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

IDENTIFICACIÓN DE DAÑO EN ESTRUCTURAS DE BARRAS UTILIZANDO MÉTODOS DE SUB-ESTRUCTURACIÓN Y REDES NEURONALES

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERA CIVIL MECÁNICA

FABIOLA MACARENA ARACENA MONTALBÁN

PROFESORA GUÍA:

VIVIANA MERUANE NARANJO

MIEMBROS DE LA COMISION:

JUAN CRISTOBAL ZAGAL MONTEALEGRE
RUBÉN BOROSCHEK KRAUSKOPF

SANTIAGO DE CHILE
MAYO 2013

RESUMEN EJECUTIVO

Toda estructura en ingeniería se encuentra expuesta al daño y deterioro durante su vida útil. La información oportuna del deterioro que presenta puede incrementar la seguridad en su uso y mejorar su confiabilidad, además de reducir significativamente los costos asociados al mantenimiento.

El uso de Redes Neuronales Artificiales (ANN en inglés artificial neural network) ha sido considerado en la detección de daño porque luego que la ANN es entrenada, utilizarla implica solo un gasto menor de recursos computacionales, lo que la hace idónea para monitoreo en tiempo real. Sin embargo, en la etapa de entrenamiento es necesaria gran capacidad computacional, la cual crece con la complejidad de la estructura. Además, mientras más elementos se quieran detectar con una red, más difícil se vuelve el entrenamiento y peores son los resultados. Esto último hace que la aplicación directa de redes neuronales en estructuras complejas sea casi imposible. Una solución es dividir la estructura en sub-estructuras y entrenar una red para cada sub-estructura por separado, este método se conoce como sub-estructuración.

En el presente trabajo se obtiene un algoritmo capaz de identificar daño en estructuras de barras por medio del método de subestructuras combinado con ANN. Las frecuencias de resonancia y anti-resonancia de la estructura son las variables de entrada sensibles al daño y las variables de salida son factores de reducción de rigidez para cada elemento.

El trabajo se divide en dos etapas principales; primero se modela en MATLAB la estructura mediante elementos finitos y se realiza un análisis con daño simulado, y luego se valida el modelo con una estructura experimental. El método de identificación de daño utiliza dos redes neuronales. La primera debe detectar la o las sub-estructura(s) con daño y la segunda debe detectar daños en cada elemento de la sub-estructura identificada por la primera red. De esta forma se reduce el tamaño de cada ANN, y con esto los recursos computacionales necesarios para entrenarlas. El daño en un elemento tiene dos representaciones; a nivel de elementos finitos se considera un factor de reducción de rigidez y a nivel experimental se considerarán cortes en las secciones.

El desarrollo de la primera red entrega buenos resultados a nivel numérico y logra detectar las subestructuras dañadas en tres de los cuatro casos de daño experimental. Para su entrenamiento se incluyeron las frecuencias de resonancia y las frecuencias de anti-resonancia de los nodos límite de cada subestructura. Las redes de la segunda etapa, una para cada subestructura, se logran entrenar a nivel numérico dependiendo principalmente del número de entradas disponibles, lo cual permite que tres de las seis presenten resultados muy buenos. La validación con datos experimentales detecta los daños 5 existentes en los casos estudiados, pero sólo logra cuantificar y localizar uno de ellos.

Por lo tanto, el uso de un método de sub-estructuración y redes neuronales para la identificación de daño en estructuras de barras, resultó correcto en la etapa de localización de subestructuras dañadas. Sin embargo, para la identificación de elementos dañados en cada sub-estructura es necesario asegurar una cantidad mínima de datos de entrada a la red (frecuencias de anti-resonancia) que permita el correcto entrenamiento de la red neuronal.

AGRADECIMIENTOS

Quiero agradecer a quienes creyeron en mí, me apoyaron y quisieron en este camino que resultó más intrincado de lo esperado. Especialmente a mi familia y mis amigos, por su apoyo incondicional, y a mi profesora guía por su dedicación en este trabajo y por ayudarme a volver a encontrar el camino.

"Mis melodías y números están aquí. Han llenado mis años, los años en que rehusé morirme. Y para eso mismo escribo, escribo, escribo, al mediodía o a las tres de la mañana.
Para no estar muerto." Ray Bradbury, 1997.

TABLA DE CONTENIDO

1 Introducción

Como parte de la vida útil de toda estructura en ingeniería, ésta se encuentra expuesta al daño y deterioro. La información oportuna del deterioro que presenta puede incrementar la seguridad en su uso y mejorar su confiabilidad, además de reducir significativamente los costos asociados al mantenimiento.

En este trabajo se busca desarrollar un algoritmo capaz de detectar y cuantificar daño en una estructura de barras, utilizando para ello la técnica de subestructuras por medio de redes neuronales artificiales (ANN en inglés artificial neural network). Las frecuencias anti-resonantes de la estructura serán las variables de entrada sensibles al daño y las variables de salida factores de reducción de rigidez para cada elemento.

El presente trabajo se basa en el desarrollado por Mahu [1], el cual presenta un método donde se entrena y prueba una red neuronal artificial de tipo Multilayer Perceptron (MLP) con 3 capas (entrada, oculta y salida) para la detección de daño en dos casos experimentales: un sistema masa resorte de 8 grados de libertad y una viga con múltiples escenarios de daño. Para esto se utilizaron como entradas las frecuencias de anti-resonancia y como salidas el nivel de daño en cada elemento, basados en la variación de la rigidez efectiva del elemento (Modulo de elasticidad de cada elemento). Como resultado del trabajo se obtuvieron los parámetros de la red y, mediante indicadores específicos, la efectividad de la red.

El uso de ANN para la detección de daño en grandes estructuras se limita debido al gran esfuerzo computacional necesario para entrenar la red en estos casos. Adicionalmente, mientras más elementos se quieran detectar con una red, más difícil se vuelve la etapa de entrenamiento y peores son los resultados de la red. Esto último hace que la aplicación directa de redes neuronales en estructuras complejas sea casi imposible. Una solución es dividir la estructura en sub-estructuras y entrenar una red para cada sub-estructura por separado, este método se conoce como sub-estructuración [3,4].

Sahoo y Maity [2] utilizan un algoritmo híbrido neuro-genético para automatizar el diseño de la red neuronal con el fin de generalizar el método para distintos tipos de estructuras. La red que utilizan tiene dos capas ocultas y usan un algoritmo retro-propagación con coeficientes de momentum para el entrenamiento. El algoritmo genético selecciona los valores adecuados para los parámetros de red como número de neuronas en cada capa oculta, el aprendizaje y los coeficientes de momentum. Para el análisis de la estructura, se utiliza el método de sub-estructuración propuesto por Yun y Bahng[3], donde la estructura se divide en varias sub-estructuras. La ventaja del método es reducir el problema y que se utilizan sólo los datos de medición de la sub estructura de interés. Los parámetros elegidos para representar a la estructura son una combinación de parámetros vibracionales (frecuencias naturales, modos de vibración, curvaturas de los modos, etc.), parámetros estáticos (desplazamientos, deformaciones, etc.) y parámetros energéticos (energía de deformación, etc.). Un elemento dañado es definido con una reducción en su rigidez. En su trabajo concluyen que las frecuencias naturales por si solas no

entregan toda la información necesaria para la identificación de daño, sin embargo, éstas pueden ser complementadas con deformaciones.

Bakhary y otros [4] usan la técnica de sub-estructuras junto a una ANN de dos etapas para detectar la localización y la extensión del daño, primero se divide la estructura en pocas estructuras y las subestructuras que presenten variaciones desde las condiciones base serán analizadas sub-dividiéndolas y repitiendo el proceso hasta identificar claramente la localización y extensión del daño. Parámetros modales como frecuencias de resonancia y modos de vibración son usados como entrada de la ANN. El daño se introduce reduciendo la rigidez local del elemento. Los autores concluyen que el método desarrollado es efectivo reduciendo el tamaño de los modelos de las ANN y el esfuerzo computacional, además pueden identificar el daño de manera más precisa que una ANN que analiza una estructura completa.

En varios trabajos [2,4,5,6] se utiliza el método de sub-estructura, a veces combinado con ANN y con otros métodos numéricos en base a modelos y parámetros modales, donde las estructuras estudiadas son principalmente vigas o barras simples, estructuras entramadas y estructuras de distinta cantidad de pisos.

En este trabajo se desarrolla un algoritmo para la detección de daños en una estructura de barras, similar a las mostradas en Figura 1.1. El algoritmo considera 2 etapas, en la primera se divide la estructura en sub-estructuras y se determina en cual existe daño, luego en la segunda etapa se identificarán el o los daños en los elementos de las subestructuras seleccionadas en la primera etapa. En ambas etapas se utilizan ANN para la resolución. En la primera etapa se utilizan las frecuencias de resonancia como variables de entrada y como salida un vector que indique el nivel de daño de cada subestructura. Para la segunda parte se utilizarán las frecuencias de anti-resonancia como entradas y los factores de reducción de rigidez como salidas de cada elemento utilizando una red neuronal para cada subestructura. El algoritmo de identificación de daño propuesto es validado primero con datos obtenidos de un modelo en elementos finitos, y luego con datos experimentales.

La distribución del presente trabajo es la que se describe a continuación. En la sección 1 se presenta la introducción en conjunto con los objetivos planteados. La sección 2 está constituida por los antecedentes principales para entender el desarrollo del trabajo, siendo los puntos más importantes el 2.1 que contiene la definición de respuesta en frecuencia y el punto 2.6 que muestra antecedentes sobre redes neuronales. La sección 3 contiene la metodología utilizada para realizar las diferentes etapas del trabajo. La sección 4 contiene el análisis de la estructura incluyendo su representación en elementos finitos y experimental. La sección 5 contiene los resultados; primero en la sección 5.1 el entrenamiento de la primera red neuronal con datos numéricos, luego en la sección 5.2 el entrenamiento de las segundas redes neuronales con datos numéricos y finalmente en la sección 5.3 se muestran los resultados que se obtienen utilizando los datos experimentales en las redes neuronales entrenadas. La sección 6 presenta las discusiones basadas en las diferentes etapas del trabajo y la sección 7 las conclusiones que se desprenden de la sección anterior.

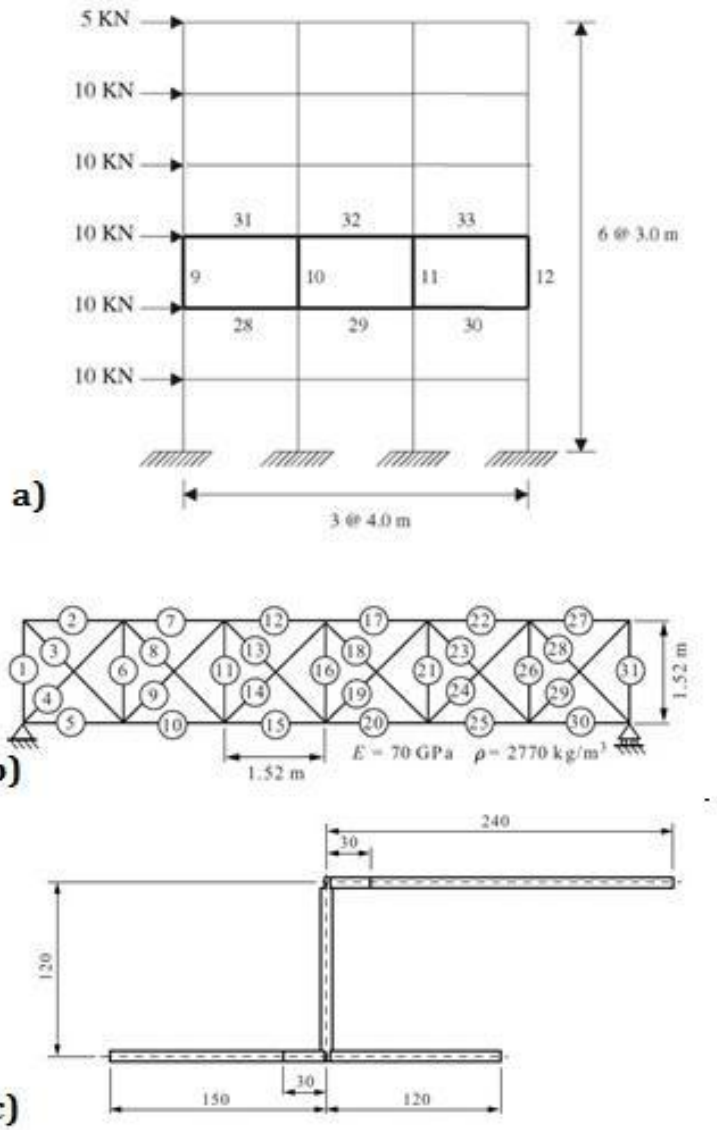


Figura 1.1 a) Estructura en pisos [2], b) Estructura entramada [7] y c) Estructura experimental utilizada en [7]

1.1 Objetivos

A continuación se detallan los objetivos de esta memoria

1.1.1 Objetivo general

Desarrollar un método para la identificación de daño en una estructura de barras, utilizando el método de sub-estructuración y redes neuronales, donde las variables de entrada son las frecuencias de resonancia y anti-resonancia, y las salidas son factores de reducción de rigidez.

1.1.2 Objetivos Específicos

- Desarrollar un modelo numérico en elementos finitos de una estructura de barras, para obtener las frecuencias de anti-resonancia y resonancia.
- Diseñar y entrenar dos tipos de redes neuronales usando los datos numéricos. La primera que identifique las subestructuras que presentan daño y la segunda que localice y cuantifique el daño en los elementos de la(s) subestructura(s) antes identificadas.
- Diseñar un montaje experimental que permita medir valores experimentales de las frecuencias de resonancia y anti-resonancia de la estructura.
- Evaluar el desempeño de las redes por medio de indicadores adecuados, usando datos numéricos y experimentales.

2 Antecedentes

2.1 Funciones de respuesta en frecuencia y frecuencias de anti-resonancia

A continuación se muestra como se relaciona la función de transferencia de un sistema con múltiples grados de libertad con los parámetros modales (frecuencias de resonancia y vectores modales).

En el caso de un sistema con múltiples grados de libertad la ecuación de movimiento equivale a la ecuación de un grado de libertad, pero matricial. Para ilustrar esto se desarrolla el caso de un sistema de dos grados de libertad mostrado en la Figura 2.1.

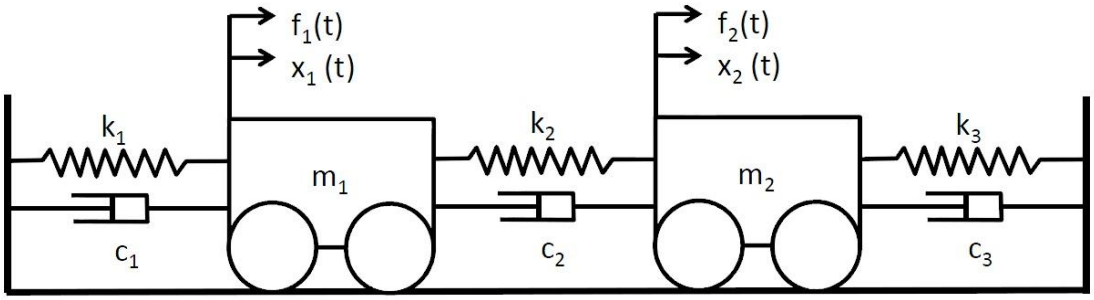


Figura 2.1 Ejemplo de un sistema de dos grados de libertad: Ilustración original en [9]

Las ecuaciones de movimiento para este sistema son:

$$m_1\ddot{x}_1 + (c_1 + c_2)\dot{x}_1 - c_2\dot{x}_2 + (k_1 + k_2)x_1 - k_2x_2 = f_1 \quad \textbf{(2.1)}$$

$$m_2\ddot{x}_2 + (c_2 + c_3)\dot{x}_2 - c_2\dot{x}_1 + (k_2 + k_3)x_2 - k_2x_1 = f_2 \quad \textbf{(2.2)}$$

En notación matricial:

$$\begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix}\begin{Bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{Bmatrix} + \begin{bmatrix} c_1 + c_2 & -c_2 \\ -c_2 & c_2 + c_3 \end{bmatrix}\begin{Bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{Bmatrix} + \begin{bmatrix} k_1 + k_2 & -k_2 \\ -k_2 & k_2 + k_3 \end{bmatrix}\begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} = \begin{Bmatrix} f_1 \\ f_2 \end{Bmatrix} \quad \textbf{(2.3)}$$

Que se puede escribir como:

$$M\{\ddot{x}\} + C\{\dot{x}\} + K\{x\} = \{f\} \quad \textbf{(2.4)}$$

Donde,

- M:matriz de masa
- C:matriz de amortiguación
- K:matriz de rigidez
- $\{x\}$:vector de respuesta
- $\{f\}$:vector de fuerzas

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

$$[FRF(\omega)] = \frac{\{X(\omega)\}}{\{F(\omega)\}} = \left(K + j\omega C - \omega^2 M \right)^{-1} = \frac{adj(K + j\omega C - \omega^2 M)}{det(K + j\omega C - \omega^2 M)} \qquad \textbf{(2.5)}$$

Donde adj(.) y det(.) definen a los operadores matriciales adjunto y determinante.

De la ecuación (2.5) se puede obtener la amplitud de la respuesta normalizada según la fuerza en función de la frecuencia ω. El resultado es una matriz de NxN (N = número de grados de libertad del sistema), donde $FRF(\omega)_{ij}$ corresponde a la función de respuesta medida en i al excitar la estructura en j, cabe notar que la matriz es simétrica. Las funciones de respuesta en frecuencia obtenidas cuando i y j son iguales, es decir, cuando el punto de medición de la respuesta es el mismo que el punto de excitación, se conocen como FRF directas (Figura 2.2), mientras que si estos puntos difieren se les denomina FRF indirectas (Figura 2.3). En ambas figuras aparecen las frecuencias resonantes, que son los puntos en que la amplitud de la respuesta vibratoria se maximiza y tiende a infinito para el caso sin amortiguamiento, se puede observar que estas frecuencias se mantiene fijas en ambos tipos de FRF. Por otro lado, se pueden definir las frecuencias de anti-resonancia como los puntos en que la respuesta vibratoria tiende a cero y éstas, a diferencia del caso anterior cambian su cantidad y valor en las distintas FRF según la posición y distancia que separan al punto de medición del punto de excitación de la estructura.

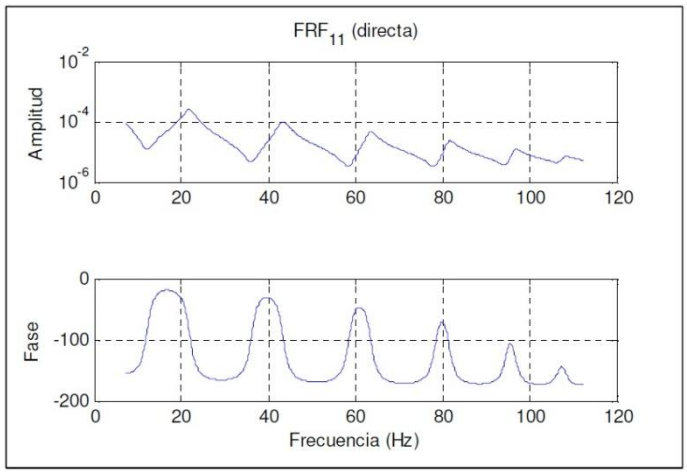


Figura 2.2 Gráfico función de respuesta en frecuencia directa (8 grados de libertad). Ilustración original en [1]

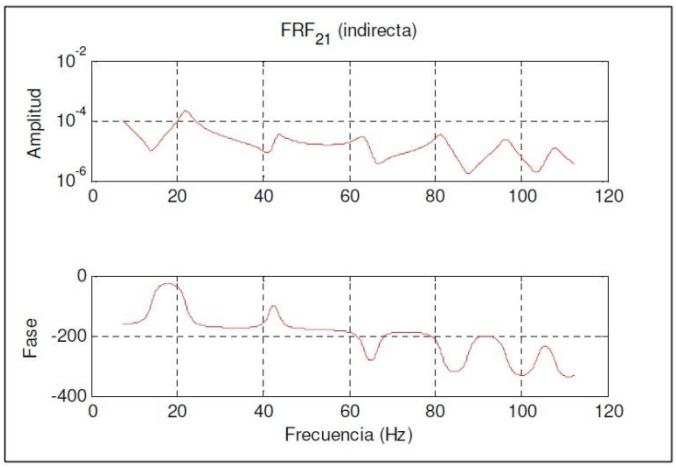


Figura 2.3 Gráfico función de respuesta en frecuencia indirecta (8 grados de libertad). Ilustración original en [1]

Analíticamente, las frecuencias de resonancia hacen cero el denominador de la ecuación (2.5) lo que maximiza su valor. Se determinan numéricamente imponiendo que el determinante de la matriz de rigidez dinámica sea igual a cero. Las frecuencias de anti-resonancia se pueden determinar resolviendo que el operador adj(.) de la matriz de rigidez dinámica sea cero, el operador adj(.) representa la matriz adjunta.

2.2 Determinación experimental de las frecuencias de anti-resonancia

Experimentalmente, se puede obtener la FRF de una estructura mediante el uso de un sistema como el presentado en la Figura 2.4. Este sistema consiste de un martillo el cual produce una fuerza de excitación sobre la estructura en forma de pulso y que contiene un sensor de fuerzas. Por otra parte, el acelerómetro es un sensor que mide la aceleración de un punto de la estructura. Ambas señales son acondicionadas y transmitidas a un analizador. Es decir, los datos de entrada que se obtienen de la estructura son la fuerza de entrada en función del tiempo $f(t)$ y la aceleración de un punto de la estructura en función del tiempo $\ddot{x}(t)$. El desplazamiento se puede obtener como:

$$\begin{aligned} x(t) &= A\,sin(\omega t) \\ \ddot{x}(t) &= -\omega^2 A\,sin(\omega t) \\ \ddot{x}(t) &= -\omega^2 x(t) \end{aligned} \qquad \textbf{(2.6)}$$

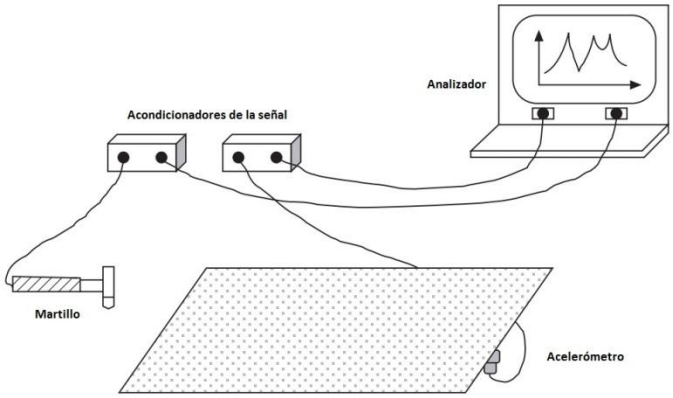


Figura 2.4 Montaje experimental para obtención de FRF usando martillo. Ilustración original en [9]

La función de respuesta en frecuencia $H(\omega)$ se puede determinar mediante los espectros en frecuencia de las señales de entrada y salida, respectivamente $F(\omega)$ para $f(t)$ la señal de entrada y $X(\omega)$ para $x(t)$ la señal de salida.

$$H(\omega) = \frac{X(\omega)}{F(\omega)} \tag{2.7}$$

Para eliminar el ruido no correlacionado, en la práctica, se utilizan las siguientes expresiones:

$$H_1(\omega) = \frac{X(\omega)}{F(\omega)}\frac{F^*(\omega)}{F^*(\omega)} = \frac{G_{XF}}{G_{FF}} \tag{2.8}$$

$$H_2(\omega) = \frac{X(\omega)}{F(\omega)}\frac{X^*(\omega)}{X^*(\omega)} = \frac{G_{XX}}{G_{FX}} \tag{2.9}$$

Finalmente se define una función de correlación que se denomina función de coherencia y es una medida del error de mínimos cuadrados.

$$\gamma^2 = \frac{\|G_{FX}\|^2}{G_{FF}G_{XX}} = \frac{H_1(\omega)}{H_2(\omega)} \tag{2.10}$$

La coherencia varía entre 0 y 1, donde 1 indica una relación perfectamente lineal de las señales de entrada y salida.

En la Figura 2.2 y Figura 2.3 se muestran ejemplos de FRF, donde buscando los mínimos y máximos locales de la función se encuentran las frecuencias de resonancia y de anti-resonancia. Además, estos máximos y mínimos están acompañados por un cambio abrupto en la fase de la FRF, en dirección 180° para las frecuencias de anti-resonancia y de -180° para las frecuencias de resonancia.

2.3 Modelo en elementos finitos para una viga

El modelo más utilizado para análisis de vibraciones es donde se asumen los desplazamientos y se calculan las fuerzas y es el que se usará en este trabajo.
El primer paso para construir un modelo en elementos finitos es discretizar la estructura en un número de elementos. Se pueden definir tres familias de elementos.

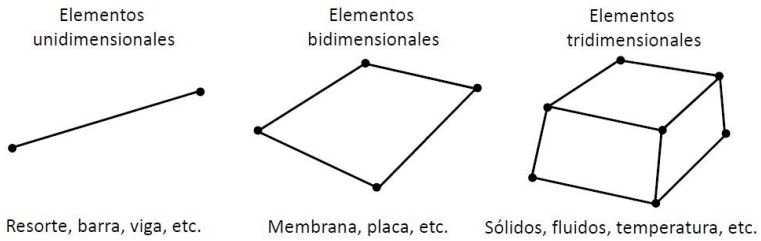


Figura 2.5 Familias de elementos finitos. Ilustración original en [9].

2.3.1 Elemento de viga en 3D

Debido a que el tipo de estructura a analizar se considerará compuesta por vigas (ver sección 4.1) es necesario agregar nociones de elemento de viga en 3D y su rotación utilizando coordenadas globales.

Un elemento de viga en 3D considera 6 grados de libertad por nodo como se muestra en la Figura 2.6.

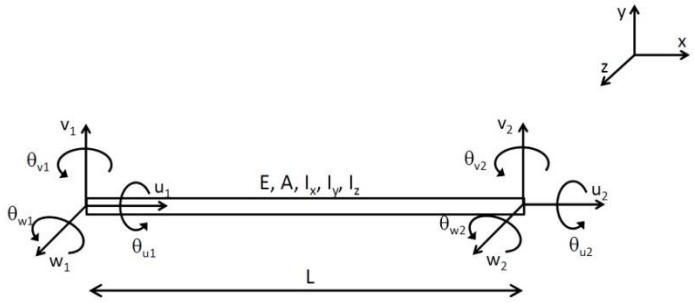


Figura 2.6 Elemento de viga en 3D. Ilustración original en [9]

Las matrices de rigidez y masa del elemento de viga en 3D en el sistema de coordenadas local vienen dadas por:

$$K_l = \begin{matrix} u_1 \\ \\ \\ \\ \\ \\ u_2 \\ \\ \\ \\ \\ \\ \end{matrix} \begin{matrix} v_1 \\ \\ \\ \\ \\ \\ v_2 \\ \\ \\ \\ \\ \\ \end{matrix} \begin{matrix} w_1 \\ \\ \\ \\ \\ \\ w_2 \\ \\ \\ \\ \\ \\ \end{matrix} \begin{matrix} \theta_{u1} \\ \\ \\ \\ \\ \\ \theta_{u2} \\ \\ \\ \\ \\ \\ \end{matrix} \begin{matrix} \theta_{v1} \\ \\ \\ \\ \\ \\ \theta_{v2} \\ \\ \\ \\ \\ \\ \end{matrix} \begin{matrix} \theta_{w1} \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{matrix} \begin{bmatrix} \frac{EA}{L} & 0 & 0 & 0 & 0 & 0 & \frac{-EA}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{12EI_z}{L^3} & 0 & 0 & 0 & \frac{6EI_z}{L^2} & 0 & \frac{-12EI_z}{L^3} & 0 & 0 & 0 & \frac{6EI_z}{L^2} \\ 0 & 0 & \frac{12EI_y}{L^3} & 0 & \frac{-6EI_y}{L^2} & 0 & 0 & 0 & \frac{-12EI_y}{L^3} & 0 & \frac{-6EI_y}{L^2} & 0 \\ 0 & 0 & 0 & \frac{GI_x}{L} & 0 & 0 & 0 & 0 & 0 & \frac{-GI_x}{L} & 0 & 0 \\ 0 & 0 & \frac{-6EI_y}{L^2} & 0 & \frac{4EI_y}{L} & 0 & 0 & 0 & \frac{6EI_y}{L^2} & 0 & \frac{2EI_y}{L} & 0 \\ 0 & \frac{6EL_z}{L^2} & 0 & 0 & 0 & \frac{4EI_z}{L} & 0 & \frac{-6EI_z}{L^2} & 0 & 0 & 0 & \frac{2EI_z}{L} \\ \frac{-EA}{L} & 0 & 0 & 0 & 0 & 0 & \frac{EA}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{-12EI_z}{L^3} & 0 & 0 & 0 & \frac{-6EI_z}{L^2} & 0 & \frac{12EI_z}{L^3} & 0 & 0 & 0 & \frac{-6EI_z}{L^2} \\ 0 & 0 & \frac{-12EI_y}{L^3} & 0 & \frac{6EI_y}{L^2} & 0 & 0 & 0 & \frac{12EI_y}{L^3} & 0 & \frac{6EI_y}{L^2} & 0 \\ 0 & 0 & 0 & \frac{-GI_x}{L} & 0 & 0 & 0 & 0 & 0 & \frac{GI_x}{L} & 0 & 0 \\ 0 & 0 & \frac{-6EI_y}{L^2} & 0 & \frac{2EI_y}{L} & 0 & 0 & 0 & \frac{6EI_y}{L^2} & 0 & \frac{4EI_y}{L} & 0 \\ 0 & \frac{6EI_z}{L^2} & 0 & 0 & 0 & \frac{2EI_z}{L} & 0 & \frac{-6EI_z}{L^2} & 0 & 0 & 0 & \frac{4EI_z}{L} \end{bmatrix} \quad \textbf{(2.11)}$$

$$M_l = \rho AL \begin{bmatrix} \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & \frac{1}{6} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{13}{35} & 0 & 0 & 0 & \frac{11L}{210} & 0 & \frac{9}{70} & 0 & 0 & 0 & \frac{-13L}{210} \\ 0 & 0 & \frac{13}{35} & 0 & \frac{-11L}{210} & 0 & 0 & 0 & \frac{9}{70} & 0 & \frac{13L}{210} & 0 \\ 0 & 0 & 0 & \frac{r^2}{3} & 0 & 0 & 0 & 0 & 0 & \frac{r^2}{6} & 0 & 0 \\ 0 & 0 & \frac{-11L}{210} & 0 & \frac{L^2}{105} & 0 & 0 & 0 & \frac{-13}{420} & 0 & \frac{-L^2}{140} & 0 \\ 0 & \frac{11L}{210} & 0 & 0 & 0 & \frac{L^2}{105} & 0 & \frac{13L}{420} & 0 & 0 & 0 & \frac{-L^2}{140} \\ \frac{1}{6} & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{9}{70} & 0 & 0 & 0 & \frac{13L}{420} & 0 & \frac{13}{35} & 0 & 0 & 0 & \frac{-11L}{210} \\ 0 & 0 & \frac{9}{70} & 0 & \frac{-13}{420} & 0 & 0 & 0 & \frac{13}{35} & 0 & \frac{11L}{210} & 0 \\ 0 & 0 & 0 & \frac{r^2}{6} & 0 & 0 & 0 & 0 & 0 & \frac{r^2}{3} & 0 & 0 \\ 0 & 0 & \frac{13L}{210} & 0 & \frac{-L^2}{140} & 0 & 0 & 0 & \frac{11L}{210} & 0 & \frac{L^2}{105} & 0 \\ 0 & \frac{-13L}{210} & 0 & 0 & 0 & \frac{-L^2}{140} & 0 & \frac{-11L}{210} & 0 & 0 & 0 & \frac{L^2}{105} \end{bmatrix} \quad \textbf{(2.12)}$$

Donde,

$$r = \frac{I_x}{A}$$

2.3.2 Rotación de elemento de viga

El operador de rotación se construye de la siguiente manera. Sean,

$$p_1 = \begin{Bmatrix} x_1 \\ y_1 \\ z_1 \end{Bmatrix}$$

$$p_2 = \begin{Bmatrix} x_2 \\ y_2 \\ z_2 \end{Bmatrix}$$

Las posiciones de los nodos del elemento en el sistema global de coordenadas.

La dirección del eje neutral del elemento, $\vec{e}_x$ se define por:

$$\vec{e}_x = \frac{p_2 - p_1}{\|p_2 - p_1\|} = \frac{p_2 - p_1}{L} \qquad \textbf{(2.13)}$$

Luego, es necesario definir el punto p_3, el cual junto a p_1 y p_2 definen el plano O_{xz} del elemento, y permite definir $\vec{e}_y$ y $\vec{e}_z$ como sigue,

$$\vec{e}_y = \frac{(p_3 - p_1) \times (p_2 - p_1)}{\|(p_3 - p_1) \times (p_2 - p_1)\|} \qquad \textbf{(2.14)}$$

$$\vec{e}_z = \vec{e}_x \times \vec{e}_y \qquad \textbf{(2.15)}$$

$\times$ es el producto cruz entre dos vectores. Por lo que se puede definir el operador rotación como:

$$R = \left[\vec{e}_x \vec{e}_y \vec{e}_z \right]^T$$

Definiendo x_g la posición de los ejes globales y x_l es la posición de los ejes locales, se tiene,

$$x_g = R x_l \qquad \textbf{(2.16)}$$

Y además se tiene:

$$\begin{Bmatrix} x \\ y \\ z \end{Bmatrix} = R \begin{Bmatrix} u \\ v \\ w \end{Bmatrix}$$

$$\begin{Bmatrix} \theta_x \\ \theta_y \\ \theta_z \end{Bmatrix} = R \begin{Bmatrix} \theta_u \\ \theta_v \\ \theta_w \end{Bmatrix}$$

Por lo que se puede expresar un vector de desplazamientos del elemento en los ejes locales y globales como:

$$\begin{Bmatrix} x_1 \\ y_1 \\ z_1 \\ \theta_{x1} \\ \theta_{y1} \\ \theta_{z1} \\ x_2 \\ y_2 \\ z_2 \\ \theta_{x2} \\ \theta_{y2} \\ \theta_{z2} \end{Bmatrix} = \begin{bmatrix} R & 0 & 0 & 0 \\ 0 & R & 0 & 0 \\ 0 & 0 & R & 0 \\ 0 & 0 & 0 & R \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ w_1 \\ \theta_{u1} \\ \theta_{v1} \\ \theta_{w1} \\ u_2 \\ v_2 \\ w_2 \\ \theta_{u2} \\ \theta_{v2} \\ \theta_{w2} \end{Bmatrix} \quad \textbf{(2.17)}$$

$$T = \begin{bmatrix} R & 0 & 0 & 0 \\ 0 & R & 0 & 0 \\ 0 & 0 & R & 0 \\ 0 & 0 & 0 & R \end{bmatrix}$$

Las matrices de rigidez y de masa del elemento quedan expresadas en el sistema global de coordenadas como,

$$K_g = T^T K_l T \quad \textbf{(2.18)}$$

$$M_g = T^T M_l T \quad \textbf{(2.19)}$$

2.3.3 Ensamble

Las matrices de masa y rigidez de un sistema, en este caso de estudio la viga, es un ensamble de las matrices de cada elemento. Para realizar este ensamble, se necesita saber que grados de libertad están asociados a que elemento en la estructura. Por ejemplo, la matriz de rigidez de la viga será entonces un ensamble de las matrices de rigidez (cuya cantidad es el número de elementos), ubicadas en los grados de libertad correspondientes. La matriz de masa se construye de manera equivalente.

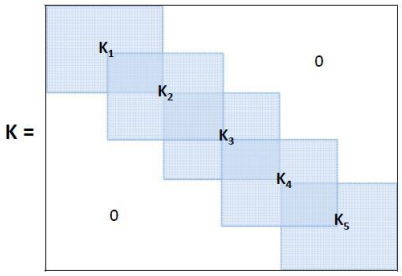


Figura 2.7 Ensamble de matrices de rigidez. Ilustración original en [9]

2.4 Ajuste de Modelos

Los métodos de ajuste de modelos permiten verificar y corregir los modelos en elementos finitos por medio de los datos experimentales. De esta manera, el resultado de un ajuste de modelo es un modelo en elementos finitos que es más confiable para predicciones futuras de la dinámica de una estructura [9].

La Figura 2.8 muestra el procedimiento típico para un ajuste de modelo.

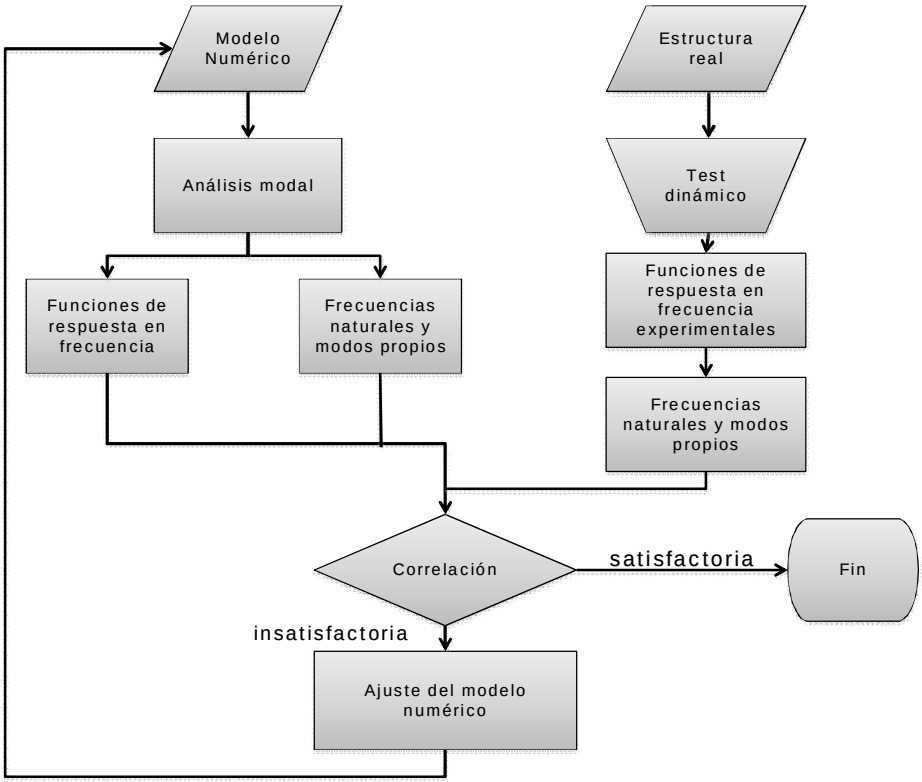


Figura 2.8 Esquema general de proceso de ajuste de modelos

2.4.1 Técnicas de correlación entre parámetros modales

Para lograr ajustar el modelo numérico a los datos experimentales es necesario definir parámetros que entreguen el nivel de correlación entre los datos numéricos y experimentales. A continuación se describen algunas técnicas de correlación usuales y sus indicadores.

2.4.1.1 Diferencia en las frecuencias de resonancia

La forma más sencilla de verificar la correlación es comparando las frecuencias naturales analíticas y experimentales. La diferencia máxima permitida dependerá de la precisión de las frecuencias naturales [9,14].

2.4.1.2 Modal Scale Factor (MSF)

La función del "Factor de escala modal (MSF)" es entregar los medios para normalizar todas las estimaciones del mismo modo de vibración, tomando en cuenta diferencias en magnitud y fase de estos. Luego, los modos experimentales se pueden escalar a los modos analíticos multiplicados por el MSF correspondiente.

$$\phi_{e,i}^{*} = \phi_{e,i} \cdot MSF_i \quad \textbf{(2.20)}$$

$$MSF_i = \frac{\phi_{a,i}^{T}\phi_{e,i}}{\phi_{e,i}^{T}\phi_{e,i}} \quad \textbf{(2.21)}$$

Donde $\phi_{e,i}^{*}$ y $\phi_{e,i}$es el i-ésimo modo experimental escalado y original,$\phi_{a,i}$ es el i-ésimo modo analítico, y MSF_i es el factor de escala modal para el modo i. Al multiplicar el modo experimental por el MSF correspondiente también se soluciona el problema de los modos analíticos y experimentales que estuviesen desfasados en 180°.

2.4.1.3 Modal Assurance Criterion (MAC)

La función del "Modal Assurance Criterion (MAC)" es proveer una medida de la consistencia (grado de linealidad) entre estimaciones de uno o varios modos de vibración, y queda definido como:

$$MAC_{ij} = \frac{(\phi_{a,i}^{T}\phi_{e,i})^2}{(\phi_{a,i}^{T}\phi_{a,i})(\phi_{e,i}^{T}\phi_{e,i})} \quad \textbf{(2.22)}$$

Donde $\phi_{a,i}$ es el i-ésimo modo analítico y $\phi_{e,j}$ es el j-ésimo modo experimental. Un valor de 0 indica que no hay correlación, mientras que un valor de 1 indica dos modos perfectamente correlacionados.
Al ordenar todos los valores $MAC_{i,j}$ en una matriz, la diagonal debería tener valores altos, sobre 0.8, para una buena correlación. Esta correlación no depende de la escala de los modos, sino que sólo de la forma de éstos.

2.5 Incorporación de daños en el modelo numérico

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

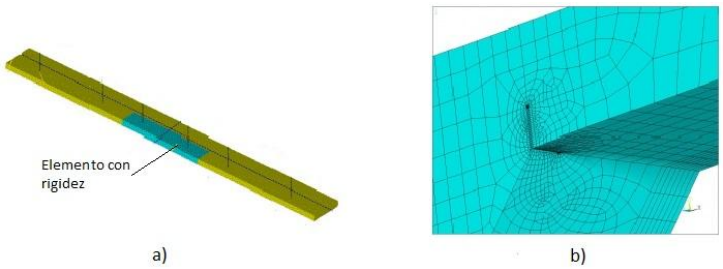


Figura 2.9 Esquema de (a) Reducción local de rigidez. Ilustración original en [15], (b) Modelo de grieta Ilustración original en [16]

La reducción de rigidez es la forma más sencilla de modelar una estructura con daños. Para esto se introduce un factor de rigidez efectiva β, el que representa la proporción efectiva de la rigidez en comparación a la rigidez original, afectando la contribución de las rigideces locales en la matriz de rigidez global.

$$[K] = \sum \beta_i [K_i] \qquad \textbf{(2.23)}$$

Donde $[K_i]$ es la matriz local de rigidez, es decir, la rigidez del elemento de viga.

Se define también una constante complementaria a la rigidez efectiva, el factor de reducción de rigidez, definido como $1 - \beta_i$.

Por otro lado, se pueden obtener mejores resultados al modelar las zonas cercanas a la grieta (Figura 2.9 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.

2.6 Redes neuronales artificiales (ANN)

Las redes neuronales artificiales (ANNs) son modelos computacionales que emulan el funcionamiento del cerebro, sobre la base del aprendizaje a través de la experiencia, con la consiguiente extracción de conocimiento a partir de ésta.

Los elementos básicos de un sistema neuronal biológico son las neuronas, agrupadas en redes y organizadas a través de una estructura de capas. En un sistema artificial se puede establecer una estructura jerárquica similar, donde una ANN se conforma de procesadores elementales (neuronas artificiales), conectadas a otras neuronas o bien a entradas externas y con una salida que permite propagar las señales por múltiples caminos. Un conjunto de neuronas cuyas entradas provienen de la misma fuente y cuyas salidas se dirigen al mismo destino constituyen lo se denomina capa o nivel, cuya agrupación conforma el sistema neuronal completo (Figura 2.10)

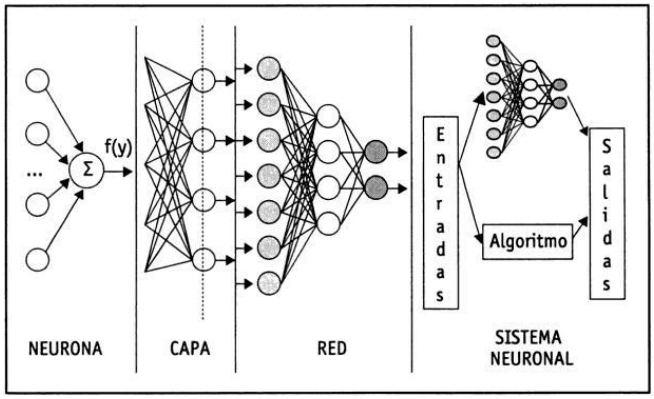


Figura 2.10 Estructura jerárquica de modelo basado en ANN. Ilustración original en [10]

Cada neurona pondera las entradas que recibe lo que constituye el funcionamiento básico de intercambio de información dentro de la red.

Las ANN pueden considerarse modelos de cálculo caracterizados por algoritmos muy eficientes que operan de forma masivamente paralela y permiten desarrollar tareas cognitivas como el aprendizaje de patrones, la clasificación entre otros. Un sistema neuronal está compuesto por los siguientes elementos básicos:

- Un conjunto de procesadores elementales o neuronas artificiales.
- Un patrón de conectividad o arquitectura.
- Una dinámica de activaciones.
- Una regla o dinámica de aprendizaje.
- El entorno donde opera.

Luego, cada neurona está caracterizada por los siguientes elementos básicos:

- Un valor o estado de activación inicial ($a_j(t - 1)$), anterior a la recepción de los estímulos. Vector que contiene los valores en el tiempo $t= t – 1$.
- Unos estímulos o entradas a la neurona (x_i), con pesos asociados w_{ij}, es decir es el peso de la conexión que va de la neurona i a la j.
- Una función de propagación, que determina la entrada total a la neurona (Net_j).
- Una función de activación o transferencia (F), que combina las entradas a la neurona con el estado de activación inicial para producir un nuevo valor de activación.
- Una función de salida (f). que transforma el estado final de activación en la señal de salida.
- Una señal de salida.
- Un regla de aprendizaje, que determina la forma de actualización de los pesos de la red (aprendizaje).

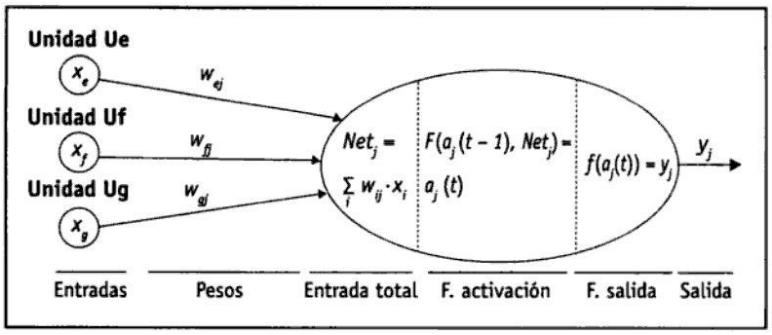


Figura 2.11 Modelo genérico de neurona. Ilustración original en [10]

Para obtener la señal de salida, la función de activación o transferencia (función F) combina la entrada total a la j-ésima neurona (Net_j), obtenido a partir de los estímulos y pesos recibidos (ver ecuación 2.24), con el estado de inicial de la neurona para producir un nuevo estado de activación $a_j(t)$ (ecuación 2.25).

$$Net_j = \sum_{i=1}^{N} w_{ij}(t) \cdot x_i(t) + \theta_j(t) \qquad \textbf{(2.24)}$$

Donde w_{ij} = peso entre neurona i y j, x_i = entrada desde neurona i, θ_j = umbral de la neurona j.

$$a_j(t) = F\left[a_j(t-1), Net_j(t)\right] \qquad \textbf{(2.25)}$$

A menudo f_j es de tipo sigmoidal, y suele ser la misma para todas las unidades de la misma capa.

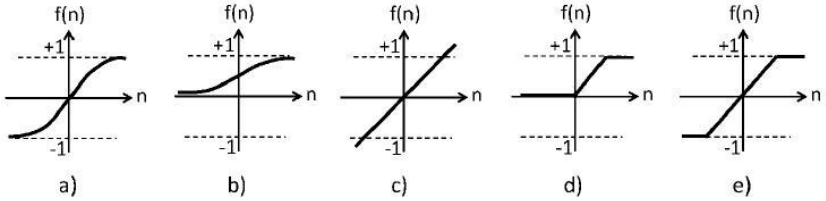


Figura 2.12 Funciones de transferencia a) Sigmoidea tangente hiperbólica, b) Sigmoidea logarítmica, c) Lineal, d) Lineal saturada y e) Lineal saturada simétrica. Ilustración original en [1]

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

Las arquitecturas neuronales pueden clasificarse según diferentes criterios:

Según su estructura en capas:

- Redes monocapa, compuestas por una única capa de neuronas.

- Redes multicapa, cuyas neuronas se organizan en varias capas (de entrada, ocultas y de salida). La capa a la que pertenece la neurona puede distinguirse mediante la observación del origen de las señales que recibe y el destino de la señal que genera.

Según el flujo de datos en la red:

- Redes unidireccionales o de propagación hacia adelante (feedforward), en las que ninguna salida neuronal es entrada de unidades de la misma capa o capas precedentes. La información circula en un único sentido, desde las neuronas de entrada hacia las neuronas de salida.
- Redes de propagación hacia atrás (feedback), en las que las salidas de las neuronas pueden servir de entradas a unidades del mismo nivel o niveles previos. Las redes de este tipo que presentan lazos cerrados se denominan sistemas recurrentes (Ver Figura 2.13).

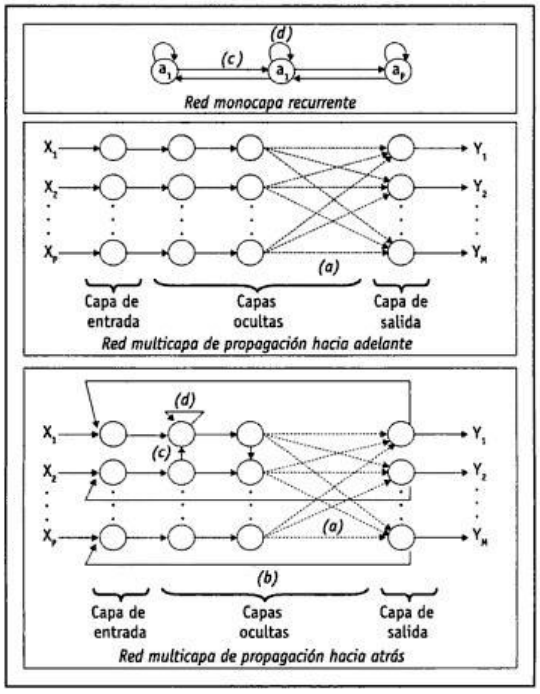


Figura 2.13 Diferentes arquitecturas según conexionado y capas. Ilustración original en [10]

Los métodos de entrenamiento para ANN pueden ser clasificados en dos grandes grupos: aprendizaje supervisado y aprendizaje no supervisado (Tabla 2.1). Los métodos de aprendizaje supervisado consideran un entrenamiento controlado donde se conocen las salidas y entradas de un cierto grupo y luego estas se comparan con la salida de la red, realizando un proceso iterativo. Por otro lado, en el aprendizaje no supervisado la red se auto organiza para agrupar y ajustar los datos sin contar con información respecto a su salida.

Tabla 2.1 Clasificación métodos de entrenamiento. Basado en [1]

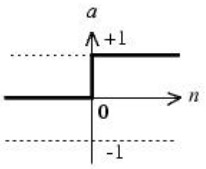
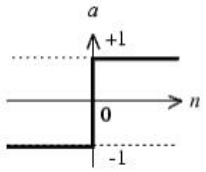
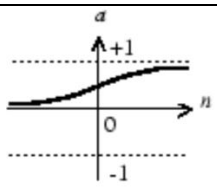
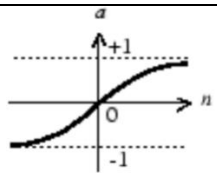
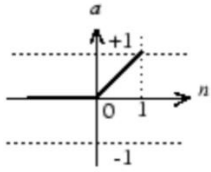
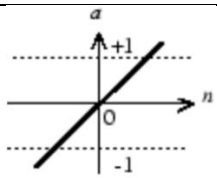
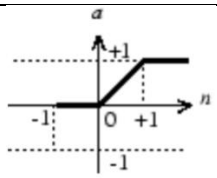
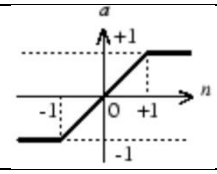
Métodos supervisados	
Entrenamiento por corrección de error	Se ajustan los pesos de las conexiones en función de las diferencias entre los valores deseados y los obtenidos en la salida de la red.
Entrenamiento por refuerzo	A diferencia del mecanismo anterior, no se cuenta con una referencia exacta respecto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

2.6.1 Funciones de transferencia

Como parte de la definición de parámetros de la red se debe determinar cuál es la función de transferencia que permite maximizar el desempeño. Con este fin, existen varias funciones de transferencia predefinidas en Matlab y también la posibilidad de ser definida por el usuario.

Las funciones que se analizan en este trabajo, se detallan a continuación.

Tabla 2.2 Funciones de transferencia en estudio

Nombre de la función	Nombre en Matlab	Gráfico	Definición
Hard-limit	Hardlim		$a = hardlim(n) = \begin{cases} 1\ si\ n \geq 0 \\ 0\ en\ otro\ caso \end{cases}$
Hard-limit simétrica	Hardlims		$a = hardlims(n) = \begin{cases} 1\ si\ n \geq 0 \\ -1\ en\ otro\ caso \end{cases}$
Sigmoidea logarítmica	Logsig		$a = logsig(n) = \frac{1}{(1 + \exp(-n))}$
Sigmoidea tangente hiperbólica	Tansig		$a = tansig(n) = \frac{2}{(1 + \exp(-2 \cdot n))} - 1$
Lineal positiva	Poslin		$a = purelin(n) = \begin{cases} n, si\ n \geq 0 \\ 0, si\ n \leq 0 \end{cases}$
Lineal	Purelin		$a = purelin(n) = n$
Lineal saturada	Satlin		$a = satlin(n) = \begin{cases} 0\ si\ n \leq 0 \\ n\ si\ 0 \leq n \leq 1 \\ 1\ si\ 1 \leq n \end{cases}$
Lineal simétrica saturada	Satlins		$a = satlins(n) = \begin{cases} -1\ si\ n \leq -1 \\ n\ si\ -1 \leq n \leq 1 \\ 1\ si\ 1 \leq n \end{cases}$

2.6.2 Función de desempeño

El proceso de entrenamiento de una red neuronal involucra la determinación de los pesos y bias de la red para optimizar el desempeño de la red, el cual queda definido por la función de desempeño de la red.

La función usada por defecto por el programa es el error cuadrático medio (MSE en inglés Mean Square Error), el cual se define como,

$$F = mse = \frac{1}{N}\sum_{i=1}^{N}(e_i)^2 = \frac{1}{N}\sum_{i=1}^{N}(t_i - a_i)^2 \qquad \textbf{(2.26)}$$

Donde t_i es el valor i de salida, a_i el valor obtenido por la red y N el número de datos de entrenamiento de la red.

Los datos de entrenamiento se dividen en dos grupos “entrenamiento”, y “validación”.

En la etapa de entrenamiento, el programa calcula el MSE en cada iteración sobre el grupo de entrenamiento y lo llama “MSE de entrenamiento” y sobre el grupo de validación y lo llama “MSE de validación”.

2.6.3 Generación del set de datos de entrenamiento y validación de la red

Considerando la naturaleza de los datos y siguiendo el criterio de Meruane y Mahu [11], el set de datos de entrenamiento se ha generado con niveles de daño distribuidos uniformemente, como:

$$\text{Número de datos de entrenamiento} = (\text{niveles de daño})^{(l)} \times C(N, l)$$

Donde l es la cantidad de daños simultáneos, N es el número de elementos de la estructura y C es el símbolo de combinatoria.

Debido a que el número de datos de entrenamiento crece de manera exponencial con la cantidad de daños simultáneos que se consideran en la estructura, en este caso se estudian dos daños simultáneos y niveles de daño distribuidos uniformemente entre 0 y 95%.

El set de datos de validación se ha generado de manera análoga, pero con un desfase que asegura que los niveles de daño de los dos set no sean los mismos.

2.7 Indicadores de desempeño de la red neuronal entrenada

El desempeño de la red se mide construyendo 3 indicadores los cuales son obtenidos de un set de datos de pruebas, el cual contiene valores diferentes a los utilizados en el entrenamiento. Estos indicadores están basados en los definidos por Yun, Yi and Bahng [5].

Estos indicadores son el Error medio de estimación (mean sizing error), Error de daño no detectado (damage missing error) y Error de detección de daño errónea (False alarm error).

El error medio de estimación (EME) es el valor promedio de los errores de estimación absolutos y se define como,

$$EME = \frac{1}{NO} \sum_{i} |y_i - o_i| \qquad \textbf{(2.27)}$$

Donde y_i y o_i, son la salida estimada por la red y la objetivo, respectivamente, para el nodo i, y NO es el número de nodos de salida.

El error de daño no detectado (DME) se define como,

$$DME = \frac{1}{NT} \sum_{i} \varepsilon_i^I , \quad 0 \leq DME \leq 1 \qquad \textbf{(2.28)}$$

Donde $\varepsilon_i^I = 0$ si el iésimo elemento dañado esta correctamente detectado y $\varepsilon_i^I = 1$ en caso contrario. El valor NT corresponde al número de lugares con daño real, si $DME = 0$ todos los lugares con daño fueron detectados correctamente.

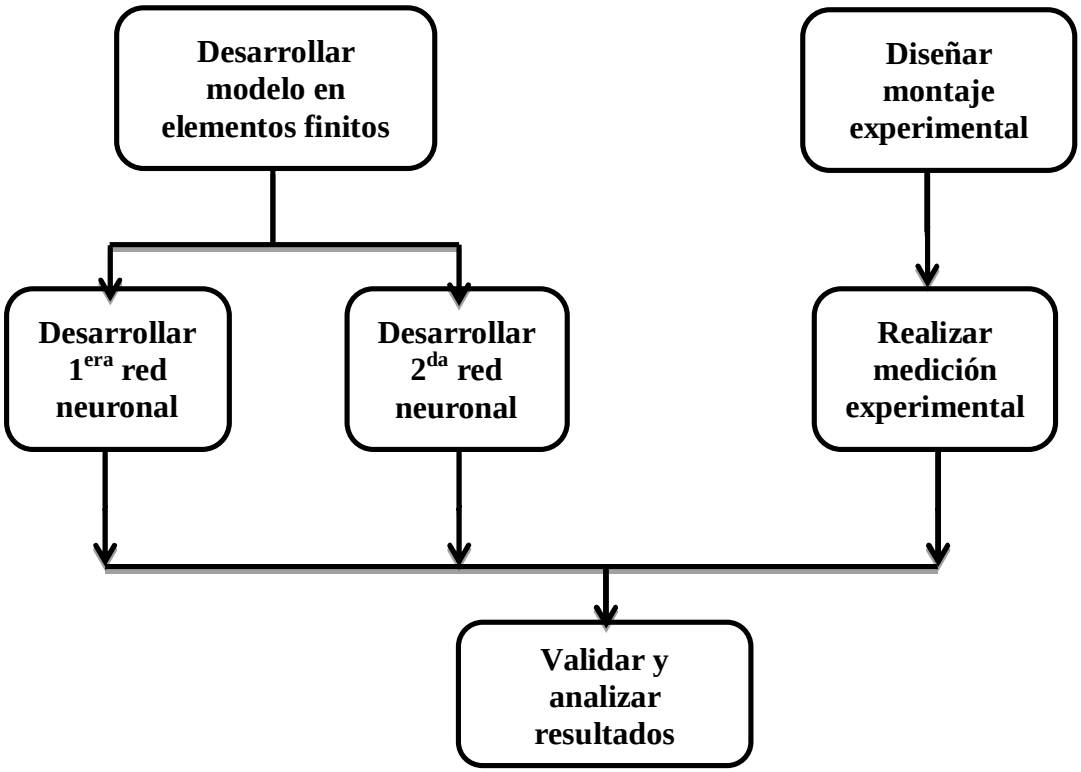
El error de detección de daño errónea (FAE) es definido como,

$$FAE = \frac{1}{NF} \sum_{i} \varepsilon_i^{II} , \quad 0 \leq FAE \leq 1 \qquad \textbf{(2.29)}$$

Donde $\varepsilon_i^{II} = 0$ si el iésimo daño detectado es daño real y $\varepsilon_i^{II} = 1$ de lo contrario. El valor NF es el número de lugares con daño predicho por la red. Si $FAE = 0$ todos los lugares de daño detectados son lugares de daño reales.

3 Metodología específica

El trabajo considera las etapas interrelacionadas indicadas en el siguiente diagrama:



Con mayor precisión, estas etapas significan:

- Desarrollar el modelo en elementos finitos de la estructura en estudio, que entregue las frecuencias de resonancia y de anti-resonancia para diferentes escenarios de daño.
- Determinar los parámetros de la primera red neuronal, entrenarla, validarla y probarla respecto a los indicadores definidos. Esta red determina cuales sub-estructuras presentan daño, y pasarán a ser analizadas por la segunda red.
- Utilizar la segunda red para la identificación de daño en las sub-estructuras.
- Diseñar el montaje experimental.
- Realizar mediciones experimentales y emplear estos datos para validar los modelos.
- Analizar y contrastar resultados numéricos y experimentales.

El trabajo de modelación en elementos finitos y el desarrollo de las redes neuronales se realizarán en Matlab.

La parte experimental se realizará utilizando el método de obtención de datos mostrado en la sección 2.2.

3.1.1 Parámetros de entrada de red

Siguiendo el procedimiento empleado por Meruane y Mahu[11], propuesto por Lee et al. [12], los valores de entrada a la red se definen como los cambios en los parámetros modales en vez de sus valores absolutos, disminuyendo de esta manera el error asociado al caso base en el modelo de elementos finitos.

Es decir, que como se explicita a continuación, los valores experimentales de los casos con daño son comparados con el caso sin daño experimental y similarmente los casos con daño numérico se comparan con el caso sin daño numérico.

En este caso, los valores de entrada corresponden al cambio de las frecuencias de anti-resonancia al haber daño, con respecto al caso sin daño:

$$x_{ixn} = \frac{\omega_{r,i,n}^{D} - \omega_{r,i,n}^{ND}}{\omega_{r,i,n}^{ND}} \quad \textbf{(3.1)}$$

Donde $\omega_{r,i,n}$ es la i-ésima frecuencia de anti-resonancia de la enésima FRF, el superíndice D se refiere a una estructura dañada y ND a la estructura no dañada.

Las frecuencias de resonancia también se incorporan a las entradas de la red, similarmente a las frecuencias de anti-resonancia, de la siguiente manera:

$$x_{m} = \frac{\omega_{R,m}^{D} - \omega_{R,m}^{ND}}{\omega_{R,m}^{ND}} \quad \textbf{(3.2)}$$

Donde $\omega_{R,m}$ es la frecuencia de resonancia del modo m, D y ND igual que en ecuación anterior

Además, para considerar el ruido experimental, los datos de entrada simulados pueden ser contaminados con ruido aleatorio. Como proponen Hjelmstad y Shin [13], cada set de datos perturbados es creado añadiendo ruido aleatorio uniformemente distribuido a los datos numéricos, de la siguiente manera,

$$\omega_{r,i,n} = \omega_{r,i,n}(1 + \xi) \quad \textbf{(3.3)}$$

Donde ξ es un número aleatorio distribuido uniformemente con una amplitud específica entre [-a,a] y a su vez a=[0,1-5]. La varianza del ruido de perturbación debería ser la misma que la del ruido medido experimentalmente.

Por otro lado, los valores de salida son niveles de daño entre 0 y 1 donde cero es un elemento sin daño y 1 un elemento completamente dañado.

4 Análisis de la estructura numérica y experimental

En este capítulo se describe la estructura estudiada. Primero se muestra conceptualmente y el modelo en elemento finitos utilizado. Luego se describe el montaje experimental y finalmente el ajuste de modelos realizado.

4.1 Definición del caso de estudio

El caso a analizar está basado en el tipo de estructura mostrado en la Figura 4.1.

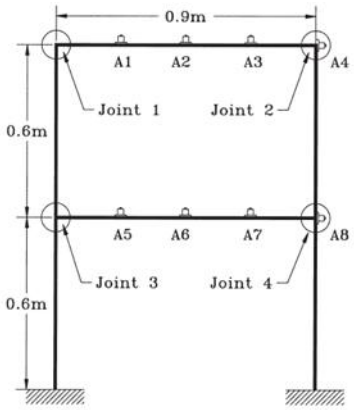


Figura 4.1 Estructura de barras utilizada en [5]

El material y las dimensiones están seleccionados para obtener frecuencias de resonancia y anti-resonancia dentro del rango de trabajo de los sensores a utilizar.

En la Figura 4.2 se muestra un diagrama de la estructura donde se muestran las divisiones por subestructura y los elementos. En las figuras siguientes se observa el mismo esquema con la numeración de los nodos y las dimensiones de la estructura construida.

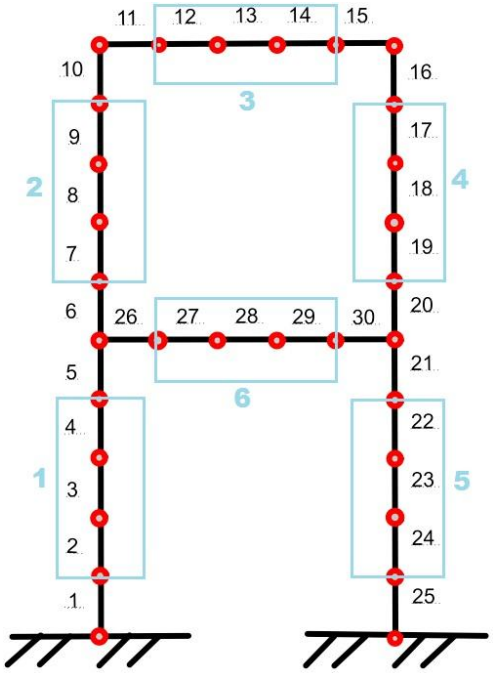


Figura 4.2 Esquema de estructura estudiada

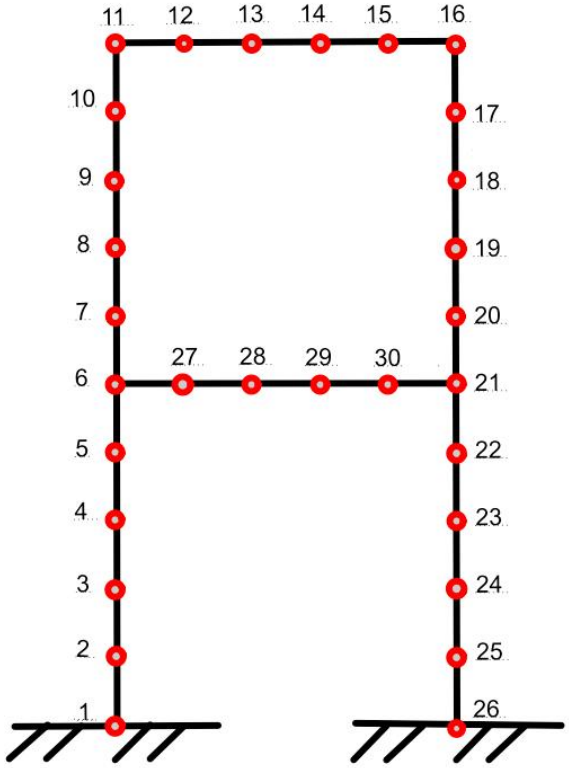


Figura 4.3 Esquema con nodos numerados

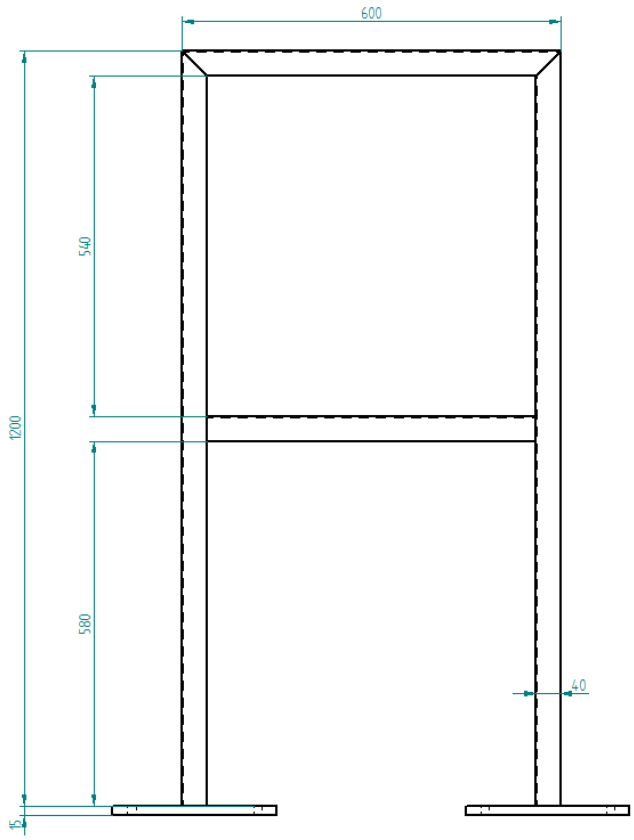


Figura 4.4 Modelo de estructura de barras (sin uniones) con medidas en milímetros

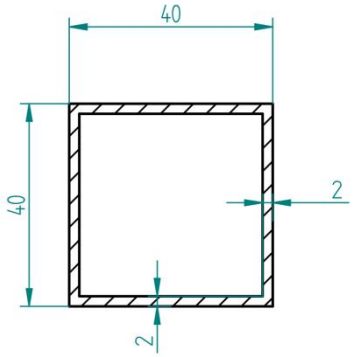


Figura 4.5 Perfil utilizado en la estructura

4.2 Modelo en elementos finitos

Como se explicó en la sección 2.3, la estructura se modeló considerando elementos de viga en 3D utilizando el software Matlab.

Las propiedades físicas consideradas para el material fueron las siguientes:

Tabla 4.1 Propiedades físicas del material estudiado

Propiedad	Valor
Módulo de Young (E)	$2.1*10^{11}$ [Pa]
Coeficiente de Poisson (ν)	0.3
Densidad (ρ)	7840 [kg/m^3]

Las primeras diez frecuencias naturales de resonancia que se obtuvieron del modelo se muestran en la Tabla 4.2.

Tabla 4.2 Primeras 10 frecuencias de resonancia del modelo en elementos finitos

	Frecuencia [Hz]
1	10,4
2	25,7
3	31,8
4	61,8
5	84,3
6	104,9
7	183,6
8	188,0
9	248,8
10	258,3

En la Figura 4.6 se observan los primeros 10 modos de vibración de la estructura, en los cuales se observa qué puntos se mueven relativamente más respecto a otros y cuál es el desplazamiento esperado de la estructura. Esta información se utilizó en la selección de los puntos de daño experimental.

Las formas de los 10 modos obtenidos fueron comparadas visualmente con los modos de la estructura experimental, para verificar su correspondencia. Además, la razón entre las diferencias de los primeros 6 modos numéricos y experimentales fue usada para ajustar el modelo numérico, como se explica en la sección 4.5.

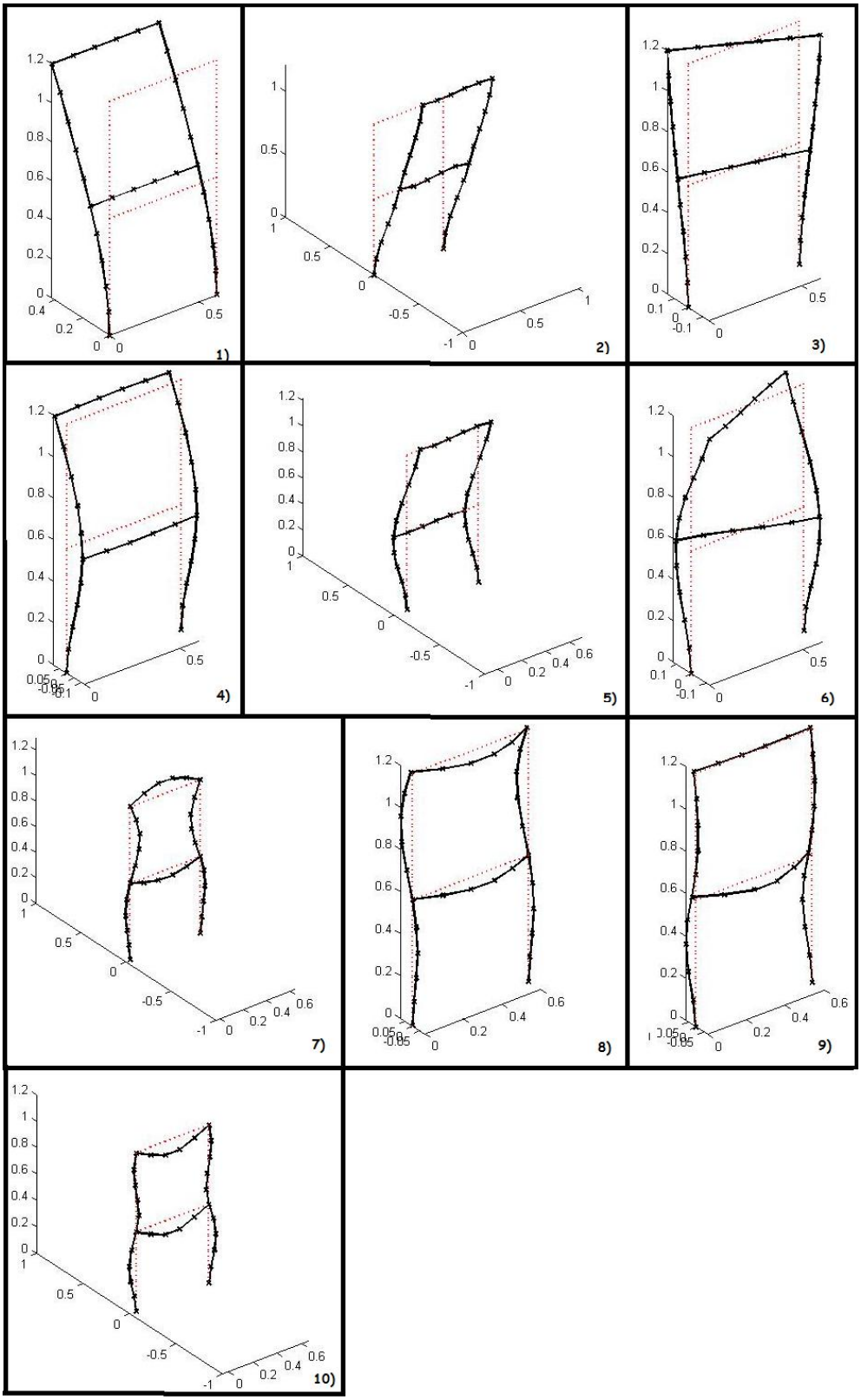


Figura 4.6 Primeros 10 modos de vibración obtenidos del modelo de EF

El presente trabajo considera la detección, localización y cuantificación de daño por medio del uso de dos redes neuronales. La primera encargada de localizar la subestructura dañada y la segunda de localizar con mayor precisión y cuantificar el daño.

4.3 Diseño de estructura experimental

El diseño de la estructura se realizó considerando las dimensiones usadas en el modelo en elementos finitos y para las cuales se pueden obtener mediciones con los instrumentos disponibles.

Para optimizar el uso de materiales se diseñó la estructura desmontable, uniéndose por medio de uniones y pernos adecuados. De esta manera es posible simular una cantidad de escenarios de daño considerable con pocos elementos. Las partes que conformaron la estructura se muestran en detalle en el Anexo A. los cortes se le realizaron a cada pieza utilizando una máquina fresadora.

La Figura 4.7 muestra el montaje de la estructura para el caso sin daño.

Figura 4.7 Montaje experimental de estructura sin daño

En la Figura 4.9 se señalan los elementos con daño, los triángulos simbolizan daño pequeños y los hexágonos daños grandes.

Los daños pequeños son cortes de 10 [mm] de profundidad y de 1 [mm] de ancho, mientras que los daños grandes son cortes de 20 [mm] y de igual ancho, tal como se aprecia en la vista lateral en la Figura 4.8.

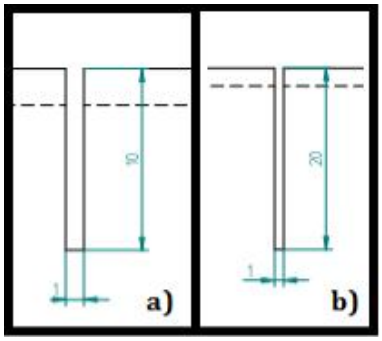


Figura 4.8 Tipos de daños infligidos a la estructura, a) daño pequeño y b) daño grande

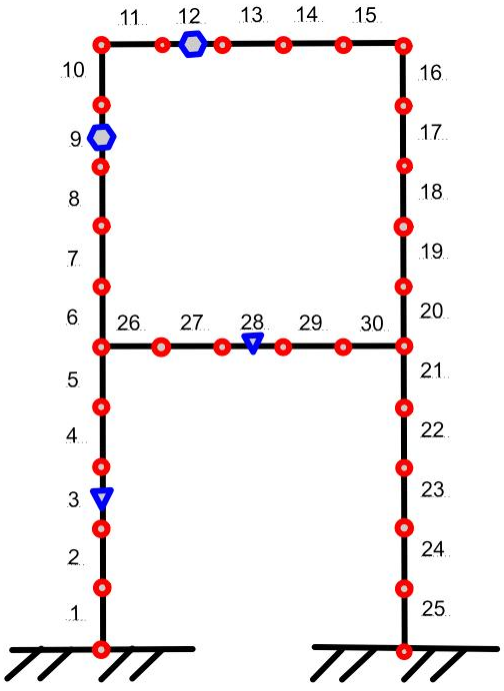


Figura 4.9 Ubicación de daño en la estructura

La estructura se divide en 4 partes principales, 2 de ellas intercambiables. Respecto a eso y a los daños mostrados en la Figura 4.9, se considera un set de 4 piezas sin daño y un set de 5 piezas con los daños detallados en la Tabla 4.3.

Tabla 4.3 Piezas con daño y su tipo

Nº de pieza	Nº de elemento	Tipo de daño
1	3	Pequeño
2	9	Grande
3	3 y 9	Pequeño (3) y grande (9)
4	12	Grande
5	28	Pequeño

Los daños se ubican transversales a la sección, y el detalle de estos y su orientación se puede observar en el Anexo A.

4.4 Mediciones experimentales

En esta sección se describirá la disposición y características de la instalación experimental.

En la Figura 4.10 se muestra en verde el sistema de ejes utilizado como referencia en las mediciones, y a los cuales se hace referencia al presentar los resultados. En todas las mediciones se utilizaron dos acelerómetros para registrar la respuesta de la estructura, lo que se muestra en celeste, ambos ubicados en el nodo 11 de la estructura (entre los elementos 10 y 11, ver Figura 4.3). Un acelerómetro se instaló para registrar en dirección –X y el otro en dirección –Z.

Cada medición consistió entonces en el promedio de al menos 6 registros de golpes del martillo modal seleccionados de entre los que consisten en un golpe seco, es decir eliminando golpes dobles y se encuentran en el rango de 80 a 200 [N]. Este proceso aplicado a cada nodo en dirección X y Z. Notar que existen nodos a los cuales no se les puede excitar en dirección X, por lo que disponen de menor información que los otros (ver Tabla 4.5).

El software de adquisición de datos utilizado fue Data Adquisition and Analysis (DAS). La frecuencia de muestreo fue de 10240 [Hz] con 32768 puntos de medición. Las características del martillo y los sensores se muestran en la Tabla 4.4.

Tabla 4.4 Características de martillo y sensores.

Elemento	Tipo de señal	Sensibilidad[1]	Filtro pasa altos [Hz][2]
Martillo	Fuerza	10.2[mV/(N)]	0.7
Acelerómetro en -Z	Aceleración	95.6[mV/(G)]	0.7
Acelerómetro en -X	Aceleración	96[mV/(G)]	0.7

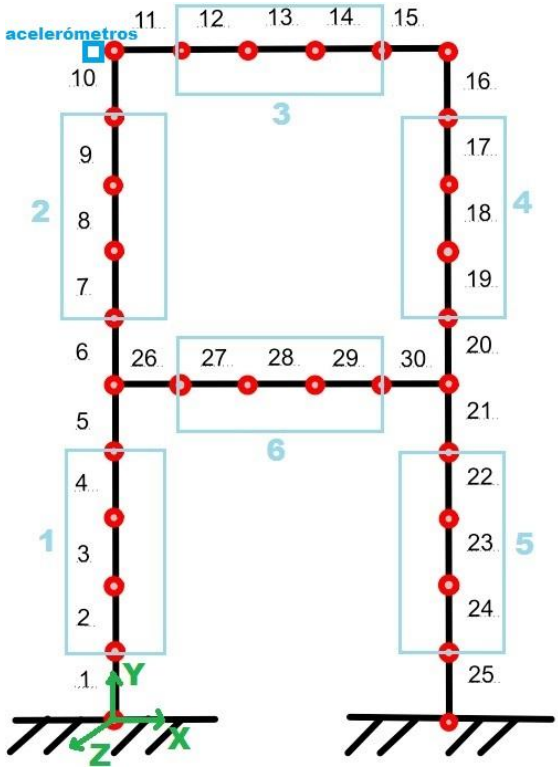


Figura 4.10 Esquema de instalación para medición

Luego de realizar la medición completa de la estructura, se procedió por medio del software Femtools a la extracción de los primeros 6 modos de vibración con las frecuencias de resonancia correspondientes.

[1] La sensibilidad (sensitivity) es el voltaje de salida producido por una fuerza medida en g. La frecuencia del voltaje de salida AC coincidirá con la frecuencia de las vibraciones. El nivel de salida será proporcional a la amplitud de las vibraciones. Los acelerómetros con salidas bajas son usados para medir niveles de vibraciones altas mientras que los acelerómetros con salidas altas son usados para medir niveles de vibración bajos.

[2] Un filtro pasa altos (high-pass filter, HPF) es un filtro electrónico que deja pasar las señales de alta frecuencia pero atenua (reduce la amplitud de) las señales con frecuencias más bajas que la frecuencia de corte.

Posteriormente, recuperando las FRF registradas por el software, se obtuvieron por medio de selección gráfica las frecuencias de anti-resonancia correspondientes a cada nodo dentro del análisis. (FRF indirectas excepto para el nodo 11). Se seleccionaron frecuencias hasta los 200[Hz], ya que los datos registran mucho ruido para frecuencias más altas que este valor.

En la Figura 4.11 se muestra como ejemplo del procedimiento una FRF del caso con un daño grande en el elemento 9. La FRF corresponde al nodo 10 excitado en dirección Z, donde la línea azul es el valor obtenido por medio del modelo en elemento finitos y el rojo el valor obtenido experimentalmente. Aquí se puede ver que se observan 3 frecuencias de anti-resonancia experimentales de las 4 que se ven en el modelo numérico, por lo que si se observa la Tabla 4.5 el nodo 10 registra 3 frecuencias de anti-resonancia en Z.

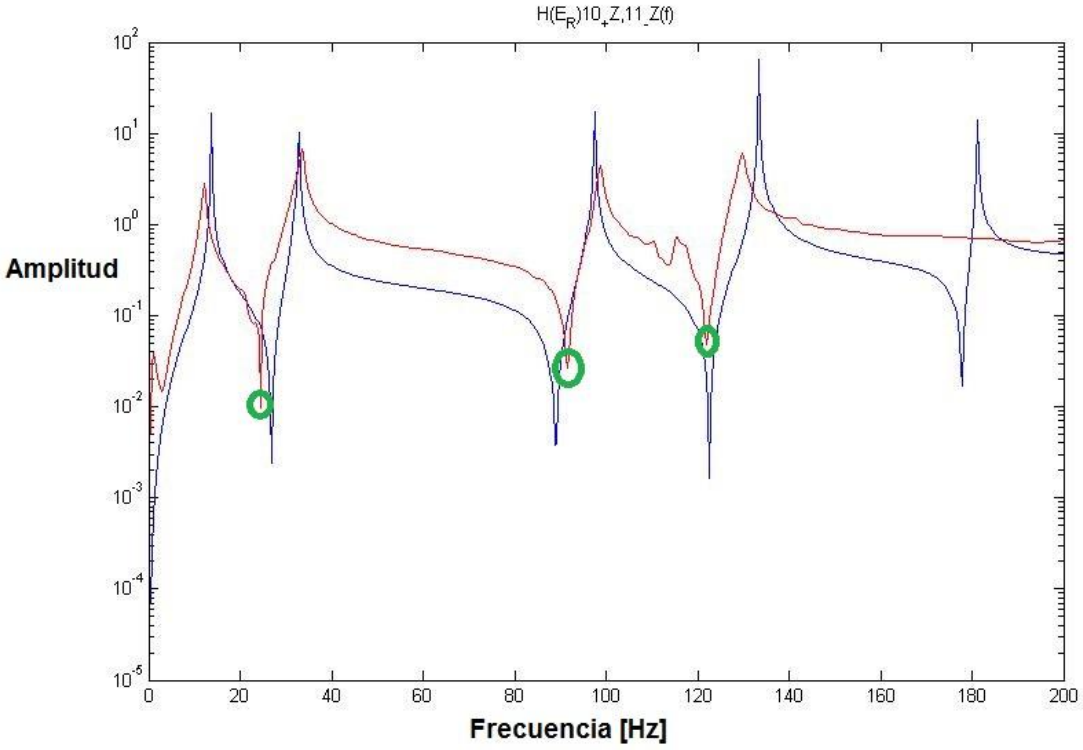


Figura 4.11 FRF indirecta, excitación en 10 en Z y respuesta en 11 en Z. Caso un daño grande en 9

Las cantidades de frecuencias de anti-resonancia experimentales obtenidas experimentalmente se muestran en la Tabla 4.5, donde se consideran las mediciones en eje Z y eje X según representación. Estas cantidades definen el número de entradas disponible para el entrenamiento de cada subestructura.

Tabla 4.5 Resumen de anti-resonancias extraíbles de FRF experimentales

Número de nodo	Cantidad de anti-resonancias		
	Z	X	
2	2	1	Subestructura 1
3	2	1	
4	2	1	
5	2	1	
7	2	1	Subestructura 2
8	3	2	
9	2	2	
10	3	2	
12	3	0	Subestructura 3
13	3	0	
14	3	0	
15	0	0	
17	0	2	Subestructura 4
18	2	1	
19	2	2	
20	1	2	
22	1	1	Subestructura 5
23	1	1	
24	1	1	
25	0	1	
27	2	0	Subestructura 6
28	2	0	
29	1	0	
30	1	0	

4.5 Ajuste de modelo

El proceso de ajuste modelo se realizó entre los modos de vibración de la estructura sin daño, obtenidos desde el modelo en elementos finitos y desde la estructura experimental. El objetivo del ajuste es conseguir que el modelo de elementos finitos represente lo más exactamente posible la situación real. Con este fin se utilizan parámetros de ajuste.

4.5.1 Frecuencias de resonancia y modos de vibración experimentales

Las frecuencias de resonancia obtenidas experimentalmente para el caso sin daño se muestran en la Tabla 4.6 junto a las correspondientes numéricas y su diferencia expresada porcentualmente. La Figura 4.12 muestra la correlación MAC de los modos. Debido a que los 6 primeros modos muestran una alta correlación y las frecuencias de resonancia correspondientes tienen diferencias en general menores al 30%, es que se decide utilizar estos datos para realizar el ajuste de modelo.

Tabla 4.6 Frecuencias experimentales y numéricas

	Frecuencia Numérica[Hz]	Frecuencia Experimental [HZ]	Diferencia [%]
1	10,36	13,8	33,13
2	25,73	26,26	2,06
3	31,79	32,79	3,13
4	61,79	103,39	67,33
5	84,27	113,22	34,36
6	104,93	133,42	27,15
7	183,57	353,07	92,34
8	188	368,24	95,87
9	248,78	410,49	65,00
10	258,30	431,37	67,01

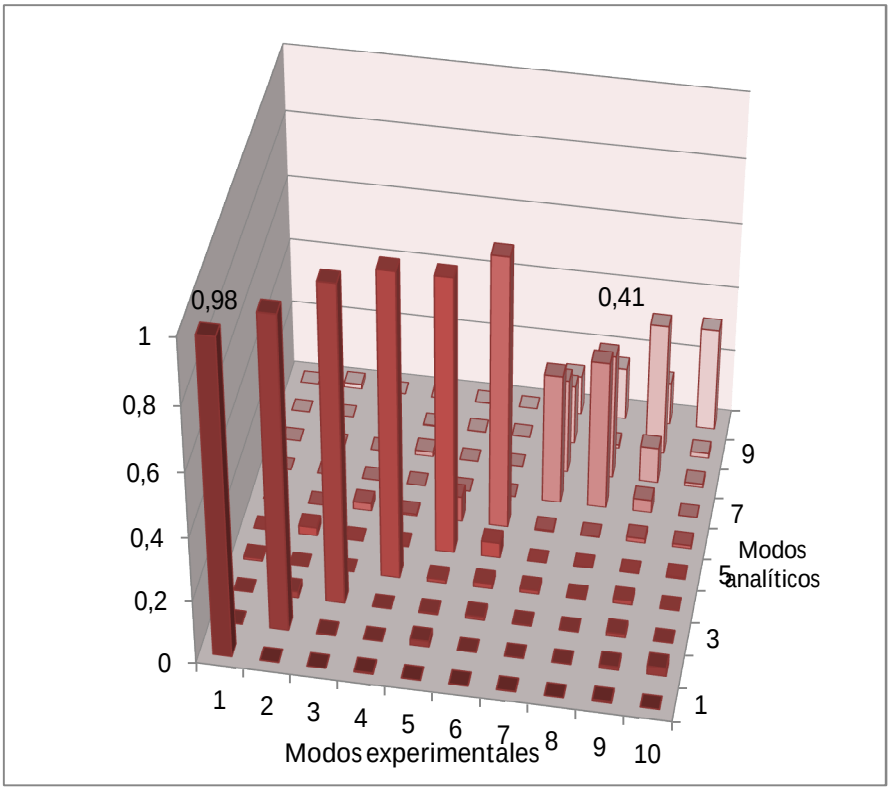


Figura 4.12 Matriz MAC primeros 10 modos

4.5.2 Parámetros de ajuste

Se consideran como variables de ajuste para el modelo numérico los siguientes parámetros.

Tabla 4.7 Variables de ajuste de modelo

Variables
Módulo de Young
Densidad
Espesor
Masas concentradas en puntos de unión
Aumento de rigidez elemento 1
Aumento de rigidez elemento 5
Aumento de rigidez elemento 10
Aumento de rigidez elemento 11
Aumento de rigidez elemento 15
Aumento de rigidez elemento 16
Aumento de rigidez elemento 21
Aumento de rigidez elemento 25
Aumento de rigidez elemento 26
Aumento de rigidez elemento 30
Elementos reducidos introducidos en las uniones (cantidad 6)

El valor del ajuste se obtuvo minimizando la siguiente función:

$$val = 20 - \left(\frac{M}{M_0} + \frac{W}{W_0} + \frac{Wa}{Wa_0}\right)$$

Donde M presenta la suma de la razón entre los modos numéricos y experimentales al cuadrado, W la suma de la razón entre las frecuencias de resonancia al cuadrado y Wa la suma de la razón entre las frecuencias de anti-resonancia al cuadrado. El sub índice 0 indica que son los valores iniciales, por lo tanto los cocientes en paréntesis al inicio son 1 cada uno y a medida que los valores experimentales se acercan a los numéricos, $M, W\ y\ Wa$ crecen, por lo que el cociente en paréntesis aumenta y val disminuye.

El valor 20 utilizado es arbitrario, haciendo con esto que el valor inicial de la función val sea 17 y aumente al acercarse los valores numéricos a los experimentales, mencionados anteriormente.

En la Tabla 4.8 se muestran las frecuencias de resonancia numéricas que entrega el modelo después del ajuste.

Tabla 4.8 Frecuencias de resonancia después de ajuste

	Frecuencia Numérica[Hz]	Frecuencia Experimental [HZ]	Diferencia [%]
1	13,1801	13,798	4,48
2	26,2574	26,258	0,00
3	32,9149	32,782	0,41
4	77,9813	103,39	24,58
5	101,8912	113,22	10,01
6	112,8403	133,42	15,42

En las Tabla 4.9 y Tabla 4.10 se presenta un resumen de los valores obtenidos de las frecuencias de anti-resonancia de las FRF experimentales y numéricas (punto de medición 11 y punto de excitación el que corresponda). En la última columna se incluyen los valores luego del ajuste.

En este análisis se separaron los planos de excitación y respuesta medidos. Cuando se hace referencia a "Z" es la respuesta y la excitación en el eje Z, y de manera análoga para X.

Tabla 4.9 Frecuencias de anti-resonancia visibles en puntos de unión de subestructuras según Z

Nodo	Anti-resonancia experimental[Hz]	Anti-resonancia numérica inicial [HZ]	Anti-resonancia numérica ajustada [HZ]
2	24,38	26,58	22,32
2	116,9	113,6	82,11
5	24,06	26,74	22,78
5	117,2	112,5	81,52
7	24,06	26,74	23
7	116,6	109,8	80,68
10	23,13	26,74	23,13
10	96,25	89,45	77,62
10	124,1	122,5	107,02
12	25	29,6	25,74
12	88,13	78,78	76,1
12	124,1	123,7	106,07
20	81,56	66,37	72,79
22	75,63	65,25	71,51
27	25,31	27,85	25,44
27	119,1	117	82,85
30	55	42,49	61,46

Tabla 4.10 Frecuencias de anti-resonancia visibles en puntos de unión de subestructuras según X

Nodo	Anti-resonancia experimental[Hz]	Anti-resonancia numérica inicial [HZ]	Anti-resonancia numérica ajustada [HZ]
2	130,9	137,4	145,79
5	130,9	137,4	101,89
7	130,6	137,5	137,71
10	95,31	92,95	93,26
10	130,9	138,5	175,15
17	95,94	93,11	95,47
17	130,9	138,5	130,89
20	130,9	135,3	130,9
20	194,1	186,4	157,71
22	131,9	136,2	168,31
25	130,9	135,8	126,88

5 Resultados

A continuación se presentan los resultados del entrenamiento de las redes neuronales para detección de daño obtenidos por medio de simulaciones en el software Matlab. También se presentan los resultados de estas redes utilizando los datos experimentales como entradas.

5.1 Entrenamiento con datos simulados de la primera red neuronal

En esta sección se presentan los resultados obtenidos a partir de los datos simulados mediante un modelo en elementos finitos para la estructura de estudio.

5.1.1 Determinación de parámetros de red

Las características principales de la red se determinan a partir de las pruebas que se muestran en la siguientes sub-secciones. Los principales parámetros a definir fueron el tipo de función de transferencia, la cantidad de capas ocultas y el número de neuronas en las capas ocultas.

Los datos de entrada se consideraron inicialmente como 10, considerando con esto que se podrían utilizar las 10 primeras frecuencias de anti-resonancia de la estructura. Similarmente, se consideraron 8 neuronas en la capa de salida, lo que correspondía a la cantidad inicial de subestructuras en las que estaba divida la estructura.

5.1.1.1 Entrenamiento de la red neuronal con diferentes funciones de salida

Utilizando la definición de datos de salida descrita en la sección 3.1.1, se realizó una prueba con una red con las características en la Tabla 5.1:

Tabla 5.1 Parámetros de primera red para prueba de función de salida

Parámetros	Valor
N° de capas ocultas	1
Función de transferencia capa oculta	Sigmoidea logarítmica (logsig)
Función de salida	Varía según prueba
N° de neuronas de capa de entrada	10
N° de neuronas de capa oculta	30
N° de neuronas de capa de salida	8
Método de entrenamiento	Levenberg-Marquardt (trainlm)
Función de desempeño	Error cuadrático medio

Los resultados se presentan en la Tabla 5.2. Estos resultados presentan el promedio de 4 pruebas para cada configuración.

La función Sigmoidea Logarítmica (logsig) fue la utilizada en la capa oculta, ya que es comúnmente usada y también porque en pruebas preliminares esta función mostró que permite un buen entrenamiento de la red. Esto es, que no haya una detención precoz debido al bajo valor del gradiente sino que la detención del entrenamiento se deba al criterio de chequeos de validación.

Las funciones que entregaron mejor desempeño fueron 3, siendo la mejor la función Lineal Saturada Simétrica (satlins). Esto se concluye en base a los valores de MSE de validación y a los valores obtenidos como salida de la red para los datos de validación. De los resultados de la red se observó que la función Lineal Pura (purelin) entrega muchos valores negativos, es decir fuera del recorrido real mientras que la función Lineal Saturada Simétrica (satlins) sólo entrega valores dentro del recorrido real.

Tabla 5.2 Resultados de entrenamiento de la red en prueba según Tabla 5.1

	Tiempo [s]	MSE entrenamiento	MSE validación
satlins	49,4	0,0023	0,0026
tansig	66,2	0,0027	0,0030
purelin	89,5	0,0024	0,0025
logsig	26,2	0,0519	0,0519
poslin	41,1	0,0519	0,0520
satlin	30,2	0,0519	0,0519

5.1.1.2 Entrenamiento de la red con diferente cantidad de capas ocultas

En la Tabla 5.3 se muestran los parámetros usados para entrenar la red. El único valor que cambia es la cantidad de capas ocultas que se utilizan.

Tabla 5.3 Parámetros de primera red para prueba de cantidad de capas

Parámetros	Valor
N° de capas ocultas	Varía entre 1 y 5
Función de transferencia capa oculta	Sigmoidea logarítmica (logsig)
Función de salida	Lineal saturada (satlins)
N° de neuronas de capa de entrada	10
N° de neuronas de capa oculta	30
N° de neuronas de capa de salida	8
Método de entrenamiento	Levenberg-Marquardt (trainlm)
Función de desempeño	Error cuadrático medio

En la Tabla 5.4 muestra el tiempo de entrenamiento para las diferentes configuraciones. La Tabla 5.4 muestra el error cuadrático medio para diferentes cantidades de capas ocultas. Se nota un empeoramiento al aumentar a 4 capas ocultas.

Se observan los mejores desempeños para 2 y 3 capas ocultas, las que presentan valores del mismo orden tanto para el entrenamiento como para la validación.

Tabla 5.4 Resultados en prueba según Tabla 5.3

Nº de capas ocultas	Tiempo [s]	MSE entrenamiento	MSE validación
1	49,4	0,0023	0,0026
2	182,4	0,0019	0,0022
3	325,6	0,0019	0,0023
4	565,0	0,0020	0,0026
5	865,1	0,0020	0,0026

De los resultados se concluye que los resultados con diferente cantidad de capas ocultas muestran resultados similares respecto al MSE de validación, pero el tiempo de entrenamiento necesitado para obtener este resultado crece rápidamente, por lo que en la práctica se utilizan sólo una y dos capas en lo posterior.

5.1.1.3 Entrenamiento de la red con diferente cantidad de neuronas en la capa oculta

En la Tabla 5.5 se muestran los parámetros usados para entrenar la red. El único valor que cambia es la cantidad de neuronas en la capa oculta. En la Tabla 5.6 se exponen los resultados para los distintos casos.

Tabla 5.5 Parámetros de primera red para prueba de cantidad de neuronas

Parámetros	Valor
N° de capas ocultas	2
Función de transferencia capa oculta	Sigmoidea logarítmica (logsig)
Función de salida	Lineal saturada (satlins)
N° de neuronas de capa de entrada	10
N° de neuronas de capa oculta	[30,40,50,60,70,80,90,100]
N° de neuronas de capa de salida	8
Método de entrenamiento	Levenberg-Marquardt (trainlm)
Función de desempeño	Error cuadrático medio

Tabla 5.6 Resultados obtenidos en prueba según Tabla 5.5

Nº de neuronas	Tiempo [s]	MSE entrenamiento	MSE validación
30	169,3	0,0019	0,0022
40	266,8	0,0019	0,0023
50	429,7	0,0022	0,0027
60	828,4	0,0019	0,0024
70	1154,0	0,0019	0,0027
80	2030,7	0,0018	0,0027
90	2943,2	0,0019	0,0028
100	3981,2	0,0020	0,0023

Se puede observar en la Tabla 5.6 que para 50 y 90 neuronas en la capa oculta se tienen los valores más bajos del MSE de validación. Ambos MSE no siguen una tendencia clara con respecto al número de neuronas.

5.1.1.4 Entrenamiento de la red variando el número de neuronas por capa y la función de transferencia

Debido a que podría existir una relación entre el número de neuronas por capa y la función de salida utilizada, presentando un mejor desempeño en una combinación específica es que se realizó un estudio posterior variando el número de neuronas por capa y la función de transferencia.

Tabla 5.7 Parámetros de primera red para prueba de cantidad de neuronas y función de transferencia

Parámetros	Valor
N° de capas ocultas	2
Función de transferencia capa oculta	Sigmoidea logarítmica (logsig)
Función de salida	Lineal saturada (satlins), Tangente sigmoidea (tansig), Lineal pura (purelin) y Sigmoidea logarítmica (logsig)
N° de neuronas de capa de entrada	10
N° de neuronas de capa oculta	[10-100]
N° de neuronas de capa de salida	8
Método de entrenamiento	Levenberg-Marquardt (trainlm)
Función de desempeño	Error cuadrático medio

En las tablas siguientes se muestran los resultados obtenidos.

Tabla 5.8 Resultados obtenidos según Tabla 5.7 para valores [10,20]

N° de neuronas	10			20		
	Tiempo [s]	MSE entrenamiento	MSE validación	Tiempo [s]	MSE entrenamiento	MSE validación
satlins	52,6	0,0021	0,0021	93,3	0,0020	0,0021
tansig	57,2	0,0140	0,0141	135,5	0,0022	0,0024
purelin	66,8	0,0022	0,0022	157,7	0,0019	0,0020
logsig	18,4	0,0519	0,0519	32,6	0,0519	0,0518

Tabla 5.9 Resultados obtenidos según Tabla 5.7 para valores [30,40]

N° de neuronas	30			40		
	Tiempo [s]	MSE entrenamiento	MSE validación	Tiempo [s]	MSE entrenamiento	MSE validación
satlins	135,8	0,0019	0,0023	292,2	0,0019	0,0022
tansig	216,7	0,0022	0,0026	447,0	0,0022	0,0027
purelin	182,5	0,0020	0,0022	389,1	0,0018	0,0023
logsig	96,9	0,0519	0,0520	135,1	0,0519	0,0519

Tabla 5.10 Resultados obtenidos según Tabla 5.7 para valores [50,60]

N° de neuronas	50			60		
	Tiempo [s]	MSE entrenamiento	MSE validación	Tiempo [s]	MSE entrenamiento	MSE Validación
satlins	505,9	0,0018	0,0026	767,6	0,0018	0,0025
tansig	765,7	0,0044	0,0048	1312,5	0,0135	0,0138
purelin	518,1	0,0018	0,0025	792,5	0,0018	0,0026
logsig	459,6	0,0519	0,0520	359,5	0,0519	0,0518

Tabla 5.11 Resultados obtenidos según Tabla 5.7 para valores [70,80]

N° de neuronas	70			80		
	Tiempo [s]	MSE entrenamiento	MSE validación	Tiempo [s]	MSE entrenamiento	MSE Validación
satlins	1332,3	0,0018	0,0025	1918,8	0,0022	0,0028
tansig	1824,2	0,0031	0,0038	3222,3	0,008	0,0083
purelin	1167,5	0,0017	0,0028	1691,9	0,0017	0,0029
logsig	625,4	0,0519	0,0519	1150,7	0,0519	0,052

Tabla 5.12 Resultados obtenidos según Tabla 5.7 para valores [90,100]

N° de neuronas	90			100		
	Tiempo [s]	MSE entrenamiento	MSE validación	Tiempo [s]	MSE entrenamiento	MSE validación
satlins	3374,8	0,0021	0,0029	20852	0,002	0,0028
tansig	4603,2	0,0032	0,0037	40112	0,0085	0,0092
purelin	2408,3	0,0018	0,0026	18079	0,0018	0,0026
logsig	1628,9	0,0519	0,0519	71325	0,0519	0,0518

Como se puede observar, las funciones ‘satlins’ y ‘purelin’ son las que presentan los mejores desempeños, teniendo su mejor valor para 20 neuronas. Con una cantidad menor o mayor los resultados empeoran.

5.1.1.5 Modificación a set de entrenamiento

Debido al resultado del ajuste de modelo se modifica el set de entrenamiento y validación. Además, se agregan las frecuencias de anti-resonancia en los nodos de unión como parámetros de entrada a la red. Esto último, para mejorar el rendimiento de la red, ya que se deduce de los resultados de este capítulo que aumentar la información de entrada conduce a un mejor entrenamiento.

Tabla 5.13 Parámetros de primera red nueva configuración

Parámetros	Valor
N° de capas ocultas	1
Función de transferencia capa oculta	Sigmoidea logarítmica (logsig)
Función de salida	Lineal Saturada simétrica (satlins), Lineal Pura (purelin)
N° de neuronas de capa de entrada	34
N° de neuronas de capa oculta	[10-120]
N° de neuronas de capa de salida	6
Método de entrenamiento	Levenberg-Marquardt (trainlm)
Función de desempeño	Error cuadrático medio

Tabla 5.14 Resultados para función 'purelin' de acuerdo a Tabla 5.13

Numero de neuronas	Tiempo [s]	MSE entrenamiento	MSE validación
10	32,9	0,0052	0,0160
20	148,1	0,0028	0,0119
30	223,1	0,0020	0,0123
40	281,0	0,0011	0,0077
50	200,9	0,0018	0,0152
60	476,8	0,0026	0,0161
70	440,3	0,0036	0,0184
80	1068,0	0,0008	0,0170
90	636,3	0,0021	0,0172
100	1263,3	0,0004	0,0109
110	940,5	0,0034	0,0172
120	1041,8	0,0034	0,0215

Tabla 5.15 Resultados para función 'satlins' de acuerdo a Tabla 5.13

Numero de neuronas	Tiempo [s]	MSE entrenamiento	MSE validación
10	40,7	0,0032	0,0111
20	98,1	0,0011	0,0082
30	266,6	0,0007	0,0069
40	264,1	0,0015	0,0098
50	320,5	0,0006	0,0096
60	550,2	0,0006	0,0084
70	881,7	0,0016	0,0079
80	577,5	0,0010	0,0102
90	1278,8	0,0011	0,0075
100	1127,8	0,0006	0,0082
110	1464,8	0,0009	0,0080
120	1842,4	0,0013	0,0083

De las Tabla 5.14 y Tabla 5.15 se puede extraer que el mejor resultado se presenta utilizando la función 'satlins' con un error de validación de 0,0069 con 30 neuronas en la capa oculta.

5.1.2 Modelo final primera red

Debido a que se busca un modelo que represente de mejor manera la estructura, se modifica la representación en elementos finitos utilizando 6 elementos extra en las uniones y luego agregando estos parámetros como parámetros de ajuste. Además, se construye una base de datos de entrenamiento que considera 6 casos de daño, donde cada elemento de la subestructura tiene el mismo nivel de daño, es decir que las 6 subestructuras consideran diferentes niveles de daño pero cada elemento de cada estructura considera solo un nivel de daño para todos sus elementos. Se consideran 2 daños simultáneos en la estructura. Con estas modificaciones adicionales los resultados obtenidos son los que se muestran en Tabla 5.17 y Tabla 5.18.

Tabla 5.16 Parámetros de primera red para modelo final

Parámetros	Valor
N° de capas ocultas	1-2
Función de transferencia capa oculta	Sigmoidea logarítmica (logsig)
Función de salida	Lineal saturada simétrica (satlins)
N° de neuronas de capa de entrada	34
N° de neuronas de capa oculta	[10-120]
N° de neuronas de capa de salida	6
Método de entrenamiento	Levenberg-Marquardt (trainlm)
Función de desempeño	Error cuadrático medio

Tabla 5.17 Resultados entrenamiento con una capa oculta. Modelo final

Numero de neuronas	Tiempo [s]	MSE entrenamiento	MSE validación
10	64,7	1,8053E-04	4,4359E-04
20	52,4	6,4906E-05	3,1320E-04
30	154,6	3,5298E-05	2,2920E-04
40	169,7	9,0347E-05	5,6883E-04
50	311,2	2,1783E-05	3,8764E-04
60	294,5	5,8344E-05	3,5992E-04
70	372,5	1,7162E-05	3,5333E-04
80	843,4	1,3819E-05	1,8778E-04
90	674,4	1,2211E-05	1,8582E-04
100	628,0	3,5622E-05	2,2108E-04
110	1090,7	4,3000E-03	4,3000E-03
120	1275,6	1,0301E-04	4,5641E-04

Tabla 5.18 Resultados entrenamiento con dos capas ocultas. Modelo final

Numero de neuronas	Tiempo [s]	MSE entrenamiento	MSE validación
10	64,9	5,7131E-05	5,1713E-04
20	152,9	4,3000E-03	4,4000E-03
30	451,2	2,2976E-06	1,8061E-04
40	493,2	9,8878E-06	2,3486E-04
50	1477,6	2,2271E-05	2,4484E-04

Se observa que los mejores resultados se presentan utilizando una capa oculta con 90 neuronas y con dos capas ocultas de 30 neuronas, aunque la red con dos capas ocultas tiene menor tiempo de entrenamiento. La selección incluyó la revisión de los indicadores DME y FAE, donde se busca la red con mejor confiabilidad y cuya distribución sea exponencialmente decreciente con el porcentaje de daño en la subestructura, como se muestra en la Figura 5.1.

Los resultados para los indicadores son los que se muestran los gráficos y tabla siguiente. Estos resultados se obtuvieron sobre el set de datos de validación.

Tabla 5.19 Valores obtenidos para las redes seleccionadas

	EME
Red una capa oculta 90 neuronas (red 90)	0,0016
Red dos capas ocultas 30 neuronas (red 3030)	0,0018

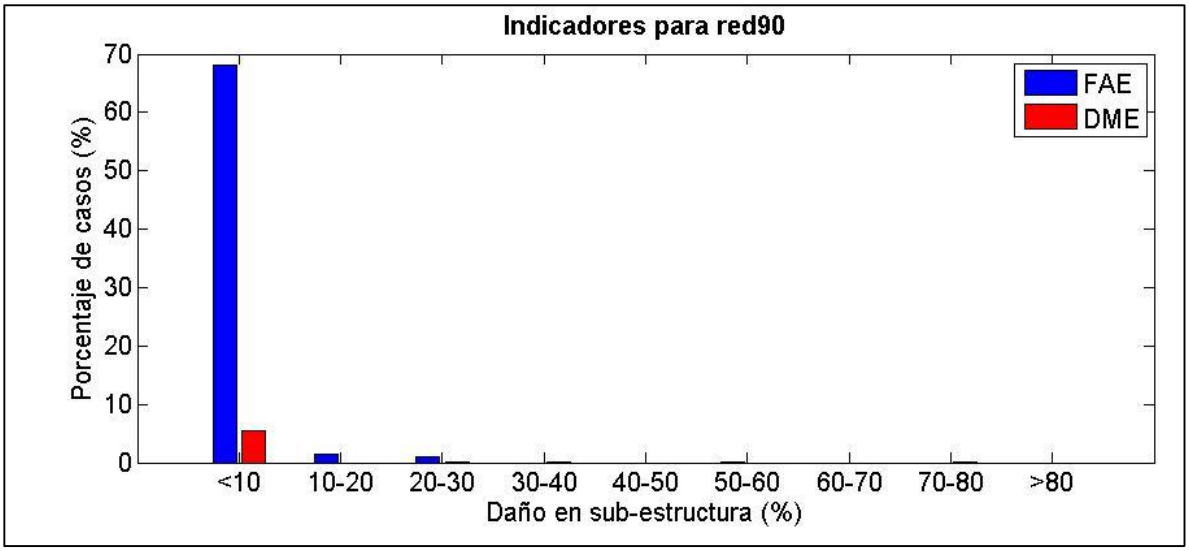


Figura 5.1 Indicadores DME y FAE para red con una capa oculta con 90 neuronas

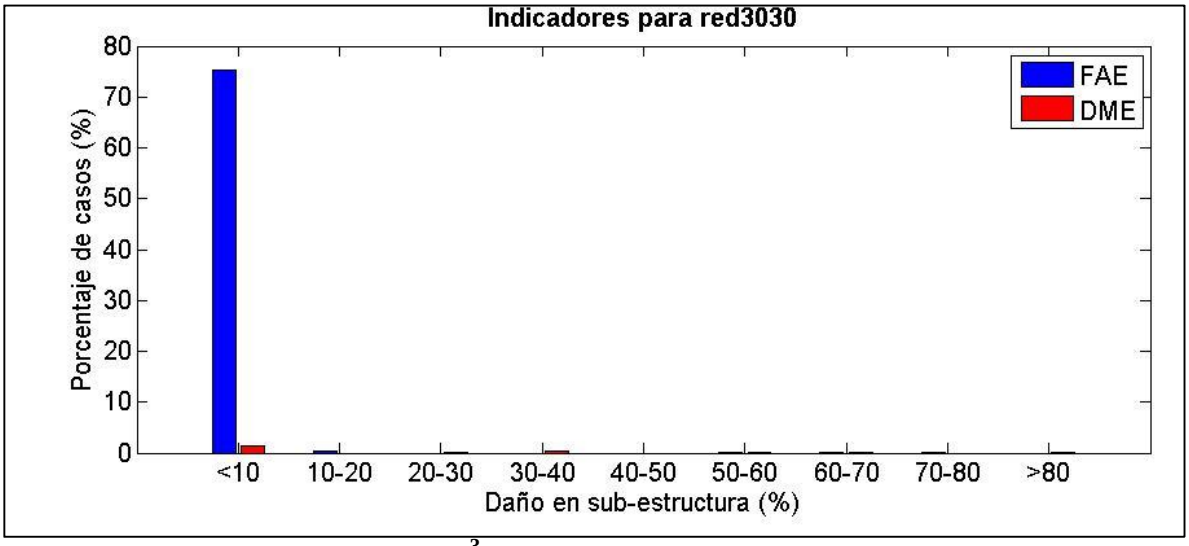


Figura 5.2 Indicadores DME y FAE[3] para red con dos capas ocultas con 30 neuronas cada una

Los resultados nos muestran que para ambas redes los daños sobre el 10% presentan menos del 5% de probabilidad de no ser detectados. También se muestra que la mayoría, es decir sobre el 70% de los daños detectados por la red con un nivel de daño inferior al 10% son daños que no existen realmente. Es decir, la red es confiable para daños sobre el 10% de daño en la sub-estructura.

El indicador más importante en este caso es el DME, ya que lo primordial de la red es detectar sin omisión los daños presentes, por lo que la red con dos capas ocultas de 30 neuronas es la elegida.

La red seleccionada finalmente tiene las siguientes características:

Tabla 5.20 Parámetros primera red seleccionada

Parámetros	Valor
N° de capas ocultas	2
Función de transferencia capa oculta	Sigmoidea logarítmica (logsig)
Función de salida	Lineal saturada simétrica (satlins)
N° de neuronas de capa de entrada	34
N° de neuronas de capa oculta	30-30
N° de neuronas de capa de salida	6
Método de entrenamiento	Levenberg-Marquardt (trainlm)
Función de desempeño	Error cuadrático medio

[3] Ver sección 2.7 para definiciones de DME y FAE

5.2 Entrenamiento con datos simulados para segunda red

La segunda red es la encargada de localizar y cuantificar el daño en cada subestructura identificada por la primera red. En este trabajo se habla singularmente de la segunda red, sin embargo lo que se hace es desarrollar 6 redes neuronales cada una entrenada para una subestructura.

En el análisis a continuación los datos utilizados para el entrenamiento y validación de las redes no fueron contaminados con ruido, a menos que se especifique lo contrario.

Para el entrenamiento de cada red se utiliza como base el trabajo desarrollo por Mahu [1] para identificación de daño en una viga, los parámetros y resultados se muestran en Tabla 5.21 y Tabla 5.22 , respectivamente.

Tabla 5.21 Parámetros de segunda red para prueba preliminar de desempeño

Parámetros	Valor
N° de capas ocultas	2
Función de transferencia capa oculta	Sigmoidea logarítmica (logsig)
Función de salida	Lineal Saturada simétrica (satlins)
N° de neuronas de capa de entrada	15-20 (depende de la subestructura)
N° de neuronas de capa oculta	30-60
N° de neuronas de capa de salida	3-4 (depende de la subestructura)
Método de entrenamiento	Levenberg-Marquardt (trainlm)
Función de desempeño	Error cuadrático medio

Tabla 5.22 Resultados preliminares segunda red

Subestructura	MSE entrenamiento	MSE validación
1	2,6625E-04	0,0067
2	5,2277 E-05	0,0085
3	2,4252E-04	0,0153
4	0,0014	0,0226
5	1,485E-05	0,0125
6	0,0021	0,0159
7	2,9181E-05	0,008
8	9,1765E-05	0,0119

Las entradas de cada red corresponden a las frecuencias de anti-resonancia encontradas en los nodos pertenecientes a cada subestructura y las salidas corresponden al nivel de daño de cada elemento de la subestructura.

Los datos utilizados para cada red corresponden a los mostrados en la Tabla 4.5.

5.2.1 Primera subestructura

Utilizando los parámetros mostrados en la Tabla 5.23 se obtienen los resultados de la Tabla 5.24.

Tabla 5.23 Parámetros de segunda red, subestructura 1

Parámetros	Valor
N° de capas ocultas	1
Función de transferencia capa oculta	Sigmoidea logarítmica (logsig)
Función de salida	Lineal Saturada simétrica (satlins)
N° de neuronas de capa de entrada	12
N° de neuronas de capa oculta	10-130
N° de neuronas de capa de salida	3
Método de entrenamiento	Levenberg-Marquardt (trainlm)
Función de desempeño	Error cuadrático medio

Tabla 5.24 Resultados de segunda red, subestructura 1

Numero de neuronas	Tiempo [s]	MSE entrenamiento	MSE validación
10	156,9	0,0062	0,0062
20	653,9	0,0032	0,0034
30	613,8	0,0064	0,0068
40	880,0	0,0011	0,0026
50	856,3	0,0038	0,0049
60	1723,7	0,0038	0,0045
70	1782,8	0,0093	0,0100
80	2325,2	0,0042	0,0055
90	3370,3	0,0065	0,0077
100	3718,4	0,0036	0,0047
110	4496,5	0,1449	0,1458
120	5838,9	0,0009	0,0024
130	7513,4	0,0707	0,0718

Se observa que los mejores valores se obtienen para 40 y 120 neuronas.

La Tabla 5.25 muestra los resultados del entrenamiento agregando las frecuencias de resonancia como entradas de la red. Se observa que estos datos mejoran el entrenamiento de la red, siendo el mejor resultado el obtenido con 20 neuronas.

Tabla 5.25 Resultados de segunda red, subestructura 1, agregando frecuencias de resonancia

Numero de neuronas	Tiempo [s]	MSE entrenamiento	MSE validación
10	154,2	3,1000E-03	3,2000E-03
20	346,0	1,7335E-04	3,8742E-04
30	500,1	2,6779E-04	5,2698E-04
40	1179,2	2,3466E-04	6,1987E-04
50	1392,8	3,0000E-03	3,1000E-03
60	1850,1	2,0014E-04	5,1962E-04
70	1349,5	6,1470E-04	1,1000E-03
80	2166,2	5,9000E-03	6,1000E-03
90	2702,6	3,1000E-03	3,4000E-03
100	4153,1	5,9000E-03	6,0000E-03

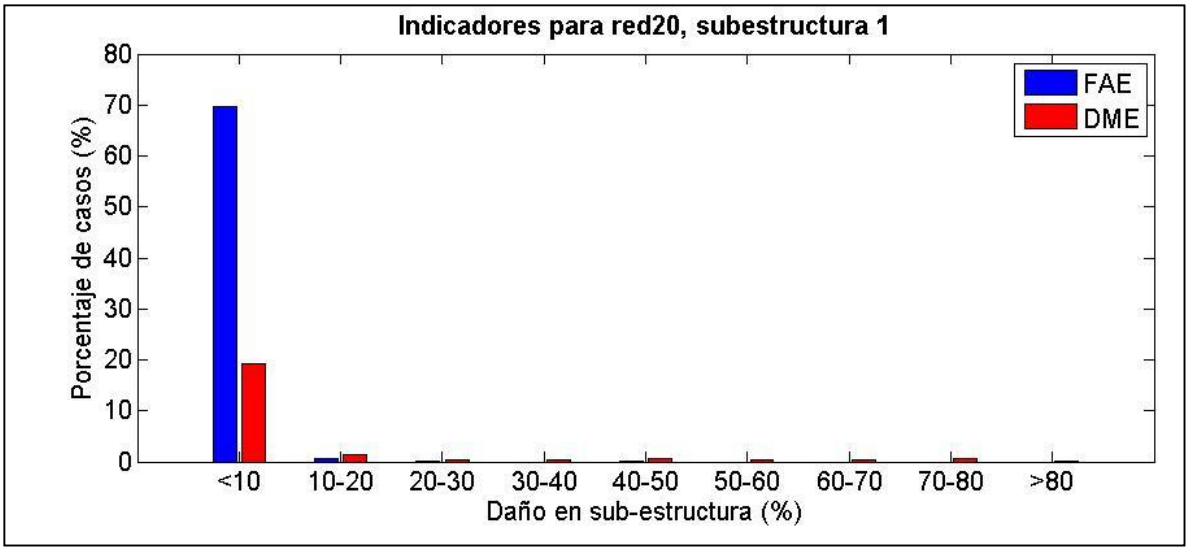


Figura 5.3 Indicadores red con 20 neuronas, subestructura 1

En la Figura 5.3 se observa los indicadores de la mejor red donde la probabilidad de no detección de daño es menor al 2% para daños sobre el 10% de severidad y menor al 20% para daños menores a este. Además, el porcentaje de daños falsos es menor al 1% para daños sobre el 10% de severidad, sin embargo en este caso los daños detectados bajo el nivel de 10% de severidad tienen un 70% de probabilidades de ser daños falsos. El EME es de 0,0028.

En resumen, la red es confiable para daños sobre el 10% en la sub-estructura.

5.2.2 Segunda subestructura

En la Tabla 5.27 se muestran los resultados obtenidos de acuerdo a la Tabla 5.26. El mejor valor de entrenamiento se obtiene con 70 neuronas, mientras que el menor error de validación se obtiene con 50 neuronas.

Tabla 5.26 Parámetros de segunda red, subestructura 2

Parámetros	Valor
N° de capas ocultas	1
Función de transferencia capa oculta	Sigmoidea logarítmica (logsig)
Función de salida	Lineal Saturada simétrica (satlins)
N° de neuronas de capa de entrada	17
N° de neuronas de capa oculta	10-130
N° de neuronas de capa de salida	3
Método de entrenamiento	Levenberg-Marquardt (trainlm)
Función de desempeño	Error cuadrático medio

Tabla 5.27 Resultados de segunda red, subestructura 2

Numero de neuronas	Tiempo [s]	MSE entrenamiento	MSE validación
10	430,0	4,7252E-05	3,4517E-05
20	1433,1	1,0122E-05	1,3694E-05
30	1501,2	3,8226E-06	6,2925E-06
40	3502,5	2,1090E-06	4,8517E-06
50	3521,8	1,3183E-05	7,5189E-06
60	6634,9	2,8789E-06	7,9230E-06
70	1197,3	7,8505E-07	3,3885E-06
80	8908,4	2,4647E-06	8,2017E-06

Tabla 5.28 Resultados de segunda red, subestructura 2, agregando las frecuencias de resonancia

Numero de neuronas	Tiempo [s]	MSE entrenamiento	MSE validación
10	588,9	1,4470E-05	1,4866E-05
20	946,3	7,2021E-06	9,7961E-06
30	2549,2	2,1146E-05	4,2589E-06
40	5669,0	5,7000E-03	5,5000E-03
50	20125,0	6,7255E-08	4,5432E-07
60	6394,1	3,2466E-07	1,8462E-06
70	18442,0	1,4025E-07	6,8762E-07

A continuación se presentan los indicadores de validación obtenidos con datos simulados.

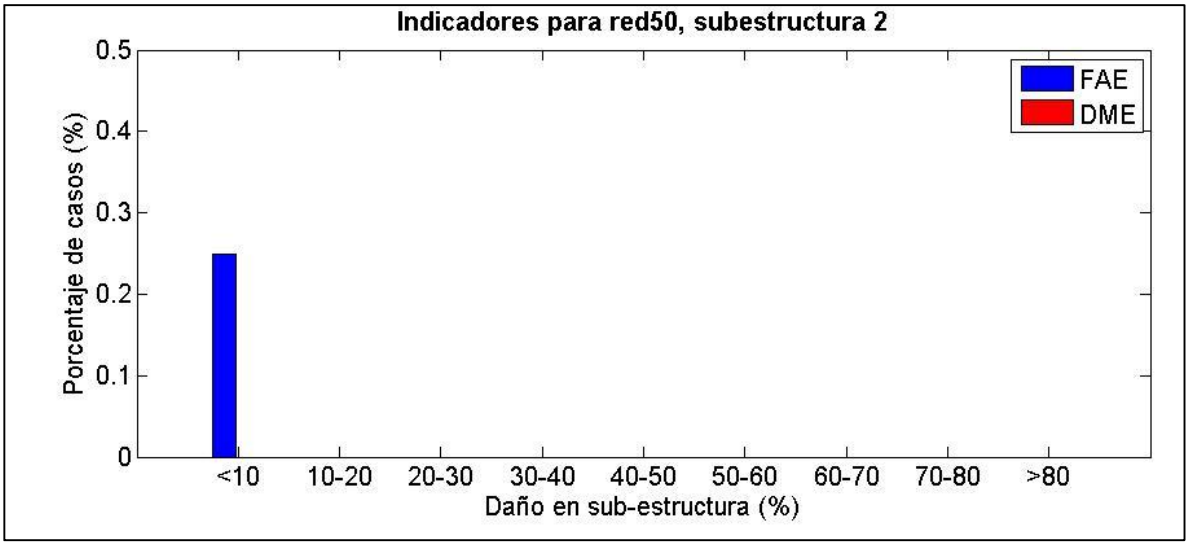


Figura 5.4 Indicadores red con 50 neuronas, subestructura 2

Considerando un EME de 1,2094E-04, la Figura 5.4 muestra que la red es capaz de detectar todos los daños presentados sin importar el nivel y sólo presenta un 0.25% de probabilidad de detectar un daño falso si este valor está bajo el 10% de daño.

5.2.3 Tercera subestructura

Al igual que en los casos anteriores, se realizó primero una prueba sólo con las frecuencias de anti-resonancia, contenido en la Tabla 5.30 y luego se realizó una segunda prueba agregando las frecuencias de resonancia. El valor mínimo obtenido para el error de validación se presenta con 20 neuronas.

Tabla 5.29 Parámetros de segunda red, subestructura 3

Parámetros	Valor
N° de capas ocultas	1
Función de transferencia capa oculta	Sigmoidea logarítmica (logsig)
Función de salida	Lineal Saturada simétrica (satlins)
N° de neuronas de capa de entrada	9
N° de neuronas de capa oculta	10-130
N° de neuronas de capa de salida	3
Método de entrenamiento	Levenberg-Marquardt (trainlm)
Función de desempeño	Error cuadrático medio

Tabla 5.30 Resultados de segunda red, subestructura 3

Numero de neuronas	Tiempo [s]	MSE entrenamiento	MSE validación
10	181,4	0,0126	0,0131
20	377,5	0,0116	0,0127
30	674,6	0,0156	0,0165
40	949,0	0,0150	0,0165
50	742,7	0,0136	0,0149
60	1159,1	0,0808	0,0832
70	2021,2	0,0153	0,0179
80	2499,4	0,0126	0,0141
90	7540,2	0,0164	0,0171
100	2608,8	0,0213	0,0229
110	5608,0	0,0133	0,0153
120	7718,4	0,0108	0,0131
130	4515,7	0,0173	0,0189

Tabla 5.31 Resultados de segunda red, subestructura 3, agregando las frecuencias de resonancia

Numero de neuronas	Tiempo [s]	MSE entrenamiento	MSE validación
10	333,2	0,0012	0,0018
20	1267,8	0,0062	0,0065
30	1707,0	0,0035	0,0046
40	2191,6	0,0005	0,0014
50	2260,9	0,0089	0,0092
60	2075,2	0,0813	0,0813
70	4511,3	0,0785	0,0788
80	8412,2	0,0724	0,0728
90	3581,6	0,0005	0,0018
100	7538,9	0,0036	0,0043

Al igual que las redes precedentes, agregar las frecuencias de resonancia como entradas de la red mejora el entrenamiento, en este caso en un orden de magnitud. En la Tabla 5.31 se observa el menor error de validación utilizando 40 neuronas, siendo por lo tanto la mejor red de las estudiadas.

Se observa en la Figura 5.5 que los daños con nivel mayor al 10% tienen una probabilidad menor al 5% de no ser detectados y los daños menores al 10% una probabilidad del 25,73% de no ser detectados. Por otro lado, los daños detectados sobre el 10% de daño tienen una probabilidad menor al 2% de ser daños falsos, aunque los daños detectados bajo el 10% tienen una probabilidad del 64,13% de ser falsos. El valor del EME es del 0,0049.

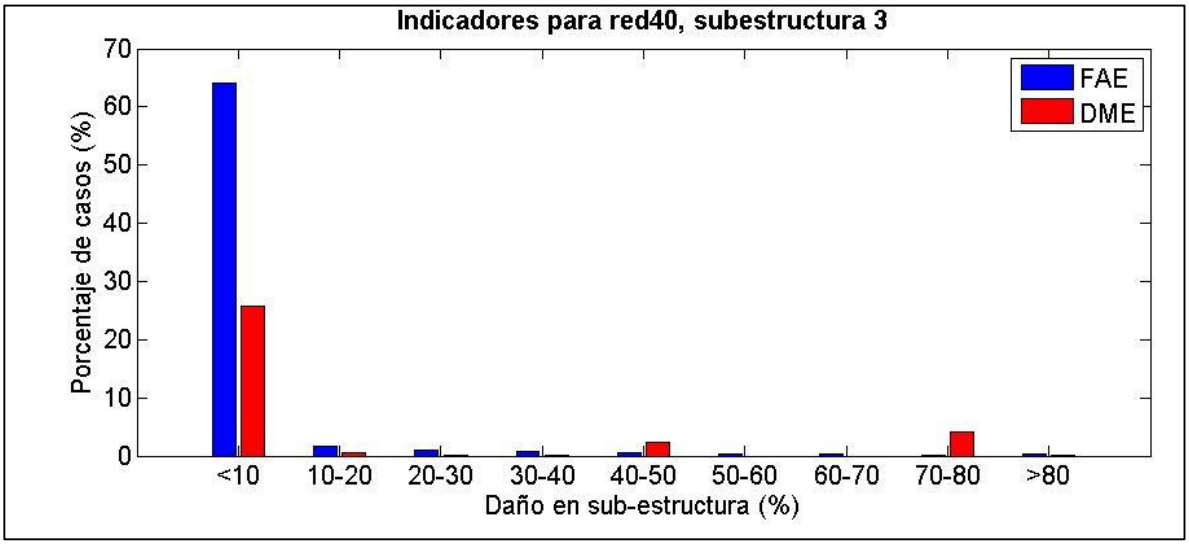


Figura 5.5 Indicadores red con 40 neuronas, subestructura 3

5.2.4 Cuarta subestructura

Al igual que en los casos anteriores, se realizó primero una prueba sólo con las frecuencias de resonancia, contenido en la Tabla 5.33 y luego se realizó una segunda prueba agregando las frecuencias de resonancia. En este caso el mejor resultado se obtiene con 40 neuronas. En la Tabla 5.34 se aprecia que el mejor resultado es el obtenido con 50 neuronas, por lo que se elige como la red final.

Tabla 5.32 Parámetros de segunda red, subestructura 4

Parámetros	Valor
N° de capas ocultas	1
Función de transferencia capa oculta	Sigmoidea logarítmica (logsig)
Función de salida	Lineal Saturada simétrica (satlins)
N° de neuronas de capa de entrada	12
N° de neuronas de capa oculta	10-130
N° de neuronas de capa de salida	3
Método de entrenamiento	Levenberg-Marquardt (trainlm)
Función de desempeño	Error cuadrático medio

Tabla 5.33 Resultados de segunda red, subestructura 4

Numero de neuronas	Tiempo [s]	MSE entrenamiento	MSE validación
10	334,8	9,6000E-03	9,2000E-03
20	1051,8	8,7082E-04	9,2633E-04
30	1532,6	8,9000E-03	9,7000E-03
40	2088,4	5,9414E-04	6,7711E-04
50	2431,7	6,4003E-04	7,5397E-04
60	3343,4	5,8468E-04	7,3062E-04
70	2878,2	5,4071E-04	6,9561E-04
80	5228,0	7,0000E-03	7,0000E-03
90	3689,3	6,3687E-04	7,8874E-04
100	6157,5	6,0854E-04	8,1182E-04
110	6314,1	6,7086E-04	7,4290E-04

Tabla 5.34 Resultados de segunda red, subestructura 4 agregando frecuencias de resonancia

Numero de neuronas	Tiempo [s]	MSE entrenamiento	MSE validación
10	231,7	1,1000E-03	1,4000E-03
20	361,1	6,2750E-04	9,3794E-04
30	651,6	3,4747E-04	7,4485E-04
40	873,1	2,9427E-04	7,1742E-04
50	1608,6	1,7231E-04	6,2076E-04
60	2030,6	1,6649E-04	6,8044E-04
70	2000,2	2,7092E-04	8,9178E-04
80	2636,3	2,3404E-04	8,0846E-04
90	2193,2	1,9130E-04	8,5339E-04
100	3298,7	2,5371E-04	8,3663E-04

De la Figura 5.6 se extrae que los daños sobre el 10% de daño tienen una probabilidad de menos del 4% de no ser detectados, mientras que bajo ese nivel la probabilidad de no ser detectados crece al 48.77%. A su vez, lo daños detectados con una severidad mayor al 10% tienen una probabilidad menor al 14% de ser falsas alarmas de daño, sin embargo los daños detectados de severidad menor al 10% tienen el 72,71% de probabilidad de ser falsa alarma de daño. El valor del EME es de 0.0043.

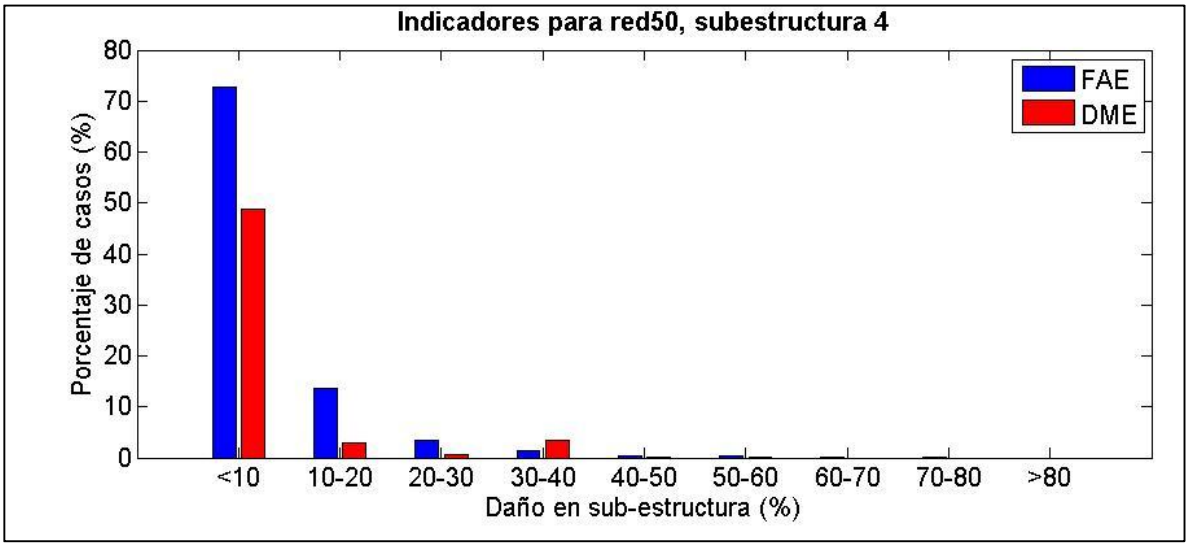


Figura 5.6 Indicadores red con 50 neuronas, subestructura 4

5.2.5 Quinta subestructura

Al igual que en los casos anteriores, se realizó primero una prueba sólo con las frecuencias de anti-resonancia, contenido en la Tabla 5.36 y luego se realizó una segunda prueba agregando las frecuencias de resonancia, los resultados se muestran en la Tabla 5.37.

Comparando los valores de MSE de validación entre la Tabla 5.36 y la Tabla 5.37 se ve una mejora de un orden de magnitud, por lo que extrayendo el mejor valor de esta última, se elige a la red con 40 neuronas como red final.

Tabla 5.35 Parámetros de segunda red, subestructura 5

Parámetros	Valor
N° de capas ocultas	1
Función de transferencia capa oculta	Sigmoidea logarítmica (logsig)
Función de salida	Lineal Saturada simétrica (satlins)
N° de neuronas de capa de entrada	7
N° de neuronas de capa oculta	10-130
N° de neuronas de capa de salida	3
Método de entrenamiento	Levenberg-Marquardt (trainlm)
Función de desempeño	Error cuadrático medio

Tabla 5.36 Resultados de segunda red, subestructura 5

Numero de neuronas	Tiempo [s]	MSE entrenamiento	MSE validación
10	352,9	0,0048	0,0046
20	496,8	0,0033	0,0035
30	704,9145	0,0092	0,0096
40	875,2998	0,0091	0,01
50	1313,4	0,0759	0,0768
60	1612,5	0,0075	0,0087
70	2999,4	0,0065	0,0075
80	2117,8	0,0079	0,0092
90	2042,3	0,0044	0,0058
100	2138,9	0,0858	0,0861
110	4360,5	0,0111	0,0119
120	3219,9	0,0067	0,0076
130	4507,9	0,0101	0,011

Tabla 5.37 Resultados de segunda red, subestructura 5 agregando frecuencias de resonancia

Numero de neuronas	Tiempo [s]	MSE entrenamiento	MSE validación
10	295,163	2,8000E-03	0,0026
20	370,4903	1,7000E-03	0,0018
30	648,1153	1,0000E-03	0,0012
40	811,3473	7,3661E-04	0,0010
50	1091,4	7,2576E-04	0,0012
60	1917,8	6,2690E-04	0,0010
70	2127,5	6,2265E-04	0,0011
80	2068,6	5,1019E-04	0,0012
90	2355,5	4,5098E-04	0,0011
100	2541,1	4,7773E-04	0,0011

De la Figura 5.7 se puede observar que los daños con nivel sobre el 20% de severidad tienen una probabilidad del 4,62% de ser detectados, mientras que estando entre el 10%y 20% de severidad la probabilidad de no ser detectados aumenta al 23,63% y siendo los daños menores al 10% la probabilidad de no ser detectados por la red aumenta al 87,95%. Por otra parte, los daños detectados con un nivel de daño menor al 10% tienen el 78,39 de probabilidad de ser falsas alarmas, valor que disminuye a 25,16% cuando el daño está entre el 10% y el 20%, mientras que para valores de daño mayor al 20% la probabilidad de ser falsas alarmas se encuentra bajo el 9%. El EME en este caso es de 0,0072.

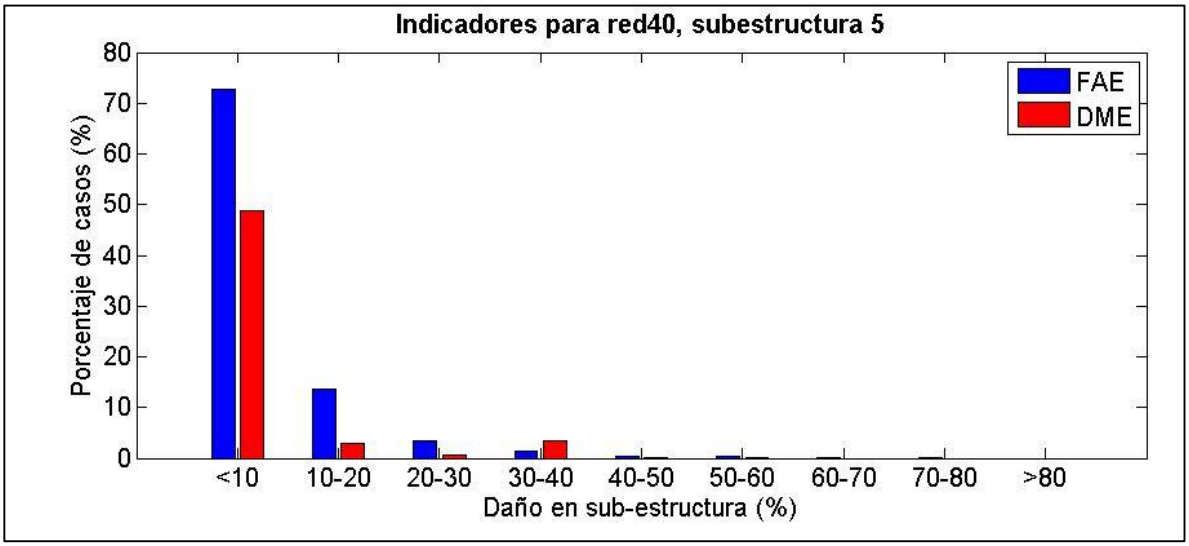


Figura 5.7 Indicadores red con 40 neuronas, subestructura 5

5.2.6 Sexta subestructura

De acuerdo a la Tabla 5.38 se obtienen los resultados de la Tabla 5.39. Como se observa en la tabla el mejor valor se obtiene con 10 neuronas. En la Tabla 5.40 se observa que el mejor valor del MSE de validación de obtiene con 50 neuronas, por lo que esta será la red final para esta subestructura.

Tabla 5.38 Parámetros de segunda red, subestructura 6

Parámetros	Valor
N° de capas ocultas	1
Función de transferencia capa oculta	Sigmoidea logarítmica (logsig)
Función de salida	Lineal Saturada simétrica (satlins)
N° de neuronas de capa de entrada	6
N° de neuronas de capa oculta	10-130
N° de neuronas de capa de salida	3
Método de entrenamiento	Levenberg-Marquardt (trainlm)
Función de desempeño	Error cuadrático medio

Tabla 5.39 Resultados de segunda red, subestructura 6

Numero de neuronas	Tiempo [s]	MSE entrenamiento	MSE validación
10	285,853	0,0157	0,0163
20	665,6913	0,0159	0,0171
30	788,8175	0,0228	0,023
40	793,3775	0,0148	0,0174
50	1289	0,0188	0,0202
60	1691,5	0,0192	0,0203
70	1947,7	0,022	0,0224
80	1581	0,0235	0,0243
90	3276,2	0,0175	0,0195
100	2939,6	0,0155	0,0175
110	3548,1	0,0217	0,0227
120	7584,6	0,0192	0,0203

Tabla 5.40 Resultados de segunda red, subestructura 6, agregando frecuencias de resonancia

Numero de neuronas	Tiempo [s]	MSE entrenamiento	MSE validación
10	220,9823	0,0033	0,0031
20	495,4925	0,0042	0,0042
30	694,4095	0,001	0,0014
40	1104	0,0008048	0,0013
50	1280,2	0,00054446	0,001
60	709,731	0,0033	0,0037
70	1676,6	0,0036	0,0042
80	1574,2	0,0063	0,0066
90	3470,3	0,0032	0,0036

En la Figura 5.8 se observa que para daños sobre el 20% de severidad la probabilidad de que la red no los detecte es menor al 7%, mientras que si se encuentran entre el 10% y el 20% la probabilidad de no ser detectados aumenta al 23,86%, llegando al 71,87% si los daños son de severidad menor al 10%. Similarmente en el caso de detección de falsas alarmas, para daños sobre el 20% la probabilidad de detección falsa es menor al 10%, lo que aumenta a 27,37% si los daños se encuentran entre el 10% y el 20% y llega al 77,09% para daños menores al 10%. El EME en este caso fue de 0,0088.

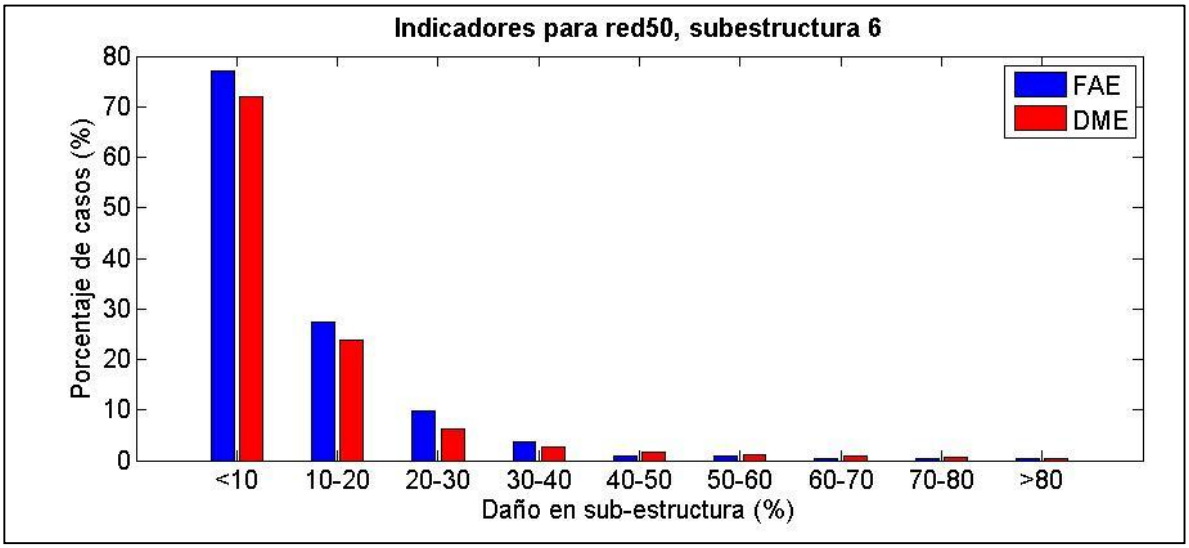


Figura 5.8 Indicadores red con 50 neuronas, subestructura 6

5.3 Resultados experimentales

A continuación se presentan los resultados obtenidos para los valores experimentales utilizando las redes entrenadas y seleccionadas en las secciones anteriores de este capítulo.

5.3.1 Resultados en primera red

En la Tabla 5.41 se muestran los resultados que entregó la red determinada la mejor en la sección 5.1.2. Los números marcados en rojo muestran en cual subestructura se localiza efectivamente el daño.

Debido a los resultados observados en la sección 5.1.2, los valores detectados menores al 10% de daño fueron considerados como falsos positivos.

Tabla 5.41 Resultados con datos experimentales red 3030

Subestructura	Daño grande 9	Daño grande 12	Daño pequeño 3	Daño pequeño 3 y grande 9
1	0,24	0,59	0,00	0,42
2	0,95	0,00	0,00	0,53
3	0,00	0,95	0,00	0,00
4	0,00	0,25	0,00	0,00
5	0,43	0,00	0,50	0,00
6	0,00	0,00	0,95	0,00

Se puede ver que la red detecta la mayoría de los daños. Se observa que puede detectar los dos casos de daños grandes únicos, sin embargo no detecta los daños pequeños. En el caso de dos daños, se detectan correctamente ambos daños.

5.3.1.1 Análisis de ruido

Utilizando la red definida en la sección 5.1.2 , se realizará el análisis de ruido agregando diferentes niveles de ruido aleatorio a los datos de entrenamiento de la red para ver su inferencia en la detección de daño de los casos estudiados (Ver sección 3.1.1).

Tabla 5.42 Resultados primera red con ruido 0.1

Subestructura	Daño grande 9	Daño grande 12	Daño pequeño 3	Daño pequeño 3 y grande 9
1	0,29	0,023	0	0,32
2	0,66	0	0,24	0,32
3	0	0	0	0
4	0	0	0	0,03
5	0,23	0	0,40	0,03
6	0	0,77	0,82	0,03

Tabla 5.43 Resultados primera red con ruido 0.25

Subestructura	Daño grande 9	Daño grande 12	Daño pequeño 3	Daño pequeño 3 y grande 9
1	0,41	0,01	0	0,27
2	0,21	0	0	0,40
3	0	0	0	0
4	0	0,01	0	0
5	0,14	0	0,31	0,06
6	0	0,73	0,90	0,17

Tabla 5.44 Resultados primera red con ruido 0.5

Subestructura	Daño grande 9	Daño grande 12	Daño pequeño 3	Daño pequeño 3 y grande 9
1	0,36	0	0	0,26
2	0,15	0	0	0,12
3	0	0	0	0
4	0	0,06	0	0
5	0,18	0,17	0,49	0,02
6	0	0,48	0,67	0,29

Tabla 5.45 Resultados primera red con ruido 1.0

Subestructura	Daño grande 9	Daño grande 12	Daño pequeño 3	Daño pequeño 3 y grande 9
1	0,39	0	0	0,32
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0
5	0,02	0,07	0,41	0
6	0	0,63	0,65	0,33

Tabla 5.46 Resultados primera red con ruido 1.5

Subestructura	Daño grande 9	Daño grande 12	Daño pequeño 3	Daño pequeño 3 y grande 9
1	0,45	0	0	0,32
2	0,05	0	0	0,06
3	0	0	0	0
4	0	0	0	0
5	0,02	0,23	0,35	0
6	0	0,49	0,47	0,03

Tabla 5.47 Resultados primera red con ruido 2.0

Subestructura	Daño grande 9	Daño grande 12	Daño pequeño 3	Daño pequeño 3 y grande 9
1	0,34	0	0	0,16
2	0	0	0	0
3	0,09	0,05	0,09	0,1
4	0	0	0	0
5	0,07	0,06	0,31	0,01
6	0	0,43	0,47	0

Según los resultados encontrados, los resultados de la red adicionando ruido a los datos de entrenamiento no muestran mejoras en la detección de daño, inclusive se puede observar en las tablas anteriores que con un nivel de 0.1 de ruido aun es capaz de detectar algunos daños, sin embargo al ir aumentando el ruido la red tiende a desviarse en todos los casos, detectando daño en otras subestructuras.

Debido a que el ruido introducido es aleatorio, se puede notar que con un nivel mayor a 1.0 los resultados de detección comienzan a desviarse demasiado, lo que podría indicar que el ruido causa variaciones mayores que las que produce el daño introducido experimentalmente.

5.3.2 Resultados en segunda red

En la siguiente sección se muestran los resultados de la detección de daño en los 4 escenarios de daño estudiados, utilizando la red que corresponde a la subestructura donde presenta daño.

El algoritmo principal considera 2 etapas; en la primera etapa (primera red) se determina cual(es) sub-estructura(s) tiene(n) daño, el análisis de la segunda red se realiza solo sobre las redes de las sub-estructuras con daño, ya que su funcionamiento es independiente al de la primera red. Es decir, que las segundas redes se probaron sólo en los lugares de daño real correspondiente.

5.3.2.1 Daño grande en 9

El elemento 9 pertenece a la subestructura 2, por lo que la detección se realiza con la red determinada para esta subestructura.

Tabla 5.48 Resultados detección red subestructura 2, caso daño grande en 9

Elemento	Valor entregado por red subestructura 2
7	0,95
8	0
9	0

La red predice un daño en el elemento opuesto al elemento realmente dañado. Además el nivel de daño que detecta es el mayor al cual fue entrenada la red.

5.3.2.2 Daño grande en 12

El elemento 12 pertenece a la subestructura 3, por lo que se utilizará la red de la subestructura 3 sobre los datos pertinentes de este caso.

Tabla 5.49 Resultados detección red subestructura 3, caso daño grande en 12

Elemento	Valor entregado por red subestructura 3
12	0
13	0
14	0,95

En este caso la red no encontró el daño, pero encontró daño en el elemento 14, el cual es de la mayor magnitud a la que fue entrenada la red e igualmente que en el caso anterior el daño detectado por la red se ubica en el elemento opuesto de la red.

5.3.2.3 Daño pequeño en 3

El elemento 3 pertenece a la subestructura 1, por lo que se utilizará la red de la subestructura 1 sobre los datos pertinentes de este caso.

Tabla 5.50 Resultados red subestructura 1

Elemento	Valor entregado por red subestructura 1
2	0,95
3	0
4	0

En este caso, la red detecta un daño pero lo detecta en el elemento 2 siendo que debería ser el 3. Además, que siendo este un daño pequeño, notar que el daño falso detectado es de 0,95.

5.3.2.4 Daño pequeño en 3 y grande en 9

En este caso es una combinación por lo que hay que utilizar las redes de las subestructuras 1 y 9 cada una con las entradas correspondientes.

Tabla 5.51 Resultados red subestructura 1

Elemento	Valor entregado por red subestructura 1
2	0,95
3	0
4	0,95

6 Discusión

6.1 Respecto al desarrollo de la primera red

Las pruebas mostraron que la función de transferencia que mejor se ajusta a los datos es la función lineal saturada simétrica (‘satlins’). Luego de entrenar la red utilizando esta función la red solo entrega valores entre 0 y el límite superior elegido para el nivel de daño. Esto último, no se consigue con el resto de las funciones estudiadas, esta propiedad la separa de la función de transferencia lineal pura (‘purelin’) la cual presenta errores de validación con valores cercanos.

La modificación del modelo presentado inicialmente donde existía sólo un elemento dañado en cada subestructura, por uno que considera que cada elemento está dañado igualmente en cada subestructura permite mejorar el entrenamiento de la red en 2 órdenes de magnitud, esto se puede observar en los resultados presentados en el modelo final (sección 5.1.2 en comparación a sección 5.1.1.1). Lo anterior significa que el error medio de validación, es decir la diferencia entre los valores reales y los entregados como salida de la red para el set de validación, disminuyó en dos órdenes de magnitud. Aunque el hecho que todos los elementos de una subestructura estén dañadas por igual magnifica la respuesta que detecta la red, ya que cada elemento dañado provoca cambios en las frecuencias de resonancia y de anti-resonancia.

Se añadió ruido a los datos de entrenamiento motivo de simular el ruido experimental de las mediciones, estos según se explicita en la sección 3.1.1. El resultado esperado al utilizar los datos experimentales en las redes entrenadas con estos datos, es que al aumentar el nivel de ruido los valores detectados son cada vez menores (ya que el ruido tiene una distribución uniforme) y que los valores de las detecciones falsas disminuyesen y finalmente desapareciesen, como se presentó en el caso estudiado de una barra por Mahu[1]. Sin embargo, los valores obtenidos con los diferentes niveles de ruido creciente no presentaron la tendencia explicada antes, lo que podría deberse a que el ruido en los datos experimentales no tiene la distribución que se utilizó.

El principal factor para lograr mejoras en el entrenamiento fue la adición de las frecuencias de anti-resonancia en los nodos límites de las subestructuras. Las frecuencias de resonancia por si solas no permiten entrenar a una red que logré resultados aceptables a nivel numérico, ni siquiera variando los otros parámetros que permiten el entrenamiento de la red.

El uso de dos capas ocultas permite un mejor resultado en un menor tiempo. Es así como aunque el error de validación es similar, el tiempo de entrenamiento es mucho menor con dos capas, como se puede ver en la Tabla 6.1.

Tabla 6.1 Resumen resultados numéricos primera red

Tipo de red	Tiempo de entrenamiento	MSE validación
Una capa oculta 90 neuronas	674,4178	1,8587E-04
Dos capas ocultas 30 neuronas	451,209	1,8061E-04

El modelo final entregó una red que presenta un DME con un valor máximo del 1,32% de casos no detectados para daños de un nivel menor al 10%. El FAE en estos casos no supera el 2% para daños sobre el 10%. Este análisis sobre la base de un EME de 0,0018.

El modelo final desarrollado permitió, utilizando los datos experimentales, identificar los dos casos con un daño grande, no así el daño pequeño. En los casos con dos daños en uno de ellos permitió identificar ambos daños pequeño y grande.

6.2 Respecto al diseño y construcción de la estructura experimental

El diseño de la estructura se realizó considerando principalmente el análisis en elementos finitos de las frecuencias de resonancia, de tal manera que éstas se ubiquen en el rango de medición de los instrumentos disponibles.

Además, se prefirió una estructura que se pudiese ensamblar, lo que permite realizar una mayor cantidad de casos con pocos elementos. Esto lleva a una variación importante respecto al modelo en elementos finitos, el cual considera uniones perfectamente sólidas.

La introducción de elementos de unión condujo a considerar masas puntuales (ver sección 4.5.2) en las uniones de la estructura. Por otro lado, la variación en la rigidez y forma conduce a una variación de comportamiento en estas zonas que no permite considerarlas dentro del análisis de daño.

El elemento de viga puede no corresponder a la mejor representación de un perfil de las características estudiadas, por lo que los resultados de detección sobre los casos experimentales se pueden mejorar con un modelo diferente como por ejemplo de placas unidas.

6.3 Respecto al desarrollo de la segunda red

Los resultados analíticos se presentan en la tabla siguiente.

Tabla 6.2 Resumen MSE de validación redes subestructuras

Numero de subestructura	MSE validación
1	3,8742E-04
2	4,5432 E-07
3	1,4E-03
4	6,2076 E-04
5	1,0 E-03
6	1,0 E-03

Las redes 1,2 y 4 logran buenos valores de error de validación lo que se debe principalmente a que estas redes poseen una mayor cantidad de datos de entrada, lo que demostró ser el mayor factor para lograr un buen entrenamiento.

Sin embargo, estas redes no mostraron resultados positivos en la detección de daños experimentales. Concordantemente a lo detectado por la primera red se detectó daño pero en elementos erróneos, excepto en un caso. El ruido aleatorio introducido, al igual que en la primera red no presentó tendencias claras en ninguna de las subestructuras probadas con datos experimentales. De las mediciones se podía notar ruido especialmente en el rango que se dejo fuera del estudio, sin embargo es necesario conocer la distribución de este ruido el cual probablemente no sea aleatorio.

Las redes probadas con datos experimentales fueron 1, 2 y 3, de estas tres redes dos de ellas mostraron un buen entrenamiento que asegura que la red entrega resultados muy ajustados a los valores de entrenamiento. Esto indica que el modelo de viga puede no ser el adecuado para representar al perfil utilizado experimentalmente, lo cual se evidencia en la necesidad de ajustar parámetros de la estructura como espesor, módulo de Young y densidad para lograr mayor similitud de los parámetros modales, entre los casos sin daño numérico y experimental usados en el ajuste de modelo.

Al considerar los resultados obtenidos por Mahu [1] el cual utilizó un modelo con elemento de viga y una red neuronal de características similares a la estudiada, cuyos resultados de detección de casos experimentales fue satisfactoria, es necesario puntualizar que la principal diferencia en este procedimiento es el tipo de elemento experimental, el cual es más afín al elemento de viga, ya que se usó una barra sólida no un perfil y la simplicidad del modelo considerado.

7 Conclusiones

En esta sección se presentan conclusiones sobre los puntos principales y una conclusión general de la metodología.

7.1 Sobre la primera red

El uso exclusivo de las frecuencias de resonancia no fue suficiente para lograr un correcto entrenamiento de la red neuronal para la detección de daño.

El uso de dos capas ocultas permite disminuir el tiempo de entrenamiento de la red con resultados mejores que el caso de una capa oculta. Esto concuerda con los resultados de Cybenko [17].

Se puede entrenar una red que permita identificar subestructuras dañadas, a nivel numérico con daños con confiabilidad del 95%.

Los datos experimentales permiten demostrar que la red detecta correctamente en el caso de un solo daño grande los dos casos presentados y dos daños en el caso con dos daños, siendo uno pequeño y uno grande.

7.2 Sobre la estructura experimental

Las diferencias en las uniones entre elementos respecto al modelo numérico no permiten incluirlas en el análisis de detección de daño. Es necesaria una modelación más detallada de las uniones para poder analizarlas.

7.3 Sobre las segundas redes

Las frecuencias de anti-resonancia obtenidas experimentalmente permitieron entrenar a las redes neuronales correctamente, aunque los mejores resultados se presentaron para las redes cuya cantidad de datos de entrada superó las 12 frecuencias de anti-resonancia (esto unido a las 6 frecuencias de resonancia consideradas).

Las redes validadas con datos experimentales entregaron sólo un resultado correcto, lo cual puede deberse a las diferencias entre el modelo analítico y la estructura real, y a la falta de mejor modelación del ruido experimental en las mediciones.

7.4 Conclusiones generales

El entrenamiento y la validación de la primera red entregó resultados satisfactorios, tanto numérica como experimentalmente, no así las segundas redes cuyos entrenamientos numéricos fueron aceptables en varios casos pero la validación experimental fue insatisfactoria.

Es decir, el uso de un método de sub-estructuración y redes neuronales para la identificación de daño en estructuras de barra, resultó correcto en la etapa de localización de subestructuras dañadas. Sin embargo, para la etapa de la identificación de elementos dañados en cada sub-estructura es necesario asegurar una cantidad mínima de datos de entrada a la red (frecuencias de anti-resonancia) que permita el correcto entrenamiento de la red neuronal.

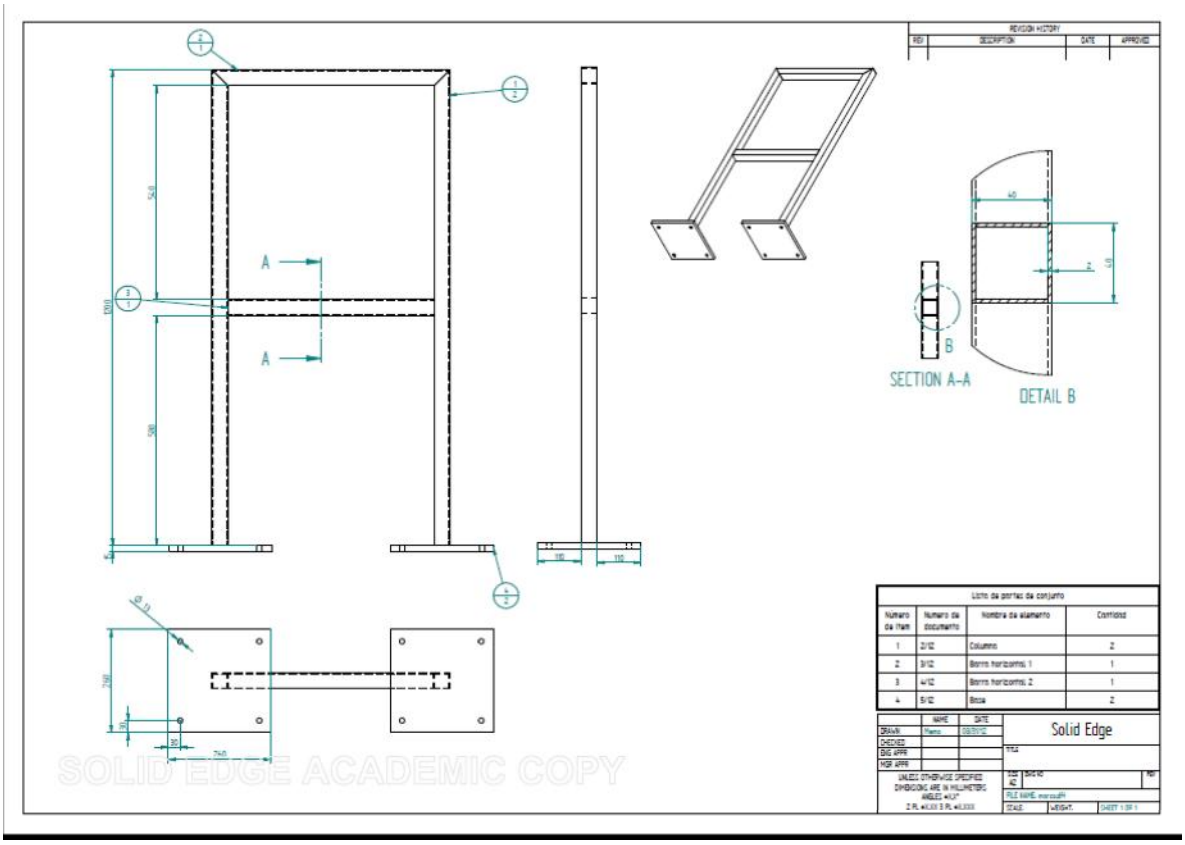
8 Bibliografía

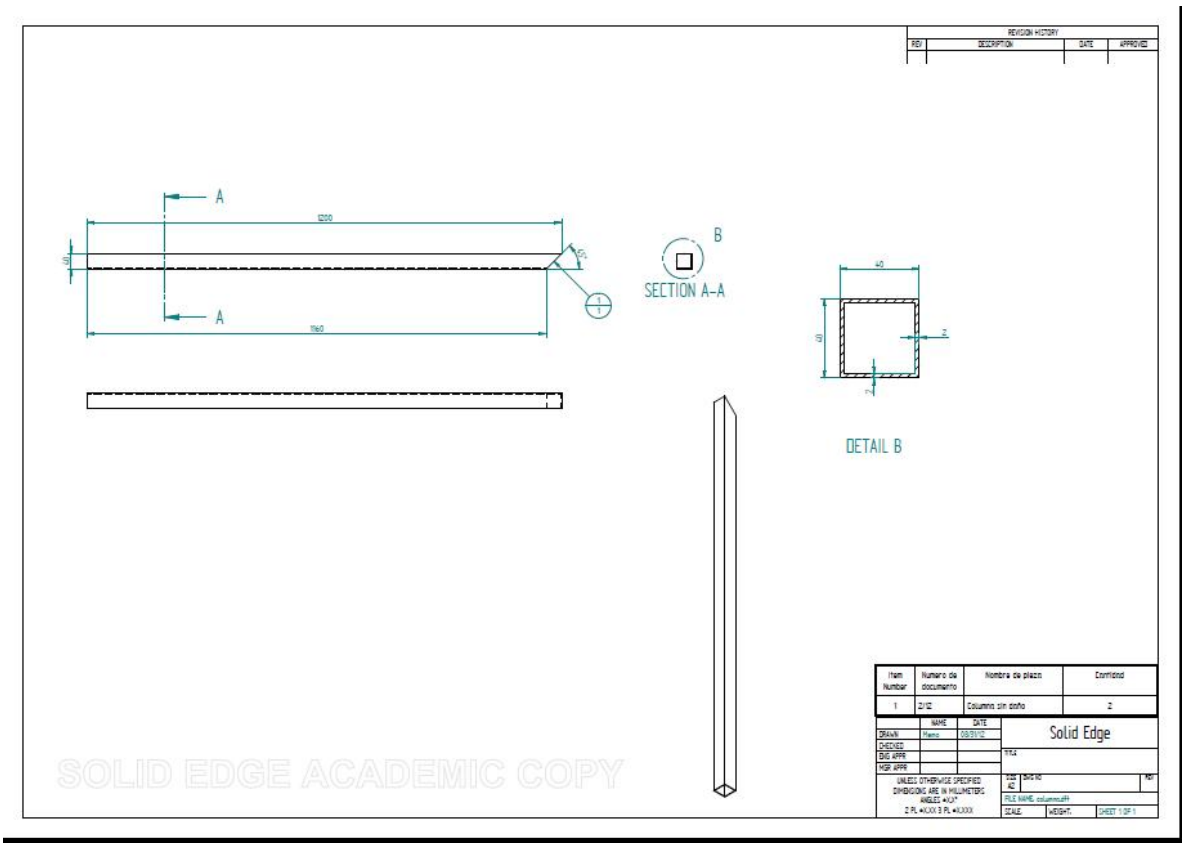
[1] J. Mahu. “Detección de daños en una viga simple mediante antiresonancia y redes neuronales”, Memoria para optar al título de Ingeniero Civil Mecánico, Universidad de Chile, 2012.
[2] B. Sahoo, D. Maity., “Damage assessment of structures using hybrid neuro-genetic algorithm”. Applied Soft computing 7, 2007, páginas 89-104.
[3] C.B. Yun, E.Y. Bahng, “Sub-structural identification using neural network”. Computer and structures 77, 2000, páginas 41-52.
[4] N. Bakhary, H. Hao, A.J.Deeks.,“Structure damage detection using neural network with multi-stage substructuring”. Advances in Structural Engineering Vol. 13 No. 1 2010.
[5] C.B. Yun, J.H. Yi, E.Y. Bahng, “Joint damage assessment of framed structures using a neural networks technique”. Engineering structures 23, 2001, páginas 425-435.
[6] F. Qu, D. Zou, X. Wang, “Substructural damage detection using neural networks and ICA”. ISNN 2004, LNCS 3173, 2004, páginas 750-754.
[7] A. Messina, E.J. Williams, T. Contursi, “Structural damage detection by a sensitivity and statistical-based method”. Journal of Sound and Vibration, 1998, páginas 791-808.
[8] V. Meruane, “Apunte vibraciones mecánicas, ME4701” [En línea] http://viviana.meruane.com/
[9] V. Meruane, “Apunte dinámica estructural, ME706” [En línea] http://viviana.meruane.com/
[10] R. Flórez, J.M. Fernández, “Las redes neuronales artificiales”. Editorial Netbiblo, 2008.
[11] V. Meruane, J. Mahu. Identificación de daño estructural utilizando redes neuronales artificiales y frecuencias de anti-resonancia. XV Congreso Chileno de Ingeniería Mecánica, COCIM 2012. Noviembre 29-30, 2012, La Serena, Chile.
[12] J.J. Lee, J.W. Lee, J.H. Yi, C.B. Yun and H.Y. Jung, “Neural networks-based damage detection for bridges considering errors in baseline finite element models”, Journal of Sound and Vibration, vol. 280, no. 3-5, 2005, páginas 555-578.
[13] K.D. Hjelmstad and S. Shin, “Damage detection and assessment of structures from static response”, Journal of Engineering Mechanics, 1997, vol. 123, no. 6, páginas 568-576.
[14] R. J. Allemang, “The Modal Assurance Criterion- Twenty years of use and abuse”, Sound and Vibration, August 2003.
[15] R. Basri and W.K. Chiu, “Numerical analysis on the interaction of guided Lamb waves with a local elastic stiffness reduction in quasi-isotropic composite plate structures”, Composite structures, Volume 66, Issues 1-4, October- December 2004, páginas 87-99, Twelfth International Conference on Composite Structures.
[16] T-Platforms Holding, “Simulation of elliptical shaped crack propagation in the zone of steel bar welded joint and lay in panel of the top column node point of the sports facility”, From http://www.t-services.ru/en/solutions/modelirovanierazvitiyaellipticheskojtrewiny.html
[17] G. Cybenko. “Approximation by superpositions of a sigmoidal function”. Mathematics of Control, Signals and Systems, 1989, páginas 303-314.

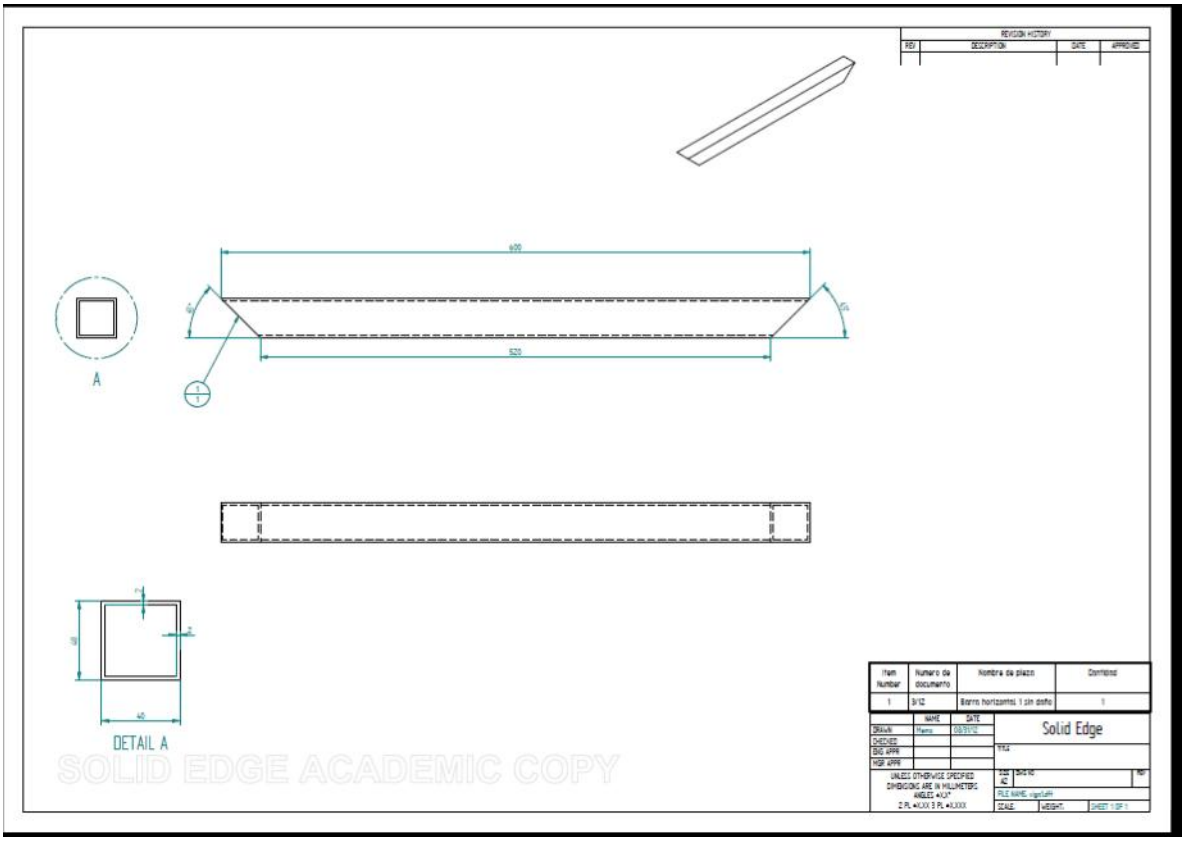
Anexos

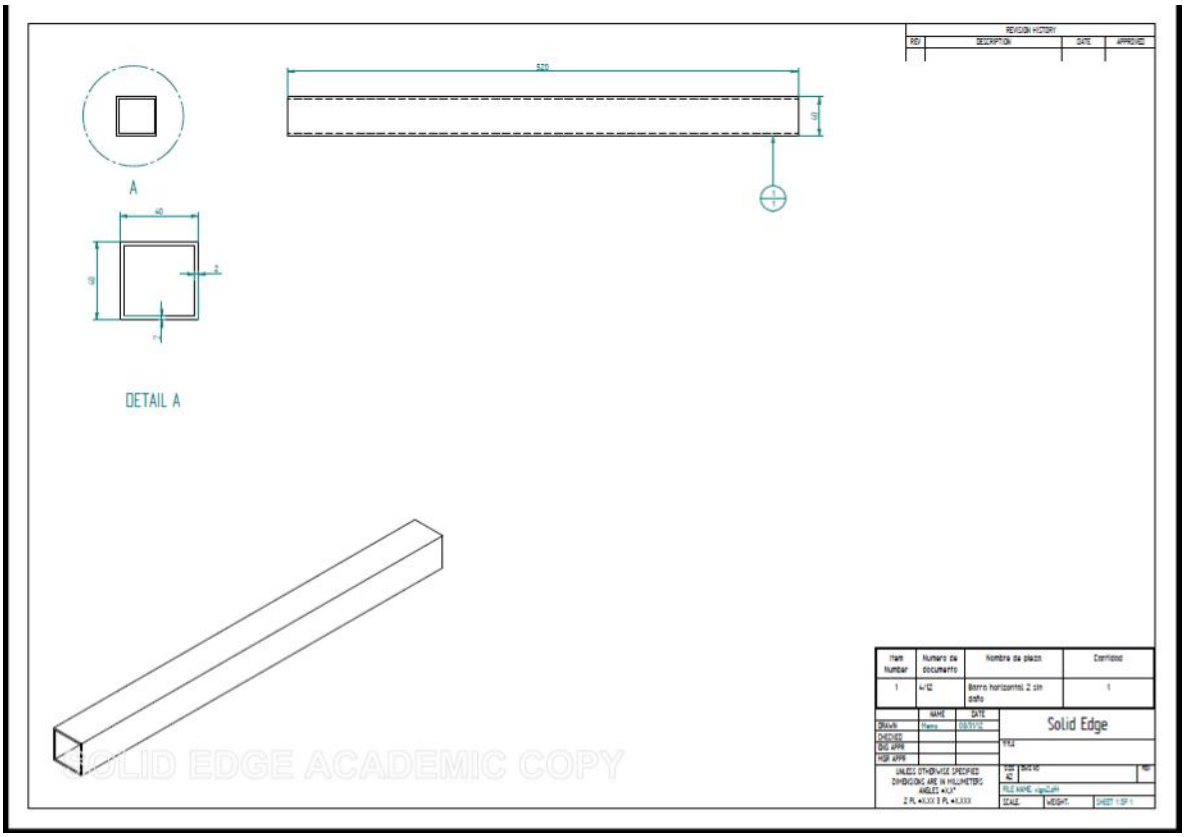
Anexo A

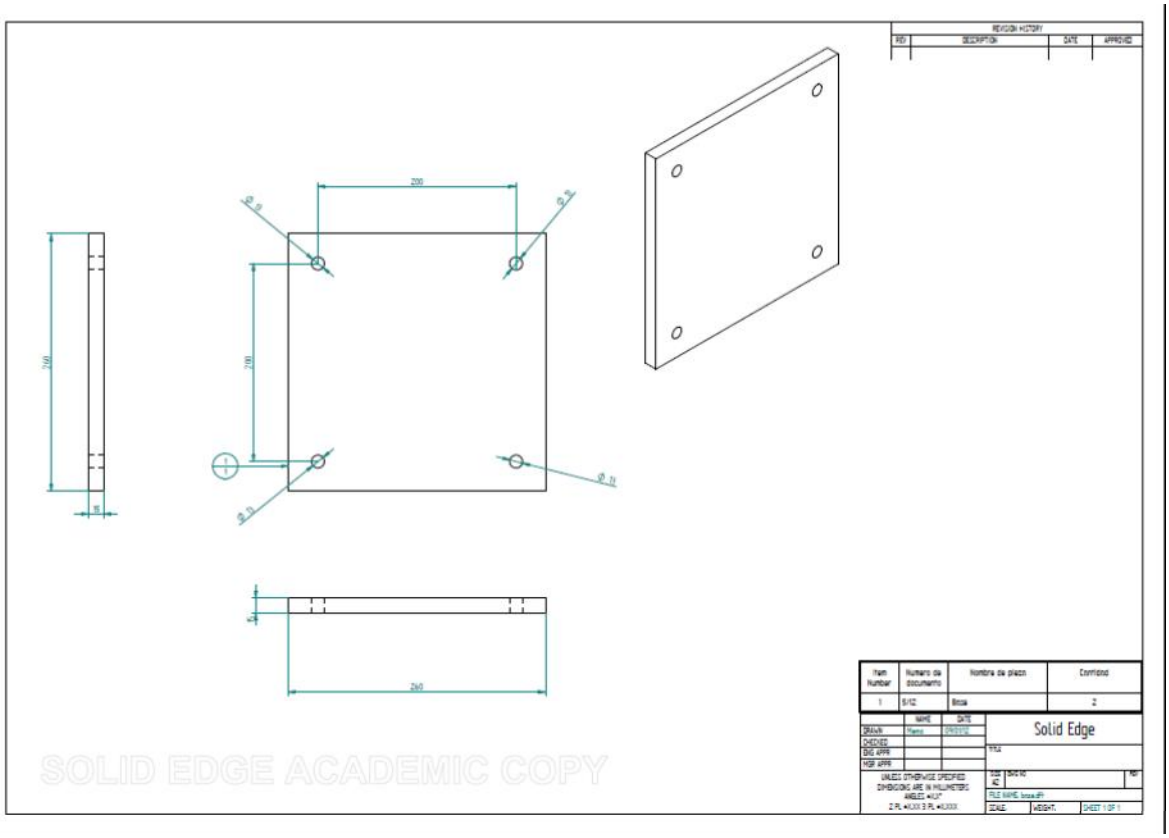
Planos de diseño de estructura experimental

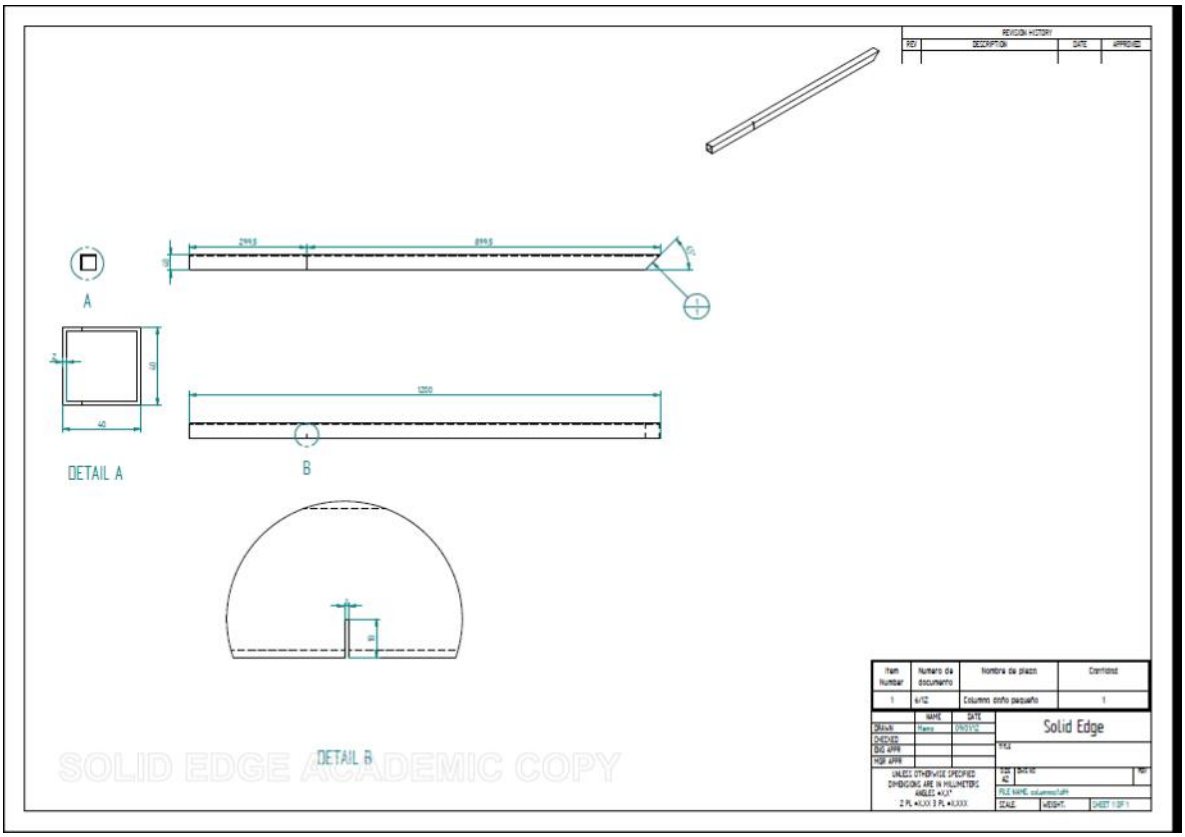


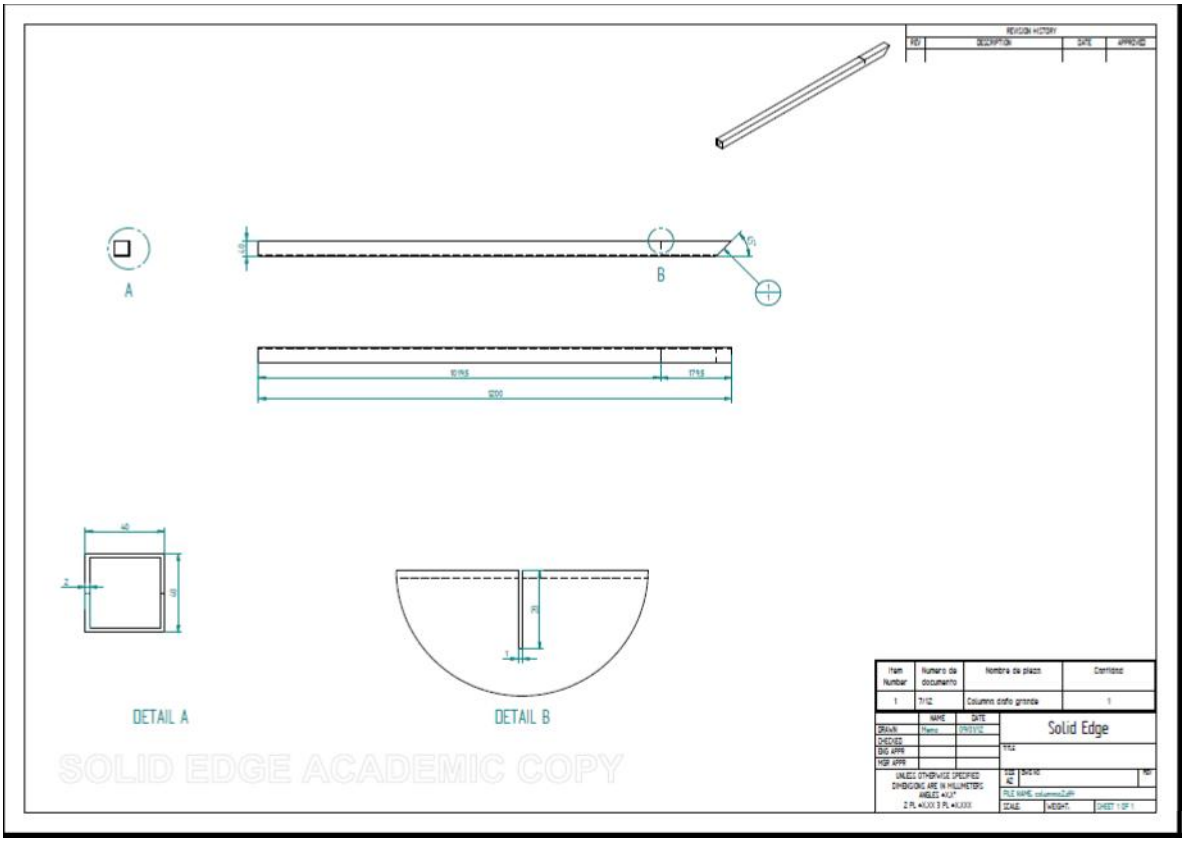

REVISION HISTORY
REV
DESCRIPTION
DATE
APPROVED
A
1200
A
1960
B
SECTION A-A
40
40
2
2
DETAIL B
Item Number
Numero de documento
Nombre de pieza
Cantidad
1
2/12
Columna sin daño
2
NAME
DATE
DRAWN
Hamo
08/31/12
CHECKED
ENG APPR
MGR APPR
Solid Edge
UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN MILLIMETERS
ANGLES ±1/2°
2 PL ±XXX 3 PL ±XXX
SIZE
A2
DWG NO
REV
FILE NAME columnasH
SCALE
WEIGHT
SHEET 1 OF 1
SOLID EDGE ACADEMIC COPY

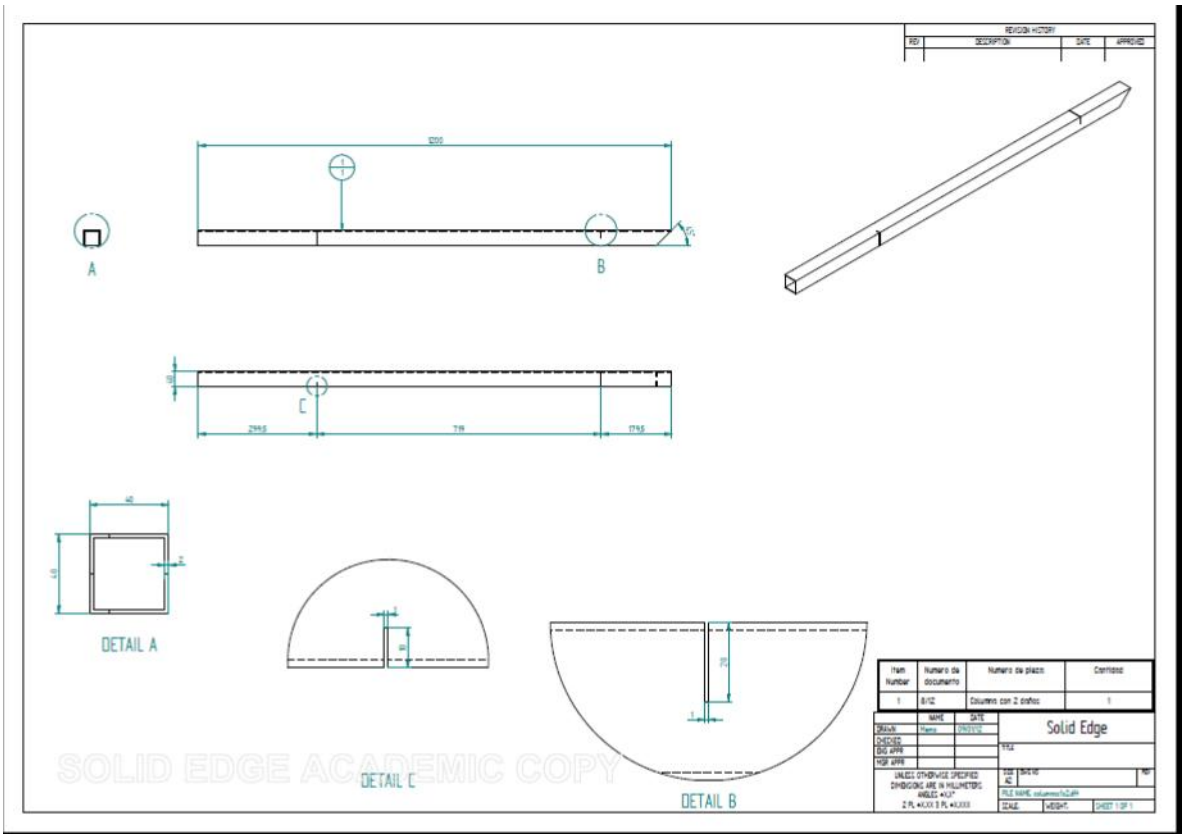

REVISION HISTORY
REV
DESCRIPTION
DATE
APPROVED
A
400
45°
45°
520
40
40
DETAIL A
Item Number
Numero de documento
Nombre de pieza
Cantidad
1
3/12
Barra horizontal 1 sin daño
1
NAME
DATE
DRAWN
Hamo
08/31/12
CHECKED
ENG APPR
MGR APPR
Solid Edge
UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN MILLIMETERS
ANGLES ±1/2°
2 PL ±XXX 3 PL ±XXX
SIZE
A2
DWG NO
REV
FILE NAME vigaSolH
SCALE
WEIGHT
SHEET 1 OF 1
SOLID EDGE ACADEMIC COPY

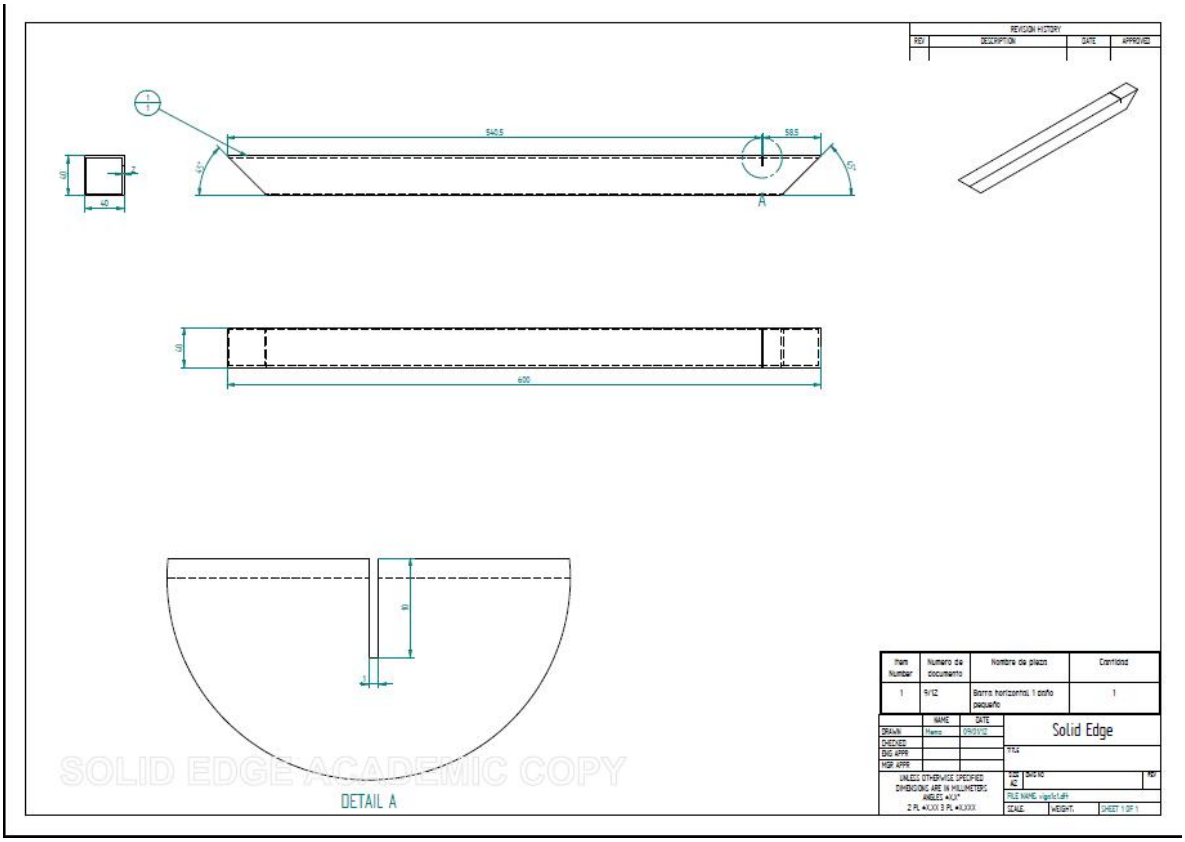

REVISION HISTORY
REV
DESCRIPTION
DATE
APPROVED
A
DETAIL A
Item Number
Numero de documento
Nombre de pieza
Cantidad
1
4/12
Barra horizontal 2 sin agujero
1
Solid Edge
SOLID EDGE ACADEMIC COPY

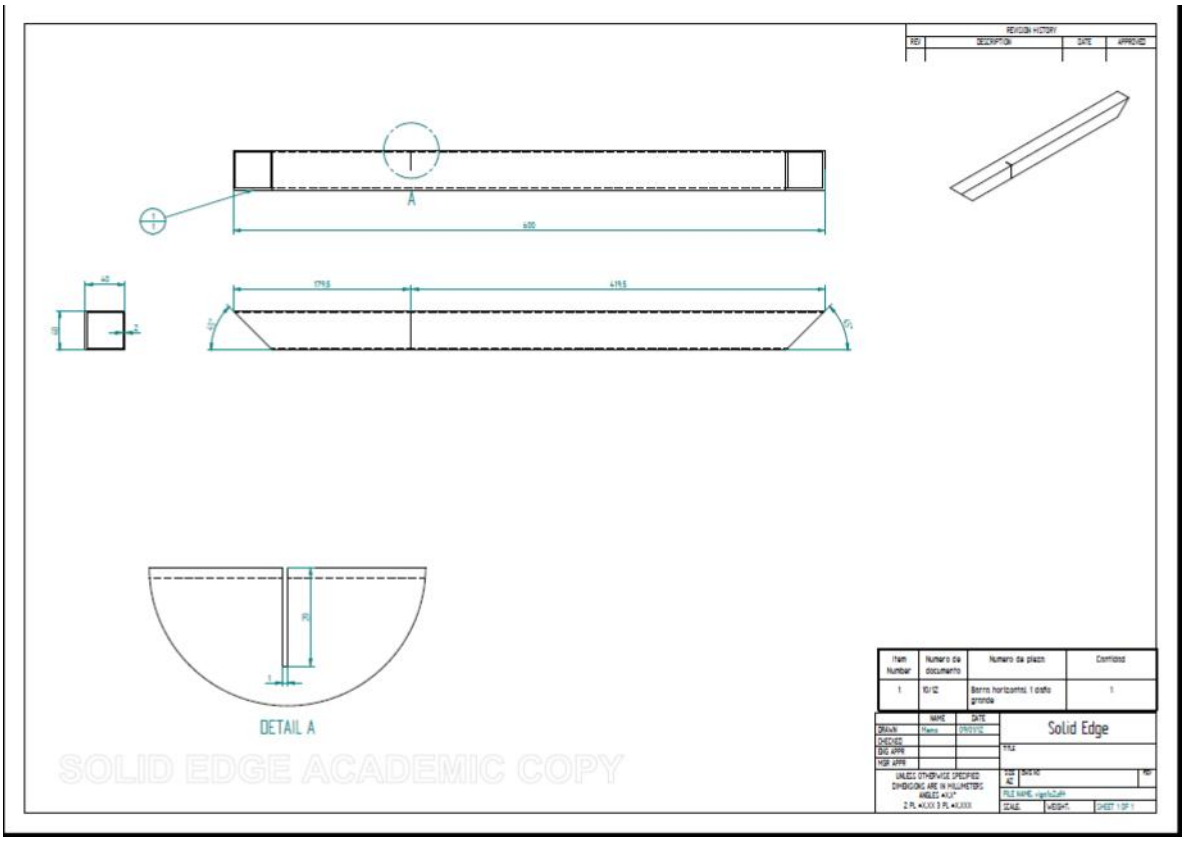

REVISION HISTORY
REV
DESCRIPTION
DATE
APPROVED
Item Number
Numero de documento
Nombre de pieza
Cantidad
1
6/12
Base
2
Solid Edge
SOLID EDGE ACADEMIC COPY

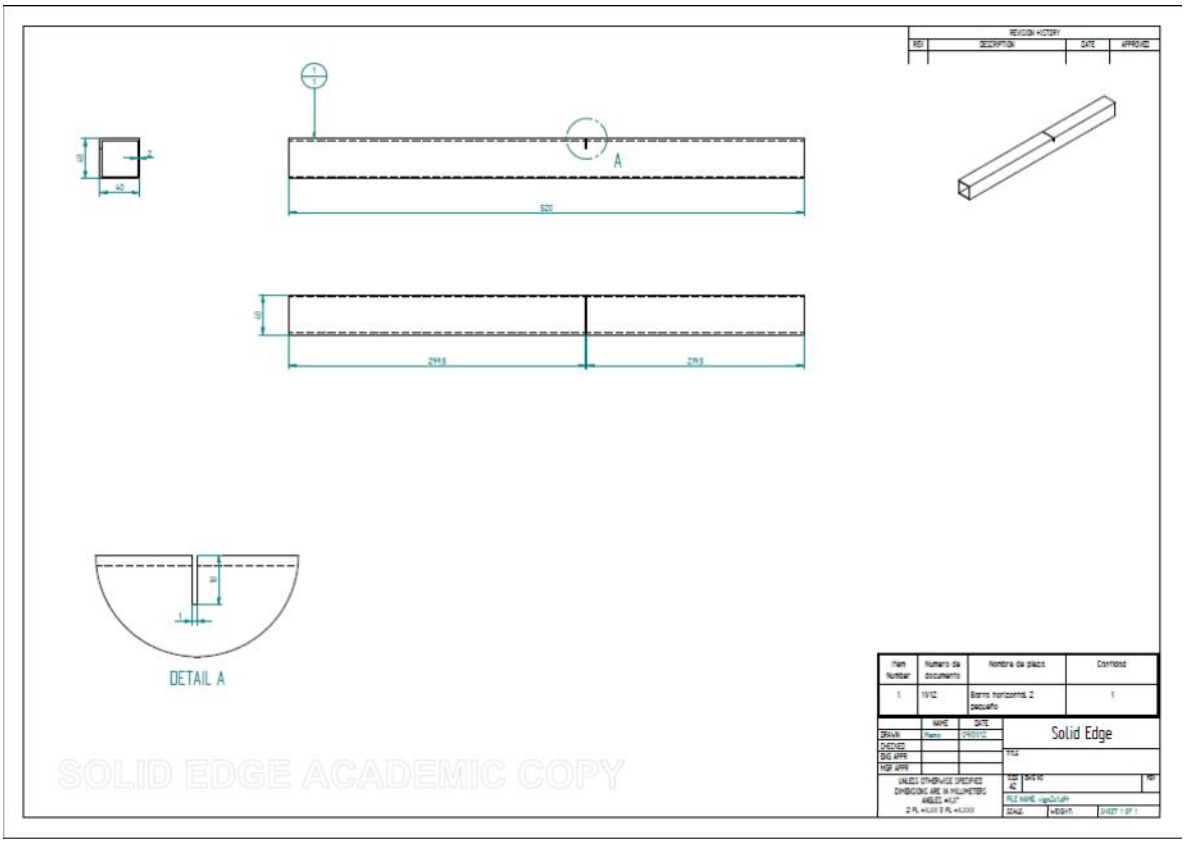

REVISION HISTORY
REV
DESCRIPTION
DATE
APPROVED
A
B
DETAIL A
DETAIL B
Item Number
Numero de documento
Nombre de pieza
Cantidad
1
6/12
Columna corta pequeña
1
NAME
DATE
DRAWN
CHECKED
ENG APPR.
MGR APPR.
Solid Edge
TITLE
SIZE
A2
DWG NO.
REV
FILE NAME: columnas2.par
SCALE:
WEIGHT:
SHEET 1 OF 1
UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN MILLIMETERS
ANGLES ±0.5°
2 PL ±0.XX 3 PL ±0.XXX
SOLID EDGE ACADEMIC COPY

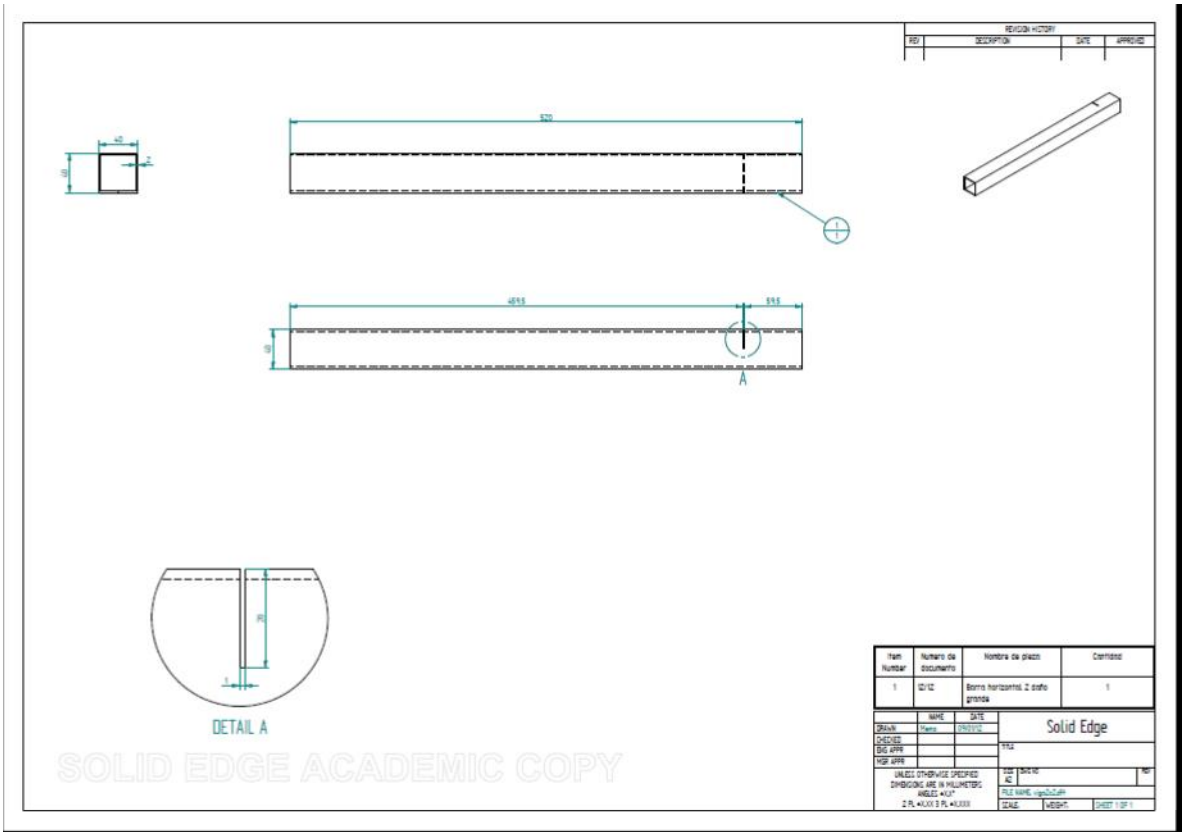

REVISION HISTORY
REV
DESCRIPTION
DATE
APPROVED
A
B
DETAIL A
DETAIL B
Item Number
Numero de documento
Nombre de pieza
Cantidad
1
7/12
Columna corta grande
1
NAME
DATE
DRAWN
CHECKED
ENG APPR.
MGR APPR.
Solid Edge
TITLE
SIZE
A2
DWG NO.
REV
FILE NAME: columnas2.par
SCALE:
WEIGHT:
SHEET 1 OF 1
UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN MILLIMETERS
ANGLES ±0.5°
2 PL ±0.XX 3 PL ±0.XXX
SOLID EDGE ACADEMIC COPY


REVISION HISTORY
REV
DESCRIPTION
DATE
APPROVED
A
B
C
DETAIL A
DETAIL B
DETAIL C
Item Number
Numero de documento
Numero de plano
Cantidad
1
B12
Columna con 2 orificios
1
Solid Edge
SOLID EDGE ACADEMIC COPY


REVISION HISTORY
REV
DESCRIPTION
DATE
APPROVED
A
DETAIL A
Item Number
Numero de documento
Nombre de plano
Cantidad
1
B12
Barra horizontal 1 orificio pequeño
1
Solid Edge
SOLID EDGE ACADEMIC COPY


REVISION HISTORY
REV
DESCRIPTION
DATE
APPROVED
A
600
40
40
3
179.5
419.5
45°
45°
R5
1
DETAIL A
Item Number
Numero de documento
Numero de piezas
Contiene
1
10/12
Barra horizontal 1 dato grande
1
NAME
DATE
DRAWN
Mario
CHECKED
ENG APPR.
MGR APPR.
Solid Edge
UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN MILLIMETERS
ANGLES ±1.0°
2 PL ±0.00 3 PL ±0.000
SIZE
A2
DWG NO
REV
FILE NAME
SCALE
WEIGHT
SHEET 1 OF 1
SOLID EDGE ACADEMIC COPY


REVISION HISTORY
REV
DESCRIPTION
DATE
APPROVED
A
40
40
520
294.5
294.5
DETAIL A
Item Number
Numero de documento
Nombre de piezas
Contiene
1
10/12
Barra horizontal 2 pequeño
1
NAME
DATE
DRAWN
Mario
CHECKED
ENG APPR.
MGR APPR.
Solid Edge
UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN MILLIMETERS
ANGLES ±1.0°
SIZE
A2
DWG NO
REV
FILE NAME
SCALE
WEIGHT
SHEET 1 OF 1
SOLID EDGE ACADEMIC COPY



Anexo B

Código modelo final primera red

Programa principal

```
clear all;
close all;
clc;

n=2; %combinaciones de daños
dc=nchoosek(1:6,n); %combinaciones de n daños en elementos del 1 al 30
dl=[0.0:0.05:0.95]; %5 niveles de daño

A=zeros(length(dl)^n*length(dc),6);% numero de patrones de entrenamiento

j=1;
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

%desordenar A
Nt=randperm(length(A)); %permutación de numeros del 1 al length(A)
At=A(Nt,:);

%Datos de validación
dl=[0.02:0.05:0.95]; %niveles de daño

A=zeros(length(dl)^n*length(dc),6);
j=1;
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

%desordenar A
Nv=randperm(length(A));
Av=A(Nv,:);
A=[At;Av];

% A=At;

Nt=1:length(At);
Nv=setdiff(1:length(A),Nt);
save Nt;
save Nv;
save A A;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for p=1:length(A)
    wa(p,:)=FEmarco(A(p,:));
    p
    %efectiva
end


wa0=FEmarco(zeros(6,1));

save wa wa;
```

```
save wa0 wa0;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
load wa wa;
load wa0 wa0;
load A;

%añadir ruido
noise=0.5;
a=-noise/100;
b=noise/100;
wa=( 1 + (a + (b-a)*rand(size(wa)))).*wa;
wa0=( 1 + (a + (b-a)*rand(size(wa0)))).*wa0;

wa0=repmat(wa0',[length(wa),1]);%igualar dimensiones de wa0 con wa

%determinar inputs
dinput=abs(wa-wa0)./wa0;

doutput=A';
dinput=dinput';

save dinput dinput;
save doutput doutput;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
load Nt;
load Nv;

h=[10:10:50];
N=4;
i=1;
for i= 1:5
k=1;
tiempo=0;
perf=0;
vperf=0;

  for k=1:N

net = newfit(dinput,doutput,[30 30],{'logsig','logsig','satlins'},'trainlm');

net.divideFcn = 'divideind';
net.divideParam.trainInd=Nt;
net.divideParam.valInd=Nv;
net.divideParam.testInd=[];
```

```
net.trainParam.max_fail=10;

[net,tr,Y,E,Pf,Af] = train(net,dinput,doutput);
save net net;
save tr tr;

tiempo=tiempo+max(tr.time)/N;
perf=perf+min(tr.perf)/N;
vperf=vperf+min(tr.vperf)/N;

save tiempo tiempo;
save perf perf;
save vperf vperf;
save net net;
   end
resultadosr(t).tiempo=tiempo;
resultadosr(t).perf=perf;
resultadosr(t).vperf=vperf;
save resultadosr resultadosr;
save t t;
end

%%%%%%%%%%%%%MSE%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
load net;
load dinput;
load doutput;
load Nv;
load Nt;
dsim=1-sim(net,dinput(:,Nv));
doutput=1-doutput(:,Nv);

MSE=sum(sum(abs(dsim-doutput)))/(size(dsim,1)*size(dsim,2));
Dr=doutput<1;
Dd=dsim<(1-2*MSE);
NT=sum(sum(Dr));
NF=sum(sum(Dd));
DME=1/NT*(NT-sum(sum(Dr.*Dd)));
FAE=1/NF*sum(sum((1-Dr).*Dd));

%%%%%%%%%%%Por nivel de daño%%%%%%%%%%%%%%%%%%%%%%%
l=[0.9:-0.1:0.1];
% for i=1:3
i=1;
%nivel 1: 0-x% daño
Dr1=(doutput>l(i)).*(doutput<1);
Dd1=(dsim>l(i)).*(dsim<(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-Dr1).*Dd1)); %que porcentaje de los daños detectados entre 0 y x% son falsos


for i=2:9
Dr1=(doutput>l(i)).*(doutput<(l(i)+0.1));
Dd1=(dsim>l(i)).*(dsim<(l(i)+0.1));
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,'w')
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('Daño en sub-estructura (%)')
ylabel('Daños no detectados (%)')
%
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,'w')
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('Daño en sub-estructura (%)')
ylabel('Daños falsos (%)')
```

Rutina de obtención de parámetros

```
function wa=FEmarco(u)
% u=zeros(1,6);
load ua0;
[n,m]=size(u);
if n>m
   u=u';
end
a=[0 u(1)*ones(1,3) 0 0 u(2)*ones(1,3) 0 0 u(3)*ones(1,3) 0 0 u(4)*ones(1,3) 0 0 u(5)*ones(1,3)
0 0 u(6)*ones(1,3) 0];
wz=marcoZ_s(ua0,a); %marcocuad recibe el vector de rigidez
wx=marcoX_s(ua0,a); %marcocuad recibe el vector de rigidez
wa=[wz' wx']';

function ws=marcoZ_s(ua,u)
% load uw;
% ua=uw;
% ua=ones(1,10);
% u=zeros(30,1);

ub=ones(1,30);
ub([1,5,10,11,15,16,21,25,26,30])=ua([11 12 13 14 14 13 12 11 15 15]);
Mm=0.1*eye(3,3)*ua(4);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%
%definir propiedades del material(revisar!!)
E=2.1e11*ua(1); %Modulo de Young Pa
nu=0.3; %Coeficiente de Poisson
rho=7840*ua(2); %densidad kg/m3

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%
%definir propiedades de las vigas
%alas
e=0.002*ua(3); %espesor m
b=0.04; %ancho m
le=0.12; %largo de un elemento m
lu=1e-2; %largo union m
n=36; %numero de elementos
nd=36; %numero de nodos
ne=nd*6; %grados de libertad
%inercias (en ejes locales)
Iy=(b^4/12)-((b-2*e)^4/12);
Iz=(b^4/12)-((b-2*e)^4/12);
```

```
Ix=Iy+Iz;
%area
A=(b-2*e)^2;

%matriz de un elemento
[Ke,Me]= matrices(E,rho,nu,le,A,Ix,Iy,Iz);

%matriz de un elemento
[Ku,Mu]= matrices(E,rho,nu,lu,A,Ix,Iy,Iz);

%Rotación de matrices
R1=[0 1 0; -1  0 0; 0 0 1]; %viga en dirección y
R2=[1 0 0; 0 1 0; 0 0 1]; %viga en dirección x
R3=[0 -1 0; 1 0 0; 0 0 1]; %viga en dirección -y

C=zeros(3,3);
T1=[R1 C  C  C ;
   C  R1 C  C ;
   C  C  R1 C ;
   C  C  C  R1 ];
T2=[R2 C  C  C ;
   C  R2 C  C ;
   C  C  R2 C ;
   C  C  C  R2 ];
T3=[R3 C  C  C ;
   C  R3 C  C ;
   C  C  R3 C ;
   C  C  C  R3 ];

%alas
K1=T1'*Ke*T1;
M1=T1'*Me*T1;
K2=T2'*Ke*T2;
M2=T2'*Me*T2;
K3=T3'*Ke*T3;
M3=T3'*Me*T3;

Ku1=T1'*Ku*T1;
Mu1=T1'*Mu*T1;
Ku2=T2'*Ku*T2;
Mu2=T2'*Mu*T2;
Ku3=T3'*Ku*T3;
Mu3=T3'*Mu*T3;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%
%Ensamble
```

```
%matrices iniciales
K=zeros(ne,ne);
M=zeros(ne,ne);

%locel: grados de libertad asociados a cada elemento
locel=zeros(n,12);

%locel para elementos consecutivos
for i=1:10
   j=6*(i-1)+1;
   locel(i,1:12)=(j):(j+11);
end

for i=11:15
   j=6*(i)+1;
   locel(i,1:12)=(j):(j+11);
end

for i=16:25
   j=6*(i+1)+1;
   locel(i,1:12)=(j):(j+11);
end

for i=26:30
   j=6*(i+2)+1;
   locel(i,1:12)=(j):(j+11);
end

locel(31,1:12)=[(6*35-5):6*35 (6*1-5):6*1];
locel(32,1:12)=[(6*6-5):6*6 (6*29-5):6*29];
locel(33,1:12)=[(6*11-5):6*11 (6*12-5):6*12];
locel(34,1:12)=[(6*17-5):6*17 (6*18-5):6*18];
locel(35,1:12)=[(6*34-5):6*34 (6*23-5):6*23];
locel(36,1:12)=[(6*36-5):6*36 (6*28-5):6*28];

%Ensamble para cada orientación de elementos
for i=1:10 %elementos del 1 a 10
   K(locel(i,:),locel(i,:))=K(locel(i,:),locel(i,:))+(1-u(i))*ub(i)*K1;
   M(locel(i,:),locel(i,:))=M(locel(i,:),locel(i,:))+M1;
end

for i=[11:15 26:30]
   K(locel(i,:),locel(i,:))=K(locel(i,:),locel(i,:))+(1-u(i))*ub(i)*K2;
   M(locel(i,:),locel(i,:))=M(locel(i,:),locel(i,:))+M2;
end

for i=16:25 %elementos del 16 al 25
   K(locel(i,:),locel(i,:))=K(locel(i,:),locel(i,:))+(1-u(i))*ub(i)*K3;
```

```
   M(locel(i,:),locel(i,:))=M(locel(i,:),locel(i,:))+M3;
end


for i=[31 36] %elementos del 16 al 25
   K(locel(i,:),locel(i,:))=K(locel(i,:),locel(i,:))+ua(5)*Ku1;
   M(locel(i,:),locel(i,:))=M(locel(i,:),locel(i,:))+Mu1;
end

for i=[32 35] %elementos del 16 al 25
   K(locel(i,:),locel(i,:))=K(locel(i,:),locel(i,:))+ua(6)*Ku2;
   M(locel(i,:),locel(i,:))=M(locel(i,:),locel(i,:))+Mu2;
end

for i=[33 34] %elementos del 16 al 25
   K(locel(i,:),locel(i,:))=K(locel(i,:),locel(i,:))+ua(7)*Ku2;
   M(locel(i,:),locel(i,:))=M(locel(i,:),locel(i,:))+Mu2;
end


locelm=[31:33;
       61:63;
       103:105;
       133:135];

for i=1:4 %masas puntuales
   M(locelm(i,:),locelm(i,:))=M(locelm(i,:),locelm(i,:))+Mm;
end

X=1:6:ne;
Y=2:6:ne;
Z=3:6:ne;
Px=4:6:ne;
Py=5:6:ne;
Pz=6:6:ne;

M1=sort([Z Px Py]); %en dirección Z

Kz=K(M1,M1);
Mz=M(M1,M1);

ne=length(Kz);

%fijar grados de libertad (60 grados)
fix=[(3*35-2):3*35 (3*36-2):3*36]; % Patas fijas al suelo

free=setdiff(1:ne,fix);
```

```
Kz=Kz(free,free);
Mz=Mz(free,free);
M1=M1(free);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%frecuencias naturales y modos propios
[P,W]=eig(Kz,Mz);

w=sqrt(diag(W))/(2*pi);
[w,j]=sort(w);
Phi=P(:,j);
Phiz=Phi(1:3:(ne-6),1:4);
wr=w(1:4);

ne=length(Kz);

J=1:ne;

dz=[2 5 7 10 13 22 24 30 33]*3-2;

NN=[2 2 2 3 3 1 1 2 1];

Waz=[];
for i=1:9
% i=4;
aux2=setdiff(J,28);
aux1=setdiff(J,dz(i));
Ma=Mz(aux1,aux2);
Ka=Kz(aux1,aux2);
[~,wa]=eig(Ka,Ma);
wa=sqrt(diag(wa))/2/pi;
wa=sort(real(wa(:,1)));
wa=wa(wa>0.1);
Waz=[Waz; wa(1:NN(i))];
% k=k+1;
end

ws=[wr;Waz];

function ws=marcoX_s(ua,u)

% load ua0;
% ua=ua0;
% ua=ones(1,16);
% u=zeros(30,1);
% u(25)=0.99;
```

```
ub=ones(1,30);
ub([1,5,10,11,15,16,21,25,26,30])=ua([11 12 13 14 14 13 12 11 15 15]);
Mm=0.1*eye(3,3)*ua(4);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%
%definir propiedades del material(revisar!!)
E=2.1e11*ua(1); %Modulo de Young Pa
nu=0.3; %Coeficiente de Poisson
rho=7840*ua(2); %densidad kg/m3

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%
%definir propiedades de las vigas
%alas
e=0.002*ua(3); %espesor m
b=0.04; %ancho m
le=0.12; %largo de un elemento m
lu=1e-2; %largo union m
n=36; %numero de elementos
nd=36; %numero de nodos
ne=nd*6; %grados de libertad
%inercias (en ejes locales)
Iy=(b^4/12)-((b-2*e)^4/12);
Iz=(b^4/12)-((b-2*e)^4/12);
Ix=Iy+Iz;
%area
A=(b-2*e)^2;

%matriz de un elemento
[Ke,Me]= matrices(E,rho,nu,le,A,Ix,Iy,Iz);

%matriz de un elemento
[Ku,Mu]= matrices(E,rho,nu,lu,A,Ix,Iy,Iz);

%Rotación de matrices
R1=[0 1 0; -1  0 0; 0 0 1]; %viga en dirección y
R2=[1 0 0; 0 1 0; 0 0 1]; %viga en dirección x
R3=[0 -1 0; 1 0 0; 0 0 1]; %viga en dirección -y

C=zeros(3,3);
T1=[R1 C  C  C ;
   C  R1 C  C ;
   C  C  R1 C ;
   C  C  C  R1 ];
T2=[R2 C  C  C ;
   C  R2 C  C ;
   C  C  R2 C ;
```

```
   C  C  C  R2 ];
T3=[R3 C  C  C ;
   C  R3 C  C ;
   C  C  R3 C ;
   C  C  C  R3 ];

%alas
K1=T1'*Ke*T1;
M1=T1'*Me*T1;
K2=T2'*Ke*T2;
M2=T2'*Me*T2;
K3=T3'*Ke*T3;
M3=T3'*Me*T3;

Ku1=T1'*Ku*T1;
Mu1=T1'*Mu*T1;
Ku2=T2'*Ku*T2;
Mu2=T2'*Mu*T2;
Ku3=T3'*Ku*T3;
Mu3=T3'*Mu*T3;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%
%Ensamble

%matrices iniciales
K=zeros(ne,ne);
M=zeros(ne,ne);

%locel: grados de libertad asociados a cada elemento
locel=zeros(n,12);

%locel para elementos consecutivos
for i=1:10
   j=6*(i-1)+1;
   locel(i,1:12)=(j):(j+11);
end

for i=11:15
   j=6*(i)+1;
   locel(i,1:12)=(j):(j+11);
end

for i=16:25
   j=6*(i+1)+1;
   locel(i,1:12)=(j):(j+11);
end
```

```
for i=26:30
   j=6*(i+2)+1;
   locel(i,1:12)=(j):(j+11);
end

locel(31,1:12)=[(6*35-5):6*35 (6*1-5):6*1];
locel(32,1:12)=[(6*6-5):6*6 (6*29-5):6*29];
locel(33,1:12)=[(6*11-5):6*11 (6*12-5):6*12];
locel(34,1:12)=[(6*17-5):6*17 (6*18-5):6*18];
locel(35,1:12)=[(6*34-5):6*34 (6*23-5):6*23];
locel(36,1:12)=[(6*36-5):6*36 (6*28-5):6*28];

%Ensamble para cada orientación de elementos
for i=1:10 %elementos del 1 a 10
   K(locel(i,:),locel(i,:))=K(locel(i,:),locel(i,:))+(1-u(i))*ub(i)*K1;
   M(locel(i,:),locel(i,:))=M(locel(i,:),locel(i,:))+M1;
end

for i=[11:15 26:30]
   K(locel(i,:),locel(i,:))=K(locel(i,:),locel(i,:))+(1-u(i))*ub(i)*K2;
   M(locel(i,:),locel(i,:))=M(locel(i,:),locel(i,:))+M2;
end

for i=16:25 %elementos del 16 al 25
   K(locel(i,:),locel(i,:))=K(locel(i,:),locel(i,:))+(1-u(i))*ub(i)*K3;
   M(locel(i,:),locel(i,:))=M(locel(i,:),locel(i,:))+M3;
end

for i=[31 36] %elementos del 16 al 25
   K(locel(i,:),locel(i,:))=K(locel(i,:),locel(i,:))+ua(8)*Ku1;
   M(locel(i,:),locel(i,:))=M(locel(i,:),locel(i,:))+Mu1;
end

for i=[32 35] %elementos del 16 al 25
   K(locel(i,:),locel(i,:))=K(locel(i,:),locel(i,:))+ua(9)*Ku2;
   M(locel(i,:),locel(i,:))=M(locel(i,:),locel(i,:))+Mu2;
end

for i=[33 34] %elementos del 16 al 25
   K(locel(i,:),locel(i,:))=K(locel(i,:),locel(i,:))+ua(10)*Ku2;
   M(locel(i,:),locel(i,:))=M(locel(i,:),locel(i,:))+Mu2;
end

locelm=[31:33;
      61:63;
      103:105;
      133:135];
```

```
for i=1:4 %masas puntuales
    M(locelm(i,:),locelm(i,:))=M(locelm(i,:),locelm(i,:))+Mm;
end

X=1:6:ne;
Y=2:6:ne;
Z=3:6:ne;
Px=4:6:ne;
Py=5:6:ne;
Pz=6:6:ne;

M2=sort([X Y  Pz]); %en dirección Z

Kx=K(M2,M2);
Mx=M(M2,M2);

ne=length(Kx);

%fijar grados de libertad (60 grados)
fix=[(3*35-2):3*35 (3*36-2):3*36]; % Patas fijas al suelo

free=setdiff(1:ne,fix);

Kx=Kx(free,free);
Mx=Mx(free,free);
M2=M2(free);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%frecuencias naturales y modos propios
[P,W]=eig(Kx,Mx);

w=sqrt(diag(W))/(2*pi);
[w,j]=sort(w);
Phi=P(:,j);
Phix=Phi(1:3:(ne-6),1:2);
Phiy=Phi(2:3:(ne-6),1:2);
wr=w(1:2);

ne=length(Kx);

J=1:ne;

dxy=[2 5 7 10 19 22 24 27]*3-2;

NN=[1 1 1 2 2 2 1 1];

Wax=[];
```

```
for i=1:8
% i=4;
aux2=setdiff(J,28);
aux1=setdiff(J,dxy(i));
Ma=Mx(aux1,aux2);
Ka=Kx(aux1,aux2);
[~,wa]=eig(Ka,Ma);
wa=sqrt(diag(wa))/2/pi;
wa=sort(real(wa(:,1)));
wa=wa(wa>0.1);
Wax=[Wax; wa(1:NN(i))];
% k=k+1;
end

ws=[wr;Wax];
```

Funcion que entrega matrices de rigidez y masa

```
function [K,M]= matrices(E,rho,nu,l,A,Ix,Iy,Iz)
% [K,M]= matrices(E,rho,nu,l,A,Ix,Iy,Iz)
%-------------------------------------------------------------
% PURPOSE
% Compute stiffness and mass matrices of a tridimensional
% beam element
%
% INPUT:
%    E = Young modulus (Pa)
%  rho = Density (Kg/m^3)
%    G = Shear modulus (Pa)
%    l = Element lenght (m)
%    A = Cross section area (m^2)
%   Ix = Inertia cross section x-axis (torsion)
%   Iy = Inertia cross section y-axis (bending)
%   Iz = Inertia cross section z-axis (bending)
%
% OUTPUT:
%    K = element stiffness matrix, dim(K)= 12 x 12
%    M = element mass matrix, dim(M)= 12 x 12
%-------------------------------------------------------------

r=Ix/A;
G=E/(2*(1+nu));
K=[E*A/l      0          0          0          0          0          -E*A/l      0          0          0          0
0;
    0       12*E*Iz/l^3 0          0          0          6*E*Iz/l^2  0          -12*E*Iz/l^3 0          0          0
6*E*Iz/l^2;
```

```
   0       0          12*E*Iy/l^3 0         -6*E*Iy/l^2  0          0          0          -12*E*Iy/l^3 0         -
6*E*Iy/l^2  0;
   0       0          0          G*Ix/l     0          0          0          0          0          -G*Ix/l     0          0;
   0       0          -6*E*Iy/l^2  0          4*E*Iy/l    0          0          0          6*E*Iy/l^2  0
2*E*Iy/l    0;
   0       6*E*Iz/l^2  0          0          0          4*E*Iz/l    0          -6*E*Iz/l^2  0          0          0
2*E*Iz/l;
   -E*A/l    0          0          0          0          0          E*A/l      0          0          0          0          0;
   0       -12*E*Iz/l^3 0          0          0          -6*E*Iz/l^2  0          12*E*Iz/l^3 0          0          0
-6*E*Iz/l^2;
   0       0          -12*E*Iy/l^3 0          6*E*Iy/l^2  0          0          0          12*E*Iy/l^3 0
6*E*Iy/l^2  0;
   0       0          0          -G*Ix/l     0          0          0          0          0          G*Ix/l     0          0;
   0       0          -6*E*Iy/l^2  0          2*E*Iy/l    0          0          0          6*E*Iy/l^2  0
4*E*Iy/l    0
   0       6*E*Iz/l^2  0          0          0          2*E*Iz/l    0          -6*E*Iz/l^2  0          0          0
4*E*Iz/l];

M=rho*A*l*[1/3   0          0          0          0          0          1/6   0          0          0          0          0;
          0     13/35      0          0          0          11/210*l  0     9/70      0          0          0          -13/210*l;
          0     0          13/35      0          -11/210*l  0          0     0          9/70       0          13/210*l  0;
          0     0          0          r^2/3      0          0          0     0          0          r^2/6      0          0;
          0     0          -11/210*l 0          l^2/105    0          0     0          -13/420*l   0     -l^2/140   0;
          0     11/210*l  0          0          0          l^2/105    0     13/420*l  0          0          0          -l^2/140;
          1/6   0          0          0          0          0          1/3   0          0          0          0          0;
          0     9/70       0          0          0          13/420*l  0     13/35      0          0          0          -11/210*l;
          0     0          9/70       0          -13/420*l  0          0     0          13/35      0          11/210*l  0;
          0     0          0          r^2/6      0          0          0     0          0          r^2/3      0          0;
          0     0          13/210*l 0          -l^2/140   0          0     0          11/210*l   0     l^2/105    0;
          0    -13/210*l  0          0          0          -l^2/140   0    -11/210*l  0          0          0          l^2/105];
```