

**DETECCIÓN DE DAÑOS EN ESTRUCTURAS UTILIZANDO LA TÉCNICA DE
OPTIMIZACIÓN ALGORITMOS GENÉTICOS**

**ROBINSON BARRERA NÚÑEZ
OSCAR MAURICIO SOTO DÍAZ**

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

2010

**DETECCIÓN DE DAÑOS EN ESTRUCTURAS UTILIZANDO LA TÉCNICA DE
OPTIMIZACIÓN ALGORITMOS GENÉTICOS**

**ROBINSON BARRERA NÚÑEZ
OSCAR MAURICIO SOTO DÍAZ**

**Trabajo de grado para optar al título de
Ingeniero Mecánico**

Directores:

**JABID QUIROGA
Magister, Ingeniero Mecánico**

**IVÁN GÓMEZ
Magister, Ingeniero Civil**

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

2010

DEDICATORIA

ROBINSON BARRERA NÚÑEZ

A mis padres por su apoyo incondicional y su gran esfuerzo, el cual me inspiro para superar cada obstáculo que se me presentó en la vida.

A mi novia por su comprensión y por animarme a culminar con éxito esta etapa de mi vida.

A mis hermanos que con su confianza me empujaron cada día a luchar por mis sueños.

DEDICATORIA

OSCAR MAURICIO SOTO DÍAZ

A mi padre ALVARO SOTO BELTRAN y a mi madre YOLANDA DÍAZ PEÑA por el amor, la excelente educación que me brindaron, los consejos, el empuje que me que han inculcado, el apoyo incondicional y la gran dedicación en todos estos años.

A mi hermano ha sido un honor haber crecido y compartir tan buenos momentos al lado tuyo.

A mi novia por tu dedicación, paciencia y comprensión.

A mis amigos por los grandes momentos compartidos en esta etapa de mi vida.

AGRADECIMIENTOS

A MSc. Jabid Quiroga, director del proyecto por su colaboración y aportar la dirección que dio como resultado un proyecto con un nuevo enfoque.

A MSc. Iván Gómez, director del proyecto por su gran disposición en cada momento y aportar su punto de vista para poder superar cada obstáculo que se presento.

Al Ingeniero Héctor Navarro por su colaboración en el inicio del proyecto.

A la Universidad Industrial de Santander, que nos permitió formarnos como nuevos profesionales que aportaran en el futuro de nuestro país.

A la escuela de Geología que nos brindo su apoyo tecnológico para poder culminar con el proyecto.

CONTENIDO

1. INTRODUCCIÓN	23
1.1. ANTECEDENTES	25
1.2. ALCANCE DEL PROYECTO	27
1.2.1 Objetivo General	27
1.2.2. Objetivos Específicos	28
2. REVISIÓN BIBLIOGRÁFICA	30
2.1. ANÁLISIS DINÁMICO	30
2.1.1. FORMULACIÓN CLÁSICA: Sistemas de múltiples grados de libertad (MDOF)	32
2.1.2. DETECCIÓN DE DAÑO	35
2.2. METODOLOGÍA EMPLEADA PARA LA LOCALIZACIÓN DE DAÑO	37
2.2.1. LA DETECCIÓN DE DAÑOS COMO UN PROCESO DE OPTIMIZACIÓN	37
2.3. FUNCIÓN OBJETIVO	38
3. OPTIMIZACIÓN CON ALGORITMOS GENÉTICOS	41
3.1. INTRODUCCIÓN A LOS ALGORITMOS GENÉTICOS	42
3.2. ALGORITMOS GENÉTICOS EN LA OPTIMIZACIÓN	44
3.2.1. APLICACIONES DE LOS ALGORITMOS GENÉTICOS	46
3.3. ALGORITMOS GENÉTICOS	48
3.3.1. GENERACIÓN	48
3.3.2. POBLACIÓN INICIAL	49
3.3.3. FUNCIÓN OBJETIVO	49
3.3.4. SELECCIÓN	50
3.3.4.1. RULETA O SELECCIÓN PROPORCIONAL	50
3.3.4.2. SELECCIÓN POR RANKING	51
3.3.4.3. SELECCIÓN POR TORNEO	51
3.3.5. CRUZAMIENTO	52

3.3.5.1. CRUCE DE UN PUNTO	52
3.3.5.2. CRUCE DE N PUNTO	52
3.3.5.3. CRUCE HEURÍSTICO	52
3.3.5.4. CRUCE UNIFORME	53
3.3.5.5. CRUCES PARA PERMUTACIÓN	53
3.3.6. MUTACIÓN	54
3.3.7. INSERCIÓN ELITISTA	55
3.3.8. CRITERIOS DE TERMINACIÓN DEL ALGORITMO GENÉTICO	55
3.3.8.1. CRITERIO DE CONVERGENCIA DE IDENTIDAD	55
3.3.8.2. CRITERIO DE CONVERGENCIA DE APTITUD	56
3.3.8.3. CRITERIO DE CANTIDAD DE GENERACIONES	56
3.3.9. RUIDO EN MEDICIONES	57
3.4. ESTRUCTURA PARA ENTORNO MATLAB CON ALGORITMOS GENÉTICOS	58
4. PROCEDIMIENTO GENERAL	60
4.1. METODOLOGÍA	60
4.2. ESTRUCTURAS ANALIZADAS	63
4.3. ESCENARIOS DE DAÑO	66
4.3.1 INCLUSIÓN DE RUIDO EN MEDICIONES	67
4.3.2 CASOS DE DAÑO	69
4.4. CONFIABILIDAD DE LOS VALORES OBTENIDOS POR OPENSEES UTILIZANDO COMO COMPROBACIÓN SAP 2000	70
4.4.1 EJECUCIÓN DE LA METODOLOGÍA EMPLEADA PARA LA TOMA DE RESULTADOS	77
5. ANÁLISIS DE RESULTADOS	80
5.1. IDENTIFICACIÓN DE ELEMENTOS DAÑADOS	81
5.1.1 Función objetivo calibraV1 (ec. 2.10)	81
5.1.2. Función objetivo calibraMAC (ec.2.9)	92

5.1.3 Comparación de resultados con la función objetivo calibraMAC (ec.2.9) para la ARMADURA con algoritmos genéticos y calibraV1 (ec. 2.10) para PSO.	96
5.2. ANÁLISIS DE RESULTADOS ESTRUCTURA TIPO VIGA	102
5.2.1. Mejor resultado para daño simple en la estructura tipo viga con la función objetivo calibraV1:	103
5.2.2. Mejor resultado para daño múltiple 2 elementos dañados en la estructura tipo viga con la función objetivo calibraV1:	104
5.2.3. Mejor resultado para daño múltiple 4 elementos dañados en la estructura tipo viga con la función objetivo calibraV1:	105
5.3. ANÁLISIS DE RESULTADOS ESTRUCTURA TIPO ARMADURA	108
5.3.1. Mejor resultado para daño simple en la estructura tipo armadura con la función objetivo calibraMAC:	109
5.3.1. Mejor resultado para daño múltiple 2 elementos dañados en la estructura tipo armadura con la función objetivo calibraMAC:	110
5.4. ANÁLISIS DE TIEMPO DE RESPUESTA PARA LAS TÉCNICAS ALGORITMOS GENÉTICOS Vs PSO.	113
6. CONCLUSIONES	115
6.1. SUGERENCIAS PARA TRABAJOS FUTUROS	116
REFERENCIAS	118
BIBLIOGRAFÍA	120
APÉNDICES	122

LISTA DE TABLAS

Tabla 4.1. Conectividad de los elementos en la estructura tipo Viga.	65
Tabla 4.2 Conectividad de los elementos en la estructura tipo Armadura.	66
Tabla 4.3. Configuraciones de ruido que serán estudiadas (a) Viga y (b) Armadura	68
Tabla 4.4. Escenarios de daño estudiados para: (a) Viga y (b) Armadura.	69
Tabla 4.5. Resultados de frecuencias y periodos OPENSEES.	71
Tabla 5.1. Parámetros escogidos para la estructura tipo viga en el algoritmo genético.	102
Tabla 5.2. Comparación de los diferentes episodios de daño y ruido de los resultados obtenidos con el algoritmo genético para la estructura tipo viga.	106
Tabla 5.3. Comparación de los diferentes episodios de daño y ruido de los resultados obtenidos con PSO para la estructura tipo viga.	107
Tabla 5.4. Parámetros escogidos para la estructura tipo armadura en el algoritmo genético.	108
Tabla 5.5. Comparación de los diferentes episodios de daño y ruido de los resultados obtenidos con el algoritmo genético para la estructura tipo armadura.	111
Tabla 5.6. Comparación de los diferentes episodios de daño y ruido de los resultados obtenidos con PSO para la estructura tipo armadura.	112
Tabla 5.7. Comparación de los diferentes tiempos de ejecución entre AG y PSO para la estructura tipo viga.	113
Tabla 5.8. Comparación de los diferentes tiempos de ejecución entre AG y PSO para la estructura tipo armadura.	114

LISTA DE FIGURAS

Figura 2.1. Modos de vibración de una estructura de 2 grados de libertad (CHOPRA, 2001)	30
Figura 3.1. Esquema de funcionamiento del algoritmo genético.	46
Figura 3.2. Estructura de operación del algoritmo genético (AG).	59
Figura 4.1. Diagrama de flujo del sistema.	62
Figura 4.2. Viga simplemente apoyada.	63
Figura 4.3. Armadura de 13 elementos simplemente apoyada.	64
Figura 4.4. Modelo viga usando SAP2000	70
Figura 4.5. Valores de periodos y frecuencias hallado por SAP2000 para la viga.	72
Figura 4.6. Modos de Vibración 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	73
Figura 4.7. Modelo armadura usando SAP2000.	74
Figura 4.8. Modos de Vibración 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.	76
Figura 4.9. Interfaz Gráfica para el caso de la viga con la función objetivo CalibraV1viga.m implementada en Algoritmos Genéticos para el entorno MATLAB.	78
Figura 4.10. Interfaz Gráfica para el caso de la armadura con la función objetivo CalibraMACarmadura.m implementada en Algoritmos Genéticos para el entorno MATLAB.	79
Figura. 5.1.	81
Figura. 5.2.	82
Figura. 5.3.	82
Figura. 5.4.	83
Figura. 5.5.	83
Figura. 5.6.	83

Figura. 5.7.	84
Figura. 5.8.	85
Figura. 5.9.	85
Figura. 5.10.	85
Figura. 5.11.	86
Figura. 5.12.	86
Figura. 5.13.	87
Figura. 5.14.	87
Figura. 5.15.	88
Figura. 5.17.	89
Figura. 5.18.	90
Figura. 5.19.	90
Figura. 5.20.	91
Figura. 5.21.	91
Figura. 5.22.	91
Figura. 5.23.	92
Figura. 5.24.	92
Figura. 5.25.	93
Figura. 5.26.	93
Figura. 5.27.	94
Figura. 5.28.	94
Figura. 5.20.	95
Figura. 5.30.	95
Figura. 5.31.	96
Figura. 5.32.	96
Figura. 5.33.	97
Figura. 5.34.	97
Figura. 5.35.	98

Figura. 5.36.	98
Figura. 5.37.	99
Figura. 5.38.	99
Figura. 5.39.	100
Figura. 5.40.	100
Figura. 5.41.	101
Figura. 5.42.	101
Figura. 5.43. Daño simple en la estructura tipo Viga.	103
Figura. 5.44. Daño múltiple 2 elementos en la estructura tipo Viga.	104
Figura. 5.45. Daño múltiple 4 elementos en la estructura tipo Viga.	105
Figura. 5.46. Daño simple en la estructura tipo Armadura.	109
Figura. 5.47. Daño múltiple 2 elementos en la estructura tipo Armadura.	110

LISTA DE ANEXOS

A.1. EJEMPLO DE ANÁLISIS MODAL	122
A.2. DETERMINACIÓN DE LA RESPUESTA DINÁMICA POR MEDIO DE OPENSEES DE LAS ESTRUCTURAS ESTUDIADAS	125
A.3. ARCHIVOS MATLAB	137

RESUMEN

TÍTULO: DETECCIÓN DE DAÑOS EN ESTRUCTURAS UTILIZANDO LA TÉCNICA DE ALGORITMOS GENÉTICOS (AG)*

AUTOR: ROBINSON BARRERA NÚÑEZ
OSCAR MAURICIO SOTO DÍAZ**

PALABRAS CLAVE:

Parámetros Dinámicos, Optimización por algoritmos genéticos, PSO, Metaheurísticas, Detección de daños, Elementos Finitos.

RESUMEN:

En el campo de la ingeniería se busca en el día a día tener mejores resultados con menores costos informáticos y económicos para la detección de fallas en estructuras, siendo la salud estructural uno de los campos de investigación de nuevos métodos para la detección de fallas mediante nuevas técnicas de inteligencia artificial y más precisamente el campo de los algoritmos evolutivos, una de las herramientas para estas nuevas técnicas en la investigaciones de la salud estructural. En el presente trabajo se busca comparar la técnica de optimización enjambre de partículas (PSO) con la técnica de optimización algoritmos genéticos (AG) mediante diferentes episodios de daños (daño simple y daño múltiple) con y sin ruido, para poder realizar la detección del daño se compararon los resultados con y sin daño de los parámetros de la estructura.

Para determinar los parámetros dinámicos de la estructura se uso la herramienta de libre distribución OPENSEES mediante el modelo de elementos finitos de esta. Se analizaron las estructuras con diferentes formas de vibración, numero de daños y ruidos mediante la implementación del análisis de la salud estructural mediante algoritmos genéticos en todas las situaciones mencionadas. Esta metodología se implemento a una estructura viga discretizada en 10 elementos y unas armadura de 13 elementos; En los resultados obtenidos se muestra el buen desempeño de la técnica AG frente a la técnica de PSO para episodios de daño y ruido promedio en estas estructuras, teniendo en cuenta mejores episodios en la detección, diagnostico, falsos positivos y tiempos de respuesta para cada una de las técnicas de optimización.

* Proyecto de Grado

** Facultad de Ingenierías Físicomecánicas, Escuela de Ingeniería Mecánica. Directores: Jabid Quiroga, Ivan Gómez.

ABSTRACT

TITLE: STRUCTURAL DAMAGE DETECTION USING GENETICS ALGORITHMS (AG)^{*}

AUTHOR: ROBINSON BARRERA NÚÑEZ
OSCAR MAURICIO SOTO DÍAZ ^{**}

KEYWORDS:

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

ABSTRACT:

In the field of engineering is looking at day to day have better results with lower costs and financial software to detect flaws in structures, structural health being one of the areas of research of new methods for fault detection using new artificial intelligence techniques and more specifically the field of evolutionary algorithms, one of the tools for these new techniques in structural health research. In the present study sought to compare the technique of particle swarm optimization (PSO) with the optimization technique genetic algorithms (GA) through different episodes of damage (damage and damage multiple simple) with and without noise, to perform detection damage compared the results with and without damage to the structure parameters.

To determine the dynamic parameters of the structure will use the freeware tool OPENS by finite element model of this. Structures were analyzed with different forms of vibration, noise and number of damage through the implementation of structural health analysis using genetic algorithms in all the above situations. This methodology was implemented to a beam structure discretized into 10 elements and a reinforcement of 13 elements, in the results obtained shows the good performance of the technical AG against PSO technique for severe damage and average noise in these structures, taking account best episodes in the detection, diagnosis, false positives and response times for each of the optimization techniques.

^{*} Undergraduate Degree Work

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

DESCRIPCIÓN DEL DOCUMENTO

El presente trabajo tiene como objetivo principal incentivar a la investigación de nuevas técnicas para el uso de la inteligencia artificial y así mejorar los análisis en cuanto a la detección de fallas, dando pautas generales para incentivar investigaciones futuras que mejoren los avances tecnológicos que marcan las nuevas tendencias en la aplicación de nuevas herramientas para resolver los problemas existentes y los venideros.

Para las investigaciones en la detección de fallas se hace primordial manejar algoritmos evolutivos que nos ayuden con la optimización de la función objetivo, siendo así como entra la técnica de algoritmos genéticos (AG) a jugar un papel muy importante, ya que es simple de manipular y tiene resultados dentro de los parámetros medios muy cercanos a lo esperado con otras técnicas implementadas en este campo de la detección de fallas en estructuras, con carácter continuo o discreto, con diferentes grados de dificultad, lo cual nos permite tener una mejor idea de la técnica a implementar en un caso real. En el presente trabajo se muestra la gran influencia del monitoreo de estructuras en el campo de la ingeniería, tales como detección de fallas para determinar el estado de las partes en ingeniería mecánica, salud estructural en ingeniería civil; lo cual marca una gran área de impacto en los campos concernientes a cada ingeniería. Para el desarrollo de este proyecto se requirió el estudio de las propiedades dinámicas de las estructuras, la aplicación de técnicas en la detección de daños, algoritmos evolutivos y la utilización de herramientas informáticas para el procesamiento de la información requerida en el análisis. El objetivo principal es motivar al lector para que se interese en los temas relacionados con la inteligencia artificial y más propiamente la detección de fallas mediante algoritmos evolutivos, también se aclara que estos métodos se pueden aplicar a diferentes problemáticas, tales

como detección de fallas en piezas mecánicas, para lo cual se debe montar el estudio y la metodología requerida ya que de esto dependen los resultados obtenidos. Pensando en la comodidad lectora para la fácil interpretación de la temática utilizada se dividió el trabajo en capítulos, anexos y referencias, para que aquellas personas interesadas en el tema puedan obtener la información requerida. A continuación se presentan los capítulos a tratar:

Capítulo 1: Muestra un enfoque general del proyecto, el cómo y porque se realizó este trabajo.

Capítulo 2: Tiene como objetivo mostrar los conceptos utilizados en el trabajo, dando mayor importancia a los temas como propiedades dinámicas de las estructuras y detección de daños en estructuras, utilizando métodos de optimización para ello.

Capítulo 3: Permite que el lector tenga una idea de la técnica de optimización algoritmos genéticos (AG) utilizada para detectar daños; también se identifica la forma en la que el AG funciona para realizar la optimización de la función objetivo.

Capítulo 4: Presenta la metodología que se utilizó para la detección de daño en las estructuras. Se muestran las estructuras a las cuales se realizó el análisis, describiendo la discretización de cada una de estas estructuras y los episodios de ruido a los que fueron expuestas; también se presenta la forma en la que se obtuvieron las propiedades dinámica y su corroboración mediante el software SAP 2000.

Capítulo 5: Se muestran los resultados obtenidos por la técnica de optimización algoritmos genéticos y estos son comparados con los resultados que se

obtuvieron con la técnica de optimización PSO y se clasifican en tablas para identificar el comportamiento en un enfoque general y determinar cual técnica presenta un mejor comportamiento.

Capítulo 6: Se presentan las conclusiones del proyecto dando énfasis en las fortalezas, debilidades y sugerencias para trabajos futuros.

Posteriormente se le permite al lector una informacional adicional para poder ampliar los conocimientos en el campo de acción del proyecto, como también bases de información donde este pueda acceder para profundizar en el tema.

Apéndice 1: Se presenta 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. 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 de simplemente el uso, el envejecimiento, las 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 en estructuras, 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.

Un estudio sobre las implicaciones económicas de detectar daño a través de un sistemas de monitoreo de salud estructural (SHM) la cual es una tecnología que combina la captura de datos de una estructura con el análisis de la misma. Los datos se consiguen desde una red sensorial no destructiva, desde la cual se obtienen indicadores que permiten detectar anomalías (daños o degradación) en la estructura. Esta información puede ser periódica o en tiempo real y continuo, y estar asociada a cambios diversos como deterioro, corrosión, fatiga, reacciones químicas, humedad, cambios en las variables del ambiente, así como a las propiedades físicas relativas a la carga, esfuerzos, desplazamientos, deformaciones, aceleraciones, agrietamiento, vibraciones, dislocaciones y otros que sean necesarios para la *evaluación operacional de la estructura*. Esta evaluación demanda contrastar la situación de la estructura con un perfil saludable de la misma. Se 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.

Por otra parte, las principales técnicas de tipo DDND consisten en inspección visual, inspección 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; 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.

A pesar de todos los argumentos mencionados anteriormente, por diferentes factores, especialmente económicos, la importancia de disponer de un método de detección de daños, es relegada a casi el olvido en gran parte de la industria nacional e internacional. Por consiguiente no se cuenta con un apoyo informático, hay que implementar desarrollar una herramienta analítica que supere las limitaciones de los métodos DDND convencionales y conlleve a garantizar la funcionalidad y la seguridad de la estructura en cuestión, generando un aporte significativo al ser implementada.

¹ Extraído de datos e índices ingeniería Perú y Pemex:
(http://www.ingenieria.peru-v.com/documentos/Monitoreo_de_salud_estructural.pdf
(<http://www.invides.com.mx/antiores/Noviembre2001/htm/plata.html>

1.1. ANTECEDENTES

A causa de la deficiencia y/o inexistencia de un plan de mantenimiento preventivo y predictivo lo cual es muy común en países subdesarrollados como por ejemplo Colombia, 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.

Por los motivos anteriores es un compromiso como estudiantes de Ingeniería Mecánica de la Universidad Industrial de Santander, el de contribuir para la solución a estos problemas de una manera eficiente, aplicando todos los conocimientos adquiridos, complementados por una debida investigación y haciendo uso de todos los recursos disponibles para la formulación de la solución optima al problema.

Recursos tales como las tecnologías informáticas abren un amplio campo de estudio en el desenvolvimiento 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.

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; y la capacidad de realizarse mediciones remotamente, lo que rápidamente minimiza el impacto sobre el funcionamiento de la estructura.

Las técnicas DDND analíticas incluyen varios métodos para examinar materiales, componentes y conexiones destinadas a identificar y cuantificar defectos internos y degradaciones. Los Ensayos No Destructivos tienen como finalidad evaluar propiedades materiales antes de que resulten fallidas -basándose en criterios de aceptación reconocidos o perfiles de degradación definidos a lo largo del tiempo- para asegurar la calidad del producto y el funcionamiento durante su producción, así como para asegurar el uso seguro de las estructuras de ingeniería. 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.

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.

El problema de detección de daños es en esencia un problema de optimización, por lo cual la aplicabilidad de metaheurísticas se convierte en una alternativa real, 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.

Una de las técnicas de optimización se conoce como **Algoritmos Genéticos (AG)**, la cual hace una analogía del proceso evolutivo y la supervivencia del individuo más apto para tener una población inicial de soluciones propuestas a nuestro problema a evolucionar a través de generaciones, y así encontrar la mejor respuesta. Los Algoritmos Genéticos son sistemas eficientes para la solución de muchos tipos de problemas en varias áreas de aplicabilidad como lo son funciones numéricas, optimizaciones combinatorias, tamaño de elementos, maquinas que aprenden, entre otras.

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.

1.2. ALCANCE DEL PROYECTO

1.2.1 Objetivo General

Aportar en los avances de investigación que se adelantan cada día en la Universidad Industrial de Santander, por medio de la aplicación de las nuevas técnicas de la inteligencia artificial, más específicamente en las técnicas de

optimización para el monitoreo de la salud estructural, buscando el avance científico y el acceso a nuevas teorías y conocimientos para poder tener un amplio y crítico campo de interacción en la Ingeniería Mecánica.

1.2.2. Objetivos Específicos

- Implementar un sistema de detección de daños (localización y cuantificación) utilizando los parámetros dinámicos de la estructura (frecuencias y formas de vibración) sin y con daño. Para lo cual se debe implementar un modelo de elementos finitos de la estructura usando la herramienta computacional de libre distribución OPENSEES 2.1.0, la cual aportara los datos de entrada para la función objetivo a ser optimizada por medio de la técnica de Algoritmos Genéticos (ver figura 1) utilizando el entorno MATLAB, con el fin de verificar su aplicabilidad en el análisis de dos ejemplos básicos de estructuras: una viga discretizada en 10 elementos y una armadura de 13 elementos (ver figura 2 y figura 3 respectivamente), en diversos escenarios de daño (daño simple y múltiple).
- Utilizar la función objetivo desarrollado por Friswell, Penny y Garvey (1998) y la función objetivo de Moslem y Nafaspour (2002) 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, sobre el desempeño de las metodologías implementadas.

- Analizar entre los resultados que se obtuvieron en el trabajo de grado detección de daños en estructuras utilizando la técnica de optimización PSO, con los resultados que se obtendrán mediante la técnica de algoritmos genéticos (AG ²), para definir cuál es más eficiente en el análisis estructural para los dos sistemas estudiados.

²DETECCIÓN DE DAÑOS EN ESTRUCTURAS UTILIZANDO LA TÉCNICA DE OPTIMIZACIÓN AG
AUTOR, ROBINSON BARRERA NÚÑEZ, OSCAR MAURICIO SOTO DÍAZ
DIRECTORES. MAGISTER, ING MECÁNICO JABID QUIROGA, ING CIVIL IVÁN GÓMEZ

2. REVISIÓN BIBLIOGRÁFICA

2.1. ANÁLISIS DINÁMICO

La teoría de dinámica de estructuras sometidas a cargas dinámicas tiende a adoptar una forma de movimiento cinemático especial, precisa que una estructura no amortiguada con múltiples grados de libertad posee un movimiento armónico simple sin cambiar la forma de deflexión. Por ejemplo en un sistema de dos grados de libertad, existen dos formas de deflexión como se muestra en la Figura 2.1 . A medida que se incrementa los grados libertad estudiados en la estructura, se incrementa el número de formas de vibración. Cuando una estructura se desplaza de una posición de equilibrio estable, la estructura tiende a volver a esta posición, debido a la acción de las fuerzas que tienden a restablecer la situación de equilibrio. Sin embargo, en el instante en que la estructura vuelve a su posición de equilibrio cuenta con un impulso que lo lleva más allá de dicha posición, presentándose una oscilación alrededor del punto de equilibrio las cuales se denominan como vibraciones mecánicas.

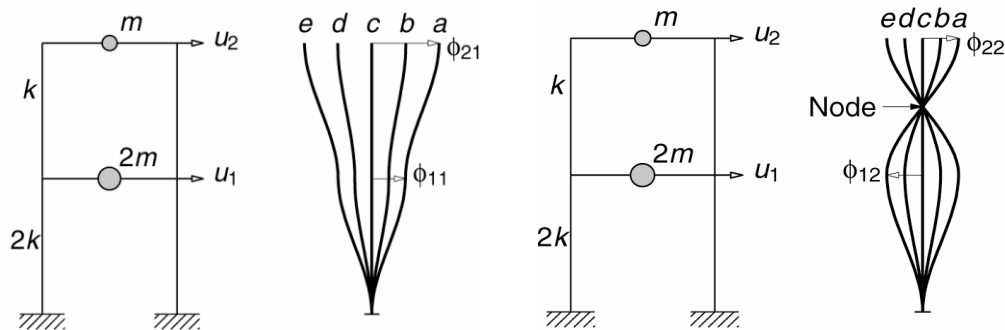


Figura 2.1. Modos de vibración de una estructura de 2 grados de libertad (CHOPRA, 2001)

La dinámica estructural generalmente se dedica a las vibraciones de los cuerpos flexibles, en donde las deformaciones relativas entre los diferentes puntos que conforman la estructura son importantes; 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.

En el análisis dinámico de una estructura existe un aspecto de gran importancia, y es la definición del modelo estructural que mejor represente el comportamiento real de la estructura. En general, una estructura en un continuo que posee una estructura definida pero que puede estar compuesto por diferentes materiales, los cuales muchas veces cuentan con ecuaciones constitutivas complicadas. Definir un modelo exacto sería muy complejo, y en la mayoría de los casos su exactitud se perdería, tanto por la complejidad de los modelos matemáticos que se utilizan en el análisis como por los errores generados durante el proceso de cálculo. Para evitar este inconveniente se definió un modelo dinámico simplificado que implica discretizar la estructura de tal manera que permita una formulación adecuada del modelo matemático el cual permita el cálculo de la respuesta. 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: SISTEMAS DE MÚLTIPLES GRADOS DE LIBERTAD (MDOF)

En aquellos casos donde el modelo matemático de un solo grado de libertad no provee la respuesta dinámica exacta, pues la estructura no puede asumir, durante su movimiento, una forma única de desplazamiento, se tiene que considerar el modelo como un sistema de múltiples grados de libertad, La respuesta dinámica de una estructura de múltiples grados de libertad se determina usando (2.1).

$$\mathbf{M}\ddot{\mathbf{q}}(t) + \mathbf{C}\dot{\mathbf{q}}(t) + \mathbf{K}\mathbf{q}(t) = \mathbf{p}(t) \quad (2.1)$$

donde \mathbf{M} , \mathbf{C} y \mathbf{K} son las 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 . $\mathbf{p}(t)$ es un vector de excitación y $\mathbf{q}(t)$ es el vector de desplazamiento del sistema. Una alternativa para solucionar (1) es la aproximación modal, que se asume el sistema no amortiguado (2.2) y cuya i -ésima solución está dada por (2.3)

$$\mathbf{M}\ddot{\mathbf{q}}(t) + \mathbf{K}\mathbf{q}(t) = \mathbf{0} \quad (2.2)$$

$$\mathbf{q}(t) = \Phi_i \sin \omega_i t - \alpha_i \quad (2.3)$$

Donde Φ_i es la i -forma del modo (vector propio) con una correspondiente frecuencia natural circular ω_i (valor propio) y ángulo de fase α_i . Substituyendo (2.3) en (2.2), y eliminando $\sin \omega_i t - \alpha_i$ se obtiene (2.4)

$$K - \omega_i^2 M \Phi_i = 0 \quad (2.4)$$

La ecuación (2.4) puede ser escrita como un sistema de N- ecuaciones por lo cual puede ser dada la ecuación (2.5):

$$\begin{array}{cccccc}
 k_{11} - \omega_1^2 m_1 & k_{12} & \dots & k_{1N} & \phi_1 & 0 \\
 k_{12} & k_{22} - \omega_2^2 m_2 & \dots & k_{2N} & \phi_2 & 0 \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 k_{N1} & k_{N2} & \dots & k_{NN} - \omega_N^2 m_N & \phi_N & 0
 \end{array} = \begin{array}{c} \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \end{array} \quad (2.5)$$

La enunciación de las ecuaciones 2.4 y 2.5 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 2.4 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 2.4 sea cero, por eso:

$$Det K - \omega_i^2 M = 0 \quad (2.6)$$

La ecuación 2.6 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 2.5 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 2.4 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:

$$K\Phi_i = \omega_i^2 M\Phi_i \quad (2.7)$$

La solución de la ecuación corresponde a la determinación de los valores y vectores propios del sistema, los cuales se normalizan a través de la matriz de masa ecuación (2.4), donde Φ_{ni} es la forma modal normalizada.

$$\Phi_n^T M \Phi_n = I \rightarrow \Phi_{ni} = \frac{\Phi_i}{\Phi_i^T M \Phi_i} \quad (2.8)$$

2.1.2. DETECCIÓN DE DAÑO

El daño se define comúnmente como una reducción de la rigidez en una o más elementos, lo que conlleva a una reducción en el módulo de elasticidad del material.

Una forma importante para evaluar el daño es a través de metodologías que se basan en los cambios de los parámetros dinámicos de la estructura. Estas metodologías parten del principio de que después de acontecido el daño, se producen variaciones en las propiedades de la estructura (matriz de masa, rigidez y amortiguación del modelo de elementos finitos), lo que implica variaciones en las propiedades dinámicas de la estructura (frecuencias naturales, formas modales y amortiguamiento modal).

Cuando se desea proponer una metodología para la detección de daños basada en las vibraciones se debe tener en cuenta los siguientes aspectos:

- Dependencia de un modelo de elementos finitos adecuado para representar la estructura y así poder identificar los datos experimentales en su condición real. Para lo cual se adoptó un modelo de elementos finitos utilizando un software de libre distribución llamado OPENSEES 2.1.0 con el cual se logró representar el estado inicial sin dañar la estructura, en la práctica actual del algoritmo tratado en este documento, las propiedades dinámicas de el modelo podrían tener diferencias en la características de la estructura real, dando lugar a la introducción de errores en los resultados ofrecidos por la metodología. Las posibles discrepancias se deben a aproximaciones en las condiciones de contorno, las diferencias entre la

conectividad de elementos del modelo y el de la estructura real, las mallas gruesas o tipos elementos incorrectos.

- Comportamiento lineal o no lineal de la estructura. Una de las principales aproximaciones encontradas en la literatura técnica es el comportamiento global de la estructura, después de haber sido dañada, es encontrarse en el régimen lineal. Cabe resaltar que hay ejemplos clásicos de no linealidad como el caso de las estructuras en concreto en las que aparecen fisuras, las cuales pueden abrir y cerrar las condiciones normales de su funcionamiento.

- Sensibilidad de los parámetros modales en presencia de daños. Se espera que el parámetro utilizado sea más susceptible al posible daño, principalmente para varios niveles de daños. Es frecuentemente aceptado que las frecuencias naturales son más sensibles a los daños en comparación con las formas modales. Zhao y DeWolf (1999) derivan los coeficientes de sensibilidad de las frecuencias naturales, formas modales y la flexibilidad entre los modos a los cambios de rigidez del sistema. Los resultados mostraron que flexibilidad entre los modos a los cambios de rigidez era el parámetro más sensible a los daños.

- La información sobre el daño necesario. Según Rytter (DOEBLING Apud, 1998) una metodología de detección de daños ofrece cuatro niveles de detalles sobre el estado de la estructura: I) La identificación de la presencia de daño estructura, II) Nivel I + Localización de elementos dañados, III) Nivel II + Cuantificación de la extensión del daño y IV) Nivel III +

determinación de la vida estructura útil restante. Actualmente, la mayoría metodología logra llegar hasta un Nivel III, en el caso de utilizar un modelo para la representación de la condición inicial de la estructura.

2.2. METODOLOGÍA EMPLEADA PARA LA LOCALIZACIÓN 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 y las denominadas metaheurísticas.

Estas técnicas convencionales de optimización 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.

2.3. FUNCIÓN OBJETIVO

El sustento de todo sistema de monitoreo es el indicador de falla. En el presente trabajo el indicador de falla, que es el cambio en el módulo de elasticidad, se obtiene de un proceso de optimización al comparar las características dinámicas experimentales con las de un modelo analítico de la estructura monitoreada. Las funciones objetivo a optimizar (ec.2.9) propuesta por M.Friswell, J.Penny, y S.Garvey (1998); esta función objetivo corresponde al archivo de MATLAB llamado **CalibraMACviga.m** con la cual se analizo la viga propuesta y **CalibraMACarmadura.m** para el análisis de la armadura; la segunda función objetivo planteada en el presente trabajo (ec.2.10) desarrollada por K.Moslem, R.Nafaspour (2002); esta segunda función objetivo corresponde al archivo de MATLAB llamado **CalibraV1viga.m** con la cual se analizó la viga propuesta y **CalibraV1armadura.m** para el análisis de la armadura.

Las dos funciones objetivos buscan minimizar la diferencia entre el comportamiento experimental y el de referencia, variando el módulo de elasticidad. El proceso de optimización permite determinar los módulos de elasticidad asociados a cada elemento de la estructura y su comparación con los valores iniciales permiten la determinación de la condición normal o de falla en la estructura.

$$\text{Min } f = \sum_{j=1}^r W_{w_j} \frac{\delta w_{mj} - \delta w_{aj}}{w_{mj}}^2 + \sum_{j=1}^r W_{\phi_j} \mathbf{1} - \text{MAC } \phi_{mj}, \phi_{aj} + W_{ns} \delta_{ns} \quad (\text{ec. 2.9})$$

$$\text{Min } f = \sum_{j=1}^r W_{w_j} \mathbf{1} - \frac{w_{mj}}{w_{aj}}^2 + \sum_{j=1}^r \sum_{i=1}^s W_{\phi_{ji}} \phi_{mji} - \phi_{aj} \quad (\text{ec. 2.10})$$

En (e.c 2.9–2.10) el subíndice m corresponde a los datos que fueron medidos o simulados y el subíndice a corresponde a los 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, 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 MAC es el Criterio de Confianza Modal de sus siglas en inglés Modal Assurance Criterion, y permite estimar el grado de correlación entre los modos de vibración (ec.2.11).

$$\text{MAC}_{ij} = \frac{(\phi_i^T \phi_j)}{\phi_i^T \phi_i (\phi_j^T \phi_j)} \quad (\text{ec.2.11})$$

Donde ϕ_i es el vector propio del modo i que contiene solo los grados de libertad medidos y ϕ_j es el correspondiente al valor experimental del vector propio del modo j . Finalmente, como se puede observar, las funciones objetivo utilizadas en esta investigación utilizan como parámetros de indicación del daño la frecuencia natural y los modos de vibración, de manera simultánea.

3. OPTIMIZACIÓN CON ALGORITMOS GENÉTICOS

Los algoritmos genéticos* están inspirados en la naturaleza, en el fenómeno de la evolución. La evolución la consideramos como la causa de los cambios en el contenido genético de una población.

Un tema polémico, con opiniones variadas dependiendo de si se trata de informáticos evolutivos o de biólogos o geneticistas, es si la evolución optimiza o no. Según los **informáticos evolutivos**, la evolución optimiza, puesto que va creando seres cada vez más perfectos, cuya cumbre es el hombre; además, indicios de esta optimización se encuentran en el organismo de los animales, desde el tamaño y tasa de ramificación de las arterias, diseñada para maximizar flujo, hasta el metabolismo, que optimiza la cantidad de energía extraída de los alimentos. Sin embargo, **los geneticistas y biólogos evolutivos** afirman que la evolución no optimiza, sino que adapta y optimiza localmente en el espacio y el tiempo; evolución no significa progreso. Un organismo más evolucionado puede estar en desventaja competitiva con uno de sus antepasados, si se colocan en el ambiente del último.

*Tomado de Algoritmos Genéticos. Aplicación al Juego de las N Reinas, Juan Carlos Pozas Bustos Ing. Telecomunicación, Univ. Carlos III de Madrid; Nieves Vázquez Vázquez Ing. Telecomunicación, Univ. Carlos III de Madrid

3.1. INTRODUCCIÓN A LOS ALGORITMOS GENÉTICOS

Los Algoritmos Genéticos, fueron inventados en 1975 por John Holland, de la Universidad de Michigan. Los Algoritmos Genéticos son, simplificando, algoritmos de optimización, es decir, tratan de encontrar la mejor solución a un problema dado entre un conjunto de soluciones posibles. Los mecanismos de los que se valen los Algoritmos Genéticos para llevar a cabo esa búsqueda pueden verse como una metáfora de los procesos de evolución biológica.

John Holland desde pequeño, se preguntaba cómo logra la naturaleza, crear seres cada vez más perfectos. No sabía la respuesta, pero tenía una cierta idea de cómo hallarla: tratando de hacer pequeños modelos de la naturaleza, que tuvieran alguna de sus características, y ver cómo funcionaban, para luego extrapolar sus conclusiones a la totalidad.

Fue a principios de los 60, en la Universidad de Michigan en Ann Arbor, donde, dentro del grupo Logic of Computers, sus ideas comenzaron a desarrollarse y a dar frutos. Y fue, además, leyendo un libro escrito por un biólogo evolucionista, R. A. Fisher, titulado La teoría genética de la selección natural, como comenzó a descubrir los medios de llevar a cabo sus propósitos de comprensión de la naturaleza. De ese libro aprendió que la evolución era una forma de adaptación más potente que el simple aprendizaje, y tomó la decisión de aplicar estas ideas para desarrollar programas bien adaptados para un fin determinado.

En esa universidad, Holland impartía un curso titulado Teoría de sistemas adaptativos. Dentro de este curso, y con una participación activa por parte de sus estudiantes, fue donde se crearon las ideas que más tarde se convertirían en los Algoritmos Genéticos.

Por tanto, cuando Holland se enfrentó a los Algoritmos Genéticos, los objetivos de su investigación fueron dos:

- Imitar los procesos adaptativos de los sistemas naturales.
- Diseñar sistemas artificiales (normalmente programas) que retengan los mecanismos importantes de los sistemas naturales.

Los Algoritmos Genéticos son métodos adaptativos que pueden usarse para resolver problemas de búsqueda y optimización. Están basados en el proceso genético de los organismos vivos. A lo largo de las generaciones, las poblaciones evolucionan en la naturaleza de acorde con los principios de la selección natural y la supervivencia de los más fuertes, postulados por Darwin (1859). Por imitación de este proceso, los AG son capaces de ir creando soluciones para problemas del mundo real. La evolución de dichas soluciones hacia valores óptimos del problema depende en buena medida de una adecuada codificación de las mismas.

En la naturaleza los individuos de una población compiten entre sí en la búsqueda de recursos tales como comida, agua y refugio. Incluso los miembros de una misma especie compiten a menudo en la búsqueda de un compañero. Aquellos individuos que tienen más éxito en sobrevivir y en atraer compañeros tienen mayor probabilidad de generar un gran número de descendientes. Por el contrario individuos poco dotados producirán un menor número de descendientes. Esto significa que los genes de los individuos mejor adaptados se propagaran en sucesivas generaciones hacia un número de individuos creciente. La combinación de buenas características provenientes de diferentes ancestros, puede a veces producir descendientes "súperindividuos", cuya adaptación es mucho mayor que la

de cualquiera de sus ancestros. De esta manera, las especies evolucionan logrando unas características cada vez mejor adaptadas al entorno en el que viven.

El poder de los Algoritmos Genéticos proviene del hecho de que se trata de una técnica robusta, y pueden tratar con éxito una gran variedad de problemas provenientes de diferentes áreas, incluyendo aquellos en los que otros métodos encuentran dificultades. Si bien no se garantiza que el Algoritmo Genético encuentre la solución óptima del problema, existe evidencia empírica de que se encuentran soluciones de un nivel aceptable, en un tiempo competitivo con el resto de algoritmos de optimización combinatoria.

3.2. ALGORITMOS GENÉTICOS EN LA OPTIMIZACIÓN

El creciente interés por los Algoritmos Genéticos es que estos son un método global y robusto de búsqueda de las soluciones de problemas. La principal ventaja de estas características es el equilibrio alcanzado entre la eficiencia y eficacia para resolver diferentes y muy complejos problemas de grandes dimensiones.

Lo que aventaja a los Algoritmos Genéticos frente a otros algoritmos tradicionales de búsqueda es que se diferencian de estos en los siguientes aspectos:

- Trabajan con una codificación de un conjunto de parámetros, no con los parámetros mismos.
- Trabajan con un conjunto de puntos, no con un único punto y su entorno (su técnica de búsqueda es global.) Utilizan un subconjunto del espacio total,

para obtener información sobre el universo de búsqueda, a través de las evaluaciones de la función a optimizar. Esas evaluaciones se emplean de forma eficiente para clasificar los subconjuntos de acuerdo con su idoneidad.

- No necesitan conocimientos específicos sobre el problema a resolver; es decir, no están sujetos a restricciones. Por ejemplo, se pueden aplicar a funciones no continuas, lo cual les abre un amplio campo de aplicaciones que no podrían ser tratadas por los métodos tradicionales.
- Utilizan operadores probabilísticos, en vez de los típicos operadores determinísticos de las técnicas tradicionales.
- Resulta sumamente fácil ejecutarlos en las modernas arquitecturas masivas en paralelo.
- Cuando se usan para problemas de optimización, resultan menos afectados por los máximos locales que las técnicas tradicionales (son métodos robustos).

Un esquema del funcionamiento general de un algoritmo genético podría ser el siguiente:

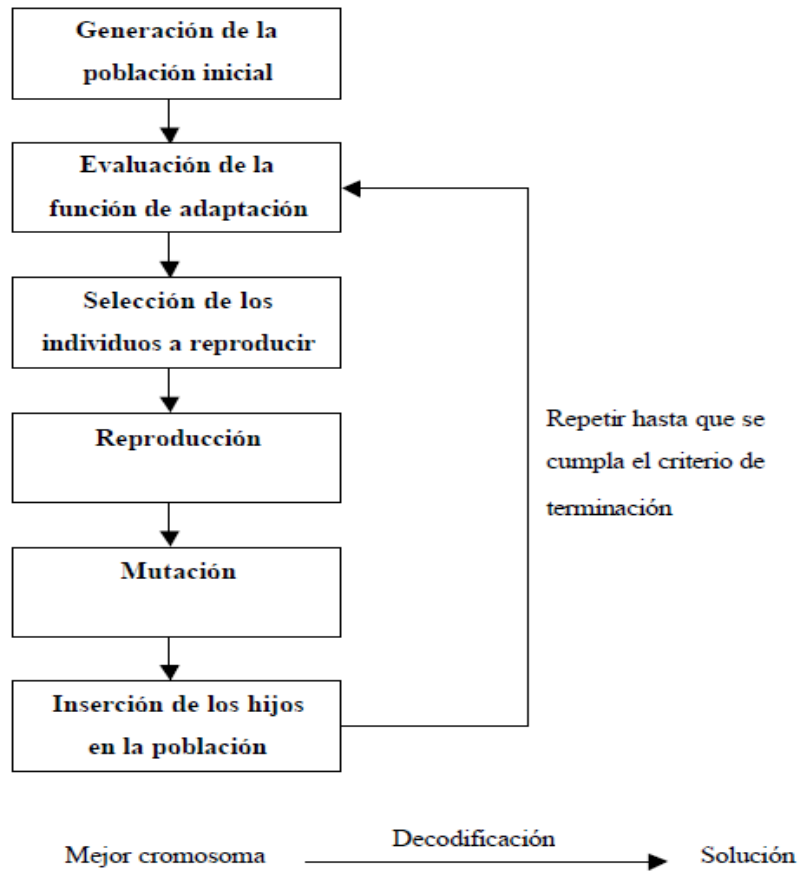


Figura 3.1. Esquema de funcionamiento del algoritmo genético.

3.2.1. APLICACIONES DE LOS ALGORITMOS GENÉTICOS

El área de aplicación de los Algoritmos Genéticos es muy amplia, y en general sus aplicaciones se pueden implementar a muchos de los problemas de la vida cotidiana, de igual forma, se hayan aplicado a diversos problemas y modelos en ingeniería, y en la ciencia en general cabe destacar entre ellos:

- **Optimización:** Se trata de un campo especialmente abonado para el uso de los Algoritmos Genéticos, por las características intrínsecas de estos problemas. No en vano fueron la fuente de inspiración para el desarrollo del presente trabajo. Los Algoritmos Genéticos se han utilizado en numerosas tareas de optimización, incluyendo la optimización numérica, y los problemas de optimización combinatoria.
- **Programación automática:** Los Algoritmos Genéticos se han empleado para desarrollar programas para tareas específicas, y para diseñar otras estructuras computacionales tales como el autómata celular, y las redes de clasificación.
- **Aprendizaje máquina:** Los Algoritmos Genéticos se han utilizado también en muchas de estas aplicaciones, tales como la predicción del tiempo o la estructura de una proteína. Han servido asimismo para desarrollar determinados aspectos de sistemas particulares de aprendizaje, como pueda ser el de los pesos en una red neuronal, las reglas para sistemas de clasificación de aprendizaje o sistemas de producción simbólica, y los sensores para robots.
- **Economía:** En este caso, se ha hecho uso de estos Algoritmos para modelar procesos de innovación, el desarrollo estrategias de puja, y la aparición de mercados económicos.
- **Sistemas inmunes:** A la hora de modelar varios aspectos de los sistemas inmunes naturales, incluyendo la mutación somática durante la vida de un individuo y el descubrimiento de familias de genes múltiples en tiempo evolutivo, ha resultado útil el empleo de esta técnica.

- **Ecología:** En la modelización de fenómenos ecológicos tales como las carreras de armamento biológico, la coevolución de parásito-huésped, la simbiosis, y el flujo de recursos.
- **Genética de poblaciones:** En el estudio de preguntas del tipo "¿Bajo qué condiciones será viable evolutivamente un gene para la recombinación?".
- **Evolución y aprendizaje:** Los Algoritmos Genéticos se han utilizado en el estudio de las relaciones entre el aprendizaje individual y la evolución de la especie.
- **Sistemas sociales:** En el estudio de aspectos evolutivos de los sistemas sociales, tales como la evolución del comportamiento social en colonias de insectos, y la evolución de la cooperación y la comunicación en sistemas multi-agentes.

Aunque esta lista no es, en modo alguno, exhaustiva, sí transmite la idea de la variedad de aplicaciones que tienen los Algoritmos Genéticos.

3.3. ALGORITMOS GENÉTICOS

3.3.1. GENERACIÓN

Una cuestión que se puede plantear es la relacionada con el tamaño idóneo de las generaciones. Parece intuitivo que las generaciones pequeñas corren el riesgo de no cubrir adecuadamente el espacio de búsqueda, mientras que el trabajar con generaciones de gran tamaño puede acarrear problemas relacionados con el

excesivo costo computacional. En el presente trabajo basándonos en resultados experimentales se llegó a la conclusión que el tamaño óptimo de generaciones para la función objetivo **CalibraV1viga.m** fue de 500 generaciones; en el caso de la segunda función objetivo **CalibraMACarmadura.m** el número de generaciones fue de 300, estas generaciones son suficientes para atacar con éxito las estructuras consideradas.

3.3.2. POBLACIÓN INICIAL

Habitualmente la población inicial se escoge generando ristas al azar. En algunos trabajos, se constata que esta inicialización no aleatoria de la población inicial, puede acelerar la convergencia del Algoritmos Genéticos. Sin embargo en algunos casos la desventaja resulta ser la prematura convergencia del algoritmo, queriendo indicar con esto la convergencia hacia óptimos locales.

La población inicial de un Algoritmos Genéticos puede ser creada de muy diversas formas, desde generar aleatoriamente el valor de cada gen para cada individuo, utilizar una función ávida o generar alguna parte de cada individuo y luego aplicar una búsqueda local. Para el presente trabajo se llegó a la conclusión basándonos en resultados experimentales que para la viga una población inicial de 50 individuos y para el caso de la armadura una de población de 150 individuos.

3.3.3. FUNCIÓN OBJETIVO

Dos aspectos que resultan cruciales en el comportamiento de los Algoritmos Genéticos son la determinación de una adecuada función de adaptación o función objetivo, así como la codificación utilizada, este tema fue abarcado más ampliamente en el numeral 2.4 *FUNCIONES OBJETIVO*.

3.3.4. SELECCIÓN

El operador de Selección es el encargado de transmitir y conservar aquellas características de las soluciones que se consideran valiosas a lo largo de las generaciones. El principal medio para que la información útil se transmita es que aquellos individuos mejor adaptados (mejor valor de función de evaluación) tengan más probabilidades de reproducirse. Sin embargo, es necesario también incluir un factor aleatorio que permita reproducirse a individuos que aunque no estén muy bien adaptados, puedan contener alguna información útil para posteriores generaciones, con el objeto de mantener así también una cierta diversidad en cada población. Algunas de las técnicas de las cuales se dispone son las siguientes:

3.3.4.1. RULETA O SELECCIÓN PROPORCIONAL

Con este método la probabilidad que tiene un individuo de reproducirse es proporcional a su valor de la función objetivo, es decir, a su adaptación. En este método se define un rango con las características de la selección por sorteo. El número al azar será un número aleatorio forzosamente menor que el tamaño del rango. El elemento escogido será aquel en cuyo rango esté el número resultante de sumar el número aleatorio con el resultado total que sirvió para escoger el elemento anterior. El comportamiento es similar al de una ruleta, donde se define un avance cada tirada a partir de la posición actual. Tiene la ventaja de que no es posible escoger dos veces consecutivas el mismo elemento, y que puede ser forzado a que sea alta la probabilidad de que no sean elementos próximos en la población; por este tipo de características y por los resultados experimentales fue el método escogido para ser implementado en el Algoritmo Genético propuesto en el desarrollo del proyecto.

3.3.4.2. SELECCIÓN POR RANKING

Desarrollado por Whitley (1989) consiste en calcular las probabilidades de reproducción atendiendo a la ordenación de la población por el valor de adaptación en vez de atender simplemente a su valor de adecuación. Estas probabilidades se pueden calcular de diversas formas, aunque el método habitual es el ranking lineal (Baker (1985)).

3.3.4.3. SELECCIÓN POR TORNEO

Reporta un valor computacional muy bajo debido a su sencillez. Se selecciona un grupo de t individuos (normalmente $t = 2$, torneo binario) y se genera un número aleatorio entre 0 y 1. Si este número es menor que un cierto umbral K (usualmente 0,75), se selecciona para reproducirse al individuo con mejor adaptación, y si este número es mayor que K , se selecciona, por el contrario, al individuo con peor adaptación.

Esta técnica tiene la ventaja de que permite un cierto grado de elitismo -el mejor nunca va a morir, y los mejores tienen más probabilidad de reproducirse y de emigrar que los peores- pero sin producir una convergencia genética prematura, si la población es, al menos, un orden de magnitud superior al del número de elementos involucrados en el torneo. En caso de que la diferencia sea menor no hemos observado mucha diferencia por lo cual fue descartado la selección por torneo para ser implementada en la configuración del Algoritmo Genético.

3.3.5. CRUZAMIENTO

El operador de cruzamiento permite realizar una exploración de toda la información almacenada hasta el momento en la población y combinarla para crear mejores individuos.

3.3.5.1. CRUCE DE UN PUNTO

Es el método de cruce más sencillo. Se selecciona una posición en las cadenas de los progenitores, y se intercambian los genes a la izquierda de esta posición. Debido a su sencillez en este método de cruzamiento se evidencio en las pruebas realizadas a las estructuras propuestas a analizarse en el Algoritmo Genético que se perdía información valiosa de sus progenitores por tal motivo de descarto para ser usado en el Algoritmo.

3.3.5.2. CRUCE DE N PUNTO

Es una generalización del método anterior. Se seleccionan varias posiciones (n) en las cadenas de los progenitores y se intercambian los genes a ambos lados de estas posiciones.

3.3.5.3. CRUCE HEURÍSTICO

Se caracteriza por producir uno o ningún hijo al cruzar dos padres regresando un hijo que se encuentra en la línea con los dos padres, a poca distancia de los padres con el mejor valor en la dirección opuesta a la de la madre con el valor del peor adaptado. Puede especificar hasta qué punto el hijo es del mejor padre por el parámetro R, se determina al seleccionar heurístico. El valor por defecto de esta

Relación es de 1,2. Si parent1 y parent2 son los padres, y parent1 tiene el mejor valor de aptitud, la función regresa el hijo.

$$\text{Hijo} = \text{parent2} + R * (\text{parent1} - \text{parent2}) \quad (3.1)$$

3.3.5.4. CRUCE UNIFORME

Se realiza un test aleatorio para decidir de cuál de los progenitores se toma cada posición de la cadena.

3.3.5.5. CRUCES PARA PERMUTACIÓN

Existe una familia de cruces específicas para los problemas de permutación, siendo algunos de ellos:

- ***Cruce de mapeamiento parcial:*** Toma una subsecuencia del genoma del padre y procura preservar el orden absoluto de los fenotipos -es decir, orden y posición en el genoma- del resto del genoma lo más parecido posible de la madre.
- ***Cruce de orden:*** toma una subsecuencia del genoma del padre y procura preservar el orden relativo de los fenotipos del resto del genoma lo más parecido posible de la madre.
- ***Cruce de ciclo:*** Tomamos el primer gen del genoma del padre, poniéndolo en la primera posición del hijo, y el primer gen del genoma de la madre, poniéndolo dentro del genoma del hijo en la posición que ocupe en el genoma del padre. El fenotipo que está en la posición que ocupa el gen del genoma del padre igual al primer gen del genoma de la madre se va a

colocar en la posición que ocupe en el genoma del padre, y así hasta rellenar el genoma del hijo.

3.3.6. MUTACIÓN

La mutación se considera un operador básico, que proporciona un pequeño elemento de aleatoriedad en la vecindad (entorno) de los individuos de la población. Si bien se admite que el operador de cruce es el responsable de efectuar la búsqueda a lo largo del espacio de posibles soluciones, también parece desprenderse de los experimentos efectuados por varios investigadores que el operador de mutación va ganando en importancia a medida que la población de individuos va convergiendo (Davis). El objetivo del operador de mutación es producir nuevas soluciones a partir de la modificación de un cierto número de genes de una solución existente, con la intención de fomentar la variabilidad dentro de la población. Existen muy diversas formas de realizar la mutación, desde la más sencilla (Puntual), donde cada gen muta aleatoriamente con independencia del resto de genes, hasta configuraciones más complejas donde se tienen en cuenta la estructura del problema y la relación entre los distintos genes.

Schaffer y col. encuentran que el efecto del cruce en la búsqueda es inferior al que previamente se esperaba. Utilizan la denominada evolución primitiva, en la cual, el proceso evolutivo consta tan sólo de selección y mutación. Encuentran que dicha evolución primitiva supera con creces a una evolución basada exclusivamente en la selección y el cruce. Otra conclusión de su trabajo es que la determinación del valor óptimo de la probabilidad de mutación es mucho más crucial que el relativo a la probabilidad de cruce. Si bien en la mayoría de las implementaciones de AG se

asume que tanto la probabilidad de cruce como la de mutación permanecen constantes, se han obtenido mejores resultados experimentales modificando la probabilidad de mutación a medida que aumenta el número de iteraciones.

3.3.7. INSERCIÓN ELITISTA

Se eligen los cromosomas menos aptos para ser reemplazados. Este método asegura que los mejores cromosomas pasarán siempre a la siguiente generación, pero puede restringir la amplitud de la búsqueda que realiza el algoritmo genético.

3.3.8. CRITERIOS DE TERMINACIÓN DEL ALGORITMO GENÉTICO

El criterio de terminación del algoritmo genético es el encargado de definir el momento en el cual debe detenerse el ciclo de evolución y adoptar el cromosoma más apto como la solución encontrada por el algoritmo genético. A continuación se describen los criterios más comúnmente utilizados.

3.3.8.1. CRITERIO DE CONVERGENCIA DE IDENTIDAD

Este criterio consiste en detener al algoritmo genético cuando un determinado porcentaje de los cromosomas de la población representan a la misma solución. Los operadores del algoritmo genético tienden a preservar y difundir el material genético de los cromosomas más aptos, por lo que es de esperar que luego de un gran número de generaciones, alguna solución con gran valor de aptitud se imponga y domine la población.

3.3.8.2. CRITERIO DE CONVERGENCIA DE APTITUD

Puede suceder que existan soluciones equivalentes o cuasi equivalentes a un problema, que obtengan valores de aptitud similares. En ese caso, es probable que no haya una solución que se imponga en la población (y el criterio de terminación por convergencia de identidad nunca se cumpla). Este criterio no espera a que la población se componga mayoritariamente de una sola solución, sino que finaliza la ejecución del algoritmo cuando los valores de aptitud de un determinado porcentaje de las soluciones son iguales, o difieren en un pequeño porcentaje. Por ejemplo, cuando el 90% de las soluciones tenga valores de aptitud que no difieran en más de un 1%.

3.3.8.3. CRITERIO DE CANTIDAD DE GENERACIONES

Los métodos anteriores apuntan a esperar a que la evolución de la población llegue a su fin. Cuando alguno de ellos se cumple, es probable que las soluciones no sigan mejorando mucho más, no importa cuántas generaciones más se ejecuten. Sin embargo, los algoritmos genéticos pueden necesitar un número de generaciones muy grande para llegar a la convergencia, dependiendo de las tasas de reproducción y mutación. Utilizando cualquiera de los dos criterios anteriores no puede estimarse un número máximo de generaciones, ya que esto dependerá no solamente de los parámetros del algoritmo genético sino también del azar.

Esto puede ser un problema, sobre todo si se quieren comparar los tiempos de resolución de un problema mediante algoritmos genéticos con otros métodos [Estivill-Castro, 2000]. El criterio de terminación por cantidad de generaciones consiste simplemente en finalizar la ejecución una vez que ha transcurrido un número determinado de generaciones. Este método permite determinar con precisión los tiempos de ejecución del algoritmo a costa de detener la evolución sin la certeza de que las soluciones no seguirán mejorando.

3.3.9. RUIDO EN MEDICIONES

Los sismogramas presentan perturbaciones causadas por lo que se llama *ruido de fondo*. Este ruido suele ser de tipo muy diverso y depende en gran parte de la situación de la estación y del tipo de sismómetros usados. Las perturbaciones más frecuentes se deben a causas atmosféricas, electromecánicas y culturales.

Tradicionalmente el ruido ha sido considerado como algo negativo, molesto y que hay que cancelar o evitar. En la actualidad, lo que hasta hace pocas décadas era un estorbo en los estudios de propagación y atenuación de ondas sísmicas se ha convertido en una herramienta excelente para caracterizar los efectos locales y las propiedades dinámicas del lugar de registro y de las estructuras. Así pues, para los estudiosos de la vibración ambiental, el ruido es *la señal* y las ondas que lo integran se suelen llamar en inglés, “*microtremors*”; en castellano no hay uniformidad en el nombre aplicado y se usan con el mismo significado las palabras *microtemblores*, *microvibraciones*, *vibración ambiental* y *ruido cultural*. Fundamentalmente son vibraciones de muy pequeña amplitud y periodos comprendidos entre 0.1 y 10 s (Architectural Institute of Japan, 1993).

La vibración ambiental se genera principalmente por la actividad humana y se debe al funcionamiento de maquinaria industrial y al tráfico de vehículos (Kanai y Tanaka, 1961). Además también contiene vibraciones producidas por el viento, que son introducidas en el subsuelo por árboles, edificaciones o por su impacto con el relieve (Seo, 1995 a). Su fuente no está asociada a terremotos y algunos investigadores consideran que el ruido de origen natural está formado por ondas superficiales generadas en zonas de interacción océano-continente, modos fundamentales de vibración del planeta, cambios en la presión atmosférica y actividad volcánica interna, además de las fuentes artificiales anteriormente citadas.

La determinación de características dinámicas del suelo utilizando vibración ambiental fue grandemente impulsada gracias a la técnica de Nakamura

(Nakamura, 1989) que en la actualidad constituye una poderosa herramienta para determinar los espesores de los sedimentos (Seht y Wohlenberg, 1999) y ha demostrado ser una de las técnicas más baratas y convenientes para estimar confiablemente las frecuencias predominantes de los depósitos de suelos blandos. Por otra parte su utilización también ha sido bastante controvertida, pero hay consenso en que la técnica es capaz de determinar bien los periodos predominantes del movimiento del suelo pero no identifica bien los factores de amplificación (Bard, 1997).

3.4. ESTRUCTURA PARA ENTORNO MATLAB CON ALGORITMOS GENÉTICOS

Los parámetros utilizados en la mejor estructura fueron seleccionados mediante la prueba de los parámetros y sus resultados con un caso particular, dando como mejor resultado la siguiente estructura seleccionada para el desarrollo del presente trabajo.

Parámetros de entrada para alimentar el proceso y definir la estructura de trabajo del algoritmo genético:

Tomado de, ALGORITMOS GENÉTICOS,
Natyhelem Gil Londoño, Escuela de estadística, UNIVERSIDAD NACIONAL DE COLOMBIA, 2006.

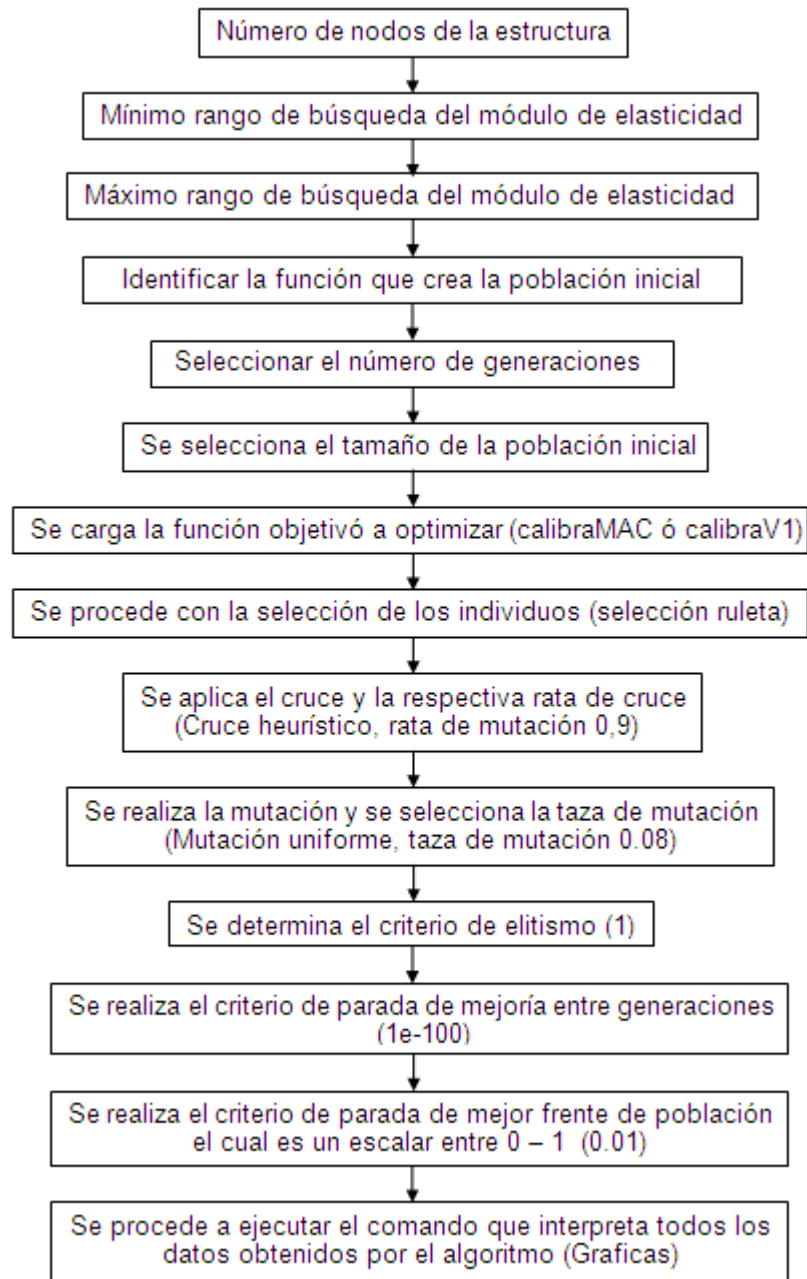


Figura 3.2. Estructura de operación del algoritmo genético (AG).

4. PROCEDIMIENTO GENERAL

4.1. METODOLOGÍA

La respuesta dinámica de las estructuras, frecuencias y formas modales, se determina usando el software de libre distribución OPENSEES[®]. El problema de la detección de fallas propuesto en el desarrollo de la presente investigación se implementa bajo las siguientes consideraciones:

- Se considerarán variaciones entre 0 - 60% del módulo de elasticidad.
- Se asume que la respuesta de la estructura se encuentra en un régimen lineal.
- La estructura se considera sin amortiguamiento (en estructuras civiles éste es despreciable).

Para el estudio de la condición en falla, el daño a la estructura es emulado disminuyendo el valor del módulo de elasticidad a uno o varios elementos de la estructura a través de un algoritmo de MATLAB[®]. Esta nueva versión de la estructura se introduce a OPENSEES, el cual determina las características dinámicas. Estas características serán utilizadas, entonces, para realizar el procedimiento inverso, es decir, a partir de ellas será determinada la rigidez (módulos de elasticidad) de cada elemento usando Algoritmos Genéticos, la cual es comparada con la original y a partir de este residuo se diagnostica las estructuras.

La selección de los parámetros asociados al Algoritmo Genético tiene un impacto significativo en el desempeño del algoritmo. El desempeño del esquema de detección propuesto se realiza en distintos escenarios en los cuales se varía: tipo

de daño, daño simple (un sólo elemento) y daños múltiples (más de un elemento); las funciones objetivos utilizadas; presencia de ruido en las mediciones y cantidad de modos de vibración.

La metodología consiste en:

- 1) Establecer daños a la estructura a partir de la reducción de los módulos de la elasticidad de los elementos, y evaluar las frecuencias y formas modales de la estructura damnificada $\{\omega_m, \Phi_m\}$.
- 2) Establecer de forma aleatoria los módulos de elasticidad de los elementos y determinar las características dinámicas para las diferentes soluciones de módulos de elasticidad usando OPENSEES $\{\omega_a, \Phi_a\}$. Estas características dinámicas son comparadas con las características de la estructura damnificada por medio de la función objetivo. Luego, los valores de la función objetivo son devueltos al Algoritmo Genético para que produzca nuevas soluciones.
- 3) El paso 2 se repite iterativamente hasta que se cumpla un error mínimo considerado en la función objetivo o se alcance un máximo de iteraciones.

El diagrama de flujo del esquema de la técnica implementada se muestra en la Figura 4.1.

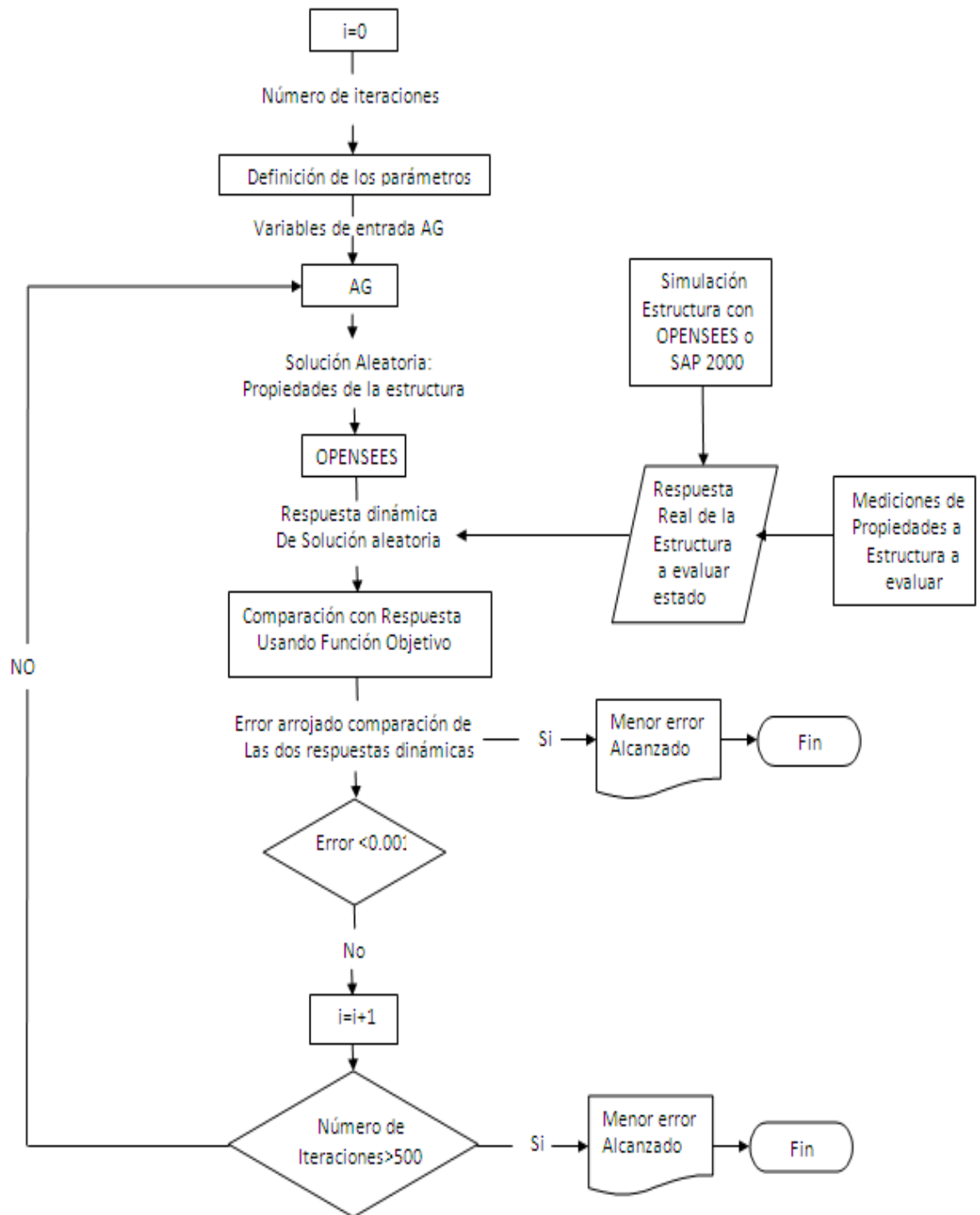


Figura 4.1. Diagrama de flujo del sistema.

4.2. ESTRUCTURAS ANALIZADAS

La metodología de detección de daño ha sido estudiada y aplicada a dos tipos de sistemas estructurales planos: viga y armadura. Las estructuras tomadas como ejemplos son mostradas en las Figuras 4.2 y 4.3.

Viga: En ingeniería y arquitectura se denomina viga a un elemento constructivo lineal que trabaja principalmente a flexión. En las vigas la longitud predomina sobre las otras dos dimensiones y suele ser horizontal. La viga a analizar posee las siguientes características:

Viga simplemente apoyada (Figura 4.2), con longitud: $L=6$ m. 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.

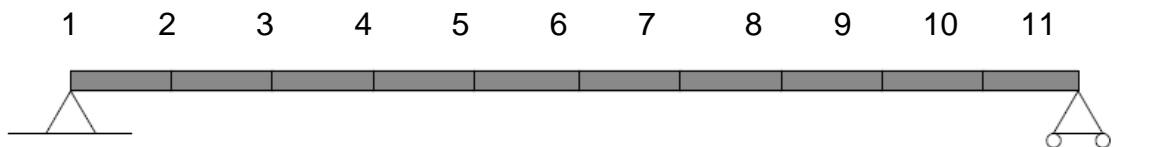


Figura 4.2. Viga simplemente apoyada.

Armadura: Una armadura es un sistema estructural reticular de barras rectas interconectadas en nudos articulados formando triángulos. Los elementos conforman, comúnmente, uno o varios triángulos en un solo plano y se disponen de forma tal que las cargas externas se aplican a los nudos, por lo que en teoría, sólo causan efectos de tensión o de compresión. En la realidad, algunos esfuerzos de flexión pueden ocurrir como resultado de la fricción en las uniones y de cargas distribuidas aplicadas a los miembros entre las juntas (como el peso propio, por

ejemplo); generalmente, estos esfuerzos son menores comparados con las fuerzas axiales y, por lo común, se ignoran para propósitos analíticos. La armadura a analizar es una cercha la cual posee las siguientes características:

Armadura de 13 elementos, simplemente apoyada, cuyas dimensiones pueden apreciarse en la Figura 4.3.

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

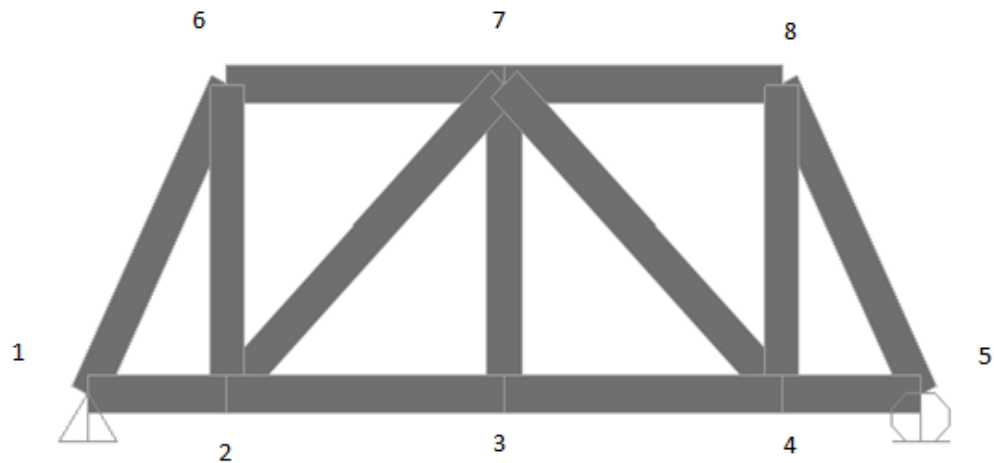


Figura 4.3. Armadura de 13 elementos simplemente apoyada.

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 4.1 y 4.2.

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 4.1. 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	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

Tabla 4.2 Conectividad de los elementos en la estructura tipo Armadura.

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 $(1-\beta)$.

4.3.1 INCLUSIÓN DE RUIDO EN MEDICIONES

La presencia de ruido en las mediciones se considera mediante la introducción de pequeñas perturbaciones en frecuencias naturales y formas modales usando las ecuaciones (4.1- 4.2).

$$f_{nr} = f_n (1 + \text{Rand}(-1,1) * \text{Ruido}_f) \quad (4.1)$$

$$\Phi_{nr} = \Phi_n (1 + \text{Rand}(-1,1) * \text{Ruido}_\phi) \quad (4.2)$$

Donde f_n y Φ_n corresponden a las frecuencias naturales y formas modales sin presencia de ruido respectivamente, f_{nr} y Φ_{nr} son los valores de los mismos parámetros con ruido. Ruido_f y Ruido_ϕ son los porcentajes de ruido introducidos. Las frecuencias naturales y formas modales son contaminadas por ruido con un error promedio de $\pm 0.15\%$ y $\pm 3\%$ 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 4.3. La configuración del ruido fue considerada para representar las condiciones experimentales reales.

CASOS 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

(a)

CASOS DE RUIDO	Ruido en Frecuencia %	Ruido en Modos %
R1 _A	0	0
R2 _A	0.1	1
R3 _A	0.15	3
R4 _A	0.2	5

(b)

Tabla 4.3. Configuraciones de ruido que serán estudiadas (a) Viga y (b) Armadura

4.3.2 CASOS DE DAÑO

Diversos escenarios de daño para ambas estructuras analizadas viga y armadura son considerados para la validación del desempeño de las metodologías estudiadas, como es presentado en la Tabla 4.4.

NUMERO DE MODOS	DAÑO SIMPLE	DAÑO MÚLTIPLE 2 ELEMENTOS	DAÑO MÚLTIPLE 4 ELEMENTOS
4	✓	✓	✓
6	✓	✓	✓
8	✓	✓	✓

(a)

NUMERO DE MODOS	DAÑO SIMPLE	DAÑO MÚLTIPLE 2 ELEMENTOS
4	✓	✓
5	✓	✓
6	✓	✓

(b)

Tabla 4.4. Escenarios de daño estudiados para: (a) Viga y (b) Armadura.

4.4. CONFIABILIDAD DE LOS VALORES OBTENIDOS POR OPENSEES UTILIZANDO COMO COMPROBACIÓN SAP 2000

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 4.4 y 4.7):

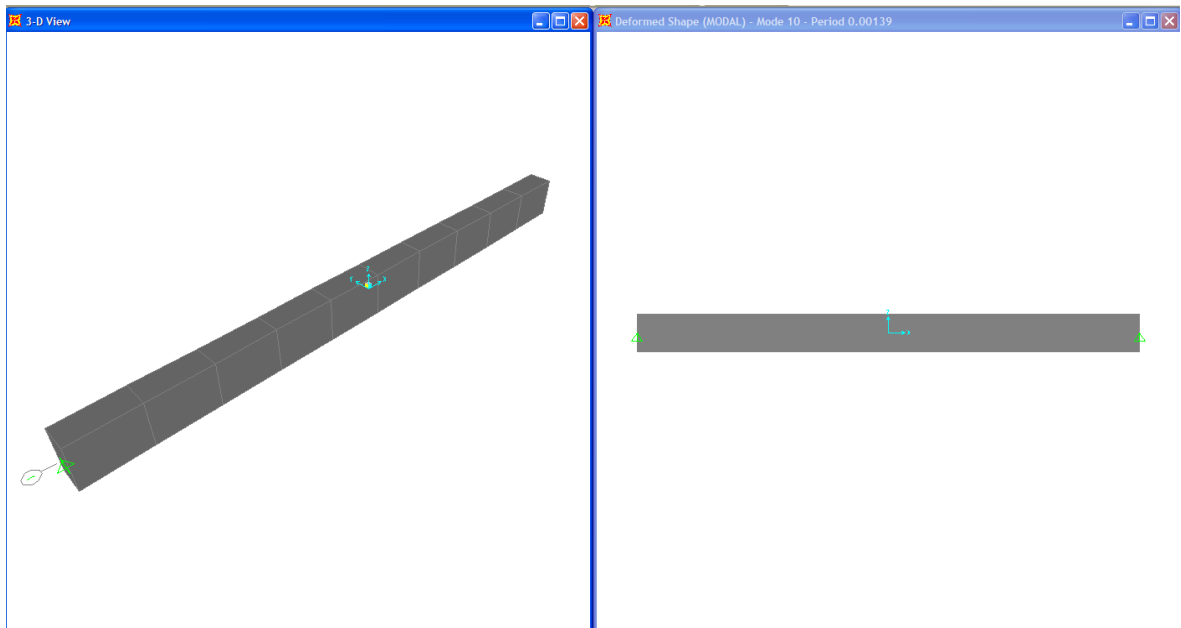


Figura 4.4. Modelo viga usando SAP2000

ELEMENTO	EIGENVALORES	FRECUENCIAS	PERIODOS
1	8.98E+03	15.0816789	0.066306
2	1.44E+05	60.3199754	0.016578
3	7.26E+05	135.644743	0.007372
4	2.29E+06	240.711789	0.004154
5	2.44E+06	248.426133	0.004025
6	5.53E+06	374.307524	0.002672
7	9.51E+06	490.735188	0.002038
8	1.12E+07	532.86753	0.001877
9	1.97E+07	707.269597	0.001414
10	2.05E+07	720.960649	0.001387

Tabla 4.5. Resultados de frecuencias y periodos OPENSEES.

Los resultados de frecuencias y periodos obtenidos con OPENSEES (Tabla 4.5) 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.

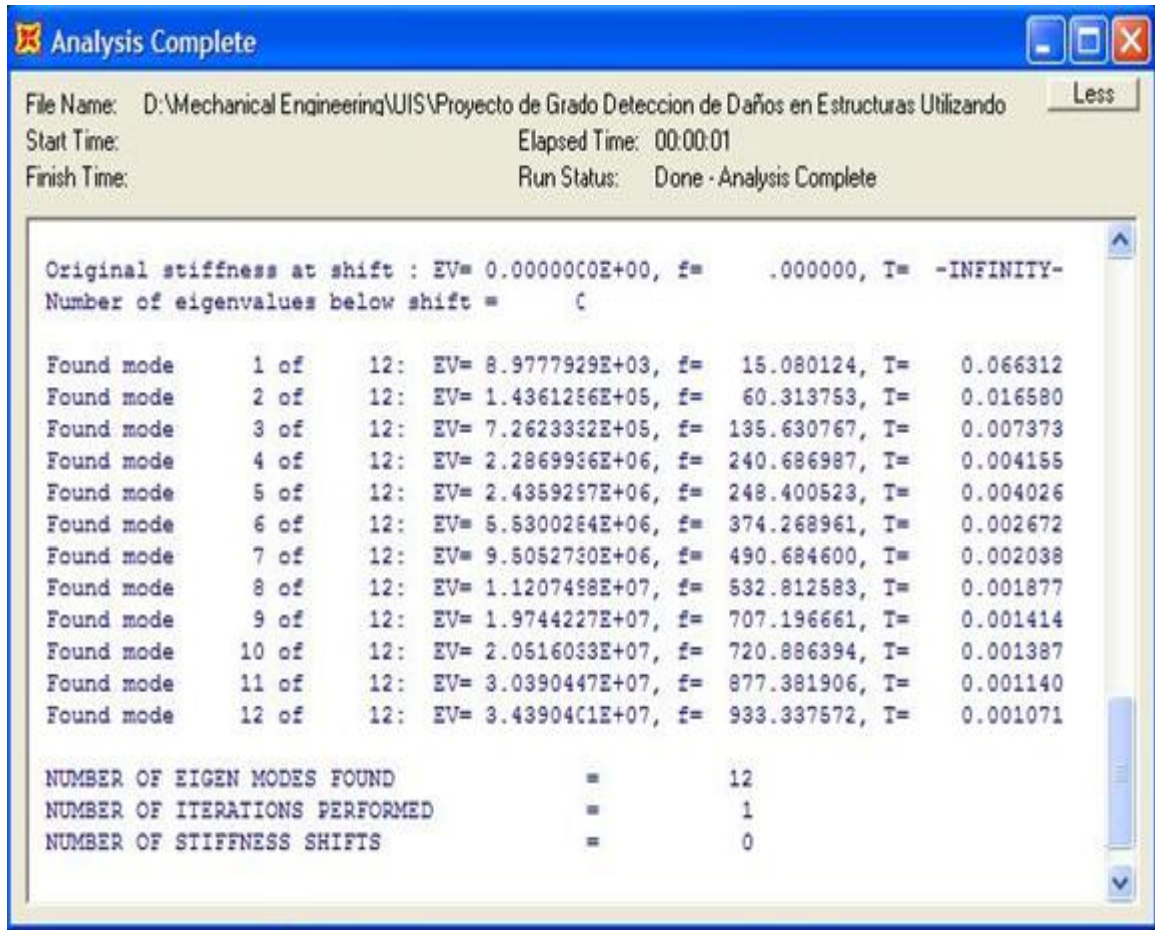


Figura 4.5. Valores de periodos y frecuencias hallado por SAP2000 para la viga.

De la misma forma, SAP 2000 nos permite apreciar las diversas formas modales (Figuras 4.6 y 4.8):



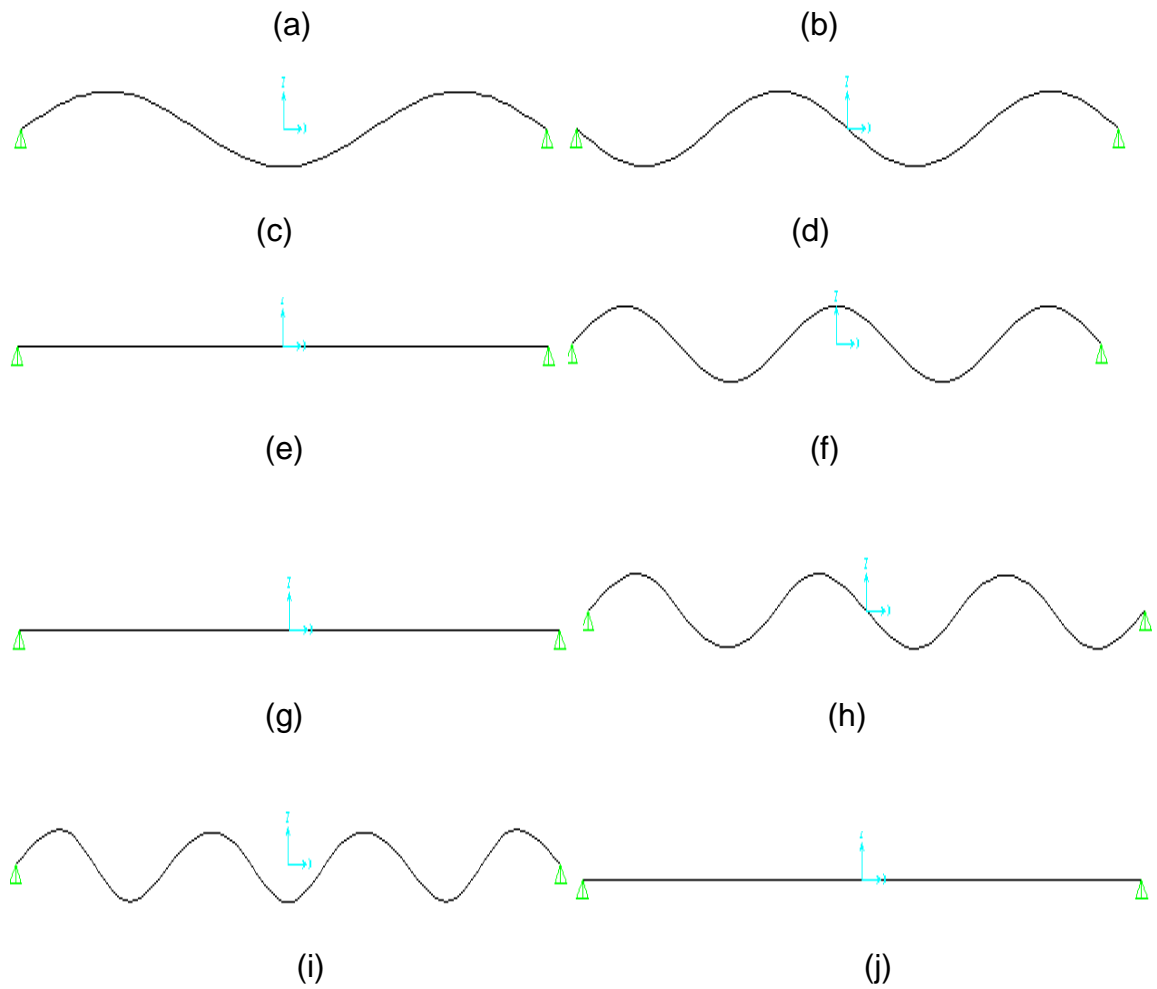


Figura 4.6. Modos de Vibración Viga (a) 1 modo, (b) 2 modos, (c) 3 modos, (d) 4 modos, (e) 5 modos, (f) 6 modos, (g) 7 modos, (h) 8 modos, (i) 9 modos, (j) 10 modos

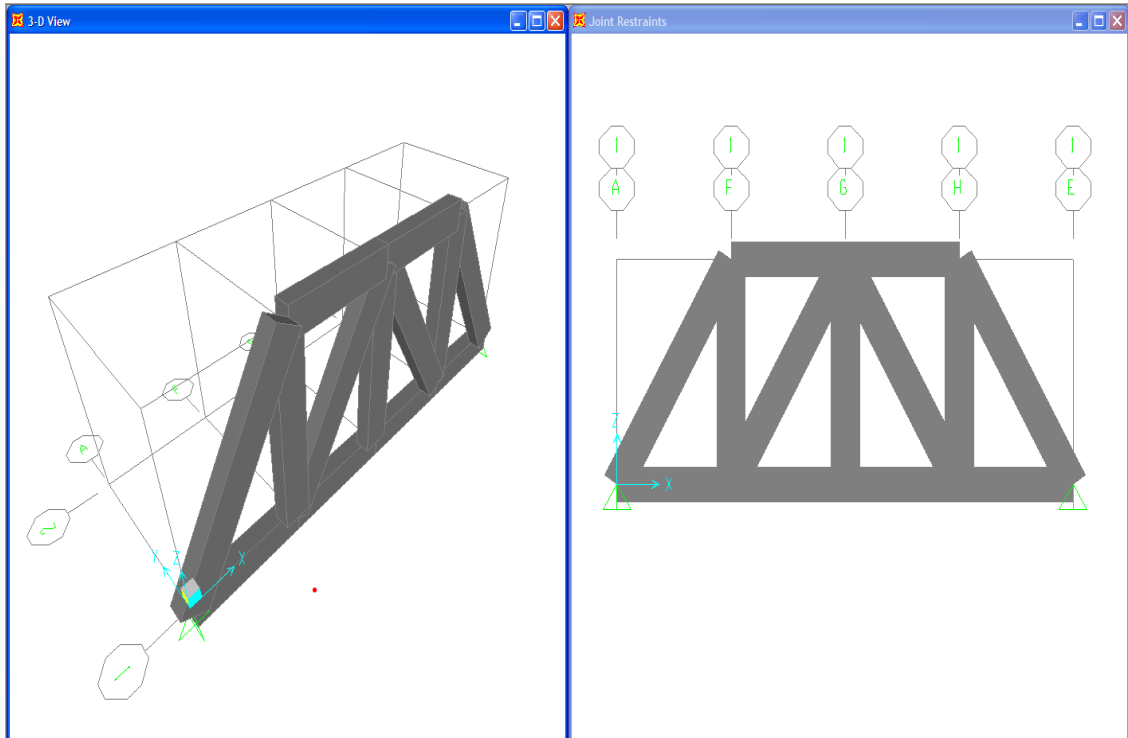
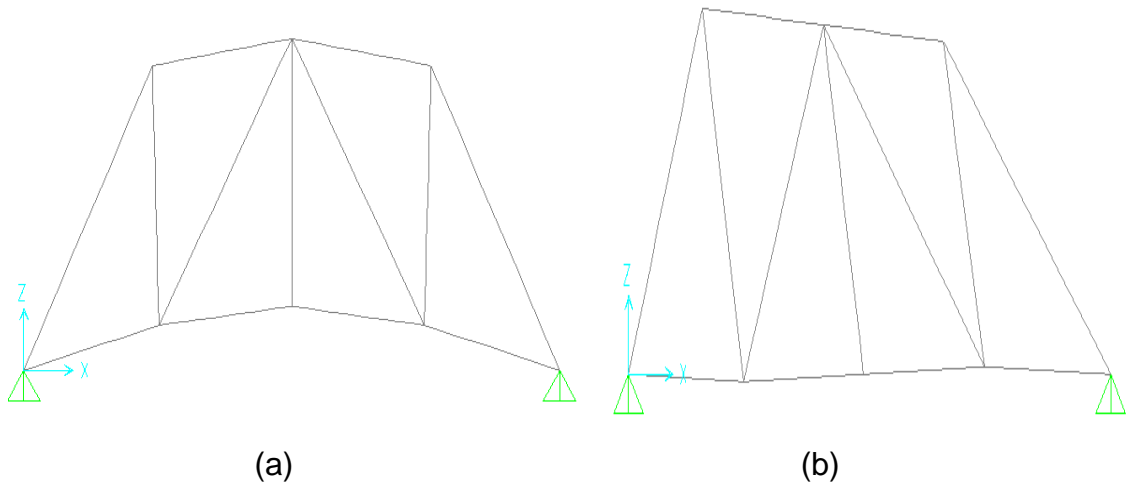
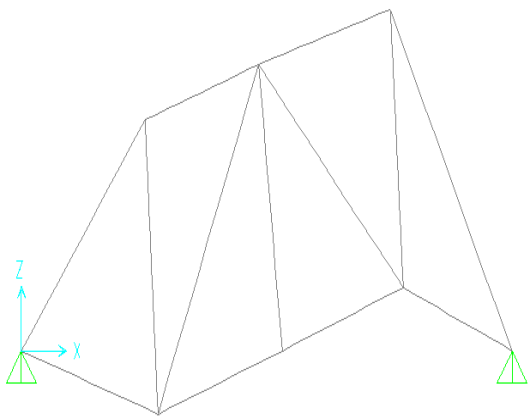
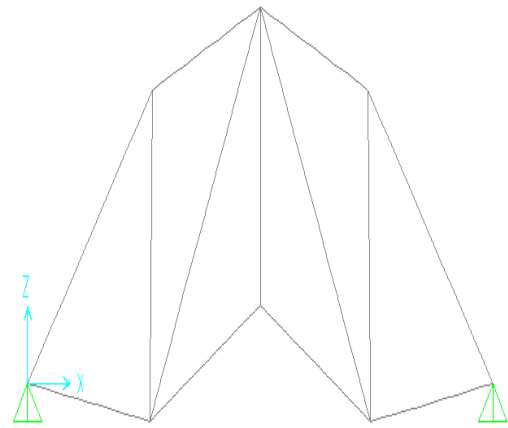


Figura 4.7. Modelo armadura usando SAP2000.

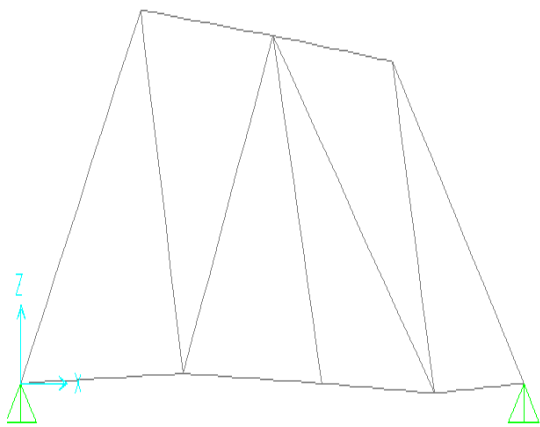




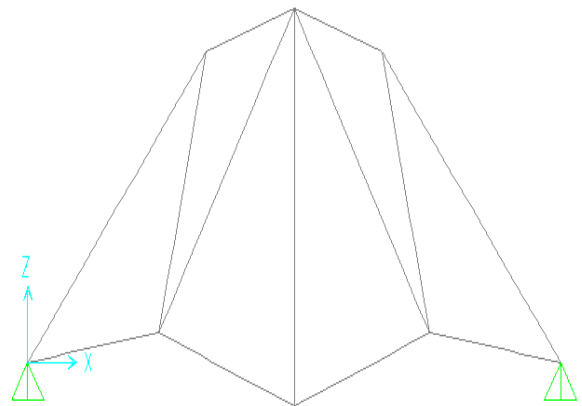
(c)



(d)



(e)



(f)

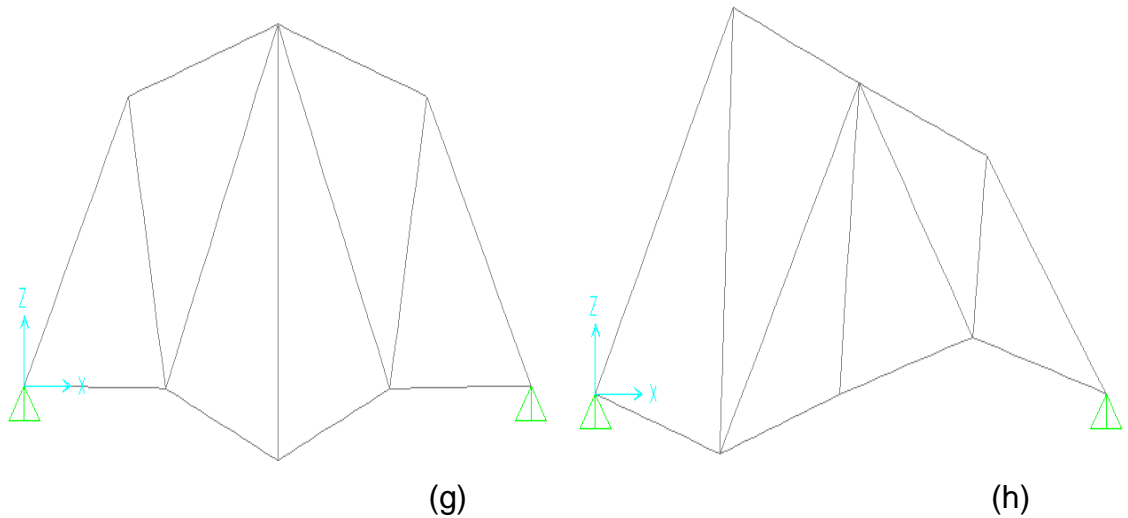


Figura 4.8. Modos de Vibración Armadura (a) 1 modo, (b) 2 modos, (c) 3 modos, (d) 4 modos, (e) 5 modos, (f) 6 modos, (g) 7 modos, (h) 8 modos.

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 la técnica de optimización Algoritmos Genéticos a través de la función objetivo.

4.4.1 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. 2.9 y 2.10), las cuales se encargan de comparar los valores de frecuencias y formas modales reales, con los valores arrojados por el Algoritmo; además en estos archivos para el caso de la viga **CalibraMACviga.m**, **CalibraV1viga.m**; y para la armadura de 13 elementos **CalibraMACarmadura.m**, **CalibraV1armadura.m** en donde se especifican la cantidad de modos de vibración que se utilizan para esta comparación. Por medio de la utilización de Algoritmos Genéticos en entorno MATLAB, 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 necesarios para condicionar el funcionamiento del Algoritmo Genético (Ver Capítulo 3 para más detalles de la utilización de Algoritmo Genético en MATLAB). Una vez se ejecute el archivo principal del Algoritmo Genético una interface gráfica muestra el mejor individuo y el error global alcanzado en la función objetivo vs el número de generaciones (ver figura 4.9 y figura 4.10.); al final obtenemos los valores de los módulos de elasticidad alcanzados junto con el mínimo error de la función objetivo

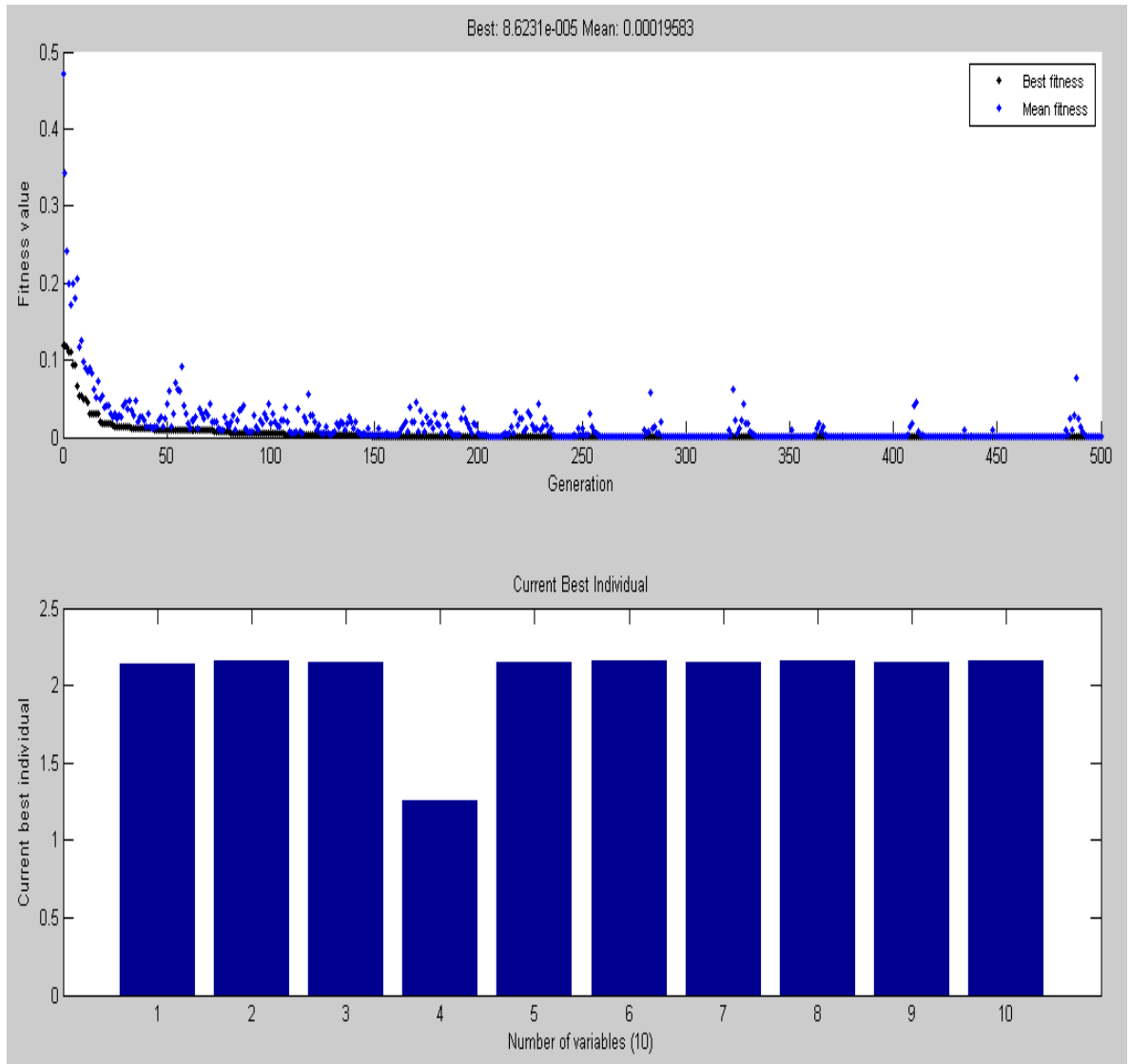


Figura 4.9. Interfaz Gráfica para el caso de la viga con la función objetivo CalibraV1viga.m implementada en Algoritmos Genéticos para el entorno MATLAB.

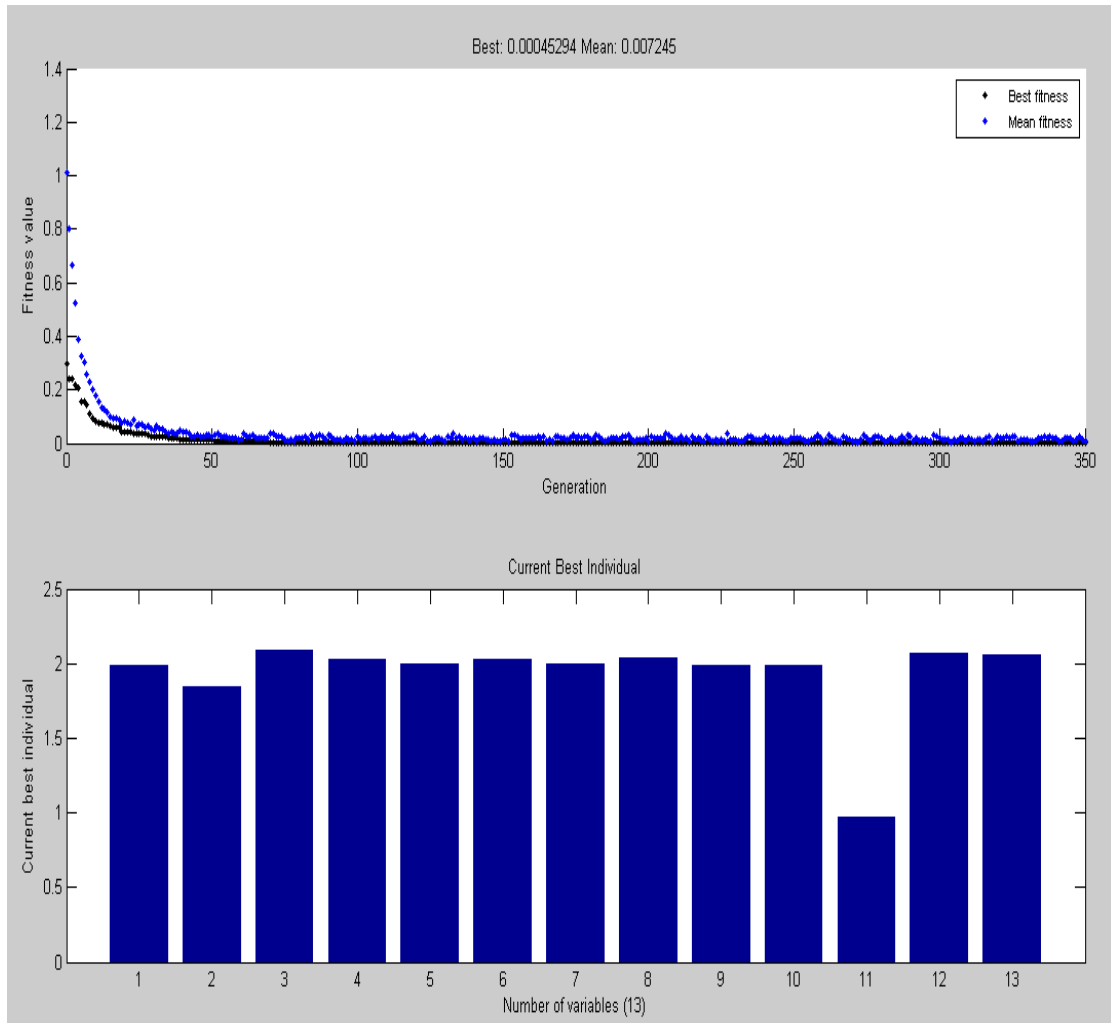


Figura 4.10. Interfaz Gráfica para el caso de la armadura con la función objetivo CalibraMACarmadura.m implementada en Algoritmos Genéticos para el entorno MATLAB.

Para simular el daño múltiple en las estructuras simplemente en el algoritmo de inclusión de esta reducción del módulo de elasticidad, se especifican la cantidad de elementos dañados deseados. Por último 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

En el proceso de la determinación de los elementos dañados en cada caso en particular (Viga, armadura) se noto que el elemento dañado tenía un porcentaje de daño superior al 5 % debido a que daños inferiores no son significativos en la detección de daños en el estudio de la salud estructural; también se tiene en cuenta la distinción de dos conceptos fundamentales en los análisis de los resultados, ya que un punto fundamental del análisis es **detectar el daño** y el otro caso es **diagnosticar el daño**, el primer caso es cuando el algoritmo detecta cual es el elemento dañado real o elementos dañados reales sin tener en cuenta el porcentaje de error y el segundo caso es cuando detecta el o los elementos damnificados pero también lo hace con una precisión de error menor al $\pm 5 \%$ del daño real; también se identificó un punto óptimo para utilizar la técnica para cada estructura analizada, en el cual se noto que no se deben utilizar ni muy pocos modos de vibración ni muchos modos de vibración en el análisis, lo que se recomienda es un intervalo que oscile entre 3 modos y 6 modos de vibración. A continuación mostramos los resultados obtenidos por la técnica de optimización algoritmos genéticos en un paralelo con los resultados que se obtuvieron con la técnica de optimización PSO*, para poder tener una perspectiva del comportamiento de cada una de las técnicas en los diferentes episodios de daño y con cada uno de los porcentajes de ruido que se implementaron.

*Proyecto de grado, Detección de daños en estructuras utilizando la técnica de optimización PSO,
Diego Andrés Ramírez Ramírez.

5.1. IDENTIFICACIÓN DE ELEMENTOS DAÑADOS

5.1.1 FUNCIÓN OBJETIVO CALIBRAV1 (EC. 2.10)

VIGA

a. Daño Simple un elemento; comparando daño real y daño calculado en la viga con los diferentes niveles de ruido (Función objetivo CalibraV1 viga,)

- **4 Modos de Vibración: (Elemento dañado 4 en 41,76%)**

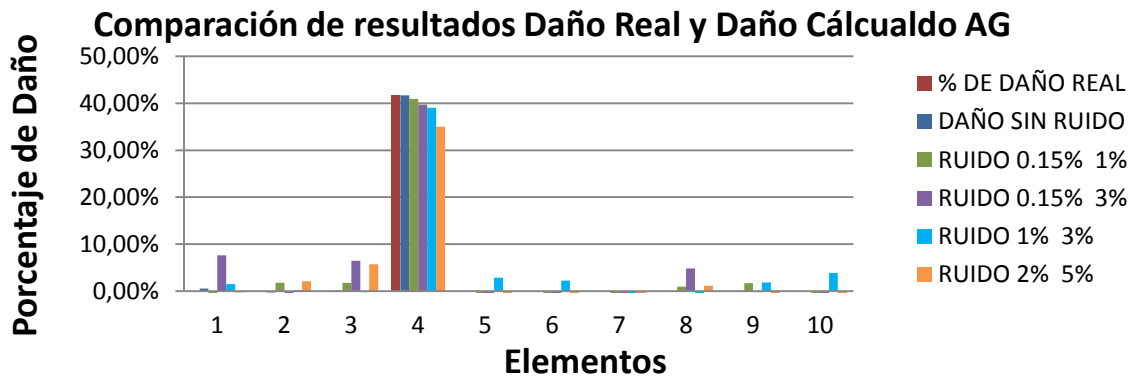


Figura. 5.1. 4 Modos de Vibración: (Elemento dañado 4 en 27,187%)

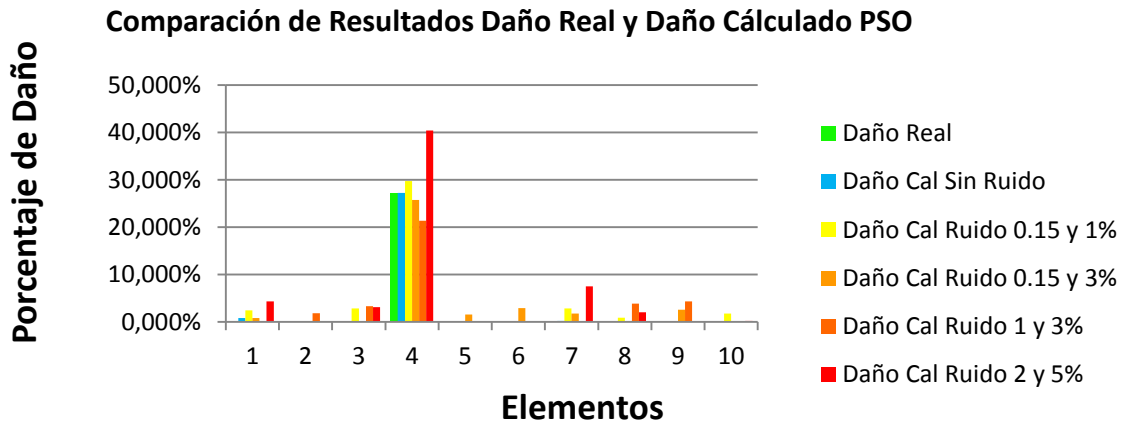


Figura. 5.2. 6 Modos de Vibración: (Elemento dañado 6 en 19,33%)

Comparación de resultados Daño Real y Daño Cálculdo AG

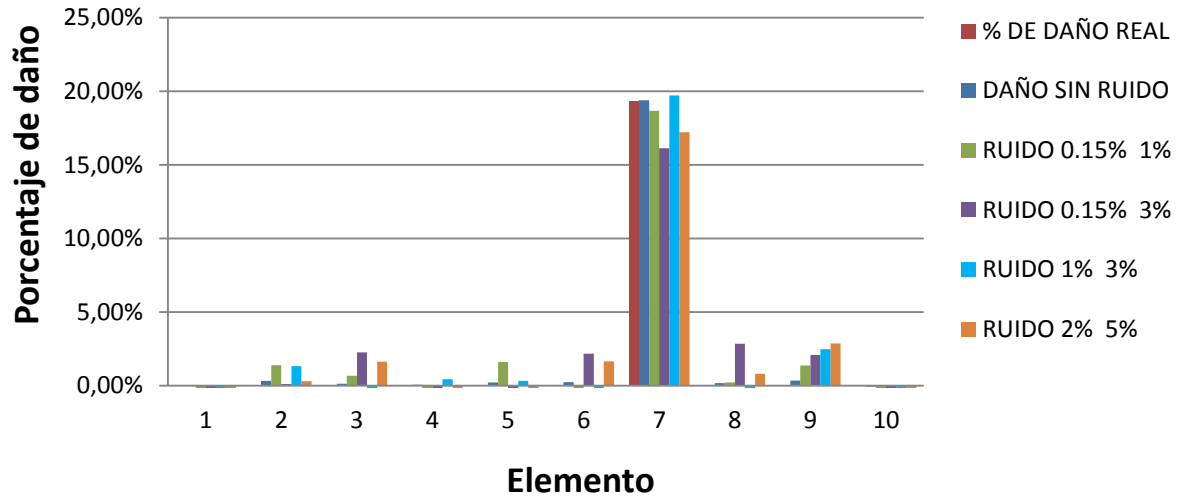


Figura. 5.3. 6 Modos de Vibración: (Elemento dañado 10 en 19,815%)

Comparación de Resultados Daño Real y Daño Cálculdo PSO

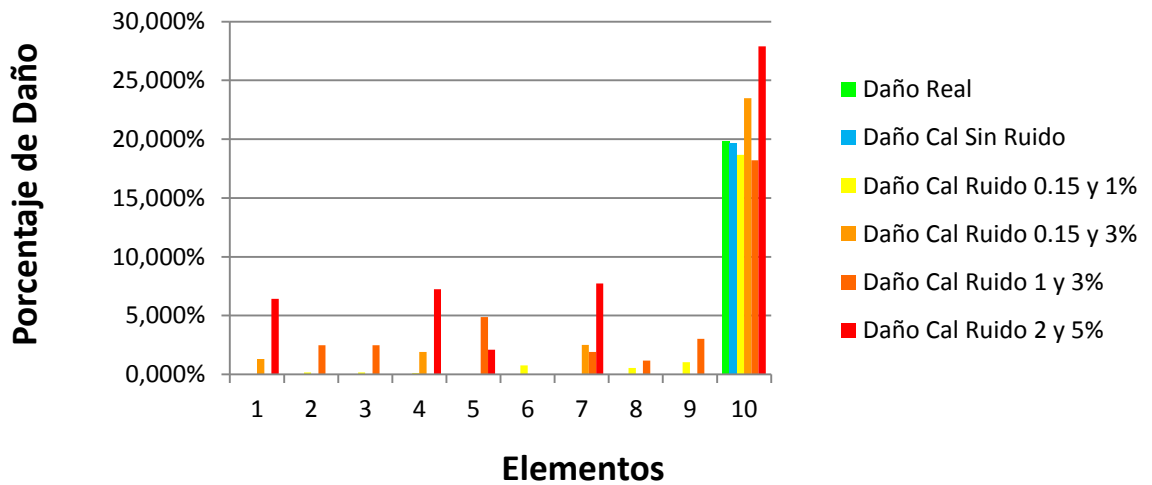


Figura. 5.4. 8 Modos de Vibración: (Elemento Dañado: 8 en 50,15%)

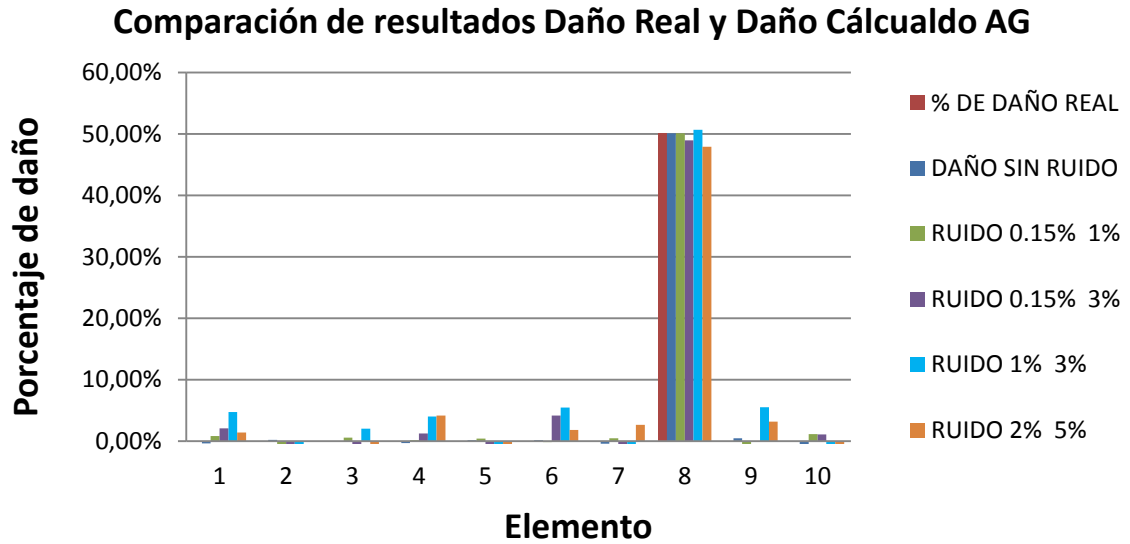
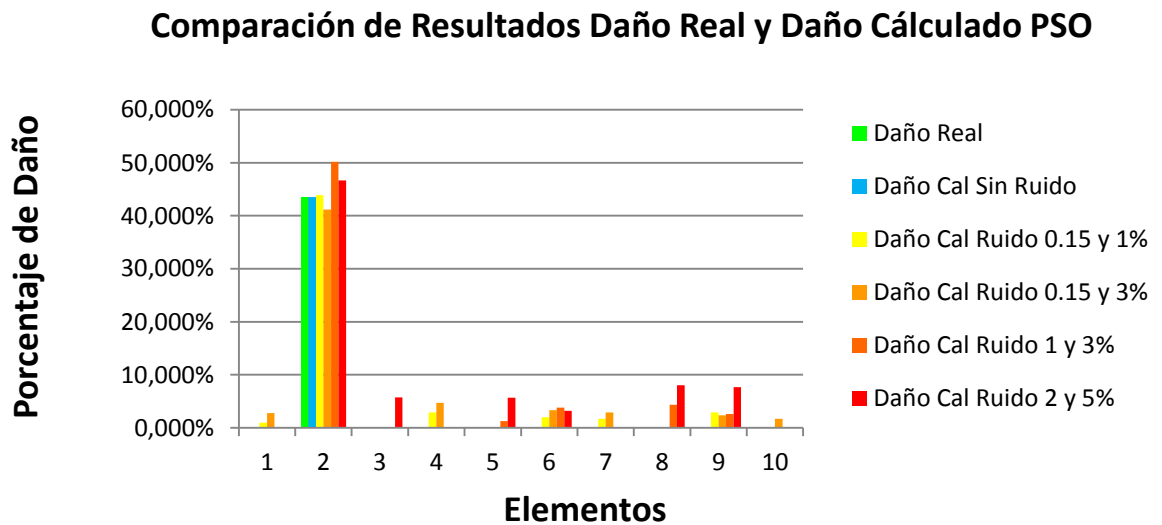


Figura. 5.5. 8 Modos de Vibración: (Elemento Dañado: 2 en 43,48%)



**Figura. 5.6.
b. Daño Múltiple (Dos Elementos)**

- **4 Modos de Vibración: (Elementos Dañados: 3 en 49,01% y 5 en 9,57%)**

Comparación de resultados Daño Real y Daño Cálculo AG

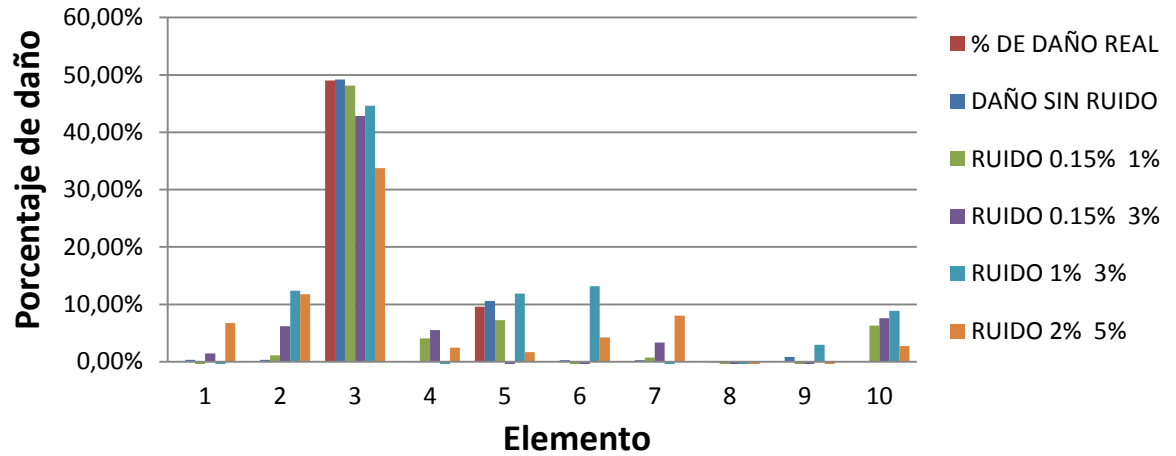


Figura. 5.7. 4 Modos de Vibración: (Elementos Dañados: 7 en 14,53% y 8 en 36,46%)

Comparación de Resultados Daño Real y Daño Cálculo PSO

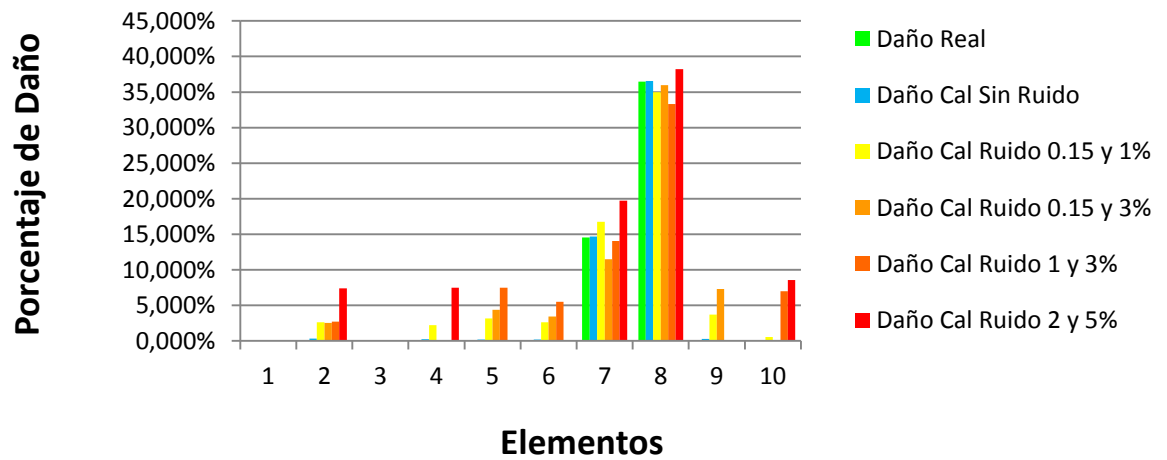


Figura. 5.8. 6 Modos de Vibración: (Elementos Dañados: 4 en 35,91% y 9 en 46,51%)

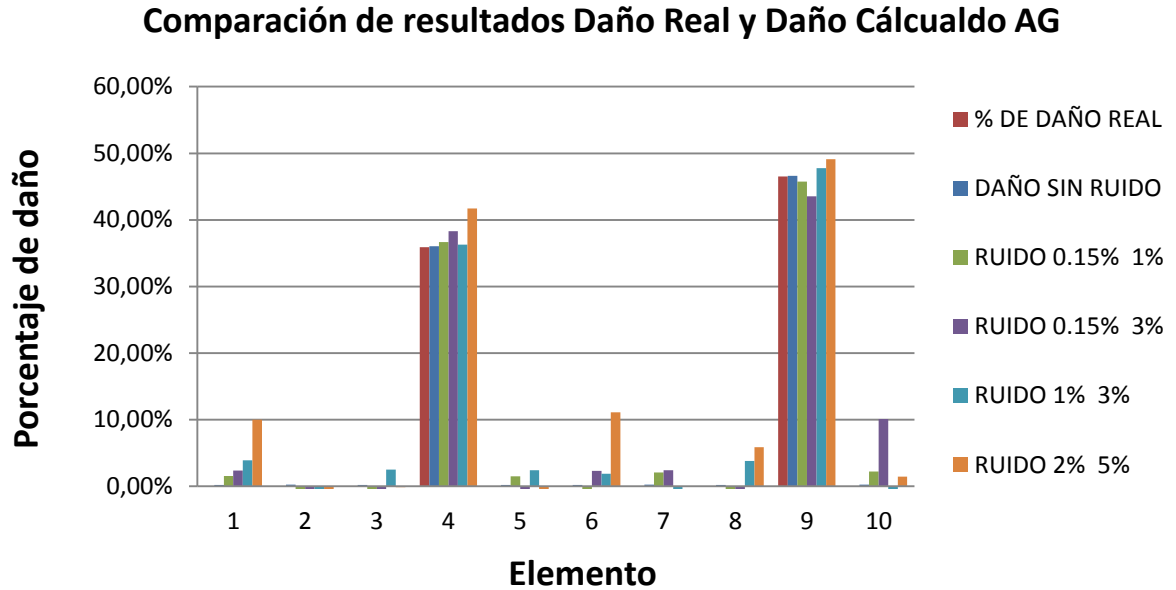


Figura. 5.9.

- **6 Modos de Vibración: (Elementos Dañados: 7 en 57,85% y 9 en 3,96%)**

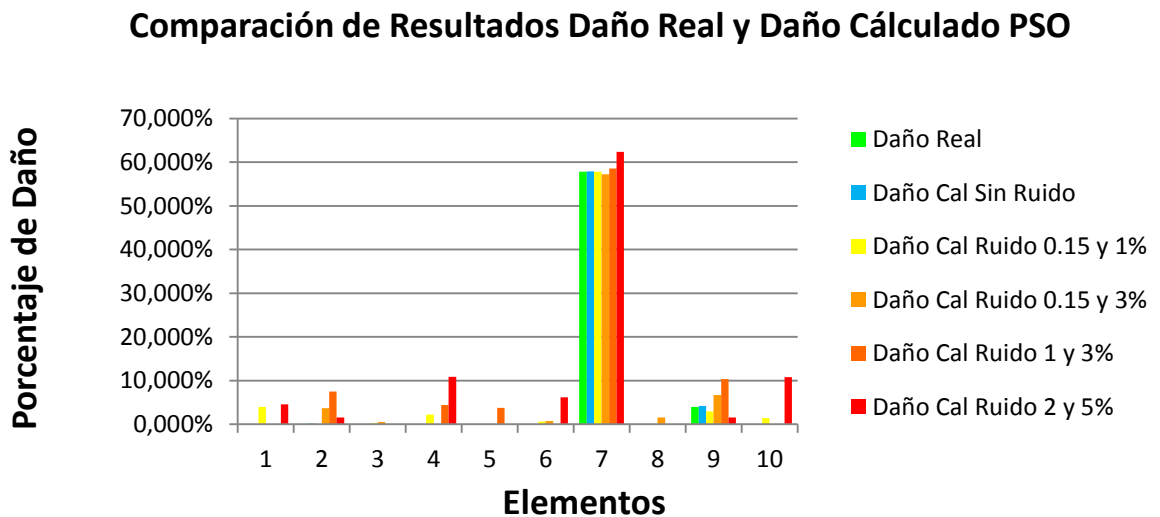


Figura. 5.10.

- **8 Modos de Vibración: (Elementos Dañados: 2 en 40,13% y 3 en 27,00%)**

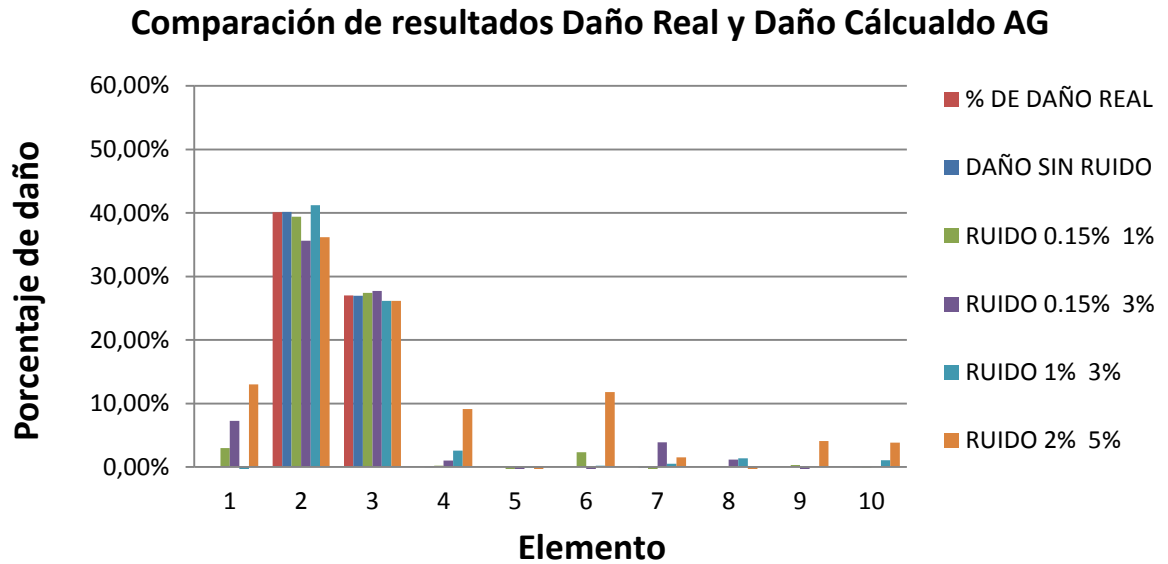


Figura. 5.11. 8 Modos de Vibración: (Elementos Dañados: 8 en 51,48% y 10 en 30,87%)

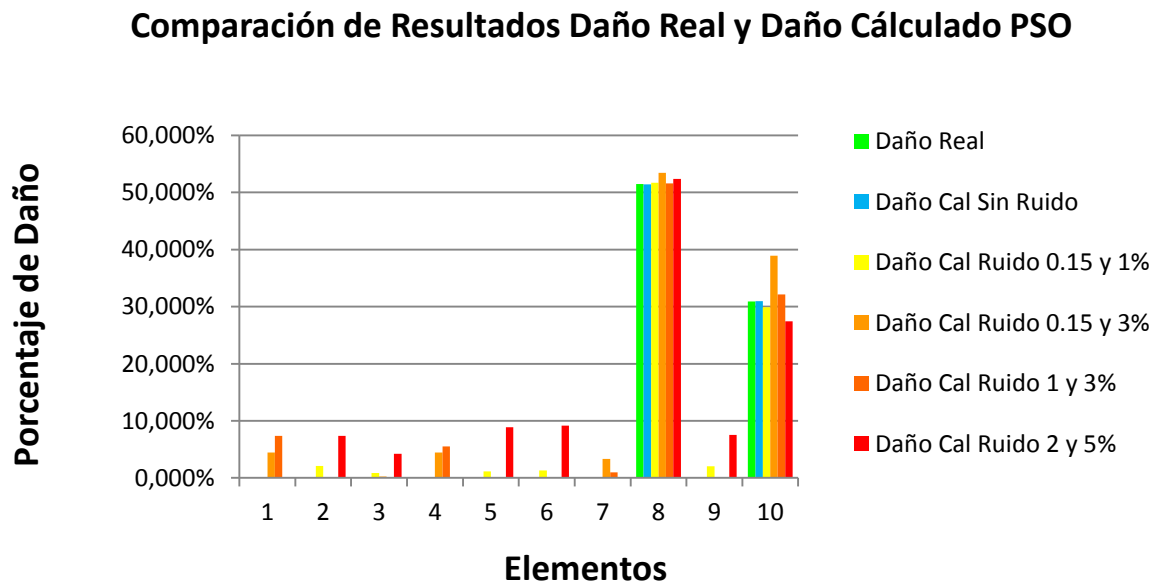


Figura. 5.12. c. Daño Múltiple (Cuatro Elementos)

- **4 Modos de Vibración:(Elementos Dañados: 2 en 24,48%, 6 en 20,95%, 7 en 31,77% y 9 en 14,66%)**

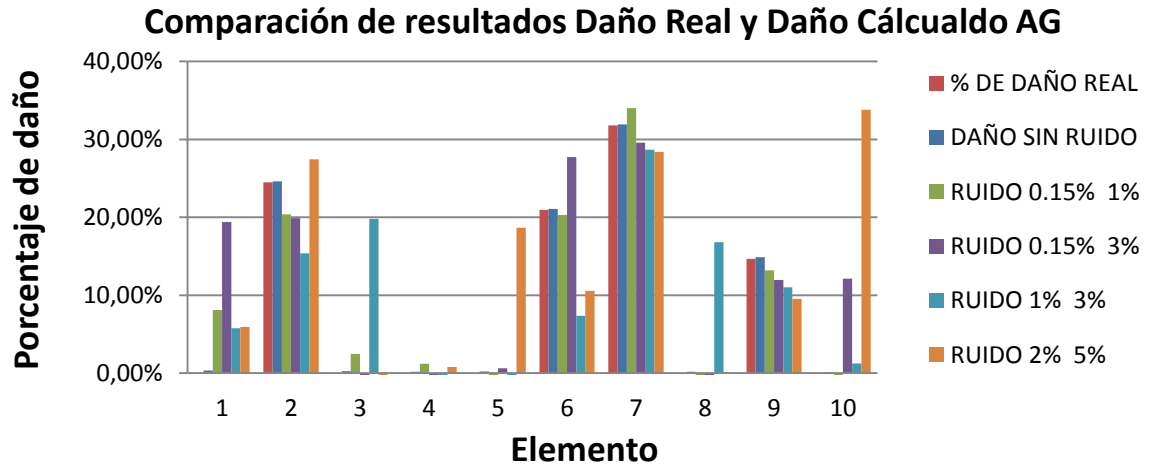


Figura. 5.13.

- **4 Modos de Vibración:(Elementos Dañados: 2 en 53,84%, 3 en 6,13%, 4 en 5,09% y 9 en 27,25%)**

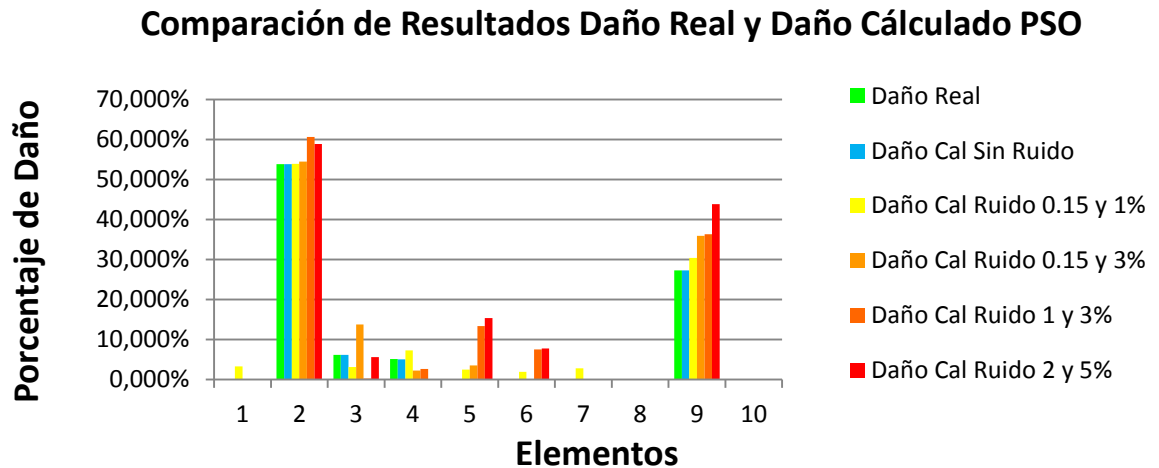


Figura. 5.14.

- **6 Modos de Vibración: (Elementos Dañados: 2 en 5,20%, 4 en 27,19%, 7 en 54,15% y 8 en 5,65%)**

Comparación de resultados Daño Real y Daño Cálculo AG

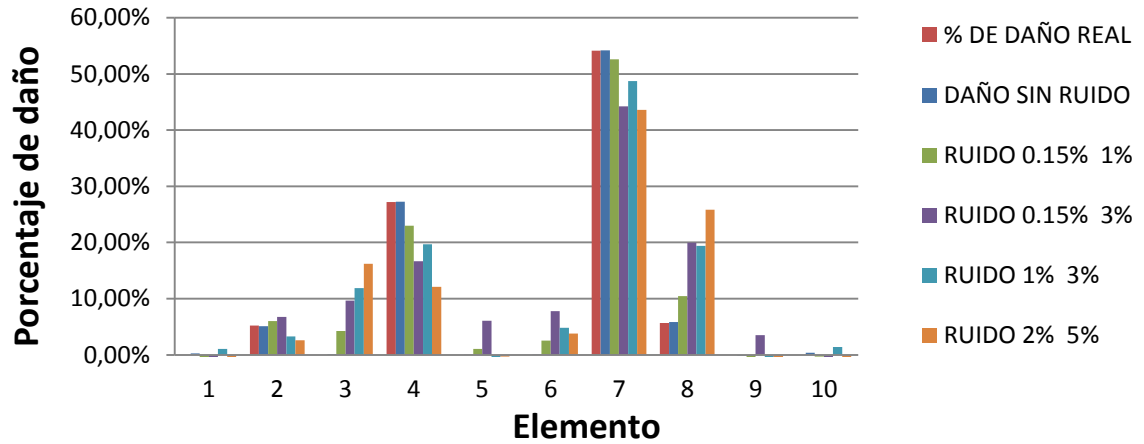


Figura. 5.15.

- **6 Modos de Vibración: (Elementos Dañados: 2 en 19,63%, 5 en 47,50%, 7 en 33,47% y 8 en 54,09%)**

Comparación de Resultados Daño Real y Daño Cálculo PSO

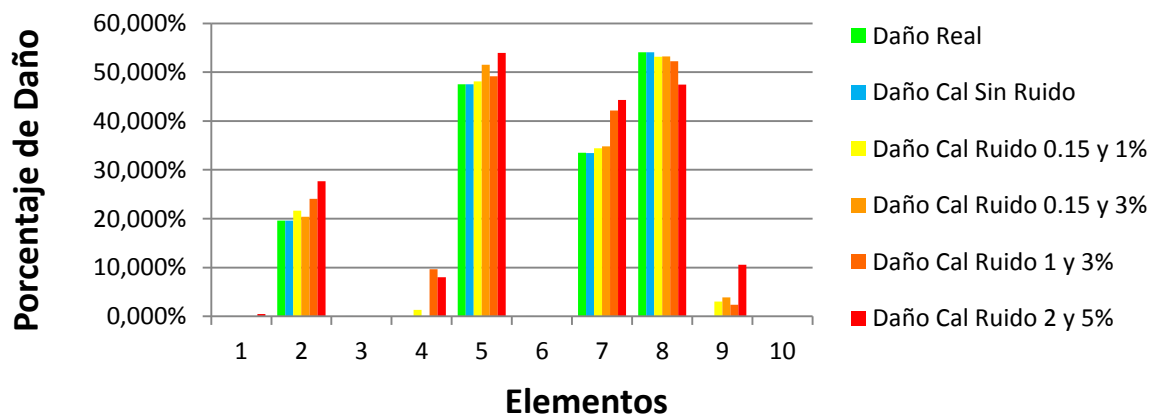


Figura. 5.16. 8 Modos de Vibración:(Elementos Dañados: 2 en 43,92%, 3 en 59,92%, 5 en 59,80% y 8 en 55,55%)

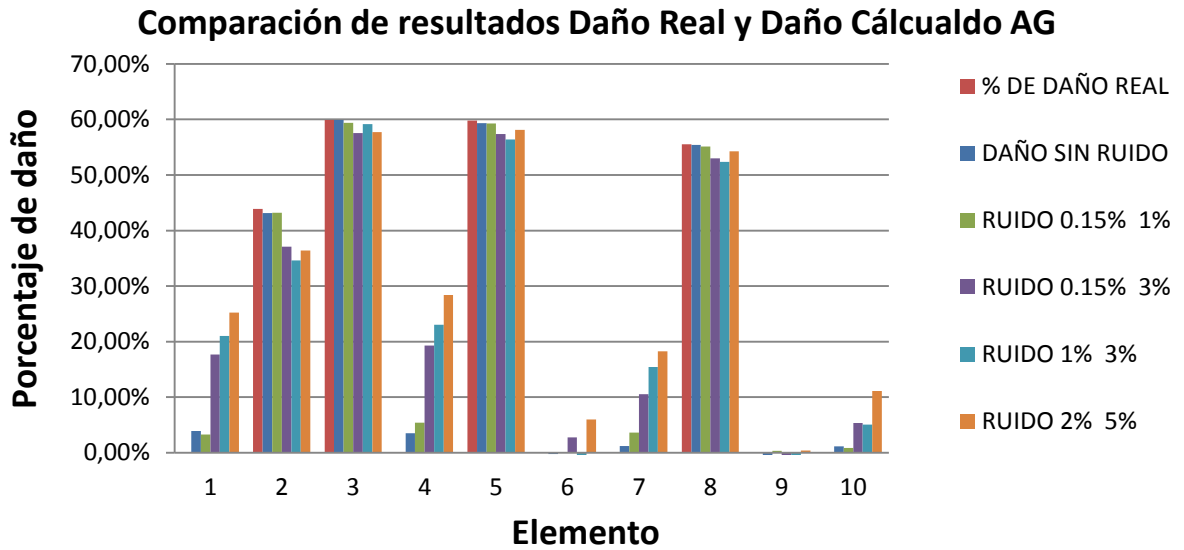


Figura. 5.17.

- **8 Modos de Vibración:(Elementos Dañados: 2 en 5,19%, 4 en 27,18%, 7 en 54,14% y 8 en 5,65%)**

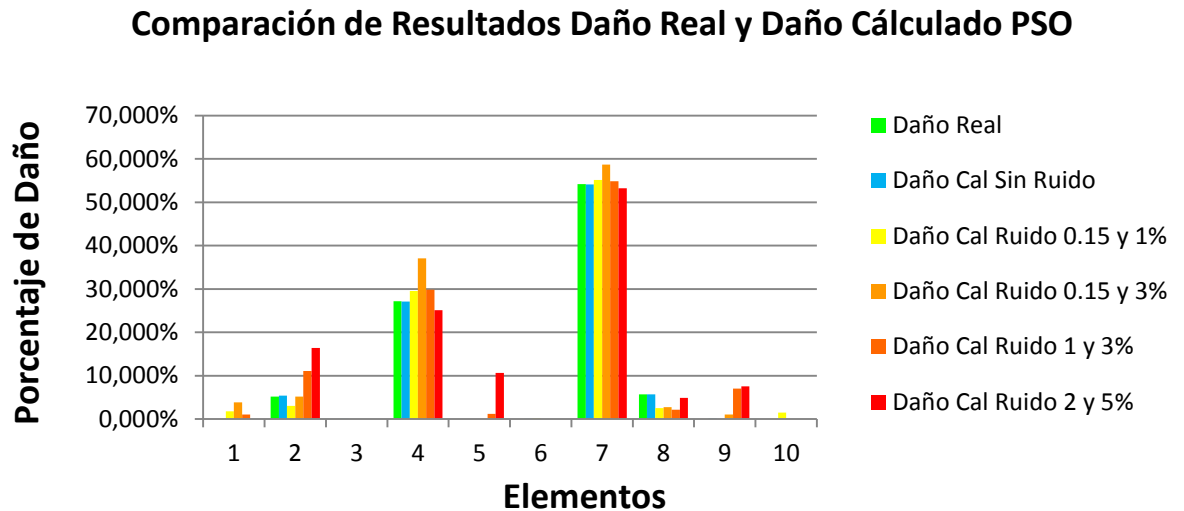


Figura. 5.18.

ARMADURA

a. Daño Simple (Un Elemento) función calibraV1armadura.

- **4 Modos de Vibración:(Elemento Dañado: 9 en 50,00%)**

Comparación de resultados Daño Real y Daño Cálculo AG

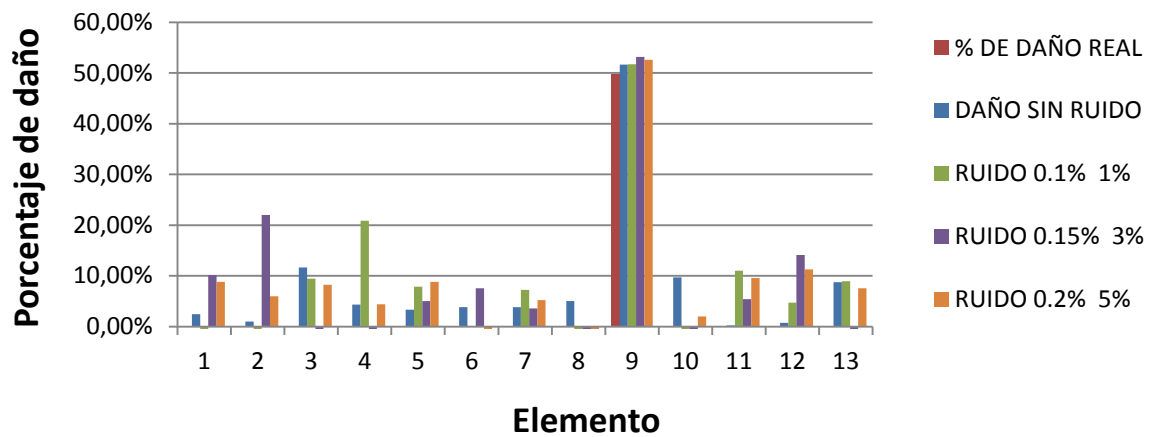


Figura. 5.19.

- **4 Modos de Vibración:(Elemento Dañado: 11 en 5,65%)**

Comparación de Resultados Daño Real y Daño Cálculo PSO

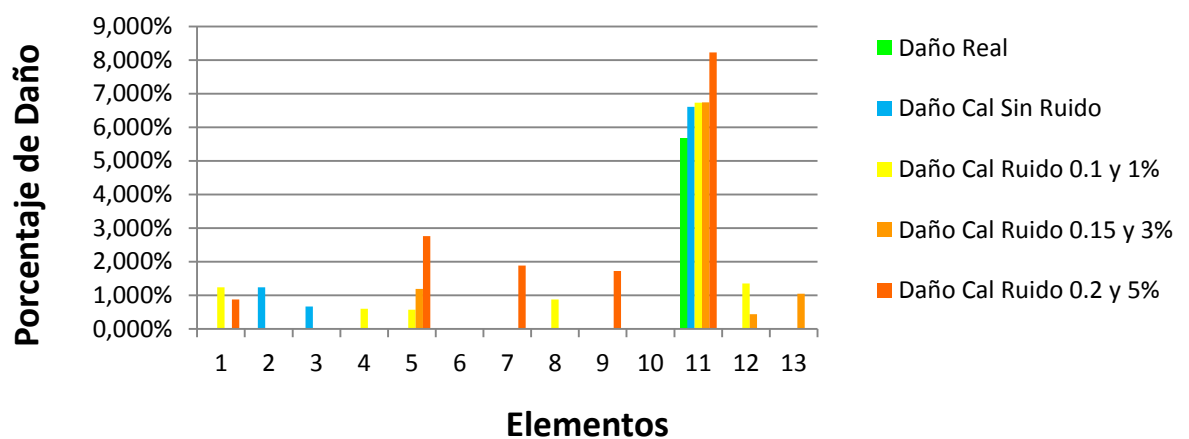


Figura. 5.20. b. Daño Múltiple (Dos Elementos)

- **4 Modos de Vibración: (Elementos Dañados: 7 en 35,56% y 11 en 26,19%)**

Comparación de resultados Daño Real y Daño Cálculdo AG

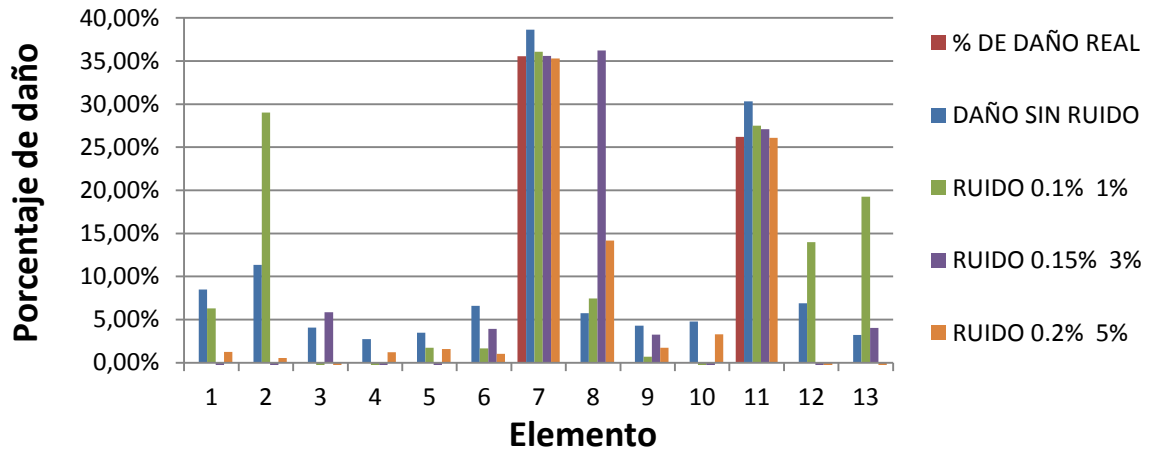


Figura. 5.21.

- **4 Modos de Vibración: (Elementos Dañados: 6 en 5,05% y 11 en 51,48%)**

Comparación de Resultados Daño Real y Daño Cálculdo PSO

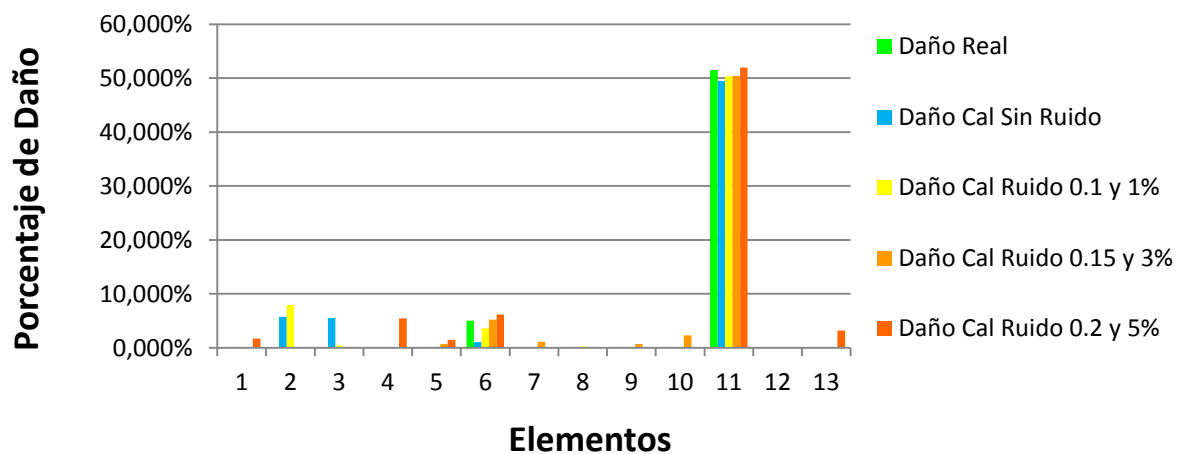


Figura. 5.22.

5.1.2. FUNCIÓN OBJETIVO CALIBRAMAC (EC.2.9)

VIGA

a. Daño Múltiple (Dos Elementos)

- 6 Modos de Vibración:(Elementos Dañados: 6 en 49.53% y 7 en 5.22%)

Comparación de resultados Daño Real y Daño Cálculo AG

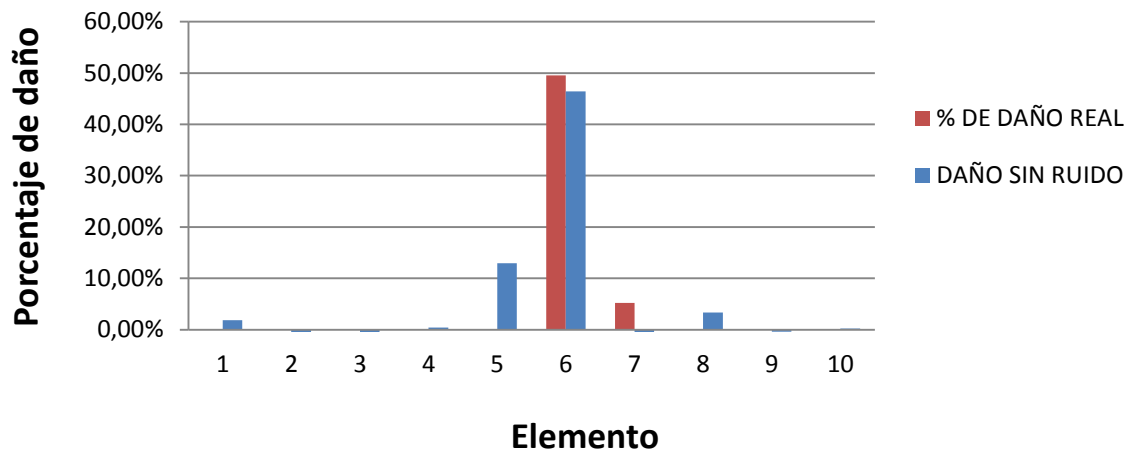


Figura. 5.23.

- 6 Modos de Vibración:(Elementos Dañados: 2 en 5.19% y 8 en 5.65%)

Comparación de Resultados Daño Real y Daño Cálculo PSO

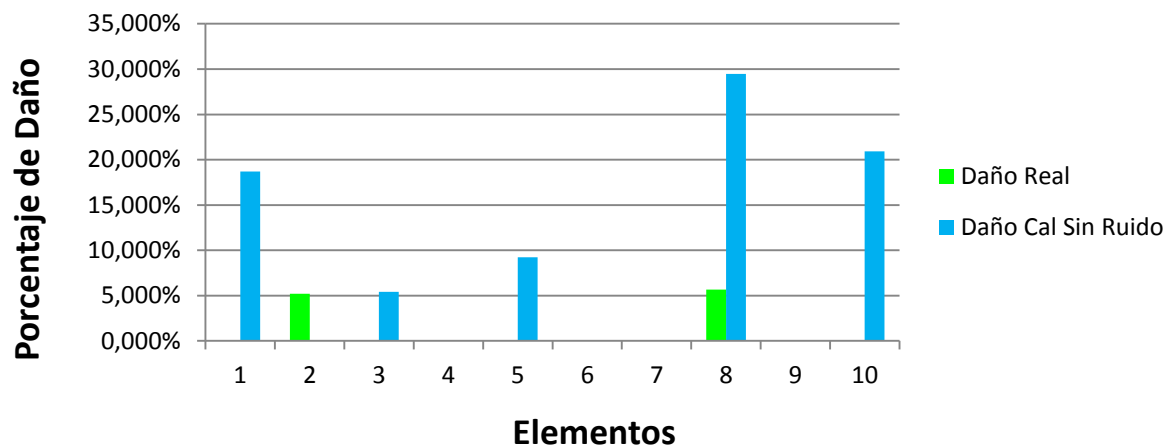


Figura. 5.24.

- **8 Modos de Vibración:(Elementos Dañados: 4 en 51.42% y 6 en 51.77%)**

Comparación de resultados Daño Real y Daño Cálculo AG

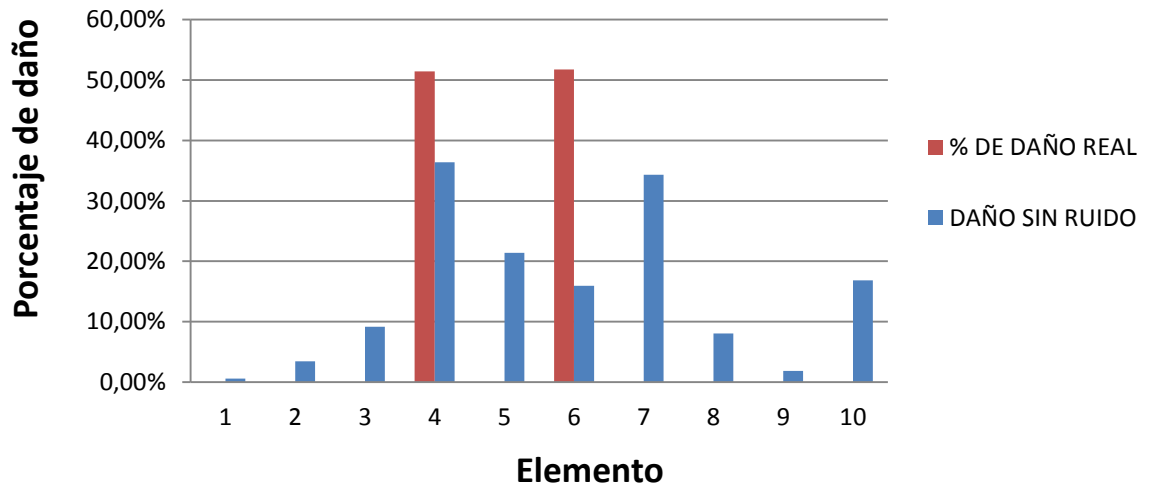


Figura. 5.25.

- **8 Modos de Vibración:(Elementos Dañados: 8 en 51.48% y 10 en 30.87%)**

Comparación de Resultados Daño Real y Daño Cálculo PSO

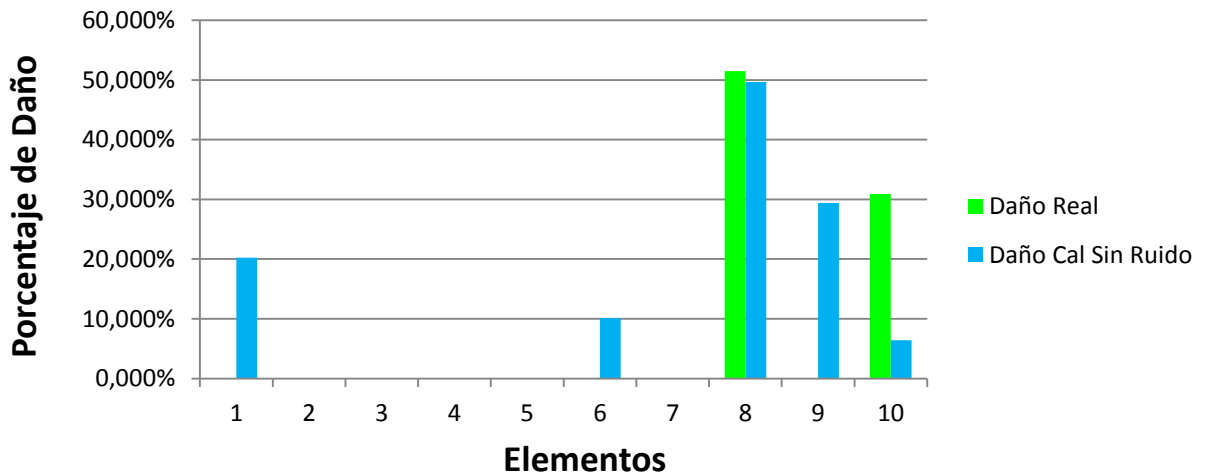


Figura. 5.26.

ARMADURA

a. Daño Múltiple (Dos Elementos))

- 5 Modos de Vibración:(Elementos Dañados: 2 en 10.59% y 4 en 57.23%)

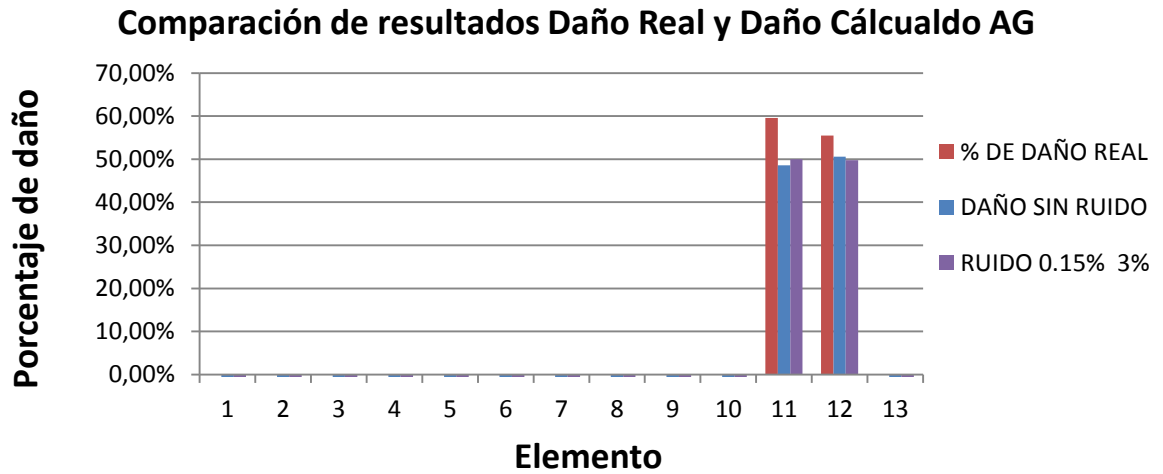


Figura. 5.27.

- 5 Modos de Vibración:(Elementos Dañados: 4 en 27.18% y 9 en 54.14%)

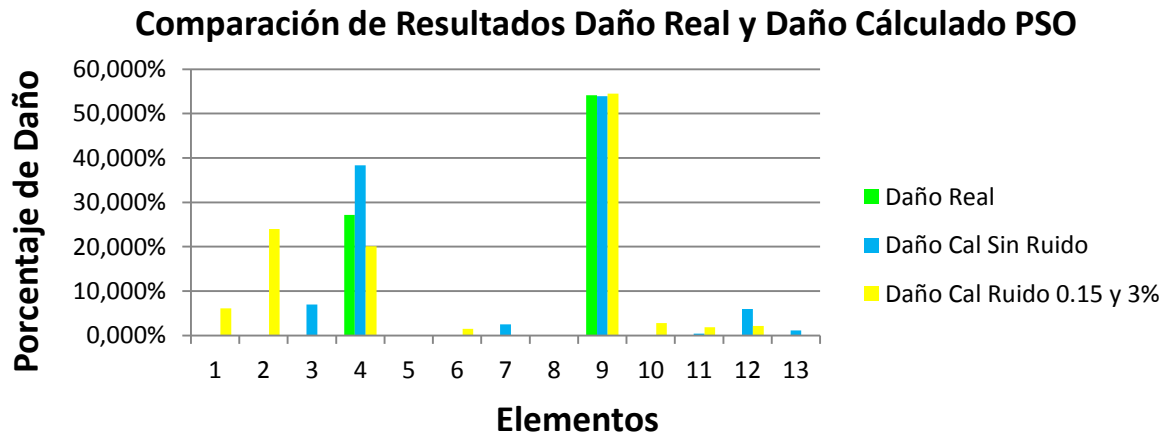


Figura. 5.28.

- **6 Modos de Vibración:(Elementos Dañados: 2 en 10,59% y 4 en 57,23%)**

Comparación de resultados Daño Real y Daño Cálculdo AG

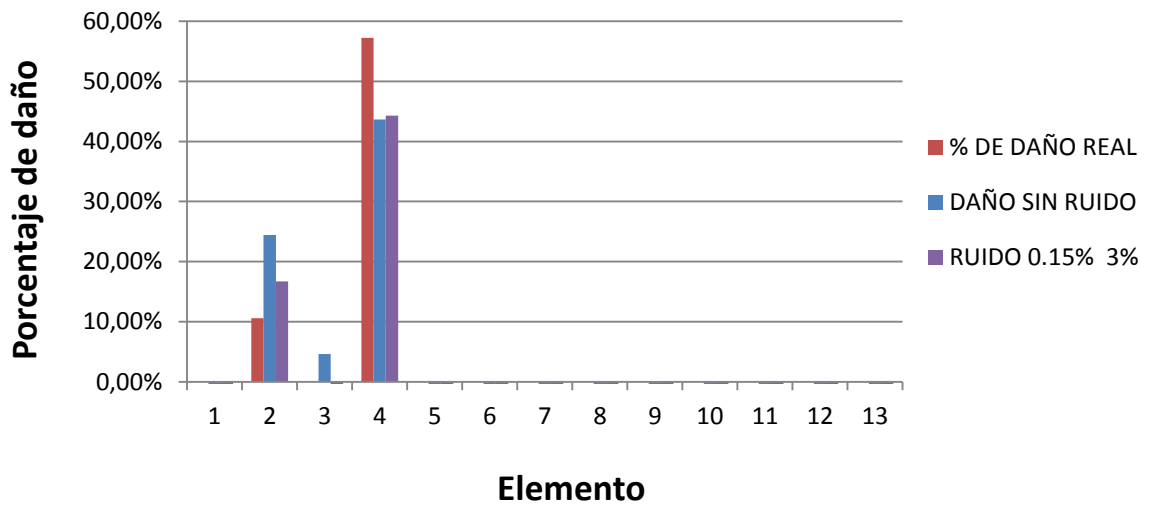


Figura. 5.20.

- **6 Modos de Vibración:(Elementos Dañados: 7 en 30.39% y 10 en 56.72%)**

Comparación de Resultados Daño Real y Daño Cálculdo PSO

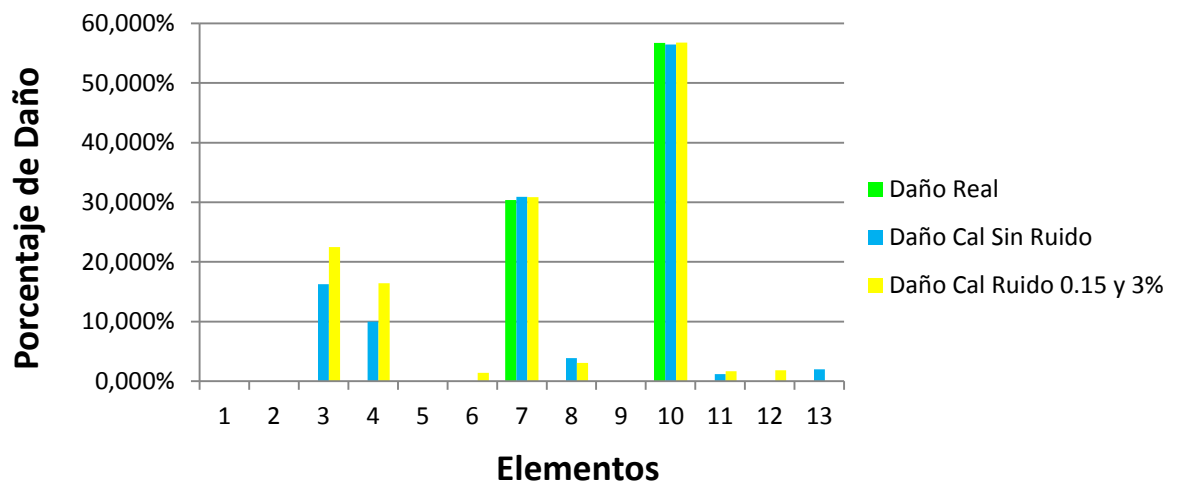


Figura. 5.30.

5.1.3 Comparación de resultados con la función objetivo calibraMAC (ec.2.9) para la ARMADURA con algoritmos genéticos y calibraV1 (ec. 2.10) para PSO.

- **4 Modos de Vibración:(Elemento Dañado: 10 en 12,29%)**

Comparación de resultados Daño Real y Daño Cálculdo AG

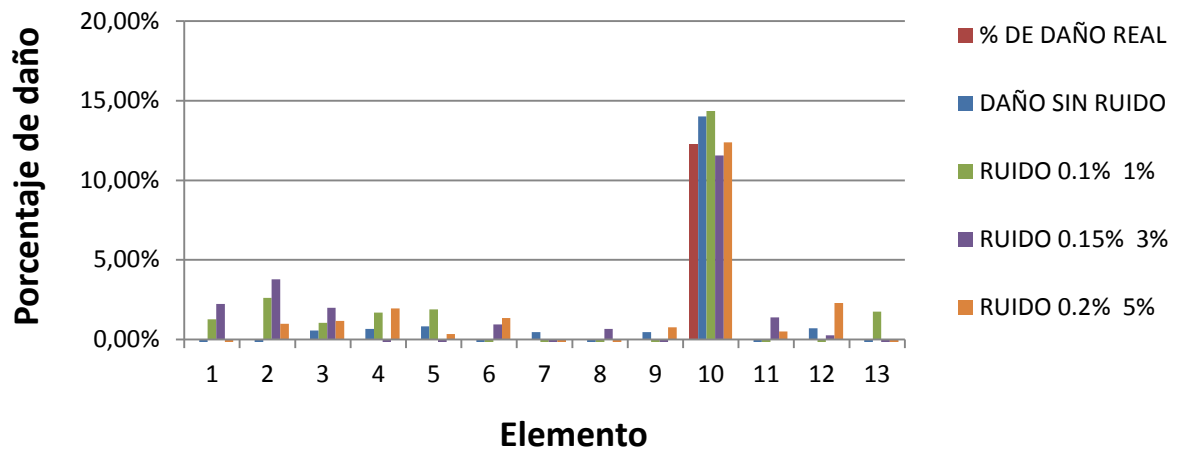


Figura. 5.31.

- **4 Modos de Vibración:(Elemento Dañado: 11 en 5,65%)**

Comparación de Resultados Daño Real y Daño Cálculdo PSO

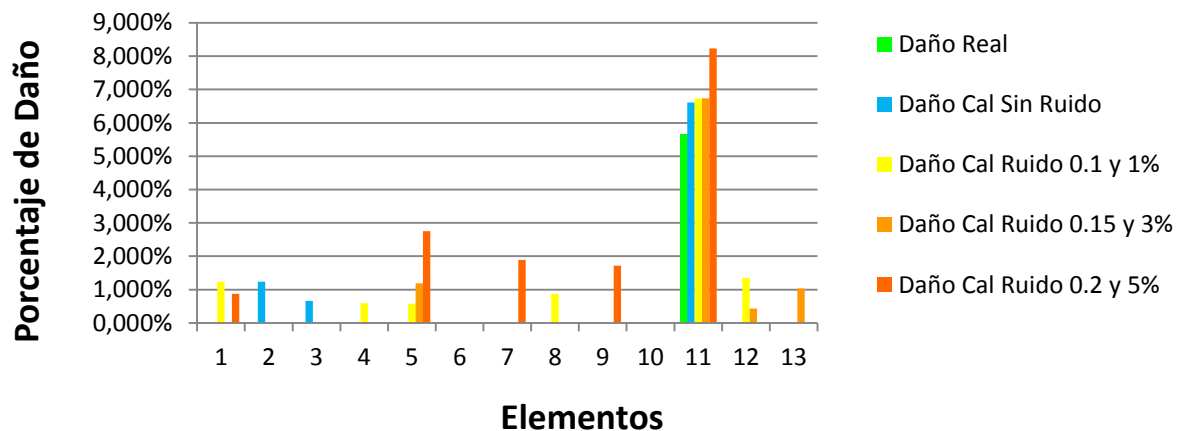


Figura. 5.32.

- 5 Modos de Vibración:(Elemento Dañado: 11 en 51,49%)

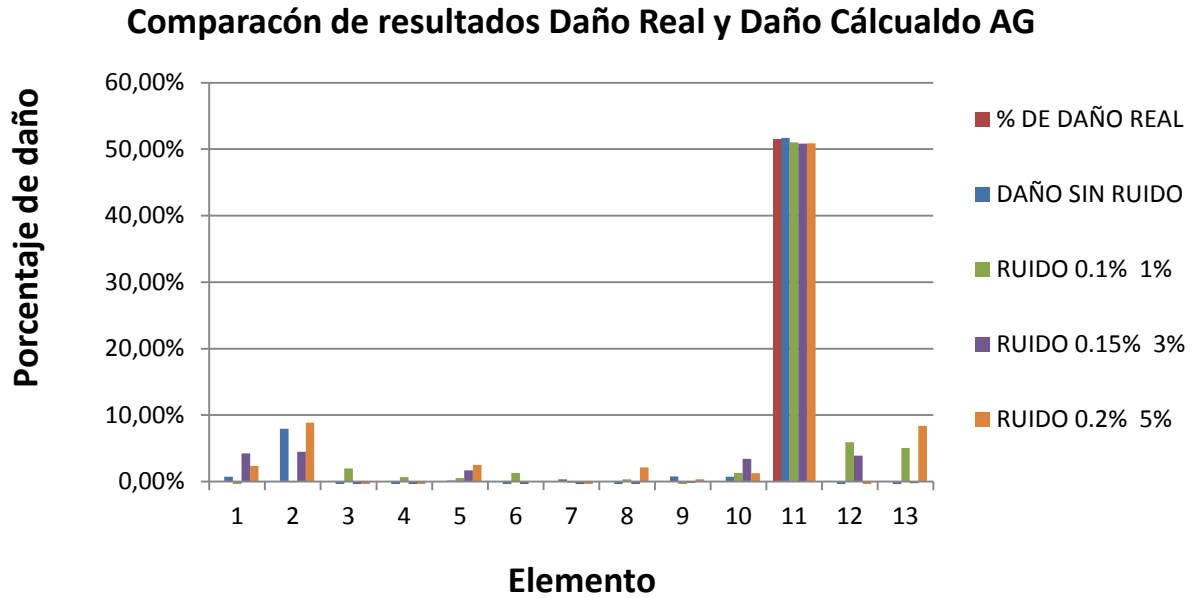


Figura. 5.33.

- 5 Modos de Vibración:(Elemento Dañado: 10 en 34,92%)

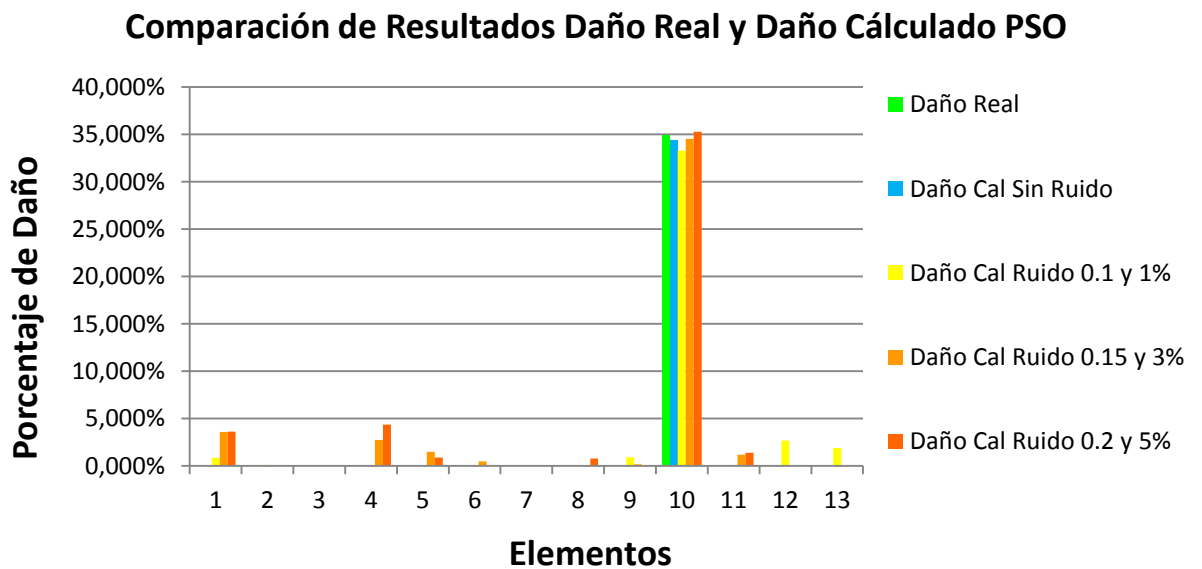


Figura. 5.34.

- 6 Modos de Vibración:(Elemento Dañado: 7 en 19,33%)

Comparación de resultados Daño Real y Daño Cálculdo AG

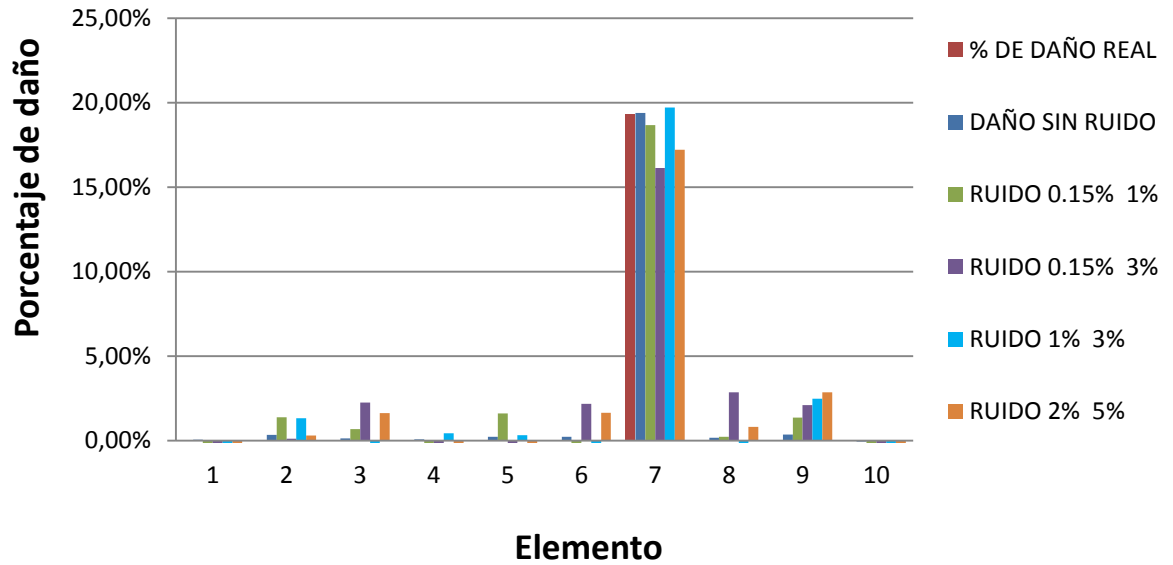


Figura. 5.35.

- 6 Modos de Vibración:(Elemento Dañado: 8 en 33,72%)

Comparación de Resultados Daño Real y Daño Cálculdo PSO

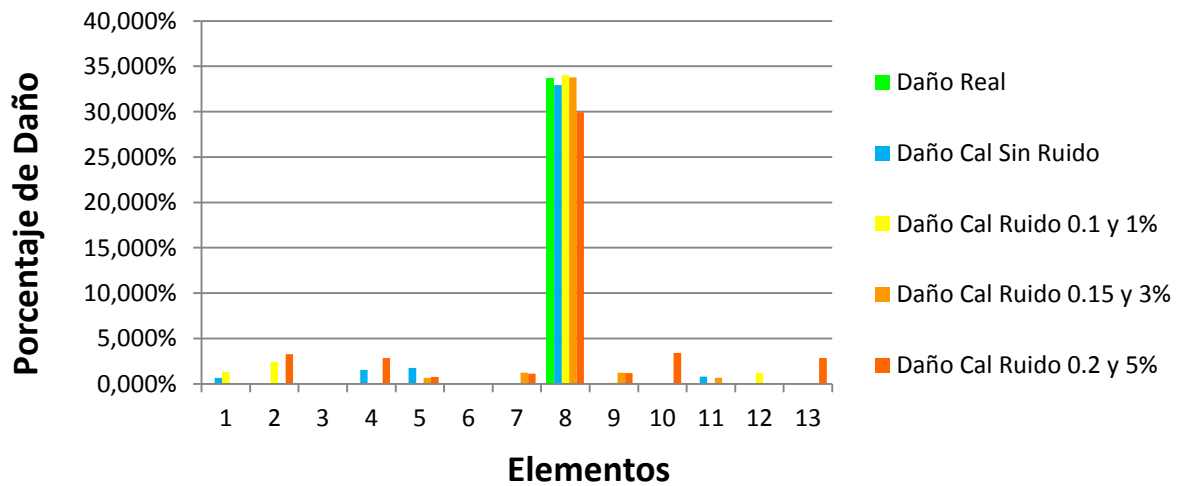


Figura. 5.36.

b. Daño Múltiple (Dos Elementos)

- **4 Modos de Vibración:(Elementos Dañados: 2 en 49,52% y 11 en 53,69%)**

Comparación de resultados Daño Real y Daño Cálculo AG

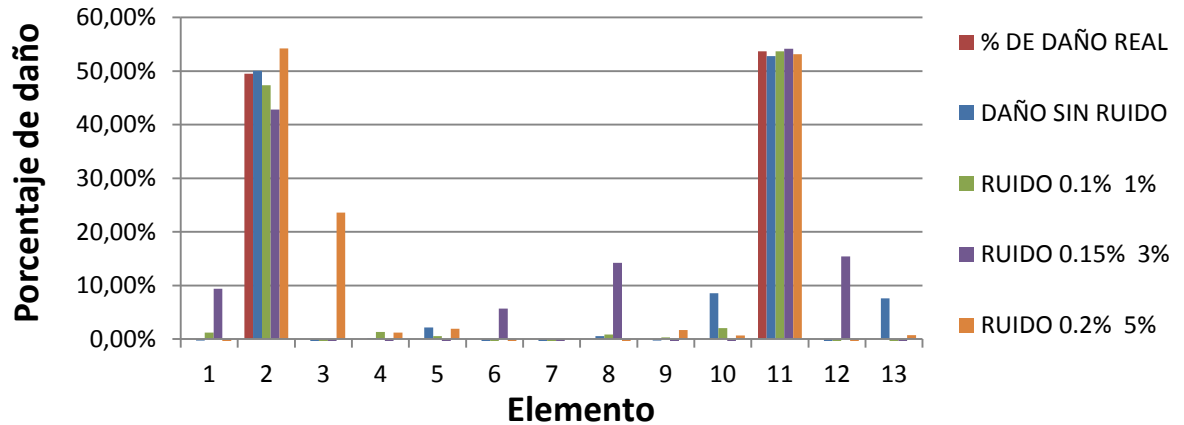


Figura. 5.37.

- **4 Modos de Vibración:(Elementos Dañados: 6 en 5,05% y 11 en 51,48%)**

Comparación de Resultados Daño Real y Daño Cálculo PSO

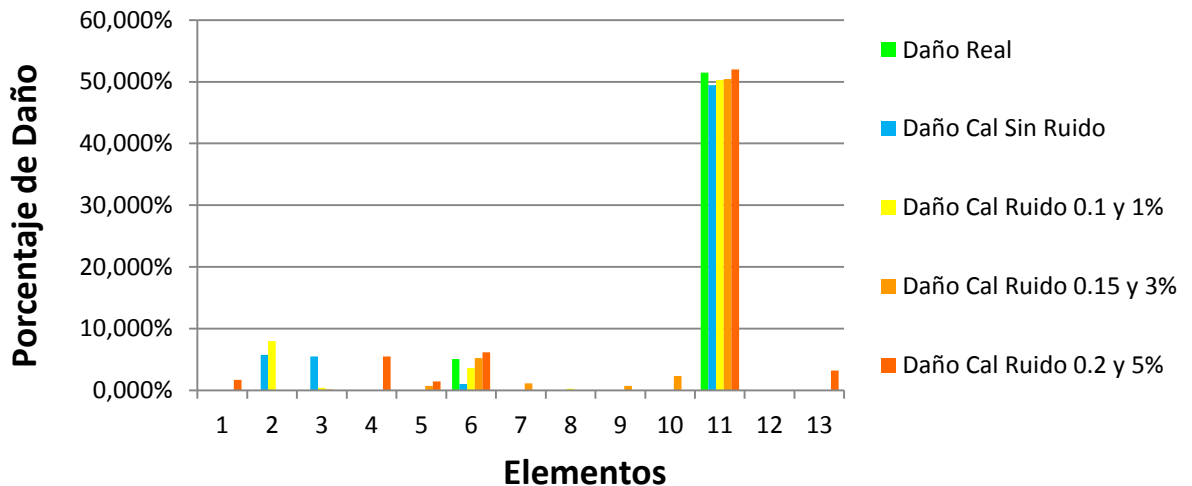


Figura. 5.38.

- 5 Modos de Vibración:(Elementos Dañados: 11 en 59,55% y 12 en 55,48%)

Comparación de resultados Daño Real y Daño Cálculdo AG

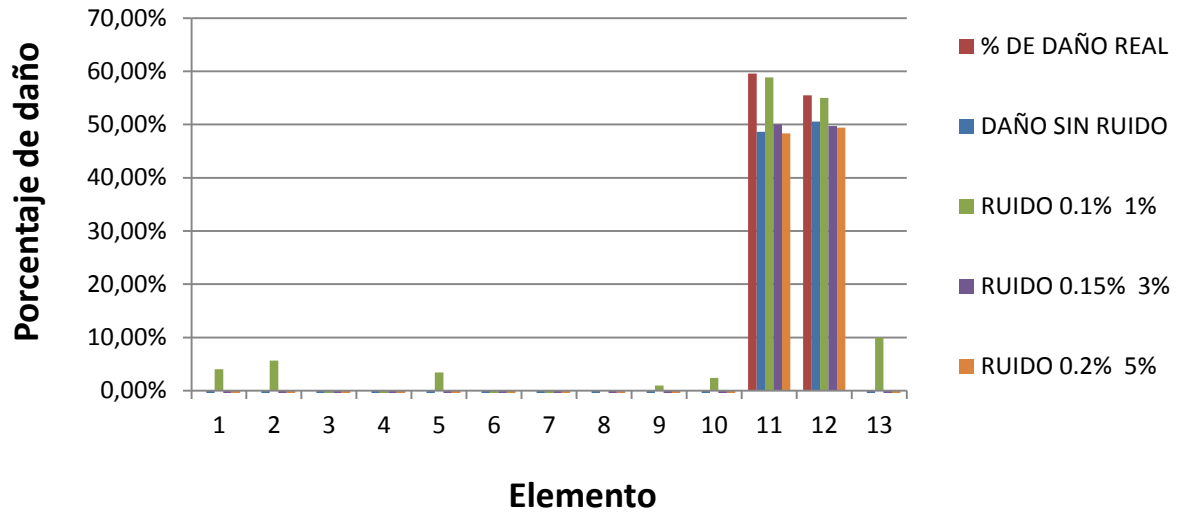


Figura. 5.39.

- 5 Modos de Vibración:(Elementos Dañados: 2 en 10,59% y 4 en 57,23%)

Comparación de Resultados Daño Real y Daño Cálculdo PSO

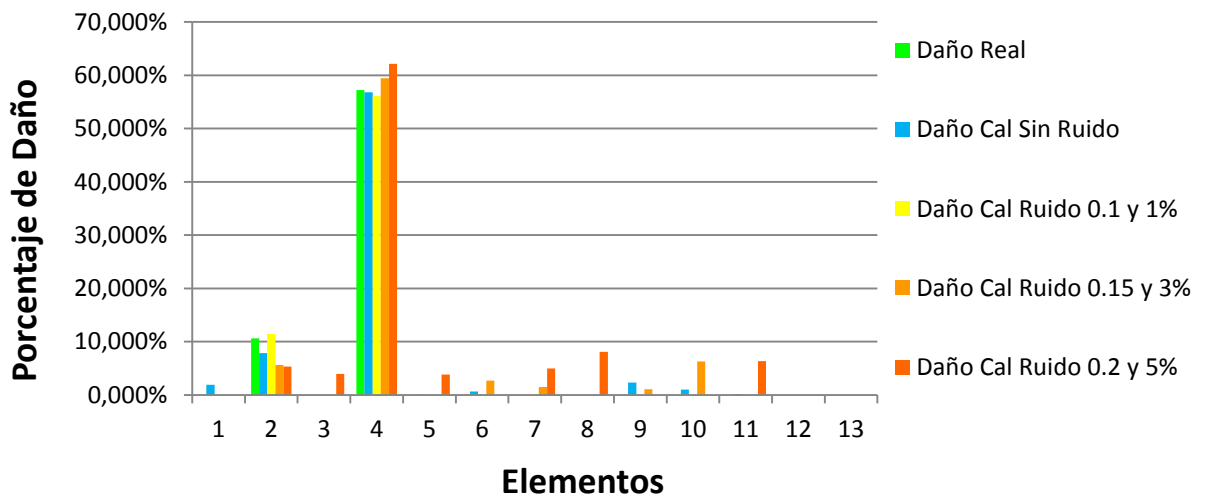


Figura. 5.40.

- **6 Modos de Vibración:(Elementos Dañados: 2 en 10,59% y 4 en 57,23%)**

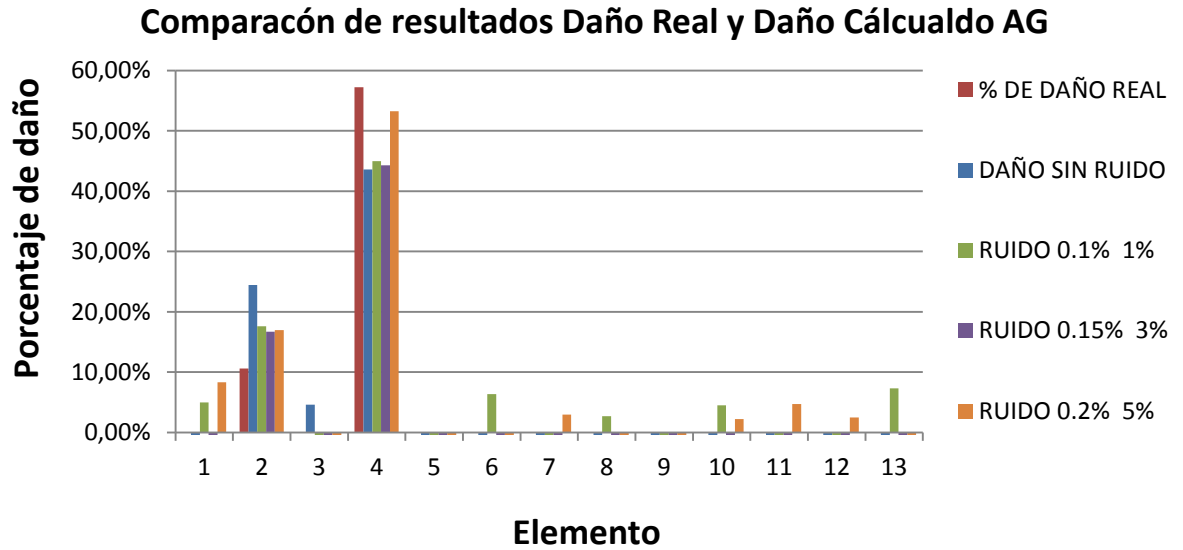


Figura. 5.41.

- **6 Modos de Vibración:(Elementos Dañados: 4 en 39,98% y 10 en 24,10%)**

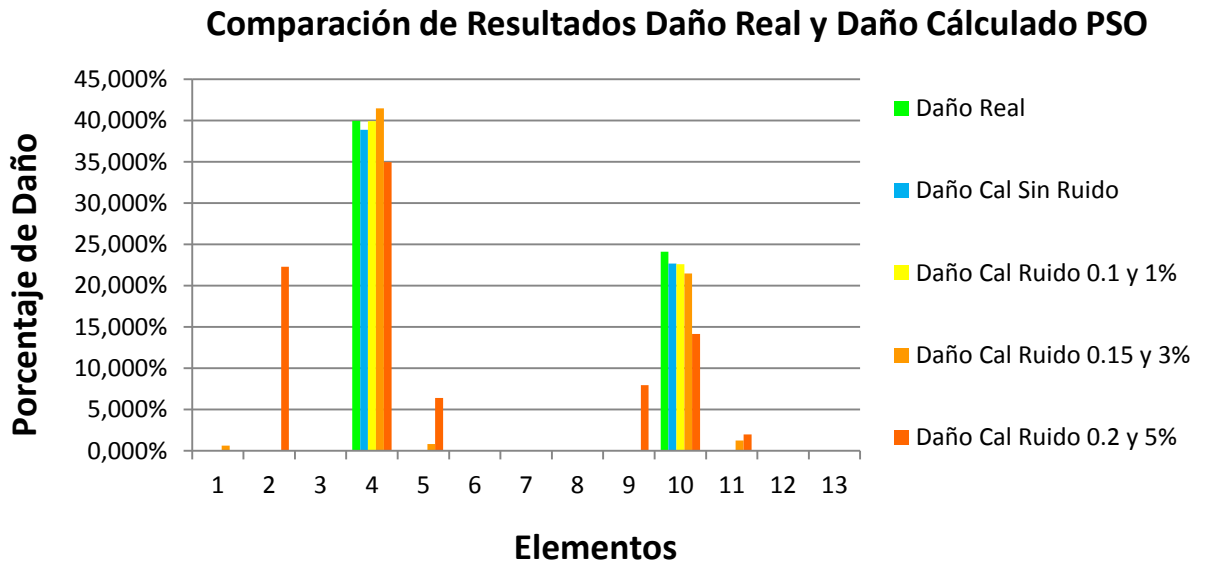


Figura. 5.42.

5.2. ANÁLISIS DE RESULTADOS ESTRUCTURA TIPO VIGA

En el proceso de la detección de daño se dividió en análisis para la viga y análisis para la armadura, siendo el análisis en la viga el procedimiento en el cual se utilizó la función objetivo con el nombre **calibraV1.m** debido a que esta función tuvo mejor comportamiento y mayor grado de exactitud a la hora de detectar el elemento dañado o los elementos dañados, lo cual se comprobó realizando varias pruebas con cada función objetivo y con diferentes estructuras de programación, también se logró identificar el rango en el cual se debe analizar esta estructura, el cual debe ser entre 4 modos de vibración y 8 modos de vibración, ya que si se utilizan más o menos modos de vibración los porcentajes de error con y sin ruido son muy poco precisos y se presentan elementos falsamente dañados.

PARÁMETROS DEL AG	PARÁMETRO SELECCIONADO
Generaciones	500
Población inicial	50
Selección	Ruleta
Cruzamiento	Heurístico
Taza de cruce	0.9
Mutación	Uniforme
Rata de mutación	0.08
Criterio de elitismo	1
Criterio de parada	$1e^{-150}$

Tabla 5.1. Parámetros escogidos para la estructura tipo viga en el algoritmo genético.

5.2.1. MEJOR RESULTADO PARA DAÑO SIMPLE EN LA ESTRUCTURA TIPO VIGA CON LA FUNCIÓN OBJETIVO CALIBRAV1:

MODO DE VIBRACIÓN	% DE DAÑO	ELEMENTO DAÑADO	% DE ERROR
4	41,76%	4	8.623 E -5

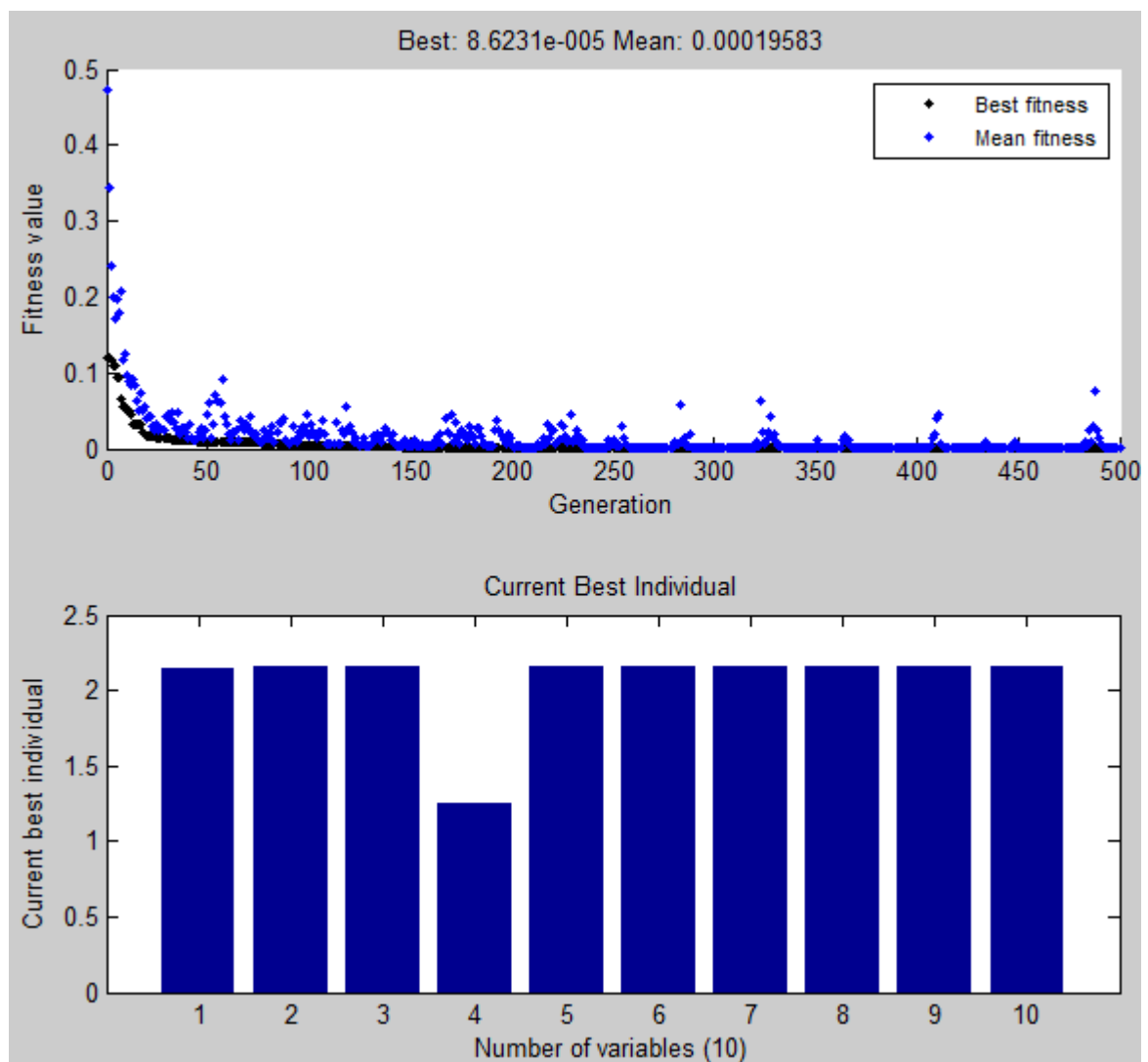


Figura. 5.43. Daño simple en la estructura tipo Viga.

5.2.2. MEJOR RESULTADO PARA DAÑO MÚLTIPLE 2 ELEMENTOS DAÑADOS EN LA ESTRUCTURA TIPO VIGA CON LA FUNCIÓN OBJETIVO CALIBRAV1:

MODO DE VIBRACIÓN	% DE DAÑO	ELEMENTO DAÑADO	% DE ERROR
4	49,01% - 9,57%	3 – 5	4.0576 E - 5

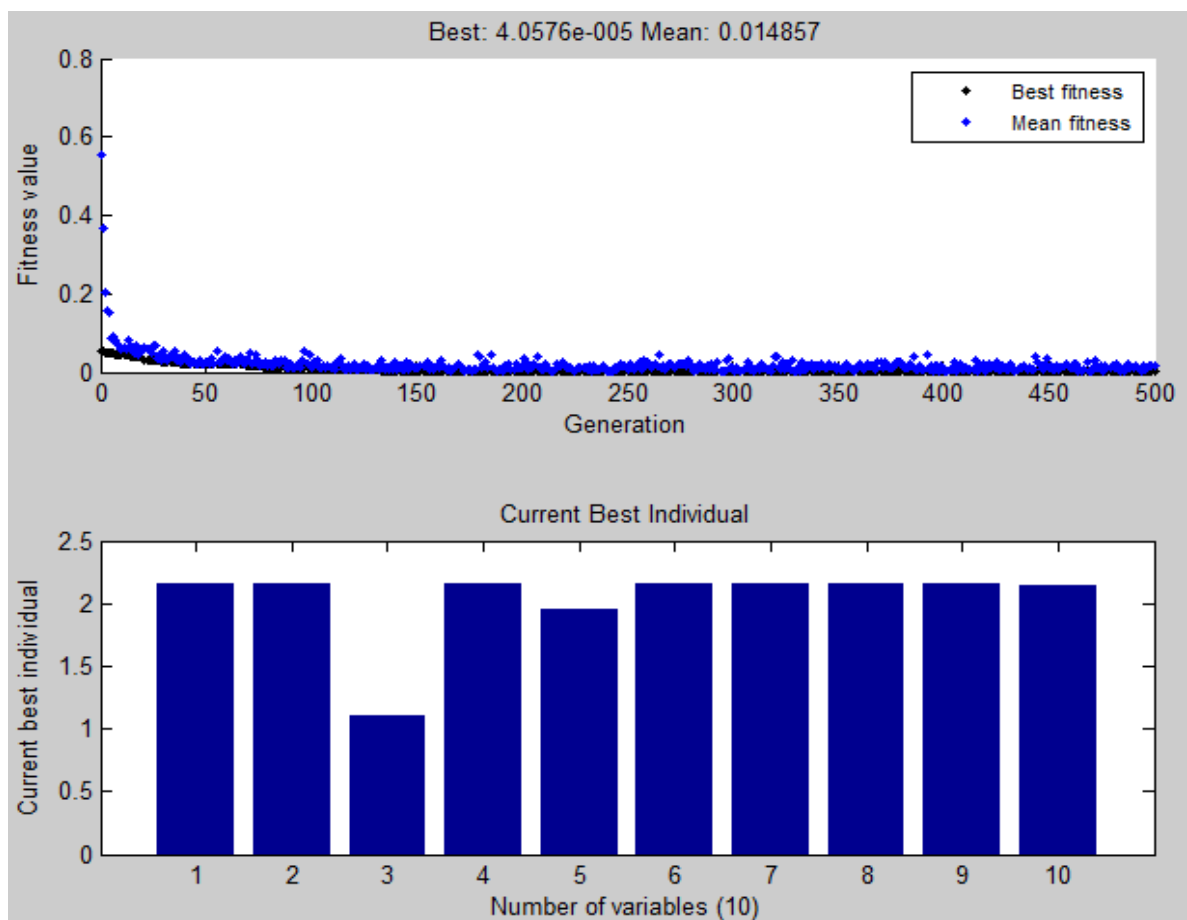


Figura. 5.44. Daño múltiple 2 elementos en la estructura tipo Viga.

5.2.3. MEJOR RESULTADO PARA DAÑO MÚLTIPLE 4 ELEMENTOS DAÑADOS EN LA ESTRUCTURA TIPO VIGA CON LA FUNCIÓN OBJETIVO CALIBRAV1:

MODO DE VIBRACIÓN	% DE DAÑO	ELEMENTO DAÑADO	% DE ERROR
4	24,48% - 20,95% - 31,77% - 4,66%	2 – 6 – 7 – 9	4.3618 E - 5

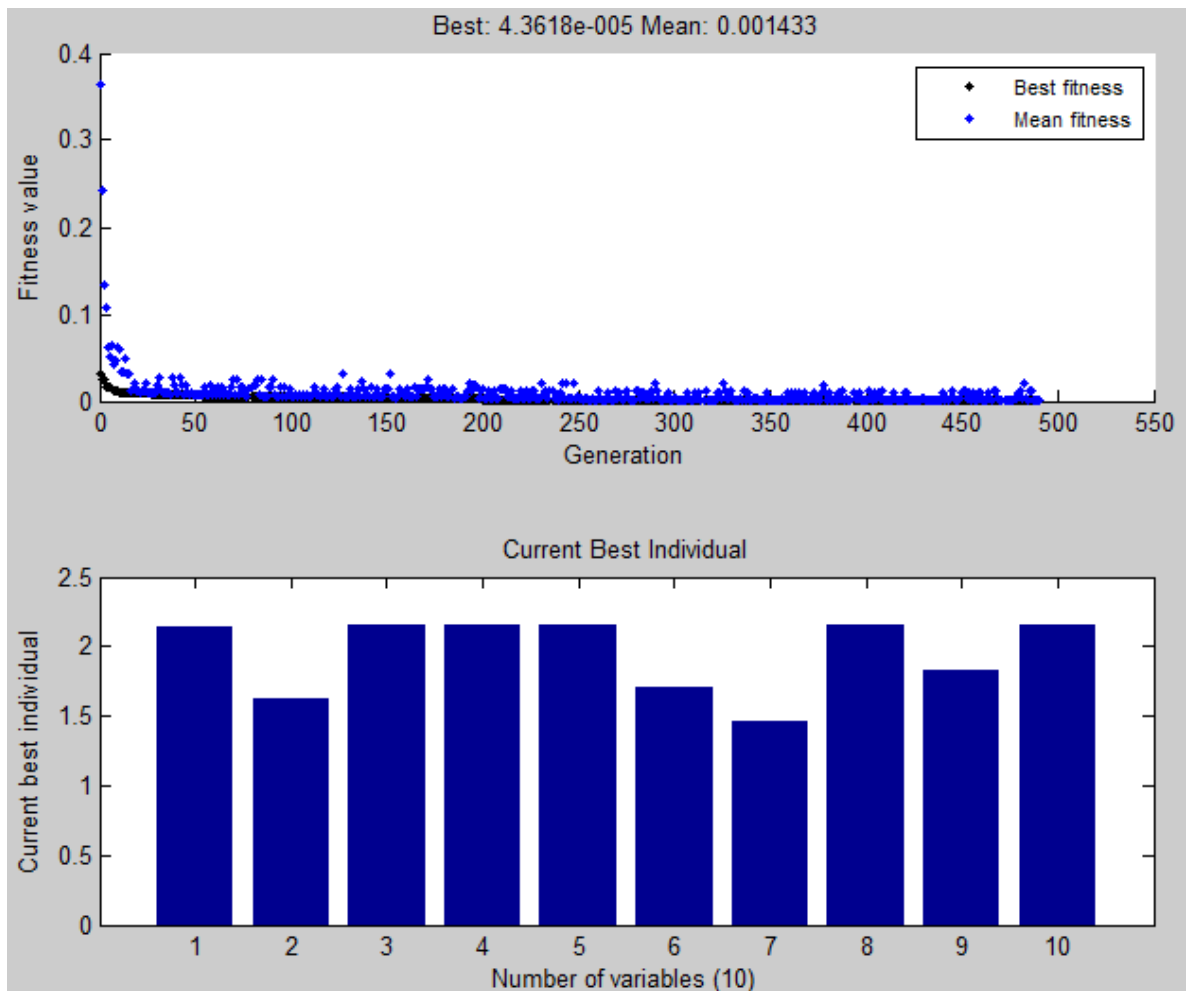


Figura. 5.45. Daño múltiple 4 elementos en la estructura tipo Viga.

En la siguiente tabla se mostrará el escenario de daño en general para la estructura tipo viga con la función objetivo calibraV1. En la tabla* se muestra la detección (Det) como la ubicación del elemento damnificado y diagnóstico (Diag) como la determinación acertada con una desviación hasta del $\pm 5\%$ del valor del cambio del módulo de elasticidad del elemento detectado. Así mismo, se listan los falsos positivos (F.P) como aquellos casos en los que la metodología registro cambio en la rigidez cuando no existía. Adicionalmente, aunque no se muestra, se encontró que la magnitud del daño afecta la detección, en general el desempeño satisfactorio se da para daños superiores al 10%.

Tipo de daño	Modos	Sin Ruido			Ruido <1%			Ruido <3%			Ruido <5%		
		Det	Diag	F.P	Det	Diag	F.P	Det	Diag	F.P	Det	Diag	F.P
Simple	4	1/1	1/1	0	1/1	1/1	0	1/1	1/1	0	1/1	0/1	0
	6	1/1	1/1	0	1/1	1/1	0	1/1	1/1	0	1/1	1/1	0
	8	1/1	1/1	0	1/1	1/1	0	1/1	1/1	0	1/1	1/1	0
2 Elementos	4	2/2	2/2	0	2/2	2/2	0	2/2	2/2	3	2/2	1/2	1
	6	2/2	2/2	0	2/2	2/2	0	2/2	2/2	0	2/2	1/2	1
	8	2/2	2/2	0	2/2	2/2	0	2/2	2/2	0	2/2	2/2	2
4 Elementos	4	4/4	4/4	0	4/4	4/4	0	4/4	1/4	2	4/4	3/4	2
	6	4/4	4/4	0	4/4	4/4	0	4/4	3/4	1	4/4	1/4	1
	8	4/4	4/4	0	4/4	4/4	0	4/4	1/4	2	4/4	3/4	4

Tabla 5.2. Comparación de los diferentes episodios de daño y ruido de los resultados obtenidos con el algoritmo genético para la estructura tipo viga.

*DETECCIÓN DE DAÑOS EN ESTRUCTURAS UTILIZANDO LA TÉCNICA DE OPTIMIZACIÓN PSO, AUTOR, Diego Ramírez; DIRECTORES, MSc Jaid Quiroga; MSc Iván Darío Gómez.

Tipo de daño	Modos	Sin Ruido			Ruido <1%			Ruido<3%			Ruido<5%		
		Det	Diag	F.P	Det	Diag	F.P	Det	Diag	F.P	Det	Diag	F.P
Simple	4	1/1	1/1	0	1/1	1/1	0	1/1	1/1	0	1/1	0/1	1
	6	1/1	1/1	0	1/1	1/1	0	1/1	1/1	0	1/1	0/1	3
	8	1/1	1/1	0	1/1	1/1	0	1/1	0/1	0	1/1	1/1	4
2 Elementos	4	2/2	2/2	0	2/2	2/2	0	2/2	2/2	3	2/2	2/2	3
	6	2/2	2/2	0	2/2	2/2	0	2/2	2/2	1	2/2	1/2	2
	8	2/2	2/2	0	2/2	2/2	0	2/2	1/2	2	2/2	2/2	4
4 Elementos	4	4/4	4/4	0	4/4	4/4	0	3/4	2/4	1	3/4	2/4	2
	6	4/4	4/4	0	4/4	4/4	0	4/4	3/4	1	4/4	2/4	2
	8	4/4	4/4	0	4/4	4/4	0	4/4	4/4	1	4/4	3/4	2

Tabla 5.3. Comparación de los diferentes episodios de daño y ruido de los resultados obtenidos con PSO para la estructura tipo viga.

De la tabla y teniendo en cuenta los resultados de PSO y AG se puede decir que algoritmos genéticos identifica y detecta mejor que PSO en un enfoque general y presenta menores episodios de elementos falsamente dañados; se identificaron las casillas en verde como mejor comportamiento para AG y en amarillo para mejor comportamiento con PSO, logrando así concluir cual técnica tiene mejor comportamiento global, lo que muestra un mejor comportamiento por parte de algoritmos genéticos con un total de 11 episodios con mejor comportamiento que PSO que mostro 6 episodios con mejor comportamiento.

5.3. ANÁLISIS DE RESULTADOS ESTRUCTURA TIPO ARMADURA

En el proceso de la detección de daño para la armadura, se realizaron pruebas con ambas funciones objetivo pero a diferencia que en la tesis de PSO la optimización con algoritmos genéticos tuvo un mejor comportamiento con la función objetivo calibraMAC tanto en daño simple como en daño múltiple con o sin ruido, lo cual nos muestra la diferencia entre las dos técnicas y sus resultados; A continuación se mostraran los mejores resultados con la técnica de algoritmos genéticos y la función objetivo calibraMAC. Se observó que aunque no siempre se logrará llegar al final de la iteración completa del algoritmo genético puede ocurrir que detecte correctamente y con un mínimo error el elemento o los elementos damnificados como se muestra en la última figura del análisis para la armadura con daño múltiple 2 elementos damnificados.

PARÁMETROS DEL AG	PARÁMETRO SELECCIONADO
Generaciones	300
Población inicial	150
Selección	Ruleta
Cruzamiento	Heurístico
Taza de cruce	0.9
Mutación	Uniforme
Rata de mutación	0.15
Criterio de elitismo	1
Criterio de parada	$1e^{-150}$

Tabla 5.4. Parámetros escogidos para la estructura tipo armadura en el algoritmo genético.

5.3.1. MEJOR RESULTADO PARA DAÑO SIMPLE EN LA ESTRUCTURA TIPO ARMADURA CON LA FUNCIÓN OBJETIVO CALIBRAMAC:

MODO DE VIBRACIÓN	% DE DAÑO	ELEMENTO DAÑADO	% DE ERROR
5	51,49%	11	4.5294 E - 4

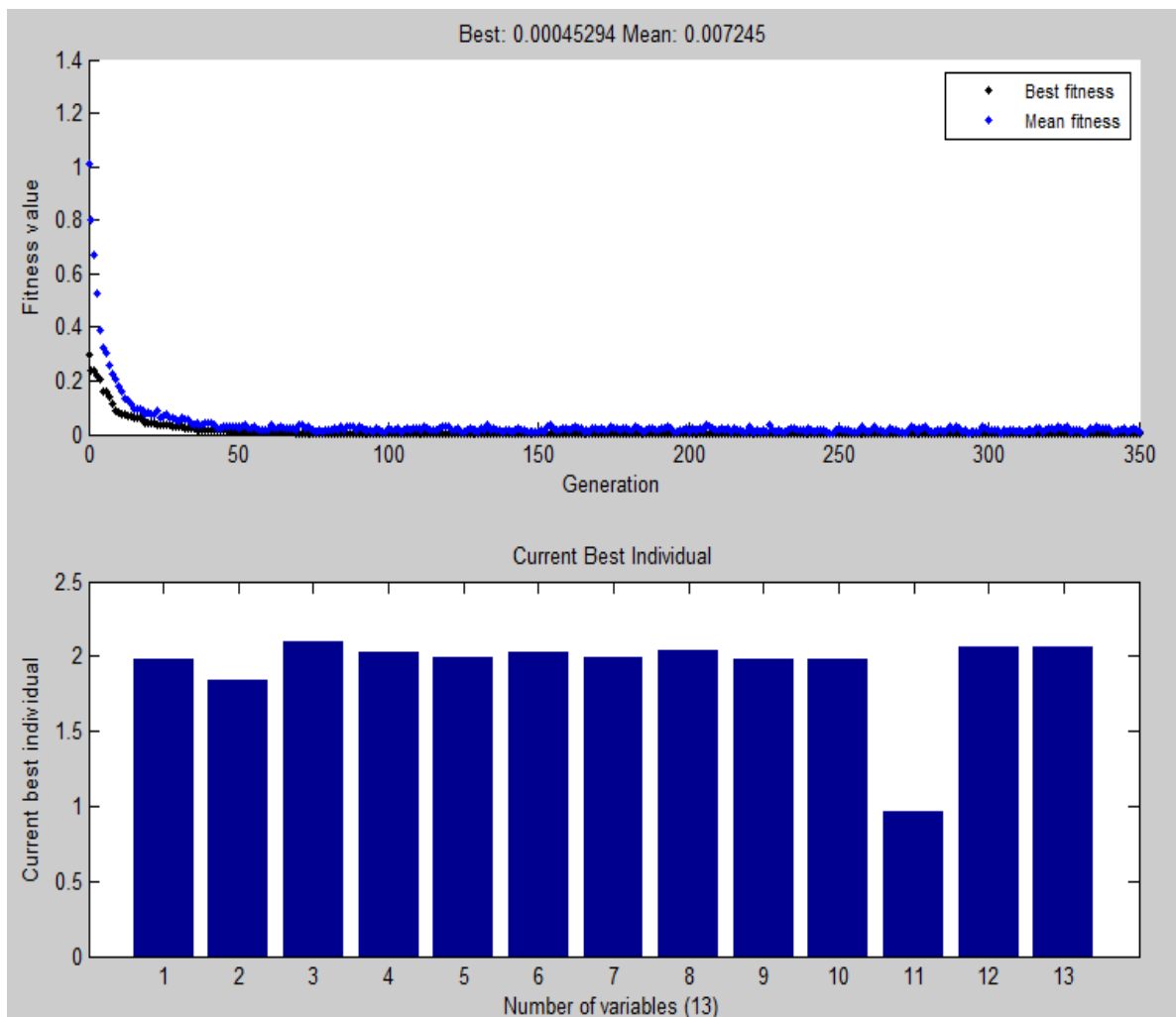


Figura. 5.46. Daño simple en la estructura tipo Armadura.

5.3.1. MEJOR RESULTADO PARA DAÑO MÚLTIPLE 2 ELEMENTOS DAÑADOS EN LA ESTRUCTURA TIPO ARMADURA CON LA FUNCIÓN OBJETIVO CALIBRAMAC:

MODO DE VIBRACIÓN	% DE DAÑO	ELEMENTO DAÑADO	% DE ERROR
5	59,56% - 55,48%	11 – 12	3.3332

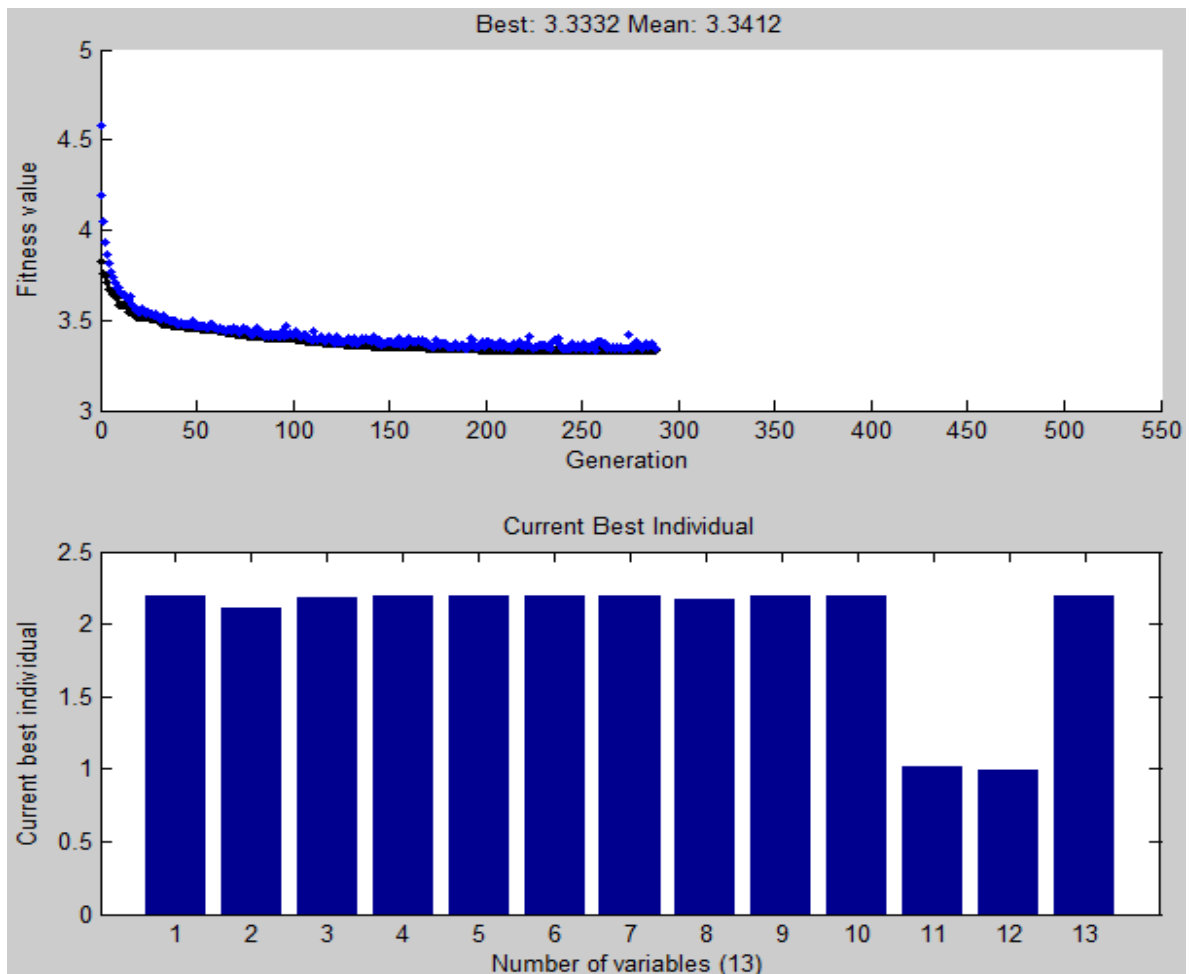


Figura. 5.47. Daño múltiple 2 elementos en la estructura tipo Armadura.

En la siguiente tabla se mostrará el escenario de daño en general para la estructura tipo armadura con la función objetivo calibraMAC. En la tabla se muestra la detección (Det) como la ubicación del elemento damnificado y diagnóstico (Diag) como la determinación acertada con una desviación hasta del $\pm 5\%$ del valor del cambio del módulo de elasticidad del elemento detectado. Así mismo, se listan los falsos positivos (F.P) como aquellos casos en los que la metodología registro cambio en la rigidez cuando no existía. Adicionalmente, aunque no se muestra, se encontró que la magnitud del daño afecta la detección, en general el desempeño satisfactorio se da para daños superiores al 10%.

Tipo de daño	Modos	Sin Ruido			Ruido <1%			Ruido<3%			Ruido<5%		
		Det	Diag	F.P	Det	Diag	F.P	Det	Diag	F.P	Det	Diag	F.P
Simple	4	1/1	1/1	0	1/1	1/1	0	1/1	1/1	0	1/1	1/1	0
	5	1/1	1/1	0	1/1	1/1	0	1/1	1/1	0	1/1	1/1	0
	6	1/1	1/1	0	1/1	1/1	0	1/1	1/1	0	1/1	1/1	0
2 Elementos	4	2/2	2/2	0	2/2	2/2	0	2/2	1/2	2	2/2	2/2	1
	5	2/2	1/2	0	2/2	2/2	0	2/2	1/2	0	2/2	0/2	0
	6	2/2	0/2	0	2/2	0/2	0	2/2	0/2	0	2/2	1/2	0

Tabla 5.5. Comparación de los diferentes episodios de daño y ruido de los resultados obtenidos con el algoritmo genético para la estructura tipo armadura.

Tipo de daño	Modos	Sin Ruido			Ruido <1%			Ruido<3%			Ruido<5%		
		Det	Diag	F.P	Det	Diag	F.P	Det	Diag	F.P	Det	Diag	F.P
Simple	4	1/1	1/1	0	1/1	1/1	0	1/1	1/1	0	1/1	0/1	0
	5	1/1	1/1	0	1/1	1/1	0	1/1	1/1	0	1/1	0/1	0
	6	1/1	1/1	0	1/1	1/1	0	1/1	1/1	0	1/1	0/1	0
2 Elementos	4	2/2	2/2	2	2/2	2/2	1	2/2	2/2	0	2/2	2/2	0
	5	2/2	2/2	0	2/2	2/2	0	2/2	2/2	1	2/2	2/2	2
	6	2/2	2/2	0	2/2	2/2	0	2/2	2/2	0	2/2	1/2	3

Tabla 5.6. Comparación de los diferentes episodios de daño y ruido de los resultados obtenidos con PSO para la estructura tipo armadura.

De la tabla de resultados cabe resaltar que los datos de algoritmos genéticos se obtuvieron de la función objetivo que con PSO no dio resultados (ec. 2.9) y se muestra que algoritmos genéticos detecto y diagnostico sin falsos positivos en daño simple al igual que PSO y para daño múltiple detecto pero no logro diagnosticar por lo cual diagnostico mejor la técnica PSO para daño múltiple. Se identificaron las casillas en verde como mejor comportamiento para AG y en amarillo para mejor comportamiento con PSO, logrando así concluir cual técnica tiene mejor comportamiento global en la estructura tipo armadura, lo cual muestra un mejor desempeño por parte de PSO con un total de 8 episodios con mejor comportamiento que algoritmos genéticos que mostro 4 episodios con mejores resultados, esto para daño múltiple.

5.4. ANÁLISIS DE TIEMPO DE RESPUESTA PARA LAS TÉCNICAS ALGORITMOS GENÉTICOS VS PSO.

En cuanto al tiempo de respuesta de la técnica de optimización Algoritmos Genéticos, se debe diferenciar con claridad que para cada estructura tuvo un tiempo promedio de respuesta y en cada caso particular (daño simple o daño múltiple) también tuvo un tiempo promedio; esto es a causa del grado de complejidad del problema, que en cuanto más complejo sea, mayor será el tiempo que tardara el algoritmo en procesar la información.

En las tablas 5.7 y 5.8 se muestra un paralelo entre los tiempos promedio de ejecución, para las dos técnicas de optimización utilizadas en las tesis AG y PSO* en cada caso en particular; se aclara que estos tiempos están ligados al desempeño de la maquina con la que se procesaron; en el caso de algoritmos genéticos se utilizaron dos maquinas diferentes, una para la viga tipo Portátil ACER Intel (R) CORE (TM)2 Duo CPU T6400 @ 2.00 GHz, 2.00 GHz, Memoria RAM 3 Gb, Sistema operativo de 32 bits, y para la armadura se dispuso de un computador tipo escritorio DELL Intel (R) CORE Cuad CPU @ 2.60 GHz, 2.60 GHz, Memoria RAM 4 Gb, Sistema operativo de 64 bits:

ESTRUCTURA TIPO VIGA		DAÑO SIMPLE			DAÑO MÚLTIPLE 2 ELEMENTOS DAMNIFICADOS			DAÑO MÚLTIPLE 4 ELEMENTOS DAMNIFICADOS		
		4 M	6M	8M	4M	6M	8M	4M	6M	8M
AG	Tiempo (min)	30	40	50	40	50	60	45	55	70
PSO	Tiempo (min)	70	80	90	60	70	80	70	90	85

Tabla 5.7. Comparación de los diferentes tiempos de ejecución entre AG y PSO para la estructura tipo viga.

ESTRUCTURA TIPO ARMADURA		DAÑO SIMPLE			DAÑO MÚLTIPLE 2 ELEMENTOS DAMNIFICADOS		
		4 M	5M	6M	4M	5M	6M
AG	Tiempo (min)	80	90	100	140	100	110
PSO	Tiempo (min)	175	185	190	220	350	140

Tabla 5.8. Comparación de los diferentes tiempos de ejecución entre AG y PSO para la estructura tipo armadura.

De los resultados en el paralelo de tiempos de ejecución de las técnicas AG y PSO se puede observar que algoritmos genéticos fue más compacto en cuanto a encontrar la solución en un menor tiempo de trabajo a cada problema descrito, pero se debe tener en cuenta que en la tesis de PSO no se especifica el equipo de computo con el que se realizaron los análisis de cada problema, por lo que no se puede asegurar cual de las técnicas requiere menor gasto informático.

6. CONCLUSIONES

En el desarrollo del proyecto se observó que hay que tener una clara pero cambiante visión del objetivo final, ya que en el presente trabajo se pretendía comparar el comportamiento de PSO con AG lo cual hubiera sido ideal si algoritmos genéticos hubieran tenido el mismo comportamiento con la función objetivo *calibraV1*.

Para la viga se pudo comparar a plenitud el desempeño de las dos técnicas siendo más acertados los resultados obtenidos por algoritmos genéticos; pero para la armadura la función objetivo *calibraV1* no dio resultados confiables así que se procedió a evaluar el desempeño de la función objetivo *calibraMAC* que para PSO no tuvo buen comportamiento, pero por el contrario con algoritmos genéticos resultó muy precisa en cuanto a detectar los elementos damnificados, lo cual es muy complicado en la armadura con daño múltiple debido a las complejas formas modales que presenta esta estructura.

Para poder realizar los cálculos con los mejores resultados, se tuvo que en forma experimental y basándose de los textos* existentes en el tema de la salud estructural y más precisamente en el campo de los algoritmos genéticos que describen las posibles estructuras para el análisis como el que se desarrolló en el presente proyecto, dando así una idea de dónde empezar a buscar la convergencia de la mejor estructura de análisis, para atacar cada daño con su respectiva dificultad (ruido).

*DETECCAO DE DANO EM ESTRUTURAS UTILIZANDO ALGORITMOS GENETICOS E PARAMETROS DINAMICOS,
Ing. Civil Jesús Daniel Villalba Morales.

6.1. SUGERENCIAS PARA TRABAJOS FUTUROS

Basándose en las dificultades que se presentaron en la realización del proyecto y esperando que las personas interesadas en el tema de la inteligencia artificial y más precisamente en la detección de daños para la salud estructural, se describan las posibles temáticas que al investigarse más a profundidad logran dar mejores y más precisos resultados en los análisis pertinentes.

- Se recomienda que en cuanto al tema que tiene que ver con las funciones objetivo, se debe buscar más opciones que las que se presentaron en este trabajo, para tener más seguridad que la función a trabajar es la indicada para el objetivo alcanzar.
- En este trabajo se utilizó una de las técnicas de optimización existentes, así que sería recomendable utilizar otra técnica de detección como las redes neuronales, para así poder saber cuál de las técnicas de optimización que han analizado este problema tiene mejor comportamiento y sería la ideal para realizar un proyecto que incluya medición de frecuencias naturales y formas modales de una estructura real.
- Implementar un modelo para lograr que el rango de daño en los parámetros iniciales sea más cercano al daño real, para que el algoritmo sea más preciso y rápido en la detección del caso particular de daño.
- Plantear una estrategia o procedimiento que logre detectar el caso de daño confiablemente, para poder ser aplicado el método de optimización a un caso real.

- Analizar y plantear una posible forma de obtener físicamente (sensores) las variables de entrada en un caso real.
- Determinar cómo se plantearía el problema en una estructura más compleja o en el caso cuando el daño no está en un punto determinado sino que se presenta en forma distribuida.

REFERENCIAS

- COLLINS J.J.** A Global Representation Scheme for Genetic Algorithms. *Proceedings of the International Conference on Computational Intelligence*. Vol 1226, 1997, pp. 1-15
- 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, pp.509-520*
- 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, pp. 403–416.
- DIGALAKIS J. AND MARGARITIS K.** 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*
- DORIGO M. AND L.M. GAMBARDELLA.** Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. *IEEE Transactions on Evolutionary Computation*. Vol.1, No.1, 1997. In press.
- DORIGO M., 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, pp.29-41.
- OPENSEES** <http://opensees.berkeley.edu/index.php>[Glover F. and Kochenberger G.G. *Handbook of Metaheuristics*. Kluwer Academic Publishers. 2003.
- GOLDBERG DAVID E.** *Genetic Algorithms in search, optimization and machine learning*. Adison Wesley, 1989 [11] Kalyanmoy Deb. Genetic algorithms in multimodal function optimization. *TCGA report no. 89002, The University of Alabama, Dept. of Engineering mechanics*. 1989
- CHAMBERS L.** *Practical Handbook of Genetic Algorithms: Applications*, volume 1. CRC Press, 1995.
- HOLLAND, JOHN.** Genetic algorithms. *Scientific American*. Vol 267, No 1, 1992, pp. 66-72. TCL <http://en.wikipedia.org/wiki/Tcl>
- KENNEDY J AND EBERHART R.** Particle Swarm Optimization. *Proceedings of IEEE International Conference on Neural Networks*. 1995, pp. 1942-1948.

Kennedy J. and Eberhart, R. *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, pp. 56-76..

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, pp. 151-152.

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, pp. 1980-1987.* [19] Melián B, Moreno Perez J and Moreno Vega J. Metaheuristics: A global view. *Revista Iberoamericana de Inteligencia Artificial.* Vol 2, N° 19, 2003, pp. 7-28.

MELIÁN B, MORENO PEREZ J AND MORENO VEGA J. Metaheuristics: A global view. *Revista Iberoamericana de Inteligencia Artificial.* Vol 2, N° 19, 2003, pp. 7-28.

BIBLIOGRAFÍA

ANTONA SORIA Antonio, ESCUDERO HERNANZ María Soledad. Algoritmos genéticos aplicados en visión artificial. Universidad de Alcalá. Departamento de Electrónica, S.N., 2002.

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

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.

CERROLAZA Miguel, ANNICCHIARICO William. Algoritmos de Optimización Estructural Basados en Simulación genética. Consejo de Desarrollo Científico y Humanístico, Universidad Central de Venezuela. Venezuela 1996.

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

Genetic Algorithm and Direct Search Toolbox User's Guide.

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

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.

PRENDES G. Maria B. Optimización del diseño y construcción de edificios metálicos en base a Algoritmos Genéticos. Universidad de Oviedo. Departamento de Construcción e Ingeniería de Fabricación. 2002.

RAMÍREZ R. Diego A. Detección de daños en estructuras utilizando la técnica de optimización PSO. Universidad Industrial de Santander. Escuela de Ingeniería

Mecánica, 2010.

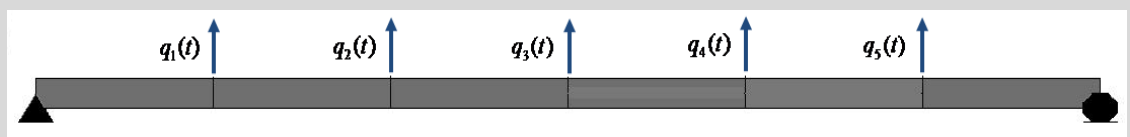
SALAZAR P. Pedro Y. Algoritmo híbrido auto configurado para optimización estructural. Universidad Industrial de Santander. Escuela de Ingeniería Civil, 2009.

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.

APÉNDICES

A.1. EJEMPLO DE ANÁLISIS 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.



$$K = \begin{pmatrix} 7445823.396 & -7166574.835 & 3127100.944 & -818485.193 & 185360.808 \\ -7166574.835 & 10572924.34 & -7985060.028 & 3273940.771 & -741443.23 \\ 3127100.944 & -7985060.028 & 10719764.17 & -7753934.141 & 2780412.114 \\ -818485.193 & 3273940.771 & -7753934.141 & 9648420.793 & -5856861.476 \\ 185360.808 & -741443.23 & 2780412.114 & -5856861.476 & 5596816.301 \end{pmatrix}$$

$$M = \begin{pmatrix} 288.36 & 0 & 0 & 0 & 0 \\ 0 & 288.36 & 0 & 0 & 0 \\ 0 & 0 & 288.36 & 0 & 0 \\ 0 & 0 & 0 & 288.36 & 0 \\ 0 & 0 & 0 & 0 & 288.36 \end{pmatrix}$$

Figura A. 4. Matrices de rigidez y masa de la viga considerada como ejemplo.

En la Figura A. 5. se presentan los vectores propios de la estructura analizada, como una representación grafica de la configuración de los vectores modales calculados.

Modo 1 $f=2.231\text{Hz}$ Modo 2 $f=8.915\text{Hz}$ Modo 3 $f=19.93\text{Hz}$ Modo 4 $f=34.526\text{Hz}$
Modo 5 $f=49.398\text{Hz}$

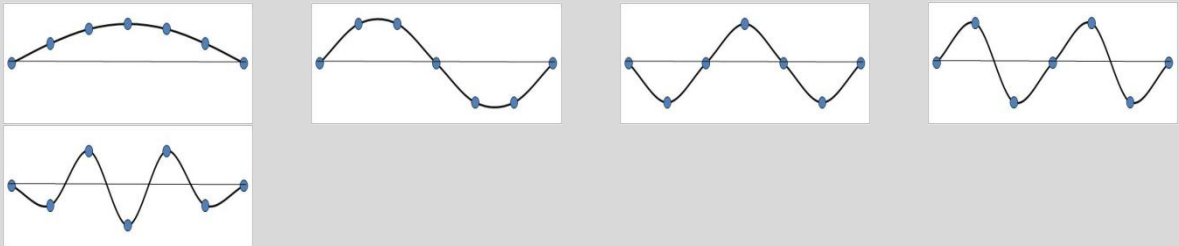


Figura A. 5. Configuraciones de modos de vibración del sistema estructural considerado como ejemplo.

A.2. DETERMINACIÓN DE LA RESPUESTA DINÁMICA POR MEDIO DE OPENSEES DE LAS ESTRUCTURAS ESTUDIADAS

De la teoría estudiada en el presente trabajo y con las referencias que tienen todo lo requerido para poder interpretar y hacer uso de OPENSEES, se presenta a continuación la forma en la que se programo la estructura, con la que se obtuvieron las respuestas dinámicas para las estructuras analizadas.

VIGA

Primero se debe crear el archivo de coordenadas nodales:

1.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:

2.tcl

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

También se deben establecer archivos como el de transformación geométrica:

3.tcl

```
geomTransf Linear 1
```

4.tcl

Archivo de Secciones Transversales y Definición de Elementos Respectivamente:

```
# 4.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
```

5.tcl

```

#Material Concrete 3000 Psi E= 2.153E+10 Mass per Mass per Unit
Length=288.335628
# Element "Element02": eleTag Nodel 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

```

6.tcl

Initialize

Del análisis y el procedimiento anterior se almacenan los resultados de la siguiente manera:

7.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
```

8.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 llama a todos los anteriores archivos y los corre de la siguiente manera:

modelo.tcl

```
##### determinando grados de libertad y dimensiones #####
```

```
model BasicBuilder -ndm 2 -ndf 3
```

```
##### caracterizando la viga #####
```

```
source 1.tcl
source 2.tcl
source 3.tcl
source 4.tcl
source 5.tcl
```

```
##### Inicializando #####
```

```
source 6.tcl
```

```
##### Creando eigenvectores #####
```

```
source 7.tcl
```

```
##### Definiendo análisis y característica #####
```

```
source 8.tcl
```

```
##### Definiendo variables #####
```

```
set eigFID [open EigenVal.out w]
```

```
##### Salidas #####
```

```
puts $eigFID [eigen generalized genBandArpack 8]
```

```
close $eigFID  
analyze 1 0.0001  
wipe
```

Por último para ejecutar el archivo con OPENSEES.exe, simplemente abrimos este programa y escribimos el comando source **modelo.tcl**, para que de esta forma se obtengan 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 analysis 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

```

A.3. ARCHIVOS MATLAB

a. VIGA

ALGORITMO DE INCLUSIÓN DE DAÑO ALEATORIO:

```
cd 'C:\Documents and Settings\ROBINSON BARRERA\Escritorio\viga AG'
clear all

% Escritura de datos
NElemeD = 1; % Numero de elementos dañados 1
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
```

```

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

end

[filanmat] = size(E);

A = 0.12;
I = 0.0016;

fid = fopen('4.tcl', 'wt');

fprintf(fid,'# 4.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);
%%llamando el programa de openses viga masa distribuida %%%
lopensees cuerpo.tcl

%Lectura de datos.

%Frecuencias

Eigen = load('EigenVal.out');
Wne = sqrt(Eigen);
Modos=length(Wne);%%%% cargando modos %%%
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%%%%Modos=4 modos se cambia por el modo a calibrar%%
EigenVece(i,:) = load(['EigenVector_',num2str(i),'.out'])+
load(['EigenVector_',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 'C:\Documents and Settings\ROBINSON BARRERA\Escritorio\viga AG'
clear all

load Wne
load EigenVece
load Elem
load Dano
load E

Modos=length(Wne);

%Porcentaje de Ruido introducido

Ruido1 = 0.15;
Ruido2 = 1;
for i = 1:Modos %%%%%%%%%Modos se cambia por el modo a calibrar%%%%%%%%

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 %%%%%%%%%Modos = 4%%%%%%%%

```

```
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 calibraMACviga:

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

```
cd 'C:\Documents and Settings\ROBINSON BARRERA\Escritorio\viga AG'
```

```
load Wne  
load EigenVece
```

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

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

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

```
    clear EigenVec Eigen
```

```
    A = 0.12;  
    I = 0.0016;
```

```
    fid = fopen('4.tcl', 'wt');
```

```
    fprintf(fid, '# 4.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  lz ');
fprintf(fid,'\n');

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

fclose(fid);

!opensees cuerpo.tcl
clc
%Lectura de datos.
%Frecuencias

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

Modos=4; %length(Wn); %%%
% Modos
for i=1:Modos
EigenVec(i,:) = load(['EigenVector_',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

```

FUNCIÓN OBJETIVO calibraV1viga (Ec. 4.4)

```
function [out] = calibraV1viga(in)

cd 'C:\Documents and Settings\ROBINSON BARRERA\Escritorio\viga AG'
% 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('4.tcl', 'wt');

    fprintf(fid, '# 4.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 cuerpo.tcl
```

```

clc
%Lectura de datos.

%Frecuencias

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

Modos=4; %length(Wn);

% Modos
for i=1:Modos

EigenVec(i,:) = load(['EigenVector_',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 INCLUSIÓN DE DAÑO

```
cd 'C:\Documents and Settings\equipo2\Escritorio\ARMADURA'
```

```
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

[filanmat] = 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 'C:\Documents and Settings\equipo2\Escritorio\ARMADURA'  
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 calibraMACarmadura:

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

cd 'C:\Documents and Settings\equipo2\Escritorio\ARMADURA'
% 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

    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 calibraV1armadura:

```
function [out] = calibraV1armadura(in)

cd 'C:\Documents and Settings\equipo2\Escritorio\ARMADURA'
% 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

```