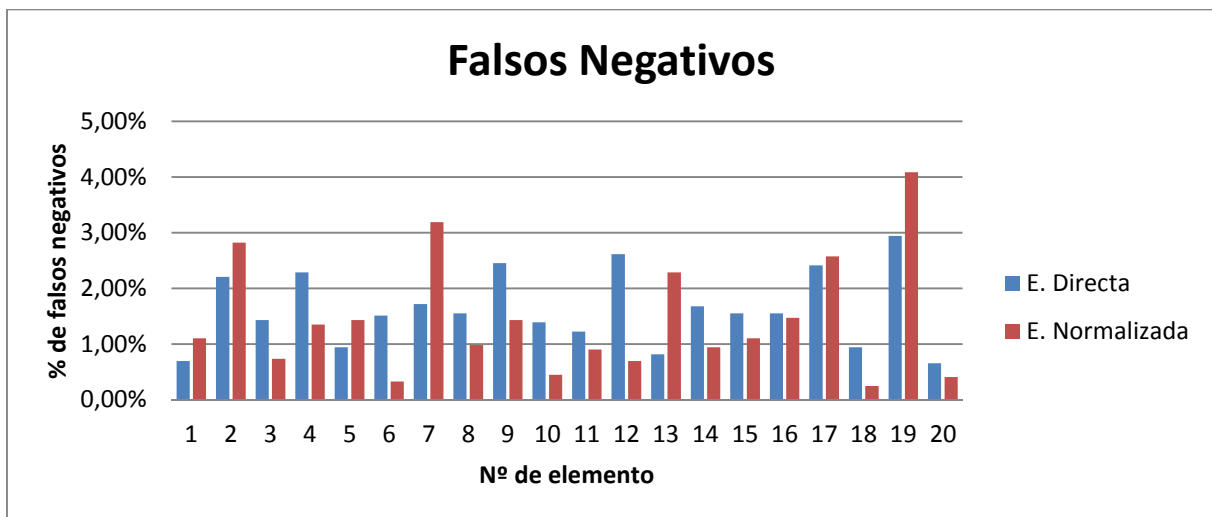


Figura 48: Número de Falsos Negativos al 2.5% (Redes sin ruido).

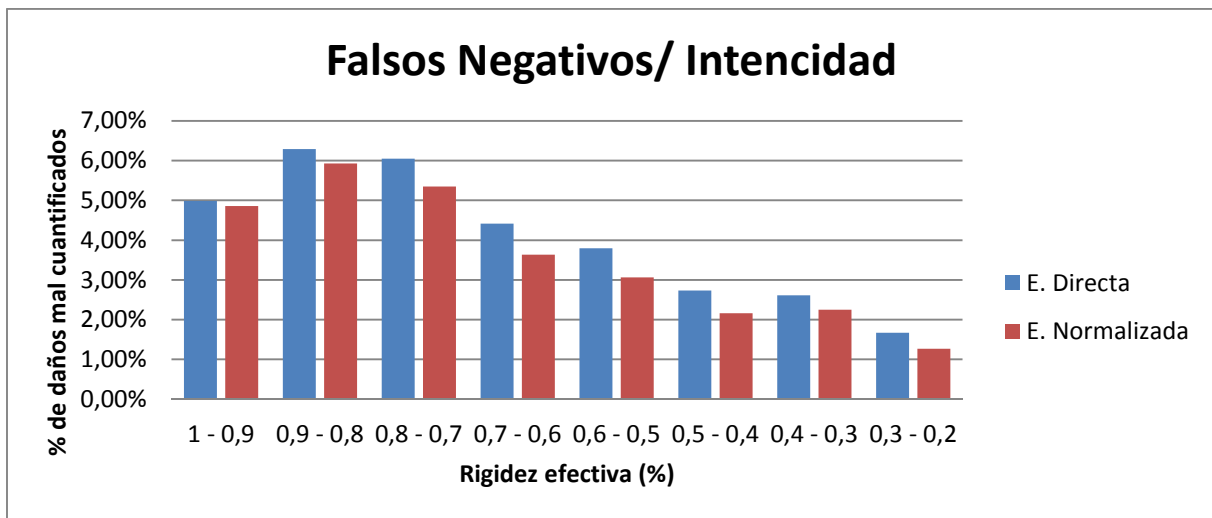


Figura 49: Número de Falsos Negativos, según rigidez efectiva al 2.5% (Redes sin ruido).

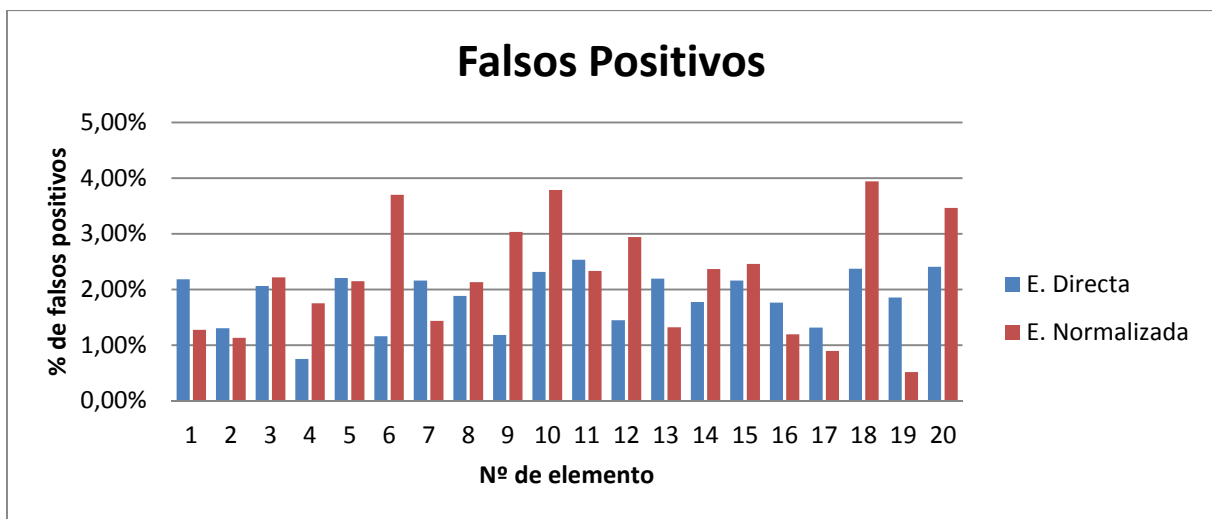


Figura 50: Número de Falsos Positivos al 2.5% (Redes sin ruido).

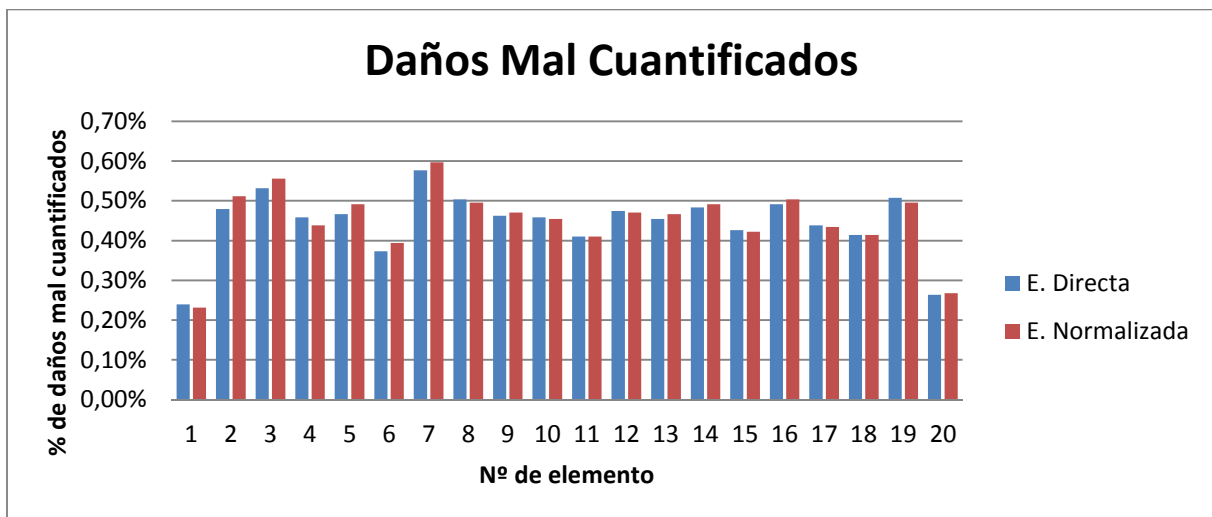


Figura 51: Daños mal cuantificado, según Nº de elemento al 5% (Redes sin ruido).

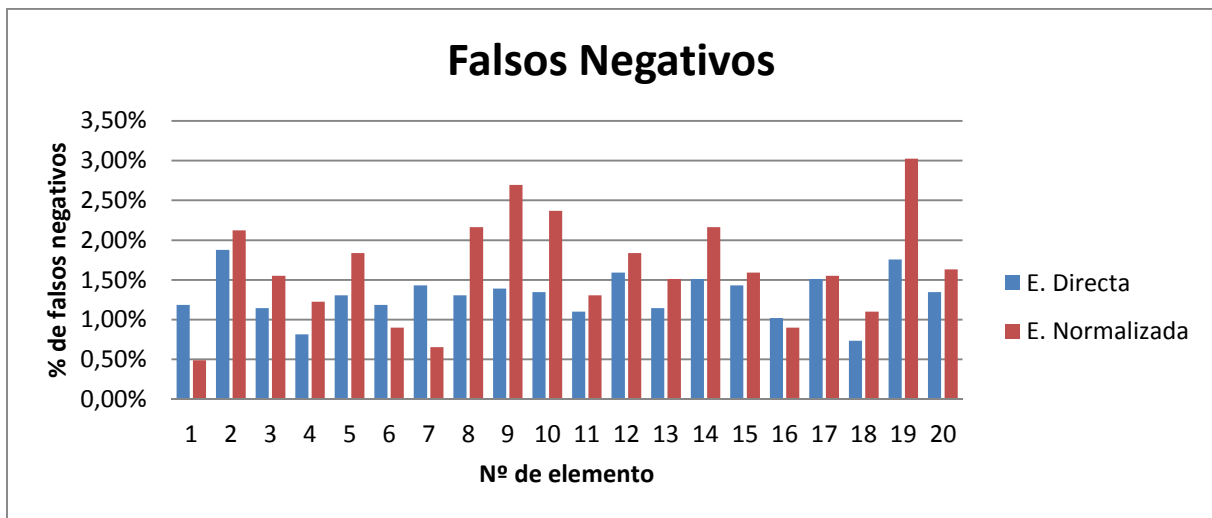


Figura 52: Número de Falsos Negativos al 2.5% (Redes con ruido).

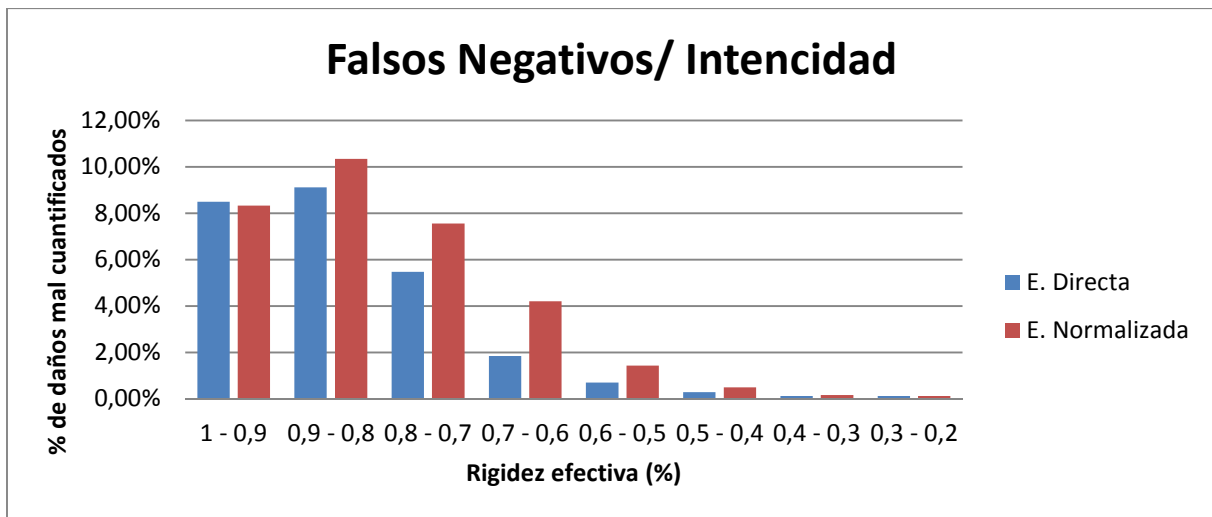


Figura 53: Número de Falsos Negativos, según rigidez efectiva al 2.5% (Redes con ruido).

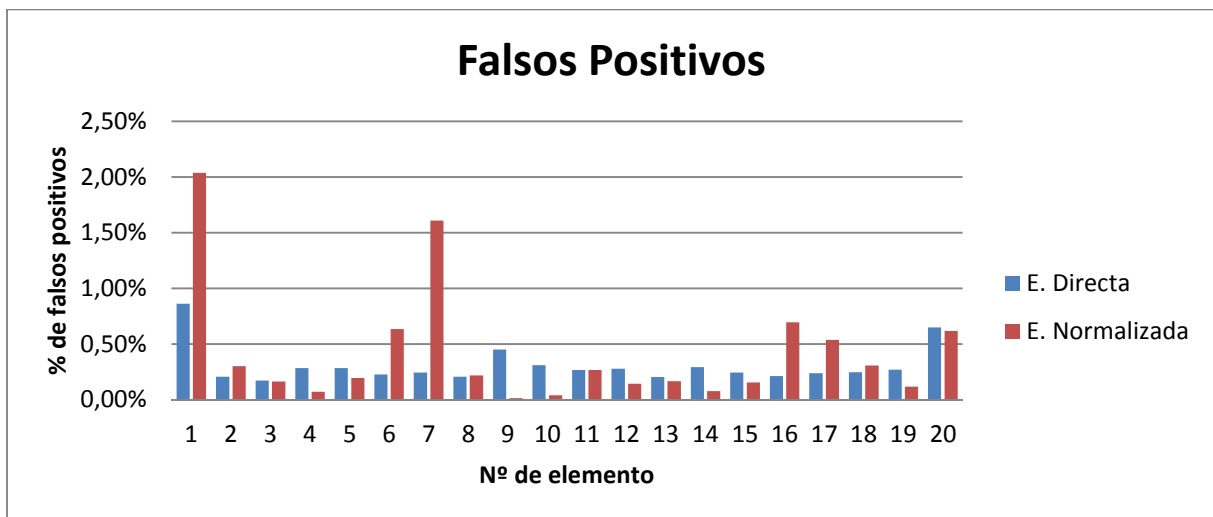


Figura 54: Número de Falsos Positivos al 2.5% (Redes con ruido).

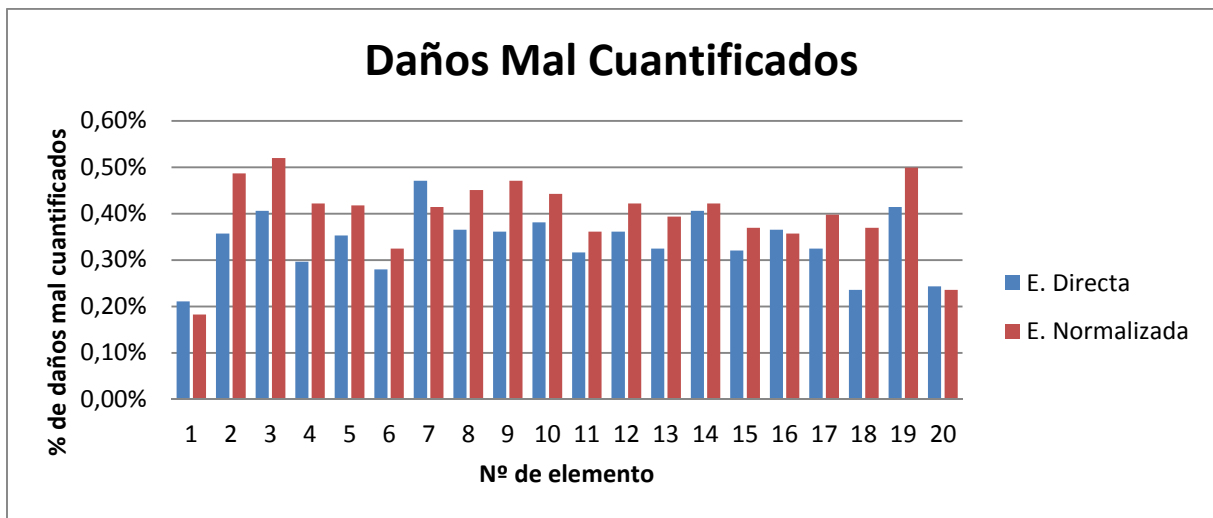


Figura 55: Daños mal cuantificado, según Nº de elemento al 5% (Redes con ruido).

5.5 Prueba Experimental (Horizontal)

Para poder probar el desempeño de las redes al enfrentarse a casos de daños reales se cuenta con las frecuencias antirresonantes horizontales, que corresponden a los mismos casos descritos en la sección 5.3. Los que generan las siguientes variaciones en las frecuencias antirresonantes (Anexo H).

En las figuras 56 a la 61, se presentan los daños detectados por las distintas redes para cada caso. Al igual que en los resultados presentados en la sección 5.3 las redes tienden a dar como salida patrones con múltiples elementos considerablemente dañados. La excepción mas importante para esto, son los resultados entregados por la red de entrada normalizada que fue entrenada utilizando patrones con ruido, esta red entrega resultados claros que puede ser relacionar con los cortes hechos a las barras. A pesar de esto, sus resultados no están libres de efecto del ruido, por lo que se ven

algunos falsos positivos entre los casos y un falso negativo en el caso 6. Además se puede ver que para ambos tipos de entrada la adición de ruido en los patrones de entrenamiento mejora los resultados de las redes, aunque esto sea menos visible en la red de entrada directa, la cual presenta resultados importantes en los cortes más profundos.

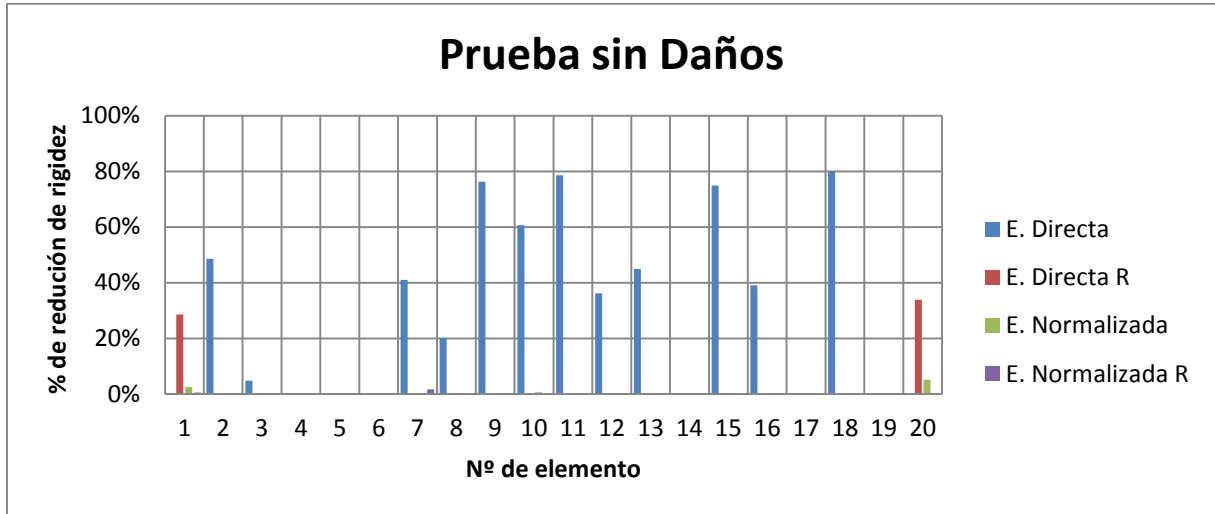


Figura 56: Resultados detección caso experimental 0 (Sin daños).

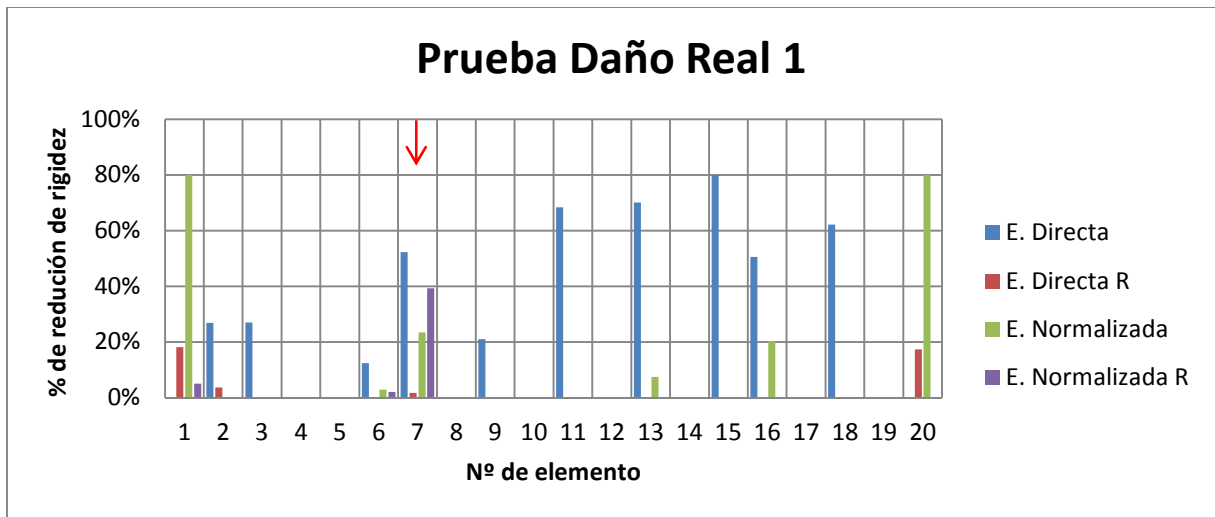


Figura 57: Resultados detección caso experimental 1 (Daño elemento 7).

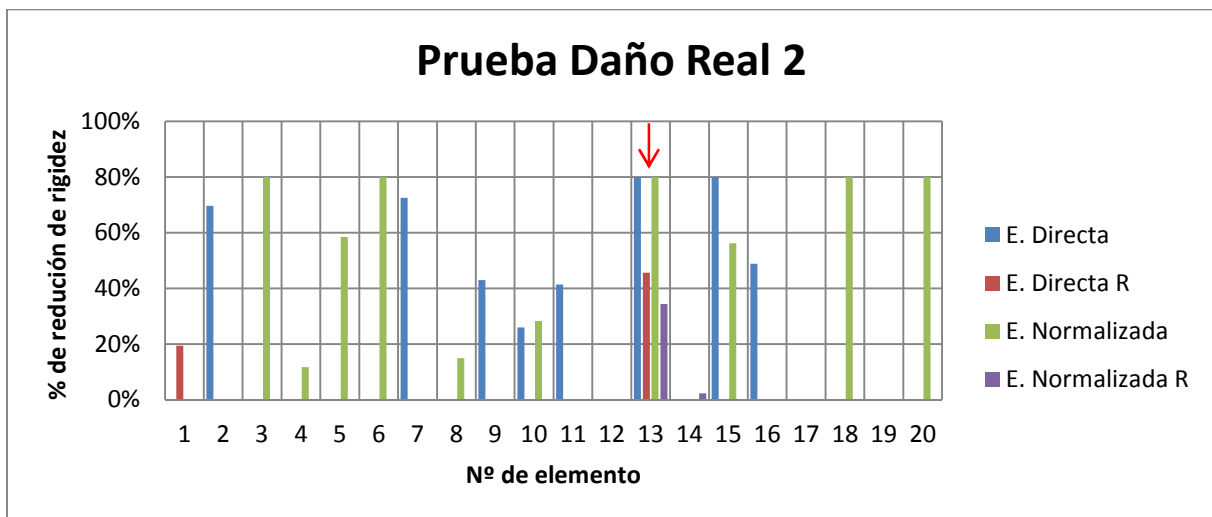


Figura 58: Resultados detección caso experimental 2 (Daño elemento 13).

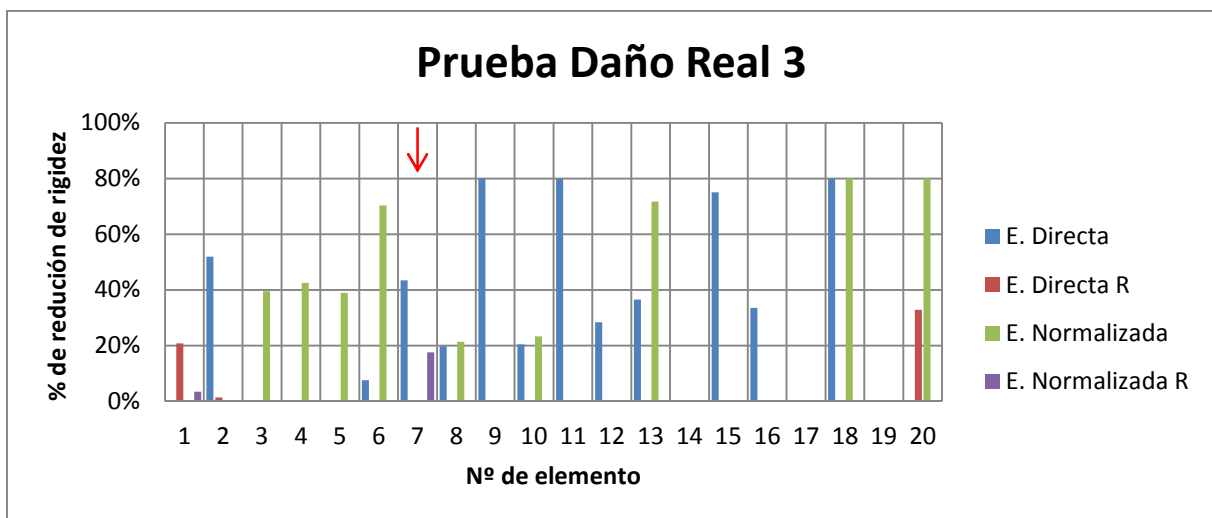


Figura 59: Resultados detección caso experimental 3 (Daño elemento 7).

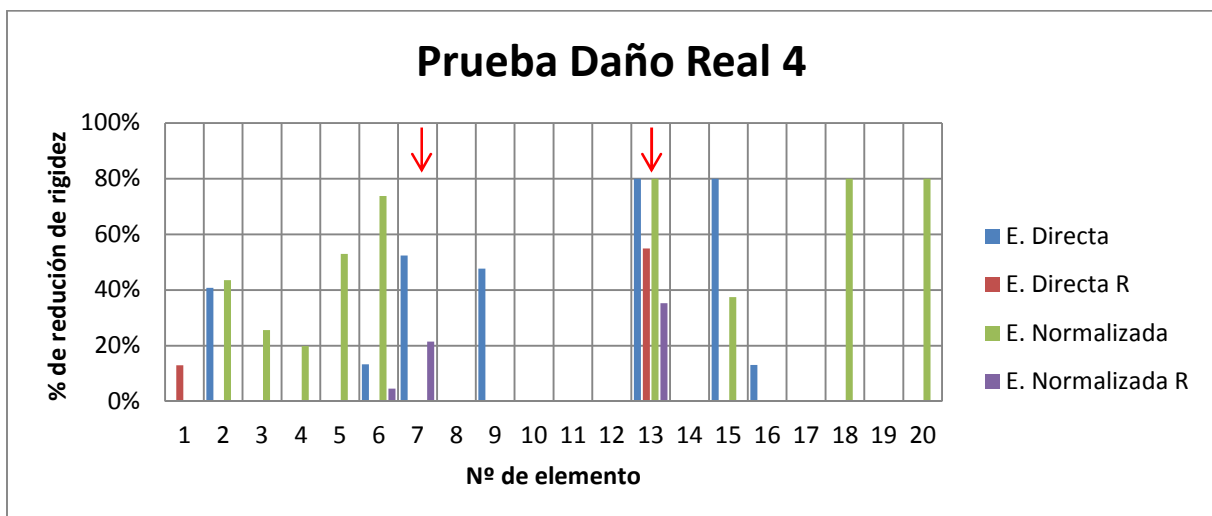


Figura 60: Resultados detección caso experimental 4 (Múltiples daños elementos 7 y 13).

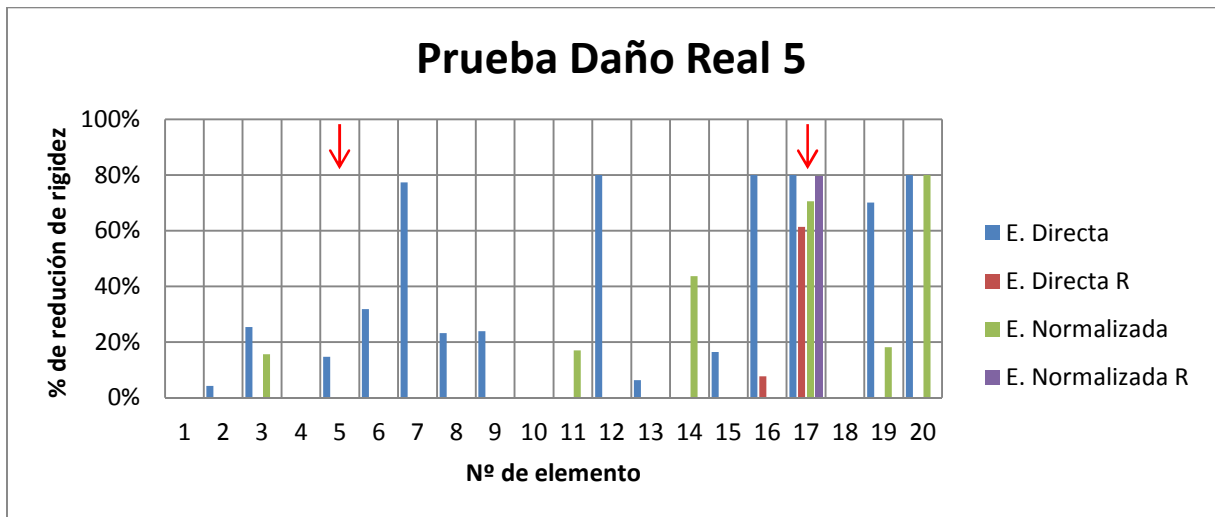


Figura 61: Resultados detección caso experimental 5 (Múltiples daños elementos 8 y 17).

5.6 Resultado Finales

En resumen, para detectar daños en esta viga los mejores resultados se obtiene al medir las vibraciones transversales de manera horizontal y usando una red normalizada con ruido. El daño detectado en cada caso se muestra en las figuras 62 a la 67, en estos resultados se aprecian 3 falso positivos y un falso negativo.



Figura 62: Resultados detección caso experimental 0 (Sin daños).

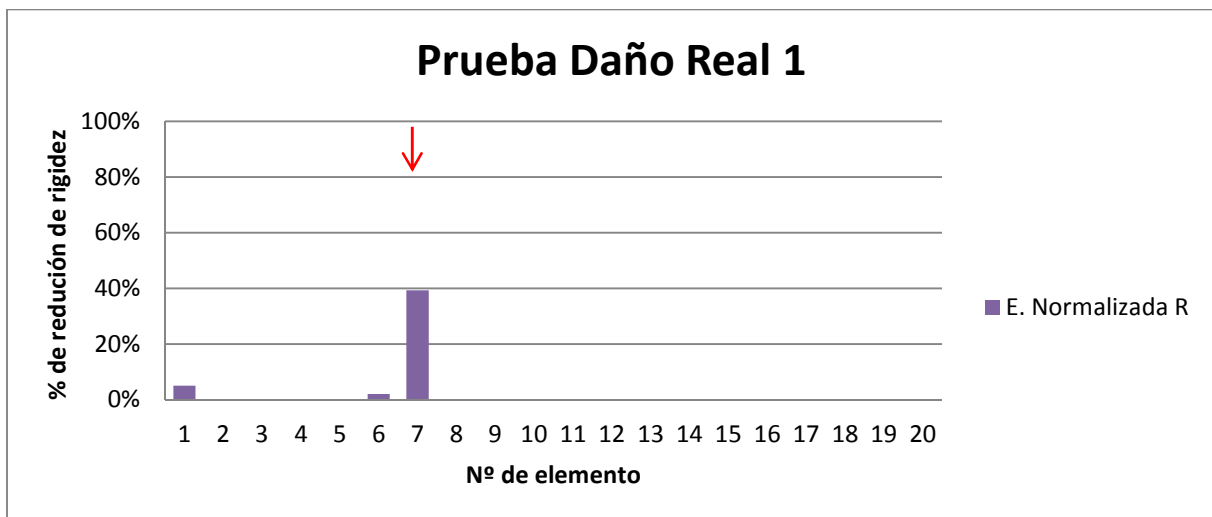


Figura 63: Resultados detección caso experimental 1 (Daño elemento 7).

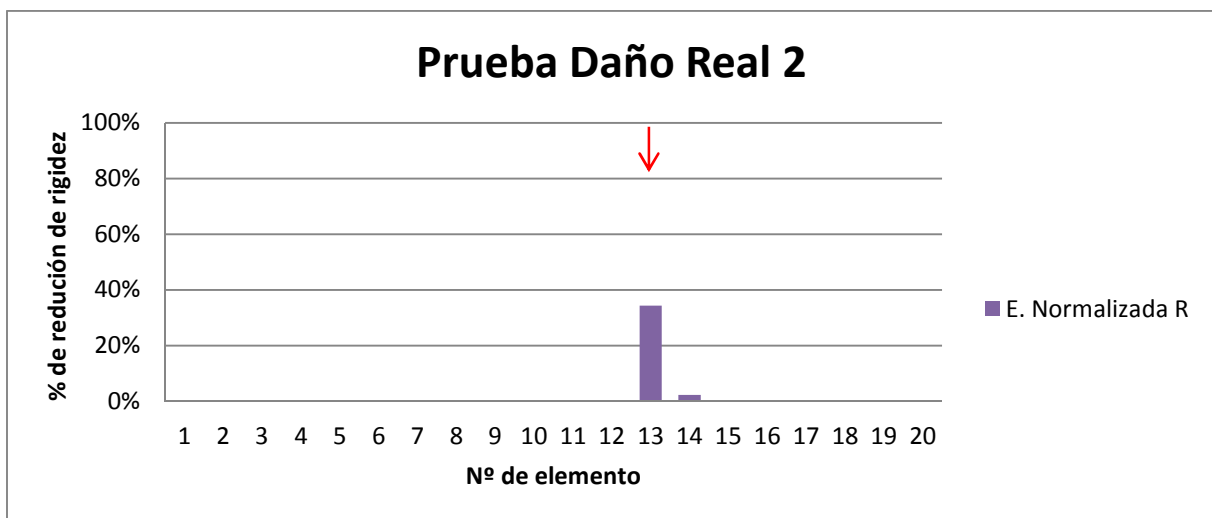


Figura 64: Resultados detección caso experimental 2 (Daño elemento 13).

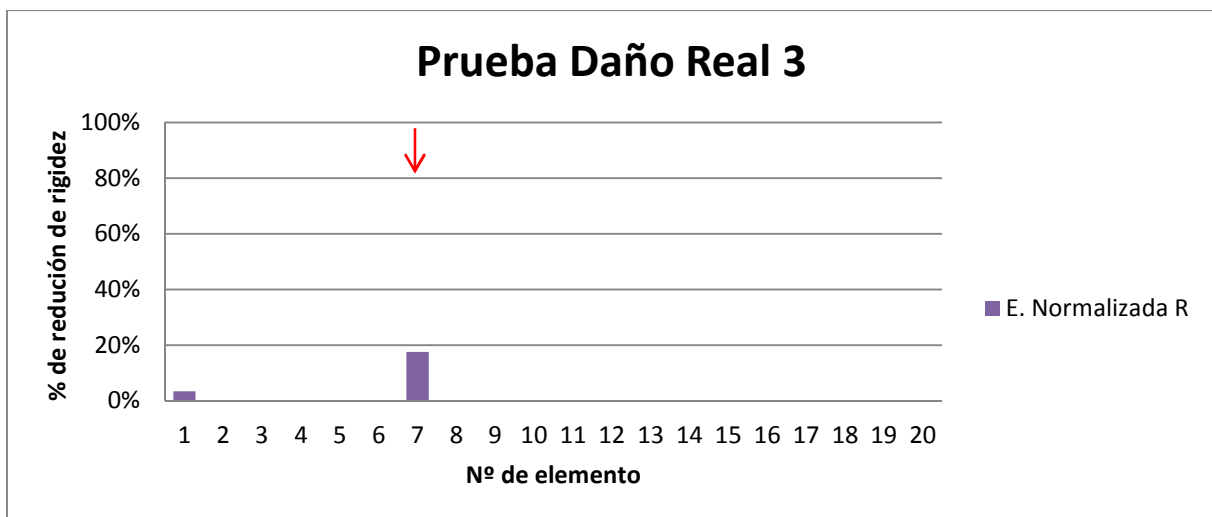


Figura 65: Resultados detección caso experimental 3 (Daño elemento 7).

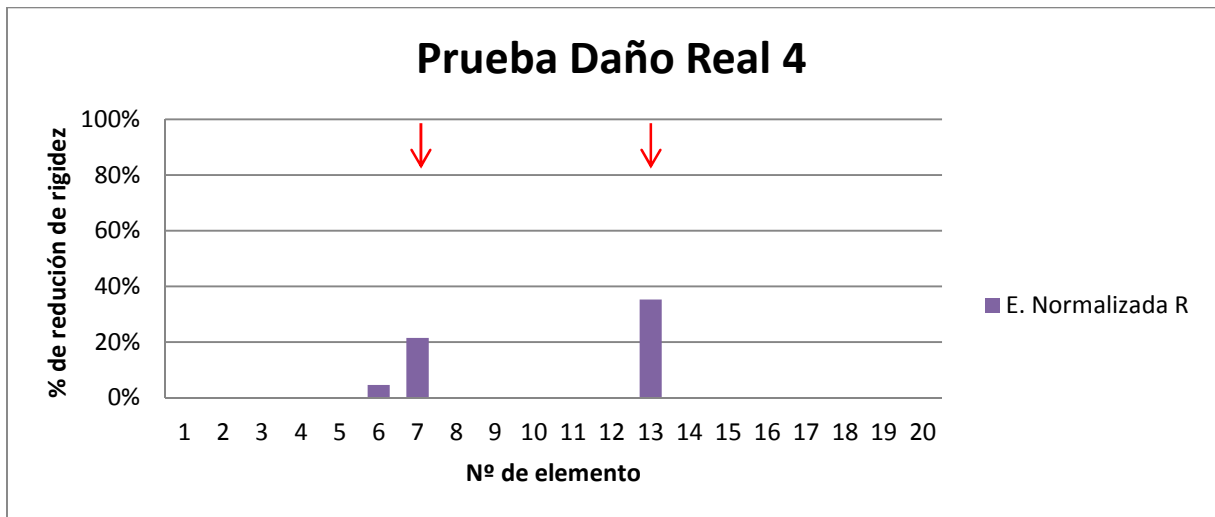


Figura 66: Resultados detección caso experimental 4 (Múltiples daños elementos 7 y 13).

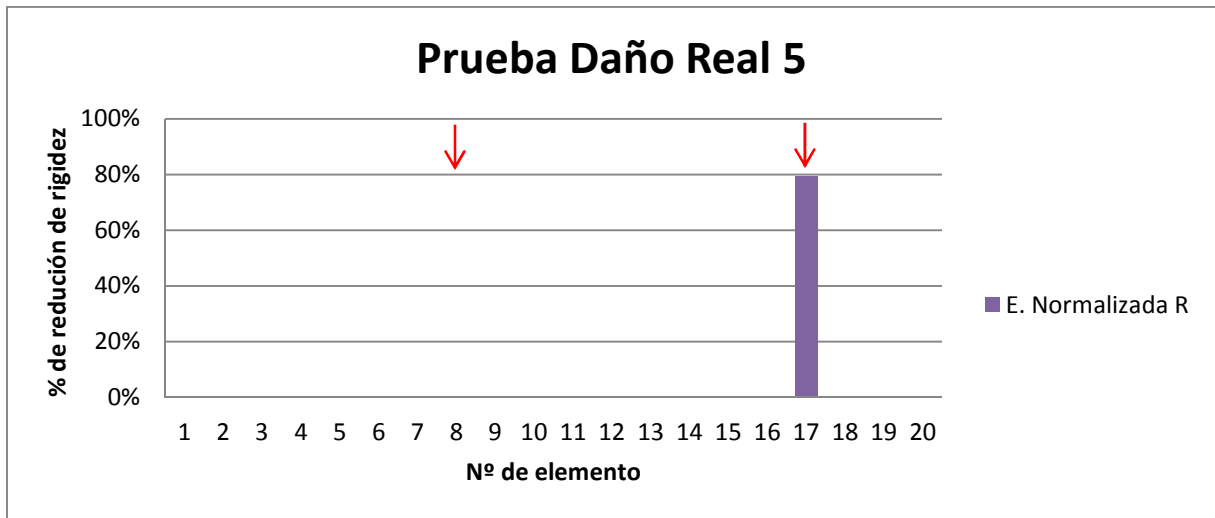


Figura 67: Resultados detección caso experimental 5 (Múltiples daños elementos 8 y 17).

6. Discusión

En este trabajo, se busca detectar hasta dos daños simultáneos para acotar el uso de recursos computacionales, ya que entrenar una red que pudiera detectar tres daños para el problema 1 es tanto o más costoso que entrenar una red que pudiera detectar dos daños en el problema 2 (19321 v/s 12321 patrones). En esta misma línea, para el problema 1 solo se contaba con datos experimentales de un daño, por lo que no se podía hacer un análisis más profundo para casos más complejos. En este trabajo, no se realizaron pruebas para mejorar la capacidad de generalización de las redes elegidas, porque en el problema 1 se consigue una detección bastante certera del caso experimental disponible. Mientras que en el problema 2 entrenar redes con un número mayor de neuronas en la capa oculta toma más tiempo del que se considera en este trabajo y aumentar el número de patrones de entrenamiento también afectaría negativamente los tiempo de entrenamiento.

De los resultados obtenidos, se puede observar que las frecuencias antiresonantes contienen información utilizable para detectar daños, tanto analítica como experimentalmente. Del mismo modo, se puede constatar que es posible entrenar RNAs para resolver casi perfectamente la modelación inversa correspondiente a la detección de daños, a nivel analítico (Anexo G). También, se puede observar que todas las redes entrenadas se comportan de manera muy similar en cuanto a la evolución del error de validación (MSE), por lo que los resultados de un problema simple se pueden aplicar a problemas más complejos.

Analizando de manera más particular los resultados del problema 1 se puede observar que, a pesar de que ninguna de las redes alcanza valores pequeños en los valores totales de los distintos indicadores de desempeño (tabla 13), todas son capaces de realizar una estimación bastante cercana al enfrentarse a la detección de daños en el caso real. Al utilizar datos experimentales, las redes mostraron errores menores al 6% en la cuantificación del daño y apariciones de falsos daños menores al 15% de magnitud. Para ambos tipos de entrada, agregar ruido a los patrones de entrenamiento produce una mejora en la detección de daños reales, siendo el mejor resultado el obtenido al utilizar una red de entrada directa, en la que se consigue una detección casi perfecta, tanto en magnitud como en número de daños detectados. Estos resultados se pueden explicar por la simpleza del problema, lo que genera un modelo muy directo cuyas diferencias con la estructura real son muy bajas.

Tabla 13: Indicadores de desempeño global, problema 1.

Tipo de entrada	Falsos Negativos		Falso Positivos		Daños Mal Cuantificados	
	Sin ruido	Con ruido	Sin ruido	Con ruido	Sin ruido	Con ruido
E. Directa	12,77%	12,77%	15,72%	13,60%	20,92%	15,00%
E. Normalizada	27,37%	19,71%	23,37%	22,52%	22,55%	18,88%

Al analizar los resultados obtenidos en el primer acercamiento al problema 2 (solo considerando vibraciones verticales), se puede notar que hay un aumento de fallas en la detección de valores extremos (falsos positivos y negativos), mientras que hay una disminución en el número de daños mal cuantificados en todo el rango de daños posibles (tabla 14). Esto último se puede deber a que existe un mayor cantidad de locaciones a analizar por lo que el promedio tiende a bajar. En este caso, no hay gran capacidad de detección de daños reales, debido a que existe una menor variación en las frecuencias antiresonantes (Anexo H). Ello sugiere que las frecuencias antiresonantes son más sensibles a las grietas en una dirección y que, en este caso, la variación que se necesita detectar es comparable con los niveles de ruido presentes en la medición.

Tabla 14: Indicadores de desempeño global, problema 2 (Vertical).

Tipo de entrada	Falsos Negativos		Falso Positivos		Daños Mal Cuantificados	
	Sin ruido	Con ruido	Sin ruido	Con ruido	Sin ruido	Con ruido
E. Directa	37,91%	24,96%	33,80%	7,41%	9,03%	6,65%
E. Normalizada	56,50%	31,86%	20,12%	10,86%	9,16%	7,53%

Por otro lado, al analizar el problema 2 utilizando frecuencias horizontales, se puede notar que, a pesar de que es un problema igual al anterior (la razón en las frecuencias antiresonantes es una constante), no se sigue una tendencia total con los resultados anteriores. Este problema muestra similitudes en cuanto al error global de la red (tabla 10 y 12) y en cuanto a la cantidad total de daños mal cuantificados (tabla 14 y 15), pero muestra un mejor desempeño en cuanto a detección de falsos positivos y negativos. Las diferencias que aparecen en estos dos indicadores, se pueden explicar por la aleatoriedad del ruido introducido a los distintos patrones. Debido a la aleatoriedad de éste hay una componente de los datos que no puede ser predicha por la red (esto también explica el aumento en el error global de las redes entrenadas con ruido), este componente es una fracción del ruido introducido y es lo que potencia la capacidad de generalización de la red. Dado que esta fracción cambia en cada grupo de patrones con ruido (incluso si tiene la misma tasa máxima de ruido), la capacidad de generalización que un patrón es capaz de entregar a una red puede ser distinta de la capacidad de generalización necesaria por la red para estimar otro patrón con ruido. Además, puede ocurrir que haya distintas fracciones para distintas porciones del conjunto de entrenamiento o validación, lo que podría causar que haya malos desempeños en la detección de daños particulares. Esto puede explicar por qué no se

observa una tendencia clara en los dos enfoques para el problema 2 e, incluso, puede explicar por qué el problema 1 a pesar de tener peores índices de desempeño, puede realizar casi perfectamente la detección del caso real.

Tabla 15: Indicadores de desempeño global, problema 2 (Horizontal).

Tipo de entrada	Falsos Negativos		Falso Positivos		Daños Mal Cuantificados	
	Sin ruido	Con ruido	Sin ruido	Con ruido	Sin ruido	Con ruido
E. Directa	32,56%	26,14%	37,01%	6,15%	8,91%	6,79%
E. Normalizada	28,51%	32,64%	44,03%	8,36%	9,01%	7,96%

Al realizar la prueba con daños reales, en las redes entrenadas para el problema 2 horizontal, se puede apreciar una detección clara en los resultados entregados por la red de entrada normalizada entrenada con ruido. Esta red es capaz de identificar la mayoría de daños objetivos, salvo el daño en el octavo elemento del caso 6. A pesar de la buena detección de los daños buscados, aparecen 3 casos de falsos positivos entre 3% y 6% de magnitud, de los cuales uno está en la posición adyacente al daño buscado. Esto puede deberse más que al ruido de la red, al hecho de que no se ha considerado el alcance que puede tener el efecto del corte para determinación de la rigidez local.

Por otro lado, aunque la red de entrada directa, tiene un mejor desempeño con datos simulados que la red normalizada, no logra generar una buena detección con los datos experimentales. De hecho, solo en el primer problema estas redes logran detectar un caso de daño real. El buen desempeño de la red directa en el problema 1 se debe a que el comportamiento del modelo analítico y la estructura son muy similares. Esto es consistente con los resultados obtenidos y se debe a la menor cantidad de simplificaciones hechas al modelo. Mientras que las redes de entrada normalizada tienen una mejor capacidad para controlar el ruido experimental y las diferencias entre el modelo analítico y la estructura, salvo la aparición de algunos falsos positivos, debido a que patrón utilizado para normalizar los datos no necesariamente tiene la misma tasa de ruido que el patrón a normalizar.

Si se consideran los valores totales de los indicadores de desempeño de las redes entrenadas con ruido, se tiene que persisten un tercio de los falsos negativos y un decimo de los falsos positivos y daños mal cuantificados. Estos indicadores no garantizan una buena detección de daños como la observada en los resultados finales de la sección 5.6. Esto se puede explicar porque las formulas de estos indicadores no son sensibles a dos posibles errores en la detección de daños. Primero, que ante la presencia de dos daños, uno mayoritariamente más severo, este último esconda o absorba la influencia del daño menor en la respuesta dinámica, como ocurre en el caso experimental 5 horizontal (Figura 67). A pesar de que esto es una falla en la detección de daños, no es tan grave porque existe la detección de un daño real importante que va a desencadenar la intervención o examinación de la estructura. Por otro lado, existe la

posibilidad de que la red detecte el daño en un elemento adyacente al daño real, como ocurre en el caso experimental 2 vertical al utilizar la red directa con ruido (Figura 44). Esto influye negativamente en los tres indicadores de desempeño, pero de todos modos se tiene una presunción de la localización del daño, como en este trabajo, lo que genera una intervención que encontrará y tratará el daño real.

Al examinar de manera general todos los resultados presentados, se puede observar que la introducción de una tasa de ruido semi-arbitrario, genera mejoras en la localización y detección de daños. Si se considera el promedio de la diferencia absoluta y la diferencia máxima entre la frecuencia antiresonantes medidas y las predichas por los modelos ajustados, se obtienen los valores presentados en la tabla 16.

Tabla 16: Diferencias promedio y máximas entre los modelos y las estructuras.

	Diferencia Promedio	Diferencia Máxima
Problema 1	0,48%	0,70%
Problema 2V	0,76%	3,90%
Problema 2H	1,29%	3,45%

Se puede observar que a pesar de que la cota establecida es superior a valores promedio, solo se puede lograr una buena detección de daños reales para los casos en que ruido tiene poca variabilidad, problema 1 y problema 2H. Esto plantea la necesidad de modelar de mejor manera el ruido a introducir en los patrones de la red y buscar una relación entre el ruido existente en los datos experimentales y el valor óptimo para la tasa de ruido a introducir.

7. Conclusiones

Las redes neuronales en conjunto con la utilización de las frecuencias antiresonantes de una estructura han mostrado ser capaces de detectar daños en casos experimentales, aún con presencia de algunos falsos positivos y negativos. A pesar de los extensos tiempos de entrenamiento ocupados en la realización de este trabajo, obtener datos de la red ya entrenada es muy rápido y dependiendo del problema se alcanzan grandes niveles de precisión. Sin embargo, aún existe holgura para mejorar estos resultados, por lo que es necesario seguir trabajando en la búsqueda de una topología y método de entrenamiento de redes neuronales que se adapte mejor al problema.

A través del método de elementos finitos es posible determinar las frecuencias resonantes y antiresonantes de la estructura de estudio, además estos resultados se pueden mejorar al ajustar las propiedades iniciales del modelo en base a datos experimentales.

Considerando los resultados de los dos problemas estudiados en este trabajo se puede concluir que las redes que mejor se adaptan al problema de detección de daños usando antiresonancia, son las que cumplen las siguientes características:

- Tipo: MLP
- N° de capas ocultas: 1
- Función de transferencia en la capa oculta: Sigmoidea tangente
- Función de transferencia en la capa de salida: Identidad saturada simétrica
- Método de entrenamiento: Levenberg-Marquardt
- Patrón de entrenamiento: Entrada normalizada, con introducción de ruido

Estas redes son capaces de detectar daños a partir de datos simulados y experimentales.

8. Trabajos Futuros

Al igual que en el presente trabajo, la realización de otros similares dependen directamente de los avances que se logren en las siguientes áreas:

- Redes neuronales y métodos de entrenamiento.
- Ajustes de modelos.
- Reducción de ruido en mediciones experimentales.

En términos más específicos, se desprende de los resultados de este trabajo las siguientes áreas o líneas de estudio.

1. Profundización del estudio de la capacidad de las redes neuronales para detectar un mayor número de daños simultáneos en una estructura (tres o más).
2. Utilización de otros tipos, topologías o métodos de entrenamiento de RNAs para mejorar los resultados obtenidos en el presente trabajo. Por ejemplo: MPL con dos o más capas ocultas.
3. Dado que hasta ahora se ha visto una mayor sensibilidad de las frecuencias antiresonantes en una dirección por sobre otras, se propone la realización de estudios sobre la direccionalidad en la detección de daños estructurales.
4. Formas de aplicación de redes neuronales para la detección de daños en estructuras más complejas. Debido a su mayor tamaño se requiere priorizar en qué elementos es más relevante la detección de daños y es necesario determinar un número óptimo de elementos a analizar.
5. En la misma línea del punto anterior, en estructuras complejas optimizar la ubicación de los sensores para obtener la mayor cantidad de información sobre los daños presentes.
6. Siendo que para modelar estructuras complejas se requieren un mayor número de entradas y salidas en la red, analizar si es más efectivo aumentar el número de mediciones o utilizar más información de las mediciones realizadas, como por ejemplo, utilización de las frecuencias resonantes en conjunto con las frecuencias antiresonantes.
7. Utilización de funciones de frecuencias indirectas en estructurales reales en que se tenga una fuente de excitación fija, como ocurre en el caso del soporte de un equipo en funcionamiento.
8. Mejorar la generación de patrones para modelar la detección en casos reales, por ejemplo, encontrar una relación entre el ruido existente en los datos y el ruido introducido a los patrones de entrenamiento.

9. Referencias

- [1] MERUANE V. "Damage detection based upon experimental and numerical dynamic analysis techniques for mechanical structures". Tesis (Doctor en Ciencia mención Ing. Mecánica). Lovaina, Bélgica. Katholieke Universiteit Leuven, Facultad de Ciencias Aplicadas, 2010. 237h.
- [2] LUNA A. "Detección de defectos mediante uso de redes neuronales artificiales y optimización de resultados con lógica neuro difusa". Tesis (Doctor en Ciencia mención Ing. Mecánica). México DF, México. Instituto Politécnico Nacional, Escuela Superior de Ingeniería Mecánica y Eléctrica, 2010. 155h.
- [3] FANG X., LUO H. y TANG J. "Structural damage detection using neural network with learning rate improvement". Computers and Structures 83: 2150-2161, 2005.
- [4] NAM D., CHOI S., PARK S. y STUBBS N. "Improved parameter identification using additional spectral information". International Journal of Solid and Structures 42: 4971-4987, 2005.
- [5] D'AMBROGIO W. y FREGOLENT A. "The use of antiresonances for robust model updating". Journal of Sound and Vibration 236 (2). 227-243, 2000.
- [6] MERUANE V. "Apunte vibraciones mecánicas, ME706" [En línea] <<http://viviana.meruane.com/>> [consulta: abril 2012]
- [7] MERUANE V. "Apunte dinámica estructural, ME706" [En línea] <<http://viviana.meruane.com/>> [consulta: diciembre 2011]
- [8] MERUANE V. "Model updating using antiresonant frequencies identified from transmissibility functions". Journal of Sound and Vibration. Submitted after revision. 2012.
- [9] FAUSETT L. "Fundamentals of neural networks: architectures, algorithms and applications". Prentice Hall, 1993.
- [10] MATSUOKA K., "Noise injection into inputs in back-propagation learning". Systems, Man and Cybernetics 22: 436-440, 2002.
- [11] BISHOP C., "Neural networks for pattern recognition". Oxford, 1995.
- [12] CONGRESO Iberoamericano de Ingeniería Mecánica (8º, 2007, Cuzco, Perú). "Detección de defectos mediante redes neuronales artificiales en evaluaciones hechas en dominio de la frecuencia".
- [13] YUN C., YI J., y BAHNG E. "Joint damage assessment of framed structure using a neural networks technique" Engineering Structure 23. 425-435, 2001.

- [14] DUFFEY T.A., DOEBLING S.W., FARRAR C.R., BAKER W.E., RHEE W.H., AND DOEBLING S.W. "Vibration-Based Damage Identification in Structures Exhibiting Axial and Torsional Response" Journal of Vibration and Acoustics 123. 84-91, 2001.
- [15] BEALE M., HAGAN M. y DEMUTH H. "Neural network toolbox™, User's guide". 2011
- [16] "Matlab's product documetation" [En línea] <<http://www.mathworks.com/help/toolbox/stats/lhsdesign.html>> [consulta: abril 2012]

10. Anexo A

```
function [Wa]=MasaResorte(Beta)
%Función que recibe el vector de rigidez efectiva
%Datos
k1=56.7*1000;
m1=0.5593;
m2=0.4194;
k=zeros(2,2,7);
M=zeros(8);

%Carga de valores de ajuste de modelo
load km
kc=km(1:7).*k1;
mc=km(8:15).*[m1 m2*ones(1,7)];

%Matriz de rigidez y masa para (2 masas y un resorte)
for i=1:7
    k(:,i)=kc(i)*[1 -1;
                 -1 1];
    M(i+1,i+1)=mc(i+1);
end
M(1,1)=mc(1);

%Procedimiento de ensamblaje K
for i=1:7
    locel(i,1:2)=(i):(i+1);
end

%Matriz de rigidez global, considerando la presencia de daños
K=zeros(8,8);
for i=1:7
    K(locel(i,:),locel(i,:))=K(locel(i,:),locel(i,:)) + (Beta(i))*k(:,i);
end

%Cálculo de frecuencia antiresonantes para cada masa
Wa=zeros(7,1);
j=1:8;
i=1;
aux=setdiff(j,i);
Ma=M(2:8,aux);
Ka=K(2:8,aux);
[~,wa]=eig(Ka,Ma);
wa=sqrt(diag(wa))/2/pi;
Wa(:,i)=sort(real(wa));
```


11. Anexo B

```
% Rutina de entrenamiento de RNAs
clear all
clc
NP=1401;
Betas= ones(NP,7);
% Un daño aleatorio en una secciones
for i=1:7
    Betas((8*(i-1)+2),i)=.9;
    Betas((8*(i-1)+3),i)=.8;
    Betas((8*(i-1)+4),i)=.7;
    Betas((8*(i-1)+5),i)=.6;
    Betas((8*(i-1)+6),i)=.5;
    Betas((8*(i-1)+7),i)=.4;
    Betas((8*(i-1)+8),i)=.3;
    Betas((8*(i-1)+9),i)=.2;
end
l=58;
% Un daño aleatorio en dos secciones
for i=1:7
    for j=i+1:7
        a=.9;
        for k=1:8
            Betas(l,i)=a;
            Betas(l,j)=.9;
            Betas(l+1,i)=a;
            Betas(l+1,j)=.8;
            Betas(l+2,i)=a;
            Betas(l+2,j)=.7;
            Betas(l+3,i)=a;
            Betas(l+3,j)=.6;
            Betas(l+4,i)=a;
            Betas(l+4,j)=.5;
            Betas(l+5,i)=a;
            Betas(l+5,j)=.4;
            Betas(l+6,i)=a;
            Betas(l+6,j)=.3;
            Betas(l+7,i)=a;
            Betas(l+7,j)=.2;
            l=l+8;
            a=a-.1;
        end
    end
end
end

% Cálculo de frecuencias antiresonantes
Fae=zeros(NP,7);
for i=1:NP
    Fae(i,:)=MasaResorte(Betas(i,:));
end
```

```

%Introducción de ruido (Opcional dependiendo del conjunto de entrenamiento deseado)
ruido=2;
a=-ruido/100;
b=ruido/100;
Fae=( 1 + (a + (b-a)*rand(size(Fae)))).*Fae;

%Normalización de entradas (Opcional dependiendo del conjunto de entrenamiento deseado)
for i=2:NP
    Fae(i,:)=(Fae(1,:)-Fae(i,:))./Fae(1,:);
end
Fae(1,:)=zeros(1,7);

%Permutación de patrones de entrenamiento
N=randperm(NP);
Betas=Betas(N,:);
Fae=Fae(N,:);
Betas=Betas';
Fae=Fae';

%Entrenamiento de RNAs
perf=inf(30,1);
vperf=inf(30,1);
for i=10:10:200
    redn=['red' num2str(i)];
    trn=['tr' num2str(i)];
    for k=1:10
        net = newfit(Fae,Betas,[i]);
        net.trainFcn='trainlm';
        net.layers{2}.transferFcn = 'satlins';
        net.trainParam.epochs = 10000;
        net.divideParam.trainRatio = .85;
        net.divideParam.valRatio = .10;
        net.divideParam.testRatio = .05;
        [net,tr] = train(net,Fae,Betas);
        if tr.vperf(tr.best_epoch+1) < vperf(k)
            vperf(k)=tr.vperf(tr.best_epoch+1);
            perf(k)=tr.perf(tr.best_epoch+1);
            assignin('base',redn,net);
            assignin('base',trn,tr);
        end
    end
    save Filename.mat
end
end

disp('Entrenamiento Terminado')

```

12. Anexo C

```
function Wa=vigalV(Beta)
%Función que recibe el vector de rigidez efectiva
%Datos
load km;
rho=(1+km(1))*7800;
E=(1+km(2))*2.1e11;
le=.05;
A=.025*.01;
Iz=.025*(.01)^3/12;;

%Matriz de masa de un elemento
m=rho*A*le/420*[156 22*le 54 -13*le;
                22*le 4*le^2 13*le -3*le^2;
                54 13*le 156 -22*le;
                -13*le -3*le^2 -22*le 4*le^2];

%matriz de rigidez de un elemento
k=E*Iz/le^3*[12 6*le -12 6*le;
             6*le 4*le^2 -6*le 2*le^2;
             -12 -6*le 12 -6*le;
             6*le 2*le^2 -6*le 4*le^2];

%Ensamble
K=zeros(42,42);
M=zeros(42,42);

for i=1:20
    K((1+2*(i-1):(2*(i+1)),(1+2*(i-1):(2*(i+1))))=K((1+2*(i-1):(2*(i+1)),(1+2*(i-1):(2*(i+1))))+Beta(i)*k;
    M((1+2*(i-1):(2*(i+1)),(1+2*(i-1):(2*(i+1))))=M((1+2*(i-1):(2*(i+1)),(1+2*(i-1):(2*(i+1))))+m;
end

Wa=zeros(5,5);
j=1:42;

for i=1:10:41
    aux=setdiff(j,i);
    Ma=M(aux,aux);
    Ka=K(aux,aux);
    [~,wa]=eig(Ka,Ma);
    wa=sqrt(diag(wa))/2/pi;
    Wa(:,1+round(i/10))=sort(real(wa(3:7,1)));
end
Wa=[Wa(:,1); Wa(:,2); Wa(:,4); Wa(:,5)];
```

```

function Wa=vigalH(Beta)
%Función que recibe el vector de rigidez efectiva
%Datos
load km;
rho=(1+km(1))*7800;
E=(1+km(2))*2.1e11;
le=.05;
A=.025*.01;
%La inercia es lo único que cambia al relsover el problema horizontal
Iz=.010*(.025)^3/12;

%Matriz de masa de un elemento
m=rho*A*le/420*[156 22*le 54 -13*le;
                22*le 4*le^2 13*le -3*le^2;
                54 13*le 156 -22*le;
                -13*le -3*le^2 -22*le 4*le^2];

%matriz de rigidez de un elemento
k=E*Iz/le^3*[12 6*le -12 6*le;
            6*le 4*le^2 -6*le 2*le^2;
            -12 -6*le 12 -6*le;
            6*le 2*le^2 -6*le 4*le^2];

%Ensamble
K=zeros(42,42);
M=zeros(42,42);

for i=1:20
    K((1+2*(i-1)):(2*(i+1)),(1+2*(i-1)):(2*(i+1)))=K((1+2*(i-1)):(2*(i+1)),(1+2*(i-1)):(2*(i+1)))+Beta(i)*k;
    M((1+2*(i-1)):(2*(i+1)),(1+2*(i-1)):(2*(i+1)))=M((1+2*(i-1)):(2*(i+1)),(1+2*(i-1)):(2*(i+1)))+m;
end

Wa=zeros(5,5);
j=1:42;

for i=1:10:41
    aux=setdiff(j,i);
    Ma=M(aux,aux);
    Ka=K(aux,aux);
    [~,wa]=eig(Ka,Ma);
    wa=sqrt(diag(wa))/2/pi;
    Wa(:,1+round(i/10))=sort(real(wa(3:7,1)));
end
Wa=[Wa(:,1); Wa(:,2); Wa(:,4); Wa(:,5)];

```

13. Anexo D

```
% Rutina de entrenamiento de RNAs
clear all
clc
NP=12321;
Betas= ones(NP,20);
% Un daño aleatorio en una secciones
for i=1:20
    Betas((8*(i-1)+2),i)=.9;
    Betas((8*(i-1)+3),i)=.8;
    Betas((8*(i-1)+4),i)=.7;
    Betas((8*(i-1)+5),i)=.6;
    Betas((8*(i-1)+6),i)=.5;
    Betas((8*(i-1)+7),i)=.4;
    Betas((8*(i-1)+8),i)=.3;
    Betas((8*(i-1)+9),i)=.2;
end
l=162;
% Un daño aleatorio en dos secciones
for i=1:20
    for j=i+1:20
        a=.9;
        for k=1:8
            Betas(l,i)=a;
            Betas(l,j)=.9;
            Betas(l+1,i)=a;
            Betas(l+1,j)=.8;
            Betas(l+2,i)=a;
            Betas(l+2,j)=.7;
            Betas(l+3,i)=a;
            Betas(l+3,j)=.6;
            Betas(l+4,i)=a;
            Betas(l+4,j)=.5;
            Betas(l+5,i)=a;
            Betas(l+5,j)=.4;
            Betas(l+6,i)=a;
            Betas(l+6,j)=.3;
            Betas(l+7,i)=a;
            Betas(l+7,j)=.2;
            l=l+8;
            a=a-.1;
        end
    end
end
end

% Cálculo de frecuencias antiresonantes
Fae=zeros(20,NP);
for i=1:NP
    Fae(:,i)=vigaV(Betas(i,:)); % Aquí se debe elegir entre vigaH y vigaV
end
```

```

%Introducción de ruido
ruido=2;
a=-ruido/100;
b=ruido/100;
Fae=( 1 + (a + (b-a)*rand(size(Fae)))).*Fae;

%Normalización de entradas
for i=2:NP
    Fae(i,:)=(Fae(1,:)-Fae(i,:))./Fae(1,:);
end
Fae(1,:)=zeros(1,7);

%Permutación de patrones de entrenamiento
N=randperm(NP);
Betas=Betas(N,:);
Fae=Fae(:,N);
Betas=Betas';

%Entrenamiento de RNAs
perf=inf(8,1);
vperf=inf(8,1);
%Se entrena hasta noventa para reducir tiempo de entrenamiento
for i=10:10:80
    redn=['red' num2str(i)]
    trn=['tr' num2str(i)];
    for k=1:10
        net = newfit(Fae,Betas,[i]);
        net.trainFcn='trainlm';
        net.layers{2}.transferFcn = 'satlins';
        net.trainParam.epochs = 10000;
        net.divideParam.trainRatio = .85;
        net.divideParam.valRatio = .10;
        net.divideParam.testRatio = .05;
        [net,tr] = train(net,Fae,Betas);
        if tr.vperf(tr.best_epoch+1) < vperf(i/10)
            vperf(i/10)=tr.vperf(tr.best_epoch+1);
            perf(i/10)=tr.perf(tr.best_epoch+1);
            assignin('base',redn,net);
            assignin('base',trn,tr);
        end
    end
    save Filename.mat
end
end

disp('Entrenamiento Terminado')

```

14. Anexo E

```
%Rutina de generación de patrones de validación
clear all
BV=ones(1232,20);
a=0.2+0.775*lhsdesign(16,1);
for i=1:16
    a1=round(rand*19+1);
    BV(i,a1)=a(i);
end
a1=0;
a2=0;
a=0.2+0.775*lhsdesign(1216,2);
for i=17:1232
    while a1==a2
        a1=round(rand*19+1);
        a2=round(rand*19+1);
    end
    BV(i,a1)=a(i-16,1);
    BV(i,a2)=a(i-16,2);
    a1=0;
    a2=0;
end

for i=1:length(BV)
    FaV(:,i)=vigaV(BV(i,:)); % Aquí se debe elegir entre vigaH y vigaV
end
BV=BV';

save('Vali.mat','BV','FaV')

Fa0(:,1)=viga(ones(1,20));
for i=1:length(BV)
    FaV(:,i)=(Fa0-FaV(:,i))./Fa0;
end

save('Valin.mat','BV','FaV')

load Vali.mat
ruido=2;
a=-ruido/100;
b=ruido/100;
FaV=( 1 + (a + (b-a)*rand(size(FaV)))).*FaV;

save('ValiR.mat','BV','FaV')

Fa0=( 1 + (a + (b-a)*rand(size(Fa0)))).*Fa0;
for i=1:length(BV)
    FaV(:,i)=(Fa0-FaV(:,i))./Fa0;
end

save('ValiRn.mat','BV','FaV')
```

15. Anexo F

%Rutina de indicadores de desempeño

```
clear all
load Filename.mat
load ValiRn.mat
aux=sim(red70,FaV);
FA=zeros(1,20);
DM=zeros(1,20);
DMi=zeros(1,8);
DMM=zeros(1,20);
for i=1:length(BV)
    for j=1:20
        if BV(j,i)==1
            if abs(BV(j,i)-aux(j,i)) >=0.025
                FA(j)=FA(j)+1;
            end
        else
            if abs(1-aux(j,i)) <=0.025
                DM(j)=DM(j)+1;
                if BV(j,i)>=0.9
                    DMi(k,1)=DMi(k,1)+1;
                elseif BV(j,i)>=0.8 && BV(j,i)<0.9
                    DMi(k,2)=DMi(k,2)+1;
                elseif BV(j,i)>=0.7 && BV(j,i)<0.8
                    DMi(k,3)=DMi(k,3)+1;
                elseif BV(j,i)>=0.6 && BV(j,i)<0.7
                    DMi(k,4)=DMi(k,4)+1;
                elseif BV(j,i)>=0.5 && BV(j,i)<0.6
                    DMi(k,5)=DMi(k,5)+1;
                elseif BV(j,i)>=0.4 && BV(j,i)<0.5
                    DMi(k,6)=DMi(k,6)+1;
                elseif BV(j,i)>=0.3 && BV(j,i)<0.4
                    DMi(k,7)=DMi(k,7)+1;
                elseif BV(j,i)<0.3
                    DMi(k,8)=DMi(k,8)+1;
                end
            end
            if abs(BV(j,i)-aux(j,i)) >=0.05
                DMM(j)=DMM(j)+1;
            end
        end
    end
end
end

FA=FA/22192;
DM=DM/2448;
DMi=DMi/2448;
DMM=DMM/24640;
```


16. Anexo G: Indicadores de desempeño para patrones sin ruido

A continuación se presentan los indicadores de desempeño para las redes entrenadas sin introducción de ruido, utilizando un conjunto de validación como los descritos en las secciones 4.3 y 5.2 respectivamente, esta vez sin introducción de ruido.

Problema 1:

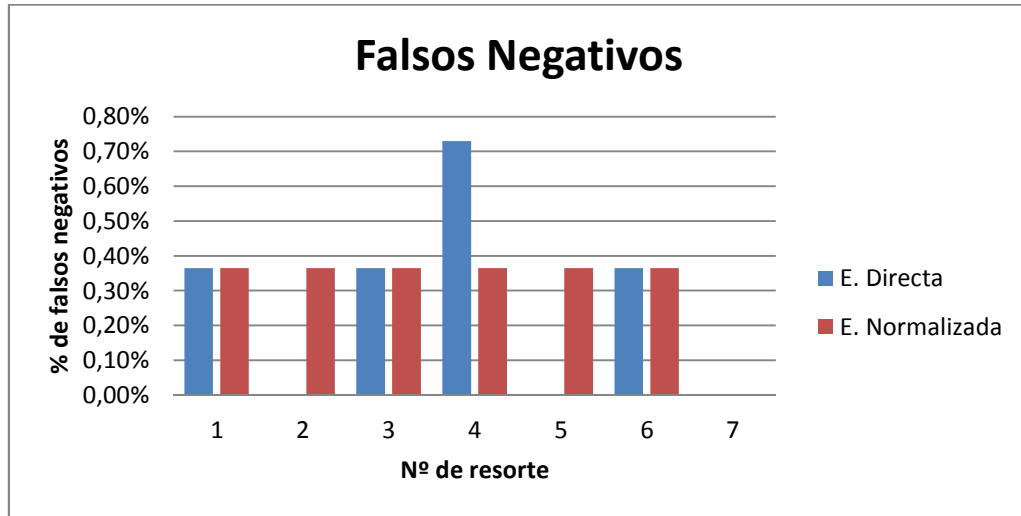


Figura 68: Número de Falsos Negativos al 2.5% (Redes sin ruido).

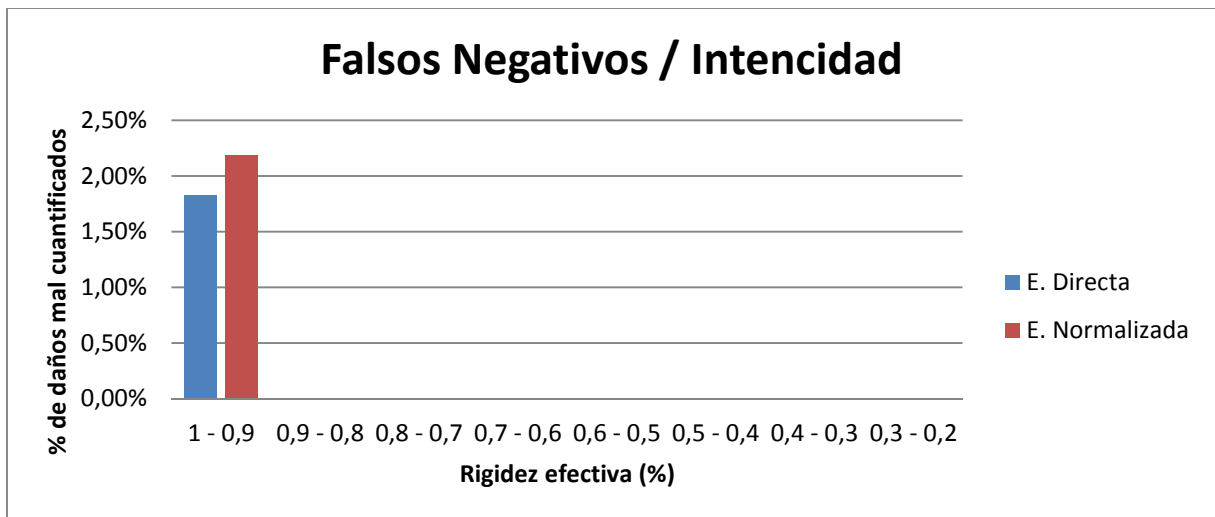


Figura 69: Número de Falsos Negativos, según rigidez efectiva al 2.5% (Redes sin ruido).

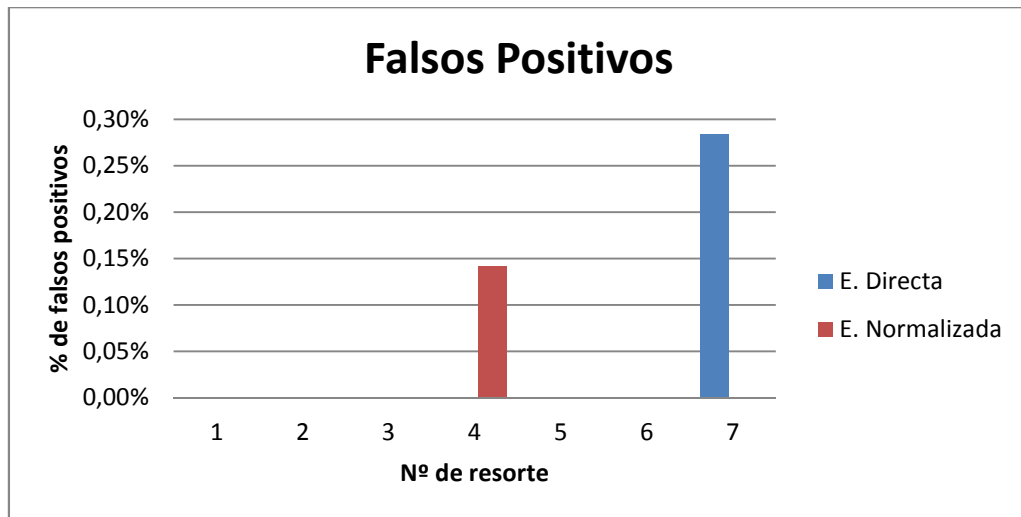


Figura 70: Número de Falsos Positivos al 2.5% (Redes sin ruido).

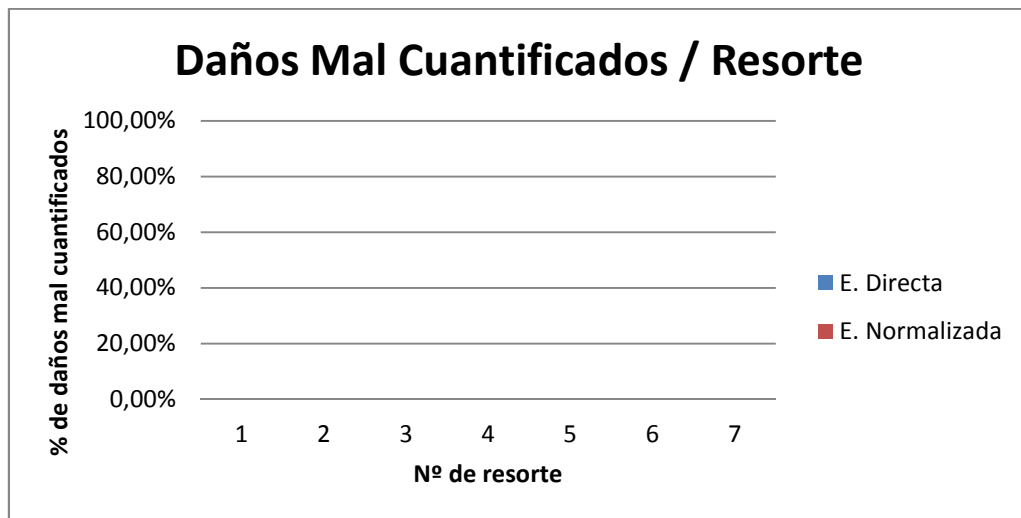


Figura 71: Daños mal cuantificado, según nº de resorte al 5% (Redes sin ruido).

Problema 2 (Vertical)

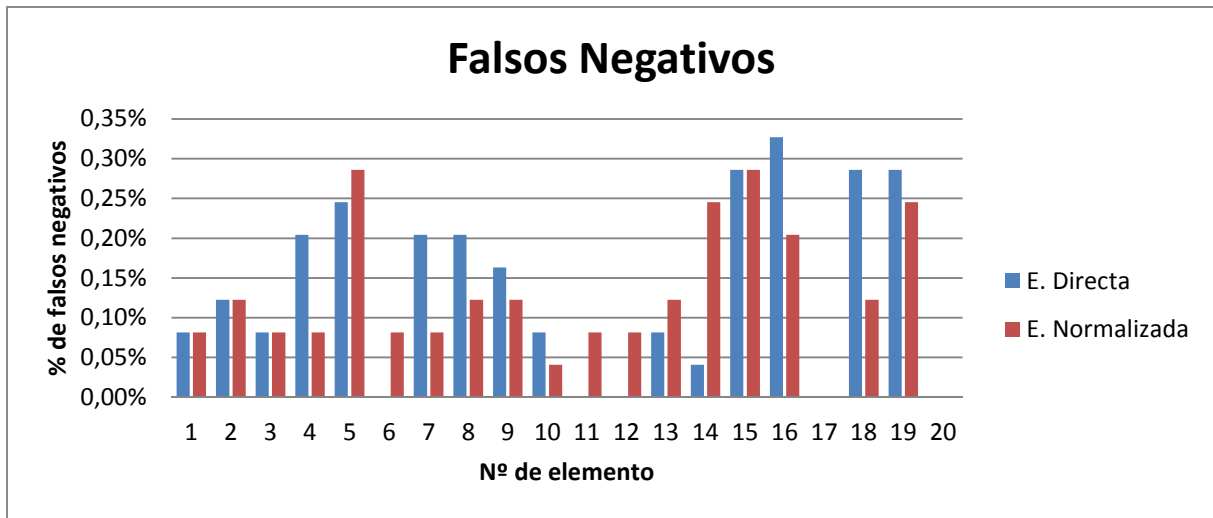


Figura 72: Número de Falsos Negativos al 2.5% (Redes sin ruido).

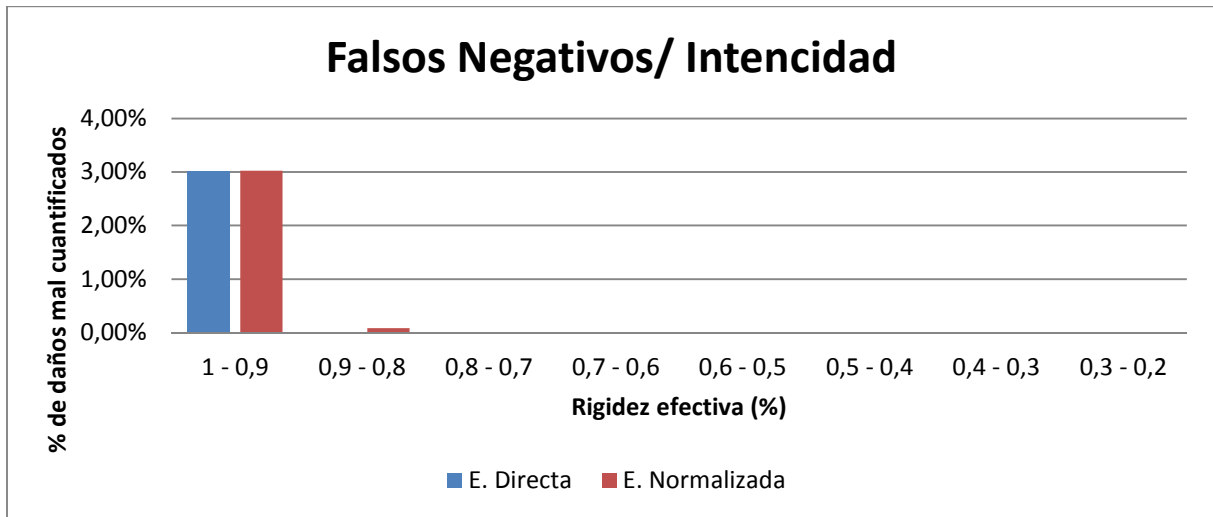


Figura 73: Número de Falsos Negativos, según rigidez efectiva al 2.5% (Redes sin ruido).

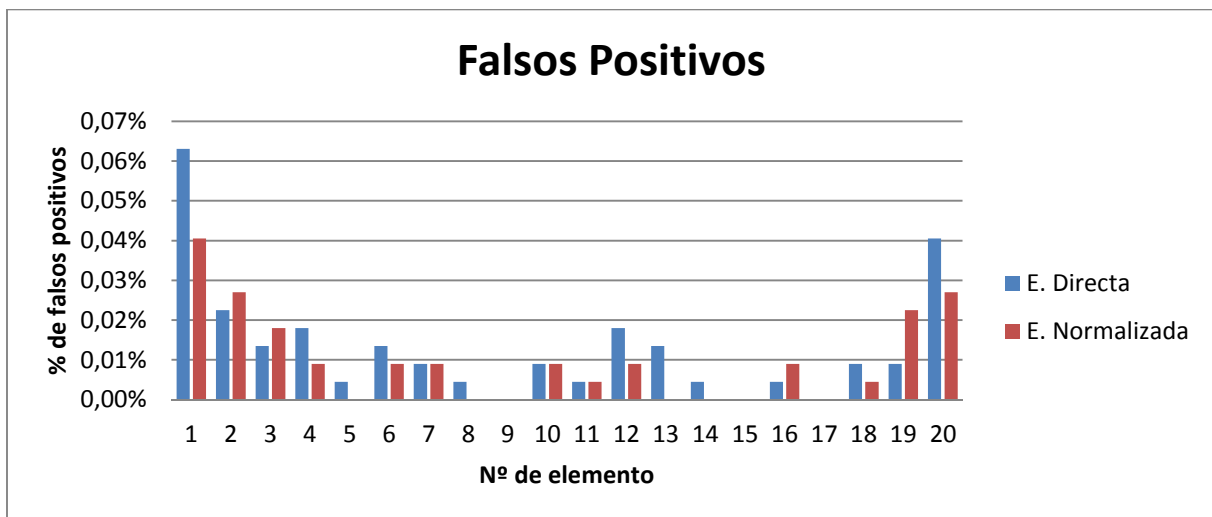


Figura 74: Número de Falsos Positivos al 2.5% (Redes sin ruido).

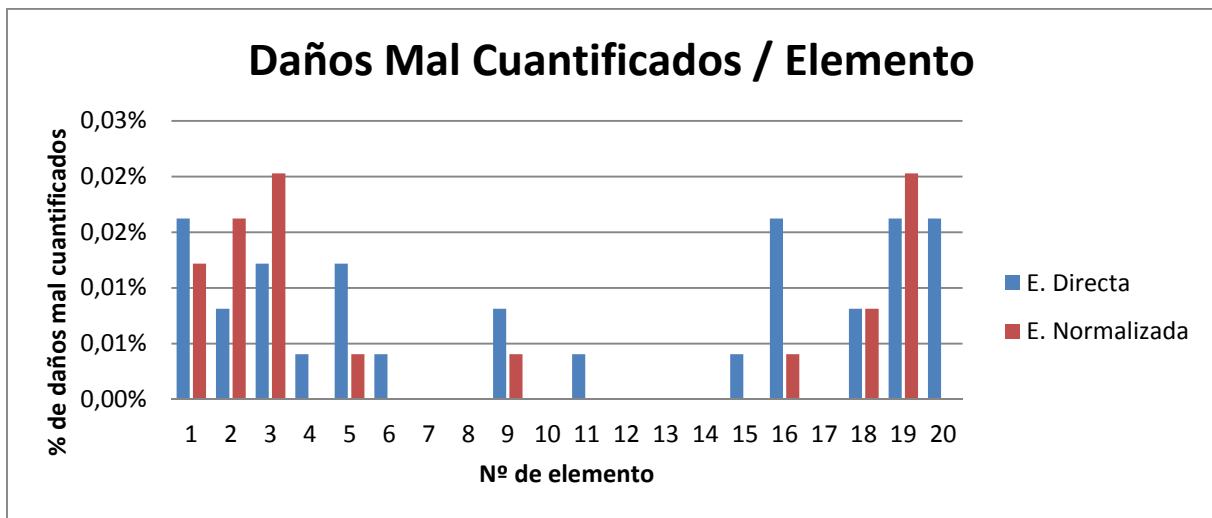


Figura 75: Daños mal cuantificado, según Nº de elemento al 5% (Redes sin ruido).

Problema 2 (Horizontal)

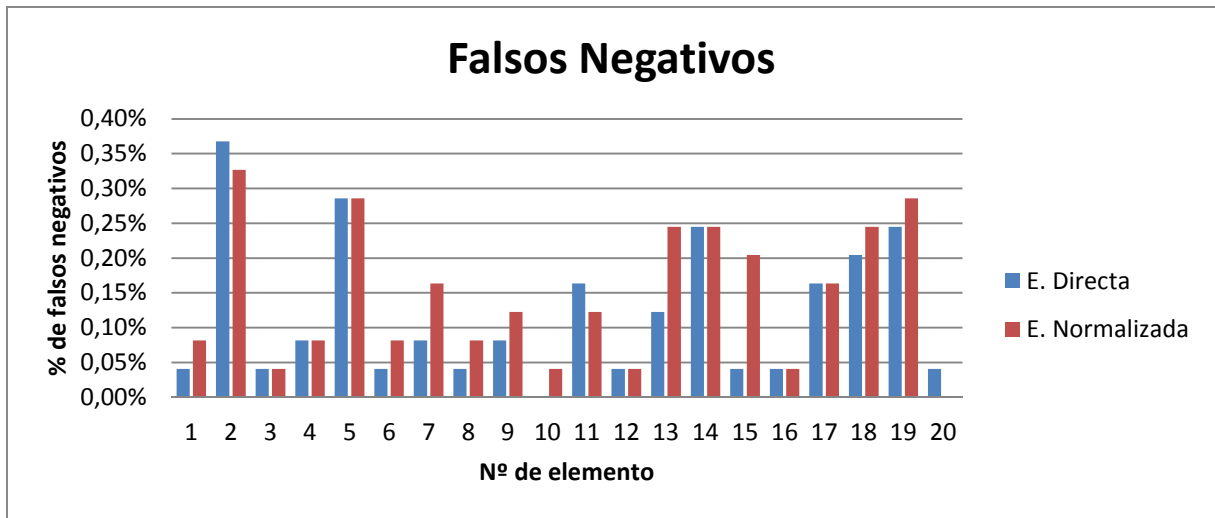


Figura 76: Número de Falsos Negativos al 2.5% (Redes sin ruido).

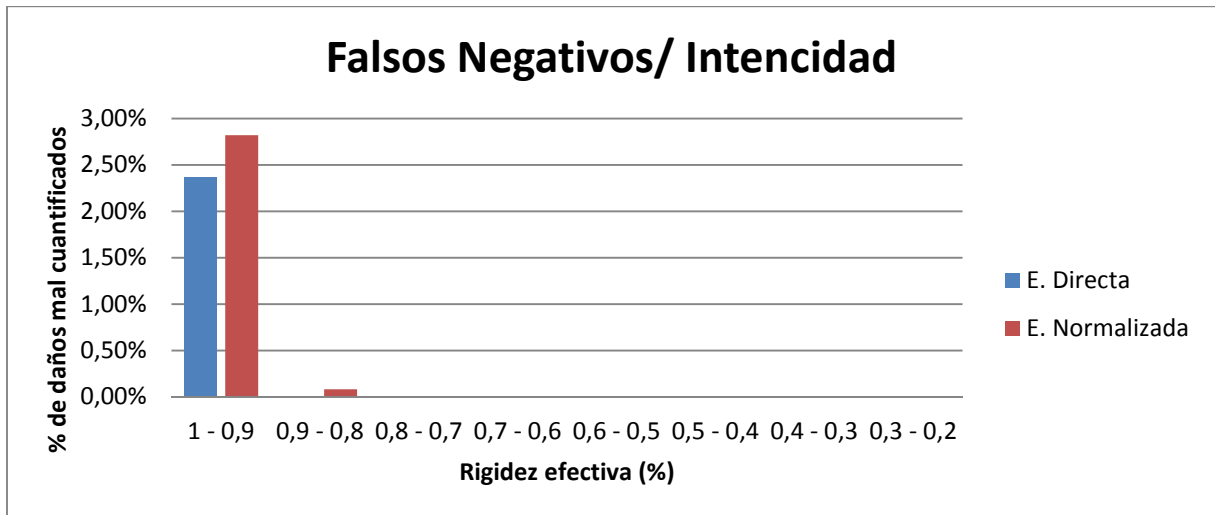


Figura 77: Número de Falsos Negativos, según rigidez efectiva al 2.5% (Redes sin ruido).

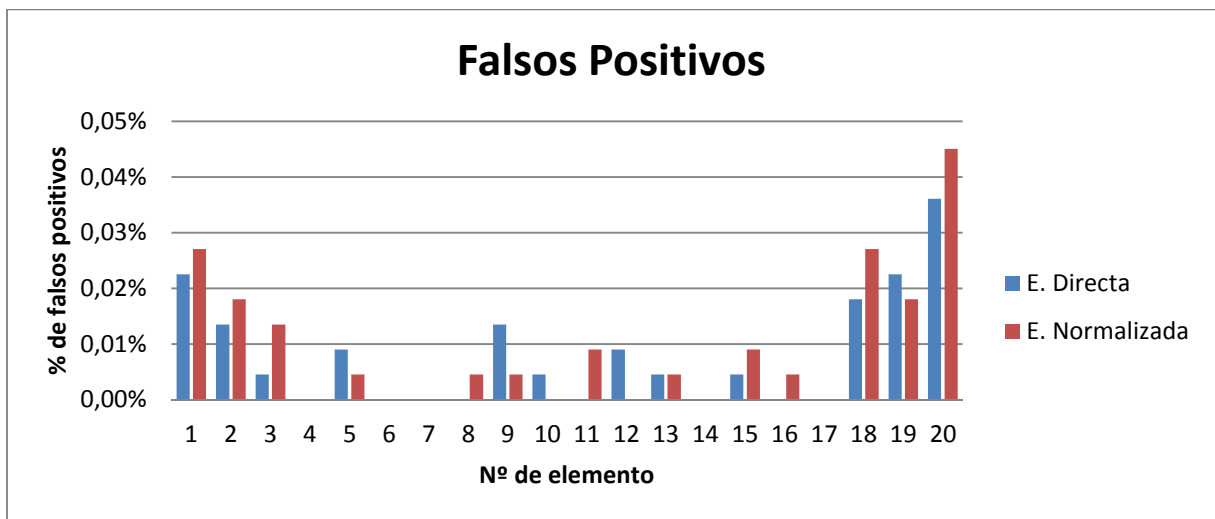


Figura 78: Número de Falsos Positivos al 2.5% (Redes sin ruido).

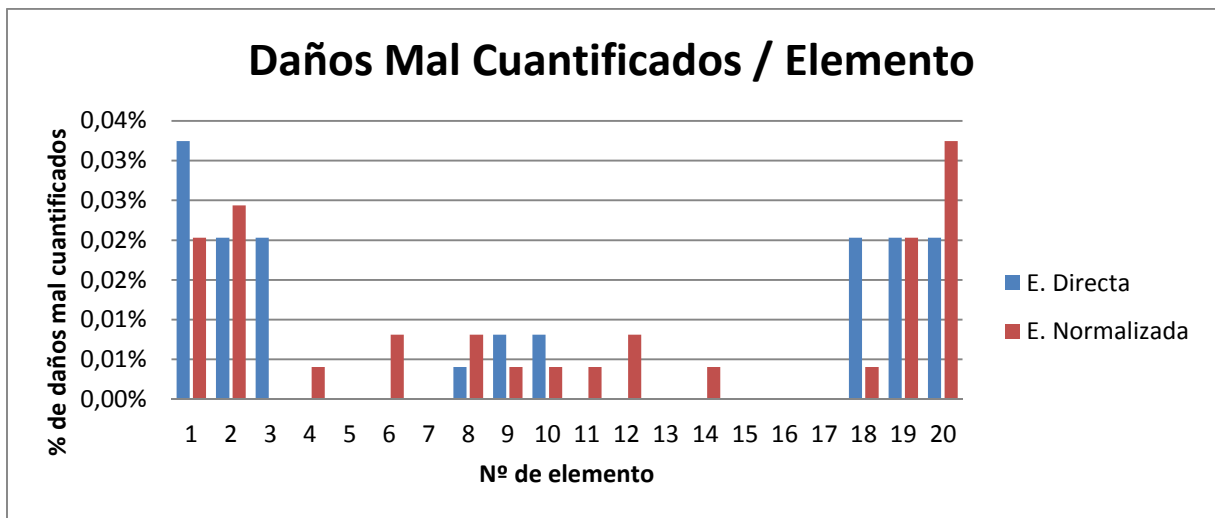


Figura 79: Daños mal cuantificado, según Nº de elemento al 5% (Redes sin ruido).

17. Anexo H: Frecuencias antiresonantes

Tabla 17: Frecuencias antiresonantes problema 1.

Frecuencias Antiresonantes		
Sin daño	Con daño*	Dif.(%)
12,31	11,66	5,28%
36,94	32,44	12,18%
59,61	59,80	-0,32%
79,96	71,74	10,28%
96,12	93,02	3,22%
109,44	106,86	2,36%
115,72	112,67	2,64%

*55% de pérdida de rigidez en el quinto resorte

Tabla 18: Frecuencias antiresonantes problema 2 (vertical).

Frecuencias Antiresonantes										
Viga 1	Viga 2	Dif.(%)	Viga 3	Dif.(%)	Viga 4	Dif.(%)	Viga 5	Dif.(%)	Viga 6	Dif.(%)
112,8	117,5	-4,17%	109,4	3,01%	116,3	-3,10%	111,3	1,33%	107,2	4,96%
242,2	246,9	-1,94%	240	0,91%	245,6	-1,40%	242,2	0,00%	230,9	4,67%
415	420,3	-1,28%	410,9	0,99%	419,1	-0,99%	412,2	0,67%	399,4	3,76%
638,8	640,6	-0,28%	631,6	1,13%	639,7	-0,14%	629,7	1,42%	625,6	2,07%
904,7	909,7	-0,55%	901,6	0,34%	907,8	-0,34%	903,4	0,14%	897,8	0,76%
103,4	103,1	0,29%	103,1	0,29%	103,8	-0,39%	101,9	1,45%	100,9	2,42%
239,7	239,1	0,25%	237,8	0,79%	238,1	0,67%	237,5	0,92%	230,9	3,67%
460	458,1	0,41%	452,8	1,57%	457,5	0,54%	451,9	1,76%	445	3,26%
659,4	663,1	-0,56%	665,9	-0,99%	663,1	-0,56%	662,2	-0,42%	658,1	0,20%
832,2	832,8	-0,07%	827,8	0,53%	827,5	0,56%	828,4	0,46%	827,8	0,53%
104,4	103,4	0,96%	102,5	1,82%	104,7	-0,29%	102,8	1,53%	101,6	2,68%
237,8	236,9	0,38%	236,6	0,50%	236,9	0,38%	235,6	0,93%	234,4	1,43%
456,6	456,6	0,00%	452,2	0,96%	455,6	0,22%	450,3	1,38%	450	1,45%
671,6	669,7	0,28%	667,5	0,61%	672,2	-0,09%	668,1	0,52%	655,9	2,34%
829,1	827,5	0,19%	825,6	0,42%	826,9	0,27%	821,6	0,90%	829,7	-0,07%
118,1	117,5	0,51%	113,1	4,23%	111,3	5,76%	110,9	6,10%	113,4	3,98%
246,9	246,6	0,12%	242,2	1,90%	240,3	2,67%	240	2,79%	240,3	2,67%
422,5	422,5	0,00%	411,3	2,65%	415	1,78%	410,3	2,89%	415,3	1,70%
643,1	640,9	0,34%	638,1	0,78%	635	1,26%	635,6	1,17%	640,3	0,44%
910	907,5	0,27%	899,7	1,13%	900,6	1,03%	895,6	1,58%	903,1	0,76%

Tabla 19: Frecuencias antiresonantes problema 2 (horizontal).

Frecuencias Antiresonantes										
Viga 1	Viga 2	Dif.(%)	Viga 3	Dif.(%)	Viga 4	Dif.(%)	Viga 5	Dif.(%)	Viga 6	Dif.(%)
293,4	287,8	1,91%	275,3	6,17%	292,8	0,20%	268,1	8,62%	257,8	12,13%
606,3	605,9	0,07%	602,8	0,58%	605,3	0,16%	605,3	0,16%	510,6	15,78%
1032,8	1010,6	2,15%	1000,3	3,15%	1028,8	0,39%	983,1	4,81%	902,2	12,65%
1552,8	1525	1,79%	1506,6	2,98%	1554,1	-0,08%	1466,60	5,55%	1471,9	5,21%
2181,3	2178,1	0,15%	2191,6	-0,47%	2181,6	-0,01%	2180	0,06%	2152,2	1,33%
260,9	251,6	3,56%	249,7	4,29%	259,4	0,57%	239,7	8,13%	238,1	8,74%
583,1	584,1	-0,17%	580,3	0,48%	584,1	-0,17%	581,6	0,26%	472,8	18,92%
1112,8	1105	0,70%	1065,3	4,27%	1113,1	-0,03%	1048,1	5,81%	966,9	13,11%
1621,9	1584,4	2,31%	1590,3	1,95%	1609,1	0,79%	1555,3	4,11%	1567,2	3,37%
1995	1989,4	0,28%	1975	1,00%	1994,1	0,05%	1982,8	0,61%	1941,3	2,69%
260	252,8	2,77%	254,4	2,15%	259,4	0,23%	249,1	4,19%	213,1	18,04%
584,4	574,1	1,76%	576,3	1,39%	582,5	0,33%	566,6	3,05%	540,6	7,49%
1113,8	1106,6	0,65%	1062,5	4,61%	1110,3	0,31%	1046,3	6,06%	1008,8 ²	9,43%
1617,5	1598,1	1,20%	1598,8	1,16%	1616,6	0,06%	1578,1	2,44%	1458,8	9,81%
1994,4	1951,9	2,13%	1926,9	3,38%	1991,3	0,16%	1878,4	5,82%	1985,6	0,44%
293,4	285,9	2,56%	288,8	1,57%	289,1	1,47%	280	4,57%	243,8	16,91%
606,3	598,8	1,24%	597,2	1,50%	602,2	0,68%	588,8	2,89%	540,9	10,79%
1032,8	1029,7	0,30%	987,5	4,39%	1028,8	0,39%	966,3	6,44%	977,8	5,33%
1552,8	1520,6	2,07%	1545,6	0,46%	1548,8	0,26%	1524,4	1,83%	1535	1,15%
2181,3	2148,1	1,52%	2125,3	2,57%	2174,1	0,33%	2088,1	4,27%	2089,4	4,21%

² En la viga 5, el valor de esta antiresonancia no se pudo determinar experimentalmente por lo que se remplazo usando los valores de las frecuencias contiguas y la misma razón que tienen estas en el caso sin daño.

