

**IDENTIFICACIÓN DE DAÑO EN UN PUENTE PEATONAL POR MEDIO DE
REDES NEURONALES
-TRABAJO DE GRADO-**

Alba Ruth Martinez Rojas

Trabajo de grado presentado al programa académico de Ingeniería Civil de
la Universidad del Valle



Universidad del Valle
Facultad de Ingeniería
Escuela de Ingeniería Civil y Geomática
Santiago de Cali
2018

**IDENTIFICACIÓN DE DAÑO EN UN PUENTE PEATONAL POR MEDIO DE
REDES NEURONALES**

-TRABAJO DE GRADO-

Alba Ruth Martinez Rojas

Directores:

Johannio Marulanda, Ph.D

Peter Thomson, Ph.D

Universidad del Valle

Facultad de Ingeniería

Escuela de Ingeniería Civil y Geomática

Santiago de Cali

2018

NOTA DE ACEPTACIÓN

El presente trabajo de grado fue aprobado por el director del Programa Académico, los jurados y los directores de investigación.

MANOLO GALVÁN CEBALLOS, Ph.D.
Director del Programa Académico de Ingeniería Civil

JOHANNIO MARULANDA CASAS, Ph.D.
Director de investigación

PETER THOMSON, Ph.D.
Director de investigación

Jurado

Jurado

Ciudad y Fecha

AGRADECIMIENTOS

Agradezco en primer lugar a Dios, a mis padres, familia y amigos por su apoyo incondicional durante esta formación tanto personal como académica. A mis directores Johannio Marulanda y Peter Thomson, por su guía, tiempo y aportes para este trabajo. Agradezco muy especialmente al profesor Wilfredo Alfonso Morales por su orientación y los conocimientos brindados para el desarrollo esta investigación. Finalmente, al grupo de investigación G-7, en especial a Diana Millán, Angélica Panesso, Paula Romero y Sebastián Castellanos por sus valiosos aportes.

TABLA DE CONTENIDO

1.	Resumen	11
2.	Abstract.....	12
3.	Introducción.....	13
4.	Planteamiento del Problema	14
	Descripción.....	14
	Formulación.....	17
	Objetivos.....	17
	Objetivo General.....	17
	Objetivos Específicos	17
	Justificación	18
5.	Marco de referencia	20
	Antecedentes y estado del arte	21
6.	Marco teórico.....	24
	Técnicas de identificación modal	24
	Detección de daño	24
	Red neuronal artificial	25
	Métodos o algoritmos de entrenamiento de una RNA	30
	Perceptrón multicapa	31
	Redes de Base Radial (RBF)	32
7.	Referente Empírico.....	34
	Modelo numérico.....	37

	6
8. Evaluación de las técnicas	39
Metodología.....	39
Metodología para entrenar las RNA.....	40
Procesamiento de los datos de aprendizaje y validación (entrada a la red).....	43
Diferencia de los ceros de la función de transferencia	45
Espacio de estados	46
Modos y frecuencias	47
Amplitudes de la función de transferencia	48
Resultados.....	50
Diferencia de los ceros de la función de transferencia	50
Espacio de estados	54
Modos y frecuencias	56
Amplitudes de la función de transferencia	62
9. Discusión	68
10. Conclusiones.....	70
11. Referencias	72

LISTA DE FIGURAS

Figura 1. Colapso de puente peatonal al norte de Bogotá, Colombia (El Espectador, recuperado de: http://www.elespectador.com/noticias/bogota/puente-caido-contrato-avaluaron-dos-subtenientes-articulo-541669)	15
Figura 2. Puente I-35W del río Mississippi, cuatro días después del colapso (Civil Engineering, recuperado de: https://twitter.com/engineeringcivi/status/561497739658534913).	19
Figura 3. Elementos de una Neurona Artificial (Modificado de Caicedo, 2017).	26
Figura 4. Estructura de una red neuronal artificial: a) monocapa b) multicapa (Modificado de Caicedo, 2017).	28
Figura 5. Estructura de una red Perceptrón multicapa (modificado de Alfonso, 2017).	32
Figura 6. Puente peatonal usado como referente.	34
Figura 7. Distribución de los sensores en el puente.	35
Figura 8. Modos de vibración experimentales (Modificado de Panesso, 2017).	36
Figura 9. Modelo del puente peatonal sin daño.	37
Figura 10. Diagrama de flujo de la metodología seguida.	39
Figura 11. Escenario sin daño (Panesso, 2017).	43
Figura 12. Escenarios de daño para validar (Panesso, 2017).	44
Figura 13. Representación de los daños seleccionados para validar las redes.	44
Figura 14. Polos y ceros de la función de transferencia.	45
Figura 15. Derivada de la función de transferencia, con los ceros de la función de transferencia.	46
Figura 16. Representación gráfica del modelo de espacio de estados con un integrador. ...	47
Figura 17. Comparación entre modos analíticos y experimentales.	49
Figura 18. Evolución del error usando la diferencia de los ceros de la función de transferencia.	51

Figura 19. Matriz de confusión con los datos de entrenamiento, usando la diferencia de los ceros de la función de transferencia.	52
Figura 20. Daños analíticos encontrados por la red tipo Perceptrón Multicapa, usando como entrada la diferencia de los ceros de la función de transferencia.	53
Figura 21. Daños analíticos encontrados por la red de base radial, usando como entrada la diferencia de los ceros de la función de transferencia.	54
Figura 22. Evolución del error usando el vector de estados x.	55
Figura 23. Daños analíticos encontrados por la red tipo Perceptrón Multicapa, usando como entrada el vector de estados x.	56
Figura 24. Evolución del error usando las frecuencias.	57
Figura 25. Daños analíticos encontrados por la red tipo Perceptrón Multicapa, usando como entrada las frecuencias.	58
Figura 26. Daños experimentales encontrados por la red tipo Perceptrón Multicapa, usando como entrada las frecuencias.	59
Figura 27. Daños analíticos encontrados por la red de Base Radial, usando como entrada las frecuencias.	60
Figura 28. Daños experimentales encontrados por la red la Red de Base Radial, usando como entrada las frecuencias.	61
Figura 29. Evolución del error usando las amplitudes de la función de transferencia.	63
Figura 30. Daños analíticos encontrados por la red tipo Perceptrón Multicapa, usando como entrada amplitudes de la función de transferencia.	63
Figura 31. Daños experimentales encontrados por la red tipo Perceptrón Multicapa, usando como entrada amplitudes de la función de transferencia.	64
Figura 32. Daños analíticos encontrados por la red de Base Radial, usando como entrada amplitudes de la función de transferencia.	65
Figura 33. Daños experimentales encontrados por la red de Base Radial, usando como entrada amplitudes de la función de transferencia.	65

Figura 34. Daños analíticos encontrados por la red tipo Perceptrón Multicapa, usando como entrada amplitudes de la función de transferencia sin el valor absoluto	67
---	----

LISTA DE TABLAS

Tabla 1. Inventario de puentes estudiados (Modificado de: G-7 & CARSIG, 2015).....	18
Tabla 2. Inventario de Box Culvert Inspeccionados (Modificado de: G-7 & CARSIG, 2015).....	18
Tabla 3. Frecuencias naturales identificadas experimentalmente del puente en estado sano	35
Tabla 4. Ajuste del modelo. Modificado de Panesso (2017).....	38
Tabla 5. Mejores topologías para la primera entrada, por cada entrenamiento.....	50
Tabla 6. Diferencias en frecuencia entre el estado sano y los cinco escenarios de daño (analíticos).	58
Tabla 7. Resumen de los resultados obtenidos al entrenar redes de tipo Perceptrón con los datos analíticos.	68

1. RESUMEN

El presente trabajo desarrolla y aplica dos modelos de redes neuronales artificiales para la identificación de daño estructural en el puente peatonal construido por el grupo de investigación G7, a partir de una identificación modal y pruebas experimentales previas y externas a este trabajo. El primer modelo es una red de tipo Perceptrón multicapa, la cual es una de las redes más utilizadas en el medio y se caracteriza por tener un aprendizaje supervisado y por usar funciones sigmoideas; y el segundo modelo es una red de Base Radial, que se diferencia de la anterior en que está compuesta por una sola capa oculta y por funciones gaussianas. En primer lugar, se realizó un análisis de sensibilidad con la información conocida de la estructura y se probaron diferentes tipos de procesamiento para obtener distintas entradas en la red neuronal. Este procesamiento se realizó tanto para el caso sano, como para los casos con daño, los cuales se simularon quitando elementos en diferentes tramos de la cercha, es decir, variando la rigidez de la estructura y la masa. En el entrenamiento se utilizaron diferentes algoritmos para Perceptrón multicapa, pero finalmente se escogió Levenberg Marquardt, mientras que la red de Base Radial utiliza un método de aprendizaje híbrido por defecto. Para la validación analítica se utilizó un modelo numérico en SAP2000 del puente (ajustado previamente); el procesamiento de las señales de respuesta del puente y el entrenamiento de las redes se realizó en MATLAB. Finalmente, se obtuvieron mejores resultados con el Perceptrón multicapa, observando que la red es capaz de predecir el daño.

PALABRAS CLAVE: Identificación de daño, redes neuronales artificiales, puente peatonal.

2. ABSTRACT

The present work of development and application of models of artificial neural networks for the identification of structural damage in the pedestrian bridge of the G7 research group, from a modal identification and experimental tests previous and external to this work. The first model is a multilayer Perceptron type sample, which is one of the most used networks in the environment and is characterized by supervised learning and by using sigmoid functions; and the second model is a radial red base, which differs from the previous one in that it is composed of a single hidden layer and by Gaussian functions. First, a sensitivity analysis was carried out with the known information of the structure and different types of processing were tested to obtain different inputs in the neural network. This processing was done both for the healthy case, and for cases with damage, the elements can be divided into different elements of the truss, that is, varying the rigidity of the structure. In the training different algorithms for multilayer Perceptron are used, but finally Levenberg Marquardt was chosen, while the Radial Base Network uses a hybrid learning method by default. For the analytical validation a numerical model was used in SAP2000 of the bridge (previously adjusted); The processing of the bridge response signals and the training of the networks was done in MATLAB. Finally, better results were obtained with the Multilayer Perceptron, noting that the Network is capable of predicting damage.

KEY WORDS: Damage identification, artificial neural networks, pedestrian bridge.

3. INTRODUCCIÓN

En la actualidad la inteligencia artificial es utilizada en diversos campos y puede ser definida como la automatización de actividades que se vinculan al pensamiento humano, como la toma de decisiones, resolución de problemas, aprendizaje, etc. (Bellman, 1978). Una de las áreas de la inteligencia artificial es la Subsimbólica, que consiste en sistemas genéricos que se van adaptando para ser capaces de resolver problemas por sí mismos. A esta área pertenecen las Redes Neuronales Artificiales (RNA), las cuales pretenden emular los sistemas neuronales de los animales y se definen como algoritmos de procesamiento de datos compuestos por neuronas interconectadas entre sí para formar una red compleja (Isasi & Galvan, 2004).

Las RNA tienen una topología distintiva que depende del número de capas, neuronas, entre otros, que son necesarios para posteriormente entrenarla para una determinada aplicación y así tener una arquitectura concreta. Hoy en día las RNA son muy utilizadas para el procesamiento de señales e imágenes, sin embargo, también se han venido usando para detectar fallas. La detección del daño oportuno en las estructuras civiles como edificios, puentes, presas, etc., es un tema de común interés para los ingenieros. Es por esto que el siguiente trabajo tiene como objetivo identificar daño estructural en un puente peatonal con RNA, para lo que se probaron diferentes arquitecturas para dos modelos de RNA.

4. PLANTEAMIENTO DEL PROBLEMA

Descripción

Las estructuras civiles son sistemas dinámicos, ya que cambian con la influencia de factores externos, como los sismos, la influencia del ambiente, agentes corrosivos, etc. Estos factores hacen que la resistencia y estabilidad de una estructura llegue a sus límites, pero no se conoce el tiempo en el que esto pueda suceder, ya que en las estructuras el daño no se identifica fácilmente, por el contrario, solo se hace evidente cuando está muy avanzado. Tales evidencias pueden ser fisuras, deflexiones excesivas, humedad, oxidación, asentamiento, entre otros. Pero la aparición de estas puede no ser suficiente para identificar el daño, ni el nivel de seguridad y/o funcionalidad de la estructura. Esta situación es preocupante cuando la estructura civil está constantemente habitada o tiene un alto valor económico y/o histórico. Si los daños estructurales no se perciben y corrigen a tiempo, podrían implicar una falla importante, generando desde pérdidas monetarias por grandes reparaciones, materiales o equipos, hasta pérdidas humanas. Sucesos como el colapso de un puente peatonal ubicado en el norte de Bogotá, dos días antes de su inauguración, mientras se realizaba una prueba de carga, con militares, dejando como resultado 42 personas heridas, es ejemplo de ello. Esta obra tuvo un costo aproximado de 3.100 millones de pesos y sería para uso privado del ejército (Figura 1).



Figura 1. Colapso de puente peatonal al norte de Bogotá, Colombia (El Espectador, recuperado de: <http://www.elespectador.com/noticias/bogota/puente-caido-contrato-avaluaron-dos-subtenientes-articulo-541669>).

Es por esto que surge la necesidad de utilizar metodologías para analizar los cambios de los parámetros dinámicos de las estructuras por medio del monitoreo, para identificar a tiempo los daños y tomar medidas de prevención y tratamiento. Una de estas técnicas son las redes neuronales artificiales, quienes al estar bien entrenadas y con una topología específica según la necesidad, pueden aprender a realizar diferentes funciones, simulando el funcionamiento del cerebro, como por ejemplo, identificar daño en una viga simple, lo que se realizó por el ingeniero Mahu, quién detectó daño (con presencia de algunos falsos positivos y negativos) usando como entrada, a una red Perceptrón Multicapa, las frecuencias de antiresonancia, mostrando la eficacia de las redes para identificar daño en una estructura simple (Mahu, 2012). Sin embargo, como ya se mencionó, es importante conocer el daño en estructuras más complejas y de constante uso, para lo que se pueden utilizar diferentes tipos de redes neuronales artificiales y a su vez, diferentes entrenamientos, como se analiza en el artículo “Detección de daño estructural utilizando redes neuronales artificiales: Un estado del Arte”, donde se recopila información sobre las

diferentes investigaciones del tema y se concluye que la mayoría de trabajos utilizan redes de tipo Perceptrón Multicapa, sin justificar su escogencia, igualmente resalta la importancia de seleccionar un adecuado método de entrenamiento, pues juega un papel importante en la capacidad de detección de daño (Villalba et al, 2017).

Formulación

¿Cuál es la efectividad de dos modelos de redes neuronales artificiales, con diferentes tipos de entrenamiento, para identificar daño estructural en un puente peatonal?

Objetivos

Objetivo General

Evaluar la efectividad de las redes neuronales artificiales para identificar daño estructural en un puente peatonal.

Objetivos Específicos

- Definir y codificar los algoritmos de dos tipos de redes neuronales, con diferentes tipos de entrenamiento, para la identificación de daño.
- Evaluar el desempeño de dos modelos de redes neuronales artificiales mediante simulaciones numéricas.
- Evaluar, a partir de datos experimentales, el desempeño de dos modelos de redes neuronales artificiales para identificar daño estructural en un puente peatonal.

Justificación

En el año 2015, en Santiago de Cali, Colombia, el Grupo de Investigación en Ingeniería Sísmica, Eólica, Geotécnica y Estructural, G-7, de la Escuela de Ingeniería Civil y Geomática de la Universidad del Valle, mediante un contrato interadministrativo con la Secretaría de Infraestructura y Valorización de la alcaldía local, llevaron a cabo el inventario, la georreferenciación y el diagnóstico visual de 682 puentes y 138 Box Culverts, que componen la red vial de la zona urbana y rural de la ciudad. En dicho proyecto se determinó visualmente el estado estructural y funcional de las estructuras mencionadas (Tabla 1 y Tabla 2).

Tabla 1. Inventario de puentes estudiados (Modificado de: G-7 & CARSIG, 2015).

Tipo	Número de puentes
Peatonal	388
Vehicular	272
Mixto	16
Férreo	6
Total	682

Tabla 2. Inventario de Box Culverts Inspeccionados (Modificado de: G-7 & CARSIG, 2015).

Inspección visual	90
Georreferenciación y registro fotográfico	48
Total	138

En este estudio se clasificó la necesidad de inspeccionar los puentes revisados, en baja, media y alta, con lo que se pudo observar que la necesidad de inspección para los puentes vehiculares es media en un 53,85% para la parte estructural y en un 30.04% para su funcionalidad. Mientras que los puentes férreos tienen una necesidad alta de inspección funcional del 66,67% y de inspección estructural media del 100%. Y en los puentes peatonales se determinó una necesidad media de inspección funcional del 47,15% y del 45,34% de inspección estructural. Lo anterior evidencia la necesidad de implementar

técnicas de detección de daño basado en sistemas de monitoreo para garantizar la seguridad y funcionalidad de las estructuras.

En Minnesota, Estados Unidos, colapsó el puente I-35W sobre el río Mississippi (tercer puente más transitado del estado, con un tráfico de aproximadamente 140.000 vehículos por día), el día 1 de agosto de 2007, debido a un mal cálculo de cargas (Figura 2). En tal fecha el puente estaba en mantenimiento, por lo que solo estaban en operación 4 de sus 8 carriles. Sobre los demás carriles se encontraba una sobrecarga adicional (suministros de construcción, maquinaria, etc.) que sumada a la del tráfico del momento, produjo el colapso, dejando 13 víctimas fatales y 145 personas heridas.



Figura 2. Puente I-35W del río Mississippi, cuatro días después del colapso (Civil Engineering, recuperado de: <https://twitter.com/engineeringcivi/status/561497739658534913>).

Este tipo de situaciones, tanto en Colombia como en el exterior, alertan sobre la necesidad de conocer la existencia del daño en su etapa más temprana, con el fin de disminuir el riesgo de los usuarios y evitar pérdidas económicas. La implementación de sistemas de identificación de daño estructural permite tener un conocimiento del nivel de funcionalidad de la estructura e identificar los diferentes factores que la afectan. Como, por ejemplo, la oportuna inspección visual realizada el 30 de septiembre de 2011 a la estructura del Puente Pudahuel, en Santiago de Chile, que permitió detectar una grieta en una de sus vigas, la cual fue creciendo con el paso de las horas, obligando a su cierre (Dirección general de concesiones, 2011). Gracias a que la falla se detectó a tiempo y se detuvo el tráfico, no se

presentaron daños en la losa del puente evitando una contingencia mayor (El Mercurio, 2011).

Por lo que existen los métodos indirectos de detección daño, dentro de los que se encuentran las redes neuronales artificiales, con los cuales se han obtenido muy buenos resultados, como es el caso de la ya mencionada tesis de grado “Detección de daños en una viga simple mediante antiresonancia y redes neuronales”, donde consiguió se identificar daño en una viga simple, con muy buenos resultados, de igual forma esta metodología fue utilizada en la tesis de grado de Panesso, 2017, siendo la que menos identifico daño donde realmente no lo había.

5. MARCO DE REFERENCIA

Antecedentes y estado del arte

En 1996, Wong, Thint y Tung presentaron un algoritmo para identificar el daño estructural causado por la vibración del suelo. Para esto tomaron datos de respuesta en tiempo real, que fueron procesados por plantillas de redes neuronales que analizan la similitud de respuesta con los tipos de daño, guardados previamente en las plantillas, y el caso original (sin daño). Para la investigación se utilizó una estructura de acero, que es el modelo de un edificio de dos pisos, en donde se aplicaron todos los procesos, y se concluyó que el método ha evolucionado mucho y que para su uso se debe usar una memoria computacional de bastante capacidad, en la que se tengan diferentes casos de daño y así no limitar el análisis (Wong et al., 1996).

En 2001 se realizó la investigación “Detección de daño a partir de cambios en las características vibratorias usando redes neuronales”, donde se implementó una red neuronal de tipo *backpropagation* en el Neural Network Toolbox de Matlab, donde se determinó el nivel de daño de una estructura tridimensional a partir de registros de desplazamientos y sus periodos de vibración, utilizando el método analítico de elementos finitos. En este caso se entrenaron cuatro redes neuronales para identificar escenarios numéricos de daño en tres modelos estructurales, concluyendo que las redes neuronales de este tipo pueden determinar la magnitud y localización del daño, siempre y cuando la red sea correctamente entrenada. Por otra parte, se obtuvieron mejores resultados cuando se

hicieron grandes reducciones en el módulo de elasticidad y con modelos de daño simples (Hernández, 2001).

Un modelo de red neuronal fue presentado en 2007 con el fin de evaluar el estado de salud de los edificios. Para esto correlacionaron el daño en un marco rígido plano de concreto reforzado, con sus características dinámicas. En este se representó el daño por la disminución de la rigidez a la flexión, variando el módulo de elasticidad, obteniendo sus resultados a partir de un índice de daño, establecido como la relación de la rigidez después del daño y la rigidez en el estado sano. La arquitectura de la red neuronal está basada en el ensayo y error. Para minimizar el error y obtener una convergencia rápida, la red utiliza en el entrenamiento tres capas ocultas con capas de entrada y salida. La capa de entrada tiene tres neuronas que representan los parámetros de entrada, que son: la altura de los niveles de piso, el coeficiente del modo de forma y la frecuencia fundamental. Observaron que el índice de daño es proporcional a la rigidez a flexión y que para cada piso la frecuencia se reduce con el aumento del daño, y esta reducción en la frecuencia incrementa aún más el índice de daño en cada piso (Kanwar et al., 2007).

En 2014 la ingeniera Meruene publicó el estudio “Real-Time structural damage assessment using artificial neural networks and antiresonant frequencies” que consistió en utilizar redes neuronales y frecuencias de vibración para simplificar la detección, localización y cuantificación del daño estructural, con un sistema multicapa de tres capas (entrada, oculta y de salida), donde las entradas corresponden a los cambios en las frecuencias anti-resonantes, las salidas son los índices de daño de cada elemento y los nodos ocultos dependen del análisis de sensibilidad. Se utilizaron dos casos experimentales: un sistema masa-resorte de 8 grados de libertad y una viga con múltiples escenarios de daño. Aunque tuvo buenos resultados, no es aplicable a estructuras con altos niveles de amortiguación o altos niveles de ruido experimental (Meruane, 2014).

Posteriormente, en 2015 se publicó la investigación “Damage quantification method using artificial neural network and static response with Limited Sensors”, donde se propuso un nuevo método para detectar y estimar daños en las estructuras utilizando desplazamientos incompletos estáticos de una estructura dañada y redes neuronales artificiales (RNA). Este se basa en condensar las matrices de rigidez para formular respuestas incompletas estáticas como parámetros de entrada a la RNA, para lo que se utilizaron tres ejemplos, una viga continua, un puente de acero y un marco de dos pisos. Los resultados obtenidos indican que el método es viable para la detección de daños estructurales a pesar de que se modeló con datos incompletos y con ruido (Kourehli, 2015).

Más recientemente, en 2016 se publicó el artículo “Real- time damage detection based on pattern recognition”, donde los autores trabajaron con dos algoritmos de reconocimiento de patrones, una red neuronal Perceptrón multicapa y el algoritmo de agrupamiento K-means. Esta investigación se realizó para estimar la respuesta estructural del puente Guadiana, generada por las acciones ambientales y operacionales típicas, como el tráfico, temperatura, viento, etc. Se concluyó que estas metodologías son capaces de detectar pequeños cambios o daños en la estructura en tiempo real (Crémona et al., 2016).

En 2017 se culminó el trabajo de grado “Diagnóstico de salud estructural de un puente peatonal mediante el análisis de su comportamiento dinámico”, de la estudiante Angélica María Panesso Libreros en la Universidad del Valle. Panesso implementó tres técnicas de identificación de daño estructural en un puente peatonal bajo excitación ambiental. Entre las técnicas implementadas se trabajó una red neuronal Perceptrón multicapa, la cual obtuvo un desempeño satisfactorio al momento de analizar datos analíticos. Sin embargo, esta técnica no fue utilizada al momento de evaluar y comparar el proceso de identificación con datos obtenidos experimentalmente, debido a que la red se entrenó con datos analíticos con entradas y salidas, pero experimentalmente solo se tenían salidas; por lo que concluye que para aplicar esta metodología se debe rediseñar la red (Panesso, 2017).

6. MARCO TEÓRICO

Técnicas de identificación modal

Las técnicas de identificación modal se utilizan para estimar las propiedades dinámicas de una estructura (frecuencias naturales, razones de amortiguamiento y modos de vibración) (Marulanda et al., 2017). Las técnicas de análisis modal operacional se diferencian en que pueden estimar el comportamiento de la estructura a partir de vibraciones ambientales, es decir, no requieren de una entrada o fuerza artificial controlada. Dos de estas técnicas son NExT-ERA y SSI. NExT es un método que correlaciona las señales de respuesta de la estructura por vibración ambiental, con una señal de respuesta de referencia, permitiendo tratar los datos de la respuesta forzada como datos de respuesta libre, es decir, se obtiene una señal con las mismas características de los datos en vibración libre (Brownjon, 2001). ERA generalmente se utiliza posterior a NExT para la identificación modal en el dominio del tiempo y así obtener frecuencias naturales, formas modales y amortiguamientos; utilizando la relación entre la matriz de Hankel y un modelo de espacio de estados para representar la estructura (He et al., 2004). SSI es una metodología de identificación en el dominio del tiempo, que calcula las propiedades dinámicas del sistema, partiendo de la representación discreta del sistema en espacio de estados y sirve para obtener las frecuencias naturales, formas modales y razones de amortiguamiento (Giraldo et al., 2009).

Detección de daño

En general, el daño se puede definir como un cambio en las propiedades de los materiales y/o geometría de una estructura, que genera que esta se comporte de forma inusual (Doebbling et al., 1996). Lo que evidencia la importancia de conocer los diferentes estados de la estructura y comparar con lo que se supone es la condición sin daño. En ingeniería civil conocer estos estados es de vital importancia, generando la necesidad de investigar diferentes metodologías para su detección, donde se utilizan diferentes parámetros para analizar el comportamiento y la respuesta dinámica del sistema. Uno de los parámetros es el cambio en las formas modales, por ejemplo, el método Modal Assurance Criterion (MAC), es usado para comparar el cambio de las formas de vibración del estado sano y dañado para detectar si hay daño. Por su parte, el Coordinate Modal Assurance Criterion (COMAC), es una metodología que correlaciona información de diferentes modos para identificar e incluso ubicar el daño. Sin embargo, existen metodologías que no se basan en un solo criterio para identificar daño, como es el caso de las redes neuronales, las cuales se pueden alimentar de los resultados de otras metodologías como las ya descritas (Pérez et al., 2012).

Red neuronal artificial

Una Red Neuronal Artificial (RNA) es un modelo matemático, compuesto de redes interconectadas y organizadas por niveles, que pretenden simular el funcionamiento del sistema nervioso biológico, interactuando con los objetos del mundo real (Matich, 2001). La neurona artificial produce una única salida, a partir de un vector de entradas. Cada neurona se representa como un nodo j que cuenta con n entradas X , las cuales se transmiten por medio de conexiones que se ponderan por su peso sináptico w (Figura 3). El modelo matemático de la RNA consiste en combinar los valores de entrada a una unidad con los pesos de tal unidad, lo que se conoce como regla de propagación o función Neta. Esta

consiste en la sumatoria del producto $w_{ji}x_i$, más el factor de corrección bias (b). A esto se le aplica una función de activación Φ y se obtiene la salida y (ecuación 1) (Caicedo, 2017).

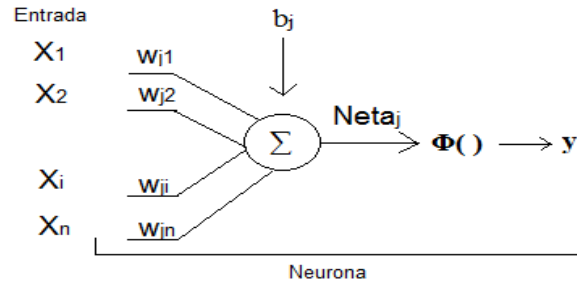


Figura 3. Elementos de una Neurona Artificial (Modificado de Caicedo, 2017).

$$Neta_j = \sum_{i=1}^n w_{ji}x_i + b_j, \quad y_j = \Phi(Neta_j) \quad (1)$$

En general, la red está compuesta por tres niveles o capas de neuronas artificiales, la capa de entrada, que es la que recibe la información del exterior, las ocultas, que reciben información de otras neuronas y que pueden tener cero o un número elevado de niveles y, finalmente, la capa de salida, quien recibe la información procesada y la devuelve al exterior (Caicedo, 2017).

Pesos sinápticos

Son la fuerza de conexión sináptica entre neuronas, que pueden ser positivos (las neuronas se excitan, es decir, si una se excita la otra también tiende a hacerlo), negativos (se inhiben, si una está activa la otra se desactiva), o cero (no hay conexión entre las neuronas). Los pesos son importantes para que la red se adapte a cualquier entorno y pueda realizar una determinada tarea, siendo necesarios para que la red adquiera conocimiento (Bertona, 2005).

Función de activación

Es la que determina el estado de activación actual de la neurona; esta transforma la entrada global en un valor de activación. Las funciones de activación más utilizadas son (Bertona, 2005):

- Lineal: $\phi(x) = x$
- Paso binario: $\phi(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases}$
- Bipolar: $\phi(x) = \begin{cases} 1 & x \geq 0 \\ -1 & x < 0 \end{cases}$
- Sigmoidea: $\phi(x) = \frac{1}{1+e^{-\sigma x}}$ y $\phi(x) = \tanh(x)$
- Sinusoidal: $\phi(x) = \sin(wx + \varphi)$
- Gaussiana: $\phi = e^{-r^2/2}$

Arquitectura de una red neuronal artificial

Las RNA tienen una forma específica de organizar las neuronas, a lo que se le llama arquitectura, que está muy relacionada con el algoritmo de aprendizaje para entrenar la red. Cuando se tiene definido el tipo de neurona a utilizar en un modelo de red, se debe indicar cuál será su tipología, es decir, la organización y disposición de las neuronas (Caicedo, 2017). Esto depende del número de capas, el grado de conectividad, el tipo de conexión y el número de neuronas por capa.

Las conexiones son la sinapsis o comunicación entre neuronas, las cuales tienen asociado un peso sináptico y una dirección. Si las conexiones son entre neuronas de una misma capa se denominan conexiones laterales o intra-capas. Si son de capas distintas se llaman conexiones inter-capas. El nivel o grado de conectividad indica si la red está completamente interconectada o si hay neuronas o capas no conectadas (Bertona, 2005). Por otra parte, dependiendo del número de capas, las redes pueden ser monocapa (una sola capa de neuronas) o multicapa (dos o más capas de neuronas) (Figura 4) y, dependiendo de cómo fluye la información entre varias capas, pueden ser redes *feedforward* (un único sentido,

hacia adelante), o recurrentes (puede haber realimentación hacia capas anteriores por medio de conexiones sinápticas) (Caicedo, 2017).

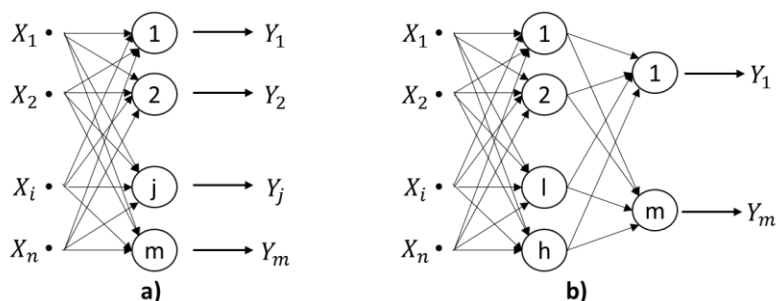


Figura 4. Estructura de una red neuronal artificial: a) monocapa b) multicapa (Modificado de Caicedo, 2017).

Aprendizaje

El aprendizaje de la RNA es la habilidad que tiene de aprender del ambiente y mejorar su desempeño, por medio de un proceso iterativo, donde se ajustan los pesos sinápticos y los niveles de los bias, proceso también denominado entrenamiento. El aprendizaje está determinado por la forma como se cambian los parámetros en el proceso (regla de aprendizaje, ecuación 2), donde se obtiene el valor actualizado del peso sináptico, a partir del valor actual del peso sináptico $w(t)$ y de la variación del peso sináptico $\Delta w(t)$ (Caicedo, 2017).

$$w(t + 1) = w(t) + \Delta w(t) \quad (2)$$

Es importante conocer cómo se modifican los valores de los pesos asignados a las conexiones, a estos criterios de cambio se les llama regla de aprendizaje. Se conocen dos grandes grupos o reglas de entrenamiento (Hilera & Martínez, 1995):

- Aprendizaje supervisado: en este el entrenamiento es controlado, ya que se le presenta a la red un conjunto de patrones de entrada con su respectiva salida esperada. Los pesos se modifican de acuerdo con el error o diferencia entre la salida real de la red y la salida esperada. En esta categoría están tres tipos de aprendizajes: por corrección de error que,

como su nombre lo indica, está en función del error (la diferencia entre lo esperado y lo obtenido de la red); el aprendizaje por refuerzo, donde los pesos se ajustan por un mecanismo de probabilidad, donde una señal de refuerzo indica si la red se ajusta (“éxito”=+1 o “fracaso”=-1), debido a que no hay una referencia exacta respecto a la salida deseada; y el entrenamiento estocástico, donde se realizan cambios en los pesos de las conexiones de la red de forma aleatoria y se comparan con el resultado deseado y/o con distribuciones de probabilidad (Mahu, 2012).

- Aprendizaje no supervisado: en este caso solo se tiene un conjunto de patrones de entrada, es decir, no hay información sobre la salida esperada. De modo que en el proceso de entrenamiento se ajustan los pesos con base en la correlación existente entre los datos de entrada. En este grupo se encuentran dos tipos de aprendizaje: el entrenamiento hebbiano, que mide la correlación de los datos de las salidas de las neuronas conectadas (si dos neuronas se activan simultáneamente, toman el mismo estado y el peso se incrementa en ambas); y el aprendizaje competitivo y comparativo, en el que las neuronas compiten unas con otras para realizar una tarea, por lo que solo una neurona se activa, siendo la vencedora (Hilera & Martínez, 1995).

La duración del entrenamiento de la red es un parámetro importante, pues si es demasiado extenso puede causar un sobre ajuste de la red, haciendo que pierda la capacidad de generalizar nuevos datos. Para acotarlo se pueden usar tres maneras y mejorar el desempeño de la red: la primera es el error objetivo, que finaliza el entrenamiento cuando el error total de la red alcanza un valor o cuando la diferencia entre una iteración y otra es menor a cierto valor objetivo; segundo, la limitación del número de iteraciones o repeticiones en que la red revisa todos los patrones de entrenamiento; y, por último, la parada temprana (“*early stopping*”) donde se considera que el set de datos de validación alcanza un máximo de error.

Métodos o algoritmos de entrenamiento de una RNA

Existen diferentes métodos de aprendizaje, como: Gradiente Descendente con Alfa variable o adaptativo, Levenberg Marquardt, Backpropagation con Regulación Bayesiana, Gradiente Descendente con momentum, Gradiente Descendente con momentum y Alfa variable. Tal variedad de aprendizajes se debe a que existen muchas aplicaciones, por lo que habrá casos donde la tasa de aprendizaje requiere de variaciones, debido a las regiones en la superficie que poseen un error singular, mientras que en otras la sintonización de los pesos sinápticos puede estancarse en los óptimos locales. A continuación, se profundiza en los que se emplean en esta investigación (Alfonso, 2017).

Backpropagation

Este algoritmo es un método de aprendizaje supervisado por corrección de error, en donde se ajustan completamente los pesos de la red antes de ponerla en funcionamiento, y es el más utilizado en redes *feedforward*. Primero se aplica un patrón de entrada que se propaga por las capas de la red hasta producir la salida de la misma. Esta salida es la que se compara con la deseada para calcular el error cometido por cada neurona de salida. Tales errores se transmiten hacia atrás, desde la capa de salida hacia las neuronas de las capas intermedias. Cada neurona tiene un error que es proporcional a su contribución sobre el error total de la red. Con este error recibido se ajustan los errores de los pesos sinápticos de cada neurona hasta obtener un error mínimo (Bertona, 2005). Este algoritmo pertenece a uno de los métodos de gradiente descendente, los cuales, a pesar de su convergencia, suelen ser muy lentos.

Gradiente Descendente con Alfa Variable o Adaptativo

La ventaja de este método es que, a diferencia del tradicional algoritmo *Backpropagation*, este no tiene una tasa de aprendizaje fija, puesto que varía de acuerdo con su aplicación y para diferentes regiones en una superficie de error singular. Esta tasa debe ser modificada durante el entrenamiento, lo que se puede hacer incrementando o disminuyendo en cada iteración dependiendo del progreso que tenga (Alfonso, 2017).

Levenberg Marquardt

Es una modificación del método de Gauss-Newton pensado para optimizar la suma de los errores cuadráticos, es decir, el error cuadrático medio. Sin embargo, este no puede ser usado como propósito general para minimizar funciones y no sirve para problemas de gran tamaño (con muchos pesos a entrenar). Es uno de los algoritmos más rápidos para entrenar redes de tamaño moderado, de tipo *feedforward*, pues aprovecha la buena convergencia de los métodos de gradiente descendente y Newton, pero mejora su eficiencia al aumentar la velocidad y variar la dirección de algoritmo, haciendo que requiera más tamaño (Alfonso, 2017).

Perceptrón multicapa

Es una extensión del Perceptrón simple (primer modelo que permite determinar los pesos sinápticos para clasificar correctamente un conjunto de patrones, a partir de ejemplos), este tiene topología compuesta por una o varias capas ocultas, una capa de entrada y una de salida. Usa funciones de activación sigmoideas para la capa oculta, lo que garantiza la capacidad de procesamiento no-lineal y su algoritmo de aprendizaje es de tipo supervisado (Bertona, 2005). En este modelo de redes, las neuronas de una capa reciben señales de entrada de otra capa anterior y envía señales a una posterior, es decir, son de tipo *feedforward*. (Hilera & Martínez, 1995). Generalmente son usadas para el reconocimiento

de patrones y para el ajuste de curvas, sin embargo, pueden ser utilizadas para la mayoría de aplicaciones de RNA, como la identificación y control de sistemas. Pueden ser entrenadas por retropropagación (*backpropagation*), utilizando un conjunto representativo de la muestra (Mahu, 2012). La topología de esta red consiste en que cada sensor de entrada está conectado con las unidades de la segunda capa, y cada unidad de proceso de la segunda capa está conectada con las de la primera y tercera capa, así sucesivamente. Las unidades de salida están conectadas solamente con las unidades de la última capa oculta, como se observa en la Figura 5 (Alfonso, 2017).

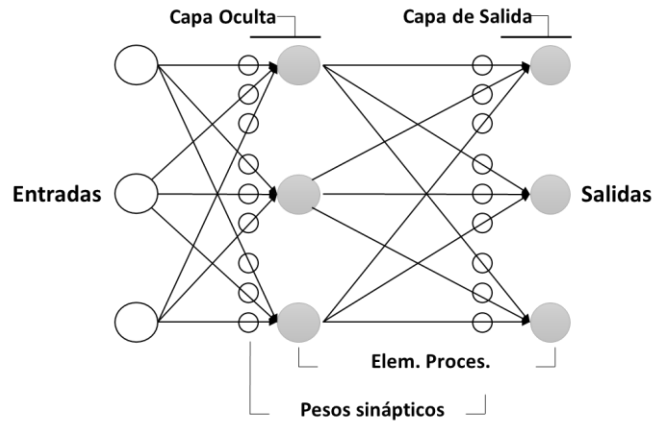


Figura 5. Estructura de una red Perceptrón multicapa (modificado de Alfonso, 2017).

El aprendizaje de este tipo de redes tiene como fin minimizar el índice de desempeño, es decir, el error global E (Ecuación 3).

$$E = \frac{1}{2} \sum_{i=1}^M (d_i - y_i)^2 \quad (3)$$

Donde M es el número de neuronas, d es la salida deseada e y es la salida de la red (Alfonso, 2017).

Redes de Base Radial (RBF)

Son redes multicapa de tipo *feedforward*, al igual que el Perceptrón multicapa, pero se diferencian en que están compuestas por una sola capa oculta y las neuronas de esta capa son de carácter local, ya que cada neurona se activa en una región diferente del espacio de patrones de entrada, es decir, la entrada es dividida por hiperesferas, para dividir el problema en varios subproblemas menos complejos, haciendo que la aproximación que tiene este tipo de redes sean combinaciones lineales de múltiples funciones locales y no lineales (Isasi & Galvan, 2004).

Se caracterizan por tener un aprendizaje veloz y simple y por su alto grado de generalización, por lo que ha sido utilizada en el reconocimiento y clasificación de patrones, y en la identificación y modelación de sistemas no lineales. La función de activación es del tipo base radial y normalmente es la función de Gauss ($\phi(r) = e^{-r^2/2}$), de modo que los parámetros característicos de la capa oculta son el centro de la función de activación (centroide) y su respectiva desviación estándar (Caicedo & López, 2009). Debido a que las capas de este tipo de neuronas realizan tareas distintas, tiene un aprendizaje híbrido, que combina dos fases: un entrenamiento supervisado en la capa de salida para determinar sus pesos y umbrales, y una fase no supervisada donde se determinan los centros y amplitudes de la capa oculta (Isasi et al., 2004).

7. REFERENTE EMPÍRICO

En esta investigación, para probar las RNA, se utilizaron los resultados experimentales del puente peatonal, objeto de estudio en el trabajo de grado de Panesso (2017), construido por el grupo de investigación G-7 en la Escuela de Ingeniería Civil y Geomática de la Universidad del Valle, el cual cuenta con un claro de 12 m y un ancho de 1.2 m (Figura 6). La estructura consta de una armadura metálica compuesta de ángulos, con unas conexiones soldadas (en los tramos extremos, cada uno de un metro) y otras pernadas (en los 10 tramos restantes), soportada por cuatro apoyos de rodillo que restringen el desplazamiento en el sentido vertical, sobre dos pórticos de concreto reforzado en cada uno de sus extremos.



Figura 6. Puente peatonal usado como referente.

Para obtener la respuesta en aceleración del puente a excitación ambiental se utilizaron seis acelerómetros sísmicos (tres sensores uniaxiales REF TEK modelo 131 B – 01/1 y un equipo triaxial REF TEK modelo 130 – SMA). Para hacer las mediciones se distribuyeron

los sensores en 22 puntos, separados cada metro (por lo que se tienen 22 grados de libertad horizontales y 22 verticales (Figura 7). La identificación dinámica de la estructura se realizó mediante análisis espectral usando la técnica de detección de picos, para obtener las frecuencias (Panesso, 2017) y se verificó usando SSI, donde además se obtuvieron el resto de propiedades dinámicas. Con esto se identificaron cinco frecuencias naturales en el estado sano de la estructura, con sus respectivos modos de vibración (Tabla 3 y Figura 8) (Panesso, 2017).

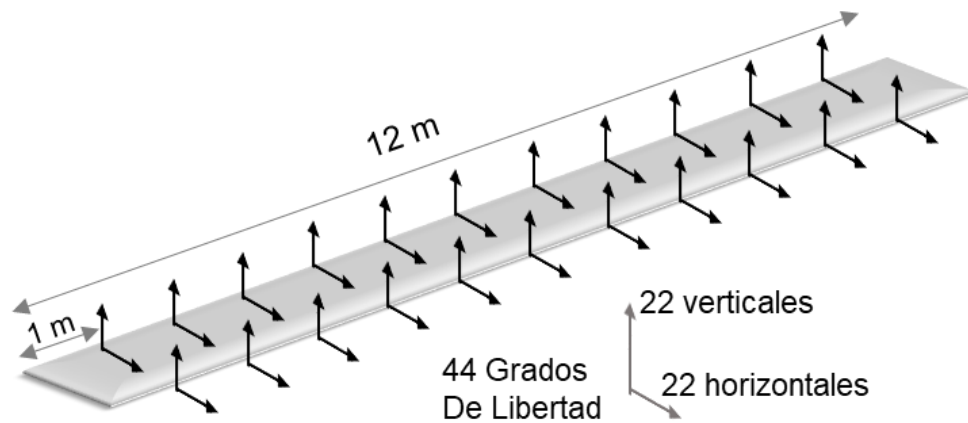


Figura 7. Distribución de los sensores en el puente.

Tabla 3. Frecuencias naturales identificadas experimentalmente del puente en estado sano

Modo	Frecuencia [Hz]
1	4.02
2	7.85
3	13.51
4	14.03
5	23.99

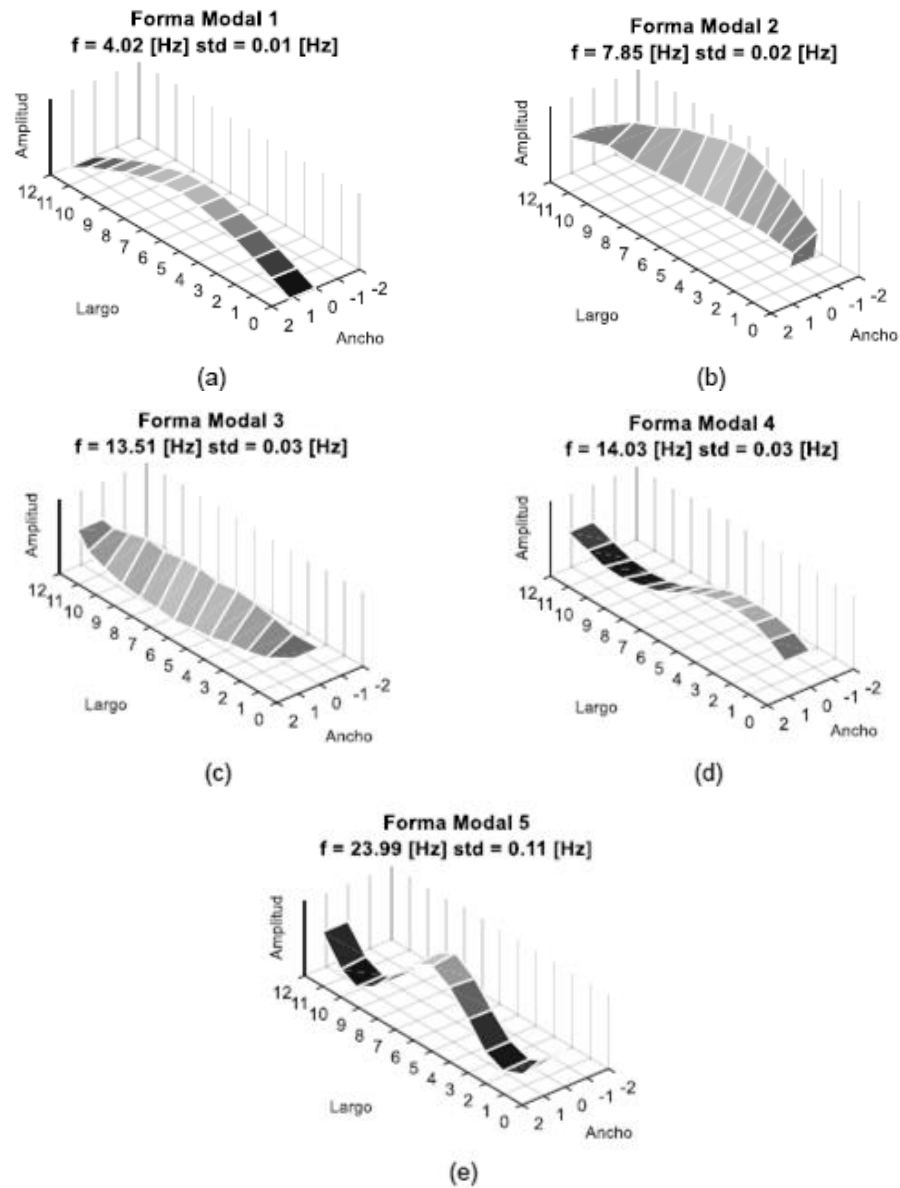


Figura 8. Modos de vibración experimentales (Modificado de Panesso, 2017).

Para simular el daño en la estructura se modificó su rigidez quitándole conexiones pernadas, modificando la configuración estructural del puente y generando los casos de daño experimentales, que se describen en la sección “Procesamiento de los datos de aprendizaje y validación”.

Modelo numérico

Se utilizó el modelo ajustado del puente desarrollado por Angélica Panesso, (2017), realizado en el programa de elementos finitos SAP2000, compuesto de elementos tipo membrana “*Shell-Thin*” para la losa y tipo “*Frame*” para los ángulos de la armadura (Figura 9).

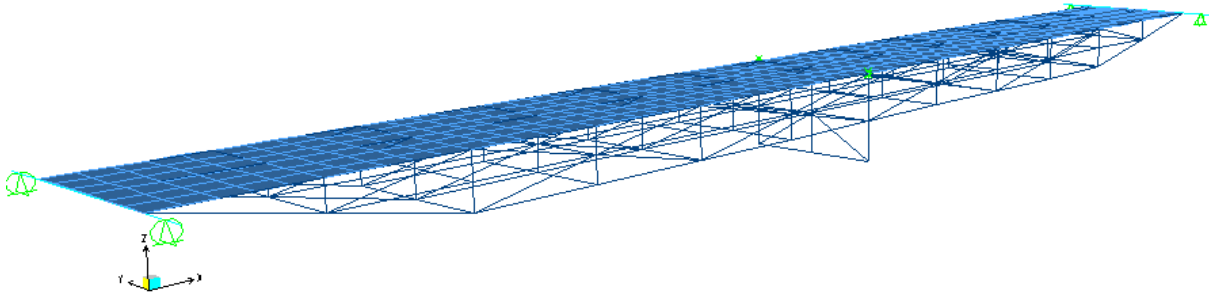


Figura 9. Modelo del puente peatonal sin daño.

Partiendo de las cinco frecuencias del puente identificadas a partir del procesamiento con detección de picos y SSI, se tiene un modelo ajustado del puente con unas diferencias mínimas de frecuencias (ecuación 4) y un MAC (ecuación 5) casi igual a uno, como se observa en la Tabla 4 (Panesso, 2017).

$$\frac{|w_{ei} - w_{ai}|}{w_{ei}} \quad (4)$$

$$MAC(\varphi_{ei}, \varphi_{ai}) = \frac{|\varphi_{ai}^T \varphi_{ei}|^2}{\varphi_{ai}^T \varphi_{ai} \varphi_{ei}^T \varphi_{ei}} \quad (5)$$

Donde w_{ei} es la i -ésima frecuencia experimental, w_{ai} es la i -ésima frecuencia del modelo, φ_{ei} es la i -ésima forma modal experimental y φ_{ai} es la i -ésima forma modal del modelo.

Tabla 4. Ajuste del modelo. Modificado de Panesso (2017)

Modo	Diferencia en frecuencia [Hz]	1 - MAC
1	0.0001	0.0025
2	0.0673	0.0167
3	0.0001	0.0393
4	0.0222	0.0551
5	0.0317	0.0268
Promedio	0.0243	0.0281

8. EVALUACIÓN DE LAS TÉCNICAS

Metodología

Para llevar a cabo este trabajo se utilizó la metodología descrita a continuación:

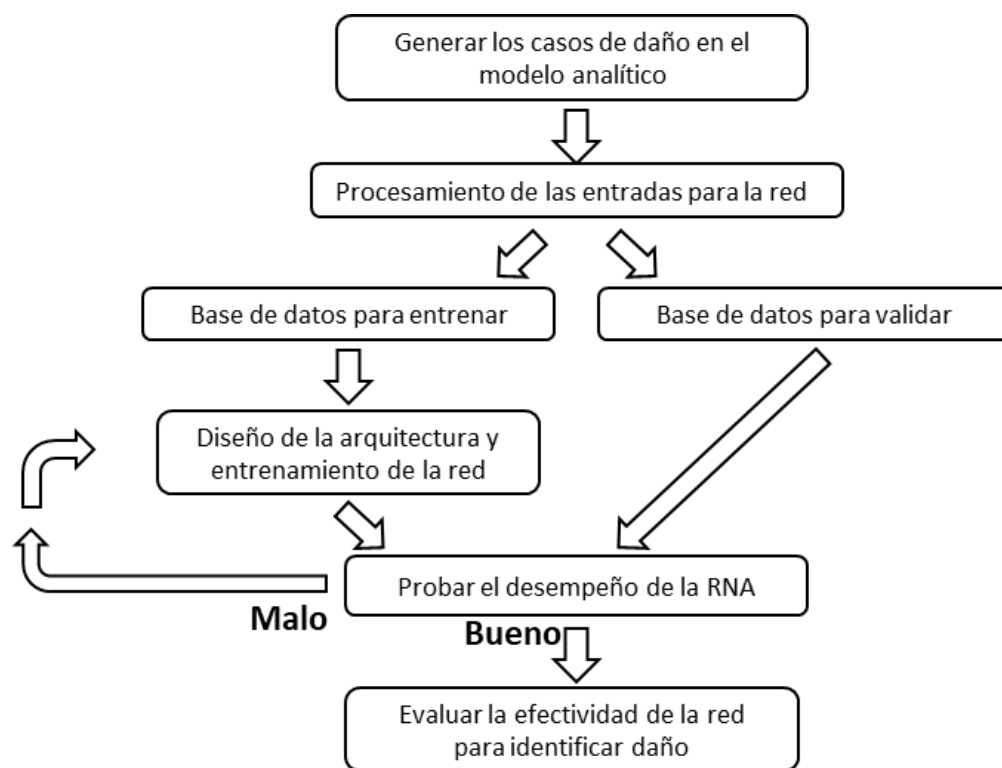


Figura 10. Diagrama de flujo de la metodología seguida.

- Generar casos de daño en el modelo analítico: esto se realizó en el modelo de elementos finitos, donde se simuló daño estructural al quitar aleatoriamente uno o varios elementos “Frame”, en los distintos diez tramos pernaados del puente (también seleccionados aleatoriamente). Con esto se generaron los daños para el

entrenamiento y para probar la capacidad de las redes al resolver el problema planteado, es decir, los datos de validación analíticos.

- Procesamiento de las entradas para la red: se probaron diferentes tipos de entradas, que se explicarán más adelante.
- Diseño de la arquitectura y entrenamiento de la red: en esta fase se definió la estructura de la red y su método de entrenamiento o aprendizaje, para lo que se siguió la metodología de entrenamiento descrita en la siguiente sección.
- Probar el desempeño de la RNA: después de entrenar las redes, son evaluadas con la matriz de confusión, usando el grupo de datos de validación (tanto analíticos como experimentales), que son nuevos para la red y así probar su precisión para identificar daño. Según fueron los resultados, se volvió a entrenar la red variando diferentes características como el número de datos de entrada, de casos a entrenar o la topología de la red.
- Evaluar la efectividad de las redes para identificar daño: luego de realizar diferentes pruebas, se analizaron los resultados de la red para los datos tanto analíticos como experimentales.

Metodología para entrenar las RNA

Al entrenar las redes se busca minimizar la función de desempeño de la red (el “*Performance*” en MATLAB), que para las redes *feedforward* está predefinida como “*mean-squared error*, MSE”, es decir, el promedio del error cuadrático entre los rendimientos de la red y los rendimientos designados, o también llamado error de validación. Para realizar el entrenamiento de las RNA se utilizó el *toolbox* de MATLAB para Redes Neuronales, con el cual se definió la estructura de las redes, influyendo

directamente en el tiempo de entrenamiento y en la precisión de la red. Por esto, para los dos tipos de redes, se tuvo en cuenta los siguientes parámetros:

- Criterio de parada: las redes se siguen entrenando hasta que el error de validación aumente 6 veces de manera consecutiva (guardando el error más bajo), o hasta que exceda un número de 10000 iteraciones (*epochs*), que indican el número de veces en que se presentan los vectores de entrenamiento entrada-salida a la red para el cálculo de pesos y desvíos.
- Número de neuronas: para el Perceptrón se varió de manera manual el número de neuronas en la capa oculta, buscando el óptimo. Mientras que para las RBF el programa aumenta el número de neuronas en la capa de base radial hasta obtener un error menor que el esperado.

Perceptrón Multicapa

Específicamente para este método se modificaron los siguientes parámetros:

- Métodos de entrenamiento: se probaron cinco métodos de entrenamiento: Gradiente descendente (*traingd*), Gradiente descendente con eta variable (*traingda*), Gradiente descendente con momentum (*traingdm*), Gradiente descendente con momentum y eta variable (*traingdx*) y Levenberg Marquardt.
- Función de activación: para generar una salida, las neuronas pueden usar diferentes funciones de transferencia, por lo que se probó la función logarítmica ('log-sigmoid', que genera salidas entre 0 y 1, de forma sigmoidea), la función tangencial ('tan-sigmoid', la cual da salidas sigmoideas entre -1 y 1), y la función de traslado lineal ('purelin', donde la salida puede tomar cualquier valor). Sin embargo, en la capa oculta se recomienda que la función incluya en su recorrido los valores objetivo de la red (que en este caso son 0 para el estado sano y 1 para el estado dañado), por lo que se usó la función sigmoidea tangente ("tansig"). Además, para solucionar problemas de clasificación y reconocimiento de patrones, lo

recomendable es utilizar una función de activación sigmoideal en la salida con el fin de garantizar una salida selectiva, por lo que también se utilizó “tansig” (Mahu, 2012).

- Porcentaje de utilización de los patrones de entrenamiento: el *toolbox* requiere que se divida el total de patrones en tres conjuntos: entrenamiento, validación y testeo. El primer conjunto es el que se utiliza en los algoritmos de entrenamiento para optimizar la red, el segundo es para los criterios de parada y participa de manera indirecta en el entrenamiento de la red, y el tercero se utiliza para probar el error de la red al enfrentar patrones que no hayan participado directa o indirectamente en el entrenamiento.

Redes de Base Radial

Para variar su entrenamiento se cambiaron los siguientes parámetros:

- *Goal*: es el error cuadrático medio.
- *SPREAD*: es el parámetro de propagación y se modificó buscando que fuese lo suficientemente grande para que las neuronas RBF respondan a las regiones superpuestas del espacio de entrada, pero no tanto para que todas las neuronas respondan esencialmente de la misma manera.

Procesamiento de los datos de aprendizaje y validación (entrada a la red)

El conjunto de datos a utilizar debe ser representativo y no debe tener ninguna tendencia (Caicedo & López, 2009), por esto se les realizaron diferentes procesamiento a las señales del estado sano de la estructura y de diferentes escenarios de daño, para usar como datos de entrada a la red y para validarla. Específicamente para entrenar la red, en el primer caso (ceros de la función de transferencia), se simularon en el modelo numérico 188 escenarios de daño, sin embargo, se decidió aumentar el número de patrones para reducir tanto el error de entrenamiento como de validación, usando finalmente 213 escenarios de daño analíticos, compuestos por 100 escenarios con más de un tramo dañado, 112 escenarios con un solo tramo dañado, y el estado sano. Aunque el daño se encuentre en un mismo tramo, el escenario varía, ya que el o los elementos retirados son diferentes, haciendo que la respuesta de la estructura cambie. Para validar la red numérica y experimentalmente se usaron los cinco escenarios de daño disponibles en la estructura real (Panesso, 2017), donde en cuatro de ellos se seleccionó el tramo con el criterio de mayor esfuerzo y el otro fue seleccionado de forma aleatoria. En cada caso se retiró un solo elemento de manera acumulada, obteniendo al final cinco elementos dañados (Figura 11 y Figura 12), que son representados gráficamente en la Figura 13, siendo este el resultado esperado de las redes.

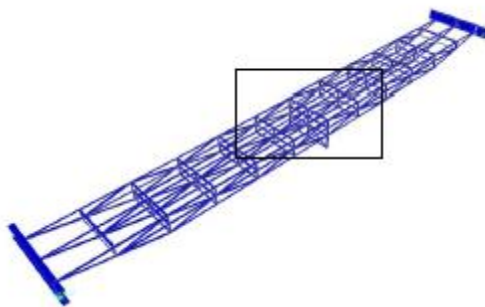


Figura 11. Escenario sin daño (Panesso, 2017).

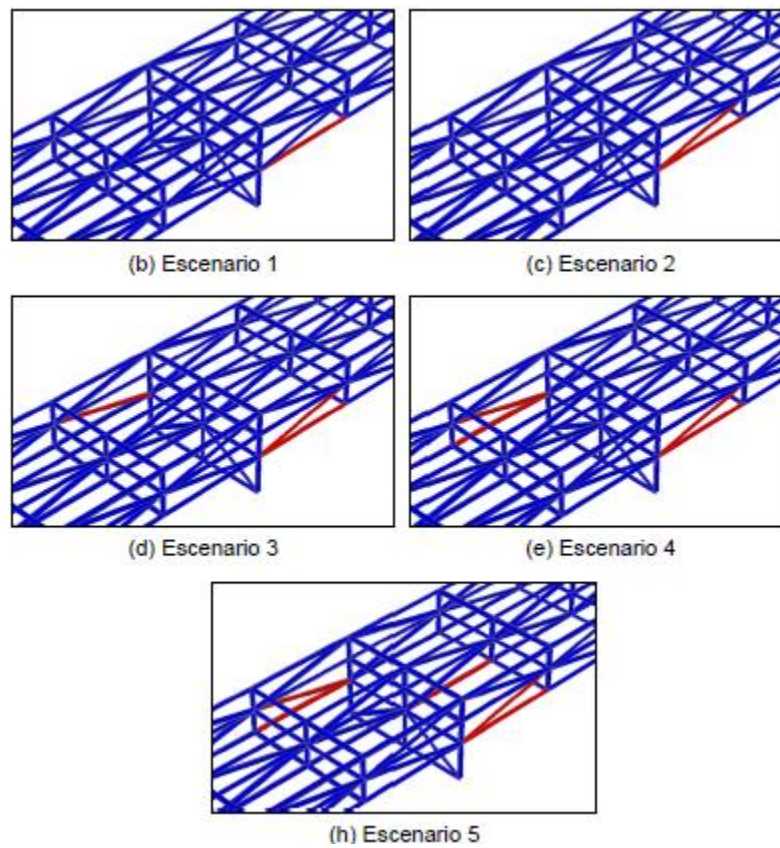


Figura 12. Escenarios de daño para validar (Panesso, 2017).

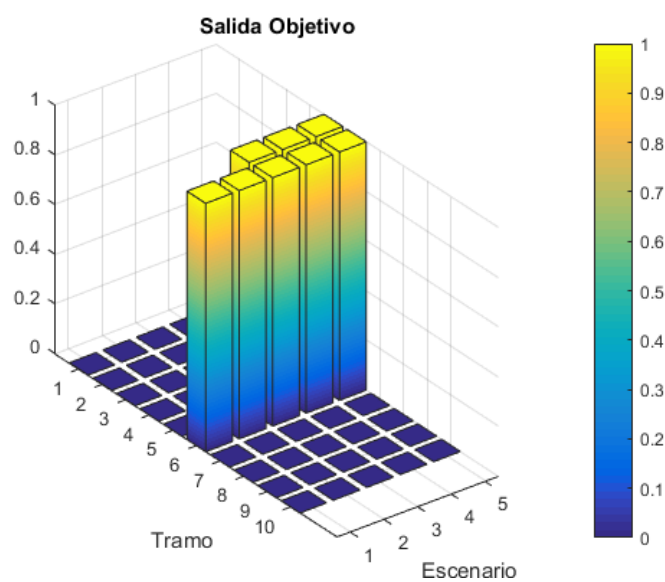


Figura 13. Representación de los daños seleccionados para validar las redes.

Diferencia de los ceros de la función de transferencia

En primer lugar, se simularon ensayos de vibración forzada y se utilizaron las aceleraciones del modelo analítico como respuesta a un Shaker (excitador dinámico), donde los registros de aceleración provienen de 13 acelerómetros verticales distribuidos sobre la estructura. Las señales fueron adquiridas a una frecuencia de muestreo de 100 Hz, se transformaron al dominio de frecuencia mediante la transformada discreta de Fourier, y se calculó la derivada del espectro de la función de transferencia, obteniendo los puntos donde la pendiente es cero, es decir, se tienen las frecuencias y amplitudes de los polos y los ceros (Figura 14 y Figura 15), para posteriormente extraer los ceros y determinar las diferencias del estado sano de la estructura con los escenarios de daño; consiguiendo tres puntos por cada sensor (puntos color rojo en la Figura 15). Con esto se obtuvo las coordenadas en frecuencias y amplitudes de los ceros, de tal forma que se tienen tres datos por 13 sensores (39 características), para cada escenario de daño (188 escenarios, al no necesitarse una base de datos muy amplia por ser características menos complejas, ya que la señal es controlada), obteniendo así una matriz de 188 x 39 para entrenar la red.

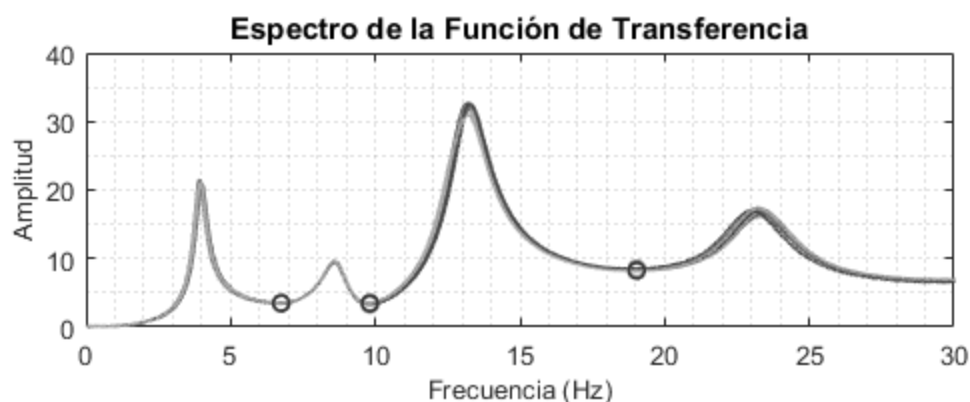


Figura 14. Polos y ceros del espectro de la función de transferencia.

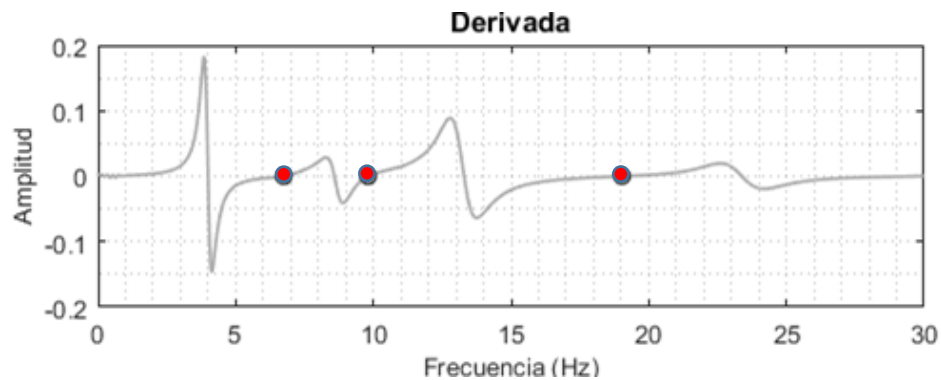


Figura 15. Derivada de la función de transferencia, con los ceros de la función de transferencia.

Espacio de estados

Sabiendo que el Estado de un sistema dinámico es el conjunto mínimo de variables en un instante t_0 que se necesita (junto con la señal de entrada en $t \geq t_0$) para conocer el comportamiento y la evolución del sistema en $t \geq t_0$ (Rodríguez et al., 2005); se puede concluir que a partir de tales variables, se puede dar una aproximación matemática del sistema. Las variables se agrupan en el llamado vector de estado, y el espacio n -dimensional que determina los posibles valores de esas variables, se denomina espacio de estados. Para un sistema lineal, las ecuaciones que representan el espacio de estados son:

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx + Du\end{aligned}$$

Donde x es el vector de estados, u el vector de entrada, y el vector de salida, A la matriz de estados, B la matriz de entrada, C la matriz de salida y D la matriz de transmisión (Rodríguez et al., 2005).

En espacio de estados los sistemas lineales se pueden representar gráficamente por tres tipos básicos de elementos: bloque integrador, multiplicación por una matriz y bloque sumador (Domínguez et al., 2006). Por lo tanto, se plantea utilizar el vector $\dot{x}(t)$, para simular la representación gráfica del integrador (Figura 16), donde A y B pasan a ser los pesos y las bias de la red. Así que se integra la respuesta en aceleración de cada grado de libertad, producida por una excitación aleatoria en la base, con la función “trapz” de MATLAB, que se encarga de calcular la integral aproximada usando el método trapezoidal, obteniendo así un solo valor que representa el área bajo la curva, para cada daño, con respecto a un periodo de 10 minutos. Con esto se obtuvo una matriz de 213 escenarios de daño x 44 características, como entrada a la red, que permiten analizar cómo afecta la interacción de cada tramo con respecto a sus velocidades.

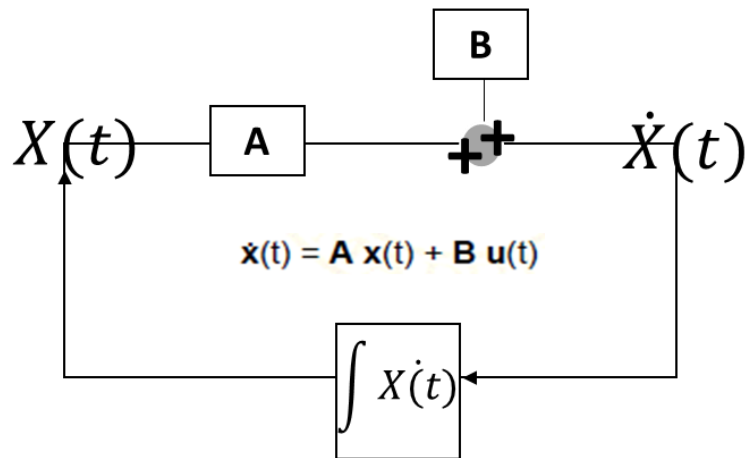


Figura 16. Representación gráfica del modelo de espacio de estados con un integrador.

Modos y frecuencias

Dado que, como ya se mencionó anteriormente, diferentes metodologías utilizan el cambio en las formas modales para identificar daño, se analizaron los cambios en las formas modales y en las frecuencias (comparando con las cinco identificadas experimentalmente).

Amplitudes de la función de transmisibilidad

Caso 1

Se calculó la función de transmisibilidad entre las respuestas en aceleración del puente (producida por una excitación aleatoria en la base) a partir de salidas, es decir, la relación entre los 44 grados de libertad medidos en el puente, para posteriormente obtener los valores de amplitud de cada una de las cinco frecuencias identificadas. Con esto se obtuvo una matriz de 44 x 44 para cada frecuencia, que se normaliza con respecto a la máxima amplitud. Al analizar gráficamente estos valores de amplitud en cada modo, y comparando lo obtenido analíticamente con lo experimental (Figura 17), se observó una diferencia en el modo 4, que genera una mala respuesta en el entrenamiento, por lo que se decidió no usarlo. Se calcularon las desviaciones del valor absoluto de las diferencias de las cuatro matrices, correspondientes a las cuatro frecuencias en cada escenario de daño, respecto a las cuatro matrices correspondientes a las cuatro frecuencias del estado sano, normalizadas con respecto a las mismas. Posteriormente, se determinaron las posiciones donde la desviación fue mayor a cinco (número al que se llegó luego de realizar un análisis de sensibilidad y hacer varias pruebas); y así se obtuvieron 97 posiciones, con las que se conformó un vector de 97 amplitudes para cada uno de los 213 escenarios de daño.

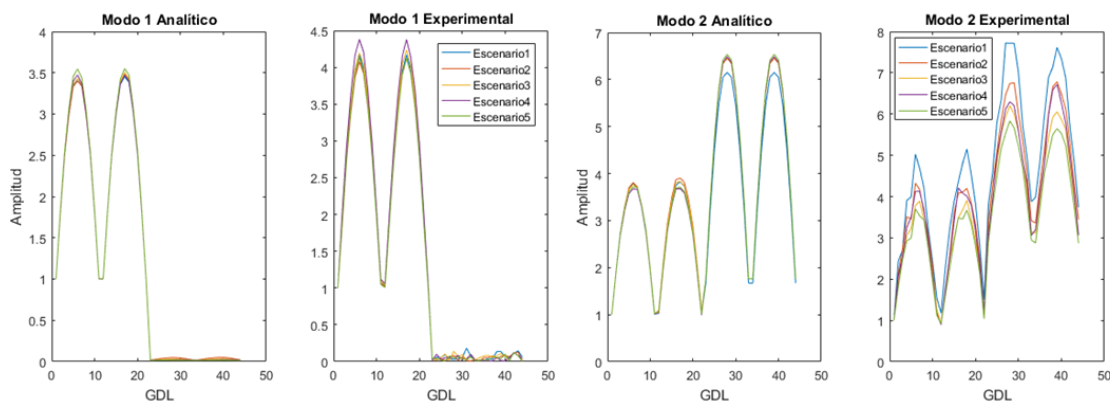


Figura 17. Comparación de la amplitud de la funciones de transmisibilidad de los grados de libertad medidos, en relación al grado de libertad 1, de los modos analíticos y experimentales.

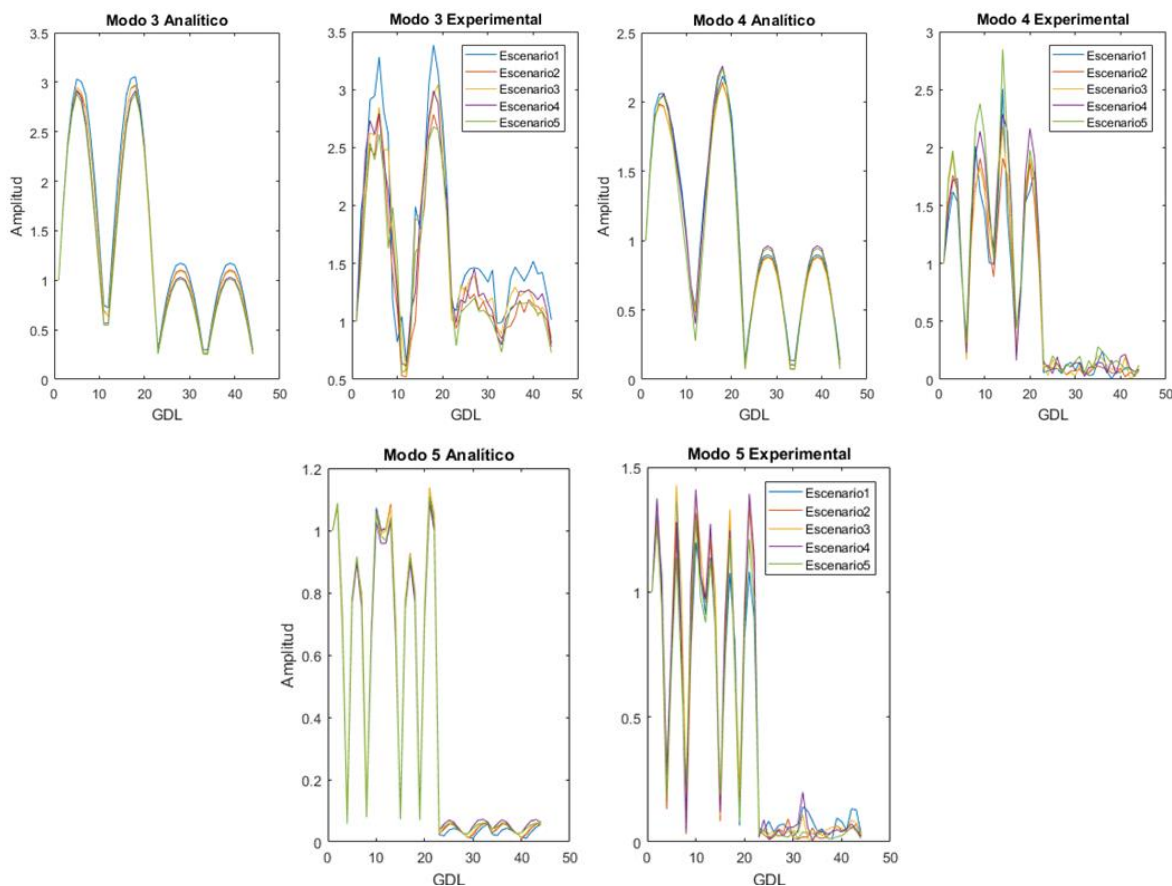


Figura 17. Comparación de la amplitud de la funciones de transmisibilidad de los grados de libertad medidos, en relación al grado de libertad 1, de los modos analíticos y experimentales.

Caso 2

Tomando como referencia el caso anterior, se parte de la misma matriz total de $44 \times 44 \times 213$ escenarios de daño, compuesta de las amplitudes de función de transmisibilidad, pero esta vez sin el absoluto de las diferencias en amplitud entre las matrices daño y la sana, es decir, se procedió de nuevo a solo utilizar los valores de amplitud que diferían más con el estado sano, sumando o restando las diferencias (según sea el signo), para encontrar las posiciones donde la desviación estándar es mayor a cinco. Dando como resultado una matriz de 114×213 .

Resultados

Para probar el desempeño de las redes neuronales, se hizo uso de la función “plotconfusion” de MATLAB, que grafica la matriz de confusión, la cual, como su nombre lo indica, muestra las formas en las que se “confunde” el clasificador, es decir, genera el porcentaje de error y acierto de la red para predecir daño en cada tramo y el porcentaje total de acierto y error. En ella, las filas representan el número de casos reales de daño en cada tramo y las columnas son el número de casos predichos.

Diferencia de los ceros de la función de transferencia

Partiendo de lo realizado por Panesso (2017), se utilizaron las diferencias de los ceros de la función de transferencia entre salidas, como la entrada para entrenar la red de tipo Perceptrón multicapa, donde se varió el número de neuronas en la capa oculta para diferentes tipos de entrenamiento, buscando la topología óptima:

Tabla 5. Mejores topologías para la primera entrada, por cada entrenamiento.

Método de entrenamiento	Gradiente descendente	Gradiente descendente con momentum y eta variable	Levenberg Marquardt
No. de neuronas en capa oculta	6	8	8
Iteraciones	10000	168	66
Best Validation Performance	0.073163	0.057627	0.000732
Éxito en datos de entrenamiento	50%	62.80%	100%
Éxito en datos de validación	40%	40%	80%

En la Tabla 5 se evidencia que la eficiencia del método de entrenamiento Levenberg Marquardt es superior a los otros, pues con un número menor de iteraciones, comparado con los otros métodos, obtiene un menor Performance y el éxito obtenido de las matrices de confusión es mayor, con un número similar de neuronas, por lo tanto, se escogió este método para continuar el entrenamiento. Finalmente, se seleccionó una topología compuesta de 8 neuronas en la capa oculta, con los siguientes parámetros de entrenamiento:

- Conjunto de entrenamiento: 60%.
- Conjunto de validación: 20%.
- Conjunto de testeo: 20%.

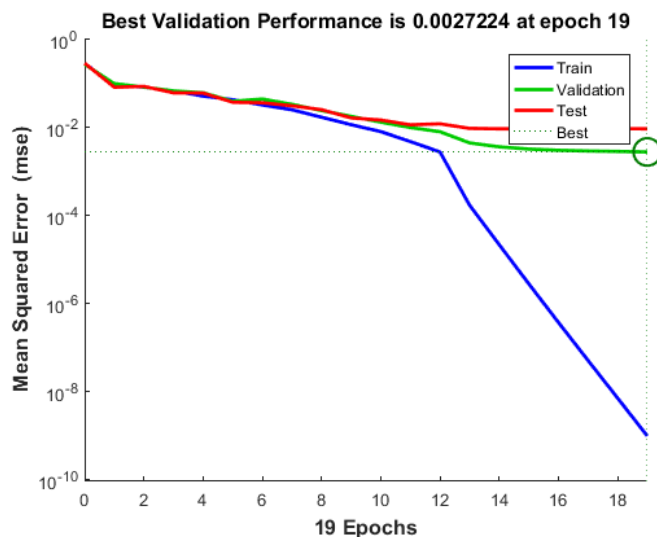


Figura 18. Evolución del error usando la diferencia de los ceros de la función de transferencia.

Aunque una topología compuesta de 8 neuronas, puede generar un poco de complejidad computacional, este sacrificio permite tener un porcentaje de éxito y clasificación mayor, sin incrementar de una manera exagerada la cantidad de neuronas utilizada. Con esta red se obtuvo un mejor Performance con un error mínimo de validación de 0.27%, pero se realiza una parada temprana en 19 iteraciones, debido a que se presentaron seis errores seguidos de validación, esto se hace con el fin de evitar una separación de las curvas de entrenamiento,

validación y prueba, y obtener una buena capacidad de generalización (Figura 18). Con las matrices de confusión se obtuvo un acierto del 100% en predecir el daño usando los mismos datos con los que se entrenó (Figura 19) y un 80% en la matriz de confusión con los datos de validación analíticos, por lo que se pueden identificar los cinco escenarios, pero no perfectamente (Figura 20), lo que muestra que la red tiene una alta efectividad para identificar el daño por tramos en el modelo analítico, considerando que en este se tiene la entrada al sistema, una excitación controlable, sin embargo, experimentalmente solo se tienen las salidas del sistema (ambientales). Es por esto que no se puede concluir sobre la efectividad de esta entrada para detectar daños reales, pues al no haber una entrada al sistema, la función de transferencia se debe realizar entre salidas, pero las señales con las que se cuenta para este trabajo son muy ruidosas, por lo que los ceros de esta función no son muy apreciables.

Confusion Matrix											
Output Class	1	2	3	4	5	6	7	8	9	10	
	12 6.4%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100%
	0 0.0%	16 8.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100%
	0 0.0%	0 0.0%	24 12.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100%
	0 0.0%	0 0.0%	0 0.0%	24 12.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100%
	0 0.0%	0 0.0%	0 0.0%	0 0.0%	24 12.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100%
	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	28 14.9%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100%
	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	20 10.6%	0 0.0%	0 0.0%	0 0.0%	100%
	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	16 8.5%	0 0.0%	0 0.0%	100%
	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	8 4.3%	0 0.0%	100%
	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	16 8.5%	100%
	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
Target Class											
	1	2	3	4	5	6	7	8	9	10	

Figura 19. Matriz de confusión con los datos de entrenamiento, usando la diferencia de los ceros de la función de transferencia.

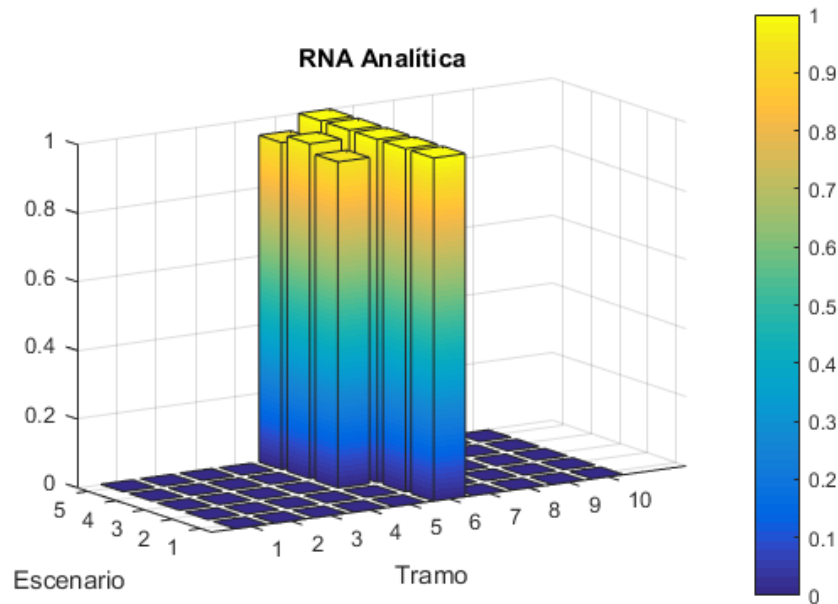


Figura 20. Daños analíticos encontrados por la red tipo Perceptrón Multicapa, usando como entrada la diferencia de los ceros de la función de transferencia.

Para las redes de Base Radial se seleccionaron los siguientes parámetros de entrenamiento:

- *Goal*: se obtuvo un “MSE” de 0.0011.
- *SPREAD*: el parámetro de propagación fue de 20.
- Número de neuronas: se requirieron 50 neuronas para alcanzar el MSE obtenido.

Sin embargo, no se obtuvieron tan buenos resultados como con el Perceptrón pues, aunque se obtuvo un 90% de acierto en la matriz de confusión con los mismos datos de entrenamiento, con los datos de validación solo acertó un 40%, por lo que no se identifican claramente los cinco escenarios de daño, como se observa en la Figura 21.

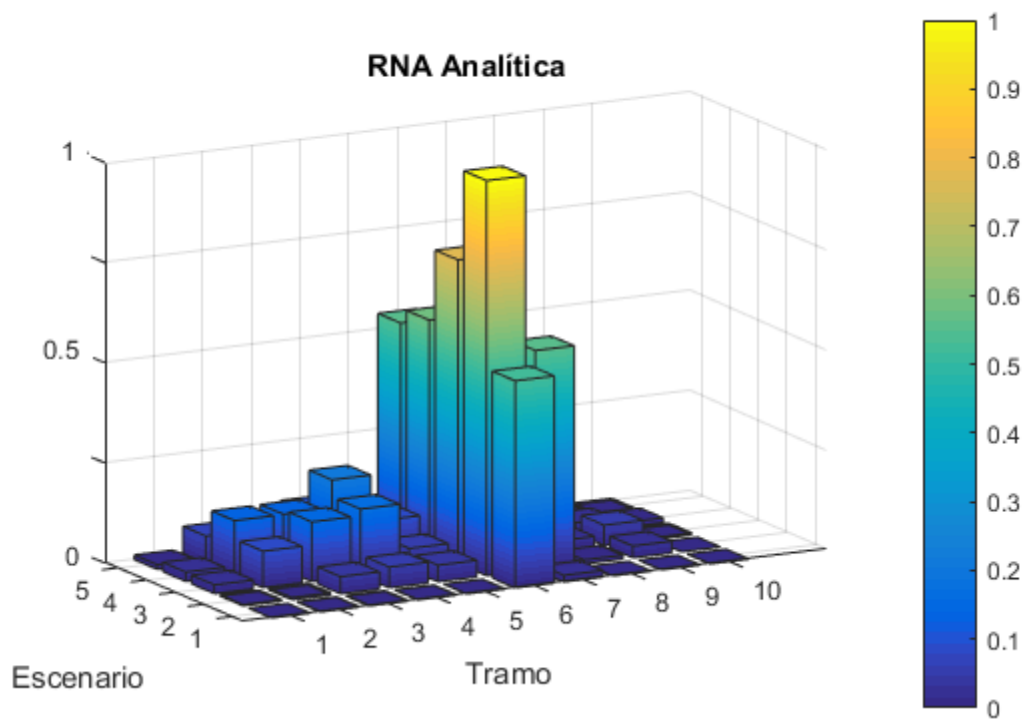


Figura 21. Daños analíticos encontrados por la red de base radial, usando como entrada la diferencia de los ceros de la función de transferencia.

Espacio de estados

Para entrenar la red usando las velocidades obtenidas para cada grado de libertad se utilizó:

- Número de neuronas en la capa oculta: 9
- Conjunto de entrenamiento: 65%.
- Conjunto de validación: 20%.
- Conjunto de testeo: 15%.

Con esto se obtuvo un buen Performance (9.5%), y se puede decir que se obtuvo un buen entrenamiento pues las tres curvas (validación, prueba y entrenamiento) están muy cercanas, presentando un comportamiento casi asintótico (Figura 22).

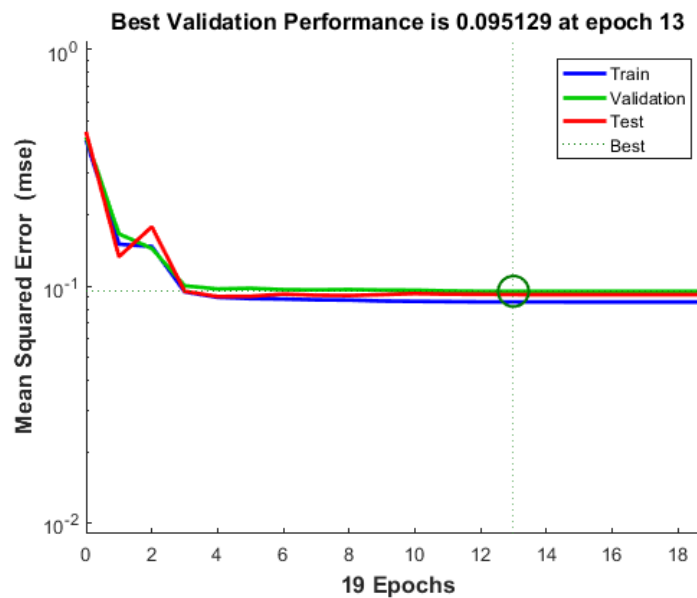


Figura 22. Evolución del error usando el vector de estados x .

Sin embargo, en las matrices de confusión solo se obtuvo un acierto del 20% con los datos de entrenamiento, mientras que con los datos de validación se obtuvo un acierto del 40%. Como se observa en la Figura 23, se pueden identificar los daños existentes, sin embargo, predice daño en donde no lo hay, lo que puede ser porque las señales utilizadas tienen una baja relación señal-ruido; así que se decide descartar esta entrada y su evaluación por redes de Base Radial.

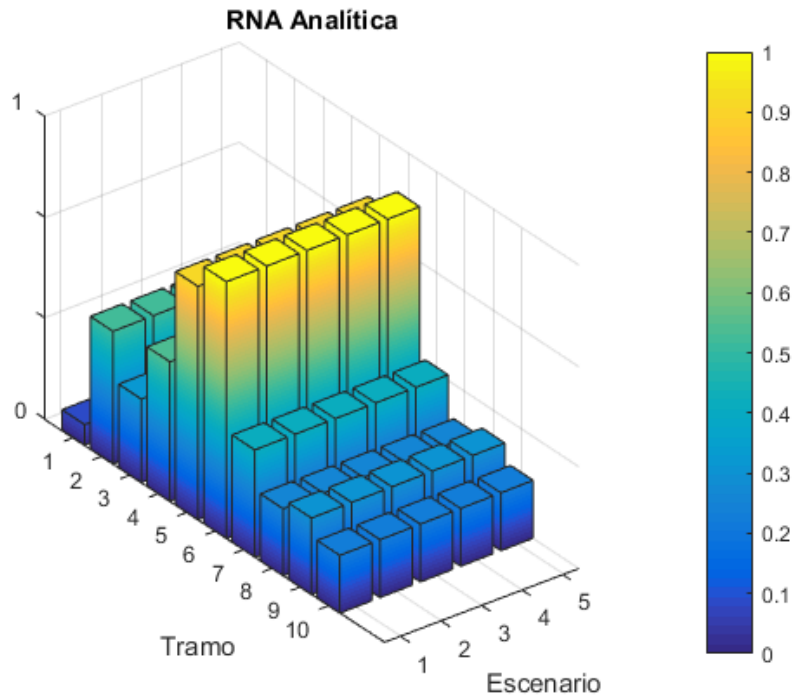


Figura 23. Daños analíticos encontrados por la red tipo Perceptrón Multicapa, usando como entrada el vector de estados x.

Modos y frecuencias

Entrenando la red solo con en el valor de las cinco frecuencias identificadas (matriz de entrada de 5 x 213) y sus respectivas variaciones según los escenarios de daño (usando dos cifras significativas, dado que esto es lo que se puede obtener en una identificación experimental), se utilizó la siguiente topología para Perceptrón:

- Número de neuronas en la capa oculta: 8
- Conjunto de entrenamiento: 65%.
- Conjunto de validación: 20%.
- Conjunto de testeo: 15%.

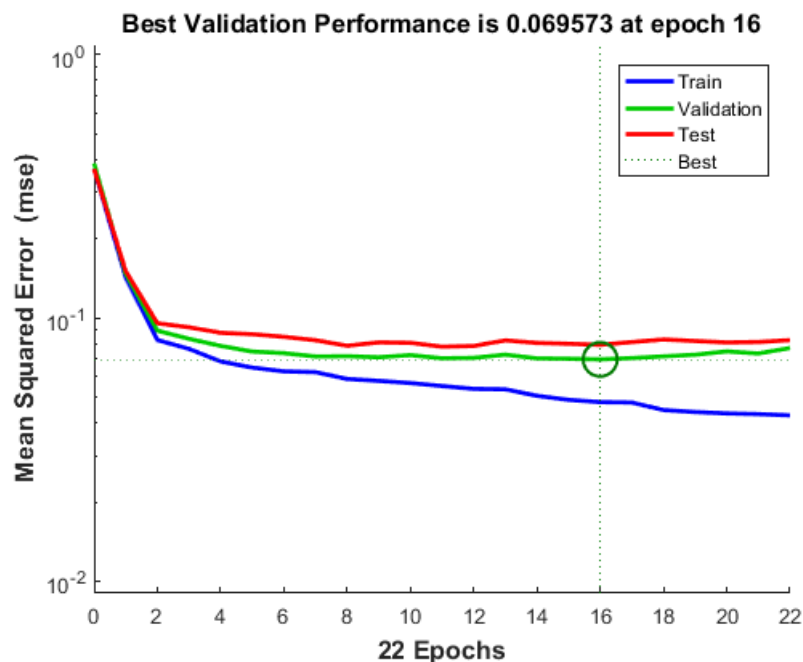


Figura 24. Evolución del error usando las frecuencias.

En general, la red tiene un error mínimo de validación de 6.9%, que es bajo considerando el tipo de datos (baja complejidad). Según las matrices de confusión, la red acertó en un 53% con los mismos datos de entrenamiento, un 60% con los datos de validación analíticos (identificando los escenarios de daño de la Figura 25), pero solo un 40% con los datos de validación experimentales (escenarios de la Figura 26), donde además se predicen más daños que los reales. Esto muestra que este tipo de redes tienen un alto potencial para predecir daño, pues con un dato tan simple como el cambio en las frecuencias (donde solo varían algunas cifras decimales, Tabla 6) alcanza a identificar el daño, especialmente en el caso analítico, como es de esperarse, pues experimentalmente los cambios son menores, por lo que tiende a identificar más daños.

Tabla 6. Diferencias en frecuencia entre el estado sano y los cinco escenarios de daño (analíticos), valores en Hz.

Modo	1	2	3	4	5
caso 1	0,04	0,00	0,04	0,01	0,11
caso 2	0,07	0,01	0,10	0,01	0,20
caso 3	0,08	0,01	0,11	0,01	0,22
caso 4	0,14	0,01	0,20	0,02	0,39
caso 5	0,19	0,00	0,20	0,03	0,56

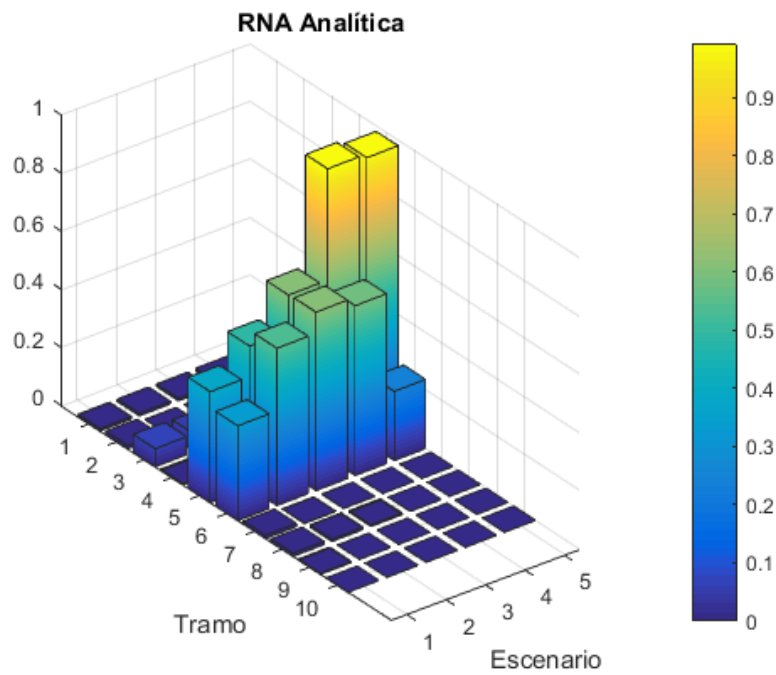


Figura 25. Daños analíticos encontrados por la red tipo Perceptrón Multicapa, usando como entrada las frecuencias.

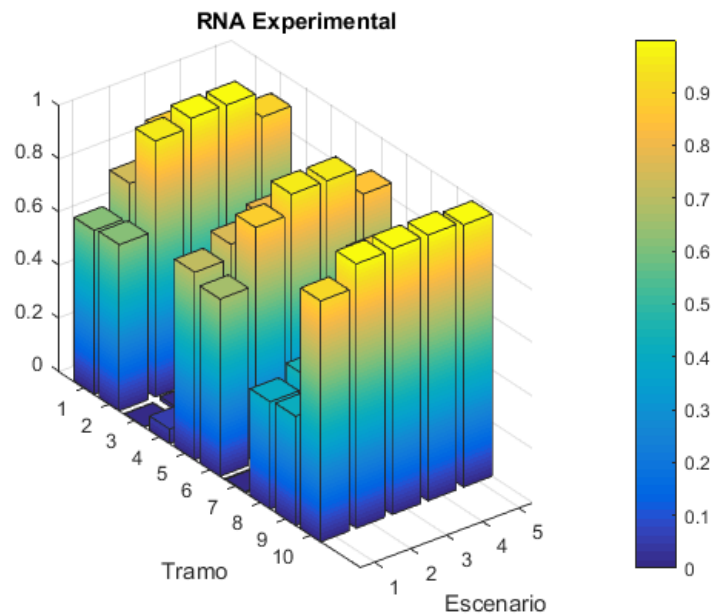


Figura 26. Daños experimentales encontrados por la red tipo Perceptrón Multicapa, usando como entrada las frecuencias.

En las redes de Base Radial se seleccionó:

- *Goal*: el “MSE” que se obtuvo fue de 0.0011.
- *SPREAD*: el parámetro de propagación, con el que se obtuvo un menor error, fue 1.
- Número de neuronas: Se necesitaron 200 neuronas para alcanzar el MSE obtenido.

Se obtuvo un acierto del 68.5% para la red en la matriz de confusión, respecto a los datos con los que se entrenó, pero un 40% con los datos analíticos para validar, identificando los escenarios de la Figura 27, y 40% con los experimentales (Figura 28), lo que evidencia de nuevo que las redes tipo Perceptrón son más eficientes que las Redes de Base Radial para identificar casos de daño diferentes a los usados como entrada a la red.

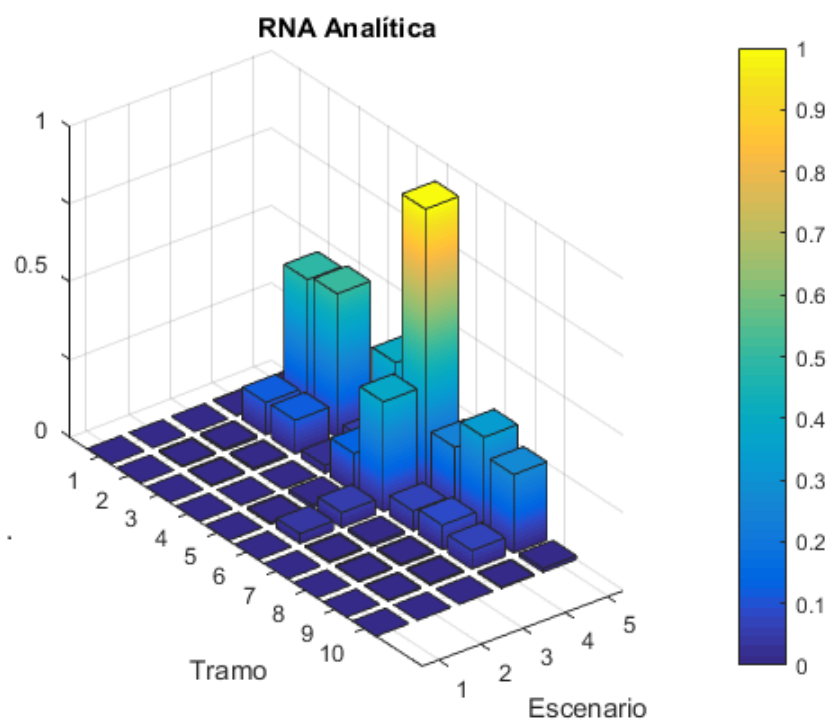


Figura 27. Daños analíticos encontrados por la red de Base Radial, usando como entrada las frecuencias.

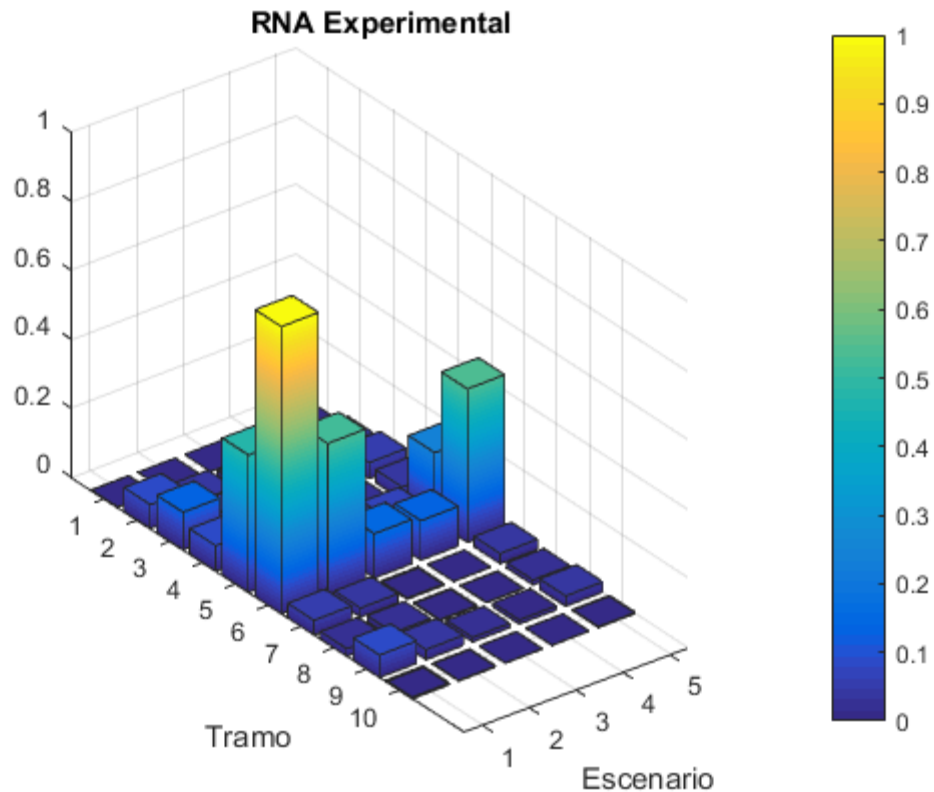


Figura 28. Daños experimentales encontrados por la red la Red de Base Radial, usando como entrada las frecuencias.

En la búsqueda de obtener mejores resultados, se entrenó la red no solo con el valor de las frecuencias, sino también con las cinco formas modales a través de la ecuación 1 – MAC, es decir, cinco valores de frecuencia y cinco correspondientes a las formas modales para cada escenario (matriz de 10 x 213). Sin embargo, aunque la red acertó el 60% con los datos de entrenamiento, solo acertó el 20% con los datos de validación analíticos, por lo que se descartó como una buena entrada, pues la red se vuelve más robusta y comienza a particularizar.

Amplitudes de la función de transmisibilidad

Caso 1

Al usar el vector compuesto de 97 amplitudes, obtenidas de usar las posiciones que más difieren con el estado sano, en la matriz compuesta de la suma de los valores de amplitud para las cuatro frecuencias, por cada escenario de daño (213), como entrada a la red, se seleccionaron como parámetros finales:

- Número de neuronas en la capa oculta: 8
- Conjunto de entrenamiento: 65%.
- Conjunto de validación: 20%.
- Conjunto de testeo: 15%.

Resultando una red con un buen entrenamiento, pues el error mínimo de validación fue 3.03% en la repetición número 118, y se pudo observar que las curvas de entrenamiento se mantienen juntas, es decir, tiene una buena capacidad de generalización (Figura 29). Lo anterior se evidencia en los resultados de las matrices de confusión, donde se logró un acierto del 95% al predecir daño en la base de datos de entrenamiento, un 80% con datos externos a la red analíticos, identificando la mayoría de escenarios reales (Figura 30), y un 60% con los datos de validación experimentales (Figura 31). Así se muestra que la red es efectiva para identificar daños externos al entrenamiento, especialmente si son analíticos, lo que es de esperarse, ya que solo se entrenó con datos del modelo.

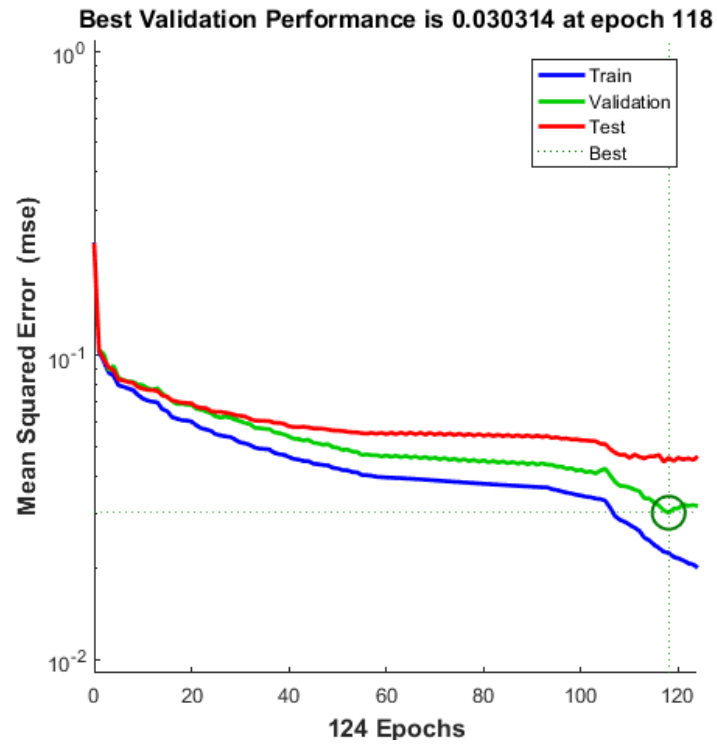


Figura 29. Evolución del error usando las amplitudes de la función de transmisibilidad.

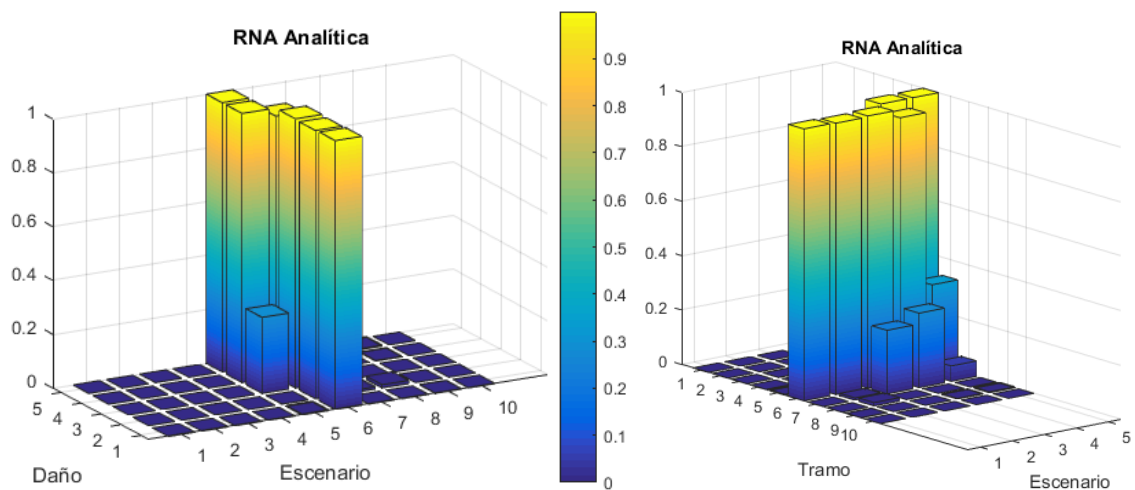


Figura 30. Daños analíticos encontrados por la red tipo Perceptrón Multicapa, usando como entrada amplitudes de la función de transmisibilidad.

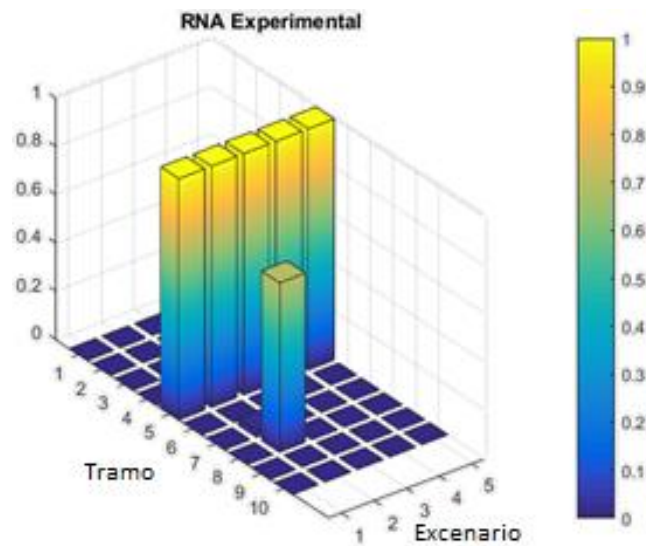


Figura 31. Daños experimentales encontrados por la red tipo Perceptrón Multicapa, usando como entrada amplitudes de la función de transmisibilidad.

Entrenando las redes de Base Radial con esta entrada se consiguió un mejor resultado con los siguientes parámetros:

- *Goal*: se alcanzó un “MSE” de 0.0027.
- *SPREAD*: el parámetro de propagación que generó el menor error fue 20.
- Número de neuronas: Se demandaron 150 neuronas para obtener el menor MSE.

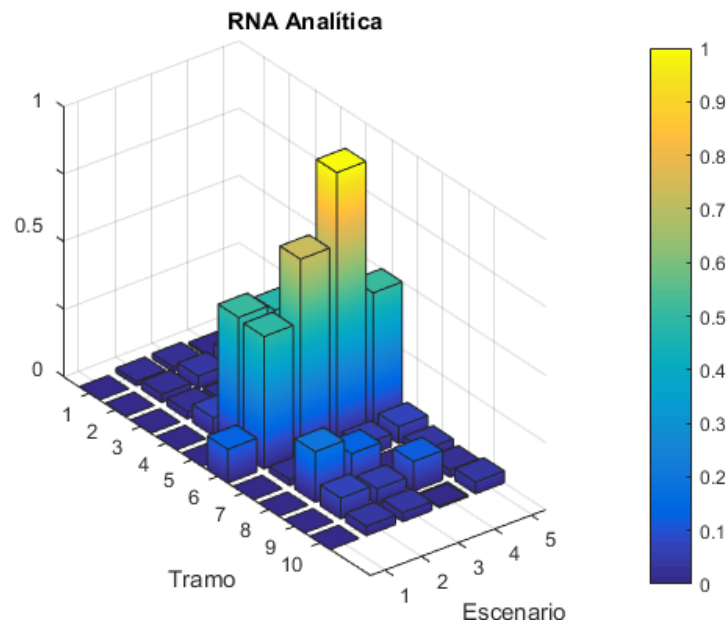


Figura 32. Daños analíticos encontrados por la red de Base Radial, usando como entrada amplitudes de la función de transmisibilidad.

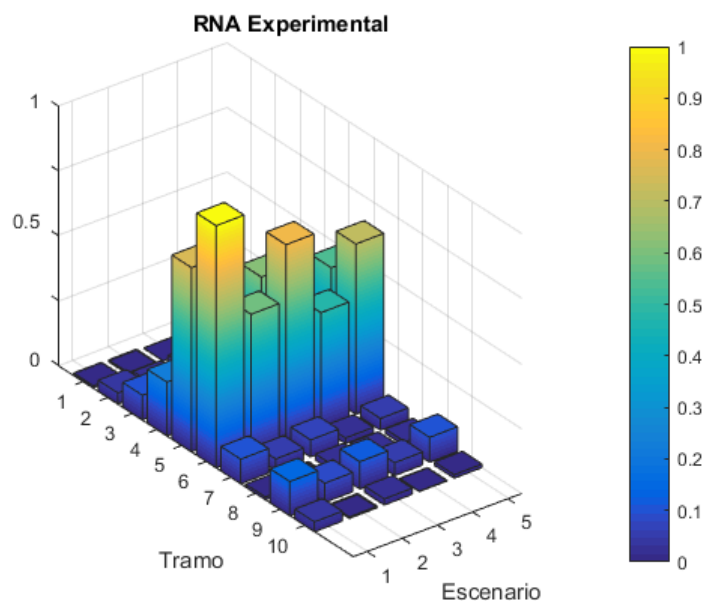


Figura 33. Daños experimentales encontrados por la red de Base Radial, usando como entrada amplitudes de la función de transmisibilidad.

Aunque el error generado en la matriz de confusión con los mismos datos de entrenamiento es solo de 0.5%, con los datos de validación analíticos se obtuvo un acierto del 40% (identificando los escenarios de la Figura 32) y con los de validación experimental 50% (Figura 33), afirmando nuevamente que las redes de tipo Perceptrón son más eficientes para identificar daño en el puente.

Caso 2

Usando la matriz compuesta de 213 escenarios de daño por 114 amplitudes, generadas de sumar o restar (según sea el signo) los valores de amplitud para las cuatro frecuencias, donde más se difiere con el estado sano de la estructura, se determinaron como parámetros finales:

- Número de neuronas en la capa oculta: 8
- Conjunto de entrenamiento: 65%.
- Conjunto de validación: 20%.
- Conjunto de testeo: 15%.

En este caso se alcanzó un acierto del 90% al hacer la matriz de confusión con los datos del entrenamiento; igualmente un acierto de 80% con los datos analíticos, sin embargo, como se observa en la Figura 34, no se evidencia un daño en el escenario 3 y se percibe muy poco en el escenario 5; además se tuvo solo un 30% de acierto al validar con los datos experimentales, por lo que se descartó esta entrada.

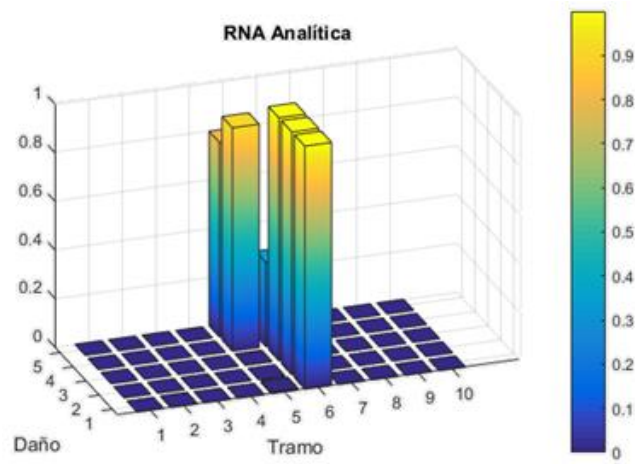


Figura 34. Daños analíticos encontrados por la red tipo Perceptrón Multicapa, usando como entrada amplitudes de la función de transmisibilidad sin el valor absoluto

9. DISCUSIÓN

En todos los casos analizados se alcanzaron mejores resultados con las redes de tipo Perceptrón Multicapa, entrenadas con Levenberg-Marquardt. Igualmente, a partir de los resultados obtenidos, se observó que los valores de amplitud donde más se difiere con el caso sano para cada una de las frecuencias fundamentales (obtenidos de la función de transmisibilidad con solo salidas), contienen información relevante para detectar daño, tanto analítica como experimentalmente. Por lo anterior, al entrenar las redes neuronales artificiales con estos valores como entrada se tuvieron muy buenos resultados, especialmente con los datos analíticos (Tabla 7). Además, experimentalmente fue la entrada que tuvo el mayor acierto y, aunque fue de 60%, se considera un buen acierto pues las redes no fueron entrenadas con datos experimentales, ya que, como se mencionó, se partió de solo cinco escenarios experimentales (donde el máximo de tramos con daño era dos), que fueron usados para validar la red.

Tabla 7. Resumen de los resultados obtenidos al entrenar redes de tipo Perceptrón con los datos analíticos.

Tipo de entrada	Falsos negativos	Falsos positivos	Aciertos
Ceros de la función de transferencia	1	0	80%
Vector de espacio de estados	3	28	40%
Frecuencias	2	1	60%
Valores de amplitud de la función de transmisibilidad	1	1	80%

Igualmente se observó que, aunque entrenar las redes de tipo Perceptrón con un número mayor de neuronas en la capa oculta disminuye el error de validación (MSE), hace que la red tome más tiempo para entrenarse, al volverse más compleja, y llega a un punto en el que la red pierde la capacidad de generalización, al igual que aumentar el número de

patrones de entrenamiento también afecta negativamente los tiempos de entrenamiento. Sin embargo, luego de tener la red entrenada, esta puede funcionar rápidamente.

Las redes que se entrenaron tienen un comportamiento muy similar en la evolución del error de validación (MSE), evidenciando que los resultados para un problema simple se pueden aplicar a otros más complejos.

10. CONCLUSIONES

- A partir de una revisión bibliográfica, se determinó que las redes neuronales artificiales de tipo Perceptrón Multicapa eran las más usadas y apropiadas para identificar daño estructural, por lo que se codificaron algoritmos para esta red con diferentes tipos de entrenamiento, e igualmente se desarrollaron e implementaron las redes de Base Radial, siendo las más similares al Perceptrón; Donde la entrada que mejor se ajusta al problema abordado en este trabajo, son los valores de amplitud de la función de transmisibilidad que más difieren con el estado sano de la estructura.
- Las redes neuronales de tipo Perceptrón Multicapa son eficaces de detectar daños analíticos, aunque tengan presencia de algunos falsos positivos y negativos, obteniendo un acierto máximo del 80%. Adicionalmente, se esperaría encontrar mejores resultados si se contara con señales con una mayor relación señal-ruido, como las obtenidas a partir una excitación artificial (caso de los ceros de la función de transferencia).
- Las redes de Base Radial analizadas en este trabajo son de rápido entrenamiento, al ser menos complejas, pero son menos precisas para detectar daño estructural en el problema abordado.
- La red que mejor se ajusta al problema abordado es de tipo Perceptrón Multicapa, entrenada con Levenberg–Marquardt y, con las siguientes características:
 - Número de neuronas en la capa oculta: 8
 - Función de activación en la capa oculta: Sigmoidea tangente
 - Función de activación en la capa de salida: Sigmoidea tangente
 - Conjunto de entrenamiento: 65%.
 - Conjunto de validación: 20%.
 - Conjunto de testeo: 15%.

- Patrón de entrada: valores de amplitud de la función de transmisibilidad donde la diferencia es mayor con el estado sano.
- Experimentalmente no se obtuvo un alto nivel de precisión (60% de acierto) para detectar daño estructural en el puente, lo que puede ser por la falta de datos experimentales para el entrenamiento.

11. REFERENCIAS

- Dirección general de concesiones (2011). Ministro Golborne inspecciona puente dañado en Ruta 68. Coordinación de concesiones de obras públicas. Noticias de la coordinación. Obtenido el 20 de septiembre 2017 de: <http://www.concesiones.cl/noticias/Paginas/DetalledeNoticias.aspx?item=73>
- El Mercurio (2011). Ministro Golborne y fisura en puente de la Ruta 68: "Se evitó un desastre mayor". Emol. Obtenido el 20 de septiembre 2017 de: <http://www.emol.com/noticias/nacional/2011/08/30/500590/ministro-golborne-y-fisura-en-puente-de-la-ruta-68-se-evito-un-desastre-mayor.html>
- ACIS (2006). Manual sobre Sismo Resistencia para Funcionarios de Planeación Municipal [PDF]. Obtenido el 25 de septiembre 2017 de: http://www.duitama-boyaca.gov.co/apc-aa-files/32663833373834396563383434623336/manual_sismoresistencia.pdf
- Alfonso, W. (2017). Redes Competitivas y Mapas Autoorganizados [PDF]. Trabajo presentado en clase de Redes Neuronales, abril, Colombia.
- Alfonso, W. (2017). Diseño y simulación de un Perceptrón Multicapa (MLP) con MATLAB [PDF]. Trabajo presentado en clase de Redes Neuronales, marzo, Colombia.
- Amézquita, J., Domínguez, A., Osornio, R., (2012). Monitoreo de vibraciones en tiempo real para detectar daños en estructuras.
- Bellman, R. (1978). An introduction to artificial intelligence: can computer think? (No. 04; Q335, B4.).
- Bertona, L. (2005). Entrenamiento de redes neuronales basado en algoritmos evolutivos. Tesis de pregrado. Universidad de Buenos Aires, Buenos Aires. Obtenido el 15 de noviembre 2017 de: <http://laboratorios.fi.uba.ar/lsi/bertona-tesisingenieriainformatica.pdf>
- Brownjohn, J. M. W., Dynamic Assessment of Tall Buildings, Building Health Monitoring, 2001, p. 82-92.

- Caicedo, E. & López, J. (2009). Una aproximación práctica a las Redes Neuronales Artificiales. Colombia: Programa Editorial Universidad del Valle.
- Caicedo, E. (2017). Arquitecturas y Aprendizaje Redes Neuronales Artificiales [diapositivas de PowerPoint]. Trabajo presentado en clase de Redes Neuronales, marzo, Colombia.
- Caicedo, E. (2017). Introducción Redes Neuronales Artificiales [diapositivas de PowerPoint]. Trabajo presentado en clase de Redes Neuronales, febrero, Colombia.
- Carroll, S.P., Owen, J.S., & Hussein, M.F. (2013). A Coupled Biomechanical/ Discrete Element Crowd Model of Crowd-Bridge Dynamic Interaction and Application to the Clifton Suspension Bridge.
- Crémona, C., et al. (2016). "Real- time damage detection based on pattern recognition." *Structural Concrete* 17(3): 338-354.
- Demuth, H., Beale, M. & Hagan, M., 1997. MATLAB: Neural Network Toolbox: User's Guide. *Math Works*.
- Domínguez, S., Campoy, P., Sebastián, J. & Jiménez, A. (2006). Control en el espacio de estado [PDF]. Madrid: Pearson Educación S.A. Obtenido el 15 de diciembre 2017 de: <ftp://www.ece.buap.mx/pub/profesor/academ91/Automatizacion2/Libros/Control%20en%20el%20Espacio%20de%20Estado.pdf>
- El Espectador. (12 de octubre de 2013). Obtenido de Se desploma el edificio Space en Medellín. Obtenido el 10 de agosto 2017: <http://www.elespectador.com/noticias/nacional/se-desploma-el-edificio-space-medellin-articulo-452128>
- Gawronski, W. K. (2004). *Advanced Structural Dynamics and Active Control of Structures*. New York: Springer.
- Giraldo, D.F., Song, W., Dyke, S.J. & Caicedo, J.M. (2009). Modal identification through ambient vibration: comparative study. *Journal of engineering mechanics* 135 (8), 759-770.

- Grupo de Investigación en Ingenierías Sísmica, Eólica, Geotécnica y Estructural (G-7) & Laboratorio de Cartografía y Sistemas de Información Geográfica - CARSIG (2015). Desarrollo e implementación de un sistema integrado de gestión de puentes para Santiago de Cali – Etapa I: inventario, diagnóstico e implementación en SIG. Informe final. Universidad del valle, Santiago de Cali.
- He, X., Moaveni, B., Conte, J. P., Elgamal, A. y Masri, S., System identification of Vincent Thomas Bridge using simulated wind response data, Proceedings of the Second International Conference on Bridge Maintenance, Safety and Management (IABMAS'04), Octubre, 2004.
- Hernández G., Miguel. (2001), Detección de daño a partir de cambios en las características vibracionales usando redes neuronales. Obtenido el 20 de agosto 2017 de: <https://ojsrevistaing.uniandes.edu.co/ojs/index.php/revista/article/viewFile/551/732>
- Hilera, R., & Martínez, V. (1995). Redes Neuronales Artificiales. Fundamentos, modelos y aplicaciones. España: RA-MA Editorial.
- I-35W Mississippi River Bridge. Wikipedia. Obtenido el 15 de septiembre 2017 de: https://en.wikipedia.org/wiki/I-35W_Mississippi_River_bridge
- Isasi, V. & Galván, L. (2004). Redes Neuronales Artificiales: Un enfoque práctico. España: PEARSON Prentice Hall.
- Kanwar, V., Kwatra, N. & Aggarwal, P (2007). Damage Detection for Framed RCC Buildings using ANN Modeling. International Journal of Damage Mechanics, 16, 457-472.
- Mahu, J. (2012). Detección de daños en una viga simple mediante antiresonancia y redes neuronales. Tesis de pregrado. Universidad de Chile, Santiago de Chile. Obtenido el 21 de agosto 2017 de: http://repositorio.uchile.cl/bitstream/handle/2250/111322/cf-mahu_js.pdf?sequence=1
- Marulanda J., Caicedo J.M. and Thomson P., “Modal Identification under Sinusoidal Excitation using Mobile Sensors”, Ingeniería y Competitividad, Universidad del Valle, Vol. 19, No. 1, pp. 159-167, 2017.

- Matich, D. J. (2001). Redes Neuronales: Conceptos básicos y aplicaciones [PDF]. Obtenido el 30 de septiembre 2017 de: https://www.fro.utn.edu.ar/repositorio/catedras/quimica/5_anio/orientadora1/monograias/matich-redesneuronales.pdf
- Meruane, V. & Mahu, J. (2014). Real-time structural damage assessment using artificial neural networks and anti-resonant frequencies. ResearchGate. Obtenido el 25 de septiembre 2017 de: <https://www.researchgate.net/publication/261833852>
- Muñoz Diaz, E. E. (2002). Estudio de las causas del colapso de algunos puentes en Colombia. Ingeniería y Universidad, 6(1), 33-48.
- Muñoz, E., Núñez, F., Rodríguez, W., Otalora, C., & Ruiz, D. (2006). Confiabilidad estructural apoyada en monitoreo e instrumentación: aplicación en un puente en acero.
- Panesso, A. (2017). Diagnóstico de salud estructural de un puente peatonal mediante el análisis de su comportamiento dinámico. Tesis de pregrado en Ingeniería Civil. Universidad del Valle, Santiago de Cali.
- Pérez, M., Quintana, J., Carrión, F., Hernández, A., Crespo, S., & López, J. (2012). Análisis de sensibilidad para la ubicación de sensores en estructuras [PDF]. Secretaría de comunicaciones y transportes. Obtenido el 16 de diciembre 2017 de: <http://imt.mx/archivos/Publicaciones/PublicacionTecnica/pt349.pdf>
- Rodríguez, D. & Bordóns, C. (2005). Apuntes de ingeniería de control. Análisis y control de sistemas en espacio de estado. Obtenido el 20 de febrero 2018 de: www.esi2.us.es/~danirr/apuntesIC4.pdf
- Sánchez, J. & Páez, O. Daños en clínicas Santillana y Red David tras temblor en Cali. El País. Obtenido el 20 de agosto 2017 de: <http://www.elpais.com.co/elpais/califotos/danos-estructurales-clinica-santillana-tras-temblor-cali>
- Santos, J., Crémona, C., Silveira, P. & Calado, L. (2016). Real-time damage detection based on pattern recognition. Structural Concrete, 17(3), 338-354.
- Kourehli, S. S. (2015). 1595. Damage quantification method using artificial neural network and static response with limited sensors. Journal of Vibroengineering, 17(3).

- Torres, Héctor. (2015). Automatización de las técnicas SSI-Data y NExT-ERA para identificación modal con vibración ambiental. Tesis de pregrado en Ingeniería Civil. Universidad del Valle, Santiago de Cali.
- Villalba, J., & Laier, J. (2017). DETECCIÓN DE DAÑO ESTRUCTURAL UTILIZANDO REDES NEURONALES ARTIFICIALES: Un estado del Arte. Cadernos de Engenharia de Estruturas, 17(67), 1-19.
- Wong F., Thint M. y Tung A. (1996), On-line detection of structural damage using neural networks, civil engineering systems vol. i4, p.p. 167-197.
- Zaballa, I. Lectura 4: El Modelo de Espacio-Estado [PDF]. Control. Obtenido el 22 de febrero 2017 de: <http://www.ehu.eus/izaballa/Control/Apuntes/lec4.pdf>