



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

ENTRENAMIENTO DE UNA RED NEURONAL PARA LA DETECCIÓN DE DAÑOS EN UNA VIGA USANDO FRECUENCIAS DE ANTI-RESONANCIA CRUZADAS

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL MECÁNICO

ÁLVARO IVÁN RIVERA TORRES

PROFESORA GUÍA:

VIVIANA MERUANE NARANJO

MIEMBROS DE LA COMISIÓN

JUAN C. ZAGAL MONTEALEGRE

ROGER BUSTAMANTE PLAZA

Este trabajo ha sido parcialmente financiado por proyecto Fondecyt

SANTIAGO DE CHILE

ENERO 2013

Resumen

La necesidad por desarrollar métodos cuantitativos para la detección de daño que puedan ser utilizados en estructuras complejas ha llevado a una continua investigación de métodos basados en los cambios en las características vibracionales de una estructura. Estos métodos son capaces de detectar daños pequeños en cualquier sector de una estructura. Tienen la ventaja que la ubicación del daño no debe ser conocida a priori y tampoco es necesario tener acceso a ella. Un área con potencial y que aún está en investigación, es el uso de las anti-resonancias en la detección de daño. Las anti-resonancias se pueden determinar de manera más precisa que los modos normales (usualmente utilizados), lo que puede resultar en una evaluación más precisa del daño.

Este trabajo busca desarrollar un algoritmo de detección de daño utilizando anti-resonancias cruzadas y redes neuronales. Las redes neuronales artificiales son eficientes técnicas computacionales, usadas ampliamente para solucionar problemas complejos en muchos campos del conocimiento. El método se valida mediante datos simulados y experimentales de una viga libre y empotrada en distintas situaciones de daño. El estudio se estructura de la siguiente manera: Se generan los modelos en elementos finitos de las vigas y se crean las bases de datos analíticas de los casos de falla. Con las bases se entrenan las RNAs, luego se validan para después hacer la toma de datos experimentales. Con estos datos se validan experimentalmente las RNAs. Más tarde se procede con la introducción de ruido a los datos analíticos y finalmente se comparan los resultados con los obtenidos anteriormente.

Se tratan los casos de una sola falla y dos fallas simultaneas en algún par de los 20 elementos en los cuales se discretizó la viga. Los daños se simulan disminuyendo porcentualmente la matriz de rigidez de alguno de los elementos de la viga.

Se logran detectar todos daños inducidos en las vigas, teniendo más éxito en la viga libre debido a que el empotramiento implementado no pudo simular las condiciones de borde completamente, lo que se traduce en que los datos obtenidos no concuerdan con los datos simulados analíticamente, los cuales obedecen a un empotramiento perfecto.

Se estudia la introducción de ruido pseudo-aleatorio que sigue una distribución uniforme y una normal. Se concluye que el ruido mejora los resultados pero no logra acercarse del todo al caso experimental, por lo que se propone analizar otro método de introducción de ruido.

Tabla de contenido

1	Introducción	1
2	Antecedentes.....	3
2.1	Modelo numérico	3
2.1.1	Elementos finitos.	3
2.1.2	Ajuste de modelos	7
2.1.3	Introducción del daño	8
2.1.4	Funciones de Respuesta en Frecuencia (FRF) y frecuencias de anti-resonancia. .	9
2.1.5	Frecuencias de anti-resonancia puntuales y cruzadas.....	11
2.2	Identificación de anti-resonancias experimentales.....	13
2.2.1	Obtención de frecuencias de anti-resonancia a través de FRF	13
2.2.2	Identificación de las frecuencias de anti-resonancia a través de funciones de transmisibilidad	14
2.3	Red Neuronal Artificial, diseño y entrenamiento	15
2.4	Entrenamiento de la red para la detección de daño	18
2.4.1	Patrones de entrenamiento y validación	18
2.4.2	Ruido experimental.....	19
2.4	Funciones de Error e Indicadores de Desempeño	19
3	Metodología.....	21
4	Resultados Problema 1.....	22
4.1	Generación de modelo y base de datos en elementos finitos viga libre	22
4.2	Variables de entrada.....	23
4.3	Entrenamiento de la red neuronal para la viga libre	24
4.4	Validación analítica de la red neuronal	26
4.5	Toma de datos experimentales.....	28
4.6	Estudio de la adición de ruido a los datos analíticos	32
4.6.1	Porcentaje de ruido agregado.....	32
4.6.2	Estabilidad de entrenamientos.....	35
5	Resultados problema 2	39
5.1	Generación de modelo y base de datos en elementos finitos viga empotrada	39
5.2	Variables de entrada.....	40
5.3	Entrenamiento de la red neuronal para la viga empotrada.....	41
5.4	Validación analítica de la red neuronal	41

5.5	Toma de datos experimentales.....	42
5.6	Estudio de la adición de ruido a los datos analíticos.....	45
5.6.1	Porcentaje de ruido agregado.....	45
5.6.2	Estabilidad de entrenamientos.....	48
6	Resultados finales.....	52
7	Discusiones.....	54
8	Conclusiones.....	57
9	Trabajos futuros.....	58
10	Bibliografía.....	59
11	Anexos.....	61
11.1	Anexo A.....	61
11.2	Anexo B.....	64
11.3	Anexo C.....	65
11.4	Anexo D.....	66
11.5	Anexo E.....	69
11.6	Anexo F.....	72

1 Introducción

La motivación de este trabajo se basa en la necesidad de detectar y evaluar oportunamente daños estructurales. Esto conlleva muchos beneficios, tales como dar mantenciones preventivas de alto rendimiento a las estructuras y maquinaria moderna, lo que ayuda a evitar daños catastróficos que pueden desencadenar en la pérdida de vidas humanas y de capital. Actualmente, los métodos de detección de fallas que predominan son los locales, es decir, solo permiten obtener información de zonas específicas. Esto presenta dos potenciales inconvenientes: Se debe tener algún indicio de la ubicación del daño en cuestión, y la existencia de zonas en las cuales el acceso se ve imposibilitado o donde los instrumentos no pueden actuar de la mejor forma. Los métodos de evaluación de daño tradicionales [3], usan la información modal obtenida de las frecuencias naturales y los modos normales de vibración. Las frecuencias naturales cuantifican y detectan la presencia de daño y los modos normales lo localizan, sin embargo, los modos normales de vibración son difíciles de identificar con exactitud. Todo esto da pie a que se investiguen nuevos métodos que permitan realizar mediciones remotas con gran precisión, sin tener que revisar exhaustivamente toda una estructura para encontrar una falla, o aún peor, nunca saber la existencia de ella y sufrir las consecuencias de su desencadenamiento. Una alternativa es el uso de la función de respuesta en frecuencia (FRF). Este método se ha convertido en una buena alternativa para el monitoreo de estructuras, en cuanto de ellas se puede extraer un muy buen detector de daño: Las frecuencias de anti-resonancia, no siendo necesario extraer los modos normales de vibración.

Las frecuencias anti-resonancias son los ceros de las FRF y son una alternativa interesante para la detección de daños, pues se pueden detectar más fácilmente y con más precisión que los modos normales de vibración y aun así, son capaces de proveer la misma cantidad de información. Además, son muy sensibles a los pequeños cambios estructurales, lo que los convierte en muy buenos indicadores de daño. What et al.[4] estudiaron las posibilidades que tenían las anti-resonancias como indicadores de modificación estructural y concluyeron que éstas deben conducir las futuras aplicaciones de detección de daños estructurales.

En el presente trabajo de título, se encuentra un algoritmo que permite la detección oportuna de daños estructurales. Este estudio corresponde a la continuación de la tesis de Javier Mahu [2], quien trabajó con una viga libre a la que se le midieron sus Funciones de Respuestas en Frecuencia (FRF) directas, es decir, se midió la respuesta en el mismo lugar donde se excitó. En este caso, se trabajará con una viga libre y una viga empotrada, el punto de medición tendrá un lugar fijo y se excitarán todos los nodos

de la barra para obtener las FRF cruzadas de todos los puntos de interés. De las FRF Cruzadas se obtienen las anti-resonancias cruzadas, las cuales tiene dos grandes ventajas frente a las anti-resonancias directas, una es que son más sensibles a los cambios de respuesta dinámica de las estructuras, lo que las hace muy buenas para detectar fallas sin tener pérdida de información y la segunda ventaja, es que no es necesario que el punto de medición sea exactamente el punto donde se excita, lo que permite hacer mediciones remotas sin problemas. Las anti-resonancias cruzadas son las entradas con las que se entrenan las RNA. Una RNA es un método de aprendizaje que consiste en emular las capacidades cerebrales humanas para procesar datos masivamente, mediante un computador. Las RNA se utilizan para desarrollar el algoritmo que logra reconocer, localizar y magnificar fallas en estructuras, usando una base de datos analítica creada mediante un modelo en elementos finitos, el que fue previamente ajustado a los datos experimentales.

La detección de daños y su evaluación pueden ser realizadas al momento de terminar el proceso de localización [6]. Si los datos medidos no tienen error, un ciclo simple de detección puede ser suficiente para evaluar los daños. Cualquier parámetro estimado que difiera de los valores base, va a estar asociado con una falla. La reducción del valor del parámetro medido sería un indicador de la severidad del daño, pero como los datos experimentales medidos siempre tendrán error, es necesario ingresar ruido a los datos analíticos simulados.

El objetivo de este trabajo es desarrollar un algoritmo que sea capaz de detectar, localizar y cuantificar daños de una estructura en tiempo real, mediante el entrenamiento de Redes Neuronales Artificiales, haciendo uso de frecuencias de anti-resonancia cruzadas. Para ello se propone utilizar redes neuronales artificiales, cuya información de entrada sean las frecuencias de anti-resonancia cruzadas, obteniendo como resultado los factores de reducción de rigidez para cada elemento de la estructura.

Para lograr el objetivo anterior se cumplen los siguientes objetivos específicos:

- Simulación de la viga libre y empotrada mediante un software de elementos finitos, para determinar las FRF y, con ellas, las anti-resonancias cruzadas.
- Validar analítica y experimentalmente los resultados.
- Determinar el efecto de las condiciones de borde en la detección de daños.
- Estudiar la introducción de ruido en los datos analíticos.
- Comparar los resultados obtenidos con las frecuencias de anti-resonancia directa.

2 Antecedentes

2.1 Modelo numérico

2.1.1 Elementos finitos.

El método de elementos finitos [1], es un método numérico general destinado para la aproximación de soluciones de ecuaciones diferenciales parciales, el cual está siendo muy utilizado en diversos problemas de ingeniería y física actualmente. Ha adquirido una gran importancia, ya que permite resolver casos que hasta hace poco tiempo eran prácticamente imposibles de resolver métodos matemáticos tradicionales.

La idea general del método de Elementos Finitos, es la división de un continuo en un conjunto de elementos pequeños interconectados por una serie de puntos llamados nodos. El dominio se divide mediante puntos (en el caso lineal), mediante líneas (en el caso bidimensional) o superficies (en el tridimensional) imaginarias, de forma que el dominio total en estudio se aproxime mediante el conjunto de porciones (elementos) en que se subdivide. Esta forma de discretización se puede ver más clara en la Figura 2-1.

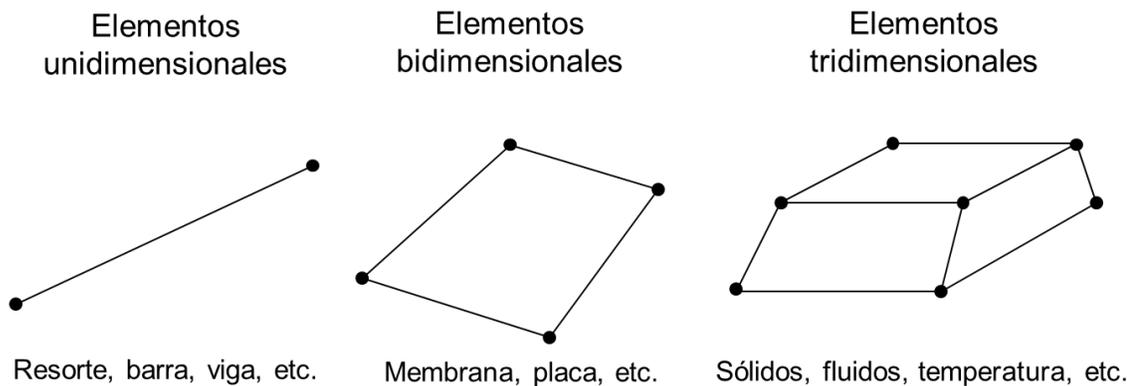


Figura 2-1: Elementos finitos clasificados según dimensión.

Las ecuaciones que rigen el comportamiento del continuo regirán también el del elemento, así se consigue pasar de un sistema continuo, que posee infinitos grados de libertad y es representado por una ecuación diferencial o un sistema de ecuaciones diferenciales, a un sistema con un número de grados de libertad finito cuyo comportamiento se modela por un sistema de ecuaciones, lineales o no. La estructura

se representa por 3 matrices: Rigidez K , Masa M y Amortiguamiento C , y se determinan como una suma de las matrices de cada elemento: $K = \sum K_i$; $M = \sum M_i$; $C = \sum C_i$.

Para estimar la respuesta dinámica de la viga, ésta se discretiza en elementos unidimensionales como el de la Figura 2-2

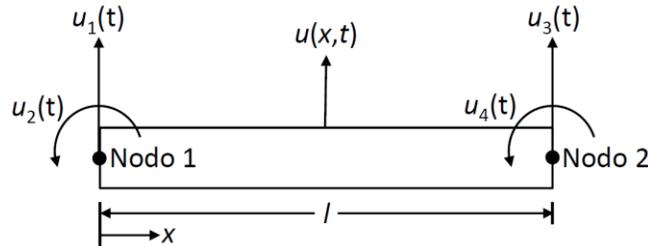


Figura 2-2: Elemento de viga simple.

Este elemento contiene dos nodos y cuatro grados de libertad. Los grados de libertad corresponden a dos desplazamientos verticales u_1 y u_3 y dos rotaciones u_2 y u_4 . Es decir, cada nodo se modela con dos grados de libertad.

La ecuación de movimiento estacionaria debe satisfacer (2.1):

$$EI \left(\frac{\partial^4 u(x, t)}{\partial x^4} \right) = 0 \quad (2.1)$$

Al integrar la ecuación (2.1) se obtiene (2.2):

$$u(x, t) = c_1(t)x^3 + c_2(t)x^2 + c_3(t)x + c_4(t) \quad (2.2)$$

La ecuación (2.2) se utiliza para aproximar el desplazamiento transversal dentro del elemento, donde $c_i(t)$ son constantes de integración con respecto a la variable espacial x . Se deben satisfacer las siguientes condiciones de borde:

$$\begin{aligned}
u(0, t) = u_1(t) \quad \frac{\partial u}{\partial x}(0, t) = u_2(t) \\
u(l, t) = u_3(t) \quad \frac{\partial u}{\partial x}(l, t) = u_4(t)
\end{aligned} \tag{2.3}$$

A partir de este conjunto de ecuaciones (2.3), se pueden despejar las constantes de integración, obteniendo:

$$c_1(t) = \frac{1}{l^3} [2(u_1 - u_3) + l(u_2 - u_4)] \tag{2.4}$$

$$c_2(t) = \frac{1}{l^2} [3(u_3 - u_1) - l(2u_2 + u_4)] \tag{2.5}$$

$$c_3(t) = u_2(t) \tag{2.6}$$

$$c_4(t) = u_1(t) \tag{2.7}$$

Reemplazando estos coeficientes en la ecuación (2.2) obtenemos:

$$\begin{aligned}
u(x, t) = \left[1 - 3\frac{x^2}{l^2} + 2\frac{x^3}{l^3} \right] u_1(t) + l \left[\frac{x}{l} - 2\frac{x^2}{l^2} + \frac{x^3}{l^3} \right] u_2(t) \\
+ \left[3\frac{x^2}{l^2} - 2\frac{x^3}{l^3} \right] u_3(t) + l \left[-\frac{x^2}{l^2} + \frac{x^3}{l^3} \right] u_4(t)
\end{aligned} \tag{2.8}$$

Con esto es posible calcular las matrices de masa M y de rigidez K . La energía cinética (2.9) y la energía potencial (2.10) del elemento de viga vienen dadas por:

$$T(t) = \frac{1}{2} \rho A \int_0^l \left[\frac{\partial u(x, t)}{\partial t} \right]^2 dx = \frac{1}{2} \dot{u}^T M \dot{u} \tag{2.9}$$

$$V(t) = \frac{1}{2} EI \int_0^l \left[\frac{\partial^2 u(x, t)}{\partial x^2} \right]^2 dx = \frac{1}{2} \dot{u}^T K \dot{u} \tag{2.10}$$

Donde:

- El vector \dot{u} es la derivada temporal del vector de coordenadas:
 $u(t) = [u_1 \ u_2 \ u_3 \ u_4]$
- E = Módulo de Young
- I = Momento de Inercia del eje en flexión
- ρ = Densidad
- A = Área de la sección transversal

Luego, integrando y factorizando, se obtienen las matrices de masa (2.11) y rigidez (2.12):

$$M = \frac{\rho A}{420} \begin{bmatrix} 156 & 22L & 54 & -13L \\ 22L & 4L^2 & 13L & -3L^2 \\ 54 & 13L & 156 & -22L \\ -13L & -3L^2 & -22L & 4L^2 \end{bmatrix} \quad (2.11)$$

$$K = \frac{EI}{L^3} \begin{bmatrix} 12 & 6L & -12 & 6L \\ 6L & 4L^2 & -6L & 2L^2 \\ -12 & -6L & 12 & -6L \\ 6L & 2L^2 & -6L & 4L^2 \end{bmatrix} \quad (2.12)$$

Las matrices K y M definen el elemento finito de una viga en vibración transversal.

Para caracterizar el comportamiento de toda la viga, se aplican las relaciones existentes entre las coordenadas nodales de cada elemento y las de coordenadas generales de la viga, habiendo hecho esto, se procede con el proceso de ensamblado de las matrices, el cual consiste en superponer las coordenadas que comparten cada elemento según el sistema de coordenadas globales del sistema, tal como se puede ver en la Figura 2-3 [9].

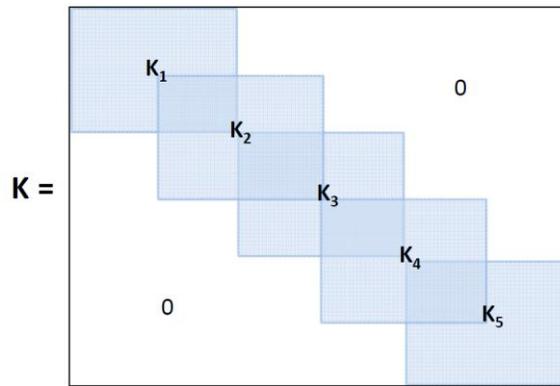


Figura 2-3: Ensamble de matriz de rigidez para una viga de 5 elementos.

2.1.2 Ajuste de modelos

Un modelo en elementos finitos permite predecir el comportamiento dinámico de la estructura bajo distintas condiciones de borde y carga, pero la confiabilidad de éste no está completamente garantizada. El propósito de los métodos de ajuste de modelo es lograr que el modelo numérico se acerque lo más posible a los datos experimentales para que entregue predicciones confiables de la dinámica de una estructura mecánica [9]. En la Figura 2-4 se ilustra el esquema general de un método de ajuste de modelos.

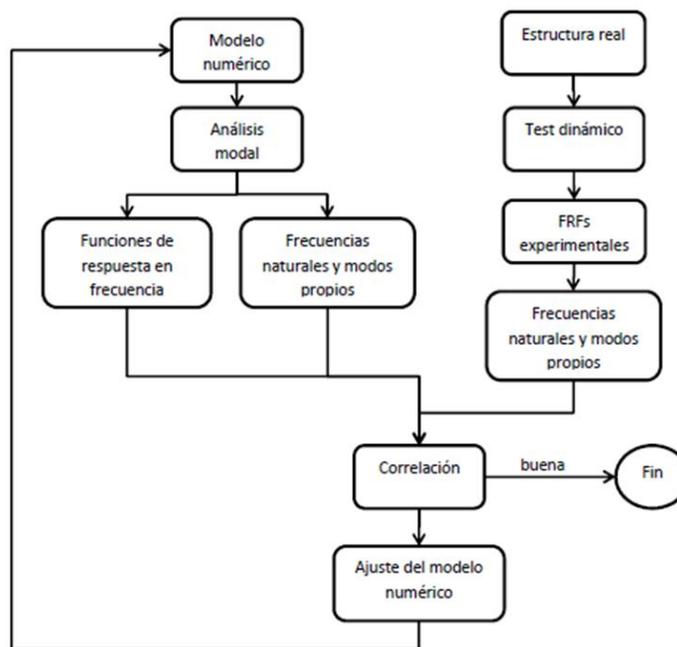


Figura 2-4: Esquema general de un ajuste de modelo.

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_i (con i entre 1 y el total de variables a ajustar) como los parámetros del modelo. Estos parámetros ponderan las propiedades originales, optimizándolas para plantear un problema no lineal que minimice la diferencia entre las frecuencias características, analíticas y experimentales. La diferencia entre el modelo numérico y experimental viene dada por la siguiente función objetivo (2.13):

$$\min \sum_n \sum_i \left\| \frac{\omega_{r,i,n}^A(\lambda_1, \lambda_2, \dots, \lambda_i)^2}{\omega_{r,i,n}^E} \right\|^{-1} \quad \text{Sujeto a } li_j \leq \lambda_j \leq ls_j \quad (2.13)$$

Donde $\omega_{r,i,n}$ es la i -ésima frecuencia anti-resonante para el n -ésimo grado de libertad, mientras que el superíndice A y E diferencia los datos analíticos de los experimentales. Para acotar el rango de soluciones se tienen los límites inferiores (li_j) y superiores (ls_j) para cada parámetro de ajuste.

2.1.3 Introducción del daño

Para simular analíticamente las influencias del daño en la estructura, generalmente causado por una grieta, se puede analizar como una reducción local de rigidez o mediante el análisis de un modelo en 2 ó 3 dimensiones (Figura 2-5) [2].

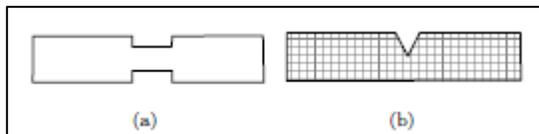


Figura 2-5: a) Reducción local de rigidez b) Modelo de grieta.

La reducción de rigidez es la manera más fácil de modelar una estructura con daños, esto se logra mediante la introducción del factor de reducción de rigidez (β). Éste representa la proporción de reducción en comparación a la rigidez original, afectando la contribución de las rigideces locales en la matriz de rigidez total (2.14).

$$[K] = \sum (1 - \beta_i) [K_i] \quad (2.14)$$

Donde $[K_i]$ es la matriz local de rigidez del i-ésimo elemento. Por otro lado, se pueden obtener mejores resultados al modelar las zona cercanas a la grieta (Figura 2-5 b). 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.

2.1.4 Funciones de Respuesta en Frecuencia (FRF) y frecuencias de anti-resonancia.

En una estructura se define como coordenadas [2], a 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 esto, los grados de libertad (gl) y las propiedades de las estructuras (tales como la masa y la rigidez), se pueden encontrar en sistemas equivalentes, como el de la Figura 2-6, cuyo movimiento queda descrito por la ecuación (2.15):

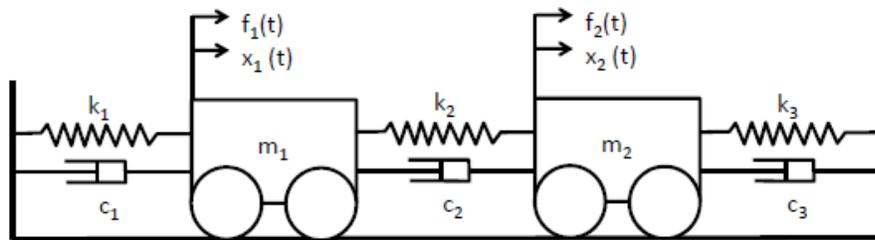


Figura 2-6: Sistema con dos grados de libertad.

$$[M]\{\ddot{x}\} + [C]\{\dot{x}\} + [K]\{x\} = \{F\} \quad (2.15)$$

- M: Matriz de masa
- K: Matriz de rigidez
- C: Matriz de amortiguamiento.
- $\{x\}$: Vector de respuesta
- $\{f\}$: Vector de fuerzas

Las soluciones de esta ecuación diferencial (2.15), homogénea y particular, representan la respuesta vibratoria transiente y estacionaria del sistema en cuestión. Para la solución homogénea de la ecuación (2.15) se supondrá una solución periódica.

$$\{\dot{x}\} = \{A\} \sin(\omega t) \Rightarrow \{\ddot{x}\} = -\omega^2 \{A\} \sin(\omega t) \quad (2.16)$$

Para la solución del problema estacionario de la ecuación (2.15) se define la Función Respuesta en Frecuencia (FRF) como el cociente entre los vectores desplazamiento y fuerza en el dominio de frecuencias, es decir, es la relación entre la respuesta del sistema y la excitación aplicado a éste. Si se calcula la transformada de Laplace de la ecuación (2.15) y se lleva al dominio de frecuencias $p = kw$, se llega a:

$$\{\text{FRF}(w)\} = \frac{\{x(w)\}}{\{F(w)\}} = ([K] + jwc - w^2[M])^{-1} = \frac{\text{adj}([K] - w^2[M])}{\det([K] - w^2[M])} \quad (2.17)$$

De la ecuación (2.17) es posible obtener la amplitud normalizada según la fuerza en función w . En la Figura 2-7 y en la Figura 2-8, es posible ver que las frecuencias resonantes son los puntos en que la amplitud de la respuesta vibratoria se maximiza, mientras que se puede definir las frecuencias anti-resonantes como los puntos en que la respuesta vibratoria tiende a cero. Analíticamente, las frecuencias resonantes hacen cero el numerador y se determinan resolviendo que el determinante de la matriz de rigidez dinámica sea cero. Las frecuencias anti-resonantes se pueden determinar resolviendo que el operador $\text{adj}()$ de la matriz de rigidez dinámica sea cero. Experimentalmente, se puede obtener la FRF de una estructura para determinar ambas frecuencias características, buscando los mínimos y máximos locales de la función, respectivamente.

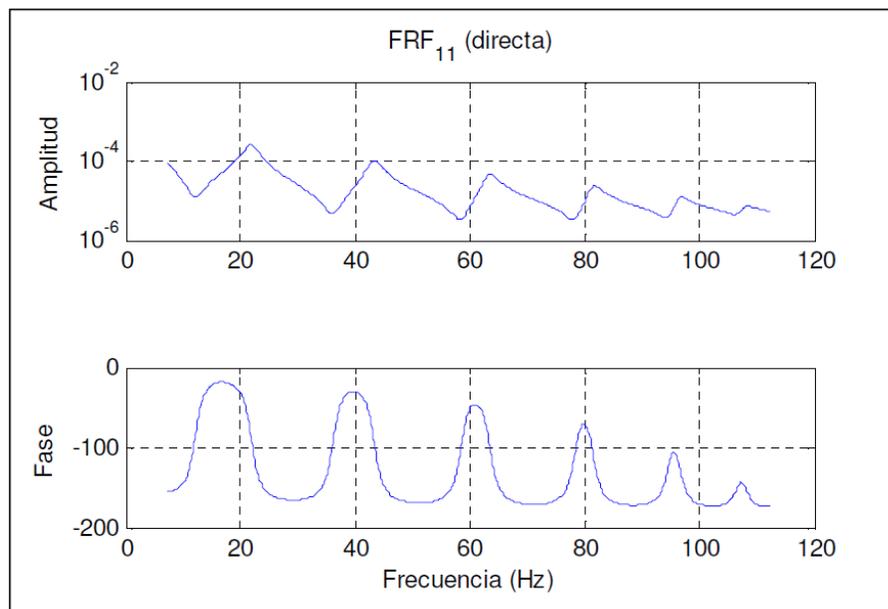


Figura 2-7: Gráfico esquemático de una FRF directa.

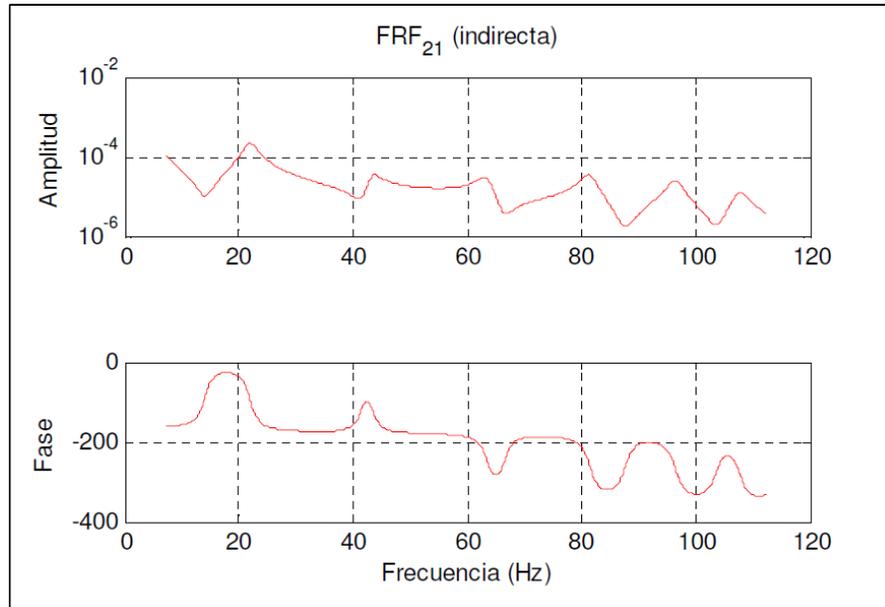


Figura 2-8: Gráfico esquemático de una FRF indirecta

2.1.5 Frecuencias de anti-resonancia puntuales y cruzadas.

Los métodos de evaluación de daño tradicionales [3], usan la información modal obtenida de las frecuencias naturales y los modos normales de vibración. Las frecuencias naturales cuantifican y detectan la presencia de daño y los modos normales lo localizan, sin embargo, los modos normales de vibración, son difíciles de identificar con exactitud. Una alternativa a este método es el uso de la función de respuesta en frecuencia (FRF). Este método se ha convertido en una buena alternativa para el monitoreo de estructuras, en cuanto de ellas se puede extraer un muy buen detector de daño: Las frecuencias de anti-resonancias, no siendo necesario extraer los modos normales de vibración, lo que evita contaminación en la información.

Las frecuencias anti-resonancias son los ceros de las FRF y son una interesante alternativa para la detección de daños, pues se pueden detectar más fácilmente y con más precisión que los modos normales de vibración y aun así, son capaces de proveer la misma cantidad de información. Además, son muy sensibles a los pequeños cambios estructurales, lo que los convierte en muy buenos indicadores de daño. Whal et al.[4] estudiaron las posibilidades que tenían las anti-resonancias como indicadores de modificación estructural y concluyeron que éstas deben conducir las futuras aplicaciones de detección de daños estructurales.

Las anti-resonancias pueden ser derivadas de las FRFs continuas y las FRFs cruzadas. En las respuestas en frecuencia continuas (Figura 2-9), el sensor de respuesta está en el mismo lugar de la excitación, mientras que en las respuestas en frecuencia cruzadas (Figura 2-10), la posición del sensor de respuesta difiere de la posición de excitación. Como una regla, las respuestas en frecuencia puntuales deben tener una anti-resonancia entre dos resonancias vecinas, sin embargo, las respuestas de frecuencia cruzadas pueden o no tener una anti-resonancia entre dos resonancias vecinas.

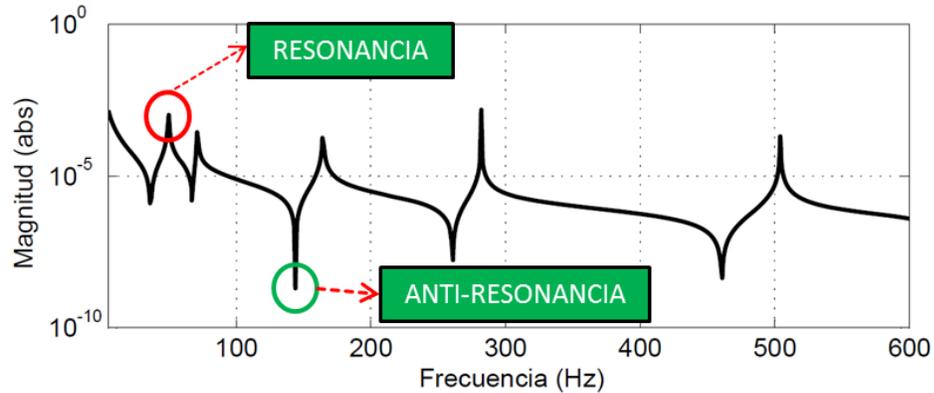


Figura 2-9: FRF continua.

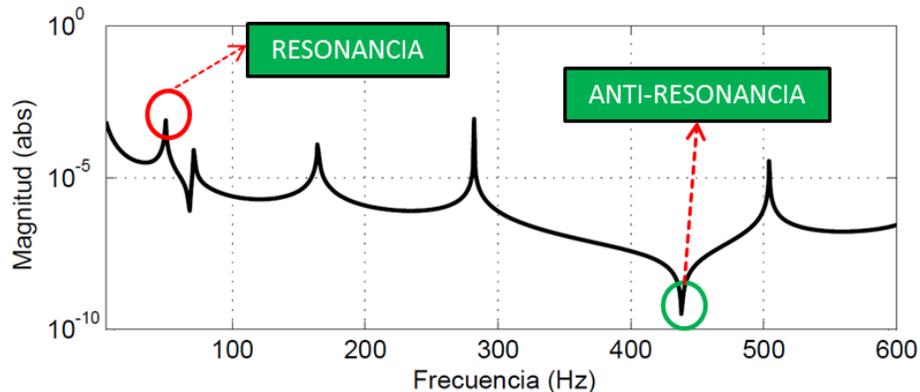


Figura 2-10: FRF cruzada.

Como se puede ver en la Figura 2-9, en el caso de las FRF puntuales, hay un sucesivo cambio de resonancias y anti-resonancias. Esta propiedad hace muy fácil el procedimiento de encontrar experimental y numéricamente las anti-resonancias continuas, sin embargo, este no es el caso de las anti-resonancias cruzadas. Algunos investigadores prefieren las FRF continuas por que aparecen problemas cuando se usan las FRF cruzadas[3]. Por otra parte, el procedimiento para obtener FRF continuas difiere del modo de testeo común, es decir, los grados de libertad de la excitación se

mueven junto con los grados de libertad de la respuesta, esto lo hace poco práctico y muy caro de aplicar en la práctica.

2.2 Identificación de anti-resonancias experimentales

Experimentalmente, la identificación de las frecuencias de anti-resonancia, continuas y cruzadas, se lleva a cabo mediante el procesamiento de los datos adquiridos por un analizador de señales. Estas señales son entregadas al analizador por acelerómetros dispuestos sobre la estructura, los cuales captan las vibraciones al momento de excitar la estructura en cuestión. Se describirán dos métodos usados para la identificación de anti-resonancias, el primero método utiliza las FRF, mientras que el segundo las transmisibilidades.

2.2.1 Obtención de frecuencias de anti-resonancia a través de FRF

Una técnica para identificar anti-resonancias experimentales es la llamada “Dip-picking” [11]. Esta técnica consiste en elegir las anti-resonancias por los mínimos que presentan las FRF al ser graficadas. Como se puede ver en las figuras Figura 2-9, Figura 2-10 y Figura 2-11, las anti-resonancias son los mínimos que presentan un cambio de fase en $+180^\circ$. Similar al “Dip picking” está el “Peak picking”, en donde se escogen las frecuencias de resonancia. El error que se pueda conllevar esta técnica estará regido por la resolución de las FRF entregadas por los acelerómetros.

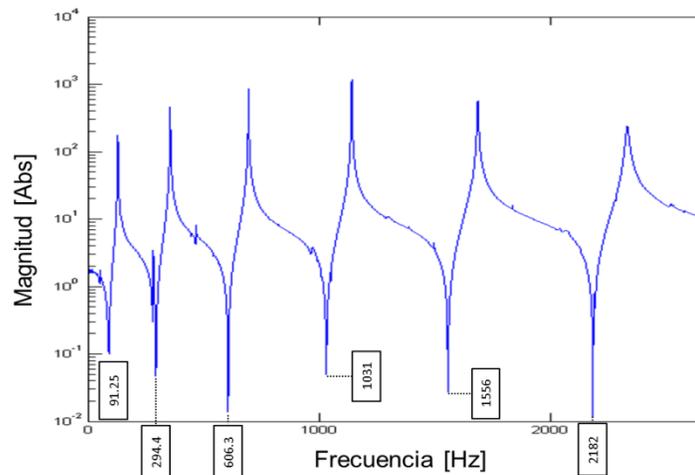


Figura 2-11 Identificación de las frecuencias de anti-resonancia experimentales a través de FRF

2.2.2 Identificación de las frecuencias de anti-resonancia a través de funciones de transmisibilidad

El algoritmo base hace uso de fracciones de polinomios racionales para identificar las frecuencias de anti-resonancia [12]. La representación de las FRF y de las funciones de transmisibilidad se hace en forma de una fracción racional. Esta representación es la razón de polinomios, donde el orden del denominador y numerador son independientes entre sí. Una función de transmisibilidad puede ser representada en forma de fracción racional como se muestra a continuación (2.18) [13]:

$$T_{ij}^k(\omega) = \frac{H_{ik}(\omega)}{H_{jk}(\omega)} = \frac{X_{ik}(\omega)}{X_{jk}(\omega)} = \frac{\sum_{p=1}^n a_p s^p}{\sum_{p=1}^m c_p s^p} \quad (2.18)$$

Cuando el polinomio del numerador se hace cero, es posible encontrar las frecuencias de anti-resonancia de la FRF de la fila i y columna k , mientras que los ceros del polinomio del denominador corresponden a las frecuencias de anti-resonancia de la FRF de la fila j y columna k .

El algoritmo propuesto usa la suma de todas las funciones de transmisibilidad cuyos denominadores tienen respuesta en j (2.19), para así entregar las anti-resonancias con una mayor exactitud.

$$T_j^k(\omega) = \sum_{p=1, p \neq j}^N T_{pj}^k = \sum_{p=1, p \neq j}^N \frac{X_{pk}(\omega)X_{jk}^*(\omega)}{X_{jk}(\omega)X_{jk}^*(\omega)} \quad (2.19)$$

N representa la cantidad de muestras medidas. Entonces, el uso de las funciones de transmisibilidad ayuda a reducir el ruido en la señal de relación, lo que tiene como consecuencia el incremento de la precisión de las anti-resonancias detectadas.

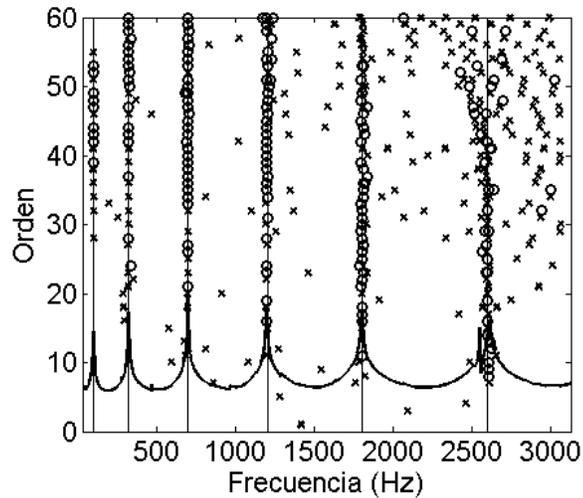


Figura 2-12 Identificación frecuencias de anti-resonancia usando la suma de funciones de transmisibilidad

La función resultante, $T_j^k(\omega)$, contiene solo los “peaks” de las frecuencias de anti-resonancia en la fila j y columna k , de la FRF. Por lo tanto, ajustando la curva $T_j^k(\omega)$, a una forma de fracción racional y resolviendo las raíces del denominador, se pueden identificar las anti-resonancias de la fila j y columna k de la FRF [13]. La Figura 2-12 muestra como el algoritmo en base a funciones de transmisibilidad es capaz de identificar las frecuencias de anti-resonancia para un grado de libertad en particular. El orden corresponde al grado de polinomio a ajustar.

2.3 Red Neuronal Artificial, diseño y entrenamiento

Las Redes Neuronales Artificiales (RNA) intentan imitar la capacidad cerebral para procesar datos de forma rápida, eficaz, en grandes cantidades y a la vez aprender de ellos. Esto lo hacen emulando los elementos constitutivos de una red neuronal humana: Las neuronas y la sinapsis. La neurona es la unidad básica de una red neuronal, y en conjunto son células altamente interconectadas, que al ser excitadas por uno o más impulsos sobre cierto nivel transmiten un impulso nervioso. La sinapsis es la unión entre una neurona y otra, y determina el tipo de comunicación que hay entre ellas, tanto en velocidad, niveles de saturación y cantidad de información comunicada.

Las Redes Neuronales pueden ser usadas para implementar funciones complejas. Si el tamaño de la red (número de interconexiones), la estructura y los pesos de la red son adecuados, ésta debería ser capaz de resolver problemas en distintas disciplinas con el nivel de precisión que se desee [5]. 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.

En una RNA se tiene que la neurona es la unidad básica de procesamiento. En la Figura 2-13 se puede ver representada por un nodo en una estructura de grafo. Cada nodo j cuenta con R entradas p_i , cada una de ellas ponderadas por un peso w_{ij} antes de entrar al nodo. Las entradas a la red corresponden a la variación de las frecuencias de anti-resonancia con respecto al caso sin daño [14], y para reducir los efectos del ruido experimental, se añade ruido a los datos del modelo numérico, como lo propusieron Hjelmstad y Shin [15]. En el nodo se suman todas las entradas ponderadas junto a un factor de corrección denominado bias (b) y se compara con el valor de activación (2.21). Al activarse la neurona se tiene una señal n (positiva). Esta señal pasa por una función de transferencia ($f(n)$) antes de convertirse en la salida de la neurona O_j . Las salidas corresponden a indicadores de daño para cada elemento del modelo numérico. Los indicadores de daño están representados como una reducción local de rigidez, estos factores representan la razón entre la variación de la rigidez y la rigidez inicial de la viga sin daños. De tenerse más de una capa de neuronas se considera un súper índice I a cada variable, que representa la capa al que corresponde dicho valor.

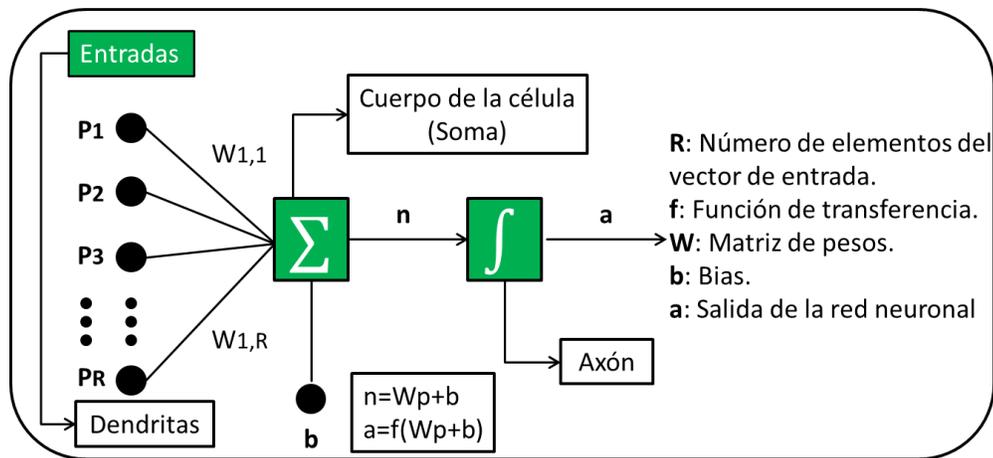


Figura 2-13: Esquema básico de una neurona.

$$\sum_{i=1}^R (w_i p_i + b_j) - \alpha_j = n \quad (2.20)$$

$$O_j = f(n) \quad (2.21)$$

El objetivo de esta función (2.21) es acotar los valores de salida de la neurona, debido a que el comportamiento de la RNA está muy influenciado por el resultado de la activación (2.20), por lo que estas funciones dan mayor control sobre la RNA. El tipo de función ocupada depende del tipo de red a entrenar. En la Figura 2-14 se presentan algunos tipos de funciones de transferencia.

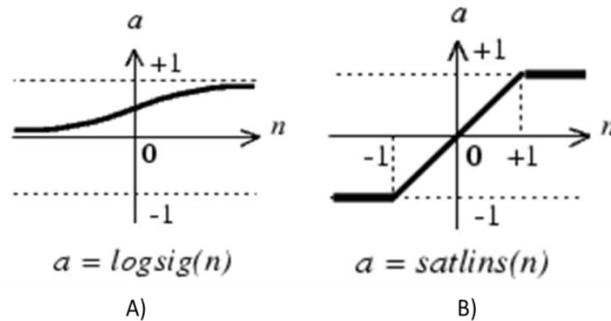


Figura 2-14: Funciones de transferencia A) Sigmoidea Logarítmica. B) Identidad Saturada Simétrica. Ref. Matlab.

La función Sigmoidea Logarítmica (Figura 2-14 A) entrega valores entre 0 y 1, es 0 cuando tiende a $-\infty$ y 1 cuando tiende a $+\infty$. La función Identidad Saturada Simétrica (Figura 2-14 B) entrega valor -1 cuando el valor ingresado es menor a -1 y entrega $+1$ cuando el valor ingresado es mayor a $+1$.

Son de particular interés las RNA utilizadas para reconocer patrones, como puede ser la configuración de rigideces de un sistema. Éstas se entrenan a partir de un conjunto de datos representativo del patrón a identificar, mediante retro-propagación. Este método se puede ilustrar de la siguiente manera, se considera el error total de una RNA (2.22) de J salidas sin capas ocultas, con G patrones de entrenamiento.

$$E = \sum_{g=1}^G E_g = \sum_{g=1}^G \left(\frac{1}{2} \sum_{i=1}^N [T_j(g) - O_j(g)]^2 \right) \quad (2.22)$$

El objetivo del método será ajustar las ponderaciones de cada conexión neuronal para minimizar el error. Para esto, se define (2.23) y se calcula la dependencia del error con el w_{ij} .

$$\Delta w_{ij} \approx -\frac{\partial E_g}{\partial w_{ij}} \quad (2.23)$$

$$-\frac{\partial E_g}{\partial w_{ij}} = \frac{\partial E_g}{\partial O_j(g)} \frac{\partial O_j}{\partial w_{ij}} = \sum_{i=1}^N [T_j(g) - O_j(g)] \frac{\partial O_j(g)}{\partial w_{ij}} \quad (2.24)$$

De este modo se genera una relación (2.24) entre la ponderación y el ajuste de sí misma, que depende de la función de transferencia de la RNA. Con esto, 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 derivar. Se puede notar, que 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.

2.4 Entrenamiento de la red para la detección de daño

Básicamente, el uso de redes neuronales para la detección de daños, consiste en construir un modelo computacional de la estructura a analizar, en este caso una viga libre y otra una en voladizo, para generar una base de datos que permita estudiar una relación entre parámetros de modos de vibrar y los parámetros estructurales, a través de un proceso de aprendizaje. Una vez que la relación se establece, entonces el modelo de RNA es capaz de detectar los daños causados por datos de los modos de vibrar, los cuales reflejan las propiedades dinámicas de la estructura.

2.4.1 Patrones de entrenamiento y validación

Los patrones de entrenamiento se generaron considerando hasta 2 daños simultáneos con 9 niveles de daño distribuidos uniformemente entre 0 y 80[%] de daño. El considerar más niveles de daño puede aumentar mucho el número de patrones necesarios para entrenar la red.

Luego del entrenamiento los datos son validados con un conjunto de patrones distinto. Los patrones de validación fueron creados considerando hasta 2 daños simultáneos con 9 niveles de daño distribuidos uniformemente entre 5 y 85% de daño. Notar que los conjuntos de entrenamiento y validación no tienen patrones en común [14].

2.4.2 Ruido experimental

Lamentablemente las medidas experimentales nunca están libres de error. El ruido en las medidas va a causar que el algoritmo de localización de daños estime valores diferentes a las actuales propiedades de la estructura. Si el experimento fuese repetido, los valores estimados serían diferentes. Entonces, los parámetros a estimar deben ser considerados como variables aleatorias. Para un conjunto de pruebas con condiciones idénticas, los parámetros estimados van a comprender una muestra de la población de las variables aleatorias.

K. D. Hjelmstad [6], para introducir el daño y verificar su eficiencia, siguió el siguiente procedimiento. Se asumió que se tiene disponible un solo conjunto de datos reales. También se asumió que la medición del ruido tiene una media cero y una varianza σ^2 . Se genera una muestra con el método de Monte Carlo de un conjunto de parámetros estimados, adicionando ruido blanco a la información real, para finalmente pasar la señal por los algoritmos de localización. Después de todo esto, se analizan las estadísticas de la población de estimaciones resultantes, donde se pudo ver qué parámetro cambió debido al ruido en las medidas y cuál de ellos cambió debido al daño.

Por todo esto, en este trabajo se estudian los efectos de la introducción de ruido a los datos analíticos generados por el modelo de elementos finitos. Se evalúa la función de introducción de ruido $RAND(N)$, incluida en el programa Matlab, la cual devuelve una matriz de n-por-n que contiene los valores pseudo-aleatorios extraídos de la distribución uniforme estándar en el intervalo abierto (0,1), y la función $RANDN(N)$ que hace lo mismo, pero utilizando una distribución normal estándar.

2.4 Funciones de Error e Indicadores de Desempeño

Con el objetivo de comparar las redes entre si, es necesario utilizar funciones de error e indicadores de desempeño para saber en qué tareas una red es superior a otra. En primera instancia, para comparar las redes se ocupa el error cuadrático medio (2.25) (Mean Square Error, MSE).

$$MSE = \frac{1}{N} \sum_N (t_i - y_i)^2 \quad (2.25)$$

Donde t_i es el valor objetivo para una entrada, y_i es la salida de la red para la misma entrada y N el número de casos analizados. Este error es independiente del número de patrones que se ocupe para entrenar o testear la red y permite tener una mirada global de la calidad de la regresión o ajuste de curva que está realizando la red.

Existen otros tres indicadores de la calidad con la que detecta daños la red, los cuales se presentan a continuación:

- Falsos positivos (2.26): Cantidad de veces por elemento que la red detecta daños en elementos sanos.

$$FP = \frac{1}{NI} * \sum_1^{NI} \varepsilon_n^I; \text{ siendo } \varepsilon_n^I = 1 \text{ si } ||1 - y_n|| \geq c \quad (2.26)$$

- Falsos negativos (2.27): El indicador inverso al anterior, cantidad de veces por elemento que la red no detecta daños en elementos con pérdida de rigidez.

$$FN = \frac{1}{ND} \sum_1^{ND} \varepsilon_n^{II}; \text{ siendo } \varepsilon_n^{II} = 1 \text{ si } ||1 - y_n|| \leq c \quad (2.27)$$

- Daños mal cuantificados (2.28): 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.

$$DMC = \frac{1}{NT} \sum_1^{NT} \varepsilon_n^{III}; \text{ siendo } \varepsilon_n^{III} = 1 \text{ si } ||t_n - y_n|| \geq c \quad (2.28)$$

Para todos los indicadores y_n corresponde a una de las salidas de la red y t_n su valor objetivo. Todos estos indicadores quedan expresados en el porcentaje de errores encontrados dentro de total de detecciones correspondientes, número de elementos dañados (ND), sanos (NS), o el total analizado (NT), esto para que los resultados sean comparables entre una estructura y otra. Todos estos indicadores van a depender de la cota c que se 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.

3 Metodología

Este trabajo de título seguirá la siguiente metodología:

- Primera etapa.

En la primera etapa construye el modelo en elementos finitos de una viga libre. A partir de este modelo, se generan las bases de datos, las que se analizan para definir el rango de la magnitud de las fallas que se estudiadas. Las bases de datos definitivas son las que se usan para entrenar la red neuronal. Finalmente se realiza la correspondencia numérica y experimental, donde se trabaja con un modelo experimental en el laboratorio de sólidos de la universidad.

- Segunda etapa.

En la segunda etapa se construye el modelo en elementos finitos de la viga en voladizo y se generan las bases de datos con las que se comprueba numéricamente el modelo, respetando el rango de variación de datos escogido en la primera etapa. Después de validar los datos obtenidos, se procede a verifican experimentalmente.

- Tercera etapa.

Se estudia el efecto de la implementación de un modelo de adición de ruido en los datos analíticos utilizados para entrenar la red neuronal.

- Cuarta etapa.

Finalmente, se procede a hacer un análisis comparativo de ambos modelos (libre y en voladizo) para estudiar el efecto de las condiciones de borde en la detección de daños y se comparan los resultados con los obtenidos con el método anterior.

4 Resultados Problema 1

4.1 Generación de modelo y base de datos en elementos finitos viga libre

La generación de un modelo en elementos finitos permite predecir el comportamiento dinámico de la estructura bajo distintas condiciones de borde y carga. En la Tabla 4-1 se muestran los datos de la viga a modelar:

Tabla 4-1: Datos de la viga a estudiar.

Largo [m]	1
Área sección transversal [m ²]	$2,5 * 10^{-4}$
Inercia sección horizontal [m ⁴]	$1,30208 * 10^{-8}$
Densidad $\left[\frac{kg}{m^3}\right]$	7800
Módulo de elasticidad [Pa]	$2,1 * 10^{11}$

La barra se discretiza en 20 elementos bidimensionales de igual tamaño, obteniéndose 21 nodos y 42 grados de libertad (dos por cada nodo, desplazamiento y rotación). En la Figura 4-1 se puede ver un esquema en donde se aprecian los elementos, los nodos y los grados de libertad.

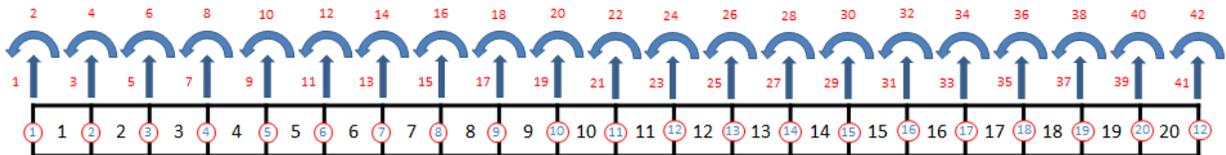


Figura 4-1: Modelo de la barra simulada con elementos finitos.

La viga se discretiza en 20 elementos porque estos dan un rango con precisión suficiente para localizar un daño y porque, de elegir más elementos, el tiempo computacional utilizado para entrenar las RNAs aumenta mucho.

Como se comentó en la sección (2.2), a cada uno de los 20 elementos le corresponde una matriz de masa M (2.11) y una matriz de rigidez K (2.12), las cuales se tienen que ensamblar con el procedimiento indicado en la misma sección. Una vez que las matrices se encuentran ensambladas, es posible calcular las anti-resonancias.

Primero se hace una base de datos con distintos conjuntos de anti-resonancias entregados por la viga para diferentes casos de fallas, con el objetivo de entrenar la RNA. Las anti-resonancias se normalizan con respecto al caso sin falla. Esto se hace para limitar las diferencias entre los datos simulados y los datos experimentales, con el objetivo de mejorar el desempeño de las redes entrenadas [2]. Las fallas en la viga se simulan disminuyendo la matriz de rigidez en alguno de los 20 elementos. Se simulan los casos de una falla y todas las combinaciones de dos fallas simultáneas sobre la viga. La disminución de la matriz de rigidez se hará multiplicándola por un factor β cuyo valor tiene 9 niveles de daño distribuidos uniformemente entre 20 y 100[%], donde 100[%] representa a un elemento sin fallas, y mientras va disminuyendo el porcentaje del factor β , va aumentando la reducción de la matriz de rigidez del elemento. Los casos de falla son los siguientes: β : [0,2; 0,3; 0,4; 0,5; 0,6; 0,7; 0,8; 0,9; 1], con los cuales se tienen 12394 posibles escenarios. La posición de los sensores sobre la viga y el lugar de aplicación de la excitación se pueden ver en la Figura 4-2:

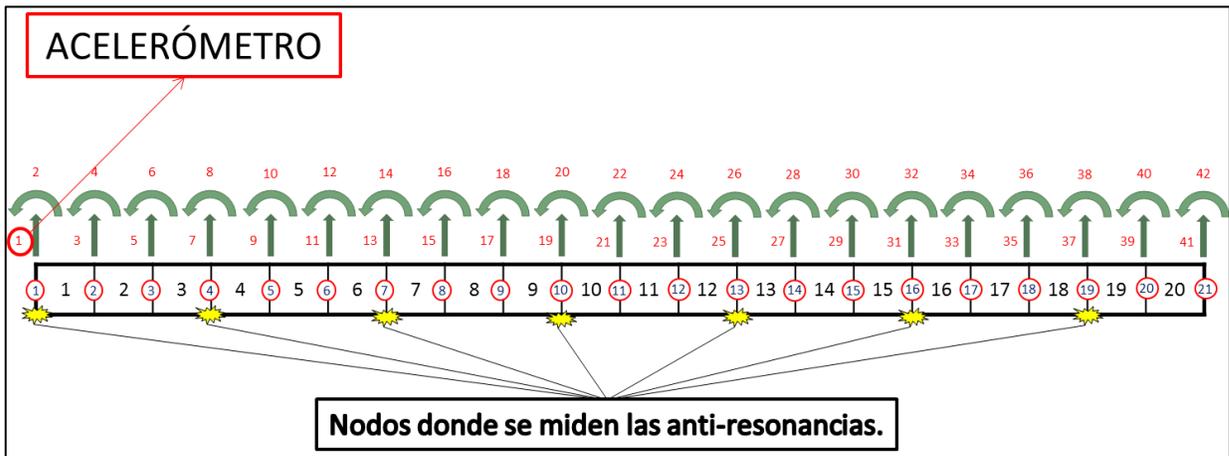


Figura 4-2: Nodos donde se medirán las anti-resonancias y lugar de posicionamiento del acelerómetro.

Se puede apreciar que las FRF medidas en el primer nodo son directas, ya que se mide en el mismo lugar donde se excita.

El código utilizado para generar el modelo y para generar la base de datos está en el anexo A y B de este informe.

4.2 Variables de entrada

El número de variables de entrada influye en varios factores, como el alcance de los recursos computacionales, el rango de resolución de los acelerómetros, la cantidad de

puntos a medir para obtener la información necesaria para detectar daños, el comportamiento de las anti-resonancias cruzadas, etc. El primer criterio que utilizado es el de la cantidad de anti-resonancias, esto se hace obedeciendo al caso anterior estudiado [2], donde se concluyó que con 20 anti-resonancias era suficiente para poder detectar daños. En [2], para detectar las 20 anti-resonancias se utilizaron 4 puntos de medición, obteniéndose 5 anti-resonancias por cada uno, en este caso, son necesarios 7 puntos de medición, ya que mientras más lejos se encuentre el punto de medición del punto de excitación, se pueden rescatar menos anti-resonancias de éste. Esto no pasaba en [2] ya que como se midieron anti-resonancias directas, el punto de medición siempre era el mismo que el de excitación. Luego, el siguiente criterio es el del rango de buena resolución de los acelerómetros, el cual está entre $(0 - 3000)[Hz]$. Se decide usar las anti-resonancias en el rango de $(1 - 2800)[Hz]$, esto para asegurar una buena resolución de los datos tomados y para obtener un número razonable de anti-resonancias por cada caso de falla. El número de anti-resonancias cruzadas utilizadas por cada acelerómetro se encuentra en la Tabla 4-2. Cabe destacar que el número de anti-resonancias encontradas en el rango de frecuencias elegido, varía dependiendo de cada caso de falla, por lo que se decide tomar un número fijo de éstas por sobre $1[Hz]$, es decir, con el primer acelerómetro se tomó las 6 primeras anti-resonancias sobre $1[Hz]$, con el segundo se tomó las 5 primeras y así sucesivamente, todo esto con el objetivo de tener una buena concordancia de datos en todos los casos simulados.

Tabla 4-2: Número de anti-resonancias cruzadas tomadas por cada sensor en la viga libre.

Acelerómetro	Anti-resonancias utilizadas [Hz]
1	6
2	5
3	4
4	3
5	2
6	2
7	1
Total	23

4.3 Entrenamiento de la red neuronal para la viga libre

Una vez escogidos el número de sensores, el rango de frecuencias y el número de anti-resonancias cruzadas a detectar por cada sensor, se procede a procesar los datos en miras a entrenar la RNA. Se calculan las anti-resonancias cruzadas para los 12394 casos simulados y luego se normalizan con respecto al caso sin fallas. Una vez arreglado todo, los datos están listos para entrenar la red.

Para entrenar la red neuronal existen distintos métodos y configuraciones, la elección óptima dependerá del problema con el que se esté trabajando. En este caso, se utilizó una red multicapa con una capa oculta, que fue la configuración óptima encontrada por Javier Mahu [2]. La configuración de red es la siguiente:

- Tipo: Perceptrón Multicapa (MLP)
- N° de entradas: 23 (anti-resonancias)
- N° de capas ocultas: 1
- N° de neuronas de salida: 18
- Método de entrenamiento: Levenberg-Marquardt

Solo falta encontrar el número óptimo de neuronas en la capa oculta, el cual está dado por el mínimo error cuadrático medio (Ec. 2.25). Se entrenan varias redes con distintos número de neuronas en la capa oculta, y los resultados del error cuadrático medio se pueden ver en la Tabla 4-3 y en la Figura 4-3. El valor mínimo se obtiene con 140 neuronas.

Tabla 4-3: Error cuadrático medio del conjunto de validación resultante para para distintas configuraciones de neuronas en la capa oculta.

Neuronas en la capa oculta	ECM
10	0,0023
20	9,92E-04
30	5,25E-04
40	3,25E-04
50	2,79E-04
60	1,21E-04
70	1,10E-04
80	1,19E-04
90	2,03E-04
100	7,81E-05
120	6,48E-05
140	5,40E-05
160	9,86E-05
200	9,19E-05

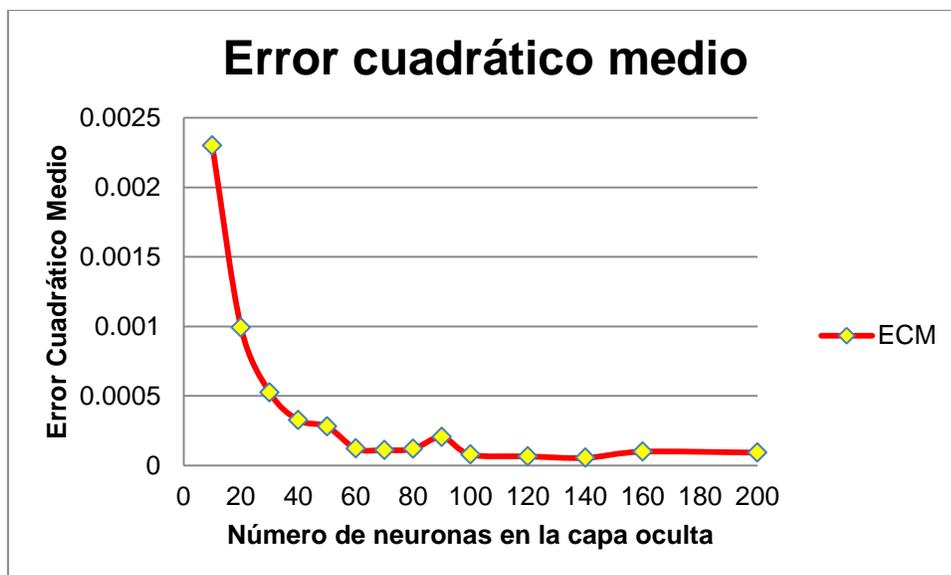


Figura 4-3: Error cuadrático medio de las redes entrenadas.

Para la capa oculta, la función de transferencia utilizada es la llamada “Logaritmo-Sigmoideo (logsig)”. Para la capa de salida, la función de transferencia utilizada es la de “Identidad Saturada Simétrica (satlins)”.

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, siendo la “Identidad Saturada Simétrica” la que mejor resultados presenta [2]. El código para entrenar la red neuronal se encuentra en el anexo C de este informe.

4.4 Validación analítica de la red neuronal

La validación analítica de la red neuronal, consiste en generar una base de datos que sea distinta a la base de datos generada para entrenar la red neuronal, es decir, que tenga información de fallas en la viga distintas a las usadas en el entrenamiento. Con esta base de datos, se procede a verificar si la RNA logra identificar, localizar y magnificar las fallas ingresadas. Tal como se mencionó en los antecedentes, los métodos que se usan para verificar la efectividad de la red son los “Falsos positivos” y los “Falsos negativos”, los cuales informan sobre los daños que la red detecta en

elementos sanos y los daños que la red no detecta en elementos que presentan pérdidas de rigidez.

En la Figura 4-4 es posible ver los Falsos Positivos, este gráfico indica que del 100% de los daños encontrados, alrededor de un 50% de ellos no existe. Esta cifra es alta, pero considerando que está concentrada en el tramo de fallas con un nivel menor al 10%, se puede decir que son despreciables.

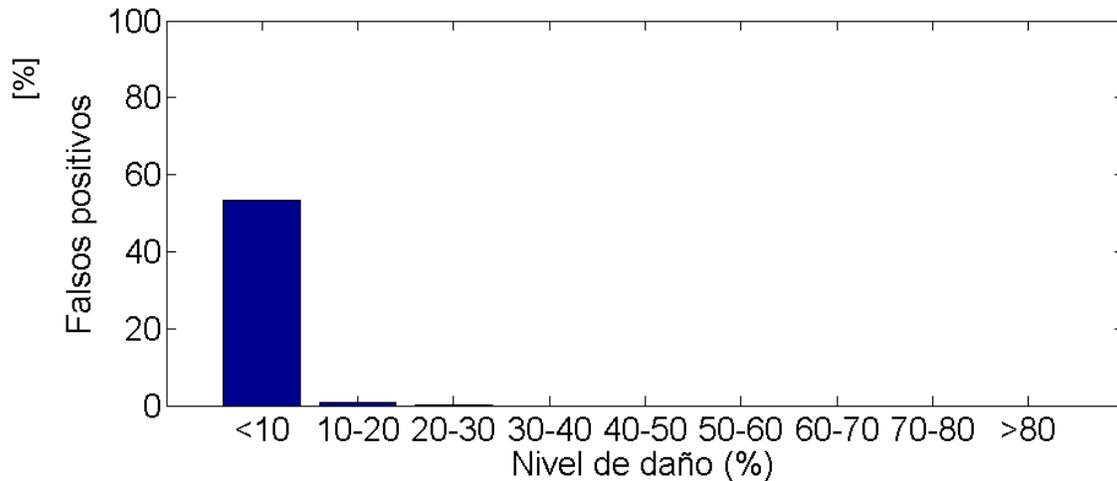


Figura 4-4: Falsos positivos encontrados al entrenar la red con 140 neuronas en la capa oculta para la viga libre.

En la Figura 4-5 es posible ver los Falsos Negativos, este gráfico indica que del 100% de los daños, alrededor de un 5% no fueron detectados. Esta cifra es baja, y considerando que está concentrada en el tramo de fallas con un nivel menor al 10%, se puede aprobar con seguridad el funcionamiento analítico de la red.

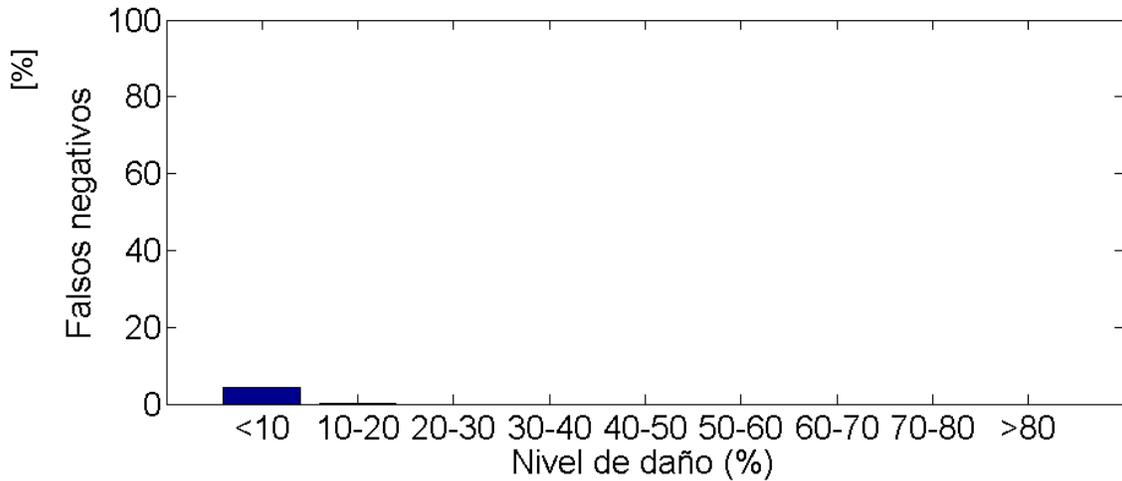


Figura 4-5: Falsos negativos encontrados al entrenar la red con 140 neuronas en la capa oculta para la viga libre.

La rutina de entrenamiento para los indicadores de desempeño se encuentra en el anexo D.

4.5 Toma de datos experimentales

La toma de datos para la viga libre se realiza haciendo uso de un martillo modal, un acelerómetro y un receptor (o acondicionador) de señales, los cuales se pueden ver en la Figura 4-6.

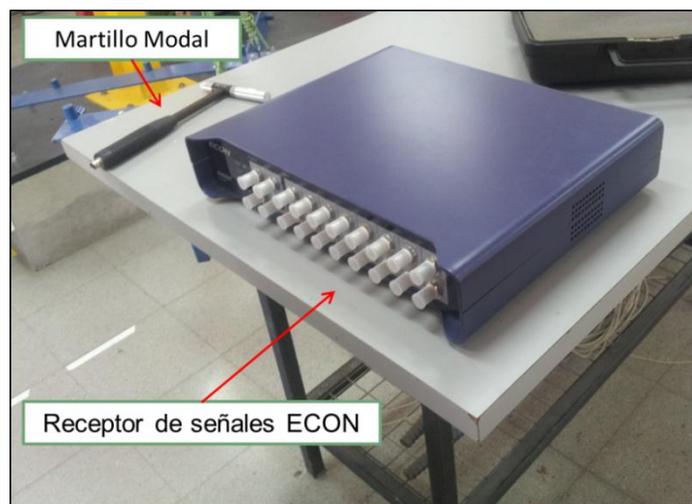


Figura 4-6: Martillo modal y Receptor de señales utilizados en la toma de datos.

El martillo es un aparato que produce una fuerza de excitación con la forma de un pulso. Consiste principalmente en un sensor de fuerzas en la punta del martillo, una masa y el mango. El acelerómetro es el sensor más utilizado en el análisis modal, éste mide la aceleración en un punto de una estructura y su señal de salida viene en forma de voltaje. Esta señal es transformada por el acondicionador de la señales, para luego ser procesada por un software.

En la Figura 4-7 es posible ver el montaje experimental de la viga libre. Éste consiste en una viga de acero colgada, por su lado más ancho, mediante dos elásticos que no influyen en la respuesta dinámica de la viga. A la viga se le adhiere un acelerómetro en uno de sus extremos usando cera. El acelerómetro está conectado al receptor de señales, y éste a un computador que contiene el software necesario para trabajar los datos recibidos.

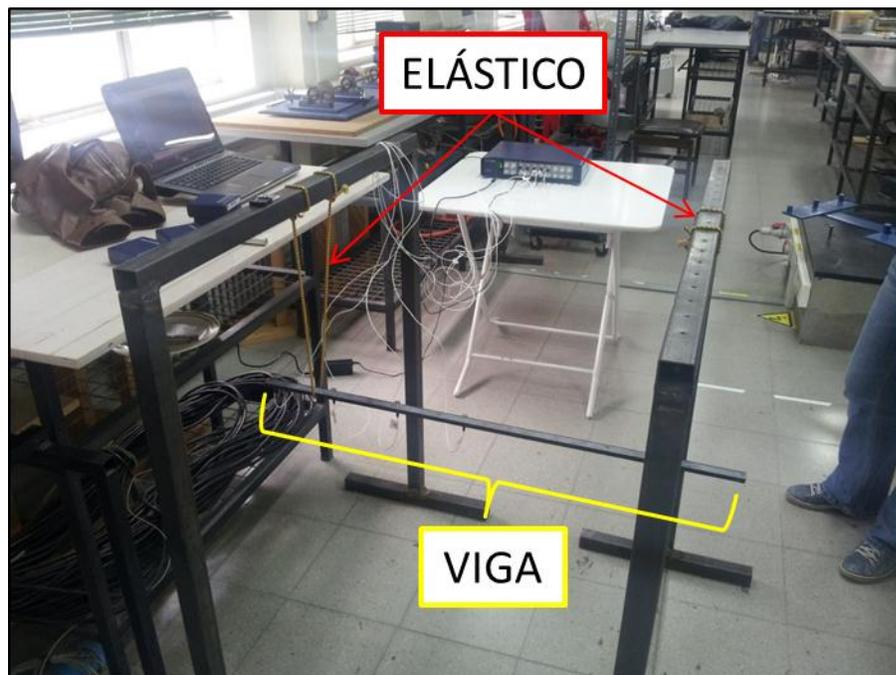


Figura 4-7: Montaje experimental viga libre.

La toma de datos se hace excitando la viga en un nodo con el martillo modal en reiteradas ocasiones, para así obtener un promedio de las anti-resonancias de ese nodo. Se procede de la misma forma con los 21 nodos de la viga. Cabe destacar que para la viga libre solo se usa la información de 7 nodos, pero se miden las FRF de todos los nodos con el fin de que, si se desea cambiar el nodo a medir, ya se tengan los datos de éste. Los detalles de la cantidad de fallas inducidas en las vigas, su magnitud y el nodo donde se ubican, se encuentran en la Tabla 4-4.

Tabla 4-4: Detalle de las vigas analizadas.

Viga	Elemento	Profundidad [mm]	Elemento	Profundidad [mm]
1	14	8	-	-
2A	8	12	14	6
2B	8	5	17	16

Como ejemplo, en Tabla 4-5 se pueden apreciar las anti-resonancias identificadas para la viga libre sin daños, y a su lado, las correspondientes anti-resonancias calculadas analíticamente. Los colores indican de qué sensor vienen las anti-resonancias.

Tabla 4-5: Anti-Resonancias experimentales y analíticas para la viga libre sin daño.

Anti-resonancias experimentales [Hz]	Anti-resonancias analíticas [Hz]
2182,6	2299,11
1558,0	1621,76
1034,1	1062,60
608,2	621,32
294,8	297,78
93,6	91,89
2112,0	2241,64
1393,9	1454,72
801,7	831,46
373,5	383,81
111,3	113,25
2043,3	2170,02
1224,1	1271,14
585,7	606,85
170,3	167,96
1969,2	2054,78
954,4	985,09
291,9	301,96
1774,5	1862,21
552,4	570,65
1392,6	1460,32
106,1	105,52
617,5	642,70

Estas anti-resonancias fueron obtenidas mediante el método “Dip Picking” explicado en la sección (2.3.1). Como es posible ver, hay una diferencia entre las anti-resonancias calculadas analíticamente y las encontradas experimentalmente. Para que los resultados analíticos se acerquen más a los experimentales se aplica un ajuste de modelo (Tabla 4-6), cuyos resultados se pueden ver en la Tabla 4-7.

Tabla 4-6: Ajuste de modelo aplicado a 4 parámetros de la viga.

	Valor original	Corrección	Valor final
Módulo de Young [Pa]	2,1*10 ¹¹	0,0109	2,122892*10 ¹¹
Densidad [kg/m3]	7800	0,04614	8159,892
Alto [m]	0,01	-0,0002047	0,009997953
Ancho [m]	0,025	-0,058104	0,0235474

Tabla 4-7: Tabla con las anti-resonancias sin ajuste y con ajuste de modelo para la viga libre.

Anti-resonancias Analíticas sin ajuste [Hz]	Anti-resonancias Analíticas con ajuste [Hz]	Anti-resonancias Experimentales [Hz]	Diferencia % de las anti-resonancias sin ajuste c/r a las experimentales	Diferencia % de las anti-resonancias con ajuste c/r a las experimentales
2299,11	2234,63	2182,59	5,34	2,38
1621,76	1576,27	1558,03	4,09	1,17
1062,60	1032,79	1034,14	2,75	0,13
621,32	603,90	608,25	2,15	0,72
297,78	289,43	294,81	1,01	1,83
91,89	89,31	93,64	1,87	4,63
2241,64	2178,76	2111,98	6,14	3,16
1454,72	1413,92	1393,91	4,36	1,44
831,46	808,14	801,68	3,71	0,81
383,81	373,05	373,52	2,75	0,13
113,25	110,07	111,32	1,73	1,12
2170,02	2109,16	2043,32	6,20	3,22
1271,14	1235,49	1224,07	3,85	0,93
606,85	589,83	585,74	3,60	0,70
167,96	163,25	170,27	1,35	4,12
2054,78	1997,15	1969,17	4,35	1,42
985,09	957,46	954,38	3,22	0,32
301,96	293,49	291,95	3,43	0,53
1862,21	1809,97	1774,53	4,94	2,00
570,65	554,65	552,36	3,31	0,41
1460,32	1419,36	1392,63	4,86	1,92
105,52	102,56	106,09	0,54	3,33
642,70	624,68	617,50	4,08	1,16

En la primera columna están las anti-resonancias calculadas sin el ajuste de modelo, en la segunda columna están las anti-resonancias con el ajuste de modelo, en la tercera columna están las anti-resonancias tomadas experimentalmente y en la cuarta y quinta columna se encuentran las diferencias porcentuales de las anti-resonancias sin el ajuste y las con ajuste de modelo respectivamente. Es posible ver que las anti-resonancias con ajuste tienen valores más cercanos a las anti-resonancias experimentales, lo que confirma la efectividad del método.

4.6 Estudio de la adición de ruido a los datos analíticos

Como se explicó en la sección (2.4.2), los datos experimentales nunca van a estar libres de ruido, debido a esto, los datos analíticos van a diferir de los experimentales en cierto porcentaje. La primera vía para disminuir esta diferencia fue aplicar un ajuste de modelo, el cual acercó los datos analíticos a los experimentales, pero aún no son iguales del todo. Otra forma de reducir la diferencia es adicionando ruido a las anti-resonancias calculadas analíticamente. Los métodos de introducción de ruido utilizados son las funciones del programa computacional MatLab RAND y RANDN (explicadas en la sección 2.4.2).

Los puntos principales estudiados son el porcentaje de ruido agregado y la estabilidad de los resultados obtenidos al entrenar las RNAs con igual configuración agregando el mismo porcentaje de ruido.

4.6.1 Porcentaje de ruido agregado

El porcentaje de ruido agregado a las anti-resonancias consiste en aumentar o disminuir en cierto porcentaje el valor de las frecuencias, todo esto con el objetivo mencionado anteriormente. Los porcentajes estudiados son: (0.5,1,1.5,2)[%].

En la Tabla 4-8 es posible ver los resultados de los entrenamientos aplicados a una viga con una falla con el método RAND. Los valores allí indicados corresponden a la disminución de la matriz de rigidez de los 20 elementos, siendo 0[%] un elemento sano, es decir, sin fallas.

Tabla 4-8: Resultado de entrenamiento con distintos % de error para la viga libre con una falla.

Elemento	Falla	Entrenamiento con 0,5 [%] de error	Entrenamiento con 1 [%] de error	Entrenamiento con 1,5 [%] de error	Entrenamiento con 2 [%] de error
1		0,0	0,0	0,0	0,0
2		0,0	0,0	5,5	0,0
3		0,0	0,0	0,0	0,0
4		19,2	21,1	15,2	12,5
5		0,0	0,0	0,0	0,0
6		0,0	0,0	0,0	0,0
7		10,9	0,0	0,0	0,0
8		0,0	0,0	0,0	0,0
9		0,0	0,0	0,0	0,0
10		0,0	0,0	0,0	0,0
11		6,2	9,5	0,0	0,0
12		0,0	0,0	0,0	0,0
13		0,0	0,0	0,0	0,0
14	X	34,3	38,5	33,9	25,7
15		0,0	0,0	0,0	0,0
16		0,0	0,0	0,0	0,0
17		0,0	0,0	0,0	0,0
18		0,0	0,0	0,0	0,0
19		0,0	0,0	0,0	0,0
20		0,0	0,0	0,0	0,0

Con celeste se indican los falsos positivos y en amarillo la ubicación de un daño menor

Como se puede apreciar en este caso, el entrenamiento que usa un 2% de error es el más estable, ya que tiene menos falsos positivos y ningún falso negativo. El falso positivo que más se destaca es el del elemento 4, el cual lo detectan todas las RNA, esto se puede deber a una falla en los datos de la viga sana o a alguna falla interna de ésta que no estaba contemplada para detectar.

Los resultados obtenidos con el método RANDN están muy alejados de encontrar una falla, por lo que no se siguió utilizando éste método. En la Tabla 4-9 hay una muestra de un entrenamiento con 2[%] de error usando el método RANDN. Es posible ver que el método no logra encontrar la falla y solo encuentra falsos positivos en otros elementos, y lo mismo pasa para el resto de las vigas.

En la Tabla 4-10 es posible ver los resultados de los entrenamientos aplicados a una viga con dos fallas. Como se trabaja con dos vigas con dos fallas, a estas se les identifica como "Viga A" y "Viga B". Al igual que en el caso anterior, la red que mejor se comporta es la que tiene un 2[%] de error, ya que ésta no presenta ni un falso positivo ni negativo, siendo que todas las otras redes tienen uno o más falsos positivos.

Tabla 4-9: Resultados del entrenamiento con el método RANDN.

Elemento	Falla	Entrenamiento con 2 [%] de error
1		0,0
2		0,5
3		0,0
4		4,8
5		0,6
6		0,0
7		0,0
8		1,0
9		0,0
10		0,0
11		0,0
12		0,0
13		0,0
14	X	0,0
15		0,0
16		0,0
17		0,0
18		0,0
19		0,0
20		0,0

Con celeste se indican los falsos positivos y en amarillo la ubicación de un daño menor

Tabla 4-10: Resultado de entrenamiento con distintos % de error para la viga libre con dos fallas (Viga A).

Elemento	Falla	Entrenamiento con 0.5 [%] de error	Entrenamiento con 1 [%] de error	Entrenamiento con 1.5 [%] de error	Entrenamiento con 2 [%] de error
1		0,0	0,0	0,0	0,0
2		0,0	0,0	0,0	0,0
3		0,0	0,0	0,0	0,0
4		0,0	0,0	0,0	0,0
5		0,0	0,0	0,0	0,0
6		0,0	0,0	0,0	0,0
7		0,0	0,0	0,0	0,0
8	X	38,7	59,1	52,2	64,0
9		0,0	0,0	0,0	0,0
10		0,0	0,0	0,0	0,0
11		0,0	0,0	0,0	0,0
12		0,0	4,2	6,4	0,0
13		0,0	0,0	0,0	0,0
14	X	4,4	23,6	16,1	12,3
15		15,8	8,1	17,3	0,0
16		0,0	0,0	0,0	0,0
17		0,0	0,0	0,0	0,0
18		0,0	0,0	0,0	0,0
19		0,0	0,0	0,0	0,0
20		0,0	0,0	0,0	0,0

Con celeste se indican los falsos positivos, en amarillo la ubicación de un daño menor y con rojo la ubicación de un daño mayor.

En la Tabla 4-11 es posible ver los resultados de los entrenamientos aplicados a una viga con dos fallas (Viga B). Nuevamente, la red que mejor desempeño tiene es la con

2[%] de error. Es posible ver que al lado del elemento 17, siempre se detecta una falla, lo mismo pasa con el elemento 8. Esto se debe a que la falla inducida está muy cerca del límite entre ambos elementos, entre el 7 y el 8 en el caso de la falla leve, y entre el 16 y el 17 para la falla grave.

Finalmente, se concluye que la red que mejor desempeño tiene es la con 2[%] de error, por lo que se trabajará con ella de aquí en adelante.

Tabla 4-11: Resultado de entrenamiento con distintos % de error para la viga libre con una falla.

Elemento	Falla	Entrenamiento con 0.5 [%] de error	Entrenamiento con 1 [%] de error	Entrenamiento con 1.5 [%] de error	Entrenamiento con 2 [%] de error
1		0,0	0,0	0,0	0,0
2		0,0	0,0	0,0	0,0
3		0,0	0,0	0,0	0,0
4		74,0	34,8	53,9	0,0
5		0,0	2,1	0,0	0,0
6		0,0	0,0	0,0	0,0
7		20,2	9,1	0,0	16,3
8	X	0,0	0,0	0,0	5,5
9		0,0	22,2	0,0	0,0
10		0,0	3,3	0,0	0,0
11		0,0	0,0	0,0	0,0
12		0,0	0,0	0,0	0,0
13		0,0	0,0	0,0	0,0
14		0,0	0,0	0,0	0,0
15		0,0	0,0	0,0	0,0
16		34,0	20,1	41,5	28,1
17	X	80,0	80,0	80,0	71,0
18		0,0	0,0	0,0	0,0
19		0,0	0,0	0,0	0,0
20		0,0	0,0	0,0	0,0

Con celeste se indican los falsos positivos, en amarillo la ubicación de un daño menor y con rojo la ubicación de un daño mayor.

4.6.2 Estabilidad de entrenamientos

Debido a que la introducción de ruido es aleatoria, se espera que los resultados de un entrenamiento y otro no sean iguales, por lo cual se estudia esto.

En las tablas 4-12, 4-13 y 4-14, es posible ver los resultados entregados al entrenar 3 veces una RNA para la viga con un daño, la viga con dos daños A y la viga con dos daños B respectivamente, con la misma configuración elegida en la sección anterior.

Al ver los resultados, es posible percatarse de que hay una variación tanto en el número de falsos positivos encontrados como en la magnitud de las matrices de rigidez de los elementos con falla. Se puede rescatar que los falsos positivos, en su gran mayoría, son de pequeño porcentaje de reducción de rigidez, lo cual aún hace funcional a las redes ya que estos daños son pequeños.

Para mitigar los efectos de la introducción aleatoria del ruido, se estudia el uso de una base de datos doble y una triple, con esto se busca que para un mismo daño, haya un espectro más amplio sobre el ruido que éste puede contener. También se estudia el uso de una base cuádruple, pero sus resultados fueron poco certeros y no vale la pena incluirlos en el informe. Los resultados se pueden apreciar en las tablas 4-15, 4-16 y 4-17.

Tabla 4-12: Resultados del entrenamiento de la RNA para la viga libre con un daño realizado 3 veces con la misma configuración.

Elemento	Falla	Entrenamiento con 2 [%] de error #1	Entrenamiento con 2 [%] de error #2	Entrenamiento con 2 [%] de error #3
1		0,0	0,0	0,0
2		0,0	0,0	0,0
3		0,0	0,0	0,0
4		12,5	15,5	6,6
5		0,0	0,0	4,2
6		0,0	18,4	0,0
7		0,0	0,0	0,0
8		0,0	4,5	0,0
9		0,0	0,0	3,8
10		0,0	0,0	0,0
11		0,0	0,0	2,1
12		0,0	0,0	2,8
13		0,0	0,0	0,0
14	X	25,7	25,9	22,3
15		0,0	0,0	1,8
16		0,0	0,0	0,0
17		0,0	0,0	0,0
18		0,0	0,0	0,0
19		0,0	0,0	0,0
20		0,0	0,0	0,0

Con celeste se indican los falsos positivos y en amarillo la ubicación de un daño menor.

Tabla 4-13: Resultados del entrenamiento de la RNA para la viga libre con daños A realizado 3 veces con la misma configuración.

Elemento	Falla	Entrenamiento con 2 [%]de error #1	Entrenamiento con 2 [%] de error #2	Entrenamiento con 2 [%] de error #3
1		0,0	0,0	0,0
2		0,0	0,0	0,0
3		0,0	2,3	0,0
4		0,0	0,0	0,0
5		0,0	0,0	0,0
6		0,0	0,0	0,6
7		0,0	0,0	0,0
8	X	64,0	59,9	55,3
9		0,0	0,0	6,9
10		0,0	11,1	0,0
11		0,0	5,0	9,5
12		0,0	0,0	4,3
13		0,0	0,0	0,0
14	X	12,3	11,2	5,6
15		0,0	0,0	7,1
16		0,0	0,0	0,0
17		0,0	0,0	0,0
18		0,0	0,0	0,0
19		0,0	0,0	0,0
20		0,0	0,0	0,0

Con celeste se indican los falsos positivos, en amarillo la ubicación de un daño menor y con rojo la ubicación de un daño mayor.

Tabla 4-14: Resultados del entrenamiento de la RNA para la viga libre con daños B realizado 3 veces con la misma configuración.

Elemento	Falla	Entrenamiento con 2 [%] de error #1	Entrenamiento con 2 [%] de error #2	Entrenamiento con 2 [%] de error #3
1		0,0	0,0	0,0
2		0,0	0,0	2,8
3		0,0	0,0	0,0
4		0,0	0,0	0,0
5		0,0	0,0	18,5
6		0,0	0,0	0,0
7		42,8	16,3	24,9
8	X	0,0	5,5	0,0
9		0,0	0,0	9,7
10		0,0	0,0	0,0
11		0,0	0,0	0,0
12		0,0	0,0	0,0
13		0,0	0,0	3,7
14		0,0	0,0	0,0
15		0,0	0,0	0,0
16		26,5	28,1	28,0
17	X	80,0	71,0	56,5
18		0,0	0,0	0,0
19		0,0	0,0	0,0
20		0,0	0,0	0,0

Con celeste se indican los falsos positivos, en amarillo la ubicación de un daño menor y con rojo la ubicación de un daño mayor.

Tabla 4-15: Resultados de 3 entrenamientos para una base de datos doble y una triple con similares configuraciones para la viga libre con un daño.

Elemento	Falla	Base Doble #1, reducción de rigidez [%]	Base Doble #2, reducción de rigidez [%]	Base Doble #3, reducción de rigidez [%]	Base Triple #1, reducción de rigidez [%]	Base Triple #2, reducción de rigidez [%]	Base Triple #3, reducción de rigidez [%]
1		0,0	0,0	0,0	0,0	0,0	0,0
2		0,0	0,0	0,0	0,0	0,0	2,7
3		0,0	0,0	0,0	0,0	0,0	0,0
4		0,0	2,3	0,0	8,5	0,0	0,0
5		0,0	0,0	0,0	5,9	0,0	0,0
6		0,0	0,0	0,0	2,5	0,0	0,0
7		0,0	0,0	0,0	1,4	0,0	0,0
8		0,0	0,0	0,0	0,0	0,0	4,9
9		0,0	0,0	0,0	0,0	0,0	0,0
10		0,0	0,0	0,0	0,0	0,0	0,0
11		0,0	0,0	0,0	0,0	0,0	0,0
12		0,0	0,0	0,0	0,0	0,0	0,0
13		0,0	0,0	0,0	0,0	0,0	0,0
14	X	28,5	37,7	37,9	37,7	37,1	34,1
15		0,0	0,0	0,0	0,0	0,0	0,8
16		0,0	0,0	0,0	0,0	0,0	0,0
17		0,0	0,0	0,0	0,0	0,0	0,0
18		0,0	0,0	0,0	0,0	0,0	0,0
19		0,0	4,6	0,0	0,0	0,0	0,0
20		0,0	0,0	0,0	0,0	0,0	0,0

Con celeste se indican los falsos positivos y en amarillo la ubicación de un daño menor.

Tabla 4-16: Resultados de 3 entrenamientos para una base de datos doble y para la base de datos triple con similares configuraciones para la viga libre con 2 daños A.

Elemento	Falla	Base Doble #1, reducción de rigidez [%]	Base Doble #2, reducción de rigidez [%]	Base Doble #3, reducción de rigidez [%]	Base Triple #1, reducción de rigidez [%]	Base Triple #2, reducción de rigidez [%]	Base Triple #3, reducción de rigidez [%]
1		0,0	0,0	0,0	0,0	0,0	0,0
2		0,0	0,0	1,0	0,0	0,0	0,0
3		0,0	0,0	0,0	0,0	0,0	0,0
4		0,0	0,0	0,0	0,0	0,0	0,0
5		0,0	0,0	0,0	0,8	0,0	0,0
6		0,0	0,0	0,0	0,0	0,0	0,0
7		17,9	0,0	0,0	12,8	0,0	0,0
8	X	48,9	44,3	62,5	46,6	50,9	66,5
9		0,0	0,0	0,0	0,0	0,0	0,0
10		0,0	0,0	0,0	0,0	0,0	0,0
11		0,0	0,0	0,0	0,0	0,0	0,0
12		1,0	0,0	0,0	2,6	0,0	0,0
13		0,0	0,0	0,0	6,6	0,0	4,8
14	X	25,1	28,6	13,0	10,1	29,6	8,0
15		3,2	3,7	3,1	13,3	15,8	16,0
16		0,0	0,0	0,0	0,0	0,0	0,0
17		0,0	0,0	0,0	0,0	0,0	0,0
18		0,0	0,0	0,0	0,0	0,0	0,0
19		0,0	0,0	0,0	0,0	0,0	0,0
20		0,0	0,0	0,0	0,0	0,0	0,0

Con celeste se indican los falsos positivos, en amarillo la ubicación de un daño menor y con rojo la ubicación de un daño mayor.

Tabla 4-17: Resultados de 3 entrenamientos para una base de datos doble y para la base de datos triple con similares configuraciones para la viga libre con 2 daños B.

Elemento	Falla	Base Doble #1, reducción de rigidez [%]	Base Doble #2, reducción de rigidez [%]	Base Doble #3, reducción de rigidez [%]	Base Triple #1, reducción de rigidez [%]	Base Triple #2, reducción de rigidez [%]	Base Triple #3, reducción de rigidez [%]
1		0,0	0,0	0,0	0,0	0,0	0,0
2		0,0	7,7	0,0	0,0	0,0	0,0
3		10,1	0,0	15,7	4,8	0,0	0,0
4		10,0	0,0	0,0	3,3	11,5	7,8
5		0,0	0,0	0,0	0,0	0,0	0,0
6		0,0	0,0	0,0	18,9	6,3	0,0
7		26,5	2,8	25,2	0,0	6,9	3,7
8	X	4,8	0,0	0,0	0,0	0,0	0,0
9		0,0	0,0	0,0	0,0	0,0	0,0
10		0,0	0,0	0,0	0,0	0,0	0,0
11		0,0	0,0	4,7	0,0	0,0	0,0
12		5,4	0,0	0,0	0,0	0,0	0,0
13		0,0	0,0	0,0	2,7	0,0	0,0
14		0,0	0,0	0,0	0,0	0,0	0,0
15		0,0	0,0	0,0	0,0	0,0	0,0
16		41,4	11,9	44,8	33,2	46,8	9,5
17	X	76,0	75,3	80,0	80,0	80,0	80,0
18		0,0	0,0	0,0	0,0	0,0	0,0
19		0,0	0,0	0,0	0,0	0,0	0,0
20		0,0	0,0	0,0	0,0	0,0	0,0

Con celeste se indican los falsos positivos, en amarillo la ubicación de un daño menor y con rojo la ubicación de un daño mayor.

5 Resultados problema 2

En el problema 2 se procede de la misma forma que el problema 1, solo que en este caso la barra estará empotrada por uno de sus extremos.

5.1 Generación de modelo y base de datos en elementos finitos viga empotrada

Los datos de la viga empotrada, su discretización y matrices de rigidez y de masa son los mismos del problema anterior, lo que cambia son los grados de libertad, la cantidad de puntos de medición de las anti-resonancias y el número de anti-resonancias utilizadas en cada caso para entrenar la RNA. En la Figura 5-1 se puede ver un esquema de la viga empotrada.

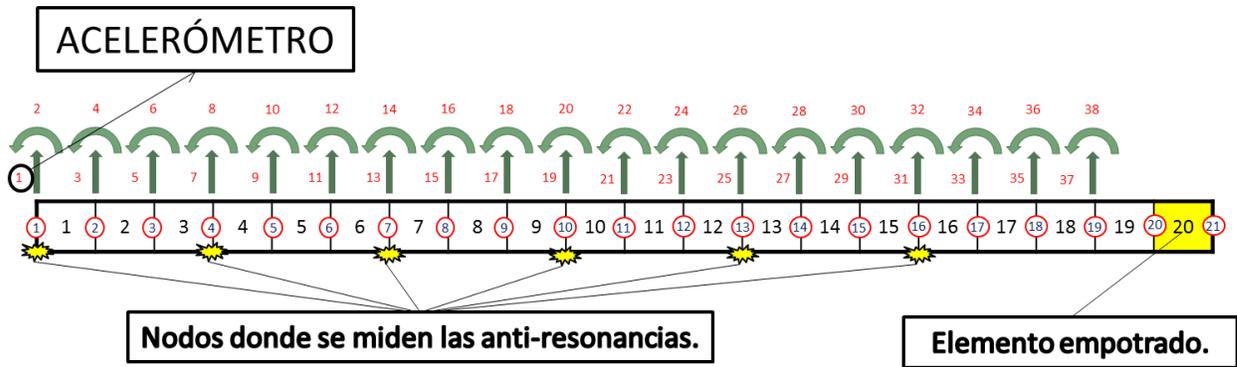


Figura 5-1: Esquema de la viga empotrada.

Acá se puede ver que el elemento 20 está empotrado, por lo cual se eliminan los grados de libertad de los nodos 20 y 21.

El código utilizado para generar el modelo y para generar la base de datos está en el anexo E y F de este informe.

5.2 Variables de entrada

En el número de anti-resonancias se mantienen los mismos principios que en el problema anterior, pero como se tiene un punto de medición menos, la cantidad de anti-resonancias usadas para detectar daños disminuye de 23 a 22. En la Tabla 5-1 se encuentra la cantidad de anti-resonancias que se toman por cada punto de medición.

Tabla 5-1: Número de anti-resonancias cruzadas tomadas por cada sensor en la viga empotrada.

Acelerómetro	Anti-resonancias utilizadas [Hz]
1	7
2	5
3	4
4	3
5	2
6	1
Total	22

5.3 Entrenamiento de la red neuronal para la viga empotrada

El entrenamiento de la RNA se hace con la misma configuración que el problema anterior, pero con 22 entradas en vez de 23:

- Tipo: Perceptrón Multicapa (MLP)
- N° de entradas: 22 (anti-resonancias)
- N° de capas ocultas: 1
- N° de neuronas de salida: 18 (Disminución de la rigidez de los elementos de la viga sin contar los extremos)
- Método de entrenamiento: Levenberg-Marquardt
- Neuronas en la capa oculta: 140

5.4 Validación analítica de la red neuronal

Para este caso, al igual que el anterior, se realiza una validación analítica de la red neuronal.

En la Figura 5-2 es posible ver los Falsos Positivos de la viga empotrada, los cuales son similares que en el caso de la viga libre. Este gráfico indica que del 100% de los daños encontrados, alrededor de un 50% de ellos no existe. Esta cifra es alta, pero considerando que está concentrada en el tramo de fallas con un nivel menor al 10%, se puede decir que son despreciables.

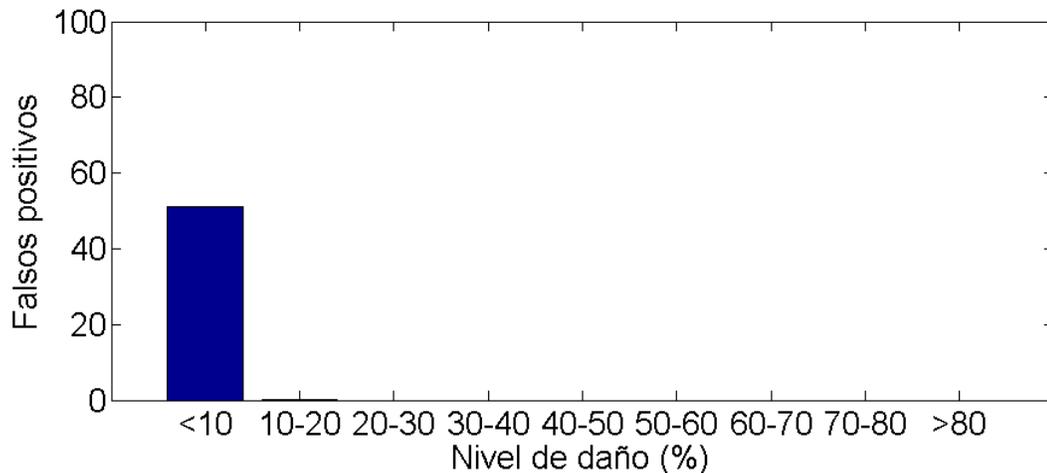


Figura 5-2: Falsos positivos encontrados al entrenar la red con 140 neuronas en la capa oculta para la viga empotrada.

En la Figura 5-3 es posible ver los Falsos negativos de la viga empotrada, este gráfico indica que del 100% de los daños, alrededor de un 2% no fueron detectados. Esta cifra es baja, y considerando que está concentrada en el tramo de fallas con un nivel menor al 10%, se puede aprobar con seguridad el funcionamiento analítico de la red.

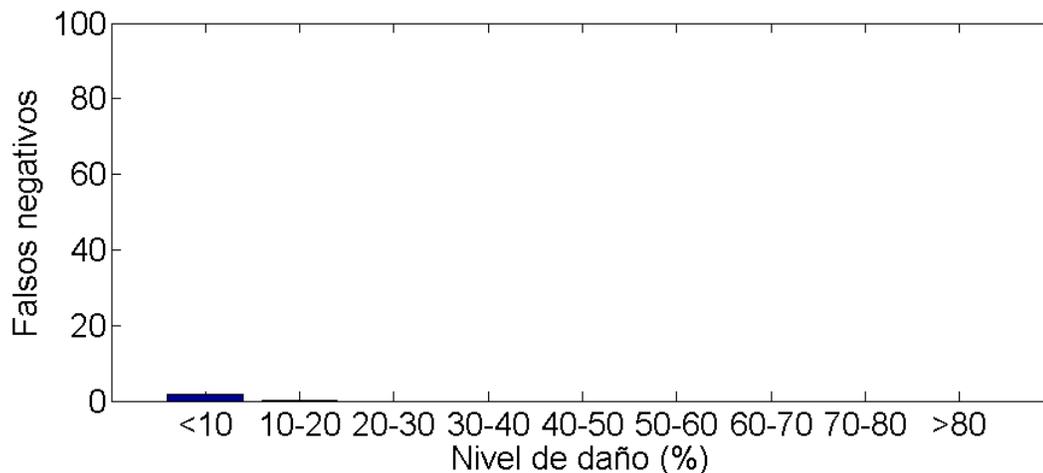


Figura 5-3 : Falsos negativos encontrados al entrenar la red con 140 neuronas en la capa oculta para la viga empotrada.

5.5 Toma de datos experimentales

Para la toma de datos se sigue el mismo procedimiento del problema anterior, solo que en este caso la viga tiene menos nodos que medir y está que ser empotrada.

En la Figura 5-4 se aprecia un diseño del sistema de empotramiento que se utilizó, luego, en la Figura 5-5 se encuentra una foto del sistema de empotramiento implementado.

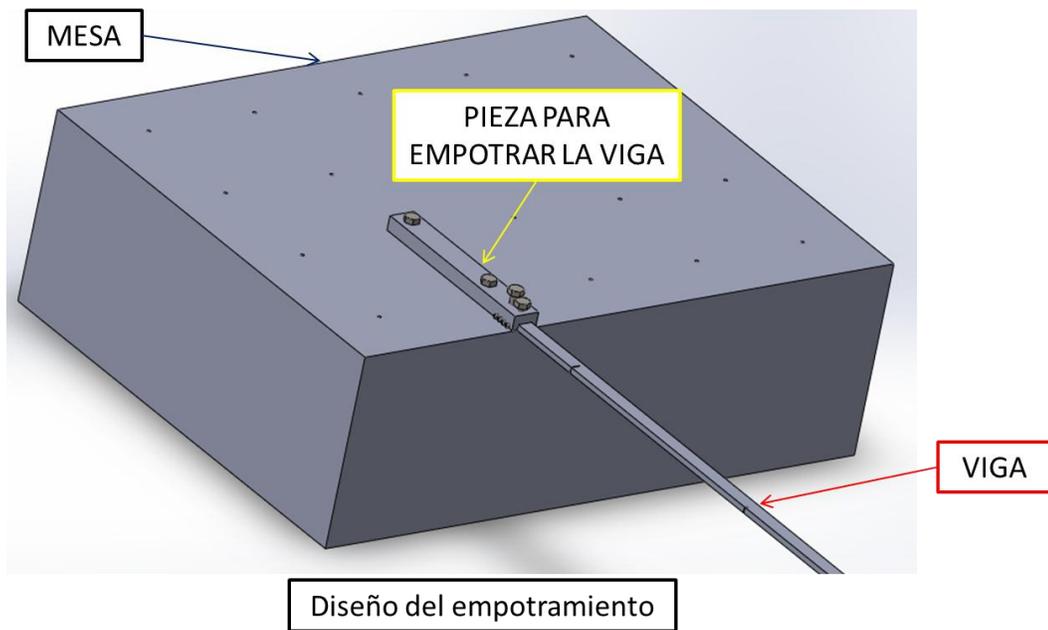


Figura 5-4: Diseño del sistema de empotramiento.

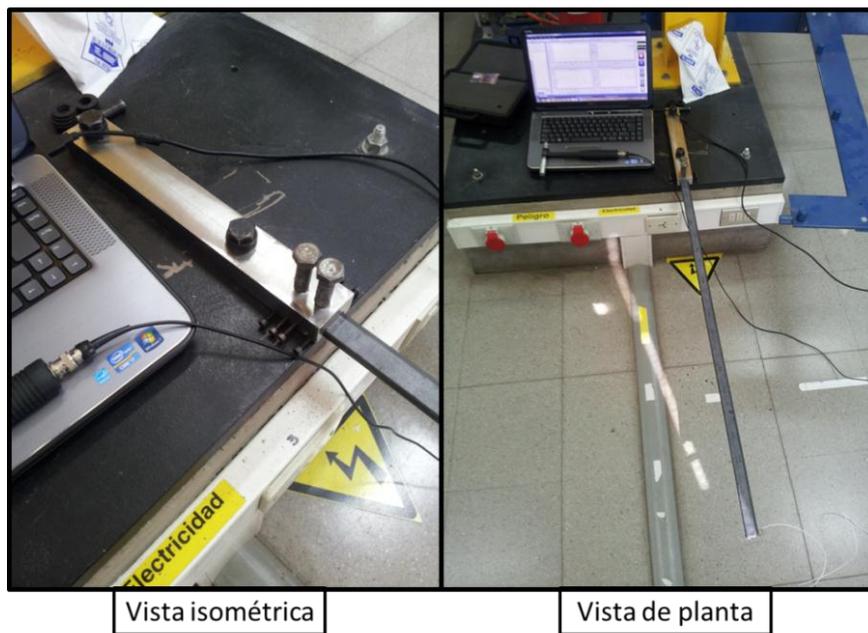


Figura 5-5: Sistema de empotramiento implementado.

En la Tabla 5-2 se encuentran las anti-resonancias experimentales y analíticas para la viga sin daños

Tabla 5-2: Anti-Resonancias experimentales y analíticas para la viga empotrada sin daño.

Anti-resonancias Analíticas [Hz]	Anti-resonancias Experimentales [HZ]
3334	3058,00
2476	2296,91
1747	1648,95
1144	1091,66
669	641,28
321	307,11
99	97,68
2463	2263,55
1601	1489,56
916	867,55
422	403,73
124	119,93
2446	2221,49
1432	1334,27
686	643,95
194	187,87
2417	2135,18
1158	1053,71
362	338,02
2364	2030,74
734	653,64
2250	1752,86

Para la obtención de estas anti-resonancias se utiliza el método del algoritmo base para la identificación de las frecuencias de anti-resonancia a través de funciones de transmisibilidad, explicado en la sección (2.2.2).

Al igual que en el caso anterior, se aplica un ajuste de modelo para que las anti-resonancias analíticas se acerquen más a las experimentales. En la Tabla 5-3 se encuentran las anti-resonancias con y sin ajuste, las anti-resonancias experimentales y los resultados de la aplicación del ajuste expresados como una diferencia porcentual entre las anti-resonancias con y sin ajuste con respecto a las experimentales.

Es posible ver que el ajuste cumple su objetivo, ya que la diferencia porcentual entre los resultados analíticos se acerca más a los experimentales.

Tabla 5-3: Tabla con las anti-resonancias sin ajuste y con ajuste de modelo para la viga empotrada.

Anti-resonancias sin ajuste [Hz]	Anti-resonancias con ajuste [Hz]	Anti-resonancias experimentales [Hz]	Diferencia % de las anti-resonancias sin ajuste c/r a las experimentales	Diferencia % de las anti-resonancias con ajuste c/r a las experimentales
3431	3334	3058,00	12,18	9,04
2548	2476	2296,91	10,93	7,81
1797	1747	1648,95	8,98	5,93
1177	1144	1091,66	7,86	4,83
688	669	641,28	7,36	4,34
329,95	320,7	307,11	7,44	4,42
102	99	97,68	4,23	1,31
2534	2463	2263,55	11,97	8,83
1647	1601	1489,56	10,60	7,49
943	916	867,55	8,64	5,60
434	422	403,73	7,49	4,47
128	124	119,93	6,36	3,38
2517	2446	2221,49	13,29	10,11
1473	1432	1334,27	10,43	7,34
706	686	643,95	9,59	6,52
199	194	187,87	6,16	3,18
2487	2417	2135,18	16,48	13,21
1191	1158	1053,71	13,04	9,87
373	362	338,02	10,21	7,12
2432	2364	2030,74	19,77	16,41
755	734	653,64	15,56	12,32
2315	2250	1752,86	32,09	28,38

5.6 Estudio de la adición de ruido a los datos analíticos

Al igual que en el caso anterior, se estudia la adición de ruido a los datos analíticos, para lograr que se acerquen más aún a los experimentales.

5.6.1 Porcentaje de ruido agregado

Se procede de la misma forma que el problema anterior. En la Tabla 5-4 están los resultados del entrenamiento de la viga empotrada con una falla. Es posible ver que los entrenamientos con 0.5 [%] de error y con 1 [%] de error son poco certeros por tener muchos falsos positivos, no así los entrenamientos con 1.5[%] de error y 2[%] de error, los cuales logran encontrar la falla y prácticamente no presentan falsos positivos.

En la tabla Tabla 5-5 están los resultados del entrenamiento de la viga empotrada con dos fallas A. Esta red se comporta similar a la anterior. Se ve que en el elemento 18 se detecta un falso positivo en todos los entrenamientos, esto se le atribuye a que ese es el primer elemento que está después del empotramiento, por lo que la información obtenida de este queda un poco distorsionada.

Tabla 5-4: Resultado de entrenamiento con distintos % de error para la viga empotrada con una falla.

Elemento	Falla	Entrenamiento con 0,5 [%] de error	Entrenamiento con 1 [%] de error	Entrenamiento con 1,5 [%] de error	Entrenamiento con 2[%] de error
1		0,00	0,00	0,00	0,00
2		0,00	0,00	0,00	0,00
3		0,00	0,00	0,00	0,00
4		0,00	0,00	0,00	2,94
5		0,00	0,00	0,00	0,00
6		0,00	0,00	0,00	0,00
7		0,00	0,00	0,00	0,00
8		0,00	0,00	0,00	0,00
9		1,27	7,94	0,00	0,00
10		0,28	1,21	0,00	0,00
11		4,16	0,00	0,00	0,00
12		0,00	0,00	0,00	0,00
13		0,00	0,00	0,00	0,00
14	X	33,71	33,71	32,84	36,29
15		1,21	0,00	0,00	0,00
16		2,64	0,00	0,00	0,00
17		0,00	0,00	0,00	0,00
18		11,95	1,72	0,00	0,00
19		0,00	1,90	0,00	0,00
20		0,00	0,00	0,00	0,00

Con celeste se indican los falsos positivos y en amarillo la ubicación de un daño menor.

Tabla 5-5: Resultado de entrenamiento con distintos % de error para la viga empotrada con 2 fallas A.

Elemento	Falla	Entrenamiento con 0,5 [%] de error	Entrenamiento con 1 [%] de error	Entrenamiento con 1,5 [%] de error	Entrenamiento con 2[%] de error
1		0,00	0,00	0,00	0,00
2		0,00	0,00	0,00	0,00
3		0,00	5,26	0,00	0,00
4		0,00	0,00	0,00	3,31
5		0,00	0,00	0,00	0,00
6		0,00	0,00	0,00	0,00
7		61,00	22,97	38,26	31,57
8	X	52,61	57,08	48,19	49,05
9		0,00	0,00	0,00	0,00
10		0,00	0,00	36,54	0,00
11		0,00	0,00	0,00	0,00
12		0,00	0,00	0,00	0,00
13		0,00	0,00	0,00	0,00
14	X	0,00	0,00	3,81	0,00
15		33,40	11,38	0,00	0,00
16		2,90	0,00	0,00	2,41
17		0,00	0,00	0,00	0,00
18		14,61	4,71	19,23	27,65
19		0,00	0,00	0,00	0,00
20		0,00	0,00	0,00	0,00

Con celeste se indican los falsos positivos, en amarillo la ubicación de un daño menor y con rojo la ubicación de un daño mayor.

En la tabla Tabla 5-6 están los resultados del entrenamiento de la viga empotrada con dos fallas B. En este caso la red entrenada con un 2[%] de error es claramente superior a las otras redes. Si bien hay un falso negativo en el elemento 8, se detecta una falla en el elemento 7, esto debido a que la falla inducida en la viga ensayada se encuentra muy cercana a este elemento.

Al igual que en el problema 1, la red que mejor se desempeña es la con 2[%] de ruido. Se trabajará con esta red en los siguientes casos.

Tabla 5-6: Resultado de entrenamiento con distintos % de error para la viga empotrada con dos fallas B.

Elemento	Falla	Entrenamiento con 0,5 [%] de error	Entrenamiento con 1 [%] de error	Entrenamiento con 1,5 [%] de error	Entrenamiento con 2[%] de error
1		0,00	0,00	0,00	0,00
2		0,00	0,00	0,00	0,00
3		16,15	0,00	56,89	0,00
4		0,00	0,00	0,00	0,00
5		0,00	1,42	0,00	0,00
6		0,00	0,00	0,00	0,00
7		60,18	0,00	1,36	21,33
8	X	0,00	0,00	0,00	0,00
9		0,00	0,00	0,00	0,00
10		0,00	0,00	0,00	0,00
11		36,71	27,60	0,00	9,87
12		11,47	0,00	0,00	0,00
13		0,00	0,00	0,00	0,00
14		9,11	0,00	0,00	0,00
15		0,00	0,00	0,00	0,00
16		64,33	49,15	47,53	70,83
17	X	26,47	24,99	0,00	34,33
18		0,00	0,00	0,00	0,00
19		0,00	0,00	0,00	0,00
20		0,00	0,00	0,00	0,00

Con celeste se indican los falsos positivos, en amarillo la ubicación de un daño menor y con rojo la ubicación de un daño mayor.

5.6.2 Estabilidad de entrenamientos

Al igual que en el problema 1, en las tablas 5-7, 5-8, y 5-9, es posible ver los resultados entregados al entrenar 3 veces una RNA para la viga con un daño, la viga con dos daños A y la viga con dos daños B respectivamente, con la misma configuración elegida en la sección anterior. También se estudia el uso de bases de datos dobles y triples en la estabilidad de la introducción de ruido a los datos. En las tablas 5-10, 5-11 y 5-12, están los resultados de este estudio.

Tabla 5-7: Resultados del entrenamiento de la RNA para la viga empotrada con un daño realizado 3 veces con la misma configuración.

Elemento	Falla	Entrenamiento con 2 [%] de error #1	Entrenamiento con 2 [%] de error #2	Entrenamiento con 2 [%] de error #3
1		0,00	0,00	0,00
2		0,00	0,00	0,00
3		0,00	0,00	0,00
4		2,94	0,00	0,00
5		0,00	0,00	0,00
6		0,00	0,00	0,00
7		0,00	0,00	0,00
8		0,00	0,00	0,00
9		0,00	0,00	0,00
10		0,00	0,00	0,00
11		0,00	0,00	0,00
12		0,00	0,00	0,00
13		0,00	0,00	0,00
14	X	36,29	35,14	26,39
15		0,00	0,00	0,00
16		0,00	0,00	0,00
17		0,00	0,00	0,00
18		0,00	0,00	0,00
19		0,00	0,00	0,00
20		0,00	0,00	0,00

Con celeste se indican los falsos positivos y en amarillo la ubicación de un daño menor

Tabla 5-8: Resultados del entrenamiento de la RNA para la viga empotrada con 2 daños A realizado 3 veces con la misma configuración.

Elemento	Falla	Entrenamiento con 2 [%] de error #1	Entrenamiento con 2 [%] de error #2	Entrenamiento con 2 [%] de error #3
1		0,00	0,00	0,00
2		0,00	3,78	1,93
3		0,00	0,00	0,00
4		3,31	0,00	0,00
5		0,00	0,00	0,00
6		0,00	0,00	0,00
7		31,57	18,85	48,12
8	X	49,05	46,06	38,10
9		0,00	9,73	0,00
10		0,00	0,00	0,00
11		0,00	0,00	0,00
12		0,00	0,00	0,00
13		0,00	0,00	0,00
14	X	0,00	2,20	7,07
15		0,00	0,00	14,24
16		2,41	0,00	0,00
17		0,00	0,00	0,00
18		27,65	12,68	0,00
19		0,00	0,00	0,00
20		0,00	0,00	0,00

Con celeste se indican los falsos positivos, en amarillo la ubicación de un daño menor y con rojo la ubicación de un daño mayor.

Tabla 5-9: Resultados del entrenamiento de la RNA para la viga empotrada con dos daños B realizado 3 veces con la misma configuración.

Elemento	Falla	Entrenamiento con 2 [%] de error #1	Entrenamiento con 2 [%] de error #2	Entrenamiento con 2 [%] de error #3
1		0,00	0,00	0,00
2		0,00	6,37	0,00
3		0,00	21,89	15,02
4		0,00	0,00	0,00
5		0,00	0,00	0,00
6		0,00	3,79	0,00
7		21,33	28,27	18,82
8	X	0,00	0,00	0,00
9		0,00	0,00	0,00
10		0,00	0,00	0,00
11		9,87	29,35	0,00
12		0,00	0,00	0,00
13		0,00	0,00	0,00
14		0,00	0,00	0,00
15		0,00	0,00	0,00
16		70,83	34,10	61,30
17	X	34,33	12,10	23,28
18		0,00	0,00	0,00
19		0,00	0,00	0,00
20		0,00	0,00	0,00

Con celeste se indican los falsos positivos, en amarillo la ubicación de un daño menor y con rojo la ubicación de un daño mayor.

Tabla 5-10: Resultados de 3 entrenamientos para una base de datos doble y para la base de datos triple con similares configuraciones para la viga empotrada con 1 daño.

Elemento	Falla	Base Doble #1, reducción de rigidez [%]	Base Doble #2, reducción de rigidez [%]	Base Doble #3, reducción de rigidez [%]	Base Triple #1, reducción de rigidez [%]	Base Triple #2, reducción de rigidez [%]	Base Triple #3, reducción de rigidez [%]
1		0,00	0,00	0,00	0,00	0,00	0,00
2		0,00	0,00	0,00	0,00	0,00	0,00
3		0,00	0,00	0,00	0,00	0,00	0,00
4		0,00	0,00	0,00	0,00	0,00	0,00
5		0,00	0,00	0,00	0,00	0,00	0,00
6		0,00	0,00	0,00	0,00	0,00	0,00
7		0,00	0,00	0,00	0,00	0,00	0,00
8		0,00	0,00	0,00	0,00	0,00	0,00
9		0,00	0,00	0,00	0,00	0,41	0,00
10		0,00	0,00	2,34	0,00	0,00	0,00
11		0,00	0,00	0,00	0,00	0,00	0,00
12		0,00	0,00	0,00	0,00	0,00	0,00
13		0,00	0,00	0,00	0,00	0,00	0,00
14	X	35,75	37,54	23,34	31,91	33,39	30,12
15		0,00	0,00	0,00	0,00	0,00	0,00
16		0,00	0,00	0,00	0,00	0,00	0,00
17		0,00	0,00	0,00	0,00	0,00	0,00
18		0,00	0,00	0,00	0,00	20,33	0,00
19		0,00	0,00	0,00	0,00	0,00	0,00
20		0,00	0,00	0,00	0,00	0,00	0,00

Con celeste se indican los falsos positivos y en amarillo la ubicación de un daño menor

Tabla 5-11: Resultados de 3 entrenamientos para una base de datos doble y para la base de datos triple con similares configuraciones para la viga empotrada con 2 daños A.

Elemento	Falla	Base Doble #1, reducción de rigidez [%]	Base Doble #2, reducción de rigidez [%]	Base Doble #3, reducción de rigidez [%]	Base Triple #1, reducción de rigidez [%]	Base Triple #2, reducción de rigidez [%]	Base Triple #3, reducción de rigidez [%]
1		0,00	0,00	0,00	0,00	0,00	0,00
2		0,00	0,00	0,00	0,00	0,00	0,00
3		0,00	0,00	0,00	0,00	0,00	3,58
4		7,53	0,00	0,00	0,00	0,00	0,00
5		0,00	0,00	0,00	0,00	0,00	0,00
6		0,00	0,00	0,00	0,00	0,00	0,00
7		0,00	0,00	38,34	9,16	46,28	0,11
8	X	64,65	61,99	55,98	60,43	42,70	31,39
9		0,00	0,00	0,00	0,00	0,00	0,00
10		0,00	0,00	0,00	0,00	0,00	0,00
11		0,00	0,00	0,00	0,00	0,00	0,00
12		0,00	0,00	0,00	0,00	0,00	0,00
13		0,00	0,00	0,00	0,00	0,00	0,00
14	X	3,99	0,00	9,01	4,45	0,00	2,31
15		7,61	10,75	0,00	7,38	18,72	0,00
16		0,00	0,00	2,68	0,00	0,00	0,00
17		0,00	0,00	0,00	0,00	0,00	0,00
18		14,10	30,37	0,00	22,91	31,34	38,19
19		1,93	0,00	0,00	0,00	0,00	0,00
20		0,00	0,00	0,00	0,00	0,00	0,00

Con celeste se indican los falsos positivos, en amarillo la ubicación de un daño menor y con rojo la ubicación de un daño mayor.

Tabla 5-12: Resultados de 3 entrenamientos para una base de datos doble y para la base de datos triple con similares configuraciones para la viga empotrada con 2 daños B.

Elemento	Falla	Base Doble #1, reducción de rigidez [%]	Base Doble #2, reducción de rigidez [%]	Base Doble #3, reducción de rigidez [%]	Base Triple #1, reducción de rigidez [%]	Base Triple #2, reducción de rigidez [%]	Base Triple #3, reducción de rigidez [%]
1		0,00	0,00	0,00	0,00	0,00	0,00
2		0,00	0,00	0,00	0,00	0,00	0,00
3		0,00	7,86	10,97	27,68	0,00	36,38
4		0,00	0,00	0,00	0,00	0,00	0,00
5		0,00	0,00	0,00	0,00	0,00	0,00
6		0,00	0,00	0,00	0,00	0,00	0,00
7		16,12	26,46	25,48	16,99	0,00	14,79
8	X	0,00	0,00	0,00	0,00	0,00	0,00
9		0,00	0,00	0,00	0,00	0,00	0,00
10		0,00	0,00	0,00	0,00	0,00	0,00
11		23,85	36,53	17,63	0,00	16,41	29,91
12		0,00	0,00	0,00	0,00	0,00	0,00
13		0,00	0,00	0,00	0,00	0,00	0,00
14		0,00	0,00	4,33	0,00	0,00	0,00
15		0,00	0,00	0,00	0,00	0,00	0,00
16		52,80	80,00	56,41	50,07	54,45	36,11
17	X	18,80	5,10	0,00	5,75	5,33	5,65
18		0,00	0,00	0,00	0,00	0,00	0,00
19		0,34	0,00	0,00	0,00	0,00	0,00
20		0,00	0,00	0,00	0,00	0,00	0,00

Con celeste se indican los falsos positivos, en amarillo la ubicación de un daño menor y con rojo la ubicación de un daño mayor.

6 Resultados finales

Finalmente, se concluye que la mejor RNA entrenada fue la con ruido de 2[%] y base doble. Los daños detectados se muestran en las figuras 6-1, 6-2 y 6-3. Estos daños son el promedio de las 3 RNA entrenadas para cada caso, esto se hace para generalizar el entrenamiento, ya que si se elige el mejor caso, no se tendrán resultados representativos.

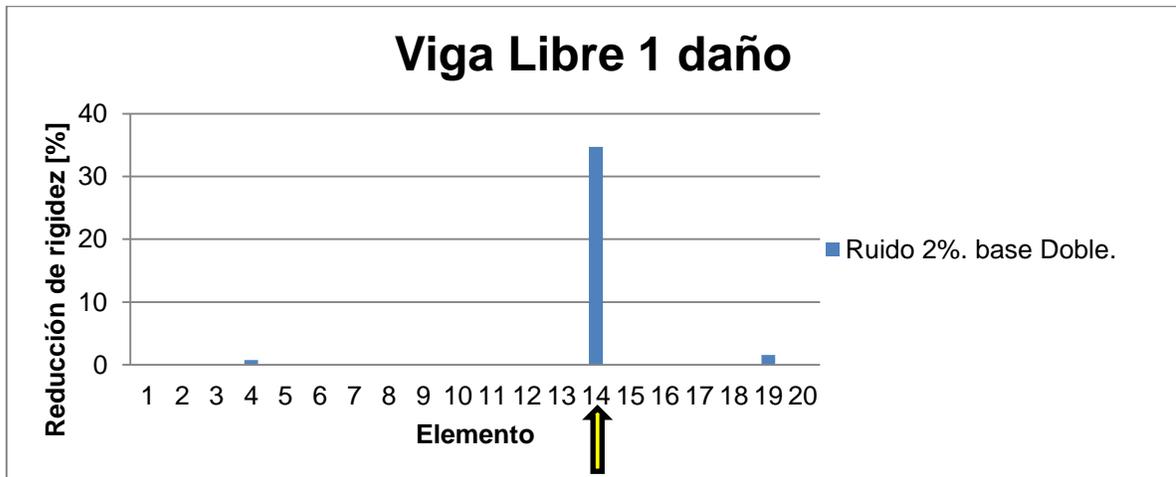


Figura 6-1: Resultados detección daños de la viga libre con 1 daño en el elemento #14.

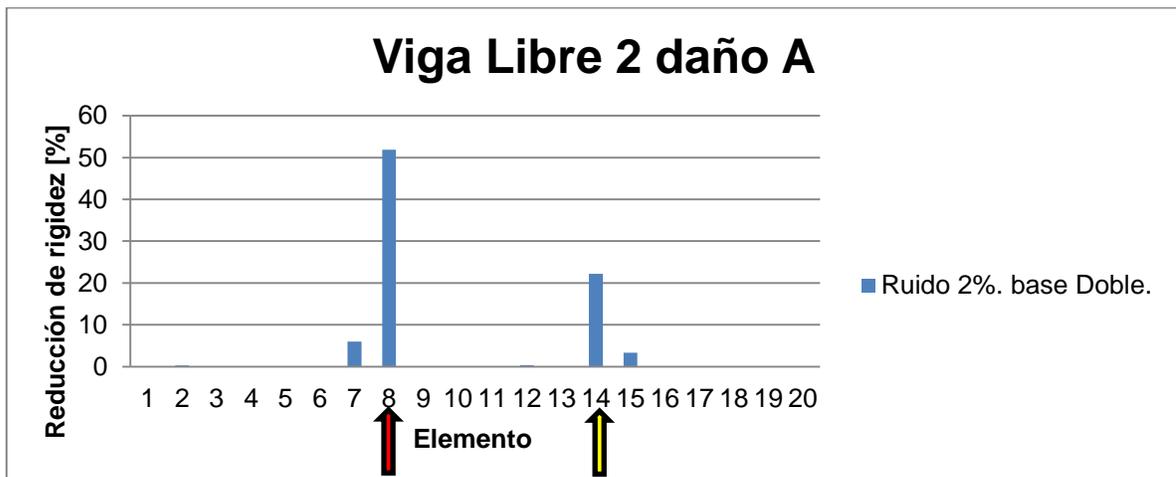


Figura 6-2: Resultados detección daños de la viga libre con 2 daños A en los elementos # 8 y #14.

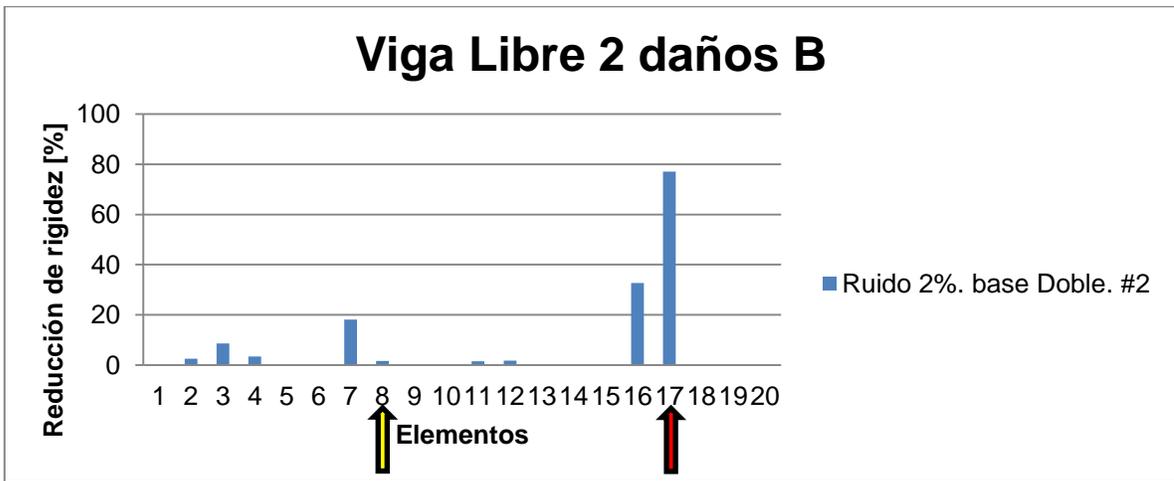


Figura 6-3: Resultados detección daños de la viga libre con 2 daños B en los elementos # 8 y #17.

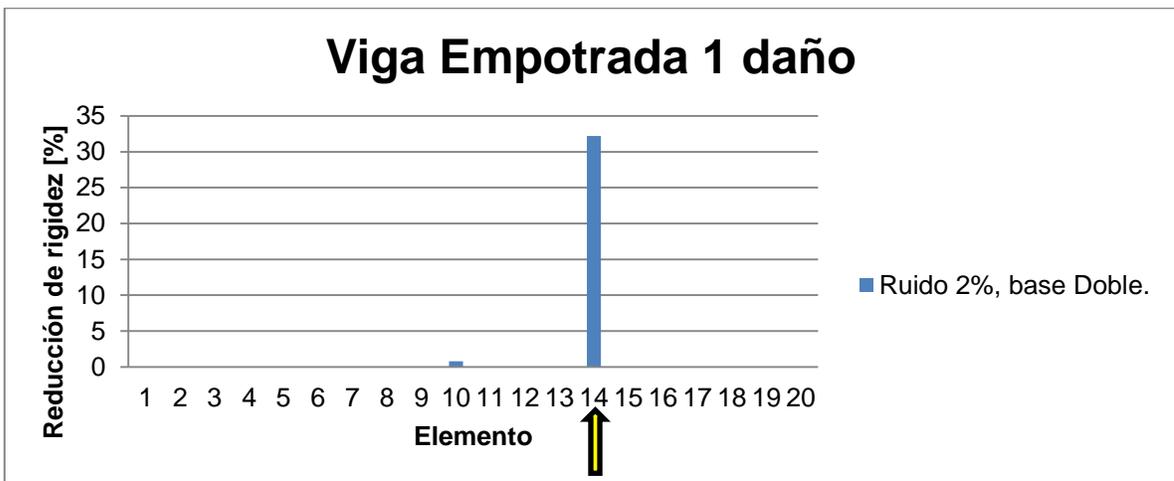


Figura 6-4: Resultados detección daños de la viga empotrada con 1 daño en el elemento #14.

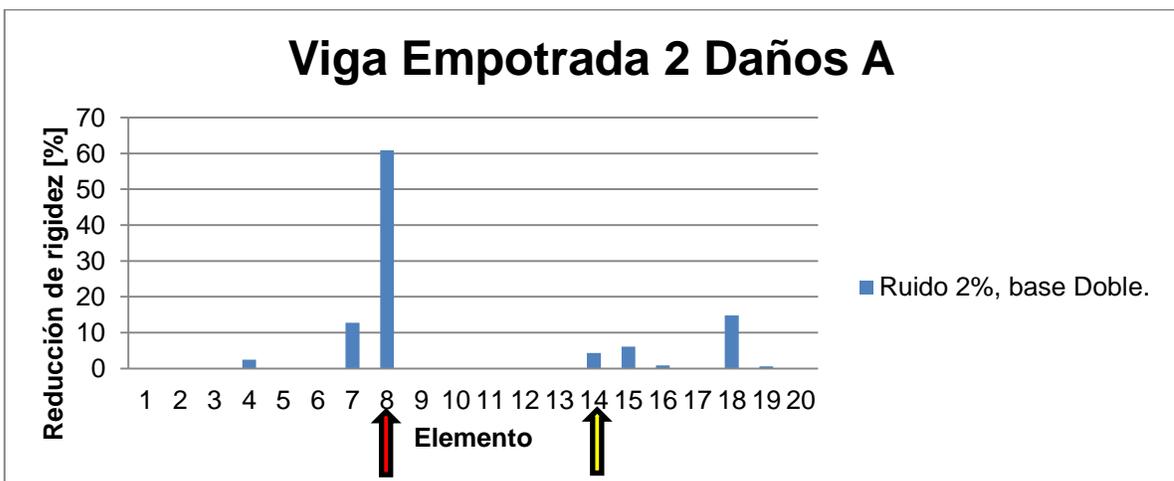


Figura 6-5: Resultados detección daños de la viga empotrada con 2 daños A en los elementos # 8 y #14.

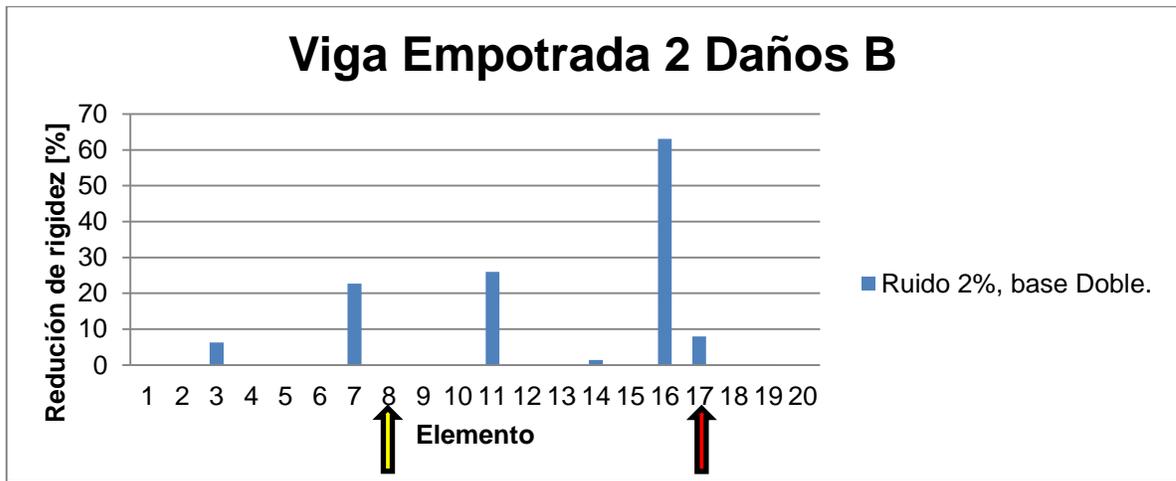


Figura 6-6: Resultados detección daños de la viga empotrada con 2 daños B en los elementos # 8 y #17.

7 Discusiones

En el presente trabajo de tesis se encuentra un algoritmo que detecta fallas en una viga con condiciones de borde libre-libre y en otra viga con condiciones empotrada-libre usando anti-resonancias cruzadas

Para el caso de la viga libre con un solo daño se logran resultados bastante certeros. Se aprecian dos falsos positivos en los elementos #4 y #19, pero son muy pequeños ($>2\%$), por lo que se puede decir que los resultados para este caso son confiables. Para la viga libre con 2 daños A se logran detectar claramente las fallas inducidas, tanto como en su posición como en su reducción de rigidez. Se aprecian dos falsos positivos adyacentes a los elementos con fallas (elementos #7 y #15), como ya se explicó en los resultados, esto se debe a la cercanía de la falla a la zona limítrofe entre un elemento y otro. Los resultados para la viga libre con dos daños B tiene más falsos positivos que los entrenamientos anteriores, y la falla pequeña inducida en el elemento #8 se detecta en el elemento #7. Con respecto al trabajo realizado anteriormente por Javier Mahu [2] en la viga libre, en las figuras 7-1, 7-2 y 7-3 se encuentran los resultados obtenidos al entrenar las redes con las mismas condiciones aplicadas en este trabajo, pero con las anti-resonancias obtenidas en el trabajo anterior [2], comparados los con los resultados obtenidos en este trabajo.

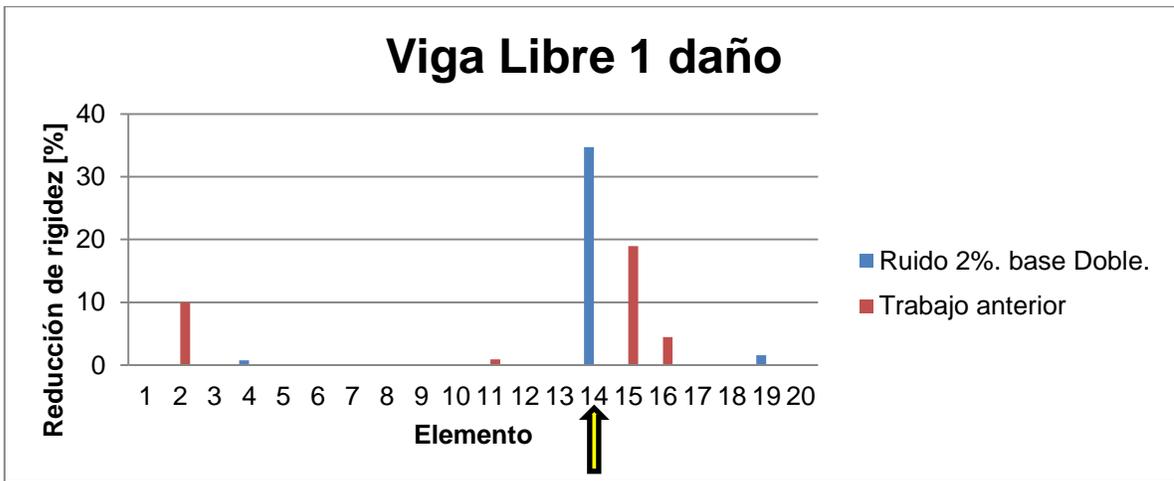


Figura 7-1: Comparación con el trabajo anterior para la viga con una falla libre.

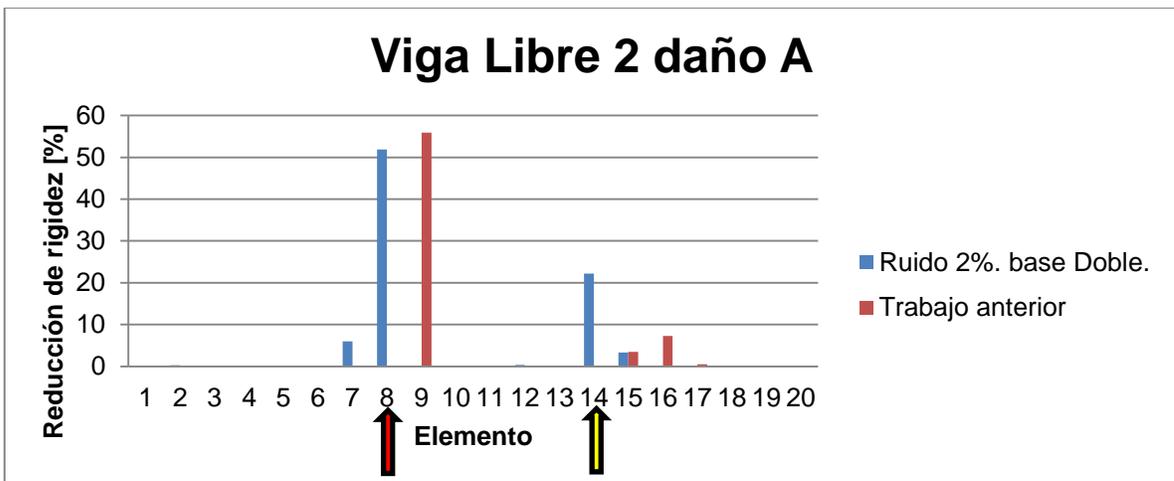


Figura 7-2: Comparación con el trabajo anterior para la viga con 2 fallas A libre.

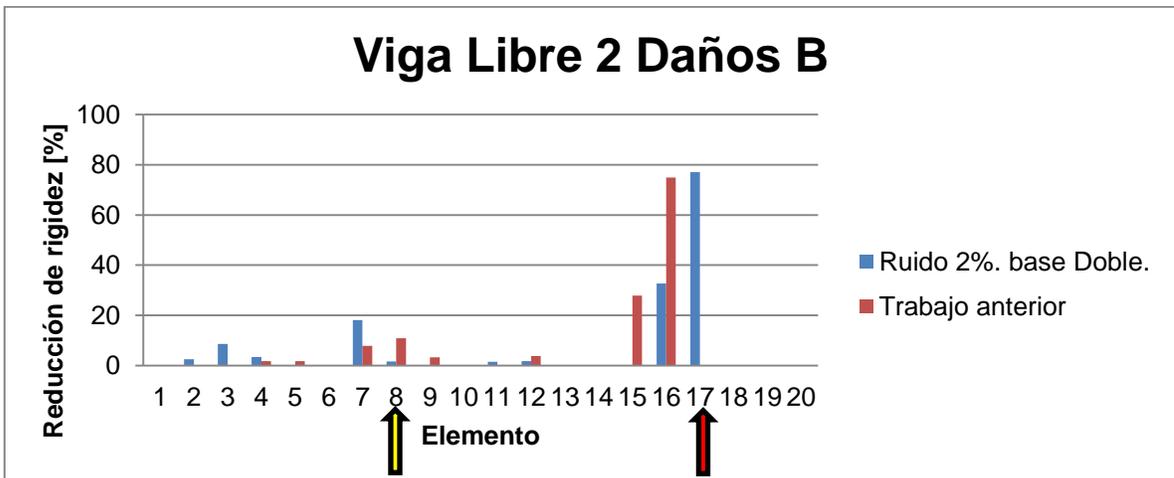


Figura 7-3: Comparación con el trabajo anterior para la viga con 2 fallas B libre.

Para el caso de una falla (Figura 7-1) se tiene que en el trabajo actual, los resultados son más certeros, se logra detectar el daño limpiamente, mientras que con el trabajo anterior, el daño se detecta en el elemento siguiente y se detecta un falso positivo grande en el elemento #2. Para el caso de dos fallas A (Figura 7-2) los resultados son claramente más favorables para el trabajo actual, se puede apreciar que la red logra detectar los daños de los elementos #8 y #14, con un pequeño falso positivo en el elemento #7 debido a la cercanía al siguiente elemento de la falla inducida, mientras que la red del trabajo anterior detecta el daño grande pero en un elemento adyacente al cual se encontraba el daño (elemento #8) y no logra detectar el daño del elemento #14. Para la viga con dos daños B (Figura 7-3) se tiene que la RNA entrenada con las anti-resonancias cruzadas, a pesar de tener falsos positivos, entrega información mucho más cercana a la realidad. La reducción de rigidez para el elemento #17 se condice con la magnitud de la falla inducida, y la reducción de rigidez para el elemento #8 es muy pequeña, pero se detecta una reducción de rigidez en el elemento #7, lo cual igual ayuda a dar indicios de la presencia de un daño, mientras que con el trabajo anterior, el daño en el elemento #17 no se detecta, pero lo hace en el elemento #16, con similar magnitud que en el caso actual, y el daño en el elemento #8 se detecta con más claridad que en el caso actual. Con estos resultados queda bastante claro lo planteado en los antecedentes, las anti-resonancias cruzadas son más sensibles que las directas, y a pesar de que se detectan falsos positivos, también se detectan fallas que con las anti-resonancias directas no se detectaban, lo que las hace más confiables

Para el caso de la viga empotrada con un solo daño, los resultados del entrenamiento, al igual que en el caso de la viga libre, son muy buenos, solo se detecta un falso positivo en el elemento #10, pero es de menor magnitud. Para el caso de la viga empotrada con 2 daños A, el daño grande del nodo #8 es detectado claramente, y al igual que en la viga libre, viene acompañado de un falso positivo en el elemento #7, por lo explicado anteriormente. La falla pequeña en el elemento #14 alcanza a ser detectada pero con menor magnitud, esto se atribuye a que se encuentra cerca del empotramiento, por lo que se pierde sensibilidad. Otro falso positivo se detecta en el nodo #18, éste se le atribuye a que es el primer nodo después del empotramiento, por lo que las anti-resonancias ahí se distorsionan. Para el caso de la viga empotrada con 2 fallas B, la falla grande del elemento #17 es detectada pero con una disminución de la matriz de rigidez muy baja, pero el elemento adyacente, el #16, presenta una disminución de la matriz de rigidez de un 68[%], esto también se le atribuye a que ésta falla se encuentra muy cercana al punto de empotramiento. La falla pequeña en el elemento #8 no se detecta directamente, ya que sí se detecta en el elemento #7, esto se le atribuye a la cercanía de la falla a éste elemento.

Cabe destacar que para el caso de la viga empotrada, el diseño del sistema de empotramiento no es perfecto, lo que daría pie a que los datos tomados no fuesen los

mejores, esto se puede ver en la Tabla 5-3, donde la última anti-resonancia registrada tiene un 28.38[%] de diferencia con la anti-resonancias analítica.

Con todo esto, es posible asegurar que las condiciones de borde sí afectan a los entrenamientos, haciendo más sensibles, o distorsionando, las anti-resonancias cruzadas detectadas en la barra empotrada. Esto se refleja en que la cantidad de falsos positivos encontrados en las RNA de las vigas empotradas es mayor que la cantidad de falsos positivos de las RNA de las vigas libres.

Con respecto al estudio de la introducción de ruido a los datos analíticos, se concluye que el método de introducción cumple con la función de acercar los datos analíticos a los experimentales, pero aún se pueden tener muchos avances en este tema. Analizando las tablas donde se entrenan las redes con las mismas características en 3 ocasiones, se ve que no se llega a una estabilidad en las reducciones de rigidez de los elementos. En la Tabla 4-14 se puede ver que en el elemento #17 la reducción de rigidez va de un 56.6[%] a un 80[%] lo cual ejemplifica la gran aleatoriedad de los resultados. El hacer entrenamientos con bases dobles o triples, no mejora sustancialmente la estabilidad de los resultados, por lo que se propone estudiar otros métodos de introducción de ruido para ver su comportamiento con respecto al método utilizado en el presente trabajo.

8 Conclusiones.

El algoritmo propuesto para la detección de fallas logra encontrar todas las fallas, pero aún quedan puntos que mejorar, como por ejemplo, la introducción aleatoria de ruido a los datos analíticos.

Se logró con éxito simular la barra con el programa Matlab mediante el uso de elementos finitos, pudiéndose obtener las anti-resonancias cruzadas que se deseaban.

Se logró entrenar con éxito la red neuronal con los datos simulados mediante elementos finitos.

Se llegó a que las condiciones de borde sí afectan a los entrenamientos de las redes, distorsionando los datos de la viga empotrada. Se propone emplear un sistema de empotramiento mejor al actual utilizado.

Se compararon los resultados obtenidos con las anti-resonancias directas y cruzadas y se concluyó que las cruzadas efectivamente son más sensibles, lo que conlleva que los entrenamientos mejoran su desempeño.

El estudio de introducción de ruido se realizó con éxito, concluyendo que aún falta por ensayar nuevos métodos de introducción para llegar a que los entrenamientos de las RNA sean más estables en su magnificación de los datos.

9 Trabajos futuros

Los trabajos futuros a realizar para aportar en esta línea de estudio son diversos, los principales son los siguientes:

- Disminución del ruido de los datos experimentales: El ruido en los datos experimentales es inevitable, pero si se puede mitigar, como por ejemplo, en el empotramiento de la viga del problema 2. En este caso el sistema utilizado no logro simular completamente las condiciones de empotramiento, por lo cual se propone desarrollar otro sistema para lograr disminuir el ruido en este caso.
- Métodos de adición de ruido a los datos analíticos: La función utilizada para agregar ruido a los datos analíticos logra acercarlos a los experimentales, pero, al ser aleatoria, no presenta una estabilidad que de la confianza suficiente, por lo que se propone encontrar una nueva función de introducción de ruido que presente más estabilidad o algún método que ayude a que los resultados sean más confiables.
- Aplicación a estructuras más complejas: La única estructura estudiada hasta el momento es una viga, con condiciones de borde libre y empotrada, se propone estudiar otra estructura para ampliar los alcances del método de detección de fallas.
- Elección de los nodos a medir: En el presente trabajo se midieron las anti-resonancias de nodos distribuidos uniformemente sobre la viga. Se propone utilizar un método que optimice la elección de los nodos para obtener la mejor información posible y con ello mejorar los resultados.

10 Bibliografía

[1] Carnicero A., "Introducción al método de los elementos finitos" Universidad Tecnológica Nacional, Facultad Regional de Cordova.

[2] J. Mahu. Detección de daños en una viga simple mediante antiresonancia y redes neurales. Memoria para optar al título de ingeniero civil mecánico, Universidad de Chile, 2012.

[3] 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.

[4] F. Wahl, G. Schmidt, and L. Forrai. On the significance of antiresonance frequencies in experimental structural analysis. *Journal of Sound and Vibration*, 219(3): 379-394, 1999.

[5] FAUSETT L. "Fundamentals of neural networks: architectures, algorithms and applications". Prentice Hall, 1993.

[6] K. D. Hjelmstad, "Damage Detection and Assessment of Structures from Static Response", *Journal Of Engineering Mechanics*,:570-571, June 1997.

[7] S. Alampalli, "Influence Of In-Service Environment On Modal Parameters" Transportation Research and Development Bureau, New York State Department Of Transportation, 111-116.

[8] M. Meo, G. Zumpano. On the optimal sensor placement techniques for a bridge structure. *International Journal of Engineering Structures* 27 (2005) 1488-1497.

[9] MERUANE V. "Apunte dinámica estructural, ME706" [En línea] <<http://viviana.meruane.com>> [consulta:Diciembre 2012]

[10] MERUANE V. "Apunte vibraciones mecánicas, ME4701" [En línea] <<http://viviana.meruane.com>> [consulta:Diciembre 2012]

[11] D'Ambrogio W. and Fregolent A., "The use of antiresonances for robust model updating", *Journal of Sound and Vibration*, vol. 236, no. 2, pp. 227-243, 2000.

[12] Meruane V., Mahu J."Real-time structural damage assessment using artificial neural networks and antiresonant frequencies", Artículo enviado a la revista "Structural Control and Health Monitoring", 2012.

[13] Meruane V. "Model updating using antiresonant identified from transmissibility functions". *Journal of Sound and Vibration*, vol. 332, no. 4, pp. 807-820, 2013.

[14] Meruane V., Mahu J.. "Identificación de daño estructural utilizando redes neuronales artificiales y frecuencias de anti-resonancia". XV Congreso Chileno de Ingeniería Mecánica, COCIM 2012, pp. 114- 120, 2012.

[15] K.D. Hjelmstad and S. Shin, "Damage Detection and Assessment of Structures from Static Response", *Journal of Engineering Mechanics*, vol. 123, no. 6, pp. 568-576, 1997.

11 Anexos

11.1 Anexo A

Generación del modelo y función para obtener las anti-resonancias cruzadas para la viga libre.

```
function Wa=VigaH(Beta)
%Propiedades de la viga
kmh=[0.0109 0.04614 -0.0002047 -0.058104];
rho=(1+kmh(1))*7800;
E=(1+kmh(2))*2.1e11;
b=(1+kmh(3))*0.01;
h=(1+kmh(4))*0.025;
A=b*h;
Iz=b*(h)^3/12;
le=0.05;%m Largo de un elemento
%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
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 n=1:20
K((1+2*(n-1)):(2*(n+1)),(1+2*(n-1)):(2*(n+1)))=K((1+2*(n-1)):(2*(n+1)),(1+2*(n-1)):(2*(n+1)))+k*Beta(n);
```

```

M((1+2*(n-1):(2*(n+1)),(1+2*(n-1):(2*(n+1))))=M((1+2*(n-1):(2*(n+1)),(1+2*(n-1):(2*(n+1))))+m;
end
Wa=zeros(41,7);
j=1:42;
aux1=setdiff(j,1);%Auxiliar que eliminará la columna del elemento donde se aplicará la fuerza.
for i=1:6:41 %7 sensores.
aux2=setdiff(j,i);%Auxiliar que eliminará la fila del elemento donde se medirá la anti-resonancia.
Ka=K(aux1,aux2);
Ma=M(aux1,aux2);
[~,wa]=eig(Ka,Ma);%Obtención de vectores propios de la matriz [MK^-1]
wa=sqrt(diag(wa))/2/pi;%Vector con las anti-resonancias de largo 41 por los grados de libertad.
    for m=1:41
        if real(wa(m,1))<1%Se filtran las anti-resonancias menores a 1[Hz], dandoles valor 0.
            wa(m,1)=0;
        end
    end
end
Wc(:,1+round(i/6))=sort(real(wa(:,1)));%Se ordenan de forma descendente las frecuencias de anti-resonancia.
end
Y=[6,11,15,18,20,22,23];
Wb=[0];
Wa=zeros(23,1);%Vector columna que tendrá las 23 anti-resonancias encontradas por los 7 sensores.
f=0;%Contador
for i=1:7
    for n=1:41
        if (Wc(n,i)>0&&length(Wb)<Y(i)) %Toma las Y(i) antiresonancias de la fila n columna i y las escribe en el vector columna Wb
            f=f+1;
            Wb(f,1)=Wc(n,i);
        end
    end
end

```

```

end
    if i==1          %Se ordenan las antiresonancias por grupo y se escriben en
el vector columna Wa
        z=sort(Wb(1:Y(i)), 'descend');
        Wa(1:Y(i))=z;
    else
        z=sort(Wb(Y(i-1)+1:Y(i)), 'descend');
        Wa(Y(i-1)+1:Y(i))=z;
    end
end
end

```

%Regresa un vector columna de dimensión [23X1] en el cual se encuentran las
 %anti-resonancias encontradas en la viga con el patron de daños introducido
 %en el vector Beta.

11.2 Anexo B

Obtención de las anti-resonancias cruzadas para la viga libre.

```
clear all;
clc;

%crear datos para entrenar 2 daños
n=2; %combinaciones de daños
dc=nchoosek(2:19,n); %combinaciones de 2 daños
dl=[0.2:0.1:1]; %niveles de daño
N=(length(dl)^n*length(dc))+1;
A=ones(N,20);
j=2;
for r=1:length(dc)
for i=1:length(dl)
    for k=1:length(dl)
        A(j,dc(r,1))=dl(i);
        A(j,dc(r,2))=dl(k);
        j=j+1;
    end
end
end
Wb=zeros(23,N);
f=0;
for i=1:N
    Wa=VigaH(A(i,:));
    f=f+1;
    for j=1:23
        Wb(j,f)=Wa(j,1);
    end
end
%Introducción del ruido
```

```

noise=1;
a=-noise/100;
b=noise/100;
Wb=( 1 + (a + (b-a)*rand(size(Wb)))).*Wb;
% Normalización de entradas
for i=2:N
Wb(:,i)=abs((Wb(:,1)-Wb(:,i))./Wb(:,1)));
end
%desordenar
Wb(:,1)=zeros(1,23);
Nta=randperm(length(A));
A=A(Nta,:);
Wb=Wb(:,Nta);
At=transpose(A);

```

11.3 Anexo C

Rutina de entrenamiento de la red neuronal.

```

dinput=Wb;
doutput=At(2:19,:);
net = newfit(dinput,doutput,140,{'logsig','satlins'},'trainlm');
net.divideParam.trainRatio=0.9;
net.divideParam.valRatio=0.1;
net.divideParam.testRatio=0.0;
net.trainParam.mem_reduc=15;
net.trainParam.epochs=10000; %Maximum number of epochs to train
[net,tr,Y,E,Pf,Af] = train(net,dinput,doutput);
save tr tr;
save net net;

```

11.4 Anexo D

Validación analítica de la red Neuronal.

```
clear all;
close all;

%crear datos para entrenar 2 daños
n=2; %combinaciones de daños
dc=nchoosek(2:19,n); %combinaciones de 2 daños
dl=0.25:0.1:1; %niveles de daño
N=length(dl)^n*length(dc)+1;
A=ones(N,20);
j=2;
for r=1:length(dc)
for i=1:length(dl)
    for k=1:length(dl)
        A(j,dc(r,1))=dl(i);
        A(j,dc(r,2))=dl(k);
        j=j+1;
    end
end
end
Wb=zeros(23,N);
f=0;
for i=1:N
    Wa=VigaH(A(i,:));
    f=f+1;
    for j=1:23
        Wb(j,f)=Wa(j,1);
    end
end
%Introducción del ruido
```

```

noise=2;
a=-noise/100;
b=noise/100;
Wb=( 1 + (a + (b-a)*rand(size(Wb)))).*Wb;
%Normalización de entradas
for i=2:N
Wb(:,i)=(Wb(:,1)-Wb(:,i))./Wb(:,1);
end
Wb(:,1)=zeros(1,23);
At=transpose(A);

dinput=Wb;
doutput=At(2:19,:);

load ERN140ruido02salto005novCONabsLibreBaseDoble3.mat; %Acá va la red que se
desea validar.

dsim=sim(net,dinput);

%%%%%%%%%%%%MSE%%%%%%%%%%%%
%%
MSE=sum(sum(abs(dsim-doutput)))/(size(dsim,1)*size(dsim,2)); %error cuantificación
Dr=doutput<1; %Sectores con daños
Dd=dsim<(1-MSE);%Daños identificados corregidos por el error de cuantificación
NT=sum(sum(Dr)); %numero total de daños reales
NF=sum(sum(Dd)); %numero total de daños identificados
DME=1/NT*(NT-sum(sum(Dr.*Dd))); %daños no identificados /sum(Dr.*Dd):Matriz con
1's en los sectores donde habían fallas pero no se identificaron
FAE=1/NF*sum(sum((1-Dr).*Dd)); %daños falsos
%%%%%%%%%%%%Por nivel de daño%%%%%%%%%%%%
l=0.9:-0.1:0.1;
for i=1:9
%nivel 1: 0-x% daño
Dr1=(doutput>l(i)).*(doutput<(l(i)+0.1));
Dd1=(dsim>=l(i)).*(dsim<(l(i)+0.1-MSE));

```

```

NT1=sum(sum(Dr1));
NF1=sum(sum(Dd1));
DME(i)=1/NT1*(NT1-sum(sum(Dr1.*Dd))); %que porcentaje de los daños entre 0 x% no
son detectados
FAE(i)=1/NF1*sum(sum((1-Dr).*Dd1)); %que porcentaje de los daños detectados entre
0 y x% son falsos
end
figure
set(gcf,'Position',[100 100 910 400])
set(gca,'FontSize',20)
set(gca,'Units','centimeters')
set(gca,'Position',[2.7 1.7 20.7 8.3])
set(gcf, 'PaperUnits', 'inches');
set(gcf, 'PaperSize', [9.3 4.2]);
set(gcf, 'PaperPosition', [-0.2 0 9.3 4.2]);
bar(DME*100)
% colormap(cmap2)
ylim([0 100])
set(gca, 'YTick',[0:20:100])
set(gca, 'XTick',[1:1:9])
set(gca, 'XTickLabel',{'<10','10-20','20-30','30-40','40-50','50-60','60-70','70-80','>80'})
xlabel('Nivel de daño (%)')
ylabel('Falsos negativos')
figure
set(gcf,'Position',[100 100 910 400])
set(gca,'FontSize',20)
set(gca,'Units','centimeters')
set(gca,'Position',[2.7 1.7 20.7 8.3])
set(gcf, 'PaperUnits', 'inches');
set(gcf, 'PaperSize', [9.3 4.2]);
set(gcf, 'PaperPosition', [-0.2 0 9.3 4.2]);
bar(FAE*100)
% load cmap2;
% colormap(cmap2)

```

```

ylim([0 100])
set(gca,'YTick',[0:20:100])
set(gca,'XTick',[1:1:9])
set(gca,'XTickLabel',{'<10','10-20','20-30','30-40','40-50','50-60','60-70','70-80','>80'})
xlabel('Nivel de daño (%)')
ylabel('Falsos positivos')

```

11.5 Anexo E

Generación del modelo y función para obtener las anti-resonancias cruzadas para la viga empotrada.

```

function Wa=VigaHE(Beta)
%Propiedades de la viga
kmh=[0.0109 0.04614 -0.0002047 -0.058104];
rho=(1+kmh(1))*7800;
E=(1+kmh(2))*2.1e11;
b=(1+kmh(3))*0.01;
h=(1+kmh(4))*0.025;
A=b*h;
Iz=b*(h)^3/12;
% rho=7800; %Kg/m3 Densidad
% E=2.1e11;%Pa Módulo de elasticidad
le=0.05;%m Largo de un elemento
% A=.025*.01;%m2 Área
% Iz=.010*(.025)^3/12;%m4 Inercia sección horizontal
%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
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 n=1:20
```

```
K((1+2*(n-1)):(2*(n+1)),(1+2*(n-1)):(2*(n+1)))=K((1+2*(n-1)):(2*(n+1)),(1+2*(n-1)):(2*(n+1)))+k*Beta(n);
```

```
M((1+2*(n-1)):(2*(n+1)),(1+2*(n-1)):(2*(n+1)))=M((1+2*(n-1)):(2*(n+1)),(1+2*(n-1)):(2*(n+1)))+m;
```

```
end
```

```
% Wa=zeros(42,6);
```

```
% ue=[3.55121141112182e-05,6.42580549825817e-10,9.71955122699819e-07,10795.6567562826];
```

```
% emp=39:42;
```

```
% Ke=[ue(1) 0 0 0;0 ue(2) 0 0;0 0 ue(3) 0; 0 0 0 ue(4)];
```

```
% K(emp,emp)=K(emp,emp)+Ke;
```

```
j=1:42;
```

```
aux1=setdiff(j,[39:42]);
```

```
K=K(aux1,aux1);
```

```
M=M(aux1,aux1);
```

aux1=setdiff(j,1);%Auxiliar que eliminará la columna del elemento donde se aplicará la fuerza y las que están restringidas por CB.

% Algoritmo que calcula las anti-resonancias cruzadas eliminando una columna

% el el elemento donde se aplicará la fuerza y una fila en el elemento donde

% se desea medir la anti-resonancia.

```
j=1:38;
```

```
aux1=setdiff(j,1);
```

```
for i=1:6:31 %6 sensores.
```

aux2=setdiff(j,i);%Auxiliar que eliminará la fila del elemento donde se medirá la anti-resonancia.

```
Ka=K(aux1,aux2);
```

```
Ma=M(aux1,aux2);
```

```

[~,wa]=eig(Ka,Ma);%Obtención de vectores propios de la matriz [MK^-1]
wa=sqrt(diag(wa))/2/pi;%Vector con las anti-resonancias de largo 42 por los grados de
libertad.
    for m=1:37
        if real(wa(m,1))<1%Se filtran las anti-resonancias menores a 1[Hz], dandoles valor
0.
            wa(m,1)=0;
        end
    end
Wc(:,1+round(i/6))=sort(real(wa(:,1)));%Se ordenan de foma ascendiente las
frecuencias de anti-resonancia.
end
Y=[7,12,16,19,21,22];
Wb=[0];
Wa=zeros(22,1);%Vector columna que tendrá las 22 anti-resonancias encontradas por
los 6 sensores.
f=0;%Contador
for i=1:6
    for n=1:37
        if (Wc(n,i)>0&&length(Wb)<Y(i)) %Toma las Y(i) antiresonancias de la fila n columna i
y las escribe en el vector columna Wb
            f=f+1;
            Wb(f,1)=Wc(n,i);
        end
    end
    if i==1                %Se ordenan las antiresonancias por grupo y se escriben en
el vector columna Wa
        z=sort(Wb(1:Y(i)),'descend');
        Wa(1:Y(i))=z;
    else
        z=sort(Wb(Y(i-1)+1:Y(i)),'descend');
        Wa(Y(i-1)+1:Y(i))=z;
    end
end
end
%Regresa un vector columna de dimensión [22X1] en el cual se encuentran las

```

%anti-resonancias encontradas en la viga con el patron de daños introducido
%en el vector Beta.

11.6 Anexo F

Obtención de las anti-resonancias cruzadas para la viga empotrada.

```
clear all;
clc;
%crear datos para entrenar 2 daños
n=2; %combinaciones de daños
dc=nchoosek(2:19,n); %combinaciones de 2 daños
dl=[0.2:0.1:1]; %niveles de daño
N=length(dl)^n*length(dc)+1;
A=ones(N,20);
j=2;
for r=1:length(dc)
for i=1:length(dl)
    for k=1:length(dl)
        A(j,dc(r,1))=dl(i);
        A(j,dc(r,2))=dl(k);
        j=j+1;
    end
end
end
Wb=zeros(22,N);
f=0;
for i=1:N
    Wa=VigaHE(A(i,:));
    f=f+1;
    for j=1:22
        Wb(j,f)=Wa(j,1);
    end
end
```

```

end
%Introducción del ruido
noise=1;
a=-noise/100;
b=noise/100;
Wb=( 1 + (a + (b-a)*rand(size(Wb)))).*Wb;
% Wb=awgn(Wb,50);
% Normalización de entradas
for i=2:N
Wb(:,i)=abs((Wb(:,1)-Wb(:,i))./Wb(:,1)));
end
%desordenar
Wb(:,1)=zeros(1,22);
Nta=randperm(length(A));
A=A(Nta,:);
Wb=Wb(:,Nta);
At=transpose(A);

```