

**DETECCIÓN DE DAÑOS EN ESTRUCTURAS UTILIZANDO LA TÉCNICA DE
OPTIMIZACIÓN PSO**

DIEGO ANDRÉS RAMÍREZ RAMÍREZ

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICOMECÁNICAS
ESCUELA DE INGENIERÍA MECÁNICA
BUCARAMANGA
2010**

**DETECCIÓN DE DAÑOS EN ESTRUCTURAS UTILIZANDO LA TÉCNICA DE
OPTIMIZACIÓN PSO**

DIEGO ANDRÉS RAMÍREZ RAMÍREZ

Trabajo de grado presentado para optar al título de Ingeniero Mecánico

Directores

**JABID QUIROGA
Magister, Ingeniero Mecánico**

**IVÁN GOMEZ
Ingeniero Civil**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICOMECÁNICAS
ESCUELA DE INGENIERÍA MECÁNICA
BUCARAMANGA
2010**

A mis papás.

A mi novia.

A mi hijo.

A mis hermanos.

A mis amigos.

AGRADECIMIENTOS

A Jabid Quiroga, MSc., director del proyecto por aportar su visión en el desarrollo de éste, por la confianza depositada en el proyecto y su incondicional apoyo.

Al profesor Iván Gómez, director por dar un rumbo al proyecto, por la colaboración brindada y su buena disposición en todo momento.

A la Universidad Industrial de Santander, centro cultural de la región y espacio de formación personal y profesional, y a los docentes que comprometidos con la academia promueven la inventiva, creatividad y el amor por la carrera.

A mi familia por brindarme su incondicional apoyo.

CONTENIDO

	Pág.
INTRODUCCIÓN	18
1. DESCRIPCIÓN DEL PROYECTO	23
1.1. ANTECEDENTES	25
1.2. ALCANCE DEL PROYECTO	26
1.2.1 Objetivos General	26
1.2.2 Objetivos Específicos	27
2. REVISIÓN BIBLIOGRÁFICA	28
2.1. ANÁLISIS DINÁMICO	28
2.1.1. FORMULACIÓN CLÁSICA: Un solo grado de libertad (SDOF)	29
2.1.2. FORMULACIÓN CLÁSICA: Sistemas de múltiples grados de libertad (MDOF)	30
2.2 METODOLOGIA EMPLEADA PARA LA LOCALIZACION DE DAÑO	33
2.2.1. La Detección de Daños como un Proceso de Optimización	34
3. OPTIMIZACIÓN POR ENJAMBRE DE PARTÍCULAS	37
3.1. Descripción General de Particle Swarm Optimization y Algoritmos de Optimización Inteligentes	37
3.2. Particle Swarm Optimization	39
3.3. Herramienta PSOt para MATLAB	43
4. PROCEDIMIENTO GENERAL	48
4.1. METODOLOGIA	48
4.2. ESTRUCTURAS ANALIZADAS:	51

4.3. ESCENARIOS DE DAÑO:	54
4.4. IMPLEMENTACION DE LA METODOLOGIA	57
4.4.1. Determinación de las propiedades dinámicas de las estructuras analizadas	57
4.4.2. Ejecución de la metodología empleada para la toma de resultados	65
5. ANÁLISIS DE RESULTADOS	67
5.1. IDENTIFICACIÓN DE ELEMENTOS DAÑADOS	67
5.1.1. Función Objetivo "CALIBRAV1" (ec. 2.14)	67
5.1.2. Función Objetivo CALIBRAV (ec. 2.12)	80
5.2. ANÁLISIS DE LA CONVERGENCIA DE PSO	84
5.2.1. Función Objetivo CalibraV1 (ec. 2.14):	85
5.2.2. Función Objetivo CalibraV (ec. 2.12):	87
5.3. TIEMPO DE EJECUCIÓN DE PSO	88
5.3.1. Función Objetivo CalibraV1 (ec. 2.14):	89
5.3.2. Función Objetivo CalibraV (ec. 2.12):	91
6. CONCLUSIONES Y SUGERENCIAS PARA TRABAJOS FUTUROS	93
6.1. METODOLOGIA DE LOCALIZACIÓN DE ELEMENTOS CON DAÑO	93
6.2. OPTIMIZACION POR ENJAMBRE DE PARTICULAS PSO	95
6.3. SUGERENCIAS PARA TRABAJOS FUTUROS	97
BIBLIOGRAFÍA	98
ANEXOS	1001

LISTA DE TABLAS

Tabla 1. Conectividad de los elementos en la estructura tipo Viga.	53
Tabla 2. Conectividad de los elementos en la estructura tipo Viga.	54
Tabla 3. Configuraciones de ruido que serán estudiadas (a) Viga y (b) Armadura.	55
Tabla 4. Escenarios de daño estudiados para: (a) Viga y (b) Armadura.	56
Tabla 5. Resultados de frecuencias y periodos OPENSEES.	59
Tabla 6. Errores mínimos alcanzados en la función objetivo CalibraV1 ante diversos niveles de ruido: Viga, daño múltiple 2 elementos.	85
Tabla 7. Errores mínimos alcanzados en la función objetivo CalibraV1 ante diversos niveles de ruido: Viga, daño múltiple 4 elementos.	86
Tabla 8. Errores mínimos alcanzados en la función objetivo CalibraV1 ante diversos niveles de ruido: Armadura, daño múltiple simple.	86
Tabla 9. Errores mínimos alcanzados en la función objetivo CalibraV1 ante diversos niveles de ruido: Armadura, daño múltiple 2 elementos.	87
Tabla 10. Errores mínimos alcanzados en la función objetivo CalibraV ante ausencia de ruido: Viga, daño múltiple 2 elementos.	87
Tabla 11. Errores mínimos alcanzados en la función objetivo CalibraV ante diversos niveles de ruido: Armadura, daño múltiple 2 elementos.	87
Tabla 12. Errores mínimos alcanzados en la función objetivo CalibraV1 ante diversos niveles de ruido: Viga, daño simple.	89
Tabla 13. Errores mínimos alcanzados en la función objetivo CalibraV1 ante diversos niveles de ruido: Viga, daño múltiple 2 elementos.	89

Tabla 14. Errores mínimos alcanzados en la función objetivo CalibraV1 ante diversos niveles de ruido: Viga, daño múltiple 4 elementos.	90
Tabla 15. Errores mínimos alcanzados en la función objetivo CalibraV1 ante diversos niveles de ruido: Armadura, daño simple.	90
Tabla 16. Errores mínimos alcanzados en la función objetivo CalibraV1 ante diversos niveles de ruido: Armadura, daño múltiple 2 elementos.	91
Tabla 17. Errores mínimos alcanzados en la función objetivo CalibraV ante ausencia de ruido: Viga, daño múltiple 2 elementos.	91
Tabla 18. Errores mínimos alcanzados en la función objetivo CalibraV ante diversos niveles de ruido: Armadura, daño múltiple 2 elementos.	92

LISTA DE FIGURAS

	Pág.
Figura 1. Modos de vibración de una estructura de 2 grados de libertad (CHOPRA, 2001)	28
Figura 2. Algoritmos secuenciales vs. Algoritmos inteligentes	39
Figura 3. Diagrama de flujo del algoritmo PSO.	42
Figura 4. Diagrama de flujo del sistema.	50
Figura 5. Viga simplemente apoyada.	52
Figura 6. Armadura de 13 elementos simplemente apoyada.	52
Figura 7. Modelo viga usando SAP2000	59
Figura 8. Valores de periodos y frecuencias hallado por SAP2000 para la viga.	60
Figura 9. Modos de Vibracion Viga (a) 1 modo, (b) 2 modos, (c) 3 modos, (d) 4 modos, (e) 5 modos, (f) 6 modos, (g) 7 modos, (h) 8 modos, (i) 9 modos, (j) 10 modos	61
Figura 10. Modelo armadura usando SAP2000	62
Figura 11. Modos de Vibracion Armadura (a) 1 modo, (b) 2 modos, (c) 3 modos, (d) 4 modos, (e) 5 modos, (f) 6 modos, (g) 7 modos, (h) 8 modos	63
Figura 12. Interface grafica herramienta PSO de Matlab	66
Figura 13. Comparación del Daño Real y el Daño Calculado en la viga con diferentes cantidades de ruido: Función objetivo CalibraV1, daño simple, 4 modos de vibración.	68

Figura 14. Comparación del Daño Real y el Daño Calculado en la viga con diferentes cantidades de ruido: Función objetivo CalibraV1, daño simple, 6 modos de vibración.	68
Figura 15. Comparación del Daño Real y el Daño Calculado en la viga con diferentes cantidades de ruido: Función objetivo CalibraV1, daño simple, 8 modos de vibración.	69
Figura 16. Comparación del Daño Real y el Daño Calculado en la viga con diferentes cantidades de ruido: Función objetivo CalibraV1, daño simple, 10 modos de vibración.	69
Figura 17. Comparación del Daño Real y el Daño Calculado en la viga con diferentes cantidades de ruido: Función objetivo CalibraV1, daño múltiple 2 elementos, 4 modos de vibración.	70
Figura 18. Comparación del Daño Real y el Daño Calculado en la viga con diferentes cantidades de ruido: Función objetivo CalibraV1, daño múltiple 2 elementos, 6 modos de vibración.	71
Figura 19. Comparación del Daño Real y el Daño Calculado en la viga con diferentes cantidades de ruido: Función objetivo CalibraV1, daño múltiple 2 elementos, 8 modos de vibración.	71
Figura 20. Comparación del Daño Real y el Daño Calculado en la viga con diferentes cantidades de ruido: Función objetivo CalibraV1, daño múltiple 2 elementos, 10 modos de vibración.	72
Figura 21. Comparación del Daño Real y el Daño Calculado en la viga con diferentes cantidades de ruido: Función objetivo CalibraV1, daño múltiple 4 elementos, 4 modos de vibración.	73
Figura 22. Comparación del Daño Real y el Daño Calculado en la viga con diferentes cantidades de ruido: Función objetivo CalibraV1, daño múltiple 4 elementos, 8 modos de vibración.	74

Figura 23. Comparación del Daño Real y el Daño Calculado en la viga con diferentes cantidades de ruido: Función objetivo CalibraV1, daño múltiple 4 elementos, 10 modos de vibración.	75
Figura 24. Comparación del Daño Real y el Daño Calculado en la armadura con diferentes cantidades de ruido: Función objetivo CalibraV1, daño simple, 4 modos de vibración.	75
Figura 25. Comparación del Daño Real y el Daño Calculado en la armadura con diferentes cantidades de ruido: Función objetivo CalibraV1, daño simple, 5 modos de vibración.	76
Figura 26. Comparación del Daño Real y el Daño Calculado en la armadura con diferentes cantidades de ruido: Función objetivo CalibraV1, daño simple, 6 modos de vibración.	76
Figura 27. Comparación del Daño Real y el Daño Calculado en la armadura con diferentes cantidades de ruido: Función objetivo CalibraV1, daño simple, 7 modos de vibración.	77
Figura 28. Comparación del Daño Real y el Daño Calculado en la armadura con diferentes cantidades de ruido: Función objetivo CalibraV1, daño simple, 8 modos de vibración.	77
Figura 29. Comparación del Daño Real y el Daño Calculado en la armadura con diferentes cantidades de ruido: Función objetivo CalibraV1, daño múltiple 2 elementos, 4 modos de vibración.	78
Figura 30. Comparación del Daño Real y el Daño Calculado en la armadura con diferentes cantidades de ruido: Función objetivo CalibraV1, daño múltiple 2 elementos, 5 modos de vibración.	78
Figura 31. Comparación del Daño Real y el Daño Calculado en la armadura con diferentes cantidades de ruido: Función objetivo CalibraV1, daño múltiple 2 elementos, 6 modos de vibración.	79

Figura 32. Comparación del Daño Real y el Daño Calculado en la armadura con diferentes cantidades de ruido: Función objetivo CalibraV1, daño múltiple 2 elementos, 7 modos de vibración.	79
Figura 33. Comparación del Daño Real y el Daño Calculado en la armadura con diferentes cantidades de ruido: Función objetivo CalibraV1, daño múltiple 2 elementos, 8 modos de vibración.	80
Figura 34. Comparación del Daño Real y el Daño Calculado en la viga sin ruido: Función objetivo CalibraV, daño múltiple 2 elementos, 6 modos de vibración	80
Figura 35. Comparación del Daño Real y el Daño Calculado en la viga sin ruido: Función objetivo CalibraV, daño múltiple 2 elementos, 8 modos de vibración	81
Figura 36. Comparación del Daño Real y el Daño Calculado en la armadura sin ruido y con nivel de ruido promedio: Función objetivo CalibraV, daño múltiple 2 elementos, 5 modos de vibración	81
Figura 37. Comparación del Daño Real y el Daño Calculado en la armadura sin ruido y con nivel de ruido promedio: Función objetivo CalibraV, daño múltiple 2 elementos, 6 modos de vibración	82

LISTA DE ANEXOS

	Pág.
Anexo A. Complementación teórica	101
Anexo B. Ejemplo de analisis modal	115
Anexo C. Determinación de la respuesta dinámica por medio de opensees de las estructuras estudiadas	118
Anexo D. Archivos matlab	126

RESUMEN

TÍTULO: DETECCIÓN DE DAÑOS EN ESTRUCTURAS UTILIZANDO OPTIMIZACIÓN POR ENJAMBRE DE PARTÍCULAS (PSO)*

AUTOR: DIEGO ANDRÉS RAMÍREZ RAMÍREZ**

PALABRAS CLAVE:

Parámetros Dinámicos, Optimización por Enjambre de Partículas, PSO, Metaheurísticas, Detección de daños, Elementos Finitos.

DESCRIPCIÓN

Este trabajo de grado se desarrolló con la intención de incentivar el desarrollo de la investigación en el campo de la inteligencia artificial y de detección de fallas en estructuras y elementos de máquinas, para que a través de la investigación y el uso de nuevas técnicas de optimización, la ingeniería local pueda estar a la vanguardia en estas áreas y por lo tanto posea herramientas alternativas para poder entender y resolver de una mejor manera problemas existentes y futuros.

La evaluación del estado de las estructuras es un tema de investigación muy importante para diversos campos de ingeniería y, por lo tanto, están siendo desarrolladas metodologías que permitan detectar daño en una estructura. El presente trabajo tiene como objetivo verificar la aplicabilidad de la técnica de búsqueda y optimización: Optimización por Enjambre de Partículas (PSO) en la detección de daño a partir de las variaciones ocurridas, entre las condiciones con y sin daño, de los parámetros dinámicos de la estructura. Los parámetros dinámicos de la estructura, sin y con daño, son determinados a partir del modelo de elementos finitos de la estructura usando la herramienta computacional de libre distribución OPENSEES. Diversas cantidades de modos de vibración, escenarios de daño y ruido fueron considerados buscando simular las características de la información obtenida por medio de un monitoreo estructural real y analizar el desempeño de la técnica empleada bajo todas las condiciones mencionadas. La metodología implementada es aplicada a una estructura de tipo viga discretizada en 10 elementos y armadura de 13 elementos. Los resultados muestran el buen desempeño de la PSO para detectar tanto daño simple como múltiple en una estructura bajo condiciones normales de ruido.

* Proyecto de Grado

** Facultad de Ingenierías Fisicomecánicas, Escuela de Ingeniería Mecánica. Directores: Jabid Quiroga, Iván Gómez.

ABSTRACT

TITLE: STRUCTURAL DAMAGE DETECTION USING PARTICLE SWARM OPTIMIZATION (PSO)

AUTHOR: DIEGO ANDRÉS RAMÍREZ RAMÍREZ**

KEYWORDS: Dynamic Parameters, Particle Swarm Optimization, PSO, Metaheuristics, Damage Detection, Finite Element Methods.

DESCRIPTION

This work of degree developed with the intention of stimulating the development of the investigation in the field of the artificial intelligence and of detection of faults in structures and elements of machines, in order that across the investigation and the use of new technologies of optimization, the local engineering could be to the forefront in these areas and therefore it possesses alternative tools to be able to deal and solve of a better way existing and future problems.

The assessment of structural health is an important research topic in many engineering fields and, for that reason; damage detection methodologies are being developed. The goal of this dissertation is to verify the applicability of the search and optimization method Particle Swarm Optimization (PSO) for detecting damage using dynamic parameters changes between undamaged and damaged conditions of the structure. Undamaged and damaged dynamic parameters are computed using the finite element model of the structure using the free distribution software OPENSEES. The effect of the amount of modes of vibration, damaged elements and noise levels in the measurements of the structural response were considered in order to simulate the characteristics of the information acquired in a real dynamic test and analyze the performance of the developed technique under all the mentioned constitutions. The implemented methodology is applied in two different structures: 10 elements quantized beam and 13 elements truss configuration. The results indicate the good performance of the PSO for detecting structural damage under average noise levels.

* Undergraduate Degree Work

** Physical-Mechanical Engineering Department, School of Mechanical Engineering. Directors: Jabid Quiroga, Iván Gómez.

INTRODUCCIÓN

Diferentes sectores industriales, tales como el mecánico, aeronáutico, civil, entre otros, se encuentran desarrollando métodos de detección de daños no destructivos (DDND) que permitan evaluar los daños en sus estructuras – aviones, plataformas marítimas de petróleo, puentes, torres de transmisión. El daño puede ser el resultado decurrente simplemente del uso, envejecimiento, sobrecargas, fatiga o el acontecimiento de un evento extremo.

La importancia de las metodologías de detección de daños se aprecia en el hecho de que ellas ayudan en la toma de decisiones sobre las necesidades de reforzar o reparar las estructuras. Estas metodologías al ser incluidas en programas de mantenimiento, ayudan a reducir al mínimo las consecuencias del daño, tales como las pérdidas económicas en caso que la estructura deje de funcionar correctamente, como los riesgos que implica una falla catastrófica.

Varios estudios adelantados por fuentes confiables sobre las implicaciones económicas de la detección de daños a través de sistemas de monitoreo de salud estructural (SHM), han presentado que las grandes empresas usuarias de estos sistemas establecen un beneficio económico significativo. Igualmente en empresas con maquinaria expuesta a constantes eventos extremos, tales como las denominadas estructuras *Offshore*, el beneficio se aprecia notablemente cuando ante sismos de grandes intensidades, gracias al SHM se puede determinar la condición actual de la estructura con el fin de no representar una amenaza para sus ocupantes¹.

¹ Extraído de datos e índices Schlumberger y Pemex:
<http://www.slb.com/media/services/resources/oilfieldreview/spanish06/spr06/composite.pdf>
<http://www.invdes.com.mx/antiores/Noviembre2001/htm/pla18ta.html>

Por otra parte, las principales técnicas de tipo DDND consisten en inspección visual, utilizando acústica, ultra sonido, campos magnéticos, rayos-X, e incluso métodos térmicos. Estas técnicas pueden realizar una detección de daño local y consiguen funcionar satisfactoriamente si el daño está localizado dentro de una región de la estructura que sea conocida a priori y que sea accesible (REN e DE ROECK, 2002; FARAVELLI e CASCIATI, 2004; RAHAI *et al.* 2007).

Debido a las limitaciones de estos métodos convencionales, el desarrollo de métodos DDND ha sido orientado hacia la consecución de procedimientos analíticos de mayor confiabilidad y versatilidad, con la capacidad de certificar la funcionalidad y la seguridad de la estructura. Tales técnicas DDND están basadas en su mayoría en los parámetros dinámicos del sistema a analizar.

Algunas de las principales ventajas de utilizar las técnicas DDND analíticas son: no se precisa conocer a priori los lugares damnificados; la posibilidad de determinar el comportamiento global de la estructura; la obtención, a partir de un número limitado de sensores, de la información suficiente para la localización y cuantificación del daño; la utilización, en la mayoría de casos, de un número de equipamiento no excesivo (HUMAR, 2006); y la capacidad de realizarse mediciones remotamente, lo que rápidamente minimiza el impacto sobre el funcionamiento de la estructura (DOEBLING, 1998).

A causa de la deficiencia y/o inexistencia de las actividades de mantenimiento preventivo y predictivo lo cual es comúnmente visto en países subdesarrollados, se observa la carencia de un método apropiadamente establecido para la detección de daños en estructuras y equipos, siendo destinados para este propósito exclusivamente métodos empíricos o procedimientos asociados con el mantenimiento correctivo, lo cual no es conveniente para la salud estructural, representando pérdidas económicas e incluso una amenaza en ciertos casos,

donde un colapso de la estructura pone en riesgo la integridad de sus ocupantes y/u operarios.

Recursos tales como las tecnologías informáticas abren un amplio campo de estudio en el desarrollo y análisis de estrategias de detección de daños, consiguiendo a través de un programa computacional, una herramienta cuyo objetivo sea el de avalar la funcionalidad y la seguridad del equipo o estructura, con la máxima eficiencia posible, junto con la versatilidad necesaria para poder adaptarse a un problema particular extrapolando el esquema básico del programa.

Dentro de las técnicas DDND analíticas, se encuentran técnicas fundamentadas en la variación de los parámetros dinámicos de la estructura entre el estado inicial y el actual. La base física de estas técnicas se encuentra en el hecho que el daño introducido altera las propiedades de una estructura – rigidez, masa y amortiguamiento- ocasionando variaciones en las propiedades dinámicas (respuestas) de la estructura, sean frecuencias naturales, formas modales y/o amortiguamientos modales. Según lo estipulado en el anterior argumento, se puede plantear un problema inverso en el cual, a partir de mediciones de estas las dinámicas, se desea encontrar los parámetros actuales del sistema que originan esa respuesta de la estructura, es decir determinar cuál es el daño al que se le atribuye esta respuesta.

Dado que las mediciones experimentales contienen una determinada cantidad de ruido, la dificultad en la resolución del problema se ve aumentada. Para lidiar con estos tipos de problemas, los datos experimentales deben ser extendidos y/o filtrados, para convertir esta información obtenida en un dominio bien estructurado que pueda ser resuelto utilizando alguna herramienta analítica. Dependiendo de la herramienta utilizada, en algunas ocasiones los datos experimentales pueden ser procesados sin necesidad del filtro requerido.

Por otro lado, el problema de detección de daños puede ser considerado esencialmente como un problema de optimización, por lo cual la aplicabilidad de metaheurísticas [19] se convierte en una alternativa plausible, ya que, dadas las condiciones del problema, estas podrían tener un mejor desempeño cuando son comparadas con algoritmos clásicos de optimización. Entre las principales razones, se encuentre el hecho que las metaheurísticas permiten encontrar soluciones globales, no requieren cálculo de derivadas de la función objetivo, no dependen del punto inicial y poseen poca sensibilidad a los ruidos (BEGAMBRE, 2007).

En la ingeniería se pueden encontrar problemas de optimización de naturaleza tanto discreta como continua y aunque ahora existen muchas técnicas de optimización, analíticas y numéricas, todavía hay un gran número de funciones que están más allá de los métodos analíticos y que presentan dificultades significativas para las técnicas numéricas. Como consecuencia, hay una continua búsqueda de nuevas y más robustas técnicas de optimización, que puedan solucionar tales problemas. Actualmente estamos viendo un creciente interés en la biología, motivado por encontrar la solución a los problemas de optimización. A partir de esto se han creado nuevas técnicas como Algoritmos Genéticos, en inglés Genetic Algorithms (GA), y Redes Neuronales en las últimas décadas. Entre las más recientes técnicas inspiradas en biología se encuentran Optimización por Colonia de Hormigas, en inglés Ant Colony Optimization (ACO) y Optimización por Enjambre de Partículas, en inglés Particle Swarm Optimization (PSO). PSO es una metaheurística propuesta por James Kennedy y Russell C. Eberhart en 1995, inspirada en el comportamiento social de organismos tales como bandada de pájaros, enjambre de abejas y cardúmenes de peces. La base de PSO está en la simulación de estos modelos sociales para así extrapolarlos a problemas de optimización reales que existen hoy en día.

La presente investigación realiza una contribución en este sentido, al estudiar la aplicabilidad de metodologías que, a partir de las variaciones de parámetros dinámicos de la estructura y la utilización de nuevas técnicas de optimización para la resolución del problema inverso, consiguen localizar y cuantificar el daño estructural.

1. DESCRIPCIÓN DEL DOCUMENTO

Este trabajo de grado se desarrolló con la intención de incentivar el desarrollo de la investigación en el campo de la inteligencia artificial y de detección de fallas en estructuras y elementos de maquinas, para que a través de la investigación y el uso de nuevas técnicas de optimización, la ingeniería local pueda estar a la vanguardia en estas áreas y por lo tanto posea herramientas alternativas para poder entender y resolver de una mejor manera problemas existentes y futuros.

A su vez podría considerarse como un complemento a una serie de proyectos orientados a la investigación y estudio de técnicas de optimización y de búsqueda, tales como en el diseño de elementos de maquinas y optimización de funciones; incorporando avances recientes en el campo de los algoritmos evolucionarios, donde PSO, uno de los algoritmos de mas reciente desarrollo, fue escogido como el algoritmo a utilizar, debido a su efectividad, simplicidad de implementación, aplicabilidad a problemas tanto de carácter continuo, como discreto con diversos grados de dificultad; entre otras características propias de éste. En este informe se pretende mostrar los resultados al incorporar estos tipos de algoritmos de una manera práctica al área de monitoreo de estructuras, cuyo aporte concierne a diversas áreas de ingeniería, tales como la ingeniería civil en el área estructural, la ingeniería mecánica en el área de mantenimiento y detección de fallas, entre otras. En esta disertación se profundiza en el estudio de las propiedades dinámicas de las estructuras, estudio de técnicas de detección de daños, algoritmos de evolución y la utilización de herramientas informáticas para la aplicación de estos conceptos. Tanto el procedimiento como los conceptos básicos de este estudio tienen como finalidad el incentivar al lector a aplicarlos en las áreas pertinentes a ellos, ya que estos principios pueden ser extrapolados a estructuras e incluso a elementos de maquinas, con diversas configuraciones, siempre y cuando su respuesta dinámica pueda ser obtenida a partir de

mediciones experimentales o de modelos analíticos adecuados; del mismo modo, la metodología implementada es susceptible a mejoramientos y aportes debido a que los algoritmos de optimización evolutivos se encuentran en constante desarrollo e investigación. Para facilitar la lectura del documento se ha decidido presentarlo de la manera más concisa posible, pero con la posibilidad de profundizar en el tema de interés del lector y observar resultados obtenidos por el autor más detalladamente a través de los anexos y referencias presentados al final del documento. Éste se encuentra organizado en cinco capítulos:

Capítulo 1: Se presenta una descripción general del trabajo desarrollado, la cual contiene las motivaciones que llevaron a su realización y el alcance del proyecto.

Capítulo 2: Se presenta una corta revisión de los principales conceptos empleados en el desarrollo del trabajo, tales como las propiedades dinámicas de las estructuras, y la detección de daños analizada como un problema de optimización.

Capítulo 3: Contiene un resumen de las principales características de la técnica conocida como optimización por enjambre de partículas (PSO), además de los parámetros que definen los respectivos algoritmos utilizados para detectar daños.

Capítulo 4: Una breve descripción de la metodología implementada es presentada. Se describen las estructuras analizadas, los escenarios de daño: cantidad de modos usados, cantidad de elementos afectados y las condiciones de ruido estudiados. Por último se explican los procedimientos utilizados para la determinación de las propiedades dinámicas de las estructuras bajo las diversas condiciones consideradas.

Capítulo 5: Son mostrados y discutidos los resultados de la aplicación de las metodologías de detección de daños estudiadas cuando son aplicadas sobre

diferentes estructuras en diversos escenarios de daño y ruido. Además se analiza la influencia de la cantidad de modos sobre los resultados.

Capítulo 6: Son presentadas las conclusiones del proyecto y algunas sugerencias para trabajos futuros.

Los capítulos que le siguen permiten al lector un acceso fácil a las bases teóricas sobre las cuales se fundamenta el proyecto sin entorpecer al lector que se encuentre familiarizado con los tópicos respectivos.

Apéndice 1: Se hace una revisión bibliográfica más detallada acerca de las metodologías de detección de daños y la teoría de métodos de elementos finitos.

Apéndice 2: Se realiza un ejemplo de análisis modal.

Apéndice 3: En este capítulo se encuentra en detalle la implementación de OPENSEES para la determinación de las propiedades dinámicas de las 2 estructuras estudiadas.

Apéndice 4: En esta sección se encuentra en detalle los archivos de MATLAB para la inclusión de daño, ruido y las dos funciones objetivo.

1.1 ANTECEDENTES

El uso de las técnicas de computación evolutiva, tales como los Algoritmos Genéticos, se han aplicado a la solución de problemas en el pasado, (Goldberg, 1989) aplico esta técnica inicialmente al diseño de armaduras planas, combinando elementos tanto de los algoritmos genéticos como del análisis por elementos finitos (FEM). Su objetivo fue disminuir la sección transversal de los miembros de la estructura de modo que pudiese reducir el consumo de materiales y el peso, con la restricción de un esfuerzo máximo, de lo cual obtuvo resultados satisfactorios.

En los últimos 20 años el interés en los métodos de búsqueda metaheurísticos para optimización ha aumentado de una manera significativa, en particular en lo que respecta a una de las vertientes más prolíficas en el campo de la optimización como son los **algoritmos evolutivos**. En los últimos 10 años se han hecho diferentes perspectivas teóricas que nos permiten entender de una mejor manera los algoritmos evolutivos, los cuales se encuentra en constante desarrollo llegando hasta nuevas técnicas como PSO y aquellas de tipo híbrido, utilizando las mejores características de cada método anteriormente establecido; por lo cual este tipo de algoritmos llegan hoy en día a tener una suma relevancia en diversos ámbitos del conocimiento humano, aunque si bien estas ideas se encuentran dispersas en la literatura, no existe un compendio de todas ellas, ni un punto final en su desarrollo e investigación, pues estos métodos siempre están abiertos a ser mejorados.

En lo que respecta a la Universidad Industrial de Santander, se han desarrollado diversos trabajos de grado enfocados a la utilización de algoritmos evolutivos para la resolución de un problema en particular, entre los cuales sobresalen en la Escuela de Ingeniería Mecánica, aquellos orientados al diseño óptimo por medio de algoritmos genéticos; en la Escuela de Ingeniería Civil, aquellos orientados a la optimización estructural y evaluación de desempeño de algoritmos evolutivos de tipo PSO e híbridos.

1.2 ALCANCE DEL PROYECTO

1.2.1 Objetivos General. Contribuir con el desarrollo de la misión de la Universidad Industrial de Santander mediante la aplicación de nuevas técnicas de optimización en el monitoreo de salud estructural, pretendiendo la continua generación y adecuación de conocimientos y la asimilación crítica de éstos en el campo de la Ingeniería Mecánica.

1.2.2 Objetivos Específicos

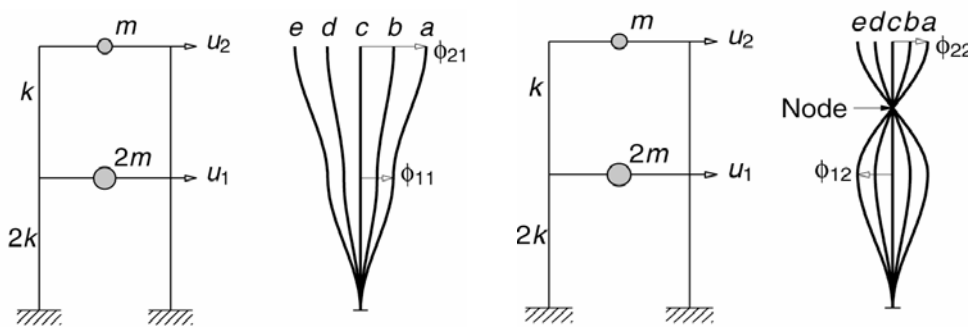
- Desarrollar un sistema de detección de daños (localización y cuantificación) utilizando los parámetros dinámicos de la estructura (frecuencias y formas de vibración) para la definición de la función objetivo a ser optimizada por medio de la técnica PSO (Particle Swarm Optimization) y verificar su aplicabilidad en el análisis de dos ejemplos básicos de estructuras: una viga discretizada en 10 elementos y una armadura de 13 elementos simplemente apoyadas, en diversos escenarios de daño (daño simple y múltiple).
- Utilizar dos tipos de función objetivo para comparar y analizar sus respectivos resultados.
- Analizar la influencia sobre los resultados de mediciones con presencia de ruido (porcentaje de error añadido al valor de las frecuencias y eigenvectores).
- Determinar el efecto de la cantidad de información utilizada, en este caso el número de modos de vibración analizados, sobre el desempeño de las metodologías implementadas.

2. REVISIÓN BIBLIOGRÁFICA

2.1 ANÁLISIS DINÁMICO

La teoría de dinámica de estructuras define que una estructura no amortiguada con múltiples grados de libertad posee un movimiento armónico simple sin cambiar la forma de deflexión (CHOPRA, 2001). Por ejemplo en un sistema de dos grados de libertad, existen dos formas de deflexión como se muestra en la Figura 1. A medida que se incrementa los grados libertad estudiados en la estructura, se incrementa el número de formas de vibración. Cada forma de deflexión o vibración es llamada como los modos naturales de vibración y dependen de las características de amortiguamiento, rigidez y masa de la estructura.

Figura 1. Modos de vibración de una estructura de 2 grados de libertad (CHOPRA, 2001)

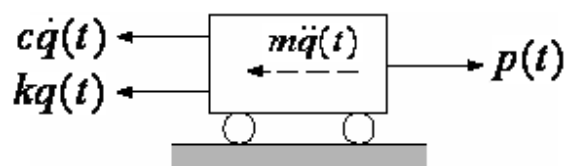


En el análisis dinámico de una estructura, ella puede ser considerada como continua o discreta dependiendo del nivel de análisis deseado. Normalmente las estructuras son idealizadas y discretizadas como puntos, hasta que el movimiento de estos, sea representativo del sistema. Cada uno de estos puntos puede tener un máximo de 6 grados de libertad: tres desplazamientos y tres rotaciones.

Existen en la actualidad tres formulaciones posibles para caracterizar el comportamiento dinámico de un sistema, las cuales son: la formulación clásica, la formulación de espacio de estados (State-Space) y modelos auto regresivos (AR). Solo la primera formulación es estudiada y empleada en el presente documento.

2.1.1 Formulación Clásica: Un solo grado de libertad (SDOF). La más importante diferencia entre un análisis estático y uno dinámico es la dependencia del tiempo. En un análisis dinámico es posible determinar la respuesta de la estructura, en términos de desplazamientos, velocidad, aceleración y fuerzas internas en cada instante de tiempo. Cada una de estas variables es asociada a la aceleración que toma lugar por las fuerzas inerciales que produce la excitación externa. El movimiento del sistema puede ser expresado matemáticamente por un equilibrio de fuerzas en cada instante de tiempo (principio de A'lembert). Las 4 fuerzas son respectivamente, la fuerza inercial, de amortiguamiento, fuerza elástica y la de excitación (Figura 2.). Si $q(t)$ es definido como el desplazamiento en el tiempo de un sistema de un solo grado de libertad, la ecuación de movimiento es dada por:

Figura 2. Equilibrio dinámico



$$m\ddot{q}(t) + c\dot{q}(t) + kq(t) = p(t) \quad (2.1)$$

Donde m es la masa del sistema, c es la constante de amortiguamiento, k es la rigidez del sistema y $p(t)$ es el vector de carga el cual es también dependiente del

tiempo. Las derivadas en el tiempo $\ddot{q}(t)$ y $\dot{q}(t)$ representan la aceleración y velocidad respectivamente. La solución de esta ecuación diferencial de segundo orden puede ser obtenida analíticamente por medio de la integral de Duhamel (CHOPRA, 2001), válida para sistemas lineales, y está dada por la siguiente expresión:

$$q(t) = \frac{1}{m\omega_D} \int_0^t p(\tau) e^{-\xi\omega_n(t-\tau)} \sin[\omega_D(t-\tau)] d\tau \quad (2.2)$$

Donde τ es el tiempo total de referencia, ω_D es la frecuencia amortiguada que puede ser relacionada a la frecuencia natural ω_N o al periodo natural T_N , $\omega_N = \sqrt{k/m} = 2\pi/T_N$.

$$\omega_D = \omega_N \sqrt{1 - \xi^2} \quad (2.3)$$

Y ξ es el coeficiente de amortiguamiento, el cual para estructuras de ingeniería civil es menor al 20%, el coeficiente de amortiguamiento está dado por la siguiente expresión:

$$\xi = \frac{c}{2m\omega_N} \quad (2.4)$$

Donde c es la constante de amortiguamiento.

2.1.2. Formulación Clásica: Sistemas de múltiples grados de libertad (MDOF).

Cuando los sistemas tienen múltiples grados de libertad, la ecuación de movimiento debe ser obtenida considerando equilibrio dinámico sobre cada uno de los nodos que se considere en el modelo, esto se configura en unas matrices de rigidez, amortiguamiento y masa de la estructura y se puede representar de la siguiente manera.

$$M\ddot{q}(t) + C\dot{q}(t) + Kq(t) = p(t) \quad (2.5)$$

Donde M, C y K son matrices de masa, amortiguamiento y rigidez de orden $n \times n$, respectivamente, con componentes m_{ik} , c_{ik} y k_{ik} que representan las fuerzas generalizadas para cada coordenada i , cuando la aceleración es aplicada en una coordenada k . $p(t)$ es un vector de excitación y $q(t)$ es el vector de respuesta del sistema.

Una alternativa para solucionar la Ecuación 2.5 es la aproximación modal, donde se empieza asumiendo un problema no amortiguado, estableciendo la ecuación diferencial homogénea como:

$$M\ddot{q}(t) + Kq(t) = 0 \quad (2.6)$$

Una solución a ésta ecuación diferencial está dada por:

$$q(t) = \Phi_i \text{sen}(\omega_i t - \alpha_i) \quad (2.7)$$

Donde Φ_i es la i - forma del modo con una correspondiente frecuencia natural circular ω_i y ángulo de fase α_i . Substituyendo la ecuación 2.7 en la ecuación 2.6, se elimina

$\text{sen}(\omega_i t - \alpha_i)$ y se obtiene:

$$\left[K - \omega_i^2 M \right] \Phi_i = 0 \quad (2.8)$$

La ecuación (2.8) puede ser escrita como un sistema de N - ecuaciones por lo cual puede ser dada la ecuación (2.9):

$$\begin{pmatrix} k_{11} - \omega_1^2 m_1 & k_{12} & \dots & k_{1N} \\ k_{12} & k_{22} - \omega_2^2 m_2 & \dots & k_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ k_{N1} & k_{N2} & \dots & k_{NN} - \omega_N^2 m_N \end{pmatrix} \begin{Bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_N \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{Bmatrix} \quad (2.9)$$

La enunciación de las ecuaciones 2.8 y 2.9 es un problema matemático importante, el cual se denomina “eigenproblema” (valor propio). La particularidad del problema de valor propio, es que la solución no es única. Para entrar en detalle, Φ_i que se calcula en el análisis modal, no representa las amplitudes del sistema bajo vibración libre, pero aun así, los cocientes normalizados de la amplitud, cuando están combinados correctamente, pueden proporcionar la reacción dinámica del sistema como se muestra posteriormente.

La función del análisis modal es el cómputo de ω_i y la reciprocidad correspondiente de Φ_i , que compensa a la ecuación 2.8 obteniendo una solución no trivial, esto es, una solución para la cual todos los ω_i y Φ_i no son cero, se requiere que el determinante de la ecuación 2.8 sea cero, por eso:

$$\text{Det}[K - \omega_i^2 M] = 0 \quad (2.10)$$

La ecuación 2.10 es una ecuación polinomial conocida como la *ecuación característica* del sistema. Para cada solución $\omega_i^2 (i = 1, 2, \dots, N)$ de la ecuación característica podemos resolver la ecuación 2.9 para Φ_i . Una solución ω_i^2 de la ecuación característica es llamada un eigenvalor (valor propio). Para cada uno de los eigenvalores corresponde un eigenvector, por lo tanto para un sistema con N grados de libertad hay N eigenvectores y sus correspondientes eigenvalores. En dinámica, ω_i es llamado la frecuencia natural circular y los correspondientes eigenvectores son llamados la forma del modo; es decir, el eigenvector y la forma

del modo son equivalentes. La frecuencia natural circular más pequeña ω_1 es llamada la *frecuencia circular fundamental*, y su correspondiente forma del modo Φ_1 es llamado el *modo fundamental* de vibración.

Aquí debe de ser de ayuda exhibir una interpretación física de una forma de modo. La ecuación 2.8 formula equilibrio entre la inercia y las fuerzas de dureza en una estructura en un tiempo t. si se reformula ésta ecuación de tal forma que:

$$K\Phi_i = \omega_i^2 M\Phi_i \quad (2.11)$$

Una forma del modo puede ser vista como una deflexión estática resultante de la fuerzas del lado derecho de la ecuación 2.11.

2.2 METODOLOGIA EMPLEADA PARA LA LOCALIZACION DE DAÑO

Diversas metodologías para la localización de daño estructural basadas en las variaciones de parámetros modales fueron formuladas en las últimas décadas. Dentro de estas se encuentran metodologías de localización de daño que utilizan como parámetros de entrada variaciones de la energía de deformación modal, variaciones de formas modales y/o frecuencias naturales, vectores de fuerza residual y variaciones en la matriz de flexibilidad de la estructura. Considerando estas técnicas mencionadas se han desarrollado funciones objetivo, las cuales son utilizadas para considerar el proceso de detección de daños, como un proceso de optimización, el cual fue el método escogido para el desarrollo del presente trabajo, por lo cual sólo se destaca el estudio de esta metodología.

2.2.1 La Detección de Daños como un Proceso de Optimización. El problema de detección de daños puede ser tratado como un problema de optimización dado que todos los algoritmos buscan minimizar las diferencias entre el comportamiento descrito por el modelo analítico y los datos experimentales. Las características propias del problema de detección de daños, mediciones incompletas y presencia de ruidos, hacen de éste un problema de cierto grado de dificultad, y, por lo tanto, requiere que el método sea robusto en el sentido de permitir exitosamente el abordaje de una variada gama de problemas.

En la implementación de un proceso de optimización pueden ser utilizadas técnicas convencionales de optimización o las denominadas metaheurísticas. Las primeras presentan una alta dependencia del punto de partida y podrían ser ineficientes en espacios de búsqueda altamente complejos. Por otro lado, las técnicas metaheurísticas se han convertido en herramientas importantes para la solución de problemas altamente no lineales como aquellos que presentan múltiples óptimos locales. Entre las principales metaheurísticas ya utilizadas en detección de daños se encuentran las Redes Neuronales Artificiales, *Simulated Annealing*, *Particle Swarm Optimization* y Algoritmos Genéticos.

A continuación se presentada una revisión de las funciones objetivo empleadas en el presente trabajo para la detección de daños.

Friswell, Penny y Garvey (1998) estudiaron un AG clásico de dos etapas. Primero, los lugares damnificados, máximo dos, son determinados y después es utilizado un método de sensibilidad de valores propios para la determinación de la extensión del daño. La función objetivo fue definida como una combinación lineal de las variaciones en las frecuencias naturales y de las variaciones en las formas modales:

$$Mtn f = \sum_{j=1}^r W_{w_j} \left(\frac{\delta w_{m_j} - \delta w_{a_j}}{w_{m_j}} \right)^2 + \sum_{j=1}^r W_{\phi_j} [1 - MAC(\phi_{m_j}, \phi_{a_j})] + W_{ns} \delta_{ns} \quad (2.12)$$

En la cual el subíndice m es referido para los datos que fueron medidos o simulados y el subíndice a para aquellos obtenidos a partir del modelo analítico. δw son las variaciones en las frecuencias naturales, w_j y ϕ_j corresponden a la j -ésima frecuencia natural y forma modal, respectivamente. Los términos W son factores de peso para cada una de las características dinámicas, en la presente investigación se asumieron pesos iguales para las frecuencias y formas modales. δ_{ns} asume un valor de cero si tiene un lugar damnificado y un valor de uno si tiene más que un lugar damnificado. r es el número de formas modales utilizadas para la comparación y s es el número de componentes del vector de la j -ésima forma modal (numero de grados de libertad). El término correspondiente a las variaciones en las frecuencias fue pensado para reducir el efecto de errores de modelamiento, en cuanto que el segundo termino ayuda a eliminar la necesidad de formas modales escaladas.

Donde el término MAC es el Criterio de Confianza Modal (Modal Assurance Criterion), el cual es un método para estimar el grado de correlación entre un modo natural de vibración del modelo virtual y un modo del experimental. Proporciona un factor de correlación entre cada una de las posibles combinaciones entre los modos teóricos y experimentales de la forma:

$$MAC_{ij} = \frac{(\phi_i^T \phi_j)}{(\phi_i^T \phi_i)(\phi_j^T \phi_j)} \quad (2.13)$$

Donde ϕ_i es el eigenvector del modo i que contiene solo los grados de libertad medidos y ϕ_j es el correspondiente al valor experimental del eigenvector del modo j .

Esta función objetivo corresponde al archivo de MATLAB llamada CalibraV.m.

Moslem y Nafaspour (2002) presentaron un método de dos etapas para localización y cuantificación de daños. Posibles áreas damnificadas, las cuales ayudan a disminuir el tamaño del espacio de búsqueda, son determinadas a partir del cálculo del vector de fuerza residual y la compatibilización entre los grados de libertad analíticos y experimentales.

Para la resolución del problema el daño es considerado como una reducción del módulo de elasticidad y una función objetivo, basada en variaciones de las frecuencias naturales y formas modales, definida como:

$$F = \sum_{j=1}^n W_{w_j} \left[1 - \left(\frac{W_{m_j}}{W_{a_j}} \right) \right]^2 + \sum_{j=1}^n \sum_{i=1}^s W_{\phi_{ji}} (\phi_{m_{ji}} - \phi_{a_{ij}}) \quad (2.14)$$

Esta función objetivo corresponde al archivo de MATLAB llamado CalibraV1.m.

3. OPTIMIZACIÓN POR ENJAMBRE DE PARTÍCULAS

Los modelos matemáticos de los sistemas cumplen un rol esencial en aplicaciones como simulación, predicción, control y diagnóstico en campos tan variados como la ingeniería, la economía, la medicina, y la fisiología, entre otros. Es teóricamente posible obtener modelos matemáticos de un sistema usando los principios de la física, la química, la biología, etc. Sin embargo, en muchos sistemas la composición de éstos ocasiona que la derivación del modelo sea una tarea compleja. En estas circunstancias el modelo matemático del sistema puede ser construido a partir de datos experimentales. El proceso de desarrollo de modelos a partir de datos experimentales es llamado identificación de sistemas. Este proceso consiste en el desarrollo y análisis de métodos para obtener modelos de cajas grises y cajas negras de sistemas.

Actualmente, muchas investigaciones en el área de identificación de sistemas se han orientado hacia el uso de algoritmos de procesamiento paralelo como son los Algoritmos Genéticos (AG) y *Particle Swarm Optimization* (PSO). El uso de estas técnicas se prefiere en procesos de modelamiento complejos en donde la función objetivo del proceso de identificación del sistema puede quedar atrapada en mínimos locales, lo cual dificulta el proceso de convergencia hacia la solución óptima. Por otro lado, las anteriores técnicas aseguran convergencia y la obtención al menos del mejor mínimo local.

3.1 DESCRIPCIÓN GENERAL DE PARTICLE SWARM OPTIMIZATION Y ALGORITMOS DE OPTIMIZACIÓN INTELIGENTES

El algoritmo PSO pertenece a las técnicas denominadas optimización inteligente y se clasifica como un algoritmo estocástico de optimización basado en población. A esta clasificación igualmente pertenecen los Algoritmos Genéticos (AG). Los

anteriores algoritmos se consideran adecuados comparados con los métodos clásicos de optimización cuando el problema de optimización es complejo, estocástico o no lineal con múltiples mínimos locales.

Las ventajas atribuidas a las técnicas inteligentes de optimización son su paralelismo intrínseco, su capacidad para resolver problemas complejos, de gran tamaño, y con un mínimo conocimiento del sistema que se está identificando. Estas ventajas se discuten a continuación.

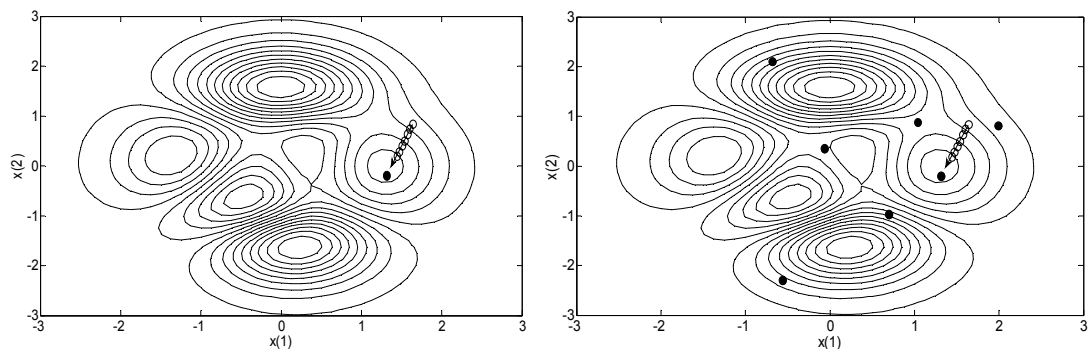
PSO es intrínsecamente paralelo. La mayoría de algoritmos clásicos operan secuencialmente y pueden explorar el espacio de solución solamente en una dirección a la vez. Esta diferencia entre algoritmos clásicos y algoritmos inteligentes se ilustra en la Figura 3.1. En esta figura se puede observar que en el evento en que la solución determinada no es la óptima, el algoritmo clásico la desecha e inicia la búsqueda en otra dirección. Por otra parte, el algoritmo de optimización inteligente puede explorar el espacio de la solución en múltiples direcciones simultáneamente. Si un camino no conduce al óptimo, el algoritmo fácilmente elimina tal camino y continúa buscando otros mejores. Esta forma de operación proporciona una mayor probabilidad de encontrar la solución óptima.

Debido al paralelismo, los algoritmos inteligentes de optimización son adecuados para problemas con gran número de variables, donde el espacio de la solución es de gran tamaño para realizar una búsqueda exhaustiva en un tiempo razonable. Otra fortaleza de los algoritmos de optimización inteligente es su buen desempeño en problemas cuyo espacio de solución presenta múltiples mínimos locales.

Muchos algoritmos clásicos de búsqueda como *backpropagation*, usado para entrenar redes neuronales, pueden quedar atrapados en mínimos locales. Los algoritmos de optimización inteligente han demostrado ser efectivos evitando mínimos locales e identificando el mínimo global en un espacio de búsqueda

complejo. Debe notarse que en ocasiones, no hay manera de determinar si una solución es un mínimo global o solamente un muy buen mínimo local. Sin embargo, aún cuando el algoritmo de optimización inteligente no puede encontrar el mínimo global, usualmente encuentra un buen mínimo local.

Figura 3. Algoritmos secuenciales vs. Algoritmos inteligentes



Finalmente, los algoritmos de optimización inteligentes no requieren de un conocimiento detallado de la estructura y comportamiento del sistema como otros algoritmos de optimización la requieren. En lugar de usar información específica del sistema para guiar la búsqueda, este algoritmo hace cambios aleatorios en las soluciones candidatas y usa una función de ajuste multivariable para determinar si las soluciones son óptimas o no.

Como un nuevo desarrollo de los algoritmos de optimización inteligentes, *Particle Swarm Optimization* es simple en concepto y altamente eficiente desde el punto de vista computacional.

3.2 PARTICLE SWARM OPTIMIZATION

PSO es un algoritmo estocástico de búsqueda basado en población, propuesto por *Kennedy* y *Eberhart* en 1995 como un modelo de las actividades sociales de insectos, pájaros y peces. Este algoritmo pretende representar el proceso natural

de comunicación grupal para compartir conocimiento individual cuando grupos de animales se desplazan, migran o cazan. Si un miembro detecta un camino deseable para desplazarse, el resto de la colonia lo sigue inmediatamente. En PSO, este comportamiento animal es imitado por partículas con ciertas posiciones y velocidades en un espacio de búsqueda, donde la población es llamada *swarm* (*enjambre*), y cada miembro del *swarm* es llamado partícula. La población inicial se determina aleatoriamente, aunque existe la opción de manualmente introducir los valores de las posiciones iniciales de las partículas en el espacio de búsqueda del problema; cada partícula se desplaza a través de dicho espacio de búsqueda y recuerda la mejor posición que ha encontrado. Cada partícula comunica las buenas posiciones a las demás y dinámicamente ajustan su propia posición y su velocidad con base en las buenas posiciones. La velocidad se ajusta con el comportamiento histórico de las partículas. De esta forma, las partículas tienden a dirigirse hacia un mejor espacio de búsqueda en el proceso de minimización de la función objetivo. Este procedimiento de búsqueda se describe por las ecuaciones 3.1 y 3.2.

$$v_i^{k+1} = w.v_i^k + c_1.rand_1.(pbest_i - x_i^k) + c_2.rand_2.(gbest - x_i^k) \quad (3.1)$$

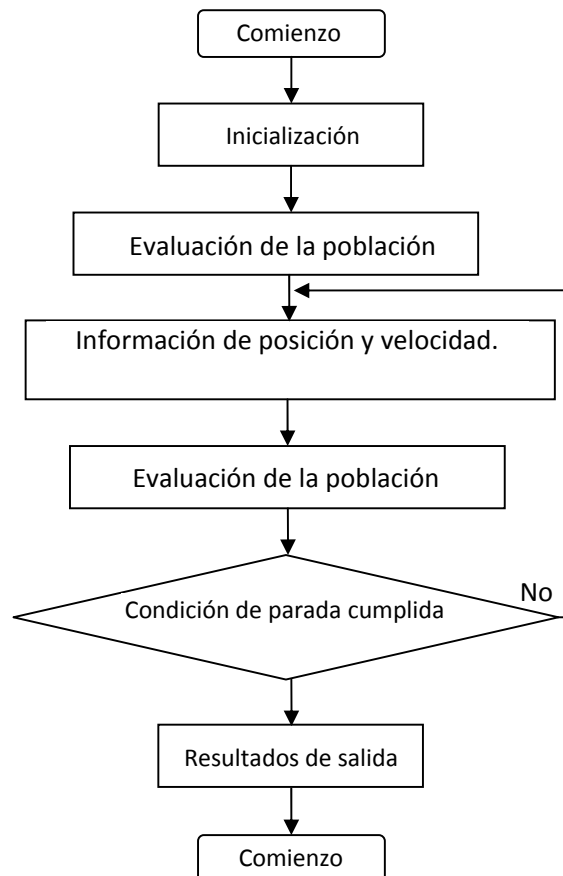
$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (3.2)$$

En 3.1 c_1 y c_2 son constantes positivas, definidas como coeficientes de aceleración; w es el factor inercial; $rand_1$ y $rand_2$ son dos números aleatorios (con distribución de probabilidad uniforme) en el rango $[0,1]$; x_i representa la *iésima* partícula y $pbest_i$ la mejor posición previa de x_i ; $gbest$ es la posición de la mejor partícula de toda la población; y v_i es la razón de cambio de la posición (velocidad) de la partícula x_i . Los cambios de velocidad en 3.1 se componen de tres partes: *momentum*, *cognitiva* y *social*. De esta forma se obtiene una velocidad que tiende a acercar la partícula a $pbest$ y $gbest$.

De acuerdo con la ec. 3.1, el factor inercial w y los dos coeficientes de aceleración c_1 y c_2 afectan el desempeño del algoritmo PSO. Desde el planteamiento de este algoritmo en 1995, una gran cantidad de trabajos se han desarrollado para mejorar la versión original modificando estos tres factores. La variación de estos factores puede balancear la búsqueda global y local durante el proceso de optimización. Las constantes de aceleración sirven para dos propósitos en el algoritmo. Primero, ellas controlan la influencia relativa hacia $gbest$ y $pbest_i$ respectivamente. Segundo, los dos coeficientes de aceleración cercanos a cero producirán una búsqueda fina en una región, mientras coeficientes cercanos a uno permitirán a la partícula la posibilidad de sobrepasar al $gbest$ y al $pbest_i$, resultando en una búsqueda amplia:

- la posición y la velocidad de las partículas están por fuera de los correspondientes límites, éstos se ajustan a los valores establecidos.
- *Evaluación de la población actualizada.* Similar al anterior, la posición actualizada de las partículas es evaluada de acuerdo al valor de la función objetivo, la cual es el criterio para evaluar el proceso de estimación de los parámetros y representa el grado de cercanía entre la salida del sistema real comparada con la salida del modelo cuyos parámetros se identifican usando PSO; las partículas $gbest$ y la $pbest$ serán actualizadas si es necesario.
- *Verificación del cumplimiento de la condición de parada.* Esta condición puede corresponder con el número de iteraciones o con el mínimo valor de la función objetivo. Si la condición de parada no ha sido cumplida, el proceso de actualización será repetido; de lo contrario, el proceso de optimización finaliza.
- *Resultados de salida.* La mejor solución obtenida durante el proceso de optimización, $gbest$, es la salida en este paso.

Figura 4. Diagrama de flujo del algoritmo PSO



En identificación de sistemas, un sistema con una estructura de modelo conocido pero con valores de parámetros desconocidos puede tratarse como una caja gris y resolverse como un problema de optimización. La idea básica es comparar la salida del sistema con la salida del modelo. La discrepancia entre las salidas del sistema y del modelo corresponde a la función objetivo a minimizar. Esta función se define con la medida de cuán bien la salida del modelo se acerca a la medida real de la salida del sistema.

Al igual que otros algoritmos de optimización, PSO no garantiza el éxito. La probabilidad de éxito se basa en la cantidad de partículas y el tamaño del espacio de búsqueda. Para un espacio de búsqueda grande, debe utilizarse una población elevada para mejorar la tasa de búsqueda exitosa. No obstante, esto involucra más cálculos, y la velocidad de búsqueda puede reducirse significativamente. Por lo tanto, es necesario introducir un espacio de búsqueda razonable y eficiente.

3.3 HERRAMIENTA PSOT PARA MATLAB

Para la implementación del algoritmo se utilizó la herramienta (toolbox) PSOT disponible para entorno MATLAB desarrollada por Brian Birge el 22 de Abril de 2005.

Esta herramienta se puede descargar desde la siguiente dirección en Internet:<http://www.mathworks.com/matlabcentral/fileexchange/7506>. En este trabajo se utilizó la versión actualizada el 20 de marzo de 2006. Los resultados de la optimización se obtuvieron con la función `pso_Trelea_vectorized.m` cuyos parámetros se introducen a través del m-file `Princ.m` de la toolbox descargada, la cual se presenta a continuación:

```
funcname='calibraV1_A2';
D=13;
mv=0.4;
VarRange=[0.05 2;0.05 2;0.05 2;0.05 2;0.05 2;0.05 2;0.05 2;0.05 2;0.05 2;0.05
2;0.5 2;0.05 2;0.05 2];      %Ei
minmax=0;
PSOparams=[1,1500,60,10,10,0.9,0.4,1500,1e-5,50,NaN,2,0];
plotfcn='goplotpso';
%PSOseedValue=ones(60,13).*2;
```

```
[optOUT,tr,te]=
```

```
pso_Trelea_vectorized(funcname,D,mv,VarRange,minmax,PSOparams,plotfcn,0);
```

A seguir se encuentra una breve descripción de este archivo, junto con el modo de uso y los datos introducidos para el problema de detección de daños, si no se menciona ningún cambio, significa que se usaron los valores por defecto.

- **Descripción:** Esta toolbox es un optimizador de enjambre de partículas genérico para encontrar el mínimo o máximo de cualquier función de múltiples entradas y de una sola salida (*MISO*, Multiple input single output).

Puede implementar los tipos de PSO Common (Común con influencia inercial), Trelea tipo 1 y 2, y Clerc clase 1. El algoritmo además tratará de automáticamente rastrear y seguir a un ambiente cambiante con éxito variado.

Esta versión por medio de la definición de los parámetros en vectores, elimina el ciclo “for” asociado con el número de partículas y maneja todas las dimensiones de búsqueda y parámetros de entrada en una sola variable.

- **Modo de uso:**

Entradas:

- **funcname:** archivo de MATLAB que contiene la función a optimizar. CalibraV o CalibraV1, según se escoja la función objetivo (mencionadas en el anterior capítulo).
- **D =** Número de dimensiones del problema, 10 para la viga, 13 para la armadura.

Entradas Opcionales:

- mv: máxima velocidad de partícula, puede ser especificada como un escalar o un vector de longitud D, lo cual permite que cada componente tenga su propia velocidad máxima. Su valor por defecto es $mv = 4$, establecido si no se especifica lo contrario o se introduce un valor que no sea un número. Para este trabajo se especificó este valor como 0.4 para dar mayor sensibilidad a las partículas en el espacio de búsqueda.
- VarRange: matriz de rangos para cada variable de entrada, por defecto -100 a 100, de la forma: [min1 max1; min2 max2;... minD maxD]. Valor mínimo_i 0.05 y valor máximo_i E (módulo de elasticidad del material de la estructura sin daño) para $i=1,2,\dots,D$.

- minmax = 0, la función es minimizada (por defecto)
= 1, la función es maximizada
= 2, la función es enfocada a P(12)(minimiza distancia a error deseado)

Se utiliza el valor por defecto ya que se desea minimizar el error.

- PSOparams: Parámetros de PSO.

P(1): Épocas para actualizar gráfico, por defecto=100, si es 0 = no grafica. Se estableció una iteración, como cantidad de épocas para la actualización de la interface grafica, con el ánimo de observar más detalladamente la dinámica de las partículas.

P(2): Máximo número de iteraciones (épocas) del ciclo, por defecto = 2000.

P(3): Tamaño de la población, por defecto = 24. El tamaño de la población se estableció de 60 partículas ya que el espacio de búsqueda con varios elementos en el caso de detección de daños es de un tamaño considerablemente extenso.

P(4): aceleración constante 1 (influencia del mejor valor personal pbest), por defecto=2. Se estableció un valor de 10 para darle mayor influencia a este factor en el movimiento de las partículas.

P(5): aceleración constante 2 (influencia del mejor valor del enjambre Gbest), por defecto=2. Se estableció un valor de 10 para darle mayor influencia a este factor en el movimiento de las partículas.

P(6): Peso inicial de inercia, por defecto = 0.9.

P(7): Peso final de inercia, por defecto = 0.4.

P(8): Valor final de épocas cuando se asigna peso inercial, por defecto = 1500.

P(9): Mínimo gradiente de error global, if $\text{abs}(\text{Gbest}(i+1)-\text{Gbest}(i)) < \text{gradient}$ durante cierta longitud de épocas, termina la corrida, por defecto=1e-25. Se tomó como 10^{-5} (1e-5) para disminuir el tiempo de ejecución del algoritmo, Esta condición genera cierta ambigüedad, ya que es posible que el gradiente no cambie en al menos 10^{-5} durante la cantidad de épocas designadas en P(), pero que si lo haga posteriormente a estas épocas de estancamiento. Para tomar una decisión acertada hay que sopesar el tiempo de ejecución y la precisión en el resultado del algoritmo.

P(10): Épocas antes de que el criterio del error del gradiente termine de correr, por defecto = 150. Se emplearon 50 para reducir tiempo de ejecución.

P(11): error deseado, si su valor es NaN (Not a number) entonces, minimizar o maximizar sin restricción, por defecto=NaN.

P(12): Tipo de PSO para utilizar []. Se utilizó P(12)=2 (Trelea) por su mejor desempeño y convergencia para los casos estudiados.

0 = PSO Común con inercia (por defecto).

1,2 = PSO tipo Trelea 1,2.

3 = PSO restringido de Clerc, Tipo 1"

P(13): PSOseed, por defecto=0

= 0 para posiciones iniciales aleatorias

= 1 para partículas iniciales ingresadas por el usuario.

- plotfcn: nombre opcional de función encargada de graficar, por defecto 'goplotpso', (se puede ingresar otra función creada por el usuario).

- PSOseedValue: Posición inicial de partículas, depende de P(13), el cual deber estar definido como 1 or 2, (no usado para P(13)=0). Este valor necesita ser de dimensiones $n \times m$ donde $n \leq$ Numero de partículas, y $m \leq$ Número de dimensiones. Si $n < p_s$ y/o $m < D$, entonces los valores restantes son escogidos aleatoriamente en Varrange.

Salidas:

- [optOUT]=PSO(funcname,D), Entradas opcionales y salidas asociadas min/max de la function, de la forma: [bestin1; bestin2; ... bestinD; bestOUT]^T.

Salidas Opcionales:

- [optOUT,tr,te]=
PSO(funcname,D,mv,VarRange,minmax,PSOparams,plotfcn,PSOseedVaue)

tr: Gbest en cada iteracion, sigue el vuelo del enjambre.

te: épocas alcanzadas, retorna como un vector de 1:época final.

En el Anexo A se encuentra el código utilizado para definir las funciones objetivo y para obtener los datos de entrada a ser comparados con los generados con PSO.

4. PROCEDIMIENTO GENERAL

4.1 METODOLOGIA

Para el desarrollo de la presente investigación se escogió seguir dos líneas de trabajo. La primera está relacionada con la obtención del conocimiento necesario para comprender la fundamentación física de las técnicas que utilizan parámetros dinámicos para detección de daños, y la otra trata de la implementación computacional de nuevas técnicas de optimización que permitan determinar el daño en estructuras.

La fundamentación teórica, fue realizada a través de una revisión bibliográfica sobre las principales metodologías de detección de daños propuestas más recientemente, que utilizan parámetros vibratoriales de la estructura. Del mismo modo, énfasis fue dado a las metodologías que permiten localizar elementos damnificados y aquellas que consideran el problema de detección de daños como un problema de optimización. En relación a la técnica de optimización aplicada (PSO), los fundamentos de ésta fueron estudiados.

Las metodologías anteriores serán probadas en 2 tipos de estructuras mencionadas inmediatamente después de este numeral, y para esto son tomadas en cuenta las siguientes consideraciones:

- Las propiedades dinámicas de la condición no damnificada son obtenidas a partir del modelo analítico original de elementos finitos de la estructura, utilizado por el software escogido OPENSEES² y SAP 2000³ en el cálculo de dicha respuesta.

² Software de libre distribución desarrollado por Berkeley University OPENSEES: <http://opensees.berkeley.edu/index.php>

³ Software estructural SAP2000 versión 7.4 desarrollado por Berkeley University: <http://www.compengineering.com/products/SAP2000/sap2000.html>

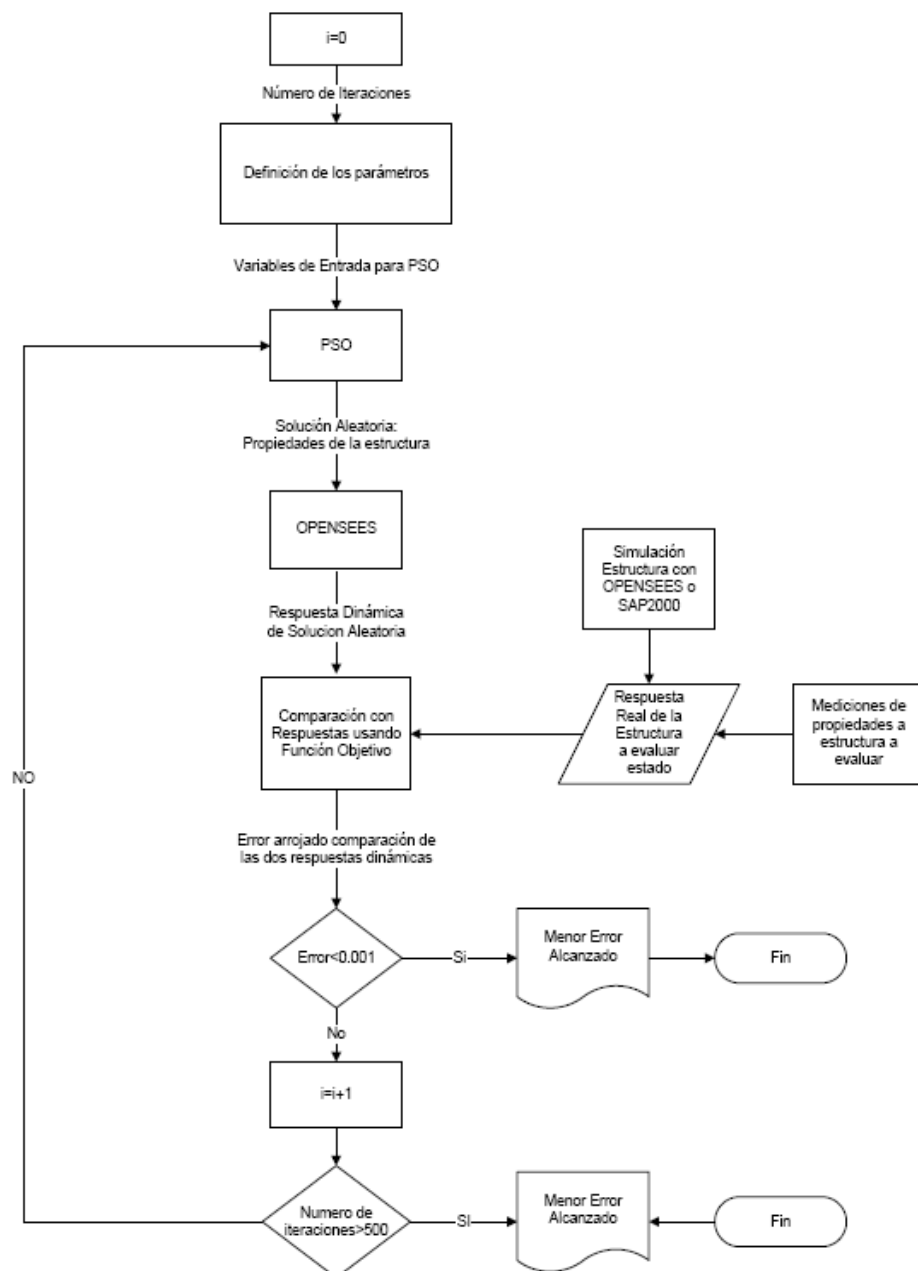
- La respuesta de la estructura, después de haber sido damnificada, se encuentra en régimen lineal.
 - La estructura no presenta amortiguamiento (en estructuras civiles éste es despreciable).
 - El daño es definido por medio de una disminución de la rigidez del (de los) elemento(s) damnificado(s).
-
- Para el estudio de la condición damnificada, el daño será introducido a través de un algoritmo de MATLAB dentro del modelo de OPENSEES. Por medio de un análisis directo del modelo, son calculadas las propiedades modales del estado damnificado. Estas propiedades serán utilizadas, entonces, para realizar el procedimiento inverso, es decir, a partir de ellas será determinada la condición actual de la estructura (rigidez de cada elemento) con esa respuesta.
 - Los datos dinámicos considerados corresponden únicamente a frecuencias naturales y formas modales, por lo tanto, la función objetivo empleada está basada en esos parámetros.

La comparación del desempeño de los diferentes algoritmos ha sido realizada con base en los siguientes criterios:

- Daños simples (un sólo elemento) y daños múltiples (más de un elemento).
- Función objetivo utilizada.
- Presencia de ruido en las mediciones. Valores típicos de niveles de ruido, encontrados durante la determinación experimental de frecuencias y modos, son incluidos en los parámetros dinámicos (frecuencias y formas modales) de la condición damnificada.
- Cantidad de modos de vibración tomados.
- Localización de los elementos damnificados.

El diagrama de flujo de la metodología de detección de daños empleada se muestra en la figura 4.1. Se aprecian cada uno de los procesos desarrollados por el algoritmo principal.

Figura 5. Diagrama de flujo del sistema



4.2 ESTRUCTURAS ANALIZADAS

Las metodologías de detección de daño han sido estudiadas y aplicadas a dos tipos de sistemas estructurales planos: viga y armadura. Las estructuras tomadas como ejemplos son mostradas en las Figuras 6 y 7.

Viga: Muchas estructuras y/o elementos de máquinas están compuestos o se pueden modelar como una viga, elemento que a pesar de su simplicidad, es en la mayoría de los casos un componente esencial, que se debe analizar por separado para una inspección estructural minuciosa. La viga a analizar posee las siguientes características:

Viga simplemente apoyada de longitud total $L=6$ m, discretizada en 10 elementos de longitud 0.6 m, además posee una sección transversal con Área $A= 0.12$ m², Momento de Inercia $I_z=1.6 \times 10^{-3}$ m⁴. Material: Concreto 3000 Psi, Módulo de Elasticidad $E= 2.153 \times 10^{10}$ Pa, Masa por Unidad de Volumen= 2402.7696 kg/m³.

Armadura: Modelando una cercha, se puede comprobar el método de detección de daños en una estructura más compleja. La cercha es uno de los principales tipos de estructuras empleadas en ingeniería, especialmente en el diseño de puentes y edificios. Una armadura consta de barras rectas unidas mediante juntas o nodos. Los elementos de una cercha se unen sólo en los extremos por medio de pasadores sin fricción para formar una armazón rígida. Cada cercha se diseña para que soporte las cargas que actúan en su plano y sean aplicadas en las uniones y no en los mismos elementos; en consecuencia, una armadura puede considerarse como una estructura bidimensional, donde sus elementos están sometidos solo a fuerzas axiales directas (tracción o compresión). La cercha a analizar posee las siguientes características:

Armadura de 13 elementos, simplemente apoyada, cuyas dimensiones son: altura de 8 ft (2.4384 m) y longitud total de la base de 24 ft (7.3152 m), los elementos horizontales están uniformemente espaciados, es decir cada uno con una longitud de 6 ft (1.8288 m)

Elementos con sección transversal de área $A= 4.0 \times 10^{-4} \text{ m}^2$. Material: Acero A-36 con modulo de elasticidad $E= 2.0 \times 10^{11} \text{ Pa}$.

Figura 6. Viga simplemente apoyada

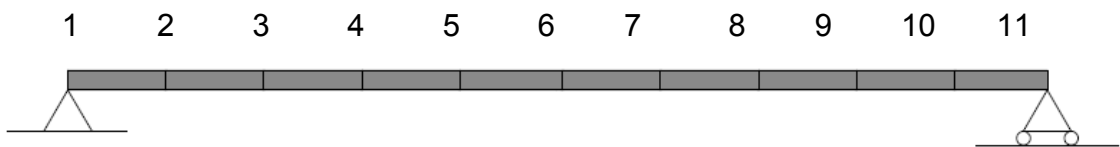
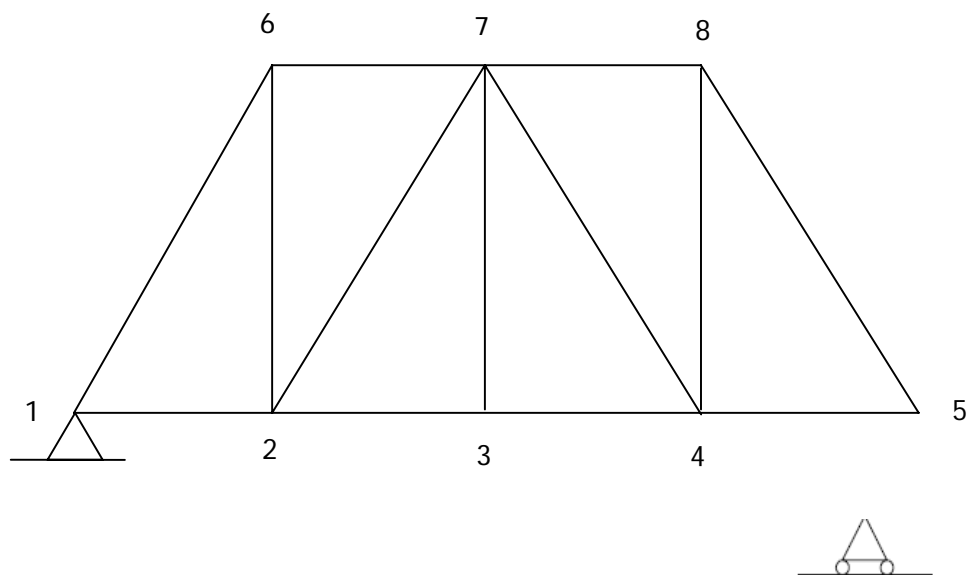


Figura 7. Armadura de 13 elementos simplemente apoyada



Nota: Todas las dimensiones están dadas en metros.

Es importante resaltar que la precisión del modelo en la localización del daño es directamente proporcional a la cantidad de elementos en que es subdividida la estructura.

La conectividad entre los distintos elementos que pertenecen a cada una de las estructuras es presentada en las Tablas 1 y 2.

Tabla 1. Conectividad de los elementos en la estructura tipo Viga

Elemento	Nodo Inicial	Nodo Final
I	1	2
II	2	3
III	3	4
IV	4	5
V	5	6
VI	6	7
VII	7	8
VIII	8	9
IX	9	10
X	10	11

Tabla 2. Conectividad de los elementos en la estructura tipo Viga

Elemento	Nodo Inicial	Nodo Final
I	1	2
II	2	3
III	3	4
IV	4	5
V	1	6
VI	2	6
VII	2	7
VIII	3	7
IX	4	7
X	4	8
XI	5	8
XII	6	7
XIII	7	8

4.3 ESCENARIOS DE DAÑO

En el presente trabajo el daño en el elemento i es considerado como una disminución del módulo de elasticidad del material de ese elemento, y representado por el coeficiente de reducción β_i . Ese coeficiente asume el valor de 0 cuando el elemento no se encuentra damnificado, y un valor de 1 indicaría la pérdida total del elemento. La matriz de rigidez damnificada del elemento i es calculada multiplicando la matriz de rigidez original por el factor $(1-\beta)$.

Inclusión de ruido en mediciones. En el presente trabajo la presencia de ruido en las mediciones fue modelada mediante la introducción de pequeñas perturbaciones en frecuencias naturales y formas modales, de la siguiente forma:

$$f_{nr} = f_n \times (1 + Rand(-1,1) * Ruido_f) \quad (4.1)$$

$$\Phi_{nr} = \Phi_n \times (1 + Rand(-1,1) * Ruido_\phi) \quad (4.2)$$

Donde f_n y Φ_n corresponden a las frecuencias naturales y formas modales originales respectivamente, f_{nr} y Φ_{nr} son los valores de los mismos parámetros ahora contaminados con ruido. $Ruido_f$ y $Ruido_\phi$ son los porcentajes de ruido introducidos. Farrar y Cone (APUD, CHEN y NAGARAJAIAH, 2007) comentan que usualmente las frecuencias naturales y formas modales son contaminadas por ruido con un error promedio de $\pm 0.15\%$ e $\pm 3\%$, respectivamente.

Para el análisis de la influencia del ruido en los resultados son utilizadas las combinaciones basadas en los valores promedios mencionados anteriormente y el grado de dificultad del problema. Estos valores de ruido tanto para la viga, como para la armadura son presentados en la Tabla 3. La configuración del ruido fue considerada para representar las condiciones experimentales reales.

Tabla 3. Configuraciones de ruido que serán estudiadas (a) Viga y (b) Armadura

(a)		
Caso de Ruido	Ruido en frecuencia %	Ruido en Modos %
R1 _v	0	0
R2 _v	0.15	1
R3 _v	0.15	3
R4 _v	1	3
R5 _v	2	5
(b)		
Caso de Ruido	Ruido en la frecuencia %	Ruido en Modos %
R1 _a	0	0
R2 _a	0.1	1
R3 _a	0.15	3
R4 _a	0.2	5

Casos de Daño

Diversos escenarios de daño para ambas estructuras analizadas son considerados para la validación del desempeño de las metodologías estudiadas, como es presentado en la Tabla 4.

Tabla 4. Escenarios de daño estudiados para: (a) Viga y (b) Armadura

Número de Modos	Daño Simple	Daño Múltiple 2 Elementos	Daño Múltiple 4 Elementos
4	✓	✓	✓
6	✓	✓	✓
8	✓	✓	✓
10	✓	✓	✓

(a)

Número de Modos	Daño Simple	Daño Múltiple 2 Elementos
4	✓	✓
5	✓	✓
6	✓	✓
7	✓	✓
8	✓	✓

(b)

4.4 IMPLEMENTACION DE LA METODOLOGIA

4.4.1 Determinación de las propiedades dinámicas de las estructuras analizadas. Gracias a herramientas informáticas que están fundamentadas en los métodos de elementos finitos para el análisis estructural, hoy en día se pueden calcular fácilmente las propiedades dinámicas de las estructuras, con tan sólo definir sus propiedades de configuración, tales como las propiedades geométricas, materiales, restricciones entre otros.

La herramienta escogida para calcular las frecuencias y formas modales fue el software de libre distribución OPENSEES que al descargarse contiene un archivo ejecutable para instalar Tcl/Tk y otro ejecutable para compilar los modelos creados.

OPENSEES ofrece innumerables ventajas para la resolución del problema. La primera ventaja es que podemos obtener la respuesta dinámica de la estructura a partir de la codificación de las características que describan a la estructura en el lenguaje de programación TCL [13], el cual es fácil de usar, especialmente por su similitud con lenguajes tales como C, C++ y Java, y debido a que la interface para escribir el código es simplemente bloc de notas o wordpad, guardando el archivo con la extensión.tcl.

Los archivos codificados de la estructura deben contener principalmente: la distribución de los nodos, materiales, propiedades de las secciones transversales, definición de las características de los elementos, restricciones, y un archivo base que concatene los anteriormente mencionados, para poder ser ejecutados con OPENSEES.exe.

Otra de las ventajas es que éste software es de fácil manejo y ofrece una variedad de ayudas para la realización de modelos estructurales, las cuales se encuentran

disponibles en su página web, previamente referenciada. Una de las ayudas encontradas en este sitio es una herramienta denominada OPENSEES NAVIGATOR, la cual nos permite por medio de una interfaz gráfica (similar a un software de tipo CAD), modelar la estructura y obtener la codificación de los archivos (.TCL) necesarios para obtener la respuesta dinámica de la estructura (eigenvalores por medio de los cuales se encuentran las frecuencias naturales y eigenvectores para hallar las formas modales).

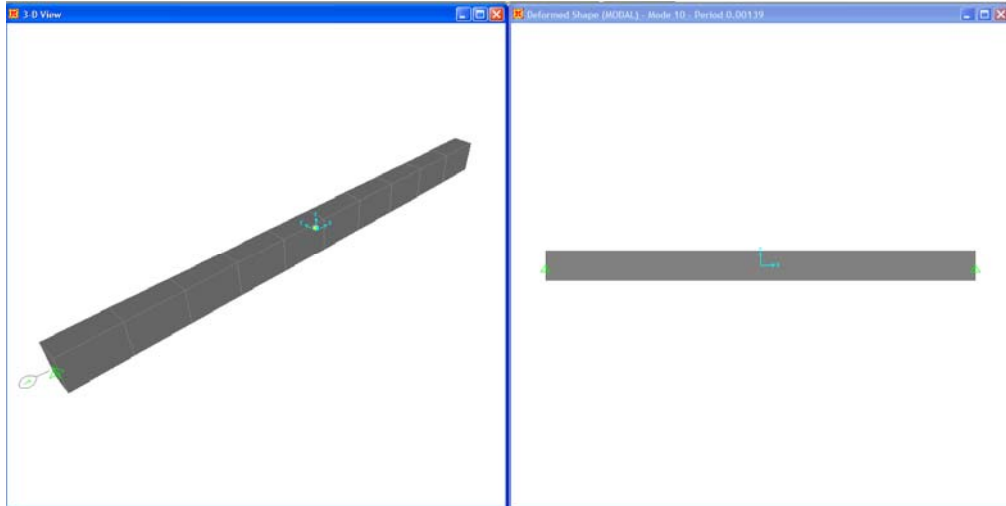
Para ilustrar un poco todo lo mencionado anteriormente se puede seguir un ejemplo básico de la utilización de OPENSEES en el sitio web: <http://soneromijaragual.netau.net/Traduccion.htm>

Con el fin de comprobar la fiabilidad de los valores obtenidos con OPENSEES, utilizamos una herramienta estructural de gran reconocimiento conocida como SAP2000 versión, (versión educacional 7.4. disponible en el CENTIC).

No se piensa detallar a fondo la utilización de este último programa mencionado, pero se piensa mostrar los resultados obtenidos para realizar la comparación deseada.

Modelamiento de las estructuras en SAP 2000 (Figuras 5 y 8):

Figura 8. Modelo viga usando SAP2000



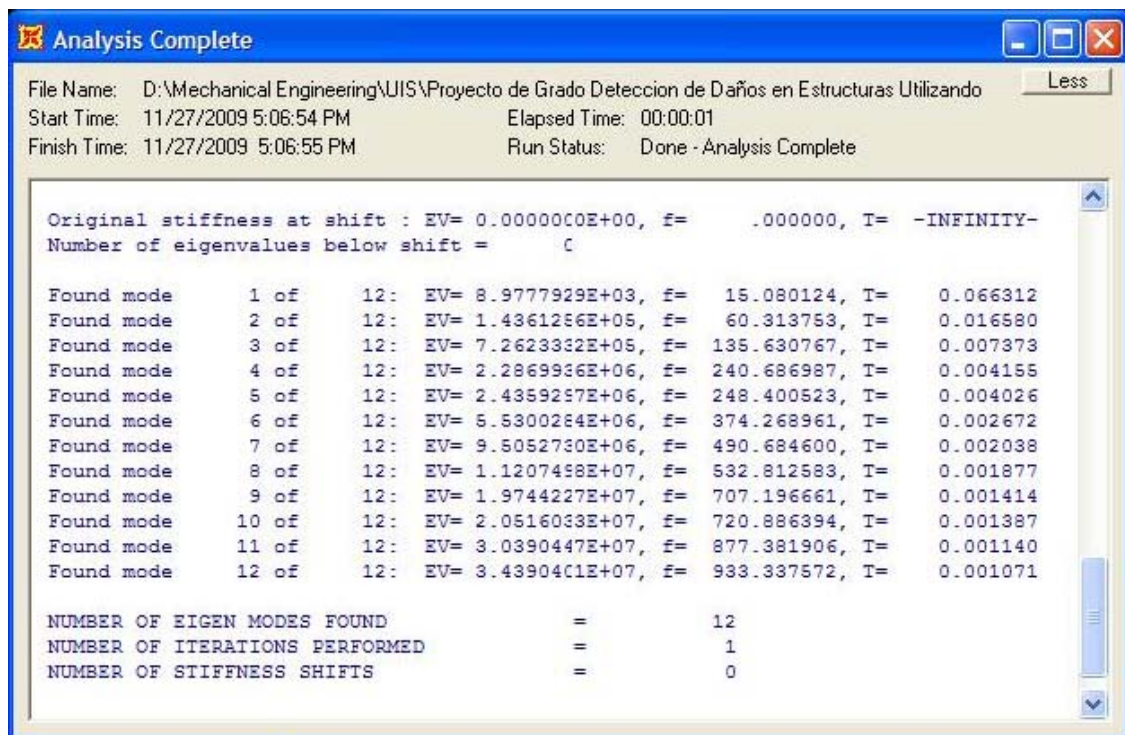
Los resultados de frecuencias y periodos obtenidos con OPENSEES (Tabla 5)

Tabla 5. Resultados de frecuencias y periodos OPENSEES

Eigenvalores	Frecuencias	Periodos
8.98E+03	15.0816789	0.066306
1.44E+05	60.3199754	0.016578
7.26E+05	135.644743	0.007372
2.29E+06	240.711789	0.004154
2.44E+06	248.426133	0.004025
5.53E+06	374.307524	0.002672
9.51E+06	490.735188	0.002038
1.12E+07	532.86753	0.001877
1.97E+07	707.269597	0.001414
2.05E+07	720.960649	0.001387

son muy cercanos a los valores hallados con el análisis de SAP 2000 (figura 4.5) por lo que a partir de este momento se confía totalmente en los resultados obtenidos con OPENSEES y se concluye la primera parte del procedimiento general, en la cual se determina la respuesta dinámica de la estructura sin daño .

Figura 9. Valores de periodos y frecuencias hallado por SAP2000 para la viga



De la misma forma, SAP 2000 nos permite apreciar las diversas formas modales (Figuras 10 y 12):

Figura 10. Modos de Vibracion Viga (a) 1 modo, (b) 2 modos, (c) 3 modos, (d) 4 modos, (e) 5 modos, (f) 6 modos, (g) 7 modos, (h) 8 modos, (i) 9 modos, (j) 10 modos

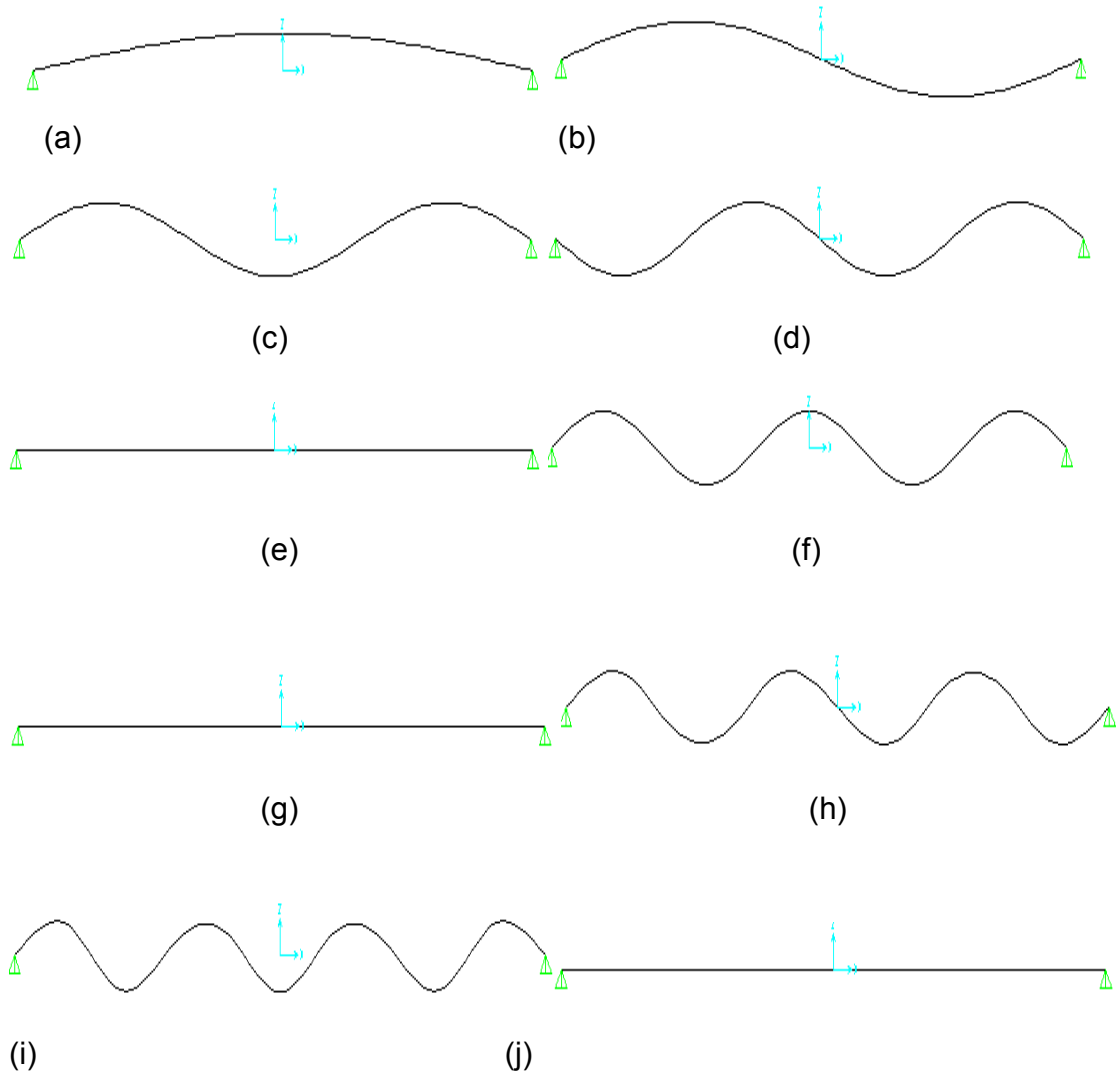


Figura 11. Modelo armadura usando SAP2000

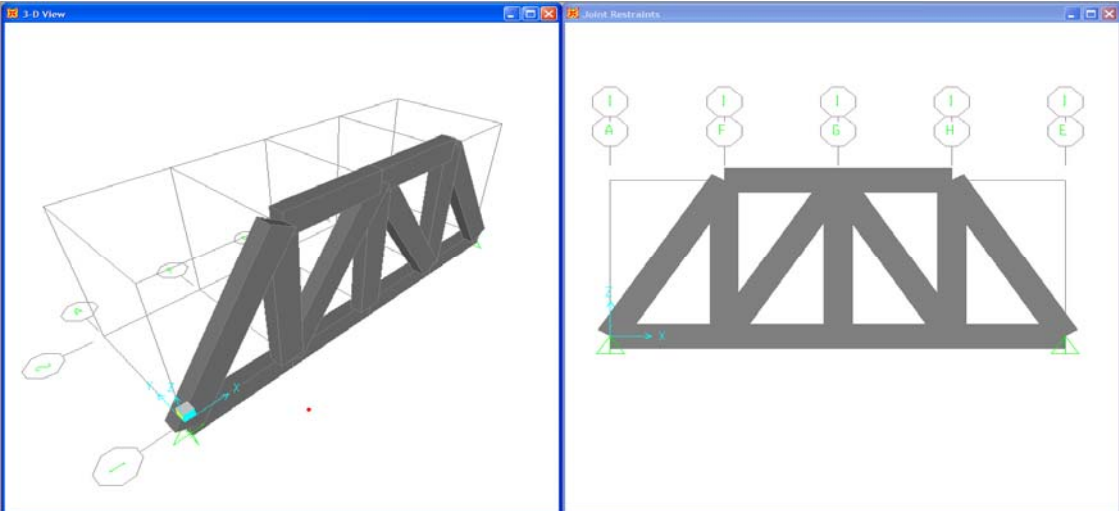
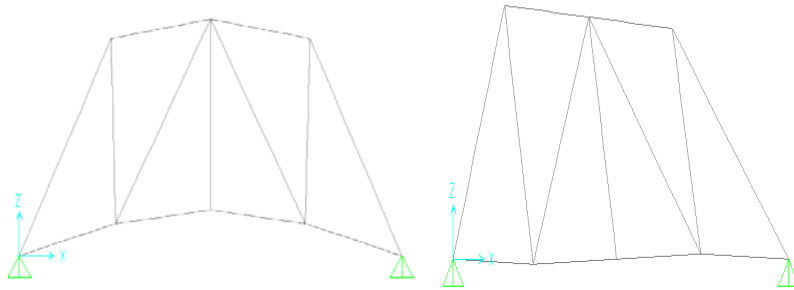
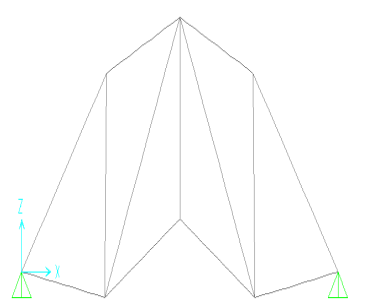
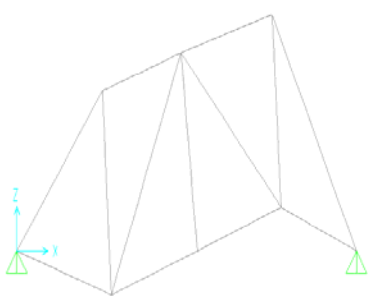


Figura 12. Modos de Vibracion Armadura (a) 1 modo, (b) 2 modos, (c) 3 modos, (d) 4 modos, (e) 5 modos, (f) 6 modos, (g) 7 modos, (h) 8 modos



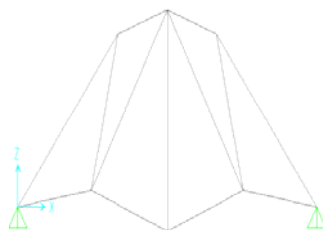
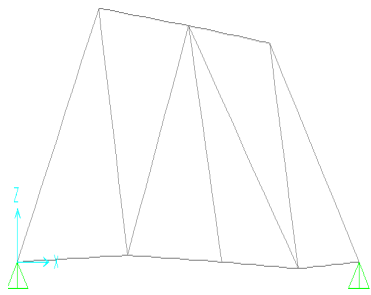
(a)

(b)



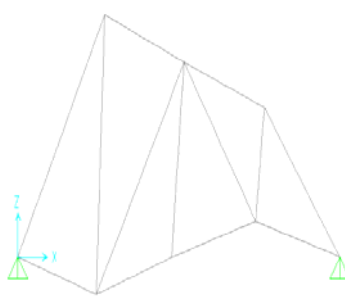
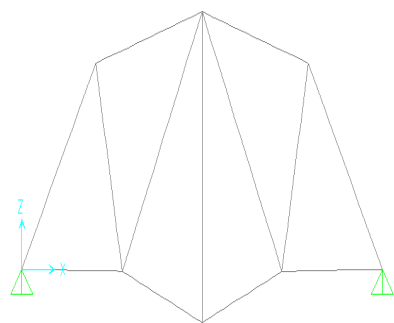
(c)

(d)



(e)

(f)



(g)

(h)

Para la armadura el procedimiento de utilización de OPENSEES para hallar los valores propios, es muy similar al de la viga, teniendo en cuenta ciertas diferencias, especialmente por la configuración. El proceso de verificación con SAP2000 es muy similar al de la viga también.

La razón de mayor importancia de la implementación de OPENSEES, es que se puede concatenar fácilmente con la interfaz de programación de MATLAB (MATLAB 7.0.4 versión comercial fue el software escogido para el desarrollo del presente trabajo, el cual se encuentra disponible en el CENTIC), ventaja que no brindan muchos programas comúnmente usados (entre estos SAP2000); de hecho los archivos obtenidos con el software estructural gratuito representan las variables de entrada para los algoritmos de MATLAB, encargados de introducir el daño a la estructura sana y de comparar esta respuesta real con la propuesta por PSO toolbox a través de la función objetivo.

Una vez calculados los eigenvalores y eigenvectores se procede a utilizar un algoritmo de MATLAB (ver Anexo 1) encargado de agregar un daño aleatorio a la cantidad de elementos deseada generando un Coeficiente de Reducción β_i . (0-60% teoría de análisis modal funciona solo en de estado lineal) De la forma:

$$E_{D_i} = E_i(1 - \beta_i) \quad (4.3)$$

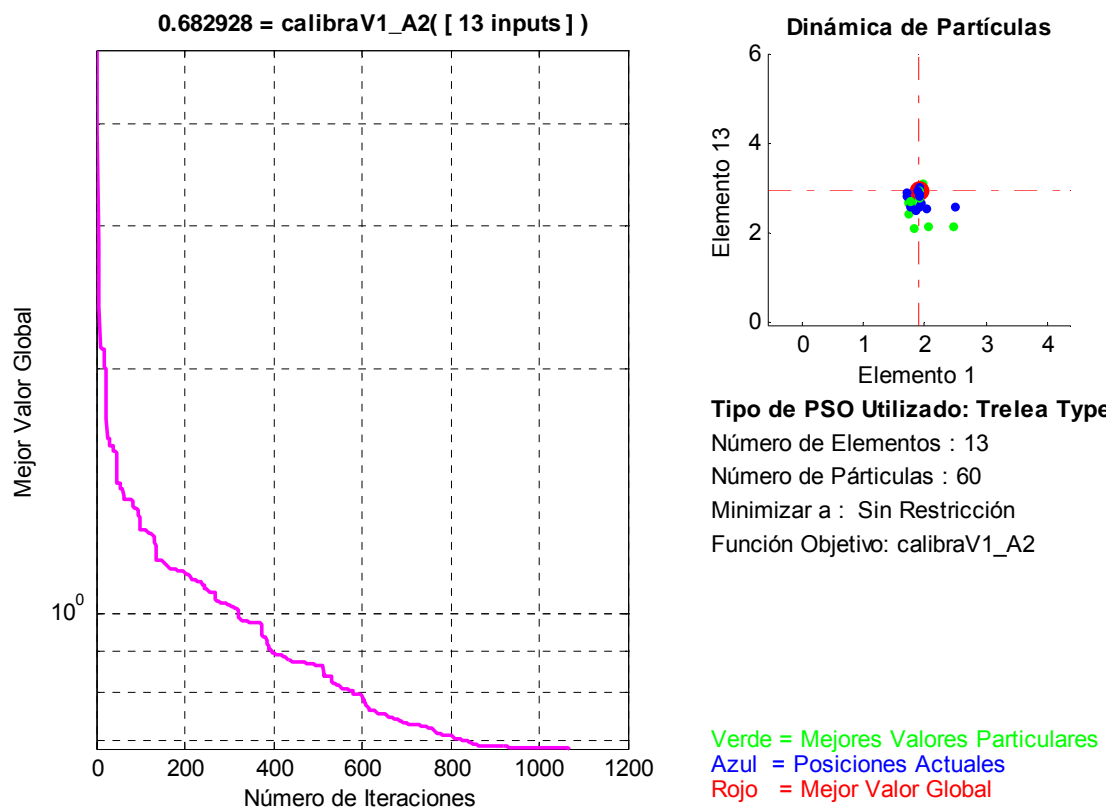
Donde E_{D_i} es el modulo de elasticidad correspondiente al elemento i después de ser introducido el daño, E_i es el modulo de elasticidad del elemento i sin daño.

En el desarrollo de la presente investigación se consideraron solo daños en el rango de [0-60%], debido a que un daño superior a estos niveles ya es un daño de colapso estructural inminente, que suele detectarse previamente por señales percibibles ante las inspecciones normalmente realizadas.

Una vez obtenidas las nuevas condiciones de la estructura, es decir un nuevo vector de módulos de elasticidad correspondientes al respectivo daño introducido, se procede a calcular la nueva respuesta dinámica de la estructura. Este paso es realizado a través del mismo algoritmo de MATLAB encargado de introducir el daño, simplemente cambiando el valor de los módulos de elasticidad en los archivos Tcl de la estructura sin afectar, creados en el paso anterior (mas exactamente en el archivo Tcl que contiene las características de las propiedades transversales); y se ejecuta nuevamente OPENSEES, generando y almacenando los nuevos eigenvalores y eigenvectores.

4.4.2 Ejecución de la metodología empleada para la toma de resultados. A partir de la respuesta dinámica del sistema damnificado, se desea obtener la localización y cuantificación del daño a través de la obtención de los módulos de elasticidad correspondientes a cada elemento. El siguiente paso del procedimiento general es la implementación de las funciones objetivo (Ec. 2.12 y 2.14), las cuales se encargan de comparar los valores de frecuencias y formas modales reales, con los valores arrojados por el algoritmo evolutivo; además en este archivo se especifica la cantidad de modos que se utilizan para esta comparación. Por medio de la utilización de la herramienta de PSO de MATLAB disponible, se obtienen los valores analíticos de la respuesta dinámica a ser tenidos en cuenta por la función objetivo que se desea minimizar; también se deben introducir los parámetros de PSO necesarios para condicionar el funcionamiento del algoritmo evolutivo (Ver Capítulo 3 para más detalles de la utilización de PSO en MATLAB). Una vez se ejecute el archivo principal de PSO una interface grafica muestra la dinámica de las partículas y el error global alcanzado en la función objetivo vs el número de épocas (ver figura 4.9.); al final obtenemos los valores de los módulos de elasticidad alcanzados junto con el mínimo error de la función objetivo.

Figura 13. Interface grafica herramienta PSO de Matlab



Para simular el daño múltiple en las estructuras simplemente en el algoritmo de inclusión de esta reducción del modulo de elasticidad, se especifican la cantidad de elementos dañados deseados. La cantidad de modos de comparación se especifican en el archivo de la función objetivo y por ultimo para la inclusión de ruido en las mediciones, simplemente se utiliza un m-file anexo que introduzca un porcentaje de error a la respuesta dinámica sin ruido.

5. ANÁLISIS DE RESULTADOS

Debido al carácter un tanto subjetivo envuelto en la determinación de lo que representa para la metodología un elemento probablemente damnificado, se estableció un criterio para la definición de esos elementos en los diferentes casos de daño estudiados:

Los elementos que presentan un daño significativo en comparación con los otros elementos en todas las condiciones de ruido, se consideran como elementos con daño.

A continuación se muestran los resultados obtenidos con la metodología implementada para la detección de daños para diferentes: función objetivo, tipo de estructura, escenario de daño (daño simple y daño múltiple), porcentaje de ruido en las mediciones, cantidad de información utilizada (número de modos)

5.1 IDENTIFICACIÓN DE ELEMENTOS DAÑADOS

5.1.1 Función Objetivo "CALIBRAV1" (ec. 2.14)

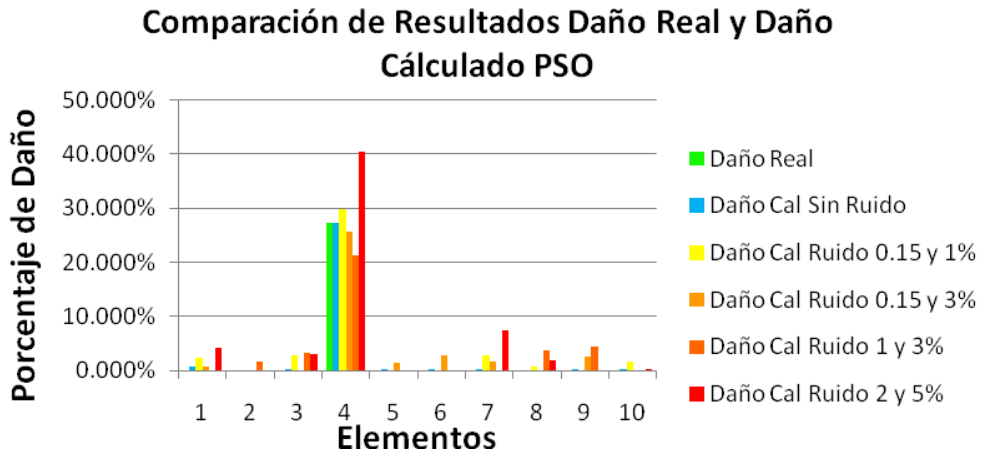
Viga

a. Daño Simple (Un Elemento)

- **4 Modos de Vibración:**

Elemento Dañado: 4 en 27,187%

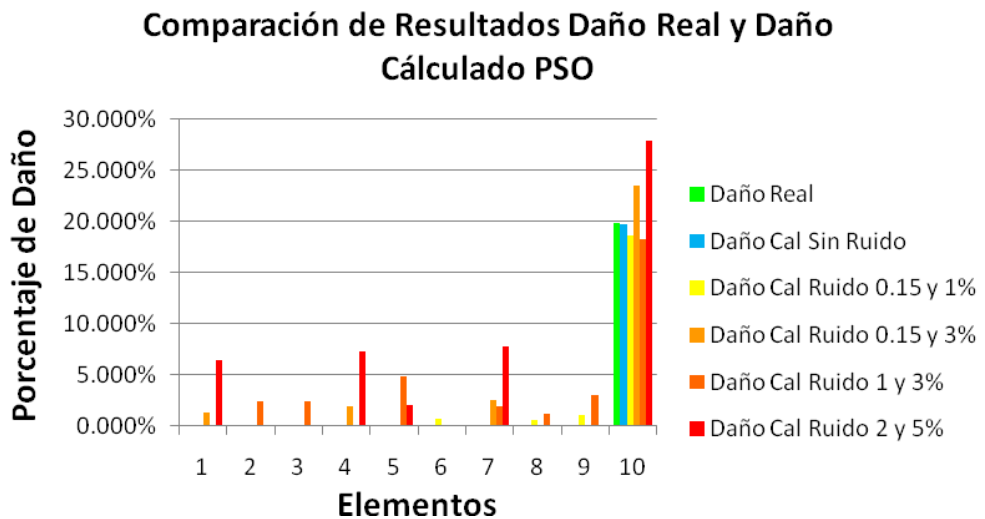
Figura 14. Comparación del Daño Real y el Daño Calculado en la viga con diferentes cantidades de ruido: Función objetivo CalibraV1, daño simple, 4 modos de vibración



- **6 Modos de Vibración:**

Elemento Dañado: 10 en 19,815%

Figura 15. Comparación del Daño Real y el Daño Calculado en la viga con diferentes cantidades de ruido: Función objetivo CalibraV1, daño simple, 6 modos de vibración



- **8 Modos de Vibración:**

Elemento Dañado: 2 en 43,488%

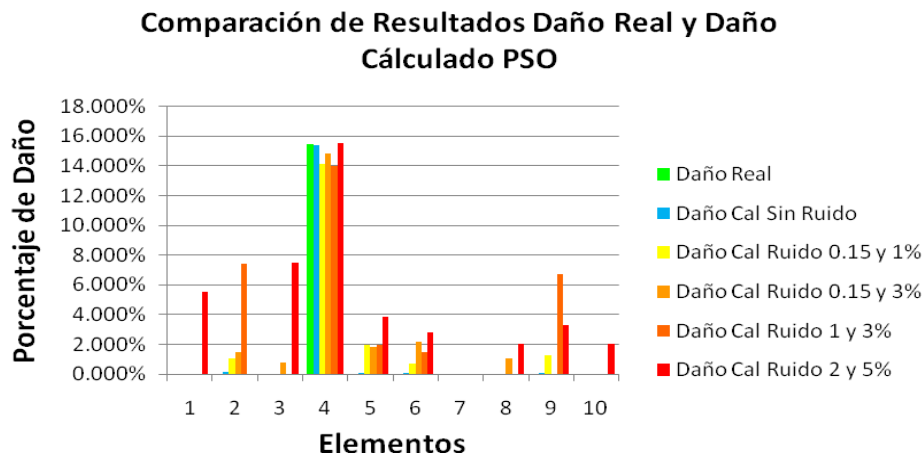
Figura 16. Comparación del Daño Real y el Daño Calculado en la viga con diferentes cantidades de ruido: Función objetivo CalibraV1, daño simple, 8 modos de vibración



- **10 Modos de Vibración:**

Elemento Dañado: 4 en 15,433%

Figura 17. Comparación del Daño Real y el Daño Calculado en la viga con diferentes cantidades de ruido: Función objetivo CalibraV1, daño simple, 10 modos de vibración

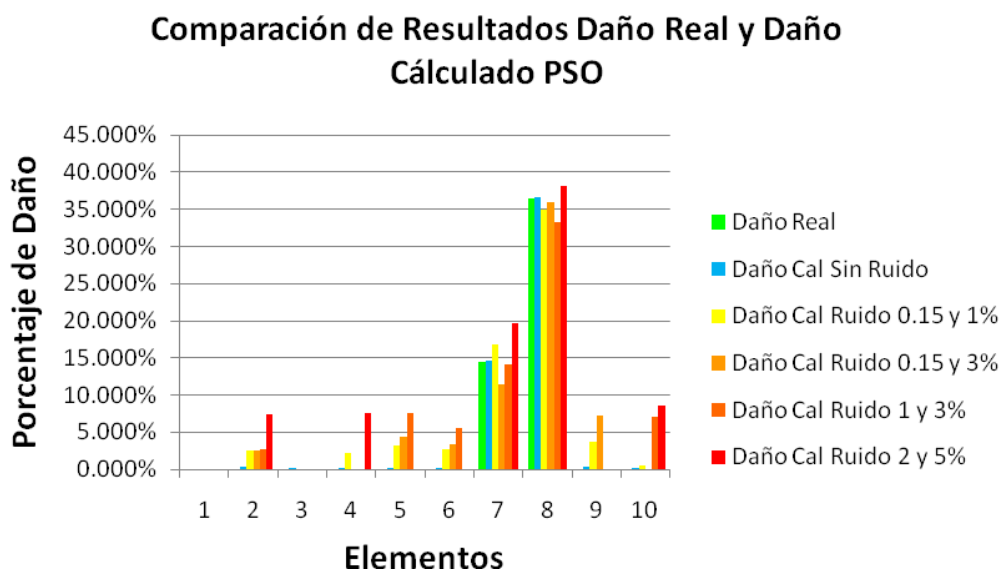


b. Daño Múltiple (Dos Elementos)

- **4 Modos de Vibración:**

Elementos Dañados: 7 en 14,536% y 8 en 36,466%

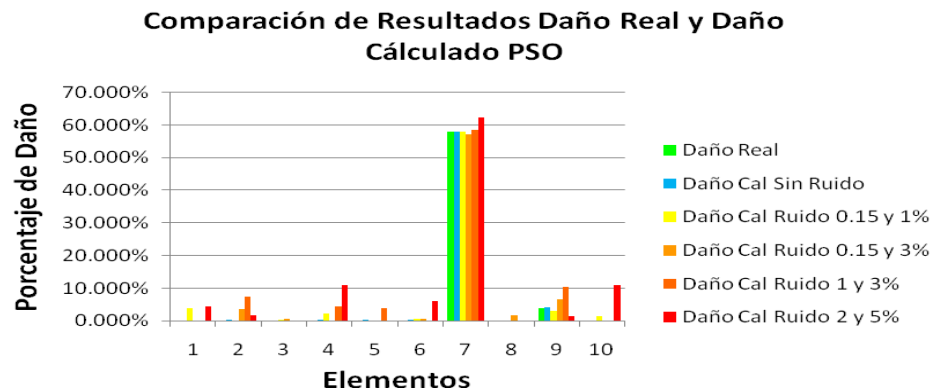
Figura 18. Comparación del Daño Real y el Daño Calculado en la viga con diferentes cantidades de ruido: Función objetivo CalibraV1, daño múltiple 2 elementos, 4 modos de vibración



- **6 Modos de Vibración:**

Elementos Dañados: 7 en 57,857% y 9 en 3,960%

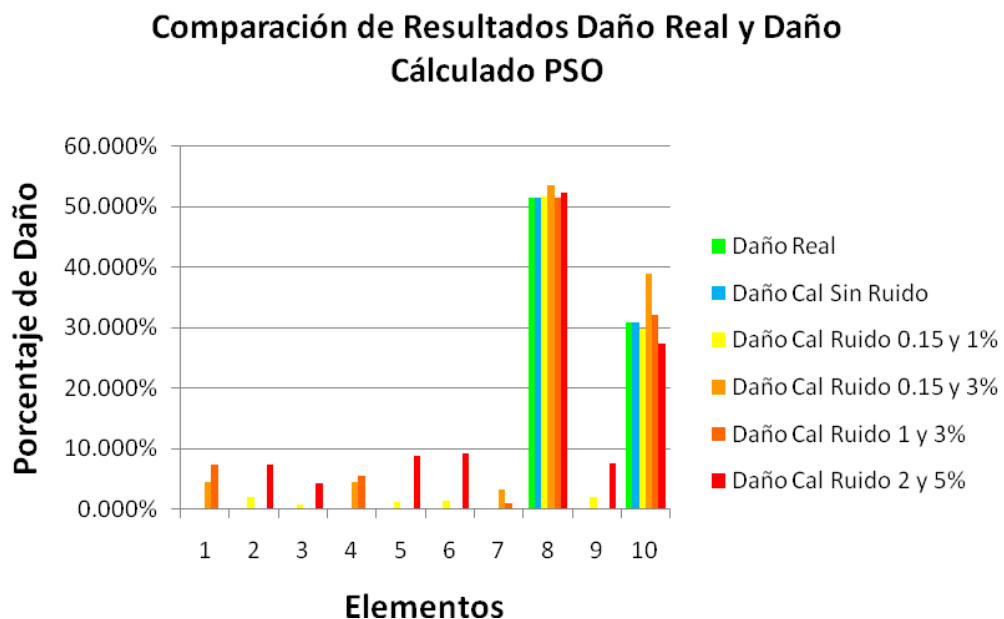
Figura 19. Comparación del Daño Real y el Daño Calculado en la viga con diferentes cantidades de ruido: Función objetivo CalibraV1, daño múltiple 2 elementos, 6 modos de vibración



• **8 Modos de Vibración:**

Elementos Dañados: 8 en 51,487% y 10 en 30,877%

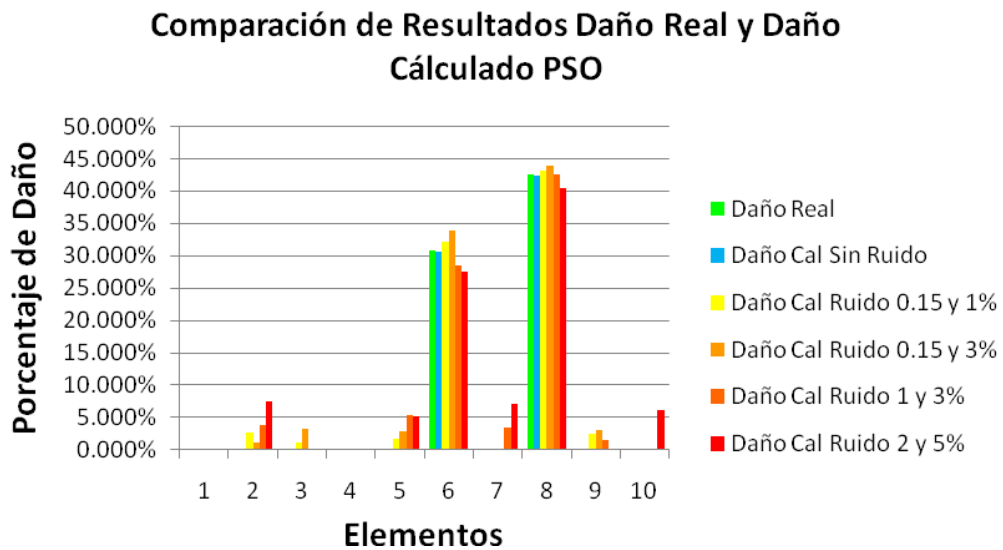
Figura 20. Comparación del Daño Real y el Daño Calculado en la viga con diferentes cantidades de ruido: Función objetivo CalibraV1, daño múltiple 2 elementos, 8 modos de vibración



- **10 Modos de Vibración:**

Elementos Dañados: 6 en 30,822% y 8 en 42,496%

Figura 21. Comparación del Daño Real y el Daño Calculado en la viga con diferentes cantidades de ruido: Función objetivo CalibraV1, daño múltiple 2 elementos, 10 modos de vibración

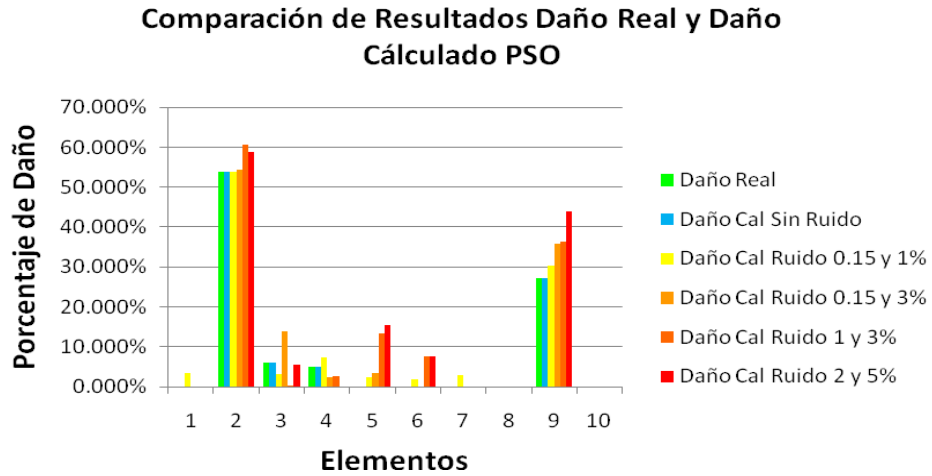


c. Daño Múltiple (Cuatro Elementos)

- **4 Modos de Vibración:**

Elementos Dañados: 2 en 53,846%, 3 en 6,134%, 4 en 5,091% y 9 en 27,252%

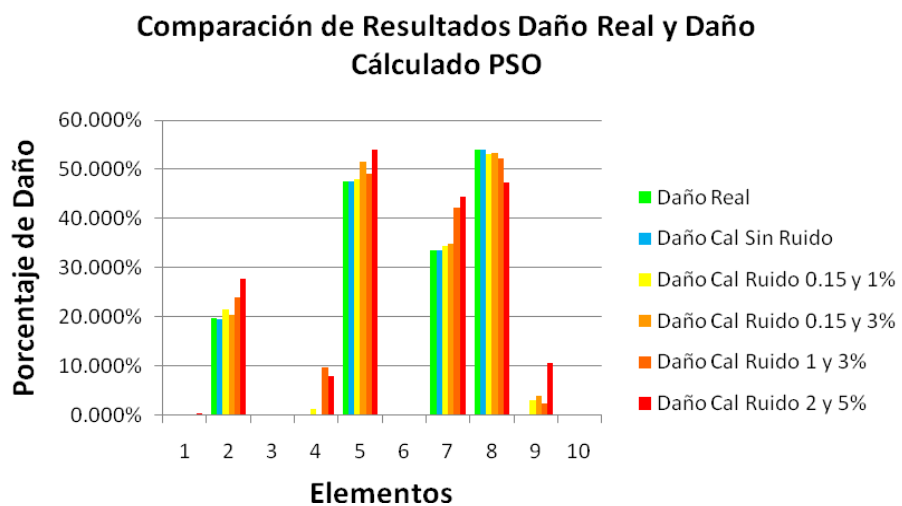
Figura 22. Comparación del Daño Real y el Daño Calculado en la viga con diferentes cantidades de ruido: Función objetivo CalibraV1, daño múltiple 4 elementos, 4 modos de vibración



• **6 Modos de Vibración:**

Elementos Dañados: 2 en 19,637%, 5 en 47,503%, 7 en 33,478% y 8 en 54,097%

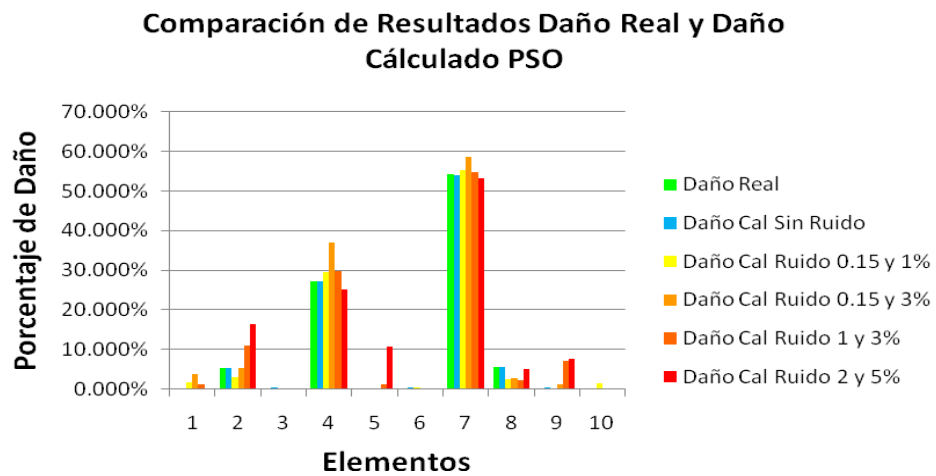
Figura 23. Comparación del Daño Real y el Daño Calculado en la viga con diferentes cantidades de ruido: Función objetivo CalibraV1, daño múltiple 4 elementos, 6 modos de vibración



- **8 Modos de Vibración:**

Elementos Dañados: 2 en 5,197%, 4 en 27,187%, 7 en 54,148% y 8 en 5,652%

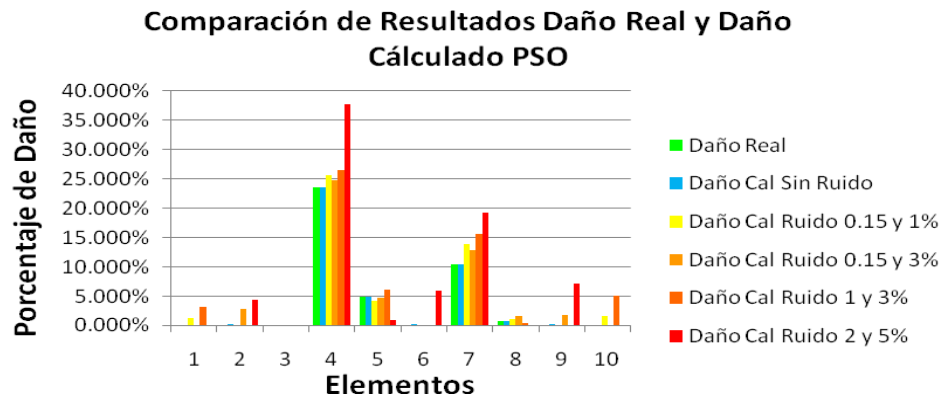
Figura 24. Comparación del Daño Real y el Daño Calculado en la viga con diferentes cantidades de ruido: Función objetivo CalibraV1, daño múltiple 4 elementos, 8 modos de vibración



- **10 Modos de Vibración:**

Elementos Dañados: 4 en 23,578%, 5 en 4,903%, 7 en 10,375%, 8 en 0,799%

Figura 25. Comparación del Daño Real y el Daño Calculado en la viga con diferentes cantidades de ruido: Función objetivo CalibraV1, daño múltiple 4 elementos, 10 modos de vibración



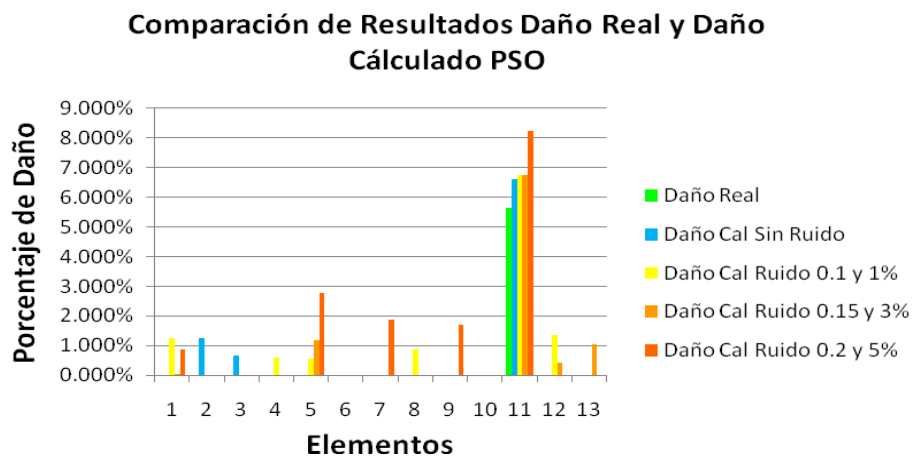
Armadura

a. Daño Simple (Un Elemento)

- **4 Modos de Vibración:**

Elemento Dañado: 11 en 5,652%

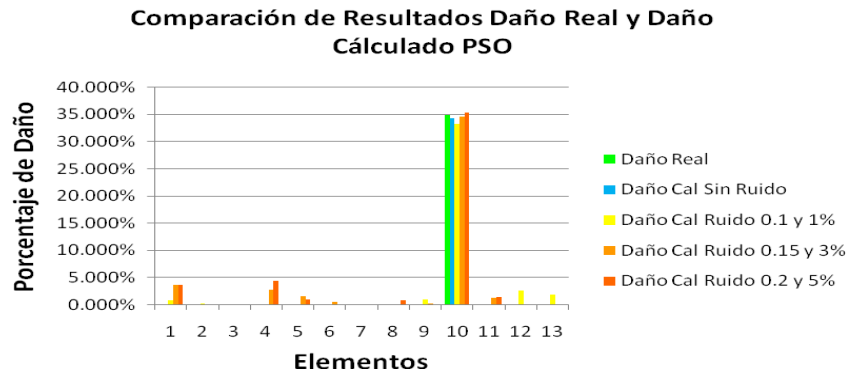
Figura 26. Comparación del Daño Real y el Daño Calculado en la armadura con diferentes cantidades de ruido: Función objetivo CalibraV1, daño simple, 4 modos de vibración



- **5 Modos de Vibración:**

Elemento Dañado: 10 en 34,927%

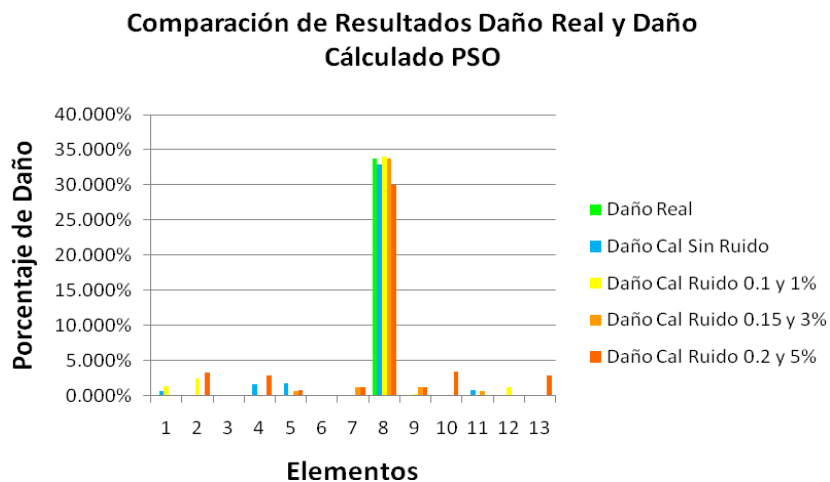
Figura 27. Comparación del Daño Real y el Daño Calculado en la armadura con diferentes cantidades de ruido: Función objetivo CalibraV1, daño simple, 5 modos de vibración



- **6 Modos de Vibración:**

Elemento Dañado: 8 en 33,721%

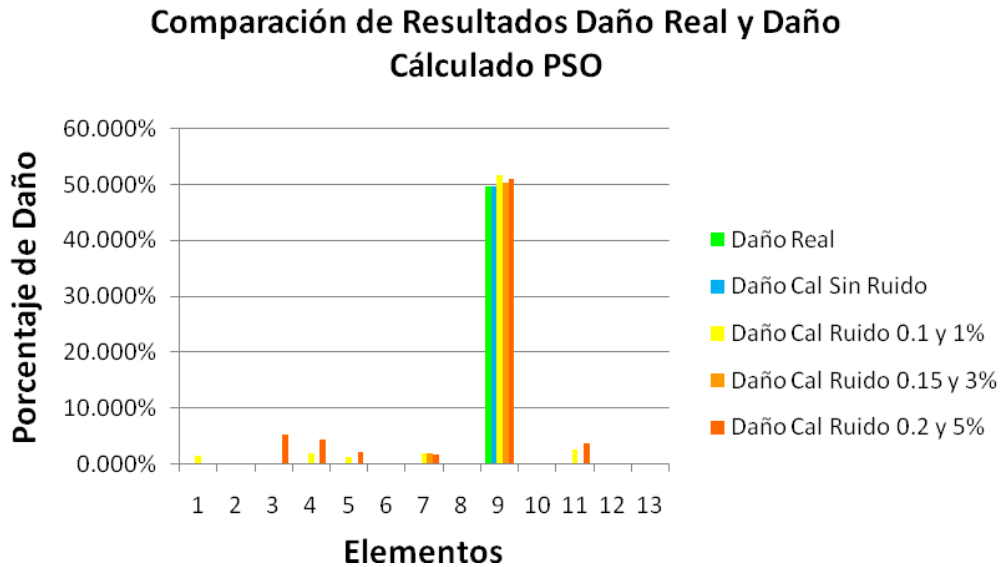
Figura 28. Comparación del Daño Real y el Daño Calculado en la armadura con diferentes cantidades de ruido: Función objetivo CalibraV1, daño simple, 6 modos de vibración



- **7 Modos de Vibración:**

Elemento Dañado: 9 en 49,729%

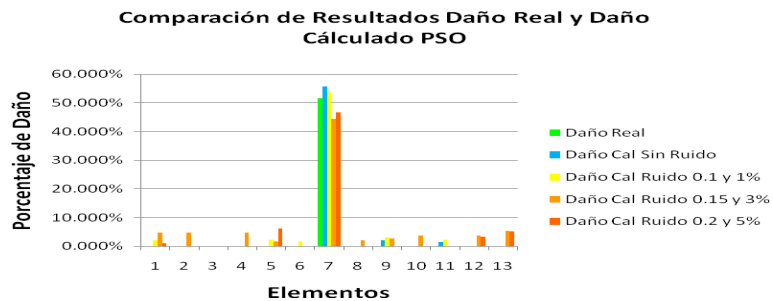
Figura 29. Comparación del Daño Real y el Daño Calculado en la armadura con diferentes cantidades de ruido: Función objetivo CalibraV1, daño simple, 7 modos de vibración



- **8 Modos de Vibración:**

Elemento Dañado: 7 en 51,596%

Figura 30. Comparación del Daño Real y el Daño Calculado en la armadura con diferentes cantidades de ruido: Función objetivo CalibraV1, daño simple, 8 modos de vibración

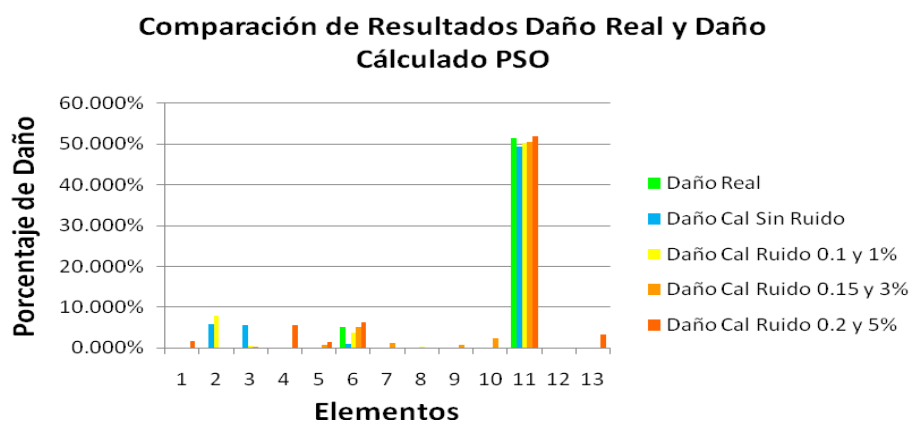


b. Daño Múltiple (Dos Elementos)

- **4 Modos de Vibración:**

Elementos Dañados: 6 en 5,056% y 11 en 51,487%

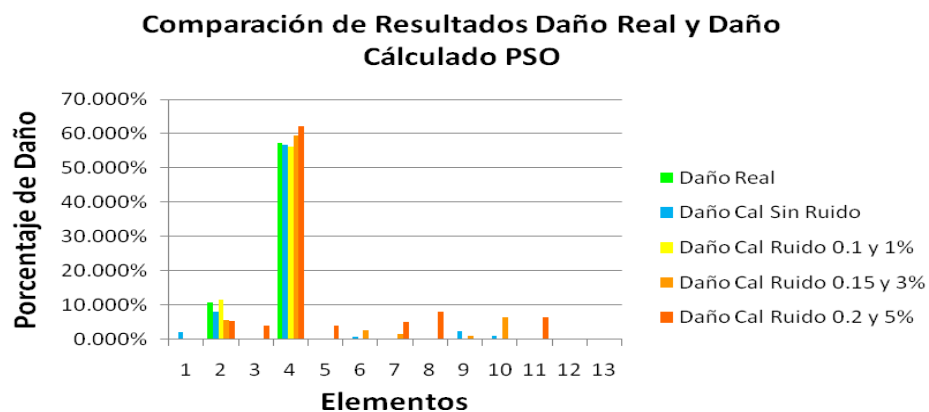
Figura 31. Comparación del Daño Real y el Daño Calculado en la armadura con diferentes cantidades de ruido: Función objetivo CalibraV1, daño múltiple 2 elementos, 4 modos de vibración



- **5 Modos de Vibración:**

Elementos Dañados: 2 en 10,593% y 4 en 57,230%

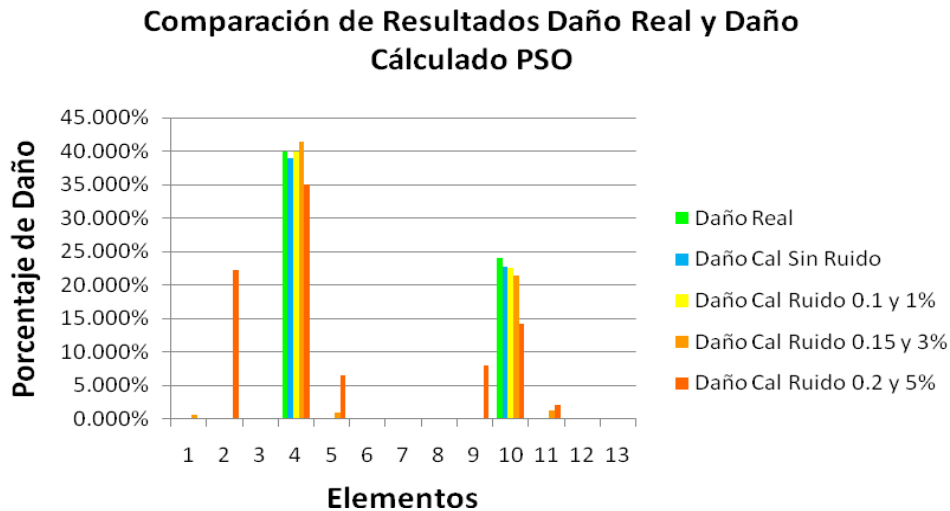
Figura 32. Comparación del Daño Real y el Daño Calculado en la armadura con diferentes cantidades de ruido: Función objetivo CalibraV1, daño múltiple 2 elementos, 5 modos de vibración



- **6 Modos de Vibración:**

Elementos Dañados: 4 en 39,987% y 10 en 24,100%

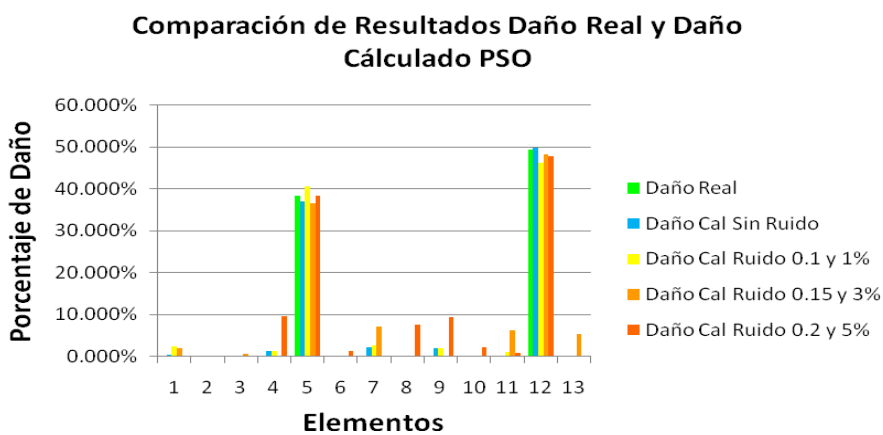
Figura 33. Comparación del Daño Real y el Daño Calculado en la armadura con diferentes cantidades de ruido: Función objetivo CalibraV1, daño múltiple 2 elementos, 6 modos de vibración



- **7 Modos de Vibración:**

Elementos Dañados: 5 en 38.299% y 12 en 49.376%

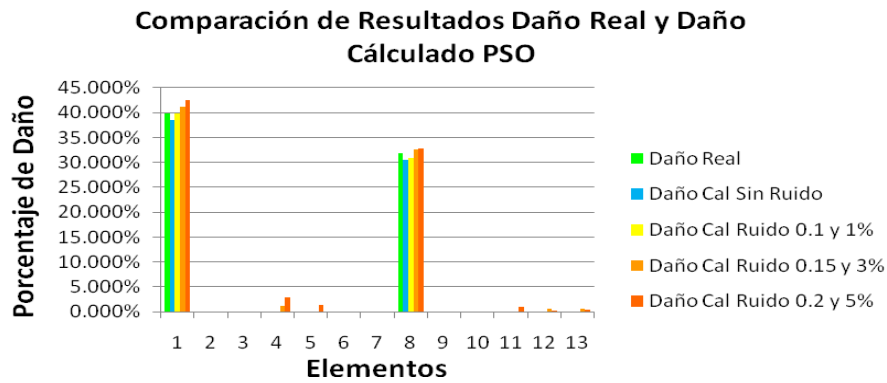
Figura 34. Comparación del Daño Real y el Daño Calculado en la armadura con diferentes cantidades de ruido: Función objetivo CalibraV1, daño múltiple 2 elementos, 7 modos de vibración



- **8 Modos de Vibración:**

Elementos Dañados: 1 en 39,773% y 8 en 31,837%

Figura 35. Comparación del Daño Real y el Daño Calculado en la armadura con diferentes cantidades de ruido: Función objetivo CalibraV1, daño múltiple 2 elementos, 8 modos de vibración



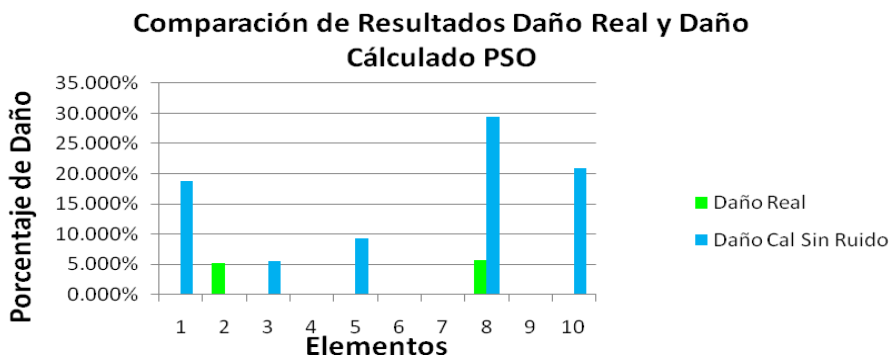
5.1.2. Función Objetivo CALIBRAV (ec. 2.12) Viga

a. Daño Múltiple (Dos Elementos)

- **6 Modos de Vibración:**

Elementos Dañados: 2 en 5.197% y 8 en 5.652%

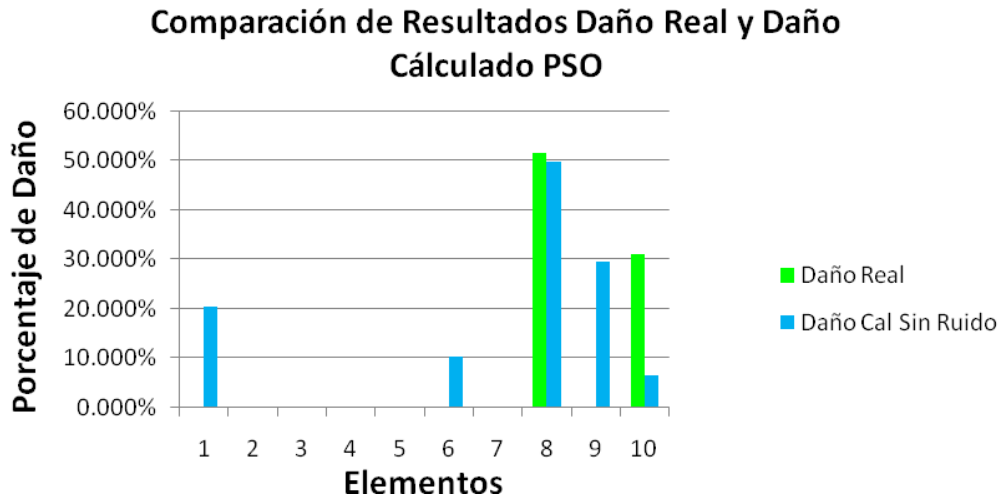
Figura 36. Comparación del Daño Real y el Daño Calculado en la viga sin ruido: Función objetivo CalibraV, daño múltiple 2 elementos, 6 modos de vibración



- **8 Modos de Vibración:**

Elementos Dañados: 8 en 51.487% y 10 en 30.877%

Figura 37. Comparación del Daño Real y el Daño Calculado en la viga sin ruido: Función objetivo CalibraV, daño múltiple 2 elementos, 8 modos de vibración



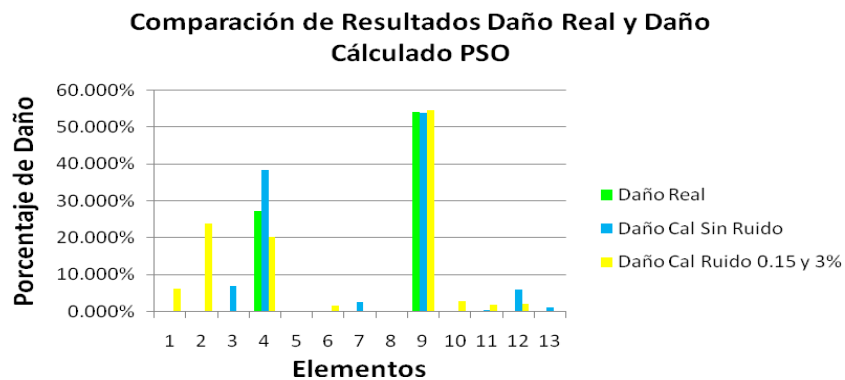
Armadura

a. Daño Múltiple (Dos Elementos))

- **5 Modos de Vibración:**

Elementos Dañados: 4 en 27.187% y 9 en 54.148%

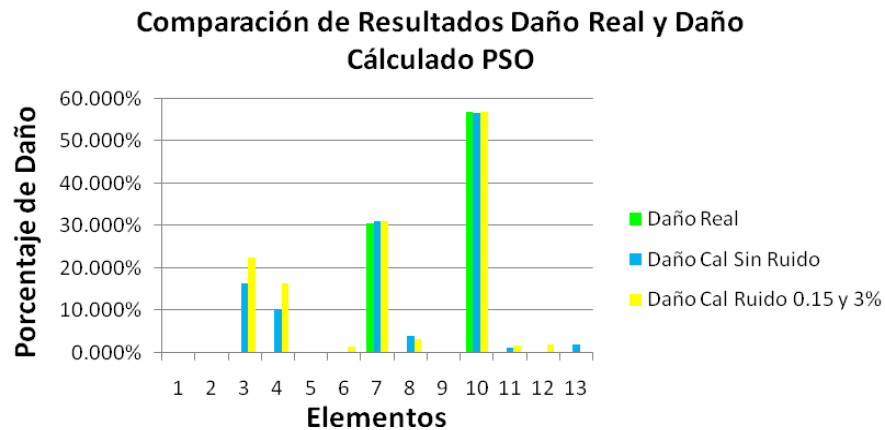
Figura 38. Comparación del Daño Real y el Daño Calculado en la armadura sin ruido y con nivel de ruido promedio: Función objetivo CalibraV, daño múltiple 2 elementos, 5 modos de vibración



- **6 Modos de Vibración:**

Elementos Dañados: 7 en 30.393 y 10 en 56.724

Figura 39. Comparación del Daño Real y el Daño Calculado en la armadura sin ruido y con nivel de ruido promedio: Función objetivo CalibraV, daño múltiple 2 elementos, 6 modos de vibración



Tras varias pruebas realizadas, los resultados obtenidos demuestran que la función objetivo “CALIBRAV1” es adecuada para la metodología de detección de daños como un proceso de optimización bajo todas las condiciones analizadas, mientras que la función objetivo “CALIBRAV” solo presentó un acercamiento confiable a los valores reales de daño cuando se usaba para analizar la estructura de tipo viga en el escenario de daño simple y cuando ésta era discretizada en menos de 7 elementos, lo cual no era adecuado para el contexto del problema, lo cual se aprecia en las figuras 5.23 y 5.24, donde a pesar de la ausencia del ruido en los valores analizados, se llegaban a resultados lejanos de los valores reales, por lo cual se optó por utilizar otra función objetivo que fuera confiable. Ambas funciones objetivo al ser aplicadas a la estructura de tipo armadura en un principio no proveían resultados cercanos a los valores reales del daño, por lo que tras analizar posibles soluciones, se encontró que al normalizar las formas modales con respecto a la masa se pudieron obtener resultados más precisos; de nuevo la función objetivo “CALIBRAV”, no alcanzaba los errores esperados, aunque debido

a la normalización de los modos con respecto a la masa, su desempeño fue notablemente mejor en esta estructura que en el caso de la viga. Por las razones mencionadas anteriormente, esta función objetivo fue desechada como posible herramienta para la solución del problema, por lo tanto, la mayoría de los resultados mostrados solo corresponden a la función objetivo "CALIBRAV1", que tras todas las pruebas realizadas, demuestra ser bastante confiable para la detección de daños en casi cualquier condición de análisis estructural.

A partir de las figuras anteriores, se aprecia que la metodología escogida en el caso de la viga, es muy precisa para determinar los elementos afectados en ausencia de ruido, llegando a tener errores menores de $1E-03$ en estos casos, pudiéndose determinar claramente cuáles son los elementos damnificados. Ante la presencia de valores de ruido cercanos a los valores promedio (mencionados anteriormente como 0.15% en las frecuencias y 3% en las formas modales), la metodología mostro un desempeño satisfactorio, llegando a errores menores de $1E-01$, pero para el caso de presencia de ruido en niveles severos (1%, 2% para las frecuencias y 5% para las formas modales), se detectaron de igual forma los elementos dañados, siempre y cuando este daño fuera superior al 10%, ya que para daños menores de este porcentaje, la técnica empleada no siempre presento la sensibilidad para su detección. También ante escenarios de ruido severo, aunque generalmente se encontraban cuales eran los elementos afectados, su porcentaje de daño arrojado difería notablemente del valor real, además se destaca la presencia de elementos falsamente afectados.

Para el caso de la armadura, debido a su mayor complejidad, los errores alcanzados no sobrepasaron $1E-02$, evidenciando características similares a las de la viga con un detrimento de precisión ante el aumento de ruido en las mediciones. Para el caso de la armadura teniendo en cuenta la complejidad de su configuración y la dificultad para encontrar el ajuste de la función objetivo que obtuviera valores acertados, se usaron solo niveles de ruido promedio y alrededor

de estos. Lo mencionado anteriormente, se evidencia en las graficas correspondientes a la cercha, donde muy pocas veces se encontraron elementos falsamente afectados significativamente, por lo que posiblemente el método se podría extender para niveles de ruido mucho más severos.

Respecto a la cantidad de información utilizada (número de modos), en el caso de la viga se encontró que el método era satisfactoriamente confiable cuando se utilizaban mínimo tres formas modales para el análisis y que a medida que se usaban más modos de vibración (recordando que el número máximo posible para el análisis es igual al número de grados de libertad del sistema) mayor era la precisión de los resultados, aunque la precisión obtenida comparada con el tiempo de ejecución del programa no hacen conveniente el análisis de una cantidad excesiva de formas modales. En el caso de la armadura, se encontró que contrario a lo esperado no siempre el desempeño de la metodología era mejor entre mayor fuera la cantidad de modos analizados, es más, se encontró que como límite superior para la obtención de resultados confiables la cantidad de modos debía ser de ocho, y como límite inferior de cuatro.

Los resultados mostrados anteriormente, de la misma manera muestran que el desempeño del método de detección de daños no depende de la cantidad de elementos dañados para la función objetivo "CALIBRAV1", pero si afectaron el desempeño para la función objetivo "CALIBRAV".

5.2 ANÁLISIS DE LA CONVERGENCIA DE PSO

A continuación se muestran los errores alcanzados en las funciones objetivo utilizadas, para todas las condiciones de ruido y escenarios de daño analizados en las dos estructuras.

5.2.1. Función Objetivo CalibraV1 (ec. 2.14):

Viga

Tabla 5. Errores mínimos alcanzados en la función objetivo CalibraV1 ante diversos niveles de ruido: Viga, daño simple

VIGA DAÑO SIMPLE	Sin Ruido	Ruido de 0.15 y 1 %	Ruido de 0.15 y 3 %	Ruido de 1 y 3 %	Ruido de 2 y 5 %
4 Modos E.D.: 4 en 27,187%	0.000133387	0.00808258	0.0248206	0.0239804	0.0417866
6 Modos E.D.: 10 en 19,815%	8.02897e-005	0.0101144	0.0304157	0.0304732	0.0503176
8 Modos E.D.: 2 en 43,488%	3.83442e-005	0.0123252	0.0365929	0.0367915	0.0610587
10 Modos E.D.: 4 en 15,433%	0.000124469	0.014558	0.0435085	0.0447577	0.0727951

Tabla 6. Errores mínimos alcanzados en la función objetivo CalibraV1 ante diversos niveles de ruido: Viga, daño múltiple 2 elementos

VIGA DAÑO MÚLTIPLE 2 ELEMENTOS	Sin Ruido	Ruido de 0.15 y 1 %	Ruido de 0.15 y 3 %	Ruido de 1 y 3 %	Ruido de 2 y 5 %
4 Modos E.D.: 7 y 8 en 14,536% y 36,466%	7.29123e-005	0.0076819	0.0245828	0.0248269	0.0425676
6 Modos E.D.: 7 y 9 en 57,857% y 3,960%	8.44666e-005	0.0103452	0.00999849	0.0312465	0.0507563
8 Modos E.D.: 8 y 10 51,487% y 30,877%	0.000317139	0.0122736	0.0371076	0.0368724	0.0614777
10 Modos E.D.: 6 y 8 en 30,822% y 42,496%	0.000191155	0.014423	0.0432487	0.0431086	0.0718563

Tabla 7. Errores mínimos alcanzados en la función objetivo CalibraV1 ante diversos niveles de ruido: Viga, daño múltiple 4 elementos

VIGA DAÑO MÚLTIPLE 4 ELEMENTOS	Sin Ruido	Ruido de 0.15 y 1 %	Ruido de 0.15 y 3 %	Ruido de 1 y 3 %	Ruido de 2 y 5 %
4 Modos E.D.: 2, 3, 4 y 9 en 53,846%, 6,134%, 5,091% y 27,252%	7.8791e-006	0.00821996	0.0249196	0.0242442	0.0413876
6 Modos E.D.: 2, 5, 7 y 8 en 19,637%, 47,503%, 33,478% y 54,097%	1.99285e-005	0.0101108	0.0301505	0.0304551	0.0498022
8 Modos E.D.: 2, 4, 7 y 8 en 5,197%, 27,187%, 54,148% y 5,652%	0.000169774	0.00996932	0.0298752	0.0301808	0.0508236
10 Modos E.D.: 4, 5, 7 y 8 en 23,578%, 4,903%, 10,375% y 0,799%	0.000104782	0.0144499	0.0433932	0.0428927	0.0734292

5.2.1.2 Armadura

Tabla 8. Errores mínimos alcanzados en la función objetivo CalibraV1 ante diversos niveles de ruido: Armadura, daño múltiple simple

ARMADURA DAÑO SIMPLE	Sin Ruido	Ruido de 0.1 y 1 %	Ruido de 0.15 y 3 %	Ruido de 0.2 y 5 %
4 Modos E.D.: 11 en 5,652%	0.0239892	0.0673277	0.094427	0.128014
5 Modos E.D.: 10 en 34,927%	0.00624027	0.124646	0.113756	0.155577
6 Modos E.D.: 8 en 33,721%	0.058541	0.151884	0.111996	0.501729
7 Modos E.D.: 9 en 49,729%	5.44716e-005	0.141508	0.136054	0.304992
8 Modos E.D.: 7 en 51,596%	0.373076	1.29169	0.533865	0.625729

Tabla 9. Errores mínimos alcanzados en la función objetivo CalibraV1 ante diversos niveles de ruido: Armadura, daño múltiple 2 elementos

ARMADURA DAÑO MÚLTIPLE 2 ELEMENTOS	Sin Ruido	Ruido de 0.1 y 1 %	Ruido de 0.15 y 3 %	Ruido de 0.2 y 5 %
4 Modos E.D.: 6 y 11 en 5,056% y 51,487%	0.0131858	0.0296509	0.0733909	0.133561
5 Modos E.D.: 2 y 4 en 10,593% y 57,230%	0.0521911	0.0338805	0.111817	0.230467
6 Modos E.D.: 4 y 10 en 39,987% y 24,100%	0.00306883	0.0396093	0.116207	0.307679
7 Modos E.D.: 5 y 12 en 38.299% y 49.376%	0.0839197	0.682928	0.267491	0.434297
8 Modos E.D.: 1 y 8 en 39,773% y 31,837%	0.00631031	0.05459468	0.15718989	0.27002869

5.2.2. Función Objetivo CalibraV (ec. 2.12): Viga

Tabla 10. Errores mínimos alcanzados en la función objetivo CalibraV ante ausencia de ruido: Viga, daño múltiple 2 elementos

VIGA DAÑO MÚLTIPLE 2 ELEMENTOS	Sin Ruido
6 Modos E.D.: 2 y 8 en 5.197% y 5.652%	0.16868
8 Modos E.D.: 8 y 10 51.487% y 30,877%	0.327365

5.2.2.2. Armadura

Tabla 11. Errores mínimos alcanzados en la función objetivo CalibraV ante diversos niveles de ruido: Armadura, daño múltiple 2 elementos

ARMADURA DAÑO MÚLTIPLE 2 ELEMENTOS	Sin Ruido	Ruido de 0.15 y 3 %
5 Modos E.D.: 4 y 9 en 27.187% y 54.148%	0.040338	0.0213745
6 Modos E.D.: 10 y 13 en 30.393% y 56.724%	0.0300086	0.0475341

A partir de las tablas anteriores se muestra como la metodología llega a errores mínimos ante ausencia de ruido y máximos con el caso contrario con la mayor cantidad de ruido introducida. Se aprecia que para el caso de la viga todos los errores son menores debido a la mayor simplicidad de la configuración estructural y a que las formas modales en la armadura son mucho más complejas, por lo que se dificulta la convergencia del algoritmo, razón por la cual el programa no funciona apropiadamente con más de 8 modos de vibración para la armadura.

Tras probar los diversos parámetros de PSO, se encontró que la cantidad adecuada de partículas para obtener un error razonable teniendo en cuenta el tiempo de ejecución fue de 60 partículas. De la misma manera se encontraron los valores de los otros parámetros de entrada de PSO que proporcionaron los errores mínimos encontrados, tales como las velocidades de partículas y el tipo de PSO escogido el cual fue el tipo Trelea 2.

5.3 TIEMPO DE EJECUCIÓN DE PSO

A continuación se muestra la cantidad de épocas necesarias para la ejecución del programa de PSO (valor mostrado en la parte superior de cada casilla) y su respectivo tiempo de ejecución en minutos (valor mostrado en la parte inferior de cada casilla).

5.3.1 Función Objetivo CalibraV1 (ec. 2.14): Viga

Tabla 12. Errores mínimos alcanzados en la función objetivo CalibraV1 ante diversos niveles de ruido: Viga, daño simple

VIGA DAÑO SIMPLE	Sin Ruido	Ruido de 0.15 y 1 %	Ruido de 0.15 y 3 %	Ruido de 1 y 3 %	Ruido de 2 y 5 %
4 Modos E.D.: 4 en 27,187%	98 33	195 65	267 89	212 77	189 63
6 Modos E.D.: 10 en 19,815%	128 43	259 86	185 62	169 56	248 83
8 Modos E.D.: 2 en 43,488%	172 57	240 80	199 66	301 100	209 70
10 Modos E.D.: 4 en 15,433%	163 54	238 79	192 64	183 61	217 72

Tabla 13. Errores mínimos alcanzados en la función objetivo CalibraV1 ante diversos niveles de ruido: Viga, daño múltiple 2 elementos

VIGA DAÑO MÚLTIPLE 2 ELEMENTOS	Sin Ruido	Ruido de 0.15 y 1 %	Ruido de 0.15 y 3 %	Ruido de 1 y 3 %	Ruido de 2 y 5 %
4 Modos E.D.: 7 y 8 en 14,536% y 36,466%	161 54	197 66	219 73	184 61	214 71
6 Modos E.D.: 7 y 9 en 57,857% y 3,960%	139 46	222 74	180 60	233 78	239 80
8 Modos E.D.: 8 y 10 51,487% y 30,877%	156 52	223 74	257 86	220 73	241 80
10 Modos E.D.: 6 y 8 en 30,822% y 42,496%	218 73	202 67	211 70	160 53	388 129

Tabla 14. Errores mínimos alcanzados en la función objetivo CalibraV1 ante diversos niveles de ruido: Viga, daño múltiple 4 elementos

VIGA DAÑO MÚLTIPLE 4 ELEMENTOS	Sin Ruido	Ruido de 0.15 y 1 %	Ruido de 0.15 y 3 %	Ruido de 1 y 3 %	Ruido de 2 y 5 %
4 Modos E.D.: 2, 3, 4 y 9 en 53,846%, 6,134%, 5,091% y 27,252%	208 69	219 73	179 60	215 72	203 68
6 Modos E.D.: 2, 5, 7 y 8 en 19,637%, 47,503%, 33,478% y 54,097%	166 55	204 68	305 102	189 63	277 92
8 Modos E.D.: 2, 4, 7 y 8 en 5,197%, 27,187%, 54,148% y 5,652%	163 54	160 53	191 64	240 80	252 84
10 Modos E.D.: 4, 5, 7 y 8 en 23,578%, 4,903%, 10,375% y 0,799%	147 49	243 81	169 56	241 80	309 103

Armadura

Tabla 15. Errores mínimos alcanzados en la función objetivo CalibraV1 ante diversos niveles de ruido: Armadura, daño simple

ARMADURA DAÑO SIMPLE	Sin Ruido	Ruido de 0.1 y 1 %	Ruido de 0.15 y 3 %	Ruido de 0.2 y 5 %
4 Modos E.D.: 11 en 5,652%	515 172	380 127	509 170	592 197
5 Modos E.D.: 10 en 34,927%	581 194	491 164	440 147	639 213
6 Modos E.D.: 8 en 33,721%	408 136	612 204	338 113	531 177
7 Modos E.D.: 9 en 49,729%	502 167	1110 370	891 297	1506 502
8 Modos E.D.: 7 en 51,596%	480 160	890 297	1410 470	995 332

Tabla 16. Errores mínimos alcanzados en la función objetivo CalibraV1 ante diversos niveles de ruido: Armadura, daño múltiple 2 elementos

ARMADURA DAÑO MÚLTIPLE 2 ELEMENTOS	Sin Ruido	Ruido de 0.1 y 1 %	Ruido de 0.15 y 3 %	Ruido de 0.2 y 5 %
4 Modos E.D.: 6 y 11 en 5,056% y 51,487%	720 240	893 298	718 239	549 183
5 Modos E.D.: 2 y 4 en 10,593% y 57,230%	578 193	1170 390	818 273	1398 466
6 Modos E.D.: 4 y 10 en 39,987% y 24,100%	417 139	436 145	421 140	304 101
7 Modos E.D.: 5 y 12 en 38.299% y 49.376%	1440 480	1066 355	1495 498	588 196
8 Modos E.D.: 1 y 8 en 39,773% y 31,837%	335 112	552 184	269 90	446 149

5.3.2 Función Objetivo CalibraV (ec. 2.12):

5.3.2.1 Viga

Tabla 17. Errores mínimos alcanzados en la función objetivo CalibraV ante ausencia de ruido: Viga, daño múltiple 2 elementos

VIGA DAÑO MÚLTIPLE 2 ELEMENTOS	Sin Ruido
6 Modos E.D.: 2 y 8 en 5.197% y 5.652%	358 119
8 Modos E.D.: 8 y 10 51.487% y 30,877%	324 108

5.3.2.2. Armadura

Tabla 18. Errores mínimos alcanzados en la función objetivo CalibraV ante diversos niveles de ruido: Armadura, daño múltiple 2 elementos

ARMADURA DAÑO MÚLTIPLE 2 ELEMENTOS	Sin Ruido	Ruido de 0.15 y 3 %
5 Modos E.D.: 4 y 9 en 27.187% y 54.148%	205 68	222 74
6 Modos E.D.: 10 y 13 en 30.393% y 56.724%	260 87	246 82

Ante los resultados obtenidos anteriormente se aprecia que para la viga no existen inconvenientes respecto al tiempo de ejecución del programa, sin importar los valores aleatorios generados por el algoritmo de PSO para las posiciones iniciales de las partículas, mientras que para la armadura, dependiendo de estas posiciones se llegaba a un número de iteraciones correspondientes, lo cual se refleja en el tiempo de ejecución del algoritmo.

6. CONCLUSIONES Y SUGERENCIAS PARA TRABAJOS FUTUROS

En el presente trabajo fue estudiada la aplicabilidad de la técnica de inteligencia artificial PSO a la detección de daños. En la herramienta de PSO para MATLAB se disponía de tres tipos de PSO, Trelea, Clerc y común con inercia, cuyos desempeños fueron analizados. El problema de detección de daños fue abordado como un problema de optimización, el cual permitió detectar elementos probablemente dañificados.

Esta metodología fue alcanzada a partir de una amplia revisión bibliográfica, que incluye diversos tópicos relacionados a la resolución del problema de detección de daño: metodologías de detección de daño que utilizan parámetros dinámicos, aplicación de técnicas de optimización en la localización y cuantificación de daños en estructuras, este último aspecto siendo de vital importancia, ya que a partir de este estudio se implementaron funciones objetivo propuestas previamente.

Varios tipos de sistemas estructurales, bajo diversos escenarios de daño, fueron estudiados mostrando la robustez de las metodologías estudiadas. A continuación son descritas las principales conclusiones relacionadas a la metodología de localización de daños y los algoritmos evolutivos implementados (PSO), así como sugerencias de trabajos futuros.

6.1 METODOLOGIA DE LOCALIZACIÓN DE ELEMENTOS CON DAÑO

En el presente trabajo fue estudiada una metodología de localización de elementos probablemente dañificados, que esta basada en variaciones de las propiedades dinámicas de la estructura (frecuencias y modos de vibración) como información requerida para considerar el problema de detección de daños como un proceso de optimización. Los resultados muestran que, con mediciones

experimentales ideales, y con ruidos promedio, la metodología consigue detectar cuales elementos presentan daño en la estructura acertadamente.

Cuando las mediciones presentan niveles de ruido severo, el desempeño de la metodología es bastante afectado, resultado que es apreciado por el aumento en el número de elementos definidos como damnificados cuando. La principal ventaja de esta metodología implementada es su simplicidad conceptual, permitiéndose la determinación de los elementos probablemente damnificados sin cálculos avanzados. El tiempo gastado en el procesamiento de las funciones objetivo es mínimo.

Como es observado, a pesar del daño no tener una extensión considerable, la metodología consigue detectar los elementos bajo condiciones normales de ruido. Se espera que para grandes estructuras su efectividad sea aún mayor debido al hecho de que el daño se concentra de forma más localizada en algunas regiones del espacio de la estructura.

Las principales desventajas de la utilización de esta metodología pueden ser:

- El carácter subjetivo de la definición de cuando un elemento se encuentra damnificado.
- La detección de daño uniforme es de difícil consecución para la metodología.
- No está definido cual es la cantidad de modos que debería ser utilizada para la obtención de resultados en forma confiable dependiendo del escenario de daño procurado.
- Si el valor del daño es muy bajo, para un determinado elemento de la estructura, y el ruido es superior a los valores promedio, puede haber grandes variaciones en el valor del daño determinado para ese elemento, lo cual podría resultar en una interpretación errada de cuáles son los elementos damnificados.

6.2 OPTIMIZACION POR ENJAMBRE DE PARTICULAS PSO

Es esta investigación, se ha propuesto el uso del algoritmo PSO para la detección de daños en estructuras planas. El uso de esta técnica puede simplificar el algoritmo necesario para la identificación del sistema. Los resultados experimentales mostraron la eficacia del sistema propuesto en identificar el sistema consistentemente. Como conclusión general se encuentra que de los tres tipos de PSO estudiados (Común con inercia, Trelea y Clerc), sólo se obtuvieron resultados positivos con el tipo Trelea, el cual fue el más adecuado para la optimización de las funciones objetivo dadas, siendo los resultados prácticamente exactos cuando las mediciones se encontraban libres de ruido.

La técnica de PSO posee una ventaja sobre muchas otras técnicas de optimización por su facilidad de implementación, por su robustez y por su poca cantidad de parámetros requeridos para el análisis.

Para los ejemplos estudiados, se encontró que la técnica PSO tiene capacidades para encontrar escenarios de daño en condiciones de mediciones reales con ruido y con pocas formas modales disponibles. Diversos tipos estructurales y diferentes tipos de daño (simples y múltiples) fueron estudiados mostrando la capacidad de la PSO para determinar daños.

En general los resultados muestran que el algoritmo realiza una buena determinación de la extensión del daño de los elementos damnificados, ya que son muy pocos los elementos considerados en forma errada como damnificados.

A continuación son mencionadas algunas conclusiones sobre los diversos análisis realizados:

- La presencia de ruido tiene un efecto directo sobre lo que es considerado como gbest (mejor valor global), lo que podría traer problemas en identificar si el

algoritmo provee o no una buena respuesta, cuando se quieren utilizar mediciones experimentales y el grado de ruido en las mediciones no sea conocido.

- De las funciones objetivo propuestas, solo una (Ec. 2.14) resultó ser adecuada para el problema de detección de daños en todos los casos analizados, ya que la otra sólo permite determinar el daño en condiciones muy excepcionales. Los factores de peso utilizados, fueron considerados iguales tanto para las frecuencias como para las formas modales, lo cual llevo a tener la misma influencia por parte de ambas propiedades, por lo cual no se pudo concluir cual inclusión de ruido en las mediciones afecta más los resultados.

- Para la metodología de localización, la cantidad de información suministrada al algoritmo puede ser un factor decisivo para tener un buen desempeño y no se tiene una regla definida sobre cuántos modos deben ser utilizados. Cabe resaltar, que esa cantidad se encuentra limitada por el número de modos que pueden ser medidos experimentalmente. Así mismo, el número y la posición de los grados de libertad medidos pueden influir en el desempeño de la PSO.

- La definición de los parámetros del AG: tipo de PSO, posición inicial de las partículas, tamaño de la población (número de partículas), número máximo de épocas, velocidad de las partículas, espacio de búsqueda, fue realizada con base en la revisión bibliográfica y en algunas pruebas. La magnitud de tales parámetros es una desventaja ya que la configuración de esos parámetros podría no funcionar cuando sean probados en otros tipos de estructuras. Por lo tanto, la utilización de PSO adaptativos (auto-configurados) es algo deseable.

- Los tiempos de procesamiento dependen del tipo de estructura analizada y de la posición inicial de las partículas. En la medida en que el número de elementos aumenta, el tiempo consumido por las metodologías se puede tornar prohibitivo. Este problema puede ser solucionado si los AG son implementados utilizando programación paralela y utilizando heurísticas que permitan disminuir el tiempo de evaluación de la función objetivo y disminuir el espacio de búsqueda para cada elemento.

6.3 SUGERENCIAS PARA TRABAJOS FUTUROS

A partir de la realización de esta investigación, se sugieren los siguientes temas para su desarrollo en trabajos futuros:

- Análisis de funciones objetivo basadas en diferentes parámetros dinámicos (energía de deformación modal, flexibilidad, vector de fuerza residual, funciones de respuesta en frecuencia), permitiéndose una determinación más adecuada de la detección de daños.
- Definición de una metodología de localización de daño que permita encontrar de una forma confiable cuales son los elementos probablemente damnificados en la estructura.
- Estudio del problema de localización óptima de sensores.
- Utilización de la técnica de redes neuronales y algoritmos genéticos para la solución del problema de detección de daños.
- Aplicación de técnicas de auto- adaptación para la definición de los parámetros de la PSO.
- Implementación de técnicas clásicas de optimización para observar su desempeño en la resolución del problema de detección de daños.
- Evaluación con un problema experimental real, de la metodología desarrollada.
- Ajustar los pesos de las frecuencias y formas modales en las funciones objetivo según lo aconsejado en la literatura, donde para determinar los pesos relativos el varios autores han presentado un análisis de la sensibilidad de estos parámetros en el daño, encontrando que las formas modales contribuyen poco; por eso, el término relacionado con las frecuencias naturales debe tener mayor relevancia en los análisis a desarrollar.
- Correr el programa de PSO asignando el valor del modulo de elasticidad sin daño a todos los elementos, como posiciones iniciales para las partículas.
- Desarrollar una metodología de detección de daños que pueda ser aplicada a daños de tipo uniforme y disperso.

BIBLIOGRAFÍA

ATIENZA P. RICARDO. Técnicas de actualización aplicadas a la detección de daños. Escuela Técnica Superior de Ingenieros Aeronáuticos. Departamento de Motopropulsión y Fluidodinámica, 2004.

BIRGE, B., "PSOt - a particle swarm optimization toolbox for use with Matlab," Swarm Intelligence Symposium, 2003. SIS '03. Proceedings of the 2003 IEEE. 24-26 April 2003.

CELIS M. CAMILO E., MILLAN P. CARLOS A. Evaluación del desempeño de algoritmos de evolución paramétrica para optimización. Universidad Industrial de Santander. Escuela de Ingeniería Civil, 2009.

CHAMBERS L. *Practical Handbook of Genetic Algorithms: Applications*, volume 1. CRC Press, 1995.

CLERC MAURICE. Particle Swarm Optimization. ISTE Ltd. USA, 2006.

COLORNI A., DORIGO M. AND MANIEZZO V. An Investigation of some Properties of an Ant Algorithm. *Proceedings of the Parallel Problem Solving from Nature Conference (PPSN 92), Brussels. Elsevier Publishing*, 1992.

DIGALAKIS J. AND MARGARITIS K. An Experimental Study of Benchmarking Function for Genetic Algorithms. *Department of applied informatics, University of Macedonia. International Journal Computer Math*. Vol. 79, 2002.

_____, _____. On Benchmarking functions for genetic algorithms. *Department of applied informatics, University of Macedonia. International Journal Computer Math*. Vol. 00, 2002, pp. 1-27.

DORIGO M. AND L.M. GAMBARDELLA. Ant Colonies for the Traveling Salesman Problem. *BioSystems, in pres (Also Technical Report TR/IRIDIA/1996-3, IRIDIA, Université Libre de Bruxelles*. 1997

_____, _____. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. *IEEE Transactions on Evolutionary Computation*. Vol.1, No.1, 1997. In press.

_____, V. MANIEZZO AND A. COLORNI. The Ant System: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics-Part B*. Vol 26, N° 1, 1996.

ESPINOZA B. FORTUNATO. Determinación de características dinámicas de estructuras. Universitat Politècnica de Catalunya. Departament de Enginyeria del Terreny Cartogràfica i Geofísica, 1999.

GLOVER F. AND KOCHENBERGER G.G. *Handbook of Metaheuristics*. Kluwer Academic Publishers. 2003.

GOLDBERG DAVID E. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison Wesley Longman, Inc. USA, 1989.

HOLLAND, JOHN. Genetic algorithms. *Scientific American*. Vol 267, No 1, 1992.

KALYANMOY DEB. Genetic algorithms in multimodal function optimization. *TCGA report no. 89002, The University of Alabama, Dept. of Engineering mechanics*, 1989

KENNEDY J AND EBERHART R. Particle Swarm Optimization. *Proceedings of IEEE International Conference on Neural Networks*, 1995.

_____, _____. *Swarm Intelligence*. Morgan Kaufmann Publishers, 2001

KENNEDY J. Thinking is social: Experiments with the adaptive culture model. *Journal of Conflict Resolution* Vol 42, N° 1, 1998.

LIU, L., "Robust fault detection and diagnosis for permanent magnet synchronous motors," Ph.D.dissertation, Dept. Mech. Eng., Florida State University, Tallahassee, FL, 2006.

MELIÁN B, MORENO PEREZ J AND MORENO VEGA J. Metaheuristics: A global view. *Revista Iberoamericana de Inteligencia Artificial*. Vol 2, N° 19, 2003.

PANDA, G.; MOHANTY, D.; MAJHI, B.; SAHOO, G., "Identification of nonlinear systems using particle swarm optimization technique," *Evolutionary Computation*, 2007. CEC 2007. IEEE Congress on Sept. 2007. pp.3253-3257.

PENG XIAO; VENAYAGAMOORTHY, G.K.; CORZINE, K.A., "Combined Training of Recurrent Neural Networks with Particle Swarm Optimization and Backpropagation Algorithms for Impedance Identification," *Swarm Intelligence Symposium*, 2007. SIS 2007. IEEE. 1-5 April 2007.

PÉREZ L. JESÚS R. Contribución a los métodos de optimización basados en procesos naturales y su aplicación a la medida de antenas en campo próximo. Universidad de Cantabria. Departamento de Ingeniería de Comunicaciones, 2005.

RIVEROS JEREZ CARLOS A. Metodología de monitoreo de daño estructural para puentes simplemente apoyados: implementación numérica. Universidad de Antioquia, Colombia, 2006.

SALAZAR P. PEDRO Y. Algoritmo híbrido auto configurado para optimización estructural. Universidad Industrial de Santander. Escuela de Ingeniería Civil, 2009.

SETTLES, M., RODEBAUGH, B., AND SOULE, T. Comparison of genetic algorithm and particle swarm optimizer when evolving a recurrent neural network. *Proceedings of the Genetic and Evolutionary Computation Conference 2003 (GECCO 2003), Chicago, IL, USA.* 2003.

TEBALDI ADRIANO, DOS SANTOS COELHO LEANDRO, LOPES JR. VICENTE. Detecção de falhas em estruturas inteligentes usando otimização por nuvem de partículas: fundamentos e estudo de casos. *Revista Controle & Automação/Vol.17 no.3, Brasil* 2006.

VESTERSTROM J. AND THOMSEN R. A Comparative Study of Differential Evolution, Particle Swarm Optimization, and Evolutionary Algorithms on Numerical Benchmark Problems. *Congress on Evolutionary Computation. CEC04, Vol, N° 1, 2004.*

VILLALBA M. JESÚS. Detecção de dano em estruturas utilizando algoritmos genéticos e parâmetros dinâmicos. Universidade de São Paulo, Escola de Engenharia de São Carlos, 2009.

Y. SHI, "Particle Swarm Optimization," Feature article, Electronic Data Systems, Inc., IEEE Neural Networks Society, Feb. 2004.

ZE DONG; PU HAN; DONGFENG WANG; SONGMING JIAO, "Thermal Process System Identification Using Particle Swarm Optimization," *Industrial Electronics, 2006 IEEE International Symposium on* , vol.1. July 2006.

ZIEGLER J. G. AND NICHOLS N. B. Optimum settings for automatic controllers. *Trans. A.S.M.E., Vol. 64.* 1942.

ANEXO A. COMPLEMENTACIÓN TEÓRICA

DETECCIÓN DE DAÑOS

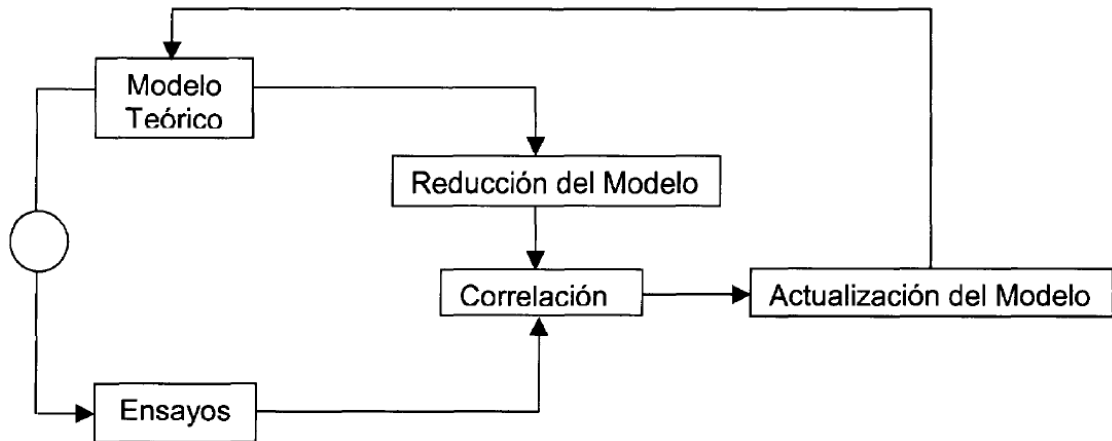
La Detección de Daños es una técnica en continua evolución y cada vez más utilizada en el mundo actual. Esta técnica aparece como el resultado de los esfuerzos de los departamentos de ingeniería de las diferentes empresas en su continua búsqueda por obtener una reducción de los niveles de vibración y ruido, aumentar en lo posible la expectativa de vida operacional de los diferentes sistemas así como incrementar la eficiencia y seguridad de los mismos.

La técnica de Detección de Daños nace como consecuencia directa de la unión de las técnicas de Sintonización de Modelos con las Técnicas Experimentales, siendo conjuntamente capaces de aumentar la potencia de las predicciones analíticas realizadas mediante cálculos por Elementos Finitos para la caracterización dinámica de los sistemas.

El principio en que se basan los algoritmos para la detección de daños es que cualquier cambio en el modelo espacial (defectos), produce cambios observables en la respuesta dinámica del sistema. No es difícil observar que la filosofía que está detrás de los procedimientos de detección de daños está en la línea que nos indica la intuición de que la aparición de un daño o de un defecto en una estructura cambia el ruido que esta produce en su funcionamiento. En términos matemáticos para dar significado al concepto de daño es necesario recurrir a la comparación entre dos diferentes estados del sistema, uno de ellos (el inicial) será en el que se asume la estructura sin daño.

Para llevar a cabo un estudio de detección de daños en una estructura, normalmente se suele realizar previamente un modelo que represente analíticamente el comportamiento dinámico de la misma, además de una fase experimental en la que se obtiene la respuesta dinámica de la estructura. Estas fases dentro del procedimiento de detección de daños no varían en gran medida de las seguidas en los procesos tradicionales de diseño y desarrollo de los diferentes sistemas estructurales, representado en el siguiente esquema:

Figura 40. Esquema de detección de daños



Modelo Analítico

En la elaboración del modelo dinámico de la estructura hubo una enorme evolución a mediados del siglo pasado con el nacimiento de la tecnología de los ordenadores. Así, en el campo de la industria aeroespacial se comenzaron a desarrollar técnicas de análisis numérico entre las décadas de los 50 y 60. Entre ellas el Método de los Elementos Finitos (FEM) (Zienckiewicz 1967) demostró ser una excelente herramienta en la obtención de soluciones aproximadas para problemas en los que su resolución por métodos tradicionales resultaba muy complicada y costosa. Más tarde estas técnicas se fueron difundiendo a otros campos de la ingeniería produciendo cambios significativos en los métodos de desarrollo y producción.

En el campo de la dinámica estructural, el análisis estructural se ha limitado a la obtención de las propiedades modales del sistema y las predicciones se comparaban frecuentemente con las formas modales y autovalores obtenidos gracias a la experimentación (Ewins 1984). Para ello existe una gran variedad de herramientas numéricas de correlación con el fin de validar las propiedades obtenidas mediante análisis y aquellas obtenidas a través de ensayos. De la misma forma existen algoritmos que permiten mejorar de forma sistemática los modelos de Elementos Finitos utilizando las propiedades dinámicas medidas, es lo que llamamos la sintonización de modelos.

En este sentido, la técnica del análisis por Elementos Finitos se ha convertido en la herramienta de análisis numérico más extendida en el campo aeroespacial, siendo la mayor parte de las veces una pieza clave en el proceso de diseño. Esta técnica de diseño continua sufriendo mejora gracias a los avances que se

producen en la tecnología de los ordenadores y en la disponibilidad de paquetes de cálculo por Elementos Finitos cada vez más intuitivos y sencillos de manejar.

El método de los Elementos Finitos permite el análisis de problemas estructurales dinámicos complejos, siendo la validación del modelo una práctica siempre recomendable. Sin embargo, conocer dónde se encuentra el error en el modelo es a menudo una tarea compleja. Hasta la fecha, el único medio de validación de un modelo de Elementos Finitos es comparar un número limitado de propiedades dinámicas predichas con sus correspondientes valores medidos a través de ensayos y llegar a conclusiones sobre la bondad del modelo.

Probablemente el método más extendido sea la correlación de los datos obtenidos mediante ensayos con las predicciones calculadas con el modelo de Elementos Finitos. Las técnicas más utilizadas son las de correlación modal y normalmente utilizan medidas correladas. Una breve introducción a este tema se puede encontrar en el libro de Ewins (1984), presentaciones más extensas del mismo pueden observarse en los artículos presentados por Heylen and Avitabile (1998) y O'Callahan (1998). En general, las técnicas de correlación son una mezcla entre medios visuales y numéricos para identificar las diferencias existentes entre las predicciones y los datos medidos. Con las técnicas de correlación numérica se obtiene un valor numérico, mientras que las técnicas de correlación visuales son de naturaleza cualitativa.

Si bien el valor de estos procedimientos ha sido largamente probado en los últimos años, el éxito de las técnicas de sintonización de modelos depende directamente de la calidad de las técnicas de extracción de los parámetros modales. De hecho, el análisis de las señales medidas para la identificación de los parámetros modales del sistema es un proceso que inevitablemente introduce imprecisiones y errores que se suman a los ya existentes en las medidas.

Como resumen de todo lo anteriormente descrito puede exponerse que el método de los elementos finitos se ha convertido en una metodología estándar en el desarrollo de vehículos aeroespaciales. Durante la etapa de diseño y desarrollo se manejan modelos de elementos finitos muy detallados que permiten predecir el comportamiento del sistema en los estados de carga que va a sufrir durante toda la vida el sistema.

Una vez construidos los primeros prototipos es habitual llevar a cabo una serie de ensayos específicos para comprobar que el modelo de cálculo desarrollado es adecuado y predice adecuadamente el comportamiento del sistema. En general, los resultados de estos ensayos se utilizan para mejorar el modelo y obtener un modelo de elementos finitos que reproduzca con mayor fidelidad los resultados del espécimen real.

Para poder comparar los resultados del modelo de elementos finitos con los de ensayo es necesario manejar modelos de tamaño reducido, muy inferiores a los

modelos de desarrollo. Para esta tarea se utilizan las técnicas de reducción de modelos de elementos finitos.

Las técnicas utilizadas para perfilar el modelo se conocen como técnicas de "Sintonización" (Güemes, López, Cuerno y Marco 2003), (Friswell and Mottershead 1995), (Imregun and Visser 1991).

Estas técnicas se usan hoy en día habitualmente durante el desarrollo de sistemas.

Técnicas Experimentales

El análisis experimental es el problema inverso, esto es, a través de la respuesta de un sistema trata de inferir las ecuaciones del mismo, y por tanto las características físicas que lo determinan dinámicamente. Estas propiedades que definen al sistema, son los llamados parámetros modales.

El objetivo del análisis modal experimental involucra las tareas y trabajos necesarios para obtener las medidas directas de la respuesta de un sistema y, a partir de ellas, obtener los parámetros modales que caracterizan el sistema dinámico. Los parámetros modales, se pueden estimar a través de una serie de medidas, almacenadas de forma discreta, en los diferentes dominios (tiempo y/o frecuencia). Estas medidas se pueden procesar una a una, o bien, en grupos parciales o completos de forma simultánea. Hay un gran abanico de posibilidades a la hora de adquirir todas estas medidas, ya que pueden obtenerse sin medir la excitaciones, midiendo únicamente en un punto, midiendo en diferentes puntos, midiendo individualmente, o de forma simultánea,...

Es decir hay una tremenda variedad a la hora de elegir la forma de la obtención de estas medidas y los tipos de condiciones de contorno.

En cuanto a la forma de la obtención de la respuesta, esto es, a las condiciones de la realización del ensayo, existen a su vez varias formas de excitación del sistema. Típicamente, se utilizan fuerzas excitadoras en diferentes puntos del sistema y se miden las señales de estas fuerzas, así como las respuestas vibratorias del sistema producidas en otros tantos puntos. Debido a la hipótesis de comportamiento lineal del sistema, no es necesario excitar todos los puntos simultáneamente. Es posible adquirir las señales de respuesta, pudiendo combinarlas más tarde durante el postproceso de los resultados del ensayo (Van der Auweraer and Hermans 1999), (Van der Auweraer 2001).

Diferentes algoritmos han permitido no sólo mejorar el modelo de elementos finitos desarrollado para sistemas, también permiten conocer el origen de las discrepancias entre el modelo y los resultados reales.

Estado Actual en Detección de Daños

Inicialmente, se lleva a cabo una búsqueda en la literatura especializada en este campo, para conocer el estado actual de la técnica de Detección de Daños y las líneas de actuación que han llevado los diferentes investigadores dedicados a la misma.

Así, los primeros estudios se deben a Lifshitz and Rotem (1969). Miden frecuencias propias y factores de amortiguamiento en piezas de poliéster y epoxi reforzadas con partículas de cuarzo. El daño se induce mediante cargas estáticas, midiéndose de forma continua durante el proceso de carga. Los autores observan que el factor de amortiguamiento es más sensible a los daños que los cambios en el resto de parámetros dinámicos.

Schultz y Warwick (1971) miden la respuesta de vigas de epoxi reforzadas con fibra de vidrio en diferentes intervalos de tiempo durante un ensayo de fatiga, encontrando que las frecuencias propias son relativamente insensibles a la presencia de daños, observando, sin embargo, que los factores de amortiguamiento y la magnitud de la respuesta son muy sensibles.

Adams et al. (1975) ensayan placas fabricadas con fibra de carbono y vidrio unidireccional, tratando de detectar la presencia de daño tras cargarlas con momentos torsores tanto estáticos como dinámicos. Observan que los factores de amortiguamiento son más sensibles a la presencia de daños que las frecuencias propias del sistema, así como que pequeños cambios en las propiedades dinámicas en las primeras fases de la aparición del daño, pueden recuperarse tras un periodo de reposo.

Stubbs et al. (1990) aplican un método de identificación de daños a una estructura en celosía plana en voladizo compuesta por 20 celdas cuadradas de 1 m de lado. Realizan una aproximación continua de la estructura consistente en una viga de Bernouilli-Euler de 10 elementos para el comportamiento a flexión y una barra uniforme para el comportamiento axial. Consideran una serie de 14 condiciones de daño, incluyendo múltiples posiciones del mismo. Construyen matrices de sensibilidad para la identificación del daño y su tamaño a partir de las frecuencias naturales y formas propias calculadas para la estructura sin daño y la estructura dañada, consiguiendo en la mayoría de los casos resultados coherentes. Sin embargo, los autores apuntan de la posibilidad de predicciones falsas en zonas cercanas al soporte y al extremo libre.

Hunt et al. (1990) describen el desarrollo y la implementación del Sistema de Inspección Modal del Shuttle (SMIS). Llevaron a cabo ensayos modales sobre diferentes especímenes para localizar el daño producido en el borde de ataque, el flap y los paneles delanteros, medios y traseros del fuselaje. En su artículo estudian los resultados obtenidos en estos ensayos, así como los obtenidos

mediante ensayos sobre el conjunto completo. También describen la utilización de la estructura de una Cessna utilizada en ensayos posteriores.

Paolozzi y Peroni (1990) estudian la variación en frecuencia debida a delaminaciones en estructuras de materiales compuestos utilizando el Método de los Elementos Finitos para su predicción analítica. El espécimen en estudio es un sándwich con núcleo de panal de abeja y las caras laminados de fibra de carbono. Considerando diferentes modelos se observa que las variaciones en frecuencia son mayores para aquellos modos en los que la longitud de onda de la forma modal es del tamaño de la delaminación.

Fox (1992) realiza ensayos de impacto en vigas de 1 m de longitud x 12 mm de espesor con grietas de profundidad de 1 mm, 3 mm y 6 mm situadas a 200 mm del extremo de las mismas. Lleva a cabo el análisis modal numéricamente mediante Elementos Finitos. Observa que, aunque las frecuencias naturales y los valores del MAC sufren pequeñas variaciones, el daño es posible localizarlo estudiando los cambios relativos en las formas modales cuyas variaciones en frecuencia son debidas a la presencia del daño.

Srinivasan y Kot (1992) ensayan una estructura cilíndrica de pared de 2,5 mm, 305 mm de diámetro y 686 mm de longitud en la que mecanizan un agujero a la mitad de su altura. Para simular condiciones de contorno de libre suspenden el cilindro mediante muelles muy poco rígidos y excitan la estructura mediante la técnica de impacto. No aplican un método específico de identificación de daño, inspeccionando las variaciones producidas en las frecuencias propias y en las formas modales. Encuentran que las formas propias son un indicador de daño más sensible que las frecuencias propias, ya que estas resultan ser insensibles a la presencia del daño, produciéndose cambios en su magnitud por debajo de la resolución de las medidas. La forma de cuantificar el daño lo realizan mediante el MAC entre las formas modales del cilindro dañado y el cilindro sin daño.

Alvelid y Gustavsson (1992) utilizan FRF's, funciones de correlación cruzada y medidas de impedancia local para caracterizar los efectos del daño en la respuesta dinámica de una placa de material compuesto reforzada con fibra de vidrio. De todos ellos recomiendan la utilización de las FRF's para la localización del daño, ya que este método goza de ciertas ventajas debidas a la naturaleza global de la respuesta contenida en las FRF's y al hecho de que el tamaño del daño puede determinarse a partir del tamaño de la longitud de onda de la forma modal correspondiente al modo más afectado por el daño.

Huang y Gu (1993) estudian una simulación por Elementos Finitos de una viga en voladizo de 400 mm de longitud y una sección transversal de 6 mm x 35 mm. Los autores utilizan un único nodo y calculan un Modelo de Movimiento Medio Autorregresivo (ARMA) para ajustar la respuesta. El parámetro que tienen en cuenta para el estudio es la acumulación del residuo, observando que, para

estudios numéricos con y sin ruido, resulta ser muy próxima a cero, mientras que en las vigas con daños es distinta de cero debido a presencia de no linealidades.

Salawu y Williams (1993) aplican el método de la matriz error a un modelo de Elementos Finitos de una viga y comparan los resultados con otro tipo de criterios como los cambios en las formas modales normalizadas con la masa y las variaciones que se producen en la matriz de rigidez experimental. Presentan el método de la matriz cursor, que usa la teoría espacial de vectores para identificar las filas y columnas distintas de cero de una matriz de error dinámica. En todos estos métodos utilizan matrices de parámetros experimentales.

Slater y Shelley (1993) aplican un esquema de filtrado modal adaptativo a una estructura en celosía suspendida compuesta por 9 celdas cuadradas de 4,5 m de lado. Demuestran la capacidad de este método de detectar variaciones en las frecuencias propias a lo largo del tiempo añadiendo o quitando condiciones de contorno.

Budipriyanto y Swamidas (1994) miden los parámetros modales de una viga en voladizo con entallas en diferentes condiciones, en aire, parcialmente y completamente sumergida en agua. A la viga le provocan entallas simétricamente en la cara superior y en la inferior, realizando medidas con acelerómetros y extensímetros. Entre los resultados presentan las frecuencias naturales, factores de amortiguamiento para todas las combinaciones de grados de inmersión y tipo de transductor utilizado.

Ko et al. (1994) aplican una técnica de identificación de daño basada en los parámetros MAC/COMAC de los datos de la respuesta obtenidos en un pórtico cuadrado de acero de 2 m de lado. El pórtico está instrumentado en once posiciones y se excita mediante la técnica de impacto para estimar las frecuencias propias y formas modales con diferentes tipos de unión de las barras, rígidas y articuladas. Posteriormente utilizan un vibrador para obtener con mayor precisión las formas modales. Simulan el daño quitando tornillos en dos puntos, uno en la unión de dos columnas y otro en la unión de la columna a la base.

Friswell et al. (1994) utilizan como especímenes de ensayo una placa en voladizo y una estructura reticulada, ambas de acero y teniendo ambas una longitud de 0,5 m. El daño lo simulan mediante cortes en ambas estructuras. Mediante el método aplicado se consigue localizar el daño pero aparecen ciertos indicios de daño cuando se utilizan las estructuras intactas.

Kim et al. (1994) presentan un método de identificación modal basado en las respuestas libres de un sistema para llevar a cabo el seguimiento de la integridad estructural de estructuras aeroespaciales utilizando técnicas de proceso de datos aleatorios en el dominio del tiempo. Los autores determinan los cambios de

frecuencia experimentados en el lanzador Delta II debidos a la pérdida de masa de propulsante durante el vuelo.

Saravanos et al. (1994) utilizan su teoría para predecir numéricamente la respuesta de sensores piezoeléctricos embebidos en laminados de material compuesto $[p/0/90/0/90]_s$ con delaminaciones. Los resultados indican la posibilidad de localizar delaminaciones de varios tamaños y en varias posiciones, en y fuera del plano medio.

Lim (1994,1995) aplica una técnica de detección de daños a una estructura en celosía compuesta por 20 celdas planas. Primeramente localiza las zonas dañadas utilizando el concepto de los mejores autovectores obtenibles y cuantifica la magnitud del daño mediante la aplicación de una combinación de restricciones a la estructura. Demuestra que el método funciona bien incluso con un número limitado de grados de libertad instrumentados.

Saitoh y Takei (1996) aplican el método de sensibilidad modal desarrollado por Richardson y Marinan (1992) a una placa con una grieta. Los resultados muestran pequeños decrementos en las frecuencias propias de la placa con el daño, los cambios en los factores de amortiguamiento son inconsistentes y los valores del MAC no presentan cambios cuando se compara la placa dañada con la placa sin daño.

Choudhury y He (1996) utilizan un método propio de detección de daños utilizando cambios en las FRF's y matrices de masa y rigidez del modelo de Elementos Finitos. Para llevar a cabo su estudio, utilizan una estructura reticular plana de dimensiones 900 mm x 420 mm. El daño provocado consiste en dos cortes de 5 mm de ancho cada uno, localizados en la misma sección en zonas opuestas de una de las vigas. Utilizando su método juntamente con un procedimiento de expansión dinámica, para estimar los valores de las FRF's en grados de libertad no medidos del modelo de Elementos Finitos, son capaces de localizar el daño en la estructura.

Robinson et al. (1996) y Doebling et al. (1995) analizan los resultados del ensayo modal para localizar el daño producido en un tirante de la estructura soporte de la piel de un DC-9. el tirante tiene cortes transversales de diferentes tamaños hasta un total de cuatro. Los datos de la pieza sin daño y los de la pieza dañada se analizan mediante un análisis de la flexibilidad, siendo los puntos en los que se produce una disminución significativa de la rigidez transversal zonas donde existe daño. Para zonas en las que el daño es pequeño, el indicador resulta poco exacto. Realizan una comparación entre dos métodos de identificación de daño, el primero basado en la evaluación directa del cambio de la flexibilidad puntual, y el segundo utilizando la disgregación de la matriz de flexibilidad experimental presumiendo una cierta conectividad de los elementos. La comparación directa de los cambios puntuales de flexibilidad proporciona una identificación del daño más exacta. Los

autores presuponen que los errores que se producen en la técnica de disgregación de la matriz de flexibilidad se deben a un patrón supuesto de conectividad de los elementos inadecuado y al hecho de que la matriz de flexibilidad experimental no es completa.

Wang et al. (1998) proponen un método para el seguimiento de la integridad estructural basado en la dinámica estructural para grandes estructuras. Comprueban la validez del método numérica y experimentalmente implementando el método en el modelo de una estructura de gran tamaño. Dividen el proceso de implementación, en una serie de pasos: 1) realización un modelo de elementos finitos con el fin de establecer una línea de referencia, 2) posicionado de los sensores para conseguir un proceso económico y 3) detección de daños.

Finalmente, Doebling et al. (1998) presentan una extensa revisión de los desarrollos actuales y comenta los diferentes aspectos prácticos de los algoritmos utilizados para la detección de daños. Utiliza una clasificación del daño y la habilidad para identificar el daño para clasificar los algoritmos de detección de daños en niveles de complejidad creciente. Hay formulaciones que determinan la existencia de daños en la estructura (nivel 1), métodos que identifican el daño y su localización (nivel 2) y métodos que tratan de localizar y cuantificar el daño en la estructura (nivel 3). Los métodos de detección de daños basados en vibraciones que no utilizan el correspondiente modelo teórico de la estructura se encuentran entre las categorías de los niveles 1 y 2. Si la identificación del daño se realiza conjuntamente con un modelo matemático, el método se clasifica dentro del nivel 3 y los procedimientos de los métodos contenidos en este nivel están íntimamente relacionados con las técnicas de Sintonización de Modelos de Elementos Finitos.

La mayor parte de las publicaciones hasta el momento se centran en el estudio de algoritmos basados en las propiedades modales. Las diferencias de estas propiedades modales entre la estructura sana y la dañada suelen ser muy pequeñas, es por ello que algunos autores proponen enriquecer los algoritmos de cálculo con las respuestas medidas directamente (Mai et al. 1997). Sin embargo, parece que la precisión necesaria en dichas medidas para este propósito es prácticamente imposible de conseguir.

MÉTODO DE ELEMENTOS FINITOS

El método de elementos finitos es ampliamente utilizado para la descripción del comportamiento de sistemas estructurales, cuyo concepto base se encuentra en la división del sistema continuo en elementos de tamaño finito que hacen que el sistema sea ahora discreto (ASSAN, 2003). En el ámbito de la detección de daños el método de los elementos finitos es utilizado para la representación de las condiciones iniciales y actuales de la estructura, es decir cuando se considera el efecto del daño.

En el caso del problema de vibración axial y de vibración transversal de una estructura no amortiguada, las ecuaciones diferenciales de equilibrio dinámico son dadas, respectivamente, por:

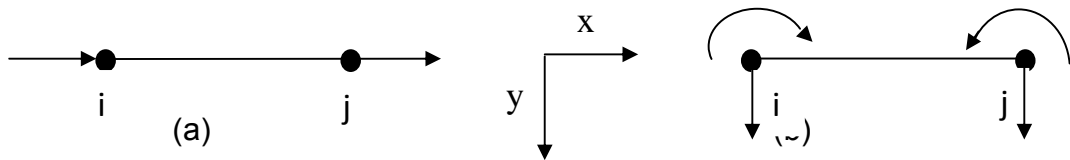
$$Eau'' - m\ddot{u} = -P(x,t) \quad (\text{A.1})$$

$$EIv^{IV} + m\ddot{v} = P(x,t) \quad (\text{A.2})$$

Siendo E el módulo de elasticidad del material, A e I el área y la inercia de la sección transversal, respectivamente.

Un elemento barra y un elemento viga presentan los grados de libertad mostrados en la Figura 41. En el caso del elemento barra se tiene un único grado de libertad por nodo, el cual corresponde a un grado de libertad axial; mientras que la viga presenta un grado de libertad transversal y uno de rotación.

Figura 41. Elementos finitos para (a) Barra y (b) Viga.



En el método de elementos finitos el campo de desplazamiento en el dominio del elemento es expresado en función de los valores de los desplazamientos en los nodos. Para el caso de la barra, considerando un polinomio de primer grado como función aproximada, se tiene:

$$u_k(x,t) = \{\varphi_1(x) \quad \varphi_2(x)\} \{u_i(t) \quad u_j(t)\}^T \quad (\text{A.3})$$

En el cual $u_k(x,t)$ se refiere al campo de desplazamiento en el elemento k para el tiempo t , u_i y u_j son los desplazamientos axiales en los nodos i y j , respectivamente, y:

$$\varphi_1(x) = 1 - \frac{x}{L_i}$$

$$\varphi_2(x) = \frac{x}{L_i} \quad (\text{A.4})$$

son las funciones de forma.

Para el caso de vibración transversal la función aproximada correspondiente se trata de un polinomio de tercer grado:

$$v_k(x,t) = \{\varphi_1(x) \quad \varphi_2(x) \quad \varphi_3(x) \quad \varphi_4(x)\} \{v_i(t) \quad \theta_i(t) \quad v_j(t) \quad \theta_j(t)\}^T \quad (A.5)$$

siendo $v_k(x,t)$ el campo de desplazamientos para el elemento k en el tiempo t , v_i y θ_i , y, v_j y θ_j son los desplazamientos verticales y rotaciones en los nodos i y j , respectivamente. Las funciones de forma están dadas por:

$$\begin{aligned} \varphi_1(x) &= 1 - 3\left(\frac{x}{L_i}\right)^2 + 2\left(\frac{x}{L_i}\right)^3 \\ \varphi_2(x) &= L_i \times \left[x - 2\left(\frac{x}{L_i}\right)^2 + \left(\frac{x}{L_i}\right)^3 \right] \\ \varphi_3(x) &= 3\left(\frac{x}{L_i}\right)^2 - 2\left(\frac{x}{L_i}\right)^3 \\ \varphi_4(x) &= L_i \times \left[-\left(\frac{x}{L_i}\right)^2 + \left(\frac{x}{L_i}\right)^3 \right] \end{aligned} \quad (A.6)$$

A partir de las ecuaciones A.4 y A.6 la aplicación del método de Galerkin conduce a un sistema de ecuaciones lineales en las coordenadas de desplazamiento adoptadas, cuya solución consiste en una solución aproximada del problema estructural.

De esta manera, las matrices de masa y rigidez consistentes del elemento barra son respectivamente:

$$[m_k] = \frac{\rho_k A_k L_k}{6} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \quad (A.7)$$

$$K_k = \frac{E_k A_k}{L_k} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad (A.8)$$

Donde A_k y L_k son el área de la sección transversal y la longitud del elemento k , respectivamente. ρ_k y E_k son, respectivamente, la densidad y el módulo de elasticidad del material del elemento k .

Igualmente para el elemento viga, es considerado el modelo de Euler-Bernoulli (ref), y se obtienen respectivamente las matrices de masa y rigidez:

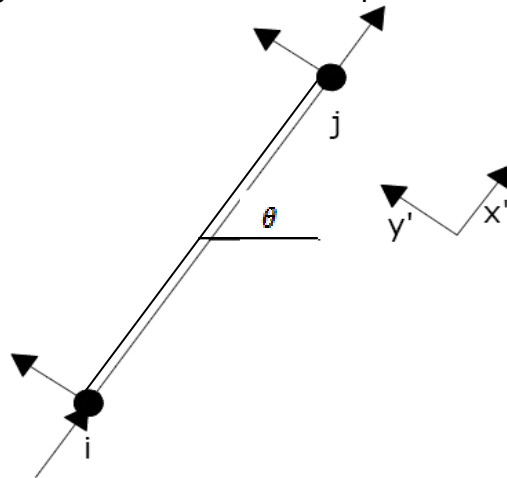
$$[m_k] = \frac{\rho_k A_k L_k}{6} \begin{bmatrix} 156 & 22L_k & 54 & -13L_k \\ 22L_k & 4L_k^2 & 13L_k & -3L_k^2 \\ 54 & 13L_k & 156 & -22L_k \\ -13L_k & -3L_k^2 & -22L_k & 4L_k^2 \end{bmatrix} \quad (\text{A.9})$$

$$[K_k] = \frac{E_k I_k}{L_k^3} \begin{bmatrix} 12 & 6L_k & -12 & 6L_k \\ 6L_k & 4L_k^2 & -6L_k & 2L_k^2 \\ -12 & -6L_k & 12 & -6L_k \\ 6L_k & 2L_k^2 & -6L_k & 4L_k^2 \end{bmatrix} \quad (\text{A.10})$$

siendo I_k el momento de inercia de la sección transversal del elemento k .

Los dos tipos de elementos anteriores son utilizados para formar los elementos correspondientes para las estructuras de tipo armadura, cuyos grados de libertad pueden ser observados en la Figura A.3. Estos grados de libertad son definidos en coordenadas locales y deben ser transformados a coordenadas globales.

Figura 42. Elementos finitos para armadura



Las matrices de masa y rigidez consistentes para una estructura de tipo cercha en el sistema local de coordenadas pueden ser dadas como:

$$[m_k] = \frac{\rho_k A_k L_k}{6} \begin{bmatrix} 2 & 0 & 1 & 0 \\ 0 & 2 & 0 & 1 \\ 1 & 0 & 2 & 0 \\ 0 & 1 & 0 & 2 \end{bmatrix} \quad (\text{A.11})$$

$$K_k = \frac{E_k A_k}{L_k} \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{A.12})$$

Para llevarse las matrices referentes a las coordenadas locales para las correspondientes en coordenadas globales, se utiliza la siguiente operación matricial de transformación (ver Figura 2.7):

$$[M_G] = [T]^T [M_L][T] \quad (\text{A.13})$$

$$[K_G] = [T]^T [K_L][T] \quad (\text{A.14})$$

Siendo la matriz de transformación T en el caso de la armadura:

$$T = \begin{bmatrix} \cos(\theta_k) & \text{sen}(\theta_k) & 0 & 0 \\ -\text{sen}(\theta_k) & \cos(\theta_k) & 0 & 0 \\ 0 & 0 & \cos(\theta_k) & \text{sen}(\theta_k) \\ 0 & 0 & -\text{sen}(\theta_k) & \cos(\theta_k) \end{bmatrix} \quad (\text{A.15})$$

En la cual θ_k es el ángulo dextrorso entre los sistemas de coordenadas global y el sistema local del elemento k .

Finalmente, la matriz de rigidez y masa de la estructura como un todo será formada a partir de la contribución de todos los elementos que la componen, para de esta forma, poder hallar la respuesta dinámica de las estructuras, es decir, aplicar el respectivo análisis modal para obtener las frecuencias y formas de vibración.

A.NEXO B. EJEMPLO DE ANALISIS MODAL

Ejemplo: Para ilustrar los conceptos presentados en el capítulo 2, se ha introducido un ejemplo que será abordado en diferentes partes, posteriormente. El ejemplo se refiere a un modelo de una viga simplemente apoyada en sus extremos, discretizada en 6 elementos, se estableció una sección cuadrada de 0.15cm y un modulo de elasticidad del material igual a 1.787×10^{10} N/m; además, masas concentradas en la direcciones verticales iguales a 288.36kg. A partir del método matricial, que fue programado en la herramienta computacional de Matlab (MathWorks, 2008), se determinó las matrices de rigidez de la viga mostrada en la **¡Error! No se encuentra el origen de la referencia..** Con el fin de reducir la dimensión del problema no se consideraron los grados de libertad axiales y los grados de libertad rotacionales fueron condensados a 5 grados de libertad verticales.

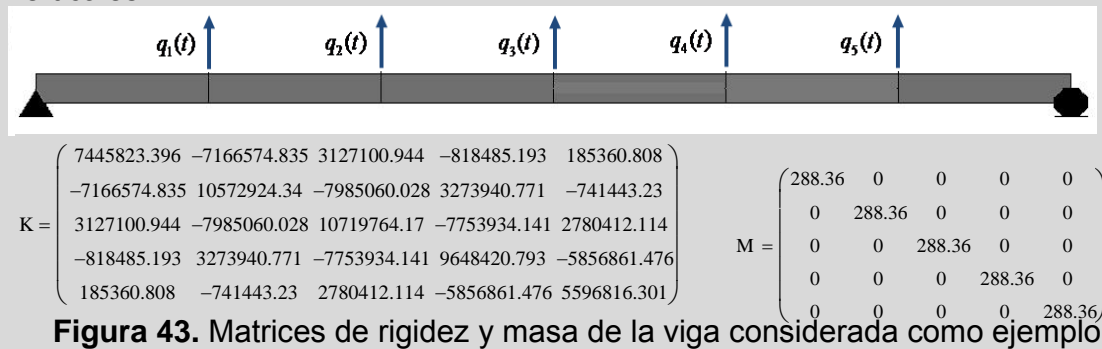
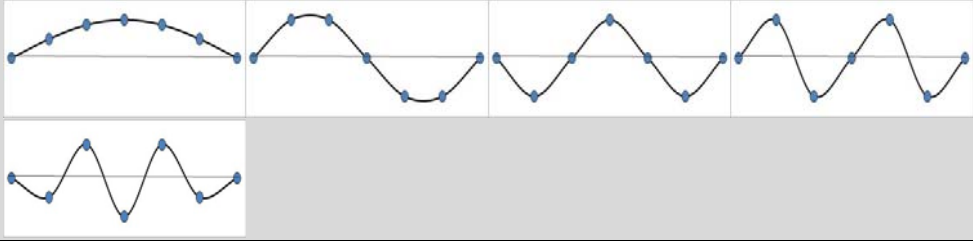


Figura 43. Matrices de rigidez y masa de la viga considerada como ejemplo.

Figura 44. Configuraciones de modos de vibración del sistema estructural considerado como ejemplo.



Anexo C. DETERMINACIÓN DE LA RESPUESTA DINÁMICA POR MEDIO DE OPENSEES DE LAS ESTRUCTURAS ESTUDIADAS

A partir de las referencias dadas para instruirse en el uso de OPENSEES, se asume que el lector se encuentra ya algo familiarizado con el tema.

Viga

Primero se debe crear el archivo de coordenadas nodales:

```
# NodeCoord.tcl
```

```
# Node tag xCrd yCrd
node 1 0.0 0.0
node 2 0.6 0.0
node 3 1.2 0.0
node 4 1.8 0.0
node 5 2.4 0.0
node 6 3.0 0.0
node 7 3.6 0.0
node 8 4.2 0.0
node 9 4.8 0.0
node 10 5.4 0.0
node 11 6.0 0.0
```

Archivo de restricciones:

```
# SPConstraint.tcl
```

```
# SPC tag Dx Dy
fix 1 1 1 0
fix 11 1 1 0
```

Archivo de Secciones Transversales y Definición de Elementos Respectivamente:

```
# Sections.tcl Material Concrete 3000 Psi E= 2.153E+10 Mass per Unit Volume=
2402.7696 Weight per Unit Volume =23563.122
# Section "ElasticDefault": secTag E A Iz
section Elastic 1 2.150000e+010 1.200000e-001 1.600000e-003
section Elastic 2 2.150000e+010 1.200000e-001 1.600000e-003
section Elastic 3 2.150000e+010 1.200000e-001 1.600000e-003
section Elastic 4 2.150000e+010 1.200000e-001 1.600000e-003
section Elastic 5 2.150000e+010 1.200000e-001 1.600000e-003
section Elastic 6 2.150000e+010 1.200000e-001 1.600000e-003
```

```

section Elastic 7 2.150000e+010 1.200000e-001 1.600000e-003
section Elastic 8 2.150000e+010 1.200000e-001 1.600000e-003
section Elastic 9 2.150000e+010 1.200000e-001 1.600000e-003
section Elastic 10 2.150000e+010 1.200000e-001 1.600000e-003

```

```
# Elements.tcl
```

```

#Material Concrete 3000 Psi E= 2.153E+10 Mass per Mass per Unit Length=288.335628
# Element "Element02": eleTag NodeI NodeJ NIP secTag geoTranTag <-mass
massDens>
element dispBeamColumn 1 1 2 10 1 1 -mass +2.883324E+002
element dispBeamColumn 2 2 3 10 2 1 -mass +2.883324E+002
element dispBeamColumn 3 3 4 10 3 1 -mass +2.883324E+002
element dispBeamColumn 4 4 5 10 4 1 -mass +2.883324E+002
element dispBeamColumn 5 5 6 10 5 1 -mass +2.883324E+002
element dispBeamColumn 6 6 7 10 6 1 -mass +2.883324E+002
element dispBeamColumn 7 7 8 10 7 1 -mass +2.883324E+002
element dispBeamColumn 8 8 9 10 8 1 -mass +2.883324E+002
element dispBeamColumn 9 9 10 10 9 1 -mass +2.883324E+002
element dispBeamColumn 10 10 11 10 10 1 -mass +2.883324E+002

```

Tambien se deben establecer archivos como el de transformación geométrica:

```
# GeoTran.tcl
```

```
geomTransf Linear 1
```

y de tipo de análisis que se desea hacer, utilizando que métodos y especificando las variables a ser almacenadas:

```
# Analysis_2_Recorder.tcl
```

```

# Node Recorder "EigenVector": fileName <nodeTag> dof respType
recorder Node -file EigenVec_1.out -time -dof 2 eigen1
recorder Node -file EigenVec_2.out -time -dof 2 eigen2
recorder Node -file EigenVec_3.out -time -dof 2 eigen3
recorder Node -file EigenVec_4.out -time -dof 2 eigen4
recorder Node -file EigenVec_5.out -time -dof 2 eigen5
recorder Node -file EigenVec_6.out -time -dof 2 eigen6
recorder Node -file EigenVec_7.out -time -dof 2 eigen7
recorder Node -file EigenVec_8.out -time -dof 2 eigen8
#recorder Node -file EigenVec_9.out -time -dof 2 eigen9
#recorder Node -file EigenVec_10.out -time -dof 2 eigen10

```

```
# AnalysisOptn_3.tcl
```

```

# AnalysisOptn "EigenDefault": Type: Eigen
# Constraint Handler
constraints Plain
# Convergence Test
test NormUnbalance +1.000000E-006 25 0 2
# Integrator
integrator Newmark +5.000000E-001 +2.500000E-001 +0.000000E+000
+0.000000E+000 +0.000000E+000 +0.000000E+000
# Solution Algorithm
algorithm Newton
# DOF Numberer
numberer RCM
# System of Equations
system ProfileSPD
# Analysis Type
analysis Transient

```

Finalmente se crea un archivo que combine a todos de la siguiente manera:

```

# modelo.tcl

#####
# Analysis-Sequence 1 #
#####

# Start of model generation
# =====

# Create ModelBuilder
# -----
model BasicBuilder -ndm 2 -ndf 3

# Define geometry
# -----
source NodeCoord.tcl

# Define Single Point Constraints
# -----
source SPConstraint.tcl

# Define geometric transformation(s)
# -----
source GeoTran.tcl

source Sections.tcl

```

```

# Define element(s)
# -----
source Elements.tcl

# Start of anaysis generation
# =====

# Get Initial Stiffness
# -----
initialize

# Analysis: EigenDefaultCase
# ++++++

# Define recorder(s)
# -----
source Analysis_2_Recorder.tcl

# Define analysis options
# -----
source AnalysisOptn_3.tcl

set eigFID [open EigenVal.out w]
puts $eigFID [eigen generalized genBandArpack 8]
close $eigFID
analyze 1 0.0001

# Clean up
# -----
wipe

```

Por último para ejecutar el archivo con OPENSEES.exe, simplemente abrimos este programa y escribimos el comando source modelo.tcl, para de esta forma obtener finalmente los eigenvalores y eigenvectores requeridos.

Armadura

```

# NodeCoord.tcl

# Node tag xCrd yCrd

node 1 +0.0 +0.0

```

```
node 2 +1.829 +0.0
node 3 +3.658 +0.0
node 4 +5.486 +0.0
node 5 +7.315 +0.0
node 6 +1.829 +2.438
node 7 +3.658 +2.438
node 8 +5.486 +2.438
```

```
# SPCConstraint.tcl
```

```
# SPC tag Dx Dy
fix 1 1 1
fix 5 1 1
```

```
# NodeMass.tcl
```

```
# Mass tag mx my
mass 1 8.39 8.39
mass 2 15.97 15.97
mass 3 10.45 10.45
mass 4 15.97 15.97
mass 5 8.39 8.39
mass 6 13.1 13.1
mass 7 21.48 21.48
mass 8 13.1 13.1
```

```
# Material "ElasticDefault": matTag E eta
uniaxialMaterial Elastic 1 1.999801e+011 0.000000e+000
uniaxialMaterial Elastic 2 1.423164e+011 0.000000e+000
uniaxialMaterial Elastic 3 1.299929e+011 0.000000e+000
uniaxialMaterial Elastic 4 1.996588e+011 0.000000e+000
uniaxialMaterial Elastic 5 1.961166e+011 0.000000e+000
uniaxialMaterial Elastic 6 1.874766e+011 0.000000e+000
uniaxialMaterial Elastic 7 1.907234e+011 0.000000e+000
uniaxialMaterial Elastic 8 1.777049e+011 0.000000e+000
uniaxialMaterial Elastic 9 1.987063e+011 0.000000e+000
uniaxialMaterial Elastic 10 1.633901e+011 0.000000e+000
uniaxialMaterial Elastic 11 9.547138e+010 0.000000e+000
uniaxialMaterial Elastic 12 1.886431e+011 0.000000e+000
uniaxialMaterial Elastic 13 1.900948e+011 0.000000e+000
```

```
# Elements.tcl
```

```
# Element "Element01": eleTag NodeI NodeJ A matTag
element truss 1 1 2 +4.000000E-004 1
```

```

element truss 2 2 3 +4.000000E-004 2
element truss 3 3 4 +4.000000E-004 3
element truss 4 4 5 +4.000000E-004 4
element truss 5 1 6 +4.000000E-004 5
element truss 6 2 6 +4.000000E-004 6
element truss 7 2 7 +4.000000E-004 7
element truss 8 3 7 +4.000000E-004 8
element truss 9 4 7 +4.000000E-004 9
element truss 10 4 8 +4.000000E-004 10
element truss 11 5 8 +4.000000E-004 11
element truss 12 6 7 +4.000000E-004 12
element truss 13 7 8 +4.000000E-004 13

```

```
# Analysis_2_Recorder.tcl
```

```

# Node Recorder "EigenVector": fileName <nodeTag> dof respType
recorder Node -file EigenVec_1.out -time -dof 1 2 eigen1
recorder Node -file EigenVec_2.out -time -dof 1 2 eigen2
recorder Node -file EigenVec_3.out -time -dof 1 2 eigen3
recorder Node -file EigenVec_4.out -time -dof 1 2 eigen4
recorder Node -file EigenVec_5.out -time -dof 1 2 eigen5
recorder Node -file EigenVec_6.out -time -dof 1 2 eigen6
recorder Node -file EigenVec_7.out -time -dof 1 2 eigen7
recorder Node -file EigenVec_8.out -time -dof 1 2 eigen8
recorder Node -file EigenVec_9.out -time -dof 1 2 eigen9
recorder Node -file EigenVec_10.out -time -dof 1 2 eigen10
recorder Node -file EigenVec_11.out -time -dof 1 2 eigen11
recorder Node -file EigenVec_12.out -time -dof 1 2 eigen12

```

```
# AnalysisOptn_3.tcl
```

```

# AnalysisOptn "EigenDefault": Type: Eigen
# -----
# Constraint Handler
constraints Plain
# Convergence Test
test NormUnbalance +1.000000E-006 25 0 2
# Integrator
integrator Newmark +5.000000E-001 +2.500000E-001 +0.000000E+000
+0.000000E+000 +0.000000E+000 +0.000000E+000
# Solution Algorithm
algorithm Newton
# DOF Numberer
numberer RCM

```

```
# System of Equations
system ProfileSPD
# Analysis Type
analysis Transient
```

```
# modelo.tcl
```

```
#####
# Analysis-Sequence 1 #
#####
```

```
# Start of model generation
# =====
```

```
# Create ModelBuilder
# -----
model BasicBuilder -ndm 2 -ndf 2
```

```
# Define geometry
# -----
source NodeCoord.tcl
```

```
# Define Single Point Constraints
# -----
source SPConstraint.tcl
```

```
# Define nodal masses
# -----
source NodeMass.tcl
```

```
# Define Multi Point Constraints
# -----
source MPCConstraint.tcl
```

```
# Define material(s)
# -----
source Materials.tcl
```

```
# Define section(s)
# -----
#source Sections.tcl
```

```
# Define geometric transformation(s)
```

```

# -----
source GeoTran.tcl

# Define element(s)
# -----
source Elements.tcl

# Define time series
# -----
source TimeSeries.tcl

# Start of anaysis generation
# =====

# Get Initial Stiffness
# -----
initialize

# Analysis: EigenDefaultCase
# ++++++
# Define recorder(s)
# -----
source Analysis_2_Recorder.tcl

# Define analysis options
# -----
source AnalysisOptn_3.tcl

set eigFID [open EigenVal.out w]
puts $eigFID [eigen generalized fullGenLapack 12]
close $eigFID
analyze 1 0.0001

# Clean up
# -----
wipe

```

Anexo D. ARCHIVOS MATLAB

a. Viga

Algoritmo de Inclusión de Daño Aleatorio:

```
cd 'D:\Mechanical Engineering\UIS\PDG Deteccion de Daños en Estructuras
Utilizando PSO\OpenSees2.1.0.exe\Modelo Viga masa distribuida'
clear all
% Escritura de datos
NElemeD = 2; % Numero de elementos dañados
N = 10; % Numero de elementos
con = 1;
Ei = 2.153E10;

if NElemeD ==0
    Dano = ones(1,N);
    Elem = [1:1:N];
else
while con-1<NElemeD && con-1 < N

    Elem(con) = round(rand(1)*(N-1)+1);

    false = sum(Elem(con) == Elem(1:con));

    if false==1
        Dano(con) = random('unif',0.4,1,1);
        if Dano(con)<=0.05
            Dano(con) = 0.05;
        end
        con = con+1;
    else
        Elem(con) = [];
    end

end

end

fila = length(Elem);
```

```

for i=1:N

    for j=1:fila
        if i ~= Elem(j)
            E(1,i)= Ei;
        else
            E(1,i)=Dano(j)*Ei;
            break
        end
    end

end

[filas,nmat] = size(E);

A = 0.12;
I = 0.0016;

fid = fopen('Sections.tcl', 'wt');

fprintf(fid, '# Sections.tcl Material Concrete 3000 Psi E= 2.153E+10 Mass per Unit
Volume= 2402.7696 Weight per Unit Volume =23563.122');
fprintf(fid, '\n');

fprintf(fid, '# Section "ElasticDefault":  secTag  E  A  Iz ');
fprintf(fid, '\n');

for i=1:nmat
    fprintf(fid, 'section Elastic %1.0f %12.6e %12.6e %12.6e', i, E(i), A, I);
    fprintf(fid, '\n');
end

fclose(fid);

!opensees modelo.tcl

%Lectura de datos.

%Frecuencias

Eigen = load('EigenVal.out');
Wne = sqrt(Eigen);
Modos=length(Wne);
Wne = sort(Wne);

```

```

Ruido1 = 0;
Ruido2 = 0;
for i = 1:Modos

ru = round(random('unif',-1,1,1,1));
if ru ==0
ru = -1;
end
ruido(i,1) = ru*Ruido1;
ruido(i,2) = ru*Ruido2;
end

Wne = Wne + Wne.*ruido(:,1);
T = 2*pi./(Wne);

% Modos
for i=1:Modos
EigenVece(i,:) = load(['EigenVec_',num2str(i),'.out'])+
load(['EigenVec_',num2str(i),'.out']).*ruido(i,2);
end

Dano = 1-Dano;
EigenVece(:,[1 2 12]) = [];
save ('EigenVece', 'EigenVece')
save ('Wne', 'Wne')
save ('Elem', 'Elem')
save ('Dano', 'Dano')
save ('E', 'E')

```

Algoritmo de Inclusión de Ruido en las Mediciones

```

cd 'D:\Mechanical Engineering\UIS\PDG Deteccion de Daños en Estructuras
Utilizando PSO\OpenSees2.1.0.exe\Modelo Armadura v2.0'
clear all

```

```

load Wne
load EigenVece
load Elem
load Dano
load E

```

```

Modos=length(Wne);

```

```

%Porcentaje de Ruido introducido

```

```

Ruido1 = 0.2;
Ruido2 = 5;
for i = 1:Modos

ru = round(random('unif',-1,1,1,1));
if ru ==0
ru = -1;
end
ruido(i,1) = ru*(Ruido1/100);
ruido(i,2) = ru*(Ruido2/100);
end

Wne = Wne + Wne.*ruido(:,1);
T = 2*pi./(Wne);

% Modos
for i=1:Modos
EigenVece(i,:) = EigenVece(i,.)+ EigenVece(i,).*ruido(i,2);
end

save ('EigenVece', 'EigenVece')
save ('Wne', 'Wne')
save ('Elem','Elem')
save ('Dano','Dano')
save ('E','E')

```

Funcion Objetivo MAC

```

function [out] = calibraV(in)

cd 'D:\Mechanical Engineering\UIS\PDG Deteccion de Daños en Estructuras
Utilizando PSO\OpenSees2.1.0.exe\Modelo Viga masa distribuida'
% Escritura de datos

load Wne
load EigenVece

E(:,.) = 1e10*in(:,.);

[filas,nmat] = size(E);
clear out

```

```

%nmat = 6;

for j=1:fila

clear EigenVec Eigen

A = 0.12;
I = 0.0016;

fid = fopen('Sections.tcl', 'wt');

fprintf(fid, '# Sections.tcl Material Concrete 3000 Psi E= 2.153E+10 Mass per Unit
Volume= 2402.7696 Weight per Unit Volume =23563.122');
fprintf(fid, '\n');

fprintf(fid, '# Section "ElasticDefault":  secTag  E  A  Iz ');
fprintf(fid, '\n');

for i=1:nmat
fprintf(fid, 'section Elastic %1.0f %12.6e %12.6e %12.6e', i, E(j,i), A, I);
fprintf(fid, '\n');
end

fclose(fid);

!opensees modelo.tcl

clc
%Lectura de datos.

%Frecuencias

Eigen = load('EigenVal.out');
Wn = sqrt(Eigen);
%Wn = sort(Wn);
T(:,j) = 2*pi./(Wn);

Modos=8; %length(Wn);

% Modos
for i=1:Modos
EigenVec(i,:) = load(['EigenVec_', num2str(i), '.out']);
end

```

```

EigenVec(:,[1 2 12]) = [];

MAC = 0;

for i = 1:Modos

MAC = MAC+ (1-(EigenVec(i,:)*EigenVece(i,:))^2
/((EigenVec(i,:)*EigenVec(i,:))*(EigenVece(i,:)*EigenVece(i,:)));

end

f = sum(abs(Wn(1,1:Modos)-Wne(1,1:Modos))./Wne(1,1:Modos));

out(j,1) = (f+MAC);

min(out)

end

end

```

Funcion Objetivo 1 (Ec. 4.4)

```

function [out] = calibraV1(in)

cd 'D:\Mechanical Engineering\UIS\PDG Deteccion de Daños en Estructuras
Utilizando PSO\OpenSees2.1.0.exe\Modelo Viga masa distribuida'
% Escritura de datos

load Wne
load EigenVece

E(:,:) = 1e10*in(:,:);

[filas,nmat] = size(E);
clear out
%nmat = 6;

for j=1:filas

clear EigenVec Eigen

A = 0.12;
I = 0.0016;

```

```

fid = fopen('Sections.tcl', 'wt');

fprintf(fid, '# Sections.tcl Material Concrete 3000 Psi E= 2.153E+10 Mass per Unit
Volume= 2402.7696 Weight per Unit Volume =23563.122');
fprintf(fid, '\n');

fprintf(fid, '# Section "ElasticDefault":  secTag  E  A  Iz ');
fprintf(fid, '\n');

for i=1:nmat
fprintf(fid, 'section Elastic %1.0f %12.6e %12.6e %12.6e', i, E(j,i), A, I);
fprintf(fid, '\n');
end

fclose(fid);

!opensees modelo.tcl

clc
%Lectura de datos.

%Frecuencias

Eigen = load('EigenVal.out');
Wn = sqrt(Eigen);
Wn = sort(Wn);
T(:,j) = 2*pi./(Wn);

Modos=10; %length(Wn);

% Modos
for i=1:Modos

EigenVec(i,:) = load(['EigenVec_', num2str(i), '.out']);

end

EigenVec(:, [1 2 12]) = [];

MA =0;

for i =1:Modos

MA1 = sum(abs(EigenVec(i,:)-EigenVece(i,:)));

```

```
MA = MA1 + MA;
```

```
end
```

```
f = sum(1-(Wn(1,1:Modos))./Wne(1,1:Modos))^2;
```

```
out(j,1) = (f+MA);  
min(out)
```

```
end
```

```
end
```

b. Armadura

Algoritmo de Inclusion de Daño

```
cd 'D:\Mechanical Engineering\UIS\PDG Deteccion de Daños en Estructuras  
Utilizando PSO\OpenSees2.1.0.exe\Modelo Armadura v2.0'
```

```
clear all
```

```
% Escritura de datos
```

```
NElemeD = 2; % Numero de elementos dañados
```

```
N = 13; % Numero de elementos
```

```
con = 1;
```

```
Ei = 2.0E11;
```

```
if NElemeD ==0
```

```
    Dano = ones(1,N);
```

```
    Elem = [1:1:N];
```

```
else
```

```
while con-1<NElemeD && con-1 < N
```

```
    Elem(con) = round(rand(1)*(N-1)+1);
```

```
    false = sum(Elem(con) == Elem(1:con));
```

```
    if false==1
```

```
        Dano(con) = random('unif',0.4,1,1);
```

```
        if Dano(con)<=0.05
```

```
            Dano(con) = 0.05;
```

```
        end
```

```
        con = con+1;
```

```
    else
```

```
        Elem(con) = [];
```

```
    end
```

```

end

end

fila = length(Elem);

for i=1:N

    for j=1:fila
        if i ~= Elem(j)
            E(1,i)= Ei;
        else
            E(1,i)=Dano(j)*Ei;
            break
        end
    end

end

[filas,nmat] = size(E);

A = 4.000000E-004;
eta = 0;

fid = fopen('Materials.tcl', 'wt');

fprintf(fid,'# Material "ElasticDefault":  matTag  E  eta ');
fprintf(fid,'\n');

for i=1:nmat
    fprintf(fid,'uniaxialMaterial Elastic %1.0f %12.6e %12.6e',i,E(i),eta);
    fprintf(fid,'\n');
end

fclose(fid);

!opensees modelo.tcl

%Lectura de datos.

%Frecuencias

Eigen = load('EigenVal.out');
Wne = sqrt(Eigen);

```

```

Modos=length(Wne);
Wne = sort(Wne);
Ruido1 = 0;
Ruido2 = 0;
for i = 1:Modos

ru = round(random('unif',-1,1,1,1));
if ru ==0
ru = -1;
end
ruido(i,1) = ru*Ruido1;
ruido(i,2) = ru*Ruido2;
end

Wne = Wne + Wne.*ruido(:,1);
T = 2*pi./(Wne);

% Modos
for i=1:Modos
EigenVece(i,:) = load(['EigenVec_',num2str(i),'.out'])+
load(['EigenVec_',num2str(i),'.out']).*ruido(i,2);
end

Dano = 1-Dano;
EigenVece(:,[1 2 3 10 11]) = [];
EigenVece = EigenVece';
M = zeros(12,12);
M(1,1) = 15.97;
M(2,2) = 15.97;
M(3,3) = 10.45;
M(4,4) = 10.45;
M(5,5) = 15.97;
M(6,6) = 15.97;
M(7,7) = 13.1;
M(8,8) = 13.1;
M(9,9) = 21.48;
M(10,10) = 21.48;
M(11,11) = 13.1;
M(12,12) = 13.1;

a = real(sqrt(inv(EigenVece'*M*EigenVece)));

for ii=1:12

```

```

for jj=1:12
  ModosN(jj,ii) = EigenVece(jj,ii)*a(ii,ii);
end
end

```

```
Mm = ModosN'*M*ModosN;
```

```
EigenVece = ModosN';
```

```

save ('EigenVece', 'EigenVece')
save ('Wne', 'Wne')
save ('Elem', 'Elem')
save ('Dano', 'Dano')
save ('E', 'E')

```

Algoritmo de Inclusión de Ruido en las Mediciones

```

cd 'D:\Mechanical Engineering\UIS\PDG Deteccion de Daños en Estructuras
Utilizando PSO\OpenSees2.1.0.exe\Modelo Armadura v2.0'
clear all

```

```
clear all
```

```

load Wne
load EigenVece
load Elem
load Dano
load E

```

```
Modos=length(Wne);
```

```
%Porcentaje de Ruido introducido
```

```

Ruido1 = 0.2;
Ruido2 = 5;
for i = 1:Modos

```

```

  ru = round(random('unif',-1,1,1,1));
  if ru ==0
    ru = -1;
  end
  ruido(i,1) = ru*(Ruido1/100);
  ruido(i,2) = ru*(Ruido2/100);
end

```

```
Wne = Wne + Wne.*ruido(:,1);  
T = 2*pi./(Wne);
```

```
% Modos
```

```
for i=1:Modos  
EigenVece(i,:) = EigenVece(i,:)+ EigenVece(i,:).*ruido(i,2);  
end
```

```
save ('EigenVece', 'EigenVece')  
save ('Wne', 'Wne')  
save ('Elem', 'Elem')  
save ('Dano', 'Dano')  
save ('E', 'E')
```

Función Objetivo calibraV_A

```
function [out] = calibraV_A(in)
```

```
cd 'C:\Ivan Maestria\Libro Tesis Maestria\PDG Deteccion de Daños en Estructuras  
Utilizando PSO\OpenSees2.1.0.exe\Modelo Armadura'
```

```
% Escritura de datos
```

```
load Wne  
load EigenVece
```

```
E(:, :) = 1e10*in(:, :);
```

```
[fila, nmat] = size(E);  
clear out  
%nmat = 6;
```

```
for j=1:fila
```

```
clear EigenVec Eigen
```

```
clear EigenVec Eigen
```

```
A = 4.000000E-004;  
eta= 0;
```

```
fid = fopen('Materials.tcl', 'wt');
```

```

fprintf(fid,'# Material "ElasticDefault":  matTag  E  eta ');
fprintf(fid,'\n');

for i=1:nmat
fprintf(fid,'uniaxialMaterial Elastic %1.0f %12.6e %12.6e',i,E(j,i),eta);
fprintf(fid,'\n');
end

fclose(fid);

!opensees modelo.tcl

clc
%Lectura de datos.
%Frecuencias

Eigen = load('EigenVal.out');
Wn = sqrt(Eigen);
%Wn = sort(Wn);
T(:,j) = 2*pi./(Wn);

Modos=length(Wn);

% Modos
for i=1:Modos
EigenVec(i,:) = load(['EigenVec_',num2str(i),'.out']);
end

EigenVec(:,[1 2 3 10 11]) = [];

EigenVec = EigenVec';
M = zeros(12,12);
M(1,1) = 15.97;
M(2,2) = 15.97;
M(3,3) = 10.45;
M(4,4) = 10.45;
M(5,5) = 15.97;
M(6,6) = 15.97;
M(7,7) = 13.1;
M(8,8) = 13.1;
M(9,9) = 21.48;
M(10,10) = 21.48;
M(11,11) = 13.1;
M(12,12) = 13.1;

```

```
a = real(sqrt(inv(EigenVec'*M*EigenVec)));
```

```
for ii=1:12  
    for jj=1:12  
        ModosN(jj,ii) = EigenVec(jj,ii)*a(ii,ii);  
    end  
end
```

```
Mm = ModosN'*M*ModosN;
```

```
EigenVec = ModosN';
```

```
MAC =0;  
Modos=10; %length(Wn);
```

```
for i =1:Modos
```

```
MAC = MAC+ (1-(EigenVec(i,:)*EigenVece(i,:))'^2  
/((EigenVec(i,:)*EigenVec(i,:))*(EigenVece(i,:)*EigenVece(i,:))));
```

```
end
```

```
f = sum(abs(Wn(1,1:Modos)-Wne(1,1:Modos))./Wne(1,1:Modos));
```

```
out(j,1) = (f+MAC);
```

```
min(out)
```

```
end
```

```
end
```

Función Objetivo calibraV1

```
function [out] = calibraV1_A2(in)
```

```
cd 'D:\Mechanical Engineering\UIS\PDG Deteccion de Daños en Estructuras  
Utilizando PSO\OpenSees2.1.0.exe\Modelo Armadura v2.0'
```

```
% Escritura de datos
```

```
load Wne
```

```
load EigenVece
```

```

E(:, :) = 1e11*in(:, :);

[filas, nmat] = size(E);
clear out
%nmat = 6;

for j=1:filas

    clear EigenVec Eigen

    A = 4.000000E-004;
    eta = 0;

    fid = fopen('Materials.tcl', 'wt');

    fprintf(fid, '# Material "ElasticDefault":  matTag  E  eta ');
    fprintf(fid, '\n');

    for i=1:nmat
        fprintf(fid, 'uniaxialMaterial Elastic %1.0f %12.6e %12.6e', i, E(j, i), eta);
        fprintf(fid, '\n');
    end

    fclose(fid);

!opensees modelo.tcl

clc
%Lectura de datos.

%Frecuencias

Eigen = load('EigenVal.out');
Wn = sqrt(Eigen);
Wn = sort(Wn);
T(:, j) = 2*pi./(Wn);

Modos=length(Wn);
% Modos
for i=1:Modos

    EigenVec(i, :) = load(['EigenVec_', num2str(i), '.out']);

```

```

end

EigenVec(:,[1 2 3 10 11]) = [];

EigenVec = EigenVec';
M = zeros(12,12);
M(1,1) = 15.97;
M(2,2) = 15.97;
M(3,3) = 10.45;
M(4,4) = 10.45;
M(5,5) = 15.97;
M(6,6) = 15.97;
M(7,7) = 13.1;
M(8,8) = 13.1;
M(9,9) = 21.48;
M(10,10) = 21.48;
M(11,11) = 13.1;
M(12,12) = 13.1;

a = real(sqrt(inv(EigenVec'*M*EigenVec)));

for ii=1:12
    for jj=1:12
        ModosN(jj,ii) = EigenVec(jj,ii)*a(ii,ii);
    end
end

Mm = ModosN'*M*ModosN;

EigenVec = ModosN';

MA = 0;
Modos=4; %length(Wn);

for i = 1:Modos

MA1 = sum(abs(EigenVec(i,:)-EigenVece(i,:)));
MA = MA1 + MA;

end

f = sum(1-(Wn(1,1:Modos))./Wne(1,1:Modos))^2;

```

```
out(j,1) = (f+MA);
```

```
min(out)
```

```
end
```

```
end
```