

**IDENTIFICACIÓN DE DAÑO EN PÓRTICOS 2D EMPLEANDO INFORMACIÓN MODAL
Y EL MÉTODO PARTICLE SWARM OPTIMIZATION (PSO)**

**FREDY SAÚL SOTELO MONROY
JAVIER ERNESTO ACEVEDO GUTIERREZ**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECAICAS
ESCUELA DE INGENIERÍA CIVIL
BUCARAMANGA
2009**

**IDENTIFICACIÓN DE DAÑO EN PÓRTICOS 2D EMPLEANDO INFORMACIÓN MODAL
Y EL MÉTODO PARTICLE SWARM OPTIMIZATION (PSO)**

**FREDY SAÚL SOTELO MONROY
JAVIER ERNESTO ACEVEDO GUTIERREZ**

**Trabajo de grado presentado como requisito para optar el título de
Ingeniero Civil**

**Director:
OSCAR JAVIER BEGAMBRE CARRILLO
Ingeniero Civil, MsC PhD**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECAICAS
ESCUELA DE INGENIERÍA CIVIL
BUCARAMANGA
2009**

*A mis padres y hermanos,
porque con su apoyo siempre
he podido alcanzar mis metas.*

*A mis sobrinos Matías y Mafe,
porque son mi inspiración.*

Fredy

A mis padres y hermanos, que fueron el motor del desarrollo en el camino de mi vida.

A mi sobrino David, que me lleno de alegría en cada instante.

Javier

AGRADECIMIENTOS

Al profesor Oscar Javier Begambre Carrillo, por tener el privilegio de contar con su orientación durante la realización de este trabajo.

A la Universidad Industrial de Santander que nos brindo un espacio, en el cual logramos desarrollar nuestras capacidades y formarnos como mejores personas.

A la señora Aura Rosario Gutiérrez, por su colaboración en la búsqueda de la información.

TABLA DE CONTENIDO

INTRODUCCIÓN.....	1
1. OBJETIVOS.....	2
1.1. GENERAL.....	2
1.2. ESPECIFICOS.....	2
2. ANTECEDENTES.....	3
2.1. MÉTODOS BASADOS EN LA FRECUENCIA NATURAL.....	5
2.2. MÉTODOS BASADOS EN LA FORMA MODAL.....	5
2.2.1. Comparación directa de las formas modales.....	5
2.2.2. Curvatura.....	6
2.3. FORMAS DE DEFLEXIÓN OPERACIONAL.....	7
2.4. ENERGÍA DE DEFORMACIÓN MODAL.....	7
2.5. MÉTODOS BASADOS EN LA MATRIZ DE FLEXIBILIDAD DINÁMICAMENTE MEDIDA.....	8
2.6. MÉTODO DEL VECTOR DE FUERZA RESIDUAL.....	8
2.7. MÉTODO DE REDES NEURONALES.....	9
2.8. MÉTODO DE ALGORITMOS GENÉTICOS (AG) Y OTRAS TÉCNICAS DE OPTIMIZACIÓN.....	10
2.9. MÉTODOS ESTADÍSTICOS.....	11
3. MARCO TEÓRICO.....	13
3.1. PÓRTICO.....	13
3.2. MATRIZ DE RIGIDEZ.....	14
3.2.1. Método matricial.....	14
3.2.1.1. Método de la rigidez.....	15
3.2.1.2. Elementos de la matriz de rigidez.....	16
3.2.1.2.1. Miembros de carga axial.....	16
3.2.1.2.2. Miembros a flexión.....	16
3.2.2. Método de elementos finitos.....	20
3.2.2.1. Principios básicos.....	20
3.2.2.2. Elección de tamaños y formas de los elementos.....	21
3.2.2.3. Elección de la función de forma de desplazamiento.....	22
3.2.2.4. Grados de libertad nodales.....	22
3.2.2.5. Función de forma del elemento.....	22

3.2.2.6.	Formulación de la matriz de rigidez para una viga	22
3.2.2.6.1.	Componentes a flexión	22
3.2.2.6.2.	Componentes axiales.....	28
3.2.3.	Rotación de ejes.....	31
3.2.4.	Matriz de rigidez de la estructura	34
3.3.	MATRIZ DE MASA.....	36
3.4.	MATRIZ DE AMORTIGUACIÓN	41
3.5.	DINÁMICA ESTRUCTURAL	41
3.5.1.	Ecuación de movimiento	42
3.5.2.	Vibración libre no amortiguada.....	43
4.	ALGORITMO PARTICLE SWARM OPTIMIZATION (PSO).....	46
4.1.	FUNDAMENTOS DEL MOVIMIENTO DE PARTÍCULAS.....	47
4.2.	OPERADOR DE VELOCIDAD Y PARÁMETROS DEL ALGORITMO.....	47
4.3.	SELECCIÓN DE PARÁMETROS	50
4.4.	ENJAMBRE INICIAL	51
4.5.	CRITERIO DE CONVERGENCIA.....	51
4.6.	EJEMPLO DE APLICACIÓN.....	53
4.7.	EVALUACIÓN DEL ALGORITMO PSO CON FUNCIONES DE LA LITERATURA INTERNACIONAL	62
4.7.1.	Resultados.....	65
4.7.1.1.	Función VENTER	66
4.7.1.2.	Función N-DIMENSIONAL (5D)	66
4.7.1.3.	Función SCHWEFEL (5D).....	67
4.7.1.4.	Función ROSENBROCK (5D)	68
5.	IMPLEMENTACIÓN DEL ALGORITMO PSO PARA LA DETECCIÓN DE DAÑO.....	70
5.1.	SIMULACION DEL DAÑO.....	70
5.1.1.	Escenarios de daño.....	71
5.1.1.1.	Daño simple.....	71
5.1.1.2.	Daño múltiple.....	72
5.2.	FUNCIONES OBJETIVO	74
5.3.	RESULTADOS NUMÉRICOS.....	76
5.3.1.	DAÑO SIMPLE I	77
5.3.2.	DAÑO SIMPLE II	81

5.3.3.	DAÑO SIMPLE III	85
5.3.4.	DAÑO SIMPLE IV.....	89
5.3.5.	DAÑO SIMPLE V.....	93
5.3.6.	DAÑO MULTIPLE I.....	97
5.3.7.	DAÑO MULTIPLE II.....	101
5.3.8.	DAÑO MULTIPLE III.....	105
5.3.9.	DAÑO MULTIPLE IV	109
5.3.10.	DAÑO MULTIPLE V	113
5.3.11.	Comparación PSO con los AG CoBin2, CoRed1 y CoRea1	117
6.	CONCLUSIONES	120
7.	RECOMENDACIONES	121
	BIBLIOGRAFIA	122
	ANEXOS	126

LISTA DE TABLAS

Tabla 1. Propiedades del perfil H.E.B. 220	14
Tabla 2. Resultados función Venter 2D.....	66
Tabla 3. Resultados función N-Dimensional 5D.....	67
Tabla 4. Resultados función Schwefel 5D.....	68
Tabla 5. Resultados función Rosenbrock 5D	68
Tabla 6. Casos de daño simple	71
Tabla 7. Casos de daño múltiple	73
Tabla 8. Resultados DAÑO SIMPLE I.....	77
Tabla 9. Resultados DAÑO SIMPLE II.....	81
Tabla 10. Resultados DAÑO SIMPLE III.....	85
Tabla 11. Resultados DAÑO SIMPE IV.....	89
Tabla 12. Resultados DAÑO SIMPLE V.....	93
Tabla 13. Resultados DAÑO MULTIPLE I.....	97
Tabla 14. Resultados DAÑO MULTIPLE II.....	101
Tabla 15. Resultados DAÑO MULTIPLE III.....	105
Tabla 16. Resultados DAÑO MULTIPLE IV	109
Tabla 17. Resultados DAÑO MULTIPLE V	113
Tabla 18. Resultados escenario de daño P3C.....	118

LISTA DE FIGURAS

Figura 1. Pórtico	13
Figura 2. Pórtico propuesto para la detección de daños	13
Figura 3. Geometría del perfil H.E.B. 220	14
Figura 4. Viga simplemente apoyada sometida a momentos $W1$ y $W2$ en sus extremos..	15
Figura 5. Elemento cargado axialmente.....	16
Figura 6. Elemento tipo viga sometido a flexión.....	17
Figura 7. Coeficientes de rigidez del elemento tipo viga	18
Figura 8. Ensamblaje de subdivisiones	20
Figura 9. Formas básicas de los elementos para discretizar el modelo.....	21
Figura 10. Viga simplemente apoyada.....	23
Figura 11. Diagrama de cuerpo libre de una sección infinitesimal de viga	23
Figura 12. Grados de libertad de un elemento de viga sometido a flexión	25
Figura 13. Funciones de forma para elementos sometido a flexión.....	25
Figura 14. Grados de libertad de un elemento de viga sometido a carga axial	29
Figura 15. Funciones de forma para elementos sometidos a carga axial	29
Figura 16. Desplazamientos de los grados de libertad referidos al sistema de referencia local	32
Figura 17. Fuerzas de los grados de libertad referidos al sistema de referencia local	32
Figura 18. Desplazamientos de los grados de libertad referidos al sistema de referencia global.....	33
Figura 19. Numeración de nodos y elementos del pórtico	35
Figura 20. Viga en vibración.....	36
Figura 21. Elemento finito de una viga en vibración.....	38
Figura 22. Sistema de un grado de libertad sometido a una excitación externa $f(t)$	42
Figura 23. Diagrama de flujo PSO básico	52
Figura 24. Función del ejemplo de aplicación	53
Figura 25. Distribución de las partículas iniciales dentro de la función.....	56
Figura 26. Primera actualización de las partículas.....	58
Figura 27. Segunda actualización de las partículas	61
Figura 28. Ultima actualización de las partículas	62
Figura 29. Función VENTER 2D	63
Figura 30. Función N-DIMENSIONAL 2D	64

Figura 31. Función Schwefel 2D	64
Figura 32. Función Rosenbrock 2D.....	65
Figura 33. Pórtico para la detección de daños.....	70
Figura 34. Casos de daño simple.....	72
Figura 35. Casos de daño múltiple.....	74
Figura 36. Resultados DAÑO SIMPLE I.....	78
Figura 37. Resultados DAÑO SIMPLE II.....	82
Figura 38. Resultados DAÑO SIMPLE III.....	86
Figura 39. Resultados DAÑO SIMPE IV	90
Figura 40. Resultados DAÑO SIMPLE V	94
Figura 41. Resultados DAÑO MULTIPLE I	98
Figura 42. Resultados DAÑO SIMPLE II.....	102
Figura 43. Resultados DAÑO MULTIPLE III	106
Figura 44. Resultados DAÑO MULTIPLE IV	110
Figura 45. Resultados DAÑO MULTIPLE V.....	114
Figura 46. Aplicación de los AG al escenario de daño P3C.....	117
Figura 47. Aplicación del PSO al escenario de daño P3C	117

LISTA DE ANEXOS

ANEXO A. RESULTADOS PSO DEL EJEMPLO DE APLICACIÓN	126
ANEXO B. CODIGO PSO	129
ANEXO C. CODIGO PARA LA SIMULACION DEL PORTICO	133
ANEXO D. VALIDACION DE LA SIMULACION DEL PORTICO CON ANSYS	137
ANEXO E. CODIGO PORTICO-PSO	138

RESUMEN

TITULO:

IDENTIFICACIÓN DE DAÑO EN PÓRTICOS 2D EMPLEANDO INFORMACIÓN MODAL Y EL MÉTODO PARTICLE SWARM OPTIMIZATION (PSO). *

AUTORES:

FREDY SAUL SOTELO MONROY. **

JAVIER ERNESTO ACEVEDO GUTIERREZ. **

PALABRAS CLAVES:

Detección de Daños, Optimización con Enjambre de Partículas, Dinámica Estructural.

DESCRIPCIÓN:

La evaluación del estado de las estructuras es un tema de investigación muy importante para diversos campos de la ingeniería y, por lo tanto, se están desarrollando metodologías para la detección de daños en estructuras. El presente trabajo tiene como finalidad estudiar el desempeño del algoritmo de optimización con enjambre de partículas para la identificación de daños en un pórtico 2D, a partir de ciertas características vibracionales del sistema estructural como las frecuencias y los periodos de vibración.

Las características vibracionales de la estructura, se determinan a partir de la dinámica estructural y un modelo de elementos finitos (MEF). El daño presente en un elemento del pórtico 2D es simulado numéricamente mediante un parámetro escalar de daño que cuantifica la pérdida de rigidez. Se estudiaron escenarios de daño simple y daño múltiple. Para facilitar la implementación numérica planteada a lo largo del trabajo se utiliza, la herramienta computacional denominada MATLAB[®].

La identificación del daño presente en la estructura se define como la detección, localización y cuantificación de la magnitud del daño. En el análisis de los resultados se compara la metodología propuesta del PSO con una aplicación de los algoritmos genéticos (AG) encontrada en la literatura especializada, mostrando la capacidad del PSO para ser aplicado al problema de detección de daños en estructuras.

* Proyecto de Grado

** Facultad de Ingenierías Fisicomecánicas, Escuela de Ingeniería Civil. Director: Oscar Begambre

ABSTRACT

TITLE:

DAMAGE IDENTIFICATION IN A TWO-DIMENSIONAL FRAME USING MODAL INFORMATION AND THE PARTICLE SWARM OPTIMIZATION ALGORITHM (PSO). *

AUTHORS:

FREDY SAUL SOTELO MONROY. **

JAVIER ERNESTO ACEVEDO GUTIERREZ. **

KEYWORDS:

Damage Identification, Particle Swarm Optimization, Structural Dynamic.

DESCRIPTION:

The structural health assessment and monitoring is a very important investigation topic on different engineering fields. In this context, several methodologies are being developed to solve the damage detection in structures. In this study, the performance of the Particle Swarm Optimization Algorithm (PSO) on the damage identification in a 2D steel frame using dynamic responses was evaluated.

The vibrational characteristics of the structure were determined from the structural dynamic theory and using a Finite Element Model (FEM). The damage, at element level, was numerically simulated by means of a damage scalar parameter that quantifies the stiffness loss. Simple and multiple damage scenarios were studied. To make possible the numerical implementation, the computer language MATLAB® was used.

The damage identification in the 2D steel frame was defined as the localization and the stiffness loss quantification of the frame affected elements. The proposed methodology was compared with an application of the Genetic Algorithms (GA) found in specialized literature and the presented numerical results establish the PSO level of confidence on solving the damage detection problem in structures.

* Degree Work

** Physical-Mechanical Engineering Faculty, Department of Civil Engineering. Director: Oscar Begambre

INTRODUCCIÓN

La necesidad de detectar los daños estructurales en sus primeras etapas ha sido de gran interés en las áreas de la ingeniería civil, mecánica y el sector aeroespacial, observando de esta forma como las industrias quieren detectar los daños en su infraestructura lo más rápidamente posible, dado el gran beneficio económico y la posible reducción de riesgos para la vida humana, que esta tecnología potencialmente ofrece. El daño puede ser definido en términos generales como los cambios introducidos en un sistema que perjudica su actual o futuro comportamiento.

Los primeros métodos de detección de daños fueron experimentales, como por ejemplo la inspección visual, el análisis de imágenes tomográficas, métodos acústicos o de ultrasonido, métodos de campo magnético, rayos X, corriente de Eddy y métodos térmicos sobre el terreno. Todas estas técnicas experimentales requieren que las proximidades de los daños se conozcan con anticipación y que la parte de la estructura a inspeccionar sea accesible. Además de estas limitaciones, estos métodos solo pueden detectar daños en o cerca de la superficie de la estructura (Nagy, 2003; Riveros, 2007).

En las últimas tres décadas se han venido desarrollando investigaciones para la detección de daños en estructuras, principalmente analizando sus características dinámicas, como son las frecuencias y los modos de vibración.

Actualmente las metodologías encontradas de monitoreo y detección de daños utilizan estrategias que involucran la respuesta dinámica para identificar la existencia, ubicación y magnitud de los daños.

Es en este contexto donde se ubica el presente trabajo, el cual por medio de la modelación de un pórtico simple en dos dimensiones, con algunos daños localizados modelados como la disminución de la rigidez, y mediante la aplicación del algoritmo Particle Swarm Optimization (PSO), pretender determinar la ubicación del daño. De esta forma, se plantea la viabilidad de la aplicación del método de ubicación y detección de daños en estructuras más complejas. Este trabajo busca también, de forma general, resaltar y dar a conocer la importancia y los beneficios que se podrían obtener al aplicar las técnicas actuales de detección de daños, además de estimular su investigación.

1. OBJETIVOS

1.1. GENERAL

Identificar los daños inducidos en un pórtico de dos dimensiones, cuyo comportamiento lineal es simulado numéricamente, mediante la aplicación de algoritmo de optimización PSO, tomando como referencia las características dinámicas del sistema estructural.

1.2. ESPECIFICOS

- Documentar el estado del arte de la detección de daños aplicado en estructuras, realizando un análisis de los métodos encontrados en la literatura mundial y nacional.
- Presentar de forma ilustrativa los principios del PSO mediante un ejemplo.
- Programar en MATLAB® el algoritmo PSO propuesto por Shi y Eberhart (2001), e implementarlo para la determinación de mínimos en varias funciones matemáticas multimodales reportadas en la literatura, y posteriormente a la detección de daños.
- Desarrollar un conjunto de herramientas computacionales que faciliten al usuario, el ingreso y la obtención de datos para la simulación del pórtico y del problema de detección de daños a partir de la respuesta dinámica del pórtico.
- Determinar la sensibilidad de 3 funciones objetivo para cuantificar el daño en el pórtico 2D.
- Evaluar el desempeño de la metodología propuesta en diversos escenarios de daño y comparar sus resultados con otras técnicas propuestas en la literatura especializada (en este caso en particular con AG's).

2. ANTECEDENTES

La supervisión de la salud estructural aparece como resultado de los departamentos de ingeniería de las diferentes empresas principalmente del sector aeroespacial y la industria del petróleo en su continuo esfuerzo por aumentar, en lo posible, la expectativa de vida operacional de los diferentes sistemas, así como incrementar la eficiencia y seguridad de los mismos (*Atienza, 2004; Carden y Fanning, 2004*).

Inicialmente se utilizaban técnicas locales de inspección para estructuras civiles tales como métodos visuales, métodos ultrasónicos, métodos magnéticos y radiografías entre otros. Las principales desventajas de estos métodos radican en que se debe conocer la ubicación del daño a priori y allí son relativamente de alto nivel de dependencia de las habilidades prácticas de los ingenieros que realizan las inspecciones estructurales. Por lo tanto métodos visuales de inspección no podrían ser la mejor solución para estructuras civiles cuando las condiciones de accesibilidad representan un peligro potencial (*Nagy, 2003; Riveros, 2007*).

Un acercamiento innovador es la supervisión basada en el análisis de las características dinámicas de los sistemas estructurales (frecuencia o modos de vibración) con el fin de detectar cambios, lo que puede indicar daño o degradación, dichas técnicas comenzaron a implementarse en los finales de los años setenta y principios de los ochenta, pero la dificultad en la instrumentación para localizar e identificar los daños obstaculizaron considerablemente el desarrollo de estos métodos. Actualmente, en la ingeniería civil, la supervisión de infraestructuras basadas en sus propiedades dinámicas es un área de investigación debido al gran impacto económico e implicaciones de seguridad relacionada con una temprana detección de daño estructural. La mayor parte de la literatura existente se encuentra enfocada a infraestructuras de puentes (*Carden y Fanning, 2004; Riveros, 2007*).

Generalmente un sistema típico de monitoreo de salud estructural, mejor conocido como Structural Health Monitoring (SHM) incluye tres componentes principales: un sistema de sensores, un sistema informático (incluyendo adquisición de datos y almacenaje) y un sistema de evaluación de salud estructural (algoritmos de diagnóstico) (*Li, Li y Song, 2004*). Una gran variedad de sensores son utilizados para medir la respuesta de la estructura, por ejemplo: acelerómetros y extensómetros. Por ejemplo *Marulanda et al. (2000)* realizaron la implementación para la adquisición de datos en el tiempo del puente Pereira-Dos quebradas mediante tres subsistemas. El primero se encargo del monitoreo de procesos lentos, tales como asentamientos de fundaciones y cambios de temperatura, el segundo subsistema se encargo de monitorear procesos rápidos como los producidos por las acciones dinámicas y el ultimo subsistema estaba encargado de monitorear la corrosión del concreto. En 2005 el sistema de identificación de daños fue propuesto empleando dos algoritmos implementados en MATLAB[®] para la identificación de los parámetros modales: la Técnica de Excitación Natural (NEXt) y el algoritmo de Realización de Valores Propios (ERA). Por otro lado, *Li et al. (2004)* realizaron un estudio de la reciente aplicación de sensores de fibra óptica enfocados a la supervisión de la salud estructural ya que la mayor parte de los sensores convencionales usados se basan en la transmisión de señales eléctricas. Estos sistemas presentan algunas limitaciones. Los mencionados autores afirman que los sensores de fibra óptica son prometedores alternativas en sistemas SHM y futuras estructuras inteligentes. Dentro de las recientes

aplicaciones de los sensores de fibra óptica se encontraron edificios, pilas, puentes, tuberías, túneles y presas. Aunque el progreso de la fibra óptica ha sido impresionante, falta investigación, sobre todo para mejorar su desempeño.

Con relación a los datos medidos, se sabe que pueden ser convertidos a partir del dominio del tiempo al dominio de la frecuencia utilizando la transformada de Fourier. Un análisis más detallado de los datos del dominio de la frecuencia se lleva a cabo para extraer los parámetros modales para producir lo que se denomina dominio de datos modales. A pesar de que las mediciones generalmente se realizan en el dominio del tiempo, el investigador puede optar por analizar los datos en el dominio del tiempo, la frecuencia o dominio modal. La mayoría de la literatura consultada hasta la fecha se ha centrado en los métodos basados en el dominio modal. Esto es probablemente debido a dos razones principales. En primer lugar, los primeros estudios se centraron en los dominios modales, y en segundo lugar, frecuencias naturales y formas modales son fácilmente identificables en sistemas lineales. La investigación de los métodos basados en los tres dominios, se encuentra en desarrollo, sobre todo porque todavía no se ha encontrado un método que identifique a cada tipo de daño en cada tipo de estructura (Carden y Fanning, 2004).

Como resultado de los diversos retos que ofrecen los diferentes sistemas estructurales y el importante esfuerzo de investigación que se ha aplicado al monitoreo estructural, ha surgido una amplia gama de técnicas, métodos y algoritmos. Las técnicas de monitoreo de la salud estructural según Rytter (1993), pueden ser categorizadas en cuatro niveles según su alcance (apud Carden y Fanning, 2004; Marulanda, Thomson y Marulanda, 2000; Riveros, 2007):

Nivel 1: Determinación de que el daño está presente en la estructura.

Nivel 2: Determinación de la ubicación geométrica de los daños.

Nivel 3: Cuantificación de la gravedad de los daños.

Nivel 4: Predicción de la vida útil restante de la estructura.

Con la aparición de aleaciones de memoria, en el contexto de las estructuras inteligentes, Park (2001) sugiere un quinto nivel en el esquema de monitoreo que incluya la llamada "auto regulación de estructuras" (apud Carden y Fanning, 2004).

Doebbling et al. (1996) presentaron un amplio estudio de los métodos de detección de daños a partir de la respuesta dinámica de la estructura. La literatura revisada por dichos autores se concentró principalmente en los niveles 1,2 y 3. La mayor parte de la literatura se centró en estructuras de laboratorio controlando los daños a las estructuras sobre el terreno, sin tener en cuenta el nivel 4. Este nivel de identificación es, sin duda, el objetivo final de cualquier sistema de monitoreo estructural. Según estos autores, dependiendo del tipo de datos medidos y de la técnica empleada para la identificación del daño, los métodos de detección de daño se pueden clasificar en los siguientes grupos (apud Begambre, 2004; Carden y Fanning, 2004):

Métodos basados en la frecuencia natural.

Métodos basados en la forma modal.

Métodos basados en la curvatura de la forma modal.

Métodos basados en la medición de la flexibilidad dinámica.

Métodos de redes neurales.

Métodos de algoritmos genéticos (AG) y otras técnicas de optimización.

Métodos estadísticos.

A continuación, se presenta una breve reseña de cada uno de los métodos arriba mencionados.

2.1. MÉTODOS BASADOS EN LA FRECUENCIA NATURAL

La relación física entre rigidez, masa y frecuencias naturales junto con la facilidad de la medición de estas (solo un sensor se requiere en muchas aplicaciones) fue el impulso para la utilización de estos métodos. Donde, básicamente, a partir de variaciones de la frecuencia natural es posible identificar la presencia, estimar la localización y cuantificar el daño en una estructura. Esta técnica se clasifica dentro de los métodos que requieren datos experimentales y modelos analíticos.

Kunert y Ambrosini (2007) propusieron tres funciones para describir la variación del módulo de elasticidad con la coordenada espacial en vigas de hormigón armado, caracterizando la distribución espacial del daño estructural. En cuanto al modelo teórico-numérico se utilizó una herramienta computacional desarrollada por R. D. Ambrosini (1991), llamada FREC. FREC es un programa que incorpora una teoría general de vigas y el método de variables de estado para encontrar las frecuencias de vibración de diferentes tipos de estructuras. Estos autores realizaron una nueva versión, llamada FREC-DAMAGE que encuentra las frecuencias de vibración variando el parámetro de daño en cualquiera de las tres funciones de daño presentadas. En el estudio presentado, se utilizó un método de integración numérica conjuntamente con el algoritmo de la transformada rápida de Fourier (FFT) para trabajar en el dominio de la frecuencia. Los resultados obtenidos son comparados con el modelo teórico-numérico, sometido a distintos escenarios de daño, inducidos artificialmente.

2.2. MÉTODOS BASADOS EN LA FORMA MODAL

La medición de las formas modales de una estructura requiere ya sea un solo punto de excitación y muchos sensores o un excitador itinerante con uno o más sensores fijos. Muchos análisis modales disponen de técnicas para la extracción de la forma modal a partir de los datos medidos en el dominio del tiempo. Los métodos de detección se han desarrollado mediante la identificación directa de las formas modales o las curvaturas de las formas modales.

2.2.1. Comparación directa de las formas modales

Dos métodos comúnmente utilizados para comparar dos conjuntos de la forma modal son el Criterio de Confianza Modal (MAC) (Allemang and Brown, 1982) y el Criterio de Confianza de Coordenadas Modales (COMAC) (Lieve et al., 1988). La evaluación MAC puede ser considerada como una medida de la similitud de dos formas modales. Un valor de $MAC=1$ es una combinación perfecta y un valor de $MAC=0$ significa que son completamente distintas. Por lo tanto, la reducción de un valor de MAC puede ser un

indicador de daño. El COMAC es una medida de la diferencia entre dos conjuntos de formas modales que toman un valor entre 1 y 0. Un bajo valor de COMAC indicaría discordancia en un punto y, por lo tanto, también es un posible indicador de daño (*apud Begambre, 2004; Carden y Fanning, 2004*).

2.2.2. Curvatura

El uso de la curvatura en la identificación de daños se basa en el supuesto de que los cambios en las curvaturas de las formas modales son muy localizados en la región de los daños, de esta manera la reducción del EI (modulo de elasticidad e inercia) producen un aumento de la curvatura en la sección. Dado que las variaciones de la curvatura son locales y dependen de la reducción del EI, se pueden utilizar para detectar, localizar y cuantificar los daños. Por lo tanto, se espera que la diferencia absoluta entre la curvatura de las formas modales de la estructura sin daño y la estructura dañada nos proporcionará la región dañada. Además, la máxima diferencia se incrementa a medida que el grado de reducción de la rigidez aumenta en la zona dañada. La curvatura se calcula a partir de la forma modal mediante una aproximación de diferencia central como lo muestra la ecuación (2.1) (*Carden y Fanning, 2004*):

$$\Phi_{ij}^{\text{II}} = \frac{\Phi_{(i+1)j} - 2\Phi_{ij} + \Phi_{(i-1)j}}{L^2} \quad (2.1)$$

Donde:

Φ_{ij}^{II} = Elemento i del vector de curvatura j .

ϕ_{ij} = Elemento i del vector de la forma modal.

L = Longitud entre los nodos.

La diferencia absoluta de la curvatura (DC) se calcula como:

$$DC = |\Phi_k^{\text{II}} - \Phi_k^{\text{IID}}| \quad (2.2)$$

Donde:

Φ_k^{II} = Curvatura de la forma modal k de la estructura sin daño.

Φ_k^{IID} = Curvatura de la forma modal k de la estructura dañada.

Para tener en cuenta todos los modos medidos, De Roeck Wahab (1999) propone un indicador de daño llamado Factor de Daño de Curvatura (FDC), que se refiere a la diferencia de la curvatura antes y después de los daños haciendo un promedio sobre el número de modos, como se muestra en la ecuación (2.3) (*apud Begambre, 2004*):

$$FDC = \frac{1}{n} \sum_{k=1}^n |\Phi_k^{\text{II}} - \Phi_k^{\text{IID}}| \quad (2.3)$$

Donde:

Φ_k^{II} = Curvatura de la forma modal k de la estructura sin daño.

Φ_k^{IID} = Curvatura de la forma modal k de la estructura dañada.

n = Número total de modos considerados.

El número de modos y de curvaturas utilizado por las rutinas de identificación de daños es, naturalmente, limitado al número de desplazamientos disponibles de la forma modal. En conclusión, las formas modales y sus derivadas han sido ampliamente utilizadas para identificar los daños. Sin embargo, existen algunas contradicciones sobre la utilidad de las formas modales por sí solas en la detección de daños. Esta incertidumbre ha dado lugar a la investigación de otros métodos como la utilización de las formas de deflexión operacionales, que tiene muchas similitudes con las formas modales. Además, formulaciones más complejas que implican el uso de las formas modales también han sido investigadas, como métodos de energía de deformación modal. Dentro de esta categoría se encuentran la flexibilidad dinámicamente medida y el vector de fuerza residual, que combinan el uso de frecuencias naturales y de las formas modales (*Carden y Fanning, 2004*).

2.3. FORMAS DE DEFLEXIÓN OPERACIONAL

Formas de deflexión operacional, más conocida en la literatura como Operational Deflection Shapes (ODS), dependen de la ubicación y las magnitudes relativas de las fuerzas aplicadas a la estructura. Si la estructura está muy excitada en un solo lugar cerca de la resonancia, entonces la forma modal y la ODS serán similares (*Carden y Fanning, 2004*).

2.4. ENERGÍA DE DEFORMACIÓN MODAL

Cuando un modo de vibración particular almacena una cantidad grande de energía de deformación en una trayectoria de carga en particular, la frecuencia y la forma de aquel modo son sumamente sensibles a cambios de aquella trayectoria de carga. Así, los cambios de la energía de deformación modal también podrían ser considerados como opción lógica de indicador de la ubicación de daño. La literatura generalmente se concentraba en métodos de deformación en 1-D, aunque estos sean aplicables a estructuras de 2-D y 3-D, que se pueden descomponer en elementos parecidos a una viga. La energía de deformación en una viga de Bernoulli-Euler asociada con una forma modal particular puede ser calculada a partir de:

$$U_i = \frac{1}{2} \int_0^1 EI \left(\frac{\partial^2 \phi_i}{\partial x^2} \right)^2 \quad (2.4)$$

Donde:

U_i = Energía de deformación del modo i .

E = Modulo de elasticidad del material.

I = Momento de inercia de la sección transversal.

ϕ_i = Curvatura del modo de vibración i .

La curvatura requerida para este cálculo comúnmente es extraída de los desplazamientos de la forma modal usando una aproximación de diferencia central.

2.5. MÉTODOS BASADOS EN LA MATRIZ DE FLEXIBILIDAD DINÁMICAMENTE MEDIDA

La matriz de flexibilidad es definida como el inverso de la matriz de rigidez y, por lo tanto, relaciona la fuerza estática aplicada con el desplazamiento resultante. Así, cada columna de la matriz de flexibilidad representa el desplazamiento asociado con una fuerza unitaria aplicada en los grados de libertad (GDL) asociados.

La matriz de flexibilidad dinámicamente medida, generalmente es estimada como:

$$[F] = [\Phi][\lambda]^{-1}[\Phi]^T \quad (2.5)$$

Donde:

$[F]$ = Matriz de flexibilidad.

$[\Phi]$ = Matriz de formas modales.

$[\lambda]$ = Matriz diagonal de frecuencias de vibración.

Debido a las dificultades prácticas en la medición de los modos más altos, la matriz de flexibilidad generalmente es estimada usando solamente los modos inferiores. Además, debido a la relación inversa al cuadrado de las frecuencias de vibración, la matriz de flexibilidad es más sensible a cambios en los modos de frecuencias inferiores. Al igual que otros métodos de detección de daños no hay un amplio consenso sobre las ventajas de utilizar la flexibilidad dinámicamente medida frente a otros parámetros estructurales (*Carden y Fanning, 2004*).

Patjawit y Kanok-Nukulchai (2005) realizaron un estudio con el fin de verificar la aplicación una técnica para la determinación de los deterioros a nivel global de un puente construido en 1999 sobre un canal en la provincia Nakhon-Nayok (Tailandia), usando la combinación de una prueba de impacto simple y un modelo de elementos finitos. Esta técnica se basa en el supuesto de que la deterioración estructural reduce la rigidez y aumenta la flexibilidad, por lo tanto el aumento de la flexibilidad estructural puede servir como un buen indicador del grado de deterioración estructural. En este estudio, se utilizó la norma de la matriz de flexibilidad para ser empleada como un índice para reflejar la flexibilidad global de la estructura del puente. Este índice se denominó Índice de Flexibilidad Global (IFG). Los autores llegaron a la conclusión que la técnica propuesta a partir del análisis de las repuesta vibratoria de la estructura podría ser aplicable, sin embargo, una mínima alteración de frecuencias y formas modales no puede ser suficientemente significativa para deducir la condición de la salud de la estructura.

2.6. MÉTODO DEL VECTOR DE FUERZA RESIDUAL

Con el acceso a las formas modales y frecuencias naturales es posible formular lo que se conoce como Vector de Fuerza Residual (VFR). Frecuencias naturales y formas modales satisfacen la ecuación del valor propio, donde considerando el modo i de la estructura dañada tenemos:

$$(K_d - \lambda_{ai}M_d)\Phi_{di} = 0 \quad (2.6)$$

Con λ_{di} siendo el cuadrado de la frecuencia natural, y Φ_{di} la i -ésima forma modal de la estructura dañada, siendo medida y por lo tanto conocida para varios modos. Asumiendo la matriz de rigidez como K_d y la matriz de masa como M_d , las matrices asociadas con la estructura dañada son definidas como:

$$K_d = K_a + \Delta K \quad (2.7)$$

$$M_d = M_a + \Delta M \quad (2.8)$$

Donde:

K_a = Matriz de rigidez de la estructura sin daño.

ΔK = Variación de la matriz de rigidez causada por el daño.

M_a = Matriz de masa de la estructura sin daño.

ΔM = Variación de la matriz de masa causada por el daño.

Sustituyendo las ecuaciones (2.7) y (2.8) en la ecuación (2.6) y reorganizando, se llega a la definición del Vector de Fuerza Residual R_i para modo i , donde el VFR está dado por:

$$R_i = (K_a - \lambda_{di}M_a)\Phi_{di} \quad (2.9)$$

Donde:

K_a = Matriz de rigidez de la estructura sin daño.

M_a = Matriz de masa de la estructura sin daño.

λ_{di} = Matriz diagonal de las frecuencias al cuadrado.

Φ_{di} = i -ésima forma modal de la estructura dañada.

Cada modo proporciona un solo VFR. Este vector físicamente puede ser interpretado como la excitación de la fuerza armónica que tendría que ser aplicada a la estructura intacta, representada por la matriz de rigidez K_a , la matriz de masa M_a y la frecuencia $\sqrt{\lambda_{di}}$ de tal manera que la estructura responda con la forma modal Φ_{di} .

Cada VFR representa un solo grado de libertad de modelo numérico de la estructura. Cuando un daño ocurre en un elemento unido a este grado de libertad, el valor del VFR se hace muy grande comparado con los demás valores asociados a los restantes grados de libertad donde ningún daño ha ocurrido. Al elegir dichas condiciones se proporciona un método para la identificación de daños (*Carden y Fanning, 2004*).

2.7. MÉTODO DE REDES NEURONALES

Las redes neuronales surgieron del estudio de las neuronas biológicas y se refieren a una estructura computacional que simula el funcionamiento del sistema nervioso biológico, que tiene como finalidad solucionar problemas complejos, modelar sistemas en los cuales no se conoce el comportamiento real, o en procesos de clasificación de patrones. Gracias a estas características, muchos investigadores han usado las redes neuronales artificiales la identificación de daños en estructuras (*Carden y Fanning, 2004; Hernández, 2005*).

Hernández (2005) estudio el desempeño de cuatro redes neuronales backpropagation para identificar escenarios analíticos de daño (simulados como reducción del módulo de

Young) en tres modelos estructurales tridimensionales, donde se consideraron dos escenarios de daño, el primero como daño simple y el segundo como daño múltiple. En el estudio se consideraron dos características vibracionales de los sistemas estructurales que fueron implementadas. El primer parámetro fue el periodo de vibración de la estructura, y el segundo, fueron los registros de los desplazamientos de la estructura al ser sometida a una excitación en la base. Todos los datos fueron obtenidos a partir de un análisis de elementos finitos de los modelos estructurales.

La primera red neuronal se diseñó específicamente para determinar los niveles de daño a partir del periodo de vibración, la segunda y la tercera para identificar el daño basándose en los registros vibracionales y la cuarta para que, a partir del periodo y los registros de vibración, identificar el daño. Como resultado se obtuvieron menores errores en la implementación al modelo estructural, correspondiente al escenario de daño simple, comparados con los escenarios de daño múltiple, donde los errores aumentaron, aunque la diferencia fue poco significativa. Concluyendo que las redes neuronales backpropagation, pueden emplearse satisfactoriamente en la detección de daños en estructuras a partir de sus cambios en las propiedades vibracionales. Tanto la magnitud del daño como su localización pueden determinarse simultáneamente siempre y cuando la red sea correctamente entrenada.

2.8. MÉTODO DE ALGORITMOS GENÉTICOS (AG) Y OTRAS TÉCNICAS DE OPTIMIZACIÓN

La optimización se puede definir como aquella ciencia encargada de determinar las mejores soluciones a problemas matemáticos, que a menudo modelan una realidad física. Los problemas complejos de optimización multidimensional, no lineal y altamente multimodal, pueden encontrarse en ingeniería, economía, geofísica y, prácticamente, en todos los campos de la ciencia. En este aspecto, la optimización numérica ha adquirido mucha atención entre la comunidad científica, durante las últimas décadas. Teniendo en cuenta que un aspecto esencial de la optimización heurística es poder decidir que algoritmo y que parámetros de ese algoritmo se ajustan mejor a las características del problema físico bajo análisis, estas técnicas, entre otras cosas, en el contexto de detección de daños pretenden dar a conocer herramientas para afrontar esta tarea (*Celis y Millan, 2009*).

Venter y Sobieszczanski-Sobiesk (2003) implementaron el algoritmo de optimización con enjambre de partículas (PSO) para estudiar su funcionamiento en un problema estructural de una viga en voladizo de cinco segmentos, minimizando el volumen de material de la viga para obtener un máximo esfuerzo de tensión para las limitaciones de cada segmento. Las variables de diseño fueron la altura h y el ancho b de cada segmento. En el problema se permitió que la altura de cada segmento variara entre 10 y 200 cm, mientras que la ancho se permitió variar entre 0,5 y 50 cm, en los cuales se permitieron variar continua y discretamente cada una de los parámetros del problema. Los resultados indicaron que el algoritmo PSO es capaz de tratar con problemas continuos y discretos.

Por otro lado los algoritmos genéticos (AG) son métodos de optimización de funciones basadas en la variación aleatoria y selección de una población de soluciones. Son parte de lo que puede describirse como algoritmos evolutivos, que se han desarrollado desde la década de 1950. Su ventaja sobre los algoritmos de optimización tradicionales radica en

que los AG son capaces de abordar la topología de solución multimodal que es típica de problemas de identificación de daños. Muchos autores, describen un proceso de dos etapas, donde un VFR es utilizado para localizar los daños inicialmente y luego en una segunda fase, un AG se utiliza para cuantificar los daños en los elementos e identificarlos con éxito (*Carden y Fanning, 2004*).

2.9. MÉTODOS ESTADÍSTICOS

Farrar y Doebling (1999) sugieren que la detección de daño basada en vibraciones es un problema fundamentalmente de reconocimiento de patrones estadísticos. Donde la detección tiene que ver con la identificación de las desviaciones en los datos obtenidos de la estructura dañada en relación a los datos en condiciones normales de funcionamiento o de la estructura sin daño. Características derivadas de las mediciones realizadas a una estructura en su estado intacto tendrá una distribución asociada con una media y varianza. Si la estructura está dañada, entonces puede haber un cambio en la media, la varianza, o en ambas cosas (*apud Carden y Fanning, 2004*).

Riveros (2007) realizó una implementación numérica al puente que hace parte del sistema ferroviario de Medellín que se abrió al público en 1995, con una longitud total de 28,8 km y 25 estaciones. La metodología propuesta se dividió en cuatro pasos: la colocación óptima de sensores, extracción de fuentes de excitaciones ambientales, obtención de su información modal y detección de daños estructurales. La posición óptima del sensor se realizó mediante el llamado Algoritmo de Ubicación Óptima de Sensor basado en la matriz de información Fisher. Para el segundo y tercer nivel se planteó una identificación del sistema estructural con base en excitaciones ambientales en el cual se utilizó el ERA (Algoritmo de Realización de Valores Propios) para la identificación del sistema y la obtención de la información modal y por último para la detección de daños estructurales se realizó una formulación matemática del enfoque probabilístico Bayesiano, donde por medio del teorema de Bayes se encuentra la mayor probabilidad, después de observar un conjunto de modos de estimación de parámetros, cada conjunto de parámetros modales consta de frecuencias naturales y el formas modales. Como conclusión de la implementación numérica realizada, se demostró que es posible solucionar el problema a partir de la identificación de deterioración gradual de estructuras civiles que usan el enfoque de vibración, que de verdad optimizan el empleo de evaluación visual de inspección y técnicas no destructivas (NDE).

Finalmente este trabajo se encuentra estructurado de la siguiente forma:

En el capítulo 3 se presenta las definiciones básicas tales como la definición de la matriz de rigidez y la matriz de masa de una estructura, además se describe el análisis dinámico por medio del cual se pretende obtener las características vibracionales del pórtico 2D.

En el capítulo 4 se describe el algoritmo PSO, selección de parámetros y algunas otras consideraciones. Antes de realizar la implementación en la detección de daños se realiza una aplicación a algunas funciones matemáticas encontradas en la literatura internacional con el fin de conocer el comportamiento del algoritmo frente a diversos problemas multidimensionales y multimodales.

En el capítulo 5 se realiza la implementación del algoritmo para la detección de daños en el pórtico, en esta sección se describe la metodología que se utiliza, además se presentan los resultados obtenidos.

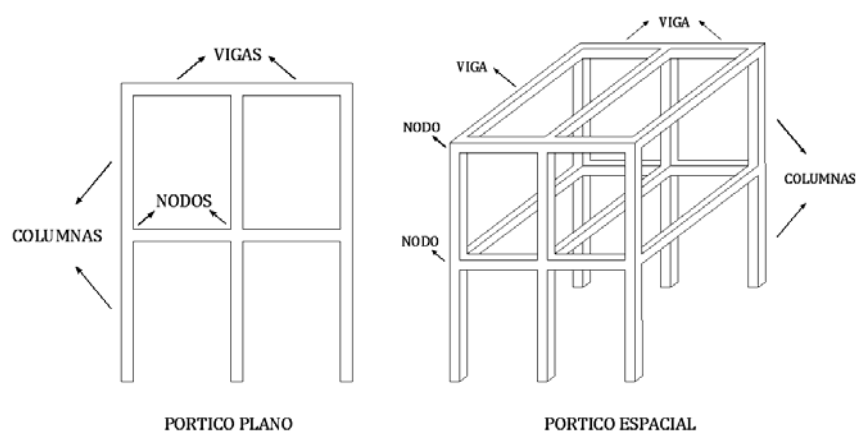
Por último en el capítulo 6 se presentan las conclusiones de este trabajo.

3. MARCO TEÓRICO

3.1. PÓRTICO

En términos generales, un pórtico se conoce como el conjunto estructural en plano o espacio constituido por vigas y columnas (ver Figura 1) unidas rígidamente mediante nodos (porción de la columna limitada por las superficies superiores e inferiores de las vigas que llegan a ella) que permiten la capacidad de transmitir momentos flectores de un elemento a otro.

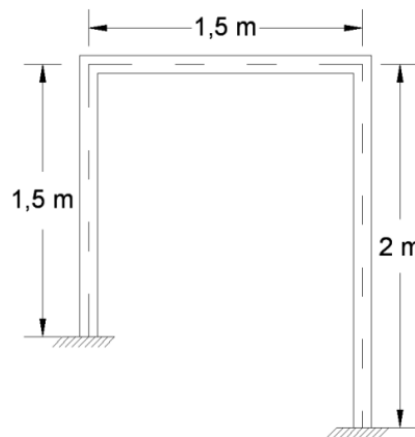
Figura 1. Pórtico



Fuente: Los autores

Para el desarrollo del presente trabajo se planteo un pórtico simple biempotrado en 2D, el cual consiste en una estructura constituida por dos columnas y una viga superior, donde sus apoyos se encuentran fijos (empotrados). En la figura 2 se pueden observar las características geométricas que fueron propuestas para el pórtico.

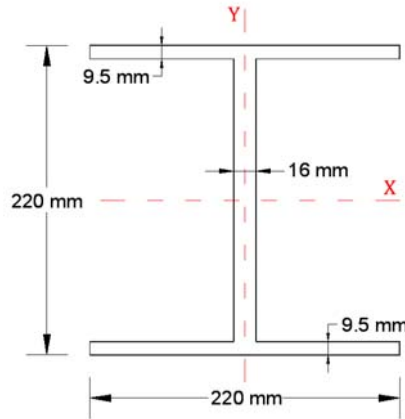
Figura 2. Pórtico propuesto para la detección de daños



Fuente: Los autores

Por otro lado, se considero un perfil de acero H.E.B. 220 para la sección transversal tanto de vigas como de columnas que componen el pórtico de la figura 2. En la figura 3 y en la tabla 1 se puede observar la geometría y las propiedades de la sección.

Figura 3. Geometría del perfil H.E.B. 220



Fuente: Tablas y equivalencias Acindar (2007)

Tabla 1. Propiedades del perfil H.E.B. 220

Área	$9.1 \times 10^{-3} \text{ m}^2$
Inercia	$8.09 \times 10^{-5} \text{ m}^4$
Densidad	7857.14 Kg/m^3
Modulo de elasticidad	$2 \times 10^{11} \text{ N/m}^2$

Fuente: Tablas y equivalencias Acindar (2007)

Dado que el presente trabajo tiene como finalidad la implementación de un algoritmo para la detección de daños teniendo en cuenta las características de vibración de la estructura (frecuencias y modos), a continuación se describirán algunos conceptos básicos de análisis y dinámica estructural mediante los cuales se pueden determinar dichas características.

3.2. MATRIZ DE RIGIDEZ

La determinación de la matriz de rigidez del sistema estructural se realizara por medio de dos métodos ampliamente usados en el campo de la ingeniería, los cuales son el método matricial y el método de elementos finitos. A continuación se explicara brevemente cada una de dichos métodos.

3.2.1. Método matricial

Este método, consiste en un conjunto de ecuaciones simultáneas que describen las características de carga-deformación de una estructura. Estas ecuaciones se resuelven

utilizando el álgebra matricial para obtener las características de carga-deformación de elementos discretos en los cuales la estructura ha sido subdividida.

El método matricial es el ideal para realizar el análisis estructural usando un ordenador. En general, existen dos enfoques para el análisis estructural usando el análisis matricial. El primero llamado método de flexibilidad donde las fuerzas son utilizadas como variables independientes y el segundo se llama el método de la rigidez, que emplea las deformaciones como las variables independientes. Los dos métodos también son conocidos como el método de las fuerzas y el método de los desplazamientos respectivamente.

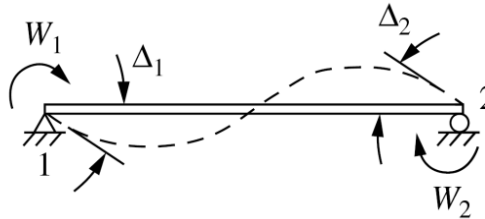
3.2.1.1. Método de la rigidez

En este método, fuerzas y deformaciones en la estructura son relacionadas entre si mediante coeficientes de influencia de rigidez. Considerando una viga simplemente apoyada sometida a los momentos W_1 y W_2 aplicados en los apoyos 1 y 2, y dejando las rotaciones respectivas denotadas como Δ_1 y Δ_2 como se muestra en la figura 4, podemos escribir las expresiones para los momentos finales W_1 y W_2 como:

$$W_1 = K_{11}\Delta_1 + K_{12}\Delta_2 \quad (3.1)$$

$$W_2 = K_{21}\Delta_1 + K_{22}\Delta_2 \quad (3.2)$$

Figura 4. Viga simplemente apoyada sometida a momentos W_1 y W_2 en sus extremos



Fuente: Richard Liew, J. Y. and Shanmugam, N.E. (2005)

Donde K_{11} y K_{12} son los coeficientes de influencia de rigidez, definidos como momentos en 1 debido a las rotaciones en 1 y 2, respectivamente. Las ecuaciones pueden escribirse en forma matricial como:

$$\begin{Bmatrix} W_1 \\ W_2 \end{Bmatrix} = \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \begin{Bmatrix} \Delta_1 \\ \Delta_2 \end{Bmatrix} \quad (3.3)$$

O

$$\{W\} = [K]\{\Delta\} \quad (3.4)$$

La matriz $[K]$ se denomina como matriz de rigidez. Puede demostrarse que la matriz de flexibilidad de una estructura es la inversa de la matriz de rigidez, y viceversa. La matriz de rigidez de toda la estructura está formada por las matrices de rigidez de los distintos elementos que componen la estructura.

Conociendo los valores de las fuerzas exteriores W que actúan sobre el sistema, se obtienen sus desplazamientos Δ por solución del sistema de ecuaciones anterior, lo cual

explica el nombre de método de los desplazamientos por el que también se conoce a este método.

3.2.1.2. Elementos de la matriz de rigidez

3.2.1.2.1. Miembros de carga axial

La figura 5(a) muestra a un miembro con área de la sección transversal constante cargado axialmente con fuerzas q_1 y q_2 y desplazamientos δ_1 y δ_2 . mostrados en sus respectivas direcciones positivas. Tomando como unidad el desplazamiento $\delta_1 = 1$ en el nodo 1, como se muestra en la figura 5(b), las fuerzas axiales en los nodos 1 y 2 son obtenidas como:

$$K_{11} = \frac{EA}{L}, \quad K_{21} = -\frac{EA}{L} \quad (3.5)$$

De la misma manera, considerando $\delta_2 = 1$ como se muestra en la Figura 5(c), las fuerzas correspondientes son obtenidas como:

$$K_{12} = -\frac{EA}{L}, \quad K_{22} = \frac{EA}{L} \quad (3.6)$$

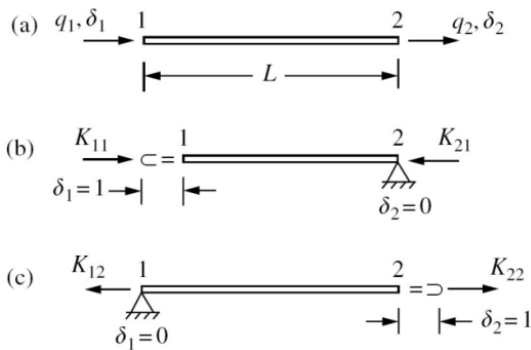
La matriz de rigidez se puede escribir como:

$$\begin{Bmatrix} q_1 \\ q_2 \end{Bmatrix} = \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \begin{Bmatrix} \delta_1 \\ \delta_2 \end{Bmatrix} \quad (3.7)$$

O

$$\begin{Bmatrix} q_1 \\ q_2 \end{Bmatrix} = \frac{EA}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{Bmatrix} \delta_1 \\ \delta_2 \end{Bmatrix} \quad (3.8)$$

Figura 5. Elemento cargado axialmente



Fuente: Richard Liew, J. Y. and Shanmugam, N.E. (2005)

3.2.1.2.2. Miembros a flexión

La matriz de rigidez para el elemento a flexión puede ser construida refiriéndose a la Figura 6, donde las fuerzas y los desplazamientos correspondientes son definidos. La ecuación de la matriz que relaciona estas fuerzas y desplazamientos puede ser escrita en la forma:

$$\begin{Bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{Bmatrix} = \begin{bmatrix} K_{11} & K_{12} & K_{13} & K_{14} \\ K_{21} & K_{22} & K_{23} & K_{24} \\ K_{31} & K_{32} & K_{33} & K_{34} \\ K_{41} & K_{42} & K_{43} & K_{44} \end{bmatrix} \begin{Bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \end{Bmatrix} \quad (3.9)$$

Los términos en la primera columna corresponde a la fuerzas generadas por los desplazamientos $\delta_1 = 1$, cuando $\delta_2 = \delta_3 = \delta_4 = 0$.

Esto significa que una unidad de desplazamiento vertical se impone en el extremo izquierdo del miembro, mientras que la rotación en ambos extremos se impide, como se muestra en la Figura 7(a). Las cuatro fuerzas de miembro correspondiente a esta deformación pueden ser obtenidas usando el teorema de castigliano, el cual enuncia que según la definición de la energía de deformación de una estructura se pueden calcular los desplazamientos y rotaciones, a partir de:

$$\Delta = \int_0^L M \frac{\partial M}{\partial q_1} \frac{dx}{EI} \quad (3.10)$$

$$\theta = \int_0^L M \frac{\partial M}{\partial q_2} \frac{dx}{EI} \quad (3.11)$$

Donde:

Δ = Desplazamiento provocado por las cargas externas.

θ = Rotación provocada por las cargas externas.

M = Momento interno de la viga o marco provocado por cargas externas.

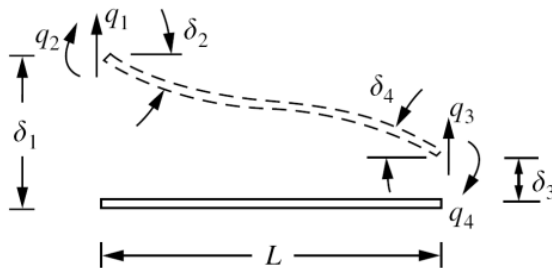
E = Modulo de elasticidad del material de la viga o marco.

I = Momento de inercia de la seccion transversal

q_1 = Fuerza externa aplicada en la viga o marco en la direccion de Δ .

q_2 = Momento externo actuante en el punto a calcular θ .

Figura 6. Elemento tipo viga sometido a flexión



Fuente: Richard Liew, J. Y. and Shanmugam, N.E. (2005)

De la figura 6, si generamos un desplazamiento en el nodo izquierdo δ_1 esto nos generara unas fuerzas en elemento q_1, q_2, q_3 y q_4 y a partir de ahí la ecuación de momento interno del elemento será:

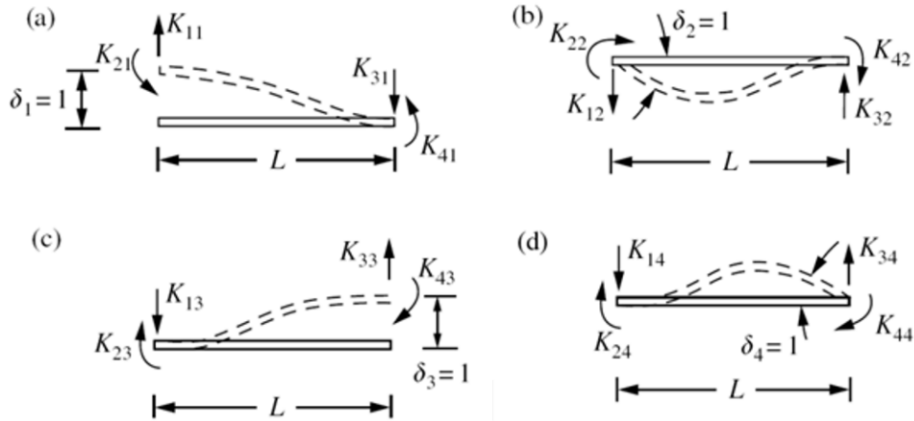
$$M = -q_1 * x + q_2 \quad (3.12)$$

Derivado parcialmente (3.12) respecto ha q_1 y q_2 tenemos:

$$\frac{\partial M}{\partial q_1} = -x \quad (3.13)$$

$$\frac{\partial M}{\partial q_2} = 1 \quad (3.14)$$

Figura 7. Coeficientes de rigidez del elemento tipo viga



Fuente: Richard Liew, J. Y. and Shanmugam, N.E. (2005)

Reemplazando (3.12) y (3.13) en (3.10), reordenando términos e integrando tenemos:

$$\Delta = \int_0^L (-q_1 * x + q_2)(-x) \frac{dx}{EI} = \frac{1}{EI} \left(\frac{q_1 * L^3}{3} - \frac{q_2 * L^2}{2} \right) \quad (3.15)$$

Reemplazando (3.12) y (3.14) en (3.11), reordenando términos e integrando tenemos:

$$\theta = \int_0^L (-q_1 * x + q_2)(1) \frac{dx}{EI} = \frac{1}{EI} \left(q_2 * L - \frac{q_1 * L^2}{2} \right) \quad (3.16)$$

Si $\Delta = 1$ y $\theta = 0$, como se muestra en la figura 7(a) entonces $q_1 = K_{11}$, $q_2 = K_{21}$, $q_3 = K_{31}$, $q_4 = K_{41}$, por lo tanto las ecuaciones (3.15) y (3.16) quedan respectivamente como:

$$\frac{K_{11} * L^3}{3} - \frac{K_{21} * L^2}{2} = EI \quad (3.17)$$

$$K_{21} * L - \frac{K_{11} * L^2}{2} = 0 \quad (3.18)$$

Despejando de (3.18) K_{21} , reemplazando en (3.17) y despejando K_{11} tenemos:

$$EI = \left(\frac{K_{11} * L^3}{3} - \frac{K_{11} * L^3}{4} \right) \Rightarrow K_{11} = \frac{12EI}{L^3} \quad (3.19)$$

Reemplazando (3.19) en (3.17) y desparjando obtenemos el valor de K_{21} como:

$$K_{21} = \frac{6EI}{L^2} \quad (3.20)$$

Aplicando las ecuaciones de equilibrio de la estática podemos determinar K_{31} y K_{41}

$$\Sigma F = 0 \Rightarrow K_{31} = \frac{12EI}{L^3} \quad (3.21)$$

$$\Sigma M = 0 \Rightarrow K_{41} = \frac{6EI}{L^2} \quad (3.22)$$

Si $\Delta = 0$ y $\theta = 1$, como se muestra en la figura 7(b) entonces $q_1 = K_{11}$, $q_2 = K_{21}$, $q_3 = K_{31}$, $q_4 = K_{41}$, por lo tanto las ecuaciones (3.15) y (3.16) quedan respectivamente como:

$$\frac{K_{12} * L^3}{3} = \frac{K_{22} * L^2}{2} \quad (3.23)$$

$$EI = \left(K_{22}L - \frac{K_{12} * L^2}{2} \right) \quad (3.24)$$

Despejando de (3.23) K_{22} , reemplazando en (3.24) y despejando K_{12} , tenemos:

$$EI = \left(\frac{2}{3}K_{12}LL - \frac{K_{12} * L^2}{2} \right) \Rightarrow K_{12} = \frac{6EI}{L^2} \quad (3.25)$$

Reemplazando (3.25) en (3.23) y despejando K_{22} , obtenemos:

$$K_{22} = \frac{2}{3} * \frac{6EI}{L^2}L \Rightarrow K_{22} = \frac{4EI}{L} \quad (3.26)$$

Aplicando las ecuaciones de equilibrio de la estática podemos determinar K_{32} y K_{42}

$$\Sigma F = 0 \Rightarrow K_{32} = \frac{6EI}{L^2} \quad (3.27)$$

$$\Sigma M = 0 \Rightarrow K_{42} = \frac{2EI}{L} \quad (3.28)$$

Usando la Figura 7(c) y la Figura 7(d), los coeficientes de influencia para las terceras y cuartas columnas pueden ser obtenidos. Los resultados de estos cálculos conducen a la matriz de rigidez de elemento como:

$$\begin{Bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{Bmatrix} = \begin{bmatrix} \frac{12EI}{L^3} & \frac{6EI}{L^2} & \frac{12EI}{L^3} & \frac{6EI}{L^2} \\ \frac{6EI}{L^2} & \frac{4EI}{L} & -\frac{6EI}{L^2} & \frac{2EI}{L} \\ -\frac{12EI}{L^3} & -\frac{6EI}{L^2} & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ \frac{6EI}{L^2} & \frac{2EI}{L} & -\frac{6EI}{L^2} & \frac{4EI}{L^2} \end{bmatrix} \begin{Bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \end{Bmatrix} \quad (3.29)$$

Si la carga axial en un elemento también es considerada, la forma general de la matriz de rigidez para el elemento será:

$$\begin{Bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \end{Bmatrix} = \begin{bmatrix} \frac{AE}{L} & 0 & 0 & -\frac{AE}{L} & 0 & 0 \\ 0 & \frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & \frac{12EI}{L^3} & \frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{4EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{2EI}{L} \\ -\frac{AE}{L} & 0 & 0 & \frac{AE}{L} & 0 & 0 \\ 0 & -\frac{12EI}{L^3} & -\frac{6EI}{L^2} & 0 & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{2EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{4EI}{L^2} \end{bmatrix} \begin{Bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \\ \delta_5 \\ \delta_6 \end{Bmatrix} \quad (3.30)$$

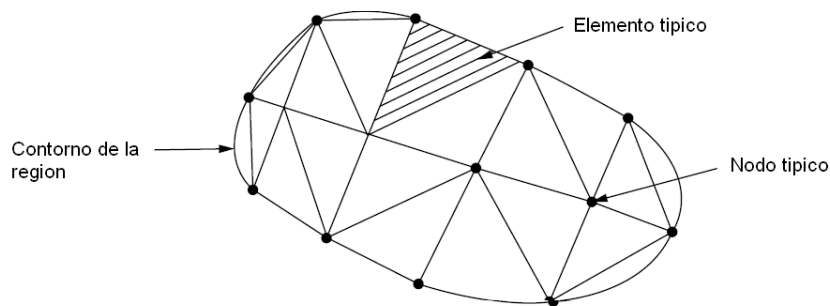
3.2.2. Método de elementos finitos

Para problemas que involucran propiedades complejas de materiales y condiciones de contorno, los métodos numéricos son empleados para suministrar soluciones aproximadas. De los muchos métodos numéricos desarrollados antes y después de la llegada de los ordenadores, el método de elementos finitos ha resultado ser una herramienta muy poderosa. Este método puede ser considerado como extensión del método matricial de análisis de estructuras. El método de elementos finitos es aplicable a una amplia gama de problemas de valor frontera de la ingeniería, y se remonta a mediados de los años 1950 con el trabajo pionero de Argyris (1960), Clough y Penzien (1963), entre otros.

3.2.2.1. Principios básicos

El método de elementos finitos está basado en la representación de un cuerpo o una estructura por ensamble de subdivisiones llamadas elementos finitos, como se muestra en la siguiente figura.

Figura 8. Ensamble de subdivisiones



Fuente: Richard Liew, J. Y. and Shanmugam, N.E. (2005)

Estos elementos son considerados unidos por nodos. Para aproximar la variación de desplazamientos sobre cada elemento finito se selecciona unas funciones de desplazamiento. Las funciones polinómicas son comúnmente empleadas para aproximar estos desplazamientos. Las ecuaciones de equilibrio para cada elemento son obtenidas por medio del principio de mínima energía potencial. Estas ecuaciones son formuladas para todo el cuerpo mediante la combinación de las ecuaciones para cada elemento individual de modo que la continuidad de desplazamientos es conservada en los nodos. Las ecuaciones resultantes son solucionadas cumpliendo con las condiciones límites para obtener los desplazamientos desconocidos.

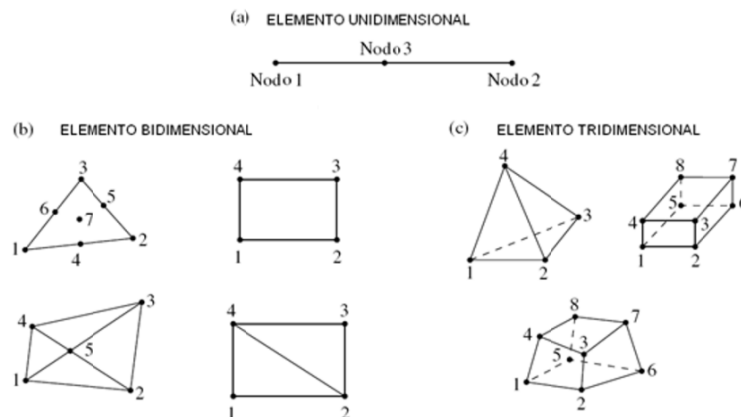
El procedimiento del método de elementos finitos implica los pasos siguientes:

- El cuerpo dado es subdividido en sistema equivalente de elementos finitos.
- Una función de desplazamiento adecuada es escogida.
- La matriz de rigidez del elemento es obtenida usando el principio de mínima energía potencial.
- La matriz de rigidez global para el cuerpo entero es formulada.
- Las ecuaciones algebraicas obtenidas son solucionadas para determinar los desplazamientos desconocidos.
- Las deformaciones y esfuerzos son calculados a partir de los desplazamientos nodales.

3.2.2.2. Elección de tamaños y formas de los elementos

Un elemento finito generalmente tiene una dos o tres dimensiones en su configuración. Los límites de los elementos son a menudo líneas rectas, y los elementos pueden ser unidimensionales, bidimensionales o tridimensionales, como se muestra en la figura 9. Al subdividir el continuo, se tiene que decidir el número, la forma, el tamaño de la configuración de los elementos de tal modo que el cuerpo original sea simulado tan estrechamente como sea posible. Los nodos deben ser ubicados en posiciones donde ocurren cambios abruptos de la geometría, carga y propiedades de los materiales.

Figura 9. Formas básicas de los elementos para discretizar el modelo



Fuente: Richard Liew, J. Y. and Shanmugam, N.E. (2005)

3.2.2.3. Elección de la función de forma de desplazamiento

La selección de la función de desplazamiento es un paso importante en el análisis de elementos finitos, ya que determina el rendimiento del análisis del elemento. Debe prestarse atención para seleccionar una función de desplazamiento que:

- Tiene el número de constantes desconocidas como el número total de grados de libertad del elemento.
- No tiene ninguna dirección definida.
- Es capaz de representar los estados constantes de esfuerzo o deformación.
- Satisface la compatibilidad de desplazamientos en los límites con elementos adyacentes.

Un polinomio es la forma más común de de la función de desplazamiento ya que estos son más fáciles de manejar en la formulación de los diferentes elementos y son más convenientes en la computación digital. El grado de aproximación se rige por el estado en el cual la función se trunca.

3.2.2.4. Grados de libertad nodales

Las deformaciones del elemento finito son especificadas completamente por el desplazamiento nodal, rotaciones y/o deformaciones, las cuales son referidas a los grados de libertad. La convergencia, la isotropía geométrica, y la función de energía potencial son los factores que determinan el número mínimo de los grados de libertad necesarios para un elemento dado.

3.2.2.5. Función de forma del elemento

El método de elementos finitos no está restringido para el uso de elementos lineales. La mayoría de elementos finitos utilizados permiten seleccionar entre elementos con funciones de interpolación lineales o cuadráticas. En el caso de funciones cuadráticas, menos elementos son necesarios para obtener el mismo grado de exactitud en los valores nodales. Las funciones de formas pueden ser desarrolladas basadas en las siguientes propiedades:

- Cada función de forma tiene un valor de 1 en su propio nodo y es cero en cada uno de los otros nodos.
- La función de forma para elementos bidimensionales es cero a lo largo de cada lado en el que el nodo no está en contacto.
- Cada función de forma es un polinomio del mismo grado que la ecuación de interpolación.

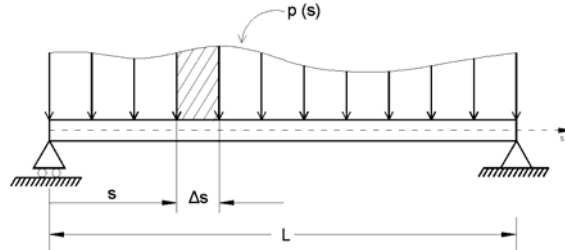
3.2.2.6. Formulación de la matriz de rigidez para una viga

3.2.2.6.1. Componentes a flexión

Las ecuaciones que rigen el comportamiento de este elemento son aquellas obtenidas a partir de la teoría de vigas en resistencia de materiales. Consideremos una viga

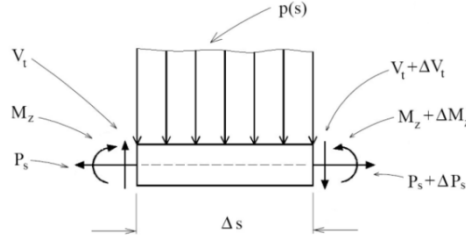
simplemente apoyada con una carga distribuida por unidad de longitud $p(s)$, de la cual extraemos un elemento infinitesimal Δs , y realizando un diagrama de cuerpo libre como se observa en la figura 10 y en la figura 11 respectivamente.

Figura 10. Viga simplemente apoyada



Fuente: Beer Ferdinand et al (2004)

Figura 11. Diagrama de cuerpo libre de una sección infinitesimal de viga



Fuente: Beer Ferdinand et al (2004)

Las ecuaciones de equilibrio del segmento de viga son:

$$V_t - (V_t + \Delta V_t) - p(s)\Delta s = 0 \quad (3.31)$$

$$(M_z + \Delta M_z) - M_z - V_t * \Delta s + p(s)\Delta s \frac{\Delta s}{2} = 0 \quad (3.32)$$

Resolviendo las ecuaciones obtenemos:

$$\Delta V_t = -p(s)\Delta s \quad (3.33)$$

$$\Delta M_z = V_t * \Delta s - p(s)\Delta s \frac{\Delta s}{2} \quad (3.34)$$

Dividiendo ambos términos de las ecuaciones por Δs , y haciendo $\Delta s \rightarrow 0$, obtenemos las ecuaciones diferenciales de equilibrio de una viga, las cuales son:

$$\frac{dV_t}{ds} = -p(s) \quad (3.35)$$

$$\frac{dM_z}{ds} = -V_t(s) \quad (3.36)$$

Combinando estas dos ecuaciones obtenemos:

$$\frac{d^2 M_z}{ds^2} = p(s) \quad (3.37)$$

El momento flector M_z resultante de la distribución de las tensiones en la sección transversal de la viga es:

$$M_z = - \int \sigma_s t \, dA \quad (3.38)$$

Donde t es el eje perpendicular al eje centroidal s de la viga y A es el área de la sección transversal de la viga. Suponiendo que el desplazamiento axial a lo largo de la viga esta aproximado por:

$$u_s = -t \frac{du_t}{ds} \quad (3.39)$$

Y a continuación sin haber deformación en el eje centroidal de la viga, es decir, los planos normales de la viga antes de la deformación permanecen normales después de la deformación. Las tensiones σ_s en la sección transversal están dadas por:

$$\sigma_s = E \varepsilon_s = E \frac{du_s}{ds} = -Et \frac{d^2 u_t}{ds^2} \quad (3.40)$$

Despreciando el efecto cortante. Considerado el material lineal. El momento flector esta dado por:

$$M_z = \left(\int_A Et^2 \, dA \right) \frac{d^2 u_t}{ds^2} \quad (3.41)$$

E introduciendo la definición del momento de inercia como:

$$I_z = \int_A t^2 \, dA \quad (3.42)$$

Tenemos la relación del momento de curvatura de una viga:

$$M_z = EI_z \frac{d^2 u_t}{ds^2} \quad (3.43)$$

Sustituyendo la relación anterior en la ecuación que relaciona el momento flector con la carga obtenemos la ecuación diferencial para vigas:

$$\frac{d^2}{ds^2} \left(EI_z \frac{d^2 u_t}{ds^2} \right) = p(s) \quad (3.44)$$

Donde u_t es el desplazamiento transversal, EI_z es la rigidez a flexión. Si EI_z es uniforme a lo largo de la viga, la anterior ecuación queda escrita como:

$$EI_z \frac{d^4 u_t}{ds^4} = p(s) \quad (3.45)$$

La energía potencial (Π_p) para un elemento de viga esta dado por:

$$\Pi_p = \frac{1}{2} \int_{\Omega} \sigma_s \varepsilon_s \, d\Omega - \int_{\Omega} \bar{p}(s) u_t \, d\Omega \quad (3.46)$$

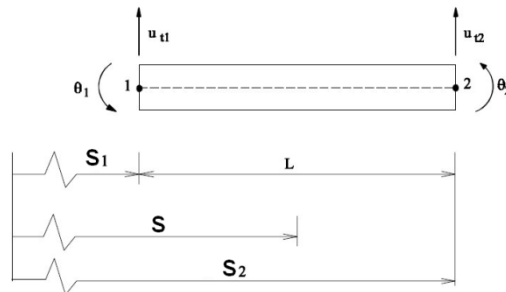
Donde Ω es el volumen del elemento de viga, $\bar{p}(s)$ es la carga distribuida por unida de área. Introduciendo las expresiones para las tensiones σ_s y deformaciones ε_s en la

expresión de la energía potencial Π_p , y teniendo en cuenta la definición de I_z en la ecuación (3.42), obtenemos:

$$\Pi_p = \frac{1}{2} \int_{s_1}^{s_2} EI_z \left(\frac{d^2 u_t}{ds^2} \right)^2 ds - \int_{s_1}^{s_2} p(s) u_t ds \quad (3.47)$$

Donde s_1 y s_2 son las coordenadas iniciales y final del elemento de viga como se muestra en la siguiente figura.

Figura 12. Grados de libertad de un elemento de viga sometido a flexión



Fuente: Los autores

El desplazamiento vertical u_t es interpolado comúnmente usando:

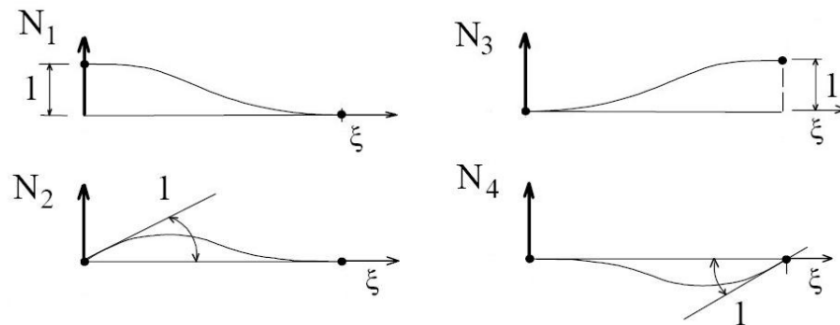
$$u_t(s) = N_1(\xi)u_{t1} + N_2(\xi)\theta_1 + N_3(\xi)u_{t2} + N_4(\xi)\theta_2 \quad (3.48)$$

Donde la coordenada está dada por $\xi = \frac{s-s_1}{L}$. Las funciones de interpolación están dadas por:

$$\begin{aligned} N_1(\xi) &= 1 - 3\xi^2 + 2\xi^3 \\ N_2(\xi) &= L\xi(1 - 2\xi + \xi^2) \\ N_3(\xi) &= \xi^2(3 - 2\xi) \\ N_4(\xi) &= L\xi^2(\xi - 1) \end{aligned} \quad (3.49)$$

Las cuales son llamadas funciones de Hermite, estas funciones son representadas gráficamente como:

Figura 13. Funciones de forma para elementos sometido a flexión



Fuente: Fonseca (2002)

Los desplazamientos $u_t(s)$ también pueden ser representados, en términos de los desplazamientos nodales $u_{t1}, \theta_1, u_{t2}, \theta_2$ en forma matricial como:

$$u_t(s) = [N]\{q\} \quad (3.50)$$

Donde

$$[N] = [N_1 \ N_2 \ N_3 \ N_4] \quad (3.51)$$

Y

$$\{q\} = \{u_{t1} \ \theta_1 \ u_{t2} \ \theta_2\}^T \quad (3.52)$$

El mismo desplazamiento $u_t(s)$ puede ser representado por un polinomio cubico

$$u_t(s) = \alpha_1 + \alpha_2 s + \alpha_3 s^2 + \alpha_4 s^3 \quad (3.53)$$

O en notación matricial

$$u_t(s) = [\Phi]\{\alpha\} \quad (3.54)$$

Donde

$$[\Phi] = [1 \ s \ s^2 \ s^3] \quad (3.55)$$

Y

$$\{\alpha\} = \{\alpha_1 \ \alpha_2 \ \alpha_3 \ \alpha_4\}^T \quad (3.56)$$

La rotación $\theta(s) = \frac{du_t}{ds}$ es dada por:

$$\theta(s) = [\Phi']\{\alpha\} \quad (3.57)$$

Donde

$$[\Phi'] = [0 \ 1 \ 2s \ 3s^2] \quad (3.58)$$

Para representar los desplazamientos $u_t(s)$ en términos de los desplazamientos nodales $u_{t1}, \theta_1, u_{t2}, \theta_2$, se evalúa la expresión para $u_t(s)$ es decir:

$$\{q\} = \begin{Bmatrix} u_{t1}(s=0) \\ \theta_1(s=0) \\ u_{t2}(s=L) \\ \theta_2(s=L) \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & L & L^2 & L^3 \\ 0 & 1 & 2L & 3L^2 \end{bmatrix} \begin{Bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{Bmatrix} \quad (3.59)$$

O

$$\{q\} = [A]\{\alpha\} \quad (3.60)$$

Donde podemos escribir

$$\{\alpha\} = [A]^{-1}\{q\} \quad (3.61)$$

Que sustituido en la expresión para $u_t(s)$ nos da como resultado:

$$u_t(s) = [\Phi][A]^{-1}\{q\} \quad (3.62)$$

Donde:

$$[A]^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -3/L^2 & -2/L & 3/L^2 & -1/L \\ 2/L^3 & 1/L^2 & -2/L^3 & 1/L^2 \end{bmatrix} \quad (3.63)$$

Y la multiplicación de matrices $[\Phi][A]^{-1}$ produce la matriz $[N]$ de la expresión:

$$u_t(s) = [N]\{q\} \quad (3.64)$$

Para una expresión completa de la energía potencial, es necesario es necesario expresar la segunda derivada u_t en relación con los desplazamientos nodales, es decir:

$$u_t''(s) = \frac{d^2 u_t}{ds^2} = \frac{1}{L^2} \frac{d^2 u_t}{d\xi^2} = \frac{1}{L^2} \frac{d^2 u_t}{d\xi^2} [N]\{q\} \quad (3.65)$$

Y

$$\frac{d^2 u_t}{ds^2} = [B]\{q\} \quad (3.66)$$

La matriz $[B]$ se obtiene mediante la correcta diferenciación de $u_t(s)$, la cual se muestra a continuación como:

$$\frac{du_t}{ds} = \frac{1}{L} \frac{d}{d\xi} (N_1(\xi)u_{t1} + N_2(\xi)\theta_1 + N_3(\xi)u_{t2} + N_4(\xi)\theta_2) \quad (3.67)$$

$$\frac{du_t}{ds} = \frac{1}{L} \begin{bmatrix} (-6\xi + 6\xi^2) & L(1 - 4\xi + 3\xi^2) & (6\xi - 6\xi^2) & L(3\xi^2 - 2\xi) \end{bmatrix} \begin{Bmatrix} u_{t1} \\ \theta_1 \\ u_{t2} \\ \theta_2 \end{Bmatrix} \quad (3.68)$$

Y

$$\frac{d^2 u_t}{ds^2} = \frac{1}{L^2} \begin{bmatrix} (-6 + 12\xi) & (-4L + 6L\xi) & (6 - 12\xi) & (6L\xi - 2L) \end{bmatrix} \begin{Bmatrix} u_{t1} \\ \theta_1 \\ u_{t2} \\ \theta_2 \end{Bmatrix} \quad (3.69)$$

Recordando la expresión de energía potencial

$$\Pi_p = \frac{1}{2} \int_{s_1}^{s_2} EI_z \left(\frac{d^2 u_t}{ds^2} \right)^2 ds - \int_{s_1}^{s_2} p(s)u_t ds \quad (3.70)$$

Cambiando las condiciones para $d^2 u_t/ds^2$ con la sustitución $ds = Ld\xi$ obtenemos

$$\Pi_p = \frac{1}{2} EI_z L \int_0^1 \{q\}^T [B]^T [B] \{q\} d\xi - L \int_0^1 [N] \{q\} p d\xi \quad (3.71)$$

Donde EI_z fue considerado uniforme a lo largo del elemento. Realizando la diferenciación de Π_p en relación con los desplazamientos nodales e igualando a cero, es decir:

$$\frac{\partial \Pi_p}{\partial u_{t1}} = 0$$

$$\frac{\partial \Pi_p}{\partial \theta_1} = 0$$

$$\frac{\partial \Pi_p}{\partial u_{t2}} = 0$$

$$\frac{\partial \Pi_p}{\partial \theta_2} = 0$$

Obtenemos cuatro ecuaciones, las cuales expresadas en forma matricial proporcionan:

$$EI_z L \int_0^1 [B]^T [B] d\xi \{q\} = L \int_0^1 [N]^T p d\xi \quad (3.72)$$

O

$$[K] \{q\} = \{Q\} \quad (3.73)$$

Que es la ecuación del elemento finito de viga. La matriz de rigidez $[K]$ esta dada por:

$$[K] = EI_z L \int_0^1 [B]^T [B] d\xi \quad (3.74)$$

Insertando la matriz $[B]$ en la expresión de la energía potencial obtenemos:

$$[K] = EI_z L \int_0^1 \begin{bmatrix} \frac{6}{L^2}(2\xi - 1) \\ \frac{2}{L}(3\xi - 2) \\ -\frac{6}{L^2}(2\xi - 1) \\ \frac{2}{L}(3\xi - 1) \end{bmatrix} \begin{bmatrix} \frac{6}{L^2}(2\xi - 1) & \frac{2}{L}(3\xi - 2) & -\frac{6}{L^2}(2\xi - 1) & \frac{2}{L}(3\xi - 1) \end{bmatrix} d\xi \quad (3.75)$$

Después de efectuar la integración, la matriz de rigidez se presenta de la siguiente forma:

$$[K] = \frac{EI_z}{L^3} \begin{bmatrix} 12 & 6L & -12 & 6L \\ 6L & 4L^2 & -6L & 2L^2 \\ -12 & -6L & 12 & -6L \\ 6L & 2L^2 & -6L & 4L^2 \end{bmatrix} \quad (3.76)$$

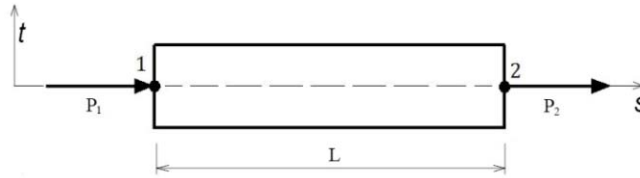
3.2.2.6.2. Componentes axiales

El estado de esfuerzos de un elemento sometido solamente a carga axial, considerando la sección transversal constante es:

$$\sigma_{ss} \neq 0 \quad (3.77)$$

$$\sigma_{zz} = \sigma_{tt} = \sigma_{ts} = \sigma_{tz} = \sigma_{sz} = 0 \quad (3.78)$$

Figura 14. Grados de libertad de un elemento de viga sometido a carga axial



Fuente: Fonseca (2002)

Como el desplazamiento u_s a lo largo de la viga varía linealmente, podemos sustituir las funciones de desplazamiento para aproximar la siguiente función:

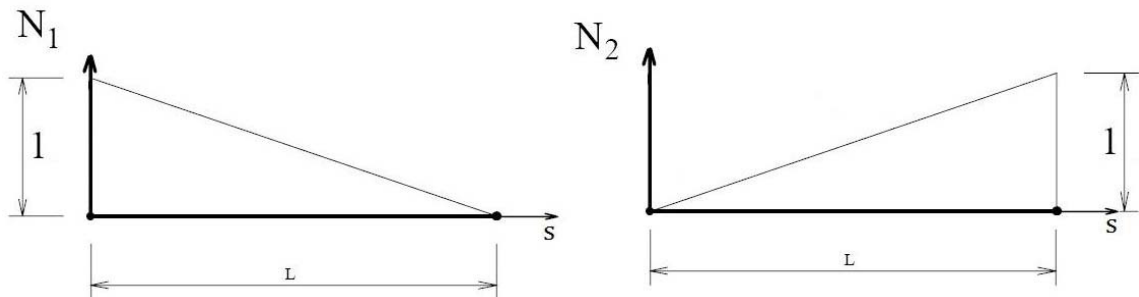
$$u_s(s) = \bar{u}_1 N_1 + \bar{u}_2 N_2 \quad (3.79)$$

Para elementos axiales podemos tomar las siguientes funciones de forma:

$$\begin{aligned} N_1(s) &= 1 - \frac{s}{L} \\ N_2(s) &= \frac{s}{L} \end{aligned} \quad (3.80)$$

Estas funciones son representadas gráficamente como:

Figura 15. Funciones de forma para elementos sometidos a carga axial



Fuente: Fonseca (2002)

Sustituyendo las ecuaciones de las funciones de forma en la ecuación (3.80) tenemos:

$$u_s(s) = \bar{u}_1 \left(1 - \frac{s}{L}\right) + \bar{u}_2 \left(\frac{s}{L}\right) \quad (3.81)$$

Para un estado uniaxial de tensiones, la deformación ϵ_{ss} es dada por:

$$\epsilon_{ss} = \frac{du_s(s)}{ds} \quad (3.82)$$

Por lo tanto:

$$\epsilon_{ss} = \bar{u}_1 \frac{dN_1(s)}{ds} + \bar{u}_2 \frac{dN_2(s)}{ds} \quad (3.83)$$

Como resultado obtenemos:

$$\epsilon_{ss} = -\frac{\bar{u}_1}{L} + \frac{\bar{u}_2}{L} = \frac{1}{L}(\bar{u}_2 - \bar{u}_1) \quad (3.84)$$

Definiendo la energía potencial como:

$$\Pi_p = U + W \quad (3.85)$$

Dónde U es la energía de deformación y W es el trabajo externo. La expresión general para U es:

$$U = \int_V \left[\frac{1}{2E} (\sigma_{ss}^2 + \sigma_{tt}^2 + \sigma_{zz}^2) + \frac{\nu}{E} (\sigma_{ss}\sigma_{tt} + \sigma_{ss}\sigma_{zz} + \sigma_{tt}\sigma_{zz}) + \frac{\nu}{2G} (\sigma_{st}^2 + \sigma_{sz}^2 + \sigma_{tz}^2) \right] dV \quad (3.86)$$

Teniendo en cuenta el estado de tensiones, ecuaciones (3.77) y (3.78), la expresión queda reducida a:

$$U = \frac{1}{2E} \int_V \sigma_{ss}^2 dV \quad (3.87)$$

Sustituyendo la ley de Hooke ($\sigma_{ss} = \epsilon_{ss}E$), además teniendo en cuenta la ecuación (3.84), obtenemos:

$$U = \frac{1}{2E} \int_L \frac{E^2}{L^2} (\bar{u}_2 - \bar{u}_1)^2 A ds \quad (3.88)$$

Esto porque σ_{ss} es constante en el área, de esta forma $dV = A ds$. Como \bar{u}_2 y \bar{u}_1 son independientes de x , podemos escribir la ecuación (3.88) en forma matricial como:

$$U = \frac{1}{2} \{ \bar{u}_1 \quad \bar{u}_2 \} \int_0^L \frac{EA}{L^2} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} ds \begin{Bmatrix} \bar{u}_1 \\ \bar{u}_2 \end{Bmatrix} \quad (3.89)$$

Integrando obtenemos:

$$U = \frac{1}{2} \{ \bar{u}_1 \quad \bar{u}_2 \} \begin{bmatrix} \frac{EA}{L} & -\frac{EA}{L} \\ -\frac{EA}{L} & \frac{EA}{L} \end{bmatrix} \begin{Bmatrix} \bar{u}_1 \\ \bar{u}_2 \end{Bmatrix} \quad (3.90)$$

Como el elemento solo estará sometido a fuerzas discretas, la expresión de trabajo es:

$$W = - \sum_{i=1}^N P_i u_{si}$$

$$W = -P_1 \bar{u}_1 - P_2 \bar{u}_2 = -\{ \bar{u}_1 \quad \bar{u}_2 \} \begin{Bmatrix} P_1 \\ P_2 \end{Bmatrix} \quad (3.91)$$

De esta forma, la energía potencial total del elemento es:

$$\Pi_p = \frac{1}{2} \{ \bar{u}_1 \quad \bar{u}_2 \} \frac{EA}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{Bmatrix} \bar{u}_1 \\ \bar{u}_2 \end{Bmatrix} - \{ \bar{u}_1 \quad \bar{u}_2 \} \begin{Bmatrix} P_1 \\ P_2 \end{Bmatrix} \quad (3.92)$$

Realizando la diferenciación de Π_p en relación con los desplazamientos nodales e igualando a cero, es decir:

$$\frac{\partial \Pi_p}{\partial u_{s1}} = 0$$

$$\frac{\partial \Pi_p}{\partial u_{s2}} = 0$$

Obtenemos:

$$\frac{EA}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{Bmatrix} \bar{u}_1 \\ \bar{u}_2 \end{Bmatrix} = \begin{Bmatrix} P_1 \\ P_2 \end{Bmatrix} \quad (3.93)$$

Siendo:

$$[K] = \frac{EA}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad (3.94)$$

Por lo tanto la matriz de rigidez de una viga considerando las fuerzas axiales será:

$$\begin{bmatrix} \frac{AE}{L} & 0 & 0 & -\frac{AE}{L} & 0 & 0 \\ 0 & \frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & \frac{12EI}{L^3} & \frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{4EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{2EI}{L} \\ -\frac{AE}{L} & 0 & 0 & \frac{AE}{L} & 0 & 0 \\ 0 & -\frac{12EI}{L^3} & -\frac{6EI}{L^2} & 0 & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{2EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{4EI}{L^2} \end{bmatrix} \quad (3.95)$$

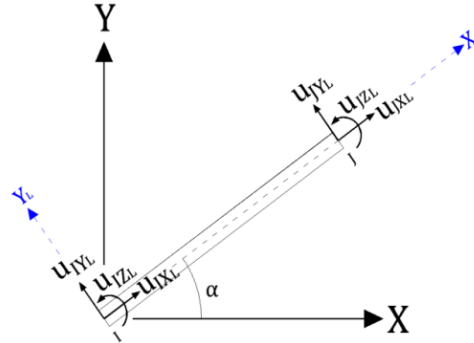
3.2.3. Rotación de ejes

Hasta el momento se ha expresado la matriz de rigidez de un elemento de un pórtico plano, según un sistema de ejes locales, estando los desplazamientos y fuerzas del extremo de la barra referida a los mismos. No obstante, es necesario transformar las características locales a unas características globales. La transformación de unas a otras se basa en la distinta orientación que tienen las características del elemento respecto a la estructura global. Esto nos obliga a la definición de un sistema de coordenadas globales ($X - Y$) que nos permiten definir de forma única las fuerzas y movimientos en los nodos de la estructura que se considera conformada como un conjunto de elementos unidos, y un sistema de coordenadas locales ($X_L - Y_L$), el cual definimos las relaciones fuerza-desplazamiento independiente de la orientación del elemento de la estructura. El paso de uno a otro se realizara mediante la correspondiente rotación de ejes.

Considerando un elemento de viga ubicada en el plano XY, y para su estudio se adopta un sistema de ejes locales con sus ejes X_L y Y_L coincidiendo con la orientación de la viga, y su origen en el nudo I del elemento. Los ejes Z general y local son coincidentes.

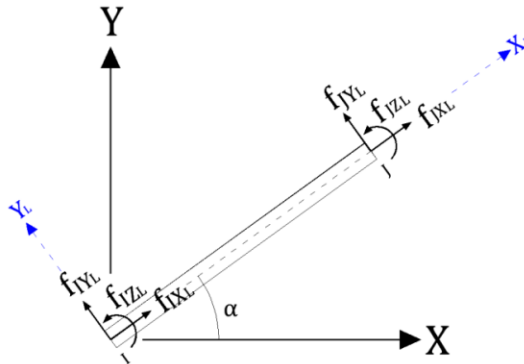
En cada nudo consideramos los tres grados de libertad que corresponden a: desplazamiento axial según X_L , desplazamiento transversal según Y_L , y giro según el eje Z. Los esfuerzos correspondientes son: fuerza axial según X_L , fuerza cortante según Y_L y momento flector según Z, como se indica en la figura 16 y la figura 17.

Figura 16. Desplazamientos de los grados de libertad referidos al sistema de referencia local



Fuente: Los autores

Figura 17. Fuerzas de los grados de libertad referidos al sistema de referencia local



Fuente: Los autores

Del método directo de rigidez, la relación entre las fuerzas y los desplazamientos en los extremos de cada barra es de la forma:

$$f_L = K_L u_L \quad (3.96)$$

Donde

f_L = Fuerzas en coordenadas locales

K_L = Matriz de rigidez de la barra en coordenadas locales

u_L = Desplazamientos en coordenadas locales

La no existencia de términos de rigidez entre los grados de libertad de deformación axial y los de flexión muestra el desacoplamiento entre ambos efectos. La ecuación de equilibrio anterior se puede poner separada para las magnitudes de cada nudo como:

$$\begin{Bmatrix} f_{LI} \\ f_{LJ} \end{Bmatrix} = \begin{bmatrix} K_{LII} & K_{LIJ} \\ K_{LJI} & K_{LJJ} \end{bmatrix} \begin{Bmatrix} u_{LI} \\ u_{LJ} \end{Bmatrix} \quad (3.97)$$

De la figura 14 las componentes de las deformaciones del nodo I referido al sistema X e Y serán:

$$u_{IXL} = U_{IX} * \cos \alpha \quad u_{IYL} = U_{IY} * \sin \alpha \quad u_{IZL} = U_{IZ} \quad (3.98)$$

Donde:

u_{IXL} = Deformación axial en coordenadas locales en el nodo I

u_{IYL} = Deformación transversal en coordenadas locales en el nodo I

u_{IZL} = Rotación en coordenadas locales en el nodo I

U_{IX} = Deformación axial en coordenadas globales en el nodo I

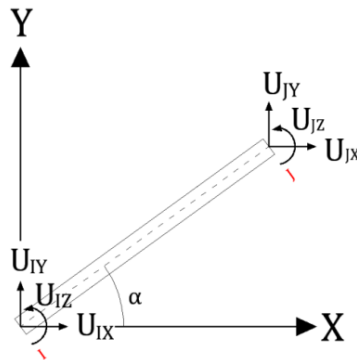
U_{IY} = Deformación transversal en coordenadas globales en el nodo I

U_{IZ} = Rotación en coordenadas globales en el nodo I

α = Angulo de desfase entre el eje global y el eje local

Los desplazamientos referidos al sistema global serán como se muestra en la siguiente figura.

Figura 18. Desplazamientos de los grados de libertad referidos al sistema de referencia global



Fuente: Los autores

Teniendo en cuenta la ecuación (3.98) y escribiendo en forma matricial obtenemos:

$$\begin{Bmatrix} u_{XL} \\ u_{YL} \\ u_{ZL} \end{Bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} U_X \\ U_Y \\ U_Z \end{Bmatrix} = TU \quad (3.99)$$

Esta es la misma matriz que se emplea para la transformación de un vector en dos dimensiones, pero añadiéndole una fila y columna más, a fin de transformar el giro, que es el mismo en los sistemas de ejes local y general. La misma matriz se emplea para la transformación de esfuerzos. En general la relación entre las magnitudes en los sistemas local y general es:

$$\begin{aligned} u_{iL} &= TU_i & u_{jL} &= TU_j \\ f_{iL} &= TF_i & f_{jL} &= TF_j \end{aligned} \quad (3.100)$$

Agrupando las propiedades de los dos nudos en un único vector obtenemos:

$$\begin{Bmatrix} u_{iL} \\ u_{jL} \end{Bmatrix} = \begin{bmatrix} T & 0 \\ 0 & T \end{bmatrix} \begin{Bmatrix} U_i \\ U_j \end{Bmatrix} = T_2 U \quad \begin{Bmatrix} f_{iL} \\ f_{jL} \end{Bmatrix} = \begin{bmatrix} T & 0 \\ 0 & T \end{bmatrix} \begin{Bmatrix} F_i \\ F_j \end{Bmatrix} = T_2 F \quad (3.101)$$

Por lo tanto la expresión de la rotación para una viga de 6 grados de libertad será:

$$T_2 = \begin{bmatrix} \cos \theta & \sin \theta & 0 & 0 & 0 & 0 \\ -\sin \theta & \cos \theta & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos \theta & \sin \theta & 0 \\ 0 & 0 & 0 & -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.102)$$

De forma compacta abreviada $u_L = T_2 U$, y análogamente $f_L = T_2 F$. Sustituyendo estas expresiones en (3.96) resulta:

$$T_2 F = K_L T_2 U \quad (3.103)$$

Multiplicando esta expresión por T_2^T , y considerando que $T_2^T = T_2^{-1}$ (por ser 1 el determinante de la matriz R), tenemos que:

$$\begin{aligned} T_2^T T_2 F &= T_2^T K_L T_2 U \\ F &= T_2^T K_L T_2 U \end{aligned} \quad (3.104)$$

La relación entre fuerzas y desplazamientos en los extremos de la barra, en coordenadas globales es de la forma $F = K U$, y realizando la comparación con la expresión obtenida, resulta finalmente:

$$K_G = T_2^T K_L T_2 \quad (3.105)$$

Donde K_G es la matriz de rigidez del elemento considerado, expresada en coordenadas globales.

3.2.4. Matriz de rigidez de la estructura

Para poder realizar el ensamble, es decir, componer la estructura a partir de sus barras. Es necesario que todas las fuerzas y todos los desplazamientos estén referidos a un mismo sistema de ejes. Para ello una vez calculada la matriz de rigidez de cada barra en coordenadas locales, hemos de realizar una rotación para referir dicha matriz a coordenadas globales.

La matriz de rigidez de la estructura K es una matriz cuadrada, de tamaño igual al número total de grados de libertad de la estructura. Si ésta tiene N nodos y cada uno de ellos tiene C grados de libertad, el tamaño de K es $N \times C$ (C es 2 para las celosías planas, 3 para celosías espaciales, pórticos planos y emparrillados, y 6 para pórticos espaciales).

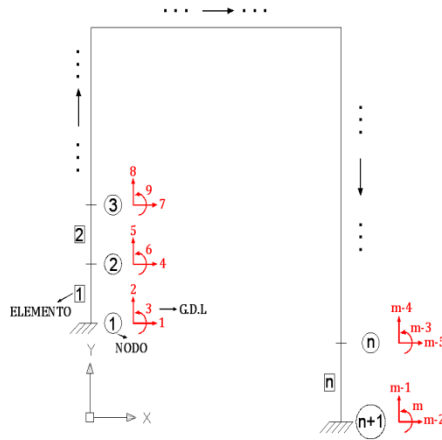
La matriz K se obtiene por ensamblaje de las matrices de rigidez de los distintos elementos que forman la estructura, en los grados de libertad correspondientes a aquellos nodos a los que se une cada elemento, es decir, cuando varios elementos se conectan en un nodo ellos tendrán la misma posición en la matriz K , por lo tanto esos coeficientes de influencia de rigidez de miembro deben sumarse algebraicamente entre sí, para determinar el coeficiente de rigidez para ese nodo de la estructura. La matriz de rigidez K es dispersa y además puede tener una estructura de banda. Esto se debe a que cada elemento solamente aporta rigidez a los grados de libertad de aquellos nodos a los que se une; por lo tanto si un nudo I no está relacionado directamente con otro nudo J , en los

términos de acople entre sus grados de libertad no se añade ninguna rigidez. Esto hace que la matriz K tenga muchos ceros, es decir sea de estructura dispersa.

Por otra parte si la numeración de los nudos es adecuada ocurre que los términos no nulos de la matriz se agrupan alrededor de la diagonal, en lo que se llama estructura de banda. Naturalmente el aspecto que adopta la matriz K depende de la numeración que se asigne a los nudos, y de cómo están relacionados éstos a través de los elementos. Por lo tanto es importante elegir una numeración correcta, pues permite generar muchos más ceros en K , facilitando su manejo.

Para poder determinar la matriz de rigidez del pórtico planteado para el desarrollo del presente trabajo, este se dividió en elementos como se muestra en la figura 18, donde se puede observar la numeración de los elementos, nudos y grados de libertad (G.D.L.) asociados a cada nodo.

Figura 19. Numeración de nudos y elementos del pórtico



Fuente: Los autores

Escribiendo las matrices de rigidez de los elementos 1, 2, ..., n-1 y n como:

$$\begin{aligned}
 K^1 &= \begin{bmatrix} K_{1,1}^1 & K_{1,2}^1 \\ K_{2,1}^1 & K_{2,2}^1 \end{bmatrix} \\
 K^2 &= \begin{bmatrix} K_{2,2}^2 & K_{2,3}^2 \\ K_{3,2}^2 & K_{3,3}^2 \end{bmatrix} \\
 K^{n-1} &= \begin{bmatrix} K_{n-2,n-2}^{n-1} & K_{n-2,n-1}^{n-1} \\ K_{n-1,n-2}^{n-1} & K_{n-1,n-1}^{n-1} \end{bmatrix} \\
 K^n &= \begin{bmatrix} K_{n-1,n-1}^n & K_{n-1,n}^n \\ K_{n,n-1}^n & K_{n,n}^n \end{bmatrix}
 \end{aligned} \tag{3.106}$$

Tomando los elementos 1 y 2 representados en la figura 17, se puede observar que el nodo 2 es común a ambos elementos, donde el desplazamiento asociado a los G.D.L 4,5 y 6 es único. Para la barra 1 de matriz de rigidez K^1 dichos desplazamientos vienen definidos por la submatriz de filas 4,5 y 6 y columnas 4,5 y 6, es decir $K_{2,2}^1$ análogamente

para la barra 2 la submatriz será $K_{2,2}^2$, Ahora para expresar los desplazamientos del nodo 2 en relación con la estructura bastara con sumar las submatrices $K_{2,2}^1$ y $K_{2,2}^2$, por lo tanto realizando el mismo razonamiento para los demás nodos, la matriz de rigidez de la estructura quedara de la siguiente manera:

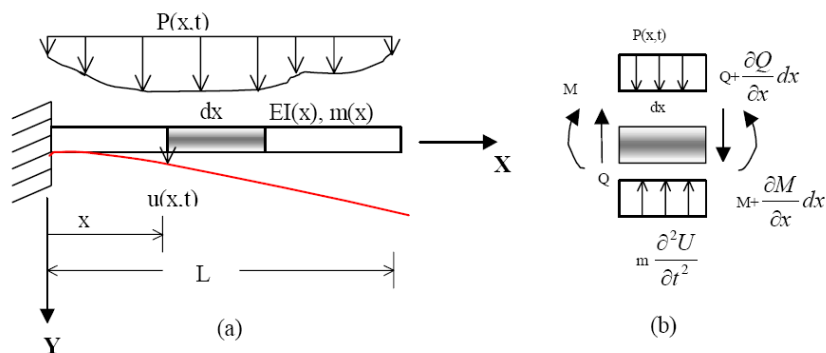
$$\begin{bmatrix}
 \begin{bmatrix} K_{1,1}^1 & K_{1,2}^1 \end{bmatrix} \\
 \begin{bmatrix} K_{2,1}^1 & \begin{bmatrix} K_{2,2}^1 + K_{2,2}^2 & K_{2,3}^2 \end{bmatrix} \\
 & \begin{bmatrix} K_{3,2}^2 & \begin{bmatrix} K_{3,3}^2 + K_{3,3}^3 \end{bmatrix} \end{bmatrix} \\
 \vdots \\
 \mathbf{0} & \dots & \begin{bmatrix} K_{n-1,n-2}^{n-1} & \begin{bmatrix} K_{n-1,n-1}^{n-2} + K_{n-1,n-1}^{n-1} & K_{n-2,n-1}^{n-1} \end{bmatrix} \\
 & & \begin{bmatrix} K_{n-1,n-1}^{n-1} + K_{n-1,n-1}^n & K_{n-1,n}^n \end{bmatrix} \\
 & & K_{n,n-1}^n & K_{n,n}^n
 \end{bmatrix}
 \end{bmatrix} \quad 3.107$$

3.3. MATRIZ DE MASA

Teniendo en cuenta la viga de la Figura 20, la ecuación de equilibrio para la viga con inercia I , módulo de elasticidad E , masa m y constante de amortiguamiento c , es:

$$EIU^{IV} + cEI\dot{U}^{IV} + m\ddot{U} = P(x, t) \quad (3.108)$$

Figura 20. Viga en vibración.



Fuente: Begambre (2004)

Multiplicando a ecuación (3.108) por una función arbitraria $\Psi(x)$, llamada función de peso y utilizando $\vartheta(x, t)$ como una función para aproximar el campo de desplazamientos $U(x, t)$ e integración en el dominio del problema, tenemos:

$$\int_0^L \Psi [EI\vartheta^{IV} + cEI\dot{\vartheta}^{IV} + m\ddot{\vartheta} - P(x,t)] dx = 0 \quad (3.109)$$

$$\underbrace{\int_0^L \Psi EI\vartheta^{IV} dx}_1 + \underbrace{\int_0^L \Psi cEI\dot{\vartheta}^{IV} dx}_2 + \int_0^L \Psi m\ddot{\vartheta} dx - \int_0^L \Psi P(x,t) dx = 0$$

Donde L es la longitud del elemento de viga de la Figura 19. Integrando dos veces por partes los términos 1 y 2 de la ecuación 3.109, tenemos:

$$\underbrace{\int_0^L \Psi EI\vartheta^{IV} dx}_1 = [\Psi EI\vartheta^{III} - \Psi^I EI\vartheta^{II}]_0^L + EI \int_0^L \Psi^{II}\vartheta^{II} dx \quad (3.110)$$

$$\underbrace{\int_0^L \Psi cEI\dot{\vartheta}^{IV} dx}_2 = cEI \left[[\Psi\dot{\vartheta}^{III} - \Psi^I\dot{\vartheta}^{II}]_0^L + EI \int_0^L \Psi^{II}\dot{\vartheta}^{II} dx \right] \quad (3.111)$$

Por otro lado, podemos determinar qué:

$$\begin{aligned} M &= -EI(\vartheta^{II} + c\dot{\vartheta}^{II}) \\ Q &= -EI(\vartheta^{III} + c\dot{\vartheta}^{III}) \end{aligned} \quad (3.112)$$

Realizando la sustitución de la ecuación (3.110), (3.111) y (3.112) en la ecuación (3.109), y agrupando términos, tenemos:

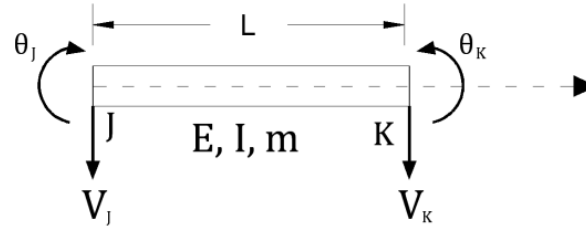
$$\int_0^L EI\Psi^{II}\vartheta^{II} dx + \int_0^L cEI\Psi^{II}\dot{\vartheta}^{II} dx + \int_0^L m\Psi\ddot{\vartheta} dx = \int_0^L \Psi P(x,t) dx + [-\Psi(-Q) + \Psi^I(-M)]_0^L \quad (3.113)$$

Como en la ecuación. (3.113) los niveles de derivación en el espacio de variables se estandarizaron entre la solución aproximada ϑ y la función de peso Ψ , la ecuación (3,113) es conocida como una forma débil del problema. Además, establece que las exigencias sean menos estrictas para las condiciones de continuidad de la función ϑ (función desconocida), además de incorporar las condiciones de contorno del problema (último término del lado derecho de la ecuación. (3,113)). De esta forma, la solución aproximada de ϑ sólo debe satisfacer las condiciones esenciales del problema. Tomando como base para aproximar el campo de desplazamiento de la viga, la función del siguiente tipo:

$$\vartheta(x,t) = [A + BX + Cx^2 + Dx^3] \quad (3.114)$$

Los coeficientes A, B, C y D pueden ser determinados por el desplazamiento de los extremos del elemento i , como se muestra en la Figura 21, y teniendo en cuenta la ecuación (3,114), es decir:

Figura 21. Elemento finito de una viga en vibración



Fuente: Begambre (2004)

$$\begin{aligned}
 \vartheta(0, t) &= V_j(t) = A \\
 \vartheta^I(0, t) &= \theta_j(t) = B \\
 \vartheta(L, t) &= V_k(t) = A + BL + CL^2 + DL^3 \\
 \vartheta^I(L, t) &= \theta_k(t) = B + 2CL + 3DL^3
 \end{aligned} \tag{3.115}$$

Resolviendo el sistema anterior para A, B, C y D, se tiene:

$$\begin{aligned}
 A &= V_j(t) \\
 B &= \theta_j(t) \\
 C &= \frac{3}{L^2} (V_k(t) - V_j(t)) - \frac{1}{L} (\theta_k(t) + \theta_j(t)) \\
 D &= -\frac{2}{L^3} (V_k(t) - V_j(t)) - \frac{1}{L^2} (\theta_k(t) + \theta_j(t))
 \end{aligned} \tag{3.116}$$

Realizando la sustitución de los valores de A, B, C y D en la ecuación (3,114), agrupando términos, tenemos:

$$\vartheta(x, t) = \left(1 - \frac{3x^2}{L^2} + \frac{2x^3}{L^3}\right) V_j(t) + \left(x - \frac{2x^2}{L} + \frac{x^3}{L^2}\right) \theta_j(t) + \left(-\frac{3x^2}{L^2} + \frac{2x^3}{L^3}\right) V_k(t) + \left(-\frac{x^2}{L} + \frac{x^3}{L^2}\right) \theta_k(t) \tag{3.117}$$

La ecuación (3.117) es la representación de la forma de vibración de la viga como una superposición de cuatro funciones polinómicas, multiplicadas por una coordenada que varía en el tiempo $V_i(t)$. Estas coordenadas actúan como incógnitas en el sistema y sus valores son determinados por la solución de la ecuación de movimiento (ecuación (3.113)). El número de coordenadas $V_i(t)$ es igual al número de grados de libertad considerados en el sistema.

Definiendo las funciones de forma como:

$$\begin{aligned}
 \xi_1 &= \left(1 - \frac{3x^2}{L^2} + \frac{2x^3}{L^3}\right) \\
 \xi_2 &= \left(x - \frac{2x^2}{L} + \frac{x^3}{L^2}\right) \\
 \xi_3 &= \left(-\frac{3x^2}{L^2} + \frac{2x^3}{L^3}\right) \\
 \xi_4 &= \left(-\frac{x^2}{L} + \frac{x^3}{L^2}\right)
 \end{aligned} \tag{3.118}$$

Que son conocidas como las funciones de forma de Hermite vistas anteriormente, que garantiza la continuidad de los desplazamientos y rotaciones de elementos que conforman un mismo nodo. Teniendo en cuenta la ecuación (3.118), la ecuación (3.114) puede escribirse como:

$$\vartheta(x, t) = \{\xi_1 \quad \xi_2 \quad \xi_3 \quad \xi_4\} \begin{Bmatrix} V_j \\ \theta_j \\ V_k \\ \theta_k \end{Bmatrix} \quad (3.119)$$

Utilizando las funciones $\Psi_i = \xi_i$ (con $i = 1, 2, 3, 4$) como la función de peso y la función expresada en la ecuación (3.119) como función para aproximar el campo de los desplazamientos de la viga, y siendo $\xi_1(x, t) = \xi_2^l(0) = \xi_3(L) = \xi_4^l = 1$, se puede escribir con base en la relación dada en la ecuación (3.113), la siguiente ecuación que permiten determinar los valores de V_j, θ_j, V_k y θ_k :

$$\begin{aligned} \int_0^L EI \begin{Bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \xi_4 \end{Bmatrix}^{II} \{\xi_1 \quad \xi_2 \quad \xi_3 \quad \xi_4\}^{II} \begin{Bmatrix} V_j \\ \theta_j \\ V_k \\ \theta_k \end{Bmatrix} dx + \int_0^L cEI \begin{Bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \xi_4 \end{Bmatrix} \{\xi_1 \quad \xi_2 \quad \xi_3 \quad \xi_4\}^{II} \begin{Bmatrix} \dot{V}_j \\ \dot{\theta}_j \\ \dot{V}_k \\ \dot{\theta}_k \end{Bmatrix} dx \\ + \int_0^L m \begin{Bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \xi_4 \end{Bmatrix} \{\xi_1 \quad \xi_2 \quad \xi_3 \quad \xi_4\} \begin{Bmatrix} \ddot{V}_j \\ \ddot{\theta}_j \\ \ddot{V}_k \\ \ddot{\theta}_k \end{Bmatrix} dx = \int_0^L EI \begin{Bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \xi_4 \end{Bmatrix} P(x, t) dx + \begin{Bmatrix} -Q_j \\ M_j \\ Q_k \\ -M_k \end{Bmatrix} \end{aligned} \quad (3.120)$$

Donde Q y M son la fuerza cortante y el momento aplicado en el contorno del elemento. Utilizando notación matricial para la ecuación (3.120), toma la forma:

$$[M_E]\{\ddot{V}_E\} + [C_E]\{\dot{V}_E\} + [K_E]\{V_E\} = \{P_E\} \quad (3.121)$$

Donde:

$\{V_E\}$ = Vector de desplazamientos nodales del elemento

Donde la matriz de amortiguamiento del elemento teniendo en cuenta que es el material visco-elástico está dada por:

$$[C_E] = \int_0^L cEI \begin{Bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \xi_4 \end{Bmatrix}^{II} \{\xi_1 \quad \xi_2 \quad \xi_3 \quad \xi_4\}^{II} dx \quad (3.122)$$

La matriz de rigidez del elemento es expresada por:

$$[K_E] = \int_0^L EI \begin{Bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \xi_4 \end{Bmatrix}^{II} \{\xi_1 \quad \xi_2 \quad \xi_3 \quad \xi_4\}^{II} dx \quad (3.123)$$

El vector de la carga dada por:

$$\{P_E\} \int_0^L EIP \begin{Bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \xi_4 \end{Bmatrix} (x, t) dx + \begin{Bmatrix} -Q_j \\ M_j \\ Q_k \\ -M_k \end{Bmatrix} \quad (3.124)$$

Y la matriz de masa del elemento está definida por:

$$[M_E] = \int_0^L m \begin{Bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \xi_4 \end{Bmatrix} \{\xi_1 \quad \xi_2 \quad \xi_3 \quad \xi_4\} dx \quad (3.125)$$

Se conoce como matriz de masa consistente. Como puede verse, se calcula usando las mismas funciones que se utilizan en la obtención de la matriz de rigidez. Para las funciones que fueron utilizadas, tenemos:

$$[M] = \frac{mL}{420} \begin{bmatrix} 156 & 22L & 54 & -13L \\ 22L & 4L^2 & 13L & -3L \\ 54 & 13L & 156 & -22L \\ -13L & -3L & -22L & 4L^2 \end{bmatrix} \quad (3.126)$$

Comparando la ecuación (3.125) con la ecuación (3.123), se puede observar que la expresión para la matriz de rigidez implica la segunda derivada para las funciones de forma. Esto sugiere que para calcular la matriz de masa de un elemento, podemos utilizar funciones de forma diferente (de orden menor, por ejemplo) de las utilizadas para determinar la matriz de rigidez.

Una aproximación razonable utilizada como una alternativa practica en el método de elementos finitos para el cálculo de la matriz de masa para vibraciones de flexión en vigas, puede realizarse mediante el uso de las siguientes funciones de forma:

$$\begin{aligned} \xi_1 &= \begin{cases} 1 & x \leq L/2 \\ 0 & x > L/2 \end{cases} \\ \xi_2 &= 0 \\ \xi_3 &= \begin{cases} 0 & x < L/2 \\ 1 & x \geq L/2 \end{cases} \\ \xi_4 &= 0 \end{aligned} \quad (3.127)$$

Para estas funciones de forma, la matriz de masa calculada con la ecuación (3.125) es:

$$[M] = mL \begin{bmatrix} 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (3.128)$$

La matriz anterior es llamada matriz de masa concentrada (**lumped**), y se refiere a la inercia de los grados de libertad rotacionales que no es tomada en cuenta, y solo considerando la masa del elemento concentrada en los grados de libertad de traslación.

Existen otras aproximaciones posibles para la matriz de masa, sin embargo, la utilizada en este trabajo es la matriz de masa consistente. La determinación de la matriz de masa únicamente para condiciones axiales es:

$$m = \frac{\rho AL}{6} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \quad (3.129)$$

Por lo tanto la matriz de masa para un elemento considerando las componentes axiales será:

$$[M] = \frac{\rho AL}{420} \begin{bmatrix} 140 & 0 & 0 & 70 & 0 & 0 \\ 0 & 156 & 22L & 0 & 54 & -13L \\ 0 & 22L & 4L^2 & 0 & 13L & -3L \\ 70 & 0 & 0 & 140 & 0 & 0 \\ 0 & 54 & 13L & 0 & 156 & -22L \\ 0 & -13L & -3L & 0 & -22L & 4L^2 \end{bmatrix} \quad (3.130)$$

Para la matriz de masa igual que para la matriz de rigidez es necesario realizar la rotación de ejes y el ensamble de la matriz de masa de la estructura, para tal fin se utiliza el mismo procedimiento descrito en la sección 3.2.3 y 3.2.4.

3.4. MATRIZ DE AMORTIGUACIÓN

Debido a la dificultad de definir los coeficientes de amortiguamiento basándose en las características físicas del sistema, es más fácil de tratar a las fuerzas de amortiguación de manera aproximada. Hay varias formas de construir una matriz de amortiguamiento aproximada, entre los que podemos mencionar los siguientes: amortiguación proporcional a la masa ($[C] = \beta[M]$), amortiguación proporcional a la rigidez ($[C] = \beta_1[K]$) y la amortiguación de Rayleigh ($[C] = \beta[M] + \beta_1[K]$), donde β y β_1 son coeficientes que deben ser calculados (*apud Begambre 2004*).

La matriz de amortiguación es necesaria en las siguientes situaciones: cuando se realiza una integración directa de las ecuaciones de movimiento en lugar de la técnica de superposición y cuando se realiza un análisis no lineal.

3.5. DINÁMICA ESTRUCTURAL

La mecánica como una rama de la física esta subdividida en la estática y la dinámica. La estática estudia sistemas en el equilibrio estático, p. ej., un estado donde el sistema fuerzas internas compensan las fuerzas externas que actúan sobre el sistema. La Estática se refiere al hecho que el estado del sistema y las fuerzas aplicadas no varía en el tiempo, por lo tanto son independientes de este. La dinámica es el estudio de sistemas sometidos a la variación en el tiempo de fuerzas aplicadas. Como una consecuencia de la variabilidad en el tiempo de las fuerzas aplicadas, las fuerzas internas del sistema y su

estado (definido en términos de desplazamiento y deformación) también varía con el tiempo (la respuesta del sistema implica el movimiento). Mientras un problema estático tiene una única solución independiente del tiempo, la solución de un problema dinámico implica una descripción del estado del sistema en cada momento dentro del período de estudio.

La dinámica estructural puede ser considerada como el estudio de un cuerpo o estructura en equilibrio dinámico. La expresión matemática de este equilibrio es la ecuación de movimiento. Mientras la ecuación de equilibrio estática expresa el equilibrio entre las fuerzas internas de la estructura y fuerzas externas aplicadas, la ecuación de movimiento expresa el equilibrio en términos de fuerzas internas, externas (que es exactamente el mismo como en la ecuación de equilibrio estática), fuerza inercial y efectos de amortiguamiento. Estas ecuaciones de equilibrio dinámico se obtienen de la aplicación de las Leyes de Newton y el principio de D'Alembert (Maldonado y Chio, 2004).

3.5.1. Ecuación de movimiento

El movimiento de una estructura estará gobernado por una ecuación diferencial, denominada ecuación de movimiento. Las propiedades esenciales físicas de un sistema elástico estructural sometido a una carga externa dinámica son: su masa, propiedades de rigidez y la capacidad de absorción de energía o amortiguamiento. El principio de análisis dinámico puede ser ilustrado considerando una estructura simple de un piso como se muestra en la Figura 1, donde la estructura es sometida a una fuerza que varía tiempo $f(t)$, k es la constante que relaciona la desviación de piso lateral x con la fuerza transversal del piso, y por último la fuerza de amortiguación que es el resultado de la velocidad por un coeficiente c . Si la masa, m , es asumida para concentrarse en la viga, la estructura se convierte en un sistema de un grado solo de libertad. La ecuación de movimiento del sistema puede ser escrita como:

$$m\ddot{x} + c\dot{x} + kx = f(t) \quad (3.131)$$

Donde:

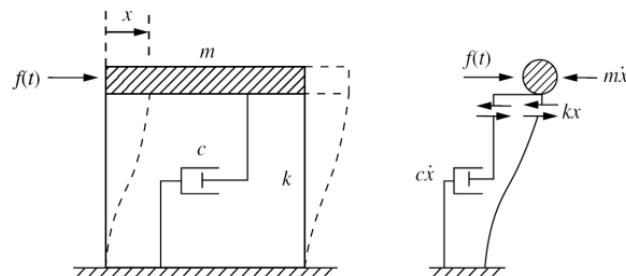
m = Masa del sistema.

c = Coeficiente de amortiguación del sistema.

k = Rigidez del sistema.

$f(t)$ = Fuerza dinámica aplicada al sistema.

Figura 22. Sistema de un grado de libertad sometido a una excitación externa $f(t)$



Fuente: Maldonado y Chio (2004)

La ecuación de movimiento de un sistema múltiples grados de libertad es similar al sistema de un grado de libertad, pero la rigidez k , la masa m y el amortiguamiento c son matrices. Por lo tanto la ecuación de movimiento de un sistema de múltiples grados de libertad puede ser escrita como:

$$[M]\{\ddot{X}\} + [C]\{\dot{X}\} + [K]\{X\} = \{F(t)\} \quad (3.132)$$

Como el moldeamiento de la estructura se realiza por medio de un análisis lineal, la matriz de amortiguamiento no es considerada ya que es despreciable (se considera amortiguamiento bajo), por lo tanto las propiedades dinámicas del sistema se determinan mediante la vibración libre no amortiguada.

3.5.2. Vibración libre no amortiguada

Cuando una estructura no está sometida a excitación externa alguna y su movimiento está gobernado únicamente por las condiciones iniciales, se considera que esta en vibración libre. El análisis de una estructura en vibración libre nos proporciona las propiedades dinámicas más importantes de la estructura, que son las frecuencias naturales y sus correspondientes modos de vibración.

Como anteriormente se menciona, el problema de vibración libre no amortiguada requiere que el vector de fuerza $\{F\}$ y la matriz de amortiguamiento sean igual a cero, por lo tanto la formulación de la ecuación de movimiento (3.132) con $\{F\} = 0$ y $[C] = 0$, es la siguiente:

$$[M]\{\ddot{X}\} + [K]\{X\} = \{0\} \quad (3.133)$$

La solución para el sistema anterior de ecuaciones diferenciales simultáneas tiene la forma:

$$x_i = a_i \text{sen}(\omega t - \delta) \quad i = 1, 2, 3, \dots, n \quad (3.134)$$

O expresado en forma matricial:

$$\{X\} = \{a\} \text{sen}(\omega t - \delta) \quad (3.135)$$

Donde a_i es la amplitud del movimiento en la coordenada i , n son los grados de libertad, δ es el ángulo de fase y ω es la frecuencia de vibración. Reemplazando la solución en la ecuación (3.132) tenemos:

$$-\omega^2 [M]\{a\} \text{sen}(\omega t - \delta) + [K]\{a\} \text{sen}(\omega t - \delta) = \{0\} \quad (3.136)$$

Reordenando los términos

$$[[K] - \omega^2 [M]]\{a\} = \{0\} \quad (3.137)$$

Que, en el caso general, es un sistema algebraico de n ecuaciones lineales homogéneas con n incógnitas, amplitudes a_i y ω^2 por determinar. Esta ecuación representa el problema conocido como problema característico. La solución de esta formulación matemática, requiere que el determinante de la matriz del factor de $\{a\}$ sea igual a cero, es decir, que la solución de las ecuaciones simultáneas lineales no homogénea sea no trivial requiere que el determinante de la matriz sea igual a cero:

$$|[K] - \omega^2 [M]| = 0 \quad (3.138)$$

Generalmente, la ecuación (3.138) resulta ser una ecuación algebraica de grado n con n incógnitas, amplitudes y ω^2 por determinar. Esta ecuación es conocida como ecuación característica del sistema. Las n raíces de esta ecuación se denominan valores propios o eigenvalores, que son siempre reales y positivas, y su raíz cuadrada corresponde a las frecuencias de vibración del sistema, en radianes por segundo. Además podemos resolver para cada ω , un vector a_1, a_2, \dots, a_n , llamado modo de vibración o eigenvector, en términos de una constante de proporcionalidad arbitraria.

De la ecuación (3.137) podemos obtener una característica muy importante de los modos normales, la propiedad de ortogonalidad. Escribiendo la ecuación (3.137) de la siguiente manera:

$$[K]\{a\} = \omega^2[M]\{a\} \quad (3.139)$$

Y tomando a ω_i^2 y ω_j^2 como dos valores propios distintos, y $\{a\}_i$ y $\{a\}_j$ sus correspondientes vectores propios, de esta manera podemos escribir:

$$[K]\{a\}_i = \omega_i^2[M]\{a\}_i \quad (3.140)$$

$$[K]\{a\}_j = \omega_j^2[M]\{a\}_j \quad (3.141)$$

Multiplicando en ambos lados la ecuación (3.140) por $\{a\}_j^T$, y la ecuación (3.141) por $\{a\}_i^T$, tenemos:

$$\{a\}_j^T [K]\{a\}_i = \omega_i^2 \{a\}_j^T [M]\{a\}_i \quad (3.142)$$

$$\{a\}_i^T [K]\{a\}_j = \omega_j^2 \{a\}_i^T [M]\{a\}_j \quad (3.143)$$

Transponiendo las matrices en ambos lados de la ecuación (3.142) y utilizando la propiedad de simetría tanto de la matriz de masa como de la matriz de rigidez, o sea:

$$[K]^T = [K] \text{ y } [M]^T = [M]$$

Tenemos:

$$\{a\}_i^T [K]\{a\}_j = \omega_i^2 \{a\}_i^T [M]\{a\}_j \quad (3.144)$$

Restando la ecuación (3.143) a la ecuación (3.144) obtenemos:

$$(\omega_i^2 - \omega_j^2)\{a\}_i^T [M]\{a\}_j = 0 \quad (3.145)$$

Si las frecuencias son distintas ($\omega_i^2 \neq \omega_j^2$) la ecuación (3.145) queda como:

$$\{a\}_i^T [M]\{a\}_j = 0 \quad (3.146)$$

Donde $\{a\}_i$ y $\{a\}_j$, son dos vectores modales cualesquiera y $[M]$ es la matriz de masa del sistema.

Como ya se mencionó anteriormente, las amplitudes de vibración en un modo normal son solo valores relativos, que conviene normalizarlos. Una manera apropiada y muy común, es la normalización con respecto a la masa, que se puede expresar de la siguiente manera:

$$\phi_{ij} = \frac{a_{ij}}{\sqrt{\{a\}_j^T [M]\{a\}_j}} \quad (3.147)$$

En la cual ϕ_{ij} es la componente normalizada i del vector modal j . Para vectores normalizados, la condición de ortogonalidad está dada, en general por:

$$\begin{aligned} \{\phi\}_i^T [M] \{\phi\}_j &= 0 \quad \text{para } i \neq j \\ \{\phi\}_i^T [M] \{\phi\}_j &= 1 \quad \text{para } i = j \end{aligned} \quad (3.148)$$

Otra condición de ortogonalidad se puede obtener a partir de la ecuación (3.144), teniendo en cuenta las condiciones de la ecuación (3.148), estas condiciones se pueden expresar como:

$$\begin{aligned} \{\phi\}_i^T [K] \{\phi\}_j &= 0 \quad \text{para } i \neq j \\ \{\phi\}_i^T [K] \{\phi\}_j &= \omega^2 \quad \text{para } i = j \end{aligned} \quad (3.149)$$

4. ALGORITMO PARTICLE SWARM OPTIMIZATION (PSO)

La optimización con enjambre de partículas, más conocida en la literatura científica como Particle Swarm Optimization (PSO) fue propuesto inicialmente por *Kennedy y Eberhart (1995)*, dicho algoritmo nace, del estudio y simulaciones del movimiento sincronizado e impredecible presentado principalmente en las bandadas de aves y bancos de peces, intrigados por la capacidad de estos grupos para separarse, reagruparse o encontrar alimento. Encontrando que había un intercambio de información entre los individuos de la especie, para encontrar el alimento, donde cada uno de los miembros puede beneficiarse de la experiencia adquirida por todos los demás miembros, ofreciendo esto una ventaja evolutiva fundamental para el desarrollo de la optimización con enjambre de partículas.

En la terminología utilizada en PSO, *Kennedy y Eberhart (1995)*, introducen el término general de partícula o agente para representar a cualquier tipo de individuos que exhiban un comportamiento social como grupo, de forma que interactúan entre sí. De acuerdo con los fundamentos teóricos del método, el movimiento de cada una de estas partículas hacia un objetivo común en dos dimensiones está condicionado por dos factores básicos, la memoria autobiográfica de la partícula y la influencia social de todo el enjambre. A nivel computacional, como método de optimización, esta filosofía puede extenderse a un espacio N-dimensional de acuerdo con el problema bajo análisis. La posición instantánea de cada una de las partículas de la población en el espacio N-dimensional representa una solución potencial, siendo N el número de incógnitas del problema original. Básicamente, el proceso evolutivo se reduce a mover cada partícula dentro del espacio de soluciones con una velocidad que variará de acuerdo a su velocidad actual, a la memoria de la partícula y a la información global que comparte el resto del enjambre, utilizando una función de fitness para cuantificar la calidad de cada partícula en función de la posición que ésta ocupe.

Dado que el PSO está basado en un concepto muy simple, y podía ser implementado en unas pocas líneas de código informático, fue desarrollado rápidamente por los autores. El PSO es similar a los algoritmos genéticos (AG) y la computación evolutiva en el que el sistema se inicia con una población (partículas o agentes) de soluciones distribuidas aleatoriamente, sin embargo, la diferencia radica en que a cada solución potencial se le asigna una velocidad aleatoriamente, por lo tanto las partículas pueden recorrer todo el espacio del problema (*Kennedy y Eberhart, 1995*).

En términos generales, la optimización de enjambre de partículas, al igual que otros algoritmos de computación evolutiva, se puede aplicar para resolver la mayoría de problemas de optimización y los problemas que puedan ser convertidos a problemas de optimización. Entre las áreas de aplicación con mayor potencial son: optimización multi-objetivo, patrón de reconocimiento, modelado de sistemas biológicos, programación, procesamiento de señales, teoría de juegos, aplicaciones de robótica, la toma de decisiones, planificación y optimización de recursos en redes de distribución de energía eléctrica, entrenamiento de redes neuronales, optimización de sistemas dinámicos, gestión de redes de sensores, planificación de red en servicios de telecomunicación, gestión empresarial entre otros (*Shi y Eberhart, 2001; Mercado Enrique y Seña Cristian, 2008; Pérez Jesús, 2005*). En el campo de la ingeniería civil se encontró una aplicación estructural al diseño de una viga en voladizo de 5 segmentos donde se minimizaban las

dimensiones b y h para obtener el máximo esfuerzo (*Venter y Sobieszczanski-Sobiesk, 2003*).

4.1. FUNDAMENTOS DEL MOVIMIENTO DE PARTÍCULAS

Para desarrollar modelos para aplicaciones en la vida artificial se realiza con base a cinco principios básicos de la inteligencia de enjambre (swarm intelligence). Primero es el principio de proximidad: la población debe ser capaz de realizar simples cálculos de espacio y tiempo. En segundo lugar está el principio de la calidad: la población debe ser capaz de responder a factores de calidad del entorno. En tercer lugar está el principio de la diversidad de respuesta: Las actuaciones se completan aunque un individuo falle. Cuarto es el principio de la estabilidad: la población no debe cambiar su modo de comportamiento cada vez que cambia el entorno. Quinto es el principio de la capacidad de adaptación: la población debe ser capaz de cambiar el modo de comportamiento, cuando sea necesario. Hay que tener en cuenta que los principios cuatro y cinco son los lados contrapuestos (*Kennedy y Eberhart, 1995*).

A continuación se expone como el PSO se desempeña de acuerdo a cada uno de los principios expuestos anteriormente. La proximidad se puede establecer con los movimientos en N dimensiones llevadas a cabo por la población durante una serie de intervalos de tiempo. La población está respondiendo a los factores de calidad con su característica de compartir información dentro de toda su población (la memoria de la partícula y el conocimiento social), para así determinar las mejores soluciones. Por las diferentes tendencias marcadas por la memoria personal de cada partícula y por la historia de la mejor posición visitada por todo el conjunto, garantiza una diversidad de respuestas. El PSO es estable mientras el individuo que hasta el momento ha reconocido la mejor posición históricamente visitada no presente cambio alguno, de lo contrario se genera la adaptación del PSO para seguir explorando el espacio del problema de acuerdo al cambio generado por la actualización de la partícula de la mejor posición históricamente visitada (*Kennedy y Eberhart, 1995*).

Cada partícula realiza un seguimiento de sus coordenadas en el espacio del problema, que son asociadas con la mejor solución que ha logrado hasta el momento (el valor es almacenado), este valor se denomina $pbest$. Otro "mejor" valor que es seguido por el PSO es el mejor valor global, y su ubicación, obtenida hasta el momento por cualquier partícula en la población, esta ubicación es llamada $gbest$. Como el sistema itera, los agentes individuales son atraídos hacia sus óptimos ($pbest$ y $gbest$), basado en la interacción de las búsquedas individuales y grupales de la población. Esto debido a que los factores estocásticos permiten una búsqueda minuciosa de los espacios entre las regiones que han demostrado ser relativamente buena, además de la exploración en regiones desconocidas en el espacio del problema (*Kennedy y Eberhart, 1995*).

4.2. OPERADOR DE VELOCIDAD Y PARÁMETROS DEL ALGORITMO

En la formulación de PSO, *Kennedy y Eberhart (1995)* definen la velocidad de la partícula como el único operador disponible para controlar la evolución de la optimización, donde cada partícula del enjambre se identifica con dos variables de estado inicializadas aleatoriamente dentro del espacio N -dimensional que establece el problema a optimizar:

un vector velocidad, v_i , y un vector de posición, x_i . Este último corresponde a una solución potencial al problema de optimización. Los límites de los parámetros a optimizar, conforman el espacio de búsqueda al cual debe restringirse el movimiento del enjambre.

Adicionalmente, cada partícula mantiene en la memoria información de la posición espacial asociada con la mejor solución históricamente visitada por ésta, p^i , y también conoce la posición de la mejor partícula o solución encontrada por todos sus congéneres, p^g .

En cada iteración del método, cada una de las partículas de la población recorre el espacio de soluciones con una velocidad v_i hacia nuevas posiciones x_i , de acuerdo con su propia experiencia p^i , y con la experiencia aportada por el mejor de sus convecinos, p^g . En las primeras versiones del algoritmo, esta formulación se reduce a las ecuaciones (4.1) y (4.2).

$$v_{k+1}^i = v_k^i + c_1 r_1 (p^i - x_k^i) + c_2 r_2 (p^g - x_k^i) \quad (4.1)$$

$$x_{k+1}^i = x_k^i + v_{k+1}^i \quad (4.2)$$

En (4.1), v_k^i y x_k^i representan, respectivamente, la velocidad y posición en la iteración o instante de tiempo k de la partícula i en la dimensión n -ésima del espacio de búsqueda. Los factores c_1 y c_2 son las denominadas constantes de aceleración cognitiva y social, que determinan en qué medida influyen sobre el movimiento de la partícula su propia memoria y la cooperación entre individuos, respectivamente. Los términos r_1 y r_2 son dos números aleatorios uniformemente distribuidos entre 0 y 1, cuyo objetivo es emular el comportamiento estocástico y un tanto impredecible que exhibe la población del enjambre. Después de calcular la nueva velocidad de la partícula i en la dimensión n , la nueva posición x_{k+1}^i se actualiza directamente de acuerdo con (4.2), donde se asume que la velocidad se aplica durante un cierto período de tiempo Δt , típicamente de valor unitario. El proceso descrito se extiende al espacio N -dimensional, de forma que se van componiendo iterativamente nuevos vectores de posición x_i , utilizando, como en cualquier otro método de cómputo evolutivo, una función de fitness para ponderar la calidad de dicha solución parcial, actualizando los vectores p^i y p^g . El movimiento de los agentes sobre el espacio de soluciones y, en consecuencia, el rendimiento del algoritmo, está condicionado por el grado de contribución de las tres componentes de la velocidad en (4.1), ya mencionadas y denominadas por este orden como momento, hábito o inercia para considerar la tendencia de la partícula; memoria o autoaprendizaje para incluir la experiencia de la propia partícula, y cooperación, conocimiento social, conocimiento de grupo o información compartida, para reflejar el intercambio de información y el comportamiento social como grupo.

Para acotar la velocidad de la partícula se especifica un valor máximo, $vmax$, que restringe la velocidad en cada dimensión al intervalo $[-vmax, vmax]$. Si el valor de $vmax$ es demasiado grande las partículas pueden sobrepasar e ignorar continuamente la zona con la solución global. En el extremo opuesto, si $vmax$ toma valores extremadamente pequeños las partículas explorarán el espacio de soluciones muy lentamente y podrán quedar atrapadas alrededor de soluciones locales, incapaces de librarse de la base de atracción. Sin embargo, si no se introdujese el parámetro de control $vmax$, el enjambre no convergería hacia un punto, sino que sufriría el fenómeno conocido como explosión del PSO, consistente en un comportamiento oscilatorio y creciente de la posición de las partículas, careciendo el método de potencial alguno como algoritmo de optimización.

El efecto de $vmax$, está fuertemente vinculado con la naturaleza del problema a optimizar, donde se convierte en el parámetro principal a sintonizar en el PSO, sin ninguna regla aparente que dicte las pautas a seguir en su elección para asegurar la convergencia del enjambre. Con el objetivo de reducir el efecto de $vmax$ y mejorar el control del alcance de la búsqueda sobre el espacio de soluciones, surgen versiones mejoradas del algoritmo que incorporan el concepto de peso inercial (*Shi y Eberhart, 2001*). Básicamente, para cada iteración, en el cálculo de la nueva velocidad de la partícula en cada dimensión, v_{k+1}^i , el modelo de peso inercial introduce como factor denominado peso inercial, w , tal que

$$v_{k+1}^i = wv_k^i + c_1r_1(p^i - x_k^i) + c_2r_2(p_k^g - x_k^i) \quad (4.3)$$

Con este factor se controla la tendencia de la partícula a continuar en la dirección en la que se estaba moviendo y se regula la relación entre capacidad de exploración del espacio de búsqueda y habilidad de convergencia hacia las soluciones locales o globales, dictadas por el segundo y tercer sumando de la ecuación (4.3). Aún en el modelo con peso inercial y dependiendo del problema a optimizar, el límite $vmax$ puede resultar imprescindible para evitar que las partículas oscilen demasiado rápido en las proximidades de una región de interés sin explorarla adecuadamente.

Una alternativa al modelo inercial lo constituye el denominado factor de constricción introducido por *Clerc (1999)*. Partiendo de la necesidad de encontrar un razonamiento a la necesidad del límite $vmax$ y a la explosión del PSO, se iniciaron estudios para analizar la trayectoria que describen las partículas. El modelo propuesto por Clerc es regido por las ecuaciones:

$$v_{k+1}^i = K[v_k^i + c_1r_1(p^i - x_k^i) + c_2r_2(p_k^g - x_k^i)] \quad (4.4)$$

$$x_{k+1}^i = x_k^i + v_{k+1}^i \quad (4.5)$$

Donde K es el denominado coeficiente de constricción, el cual se expresa como

$$K = \frac{2}{\left|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}\right|}, \quad \text{Donde } \varphi = c_1 + c_2, \quad \varphi > 4 \quad (4.6)$$

Tal y como se indica en la ecuación (4.6), el coeficiente de constricción no está definido para $\varphi \leq 4$. A medida que φ toma valores por encima de 4, K decrece paulatinamente, acentuando el efecto de amortiguamiento del movimiento. En este modelo de constricción típicamente se utilizan valores de $\varphi = 4.1$ ($c_1 = c_2 = 2.05$), lo cual reporta un valor para el coeficiente de constricción de $K = 0.729$. Algebraicamente, esta configuración es equivalente a utilizar el modelo inercial de (4.3) con $w = 0.729$ y $c_1 = c_2 = 1.49445$. La utilización de ambos modelos de PSO, con peso inercial y con factor de constricción, está muy extendida en la bibliografía y aunque es el modelo de Clerc el que ofrece mejores resultados, los parámetros óptimos del algoritmo están íntimamente ligados con la naturaleza del problema físico a optimizar (*apud Shi y Eberhart, 2001; Pérez Jesús, 2005*).

Otros enfoques del PSO fueron realizados por: *CUI et al. (2008)*, *LEE et al. (2008)*, *PARSOPOULOS et al. (2007)*, *KUMAR et al. (2007)*, *JIAO et al. (2008)* y *XI-HUAI et al. (2004)*

4.3. SELECCIÓN DE PARÁMETROS

Como en cualquier otro método de cómputo evolutivo, en el PSO el ajuste de los parámetros condiciona el rendimiento del algoritmo de optimización, dicha selección está íntimamente ligada con la naturaleza del problema a optimizar, siendo preciso lograr un balance óptimo entre exploración y convergencia. Al margen de los parámetros intrínsecos del algoritmo, debe resaltarse la importancia que adquiere la función de fitness. La utilización de una función de fitness que no introduzca una métrica adecuada para pesar la bondad de cada partícula hará emerger carencias del PSO que erróneamente pueden asociarse con el algoritmo en sí, aún siendo más propias de los defectos de la propia función de fitness (*Pérez Jesús, 2005*).

En lo referente al peso inercial, se utilizan comúnmente diferentes valores dependiendo del compromiso que se establezca entre exploración y rapidez de convergencia. Un peso inercial alto reduce la influencia de la nostalgia y de la cooperación sobre el movimiento de las partículas, favoreciendo la exploración. Por el contrario, un peso inercial muy bajo promueve la convergencia hacia la solución propuesta por la propia partícula p^i , y por la mejor solución de conjunto, p^g . Shi y Eberhart (1998) sugieren un peso inercial constante entre $0.8 < w < 1.4$. Por otra parte, para aunar exploración y convergencia del espacio de soluciones, se suele utilizar una variante con un peso inercial que decrece linealmente durante el transcurso de la simulación (*apud Shi y Eberhart, 2001; Venter and Sobieszczanski-Sobiesk, 2003; Pérez Jesús, 2005*).

La selección del peso inercial no se debe dissociar de la elección de las constantes de aceleración cognitiva (c_1) y social (c_2) que controlan el flujo de información dentro del enjambre, si $C_2 > C_1$ entonces la partícula va a confiar más en los resultados de la búsqueda del enjambre, de lo contrario, $C_2 < C_1$, la partícula va a tener más confianza en los resultados de su búsqueda que en los del enjambre. Con el fin de mantener un equilibrio entre las capacidades de búsqueda local y global se propone usar de forma general las condiciones de $0.5 < C_1 < 2$ y $0.5 < C_2 < 2$. La literatura propone un valor de 2.0 para ambas tasas de aprendizaje (c_1 y c_2), con el objetivo de que estos factores estocásticos tengan en media un valor de 1.0, de forma que los agentes sobrevuelen el objetivo la mitad del tiempo. Esta versión suele combinarse con el peso inercial linealmente decreciente (*apud Venter and Sobieszczanski-Sobiesk, 2003; Pérez Jesús, 2005*).

Otro parámetro que precisa ser seleccionado con extrema rigurosidad es el tamaño de la población. Poblaciones demasiado grandes exploran minuciosamente el espacio de búsqueda, pero el coste computacional asociado con el aumento del número de evaluaciones de la función de fitness puede resultar excesivamente elevado. Estudios paramétricos revelan que una población de alrededor de 30 agentes es suficiente para múltiples problemas, y típicamente se utilizan poblaciones que oscilan entre 10 y 50 partículas, o entre 100 y 200 partículas para abordar problemas más complejos (*apud Pérez Jesús, 2005*).

En este trabajo, los valores de $c_1 = 2$ y $c_2 = 2$ fueron elegidos para los dos parámetros de confianza como es sugerido en la literatura. El parámetro de peso de inercia es de $w = 1$ ajustado dinámicamente durante la optimización, como lo propone Shi y Eberhart (2001). Por otra parte se realiza una aplicación del algoritmo PSO a varias funciones

reportadas en la literatura internacional con el fin de observar la influencia tanto de la función objetivo como del tamaño de las partículas en el desempeño del mismo.

4.4. ENJAMBRE INICIAL

El enjambre inicial generalmente es creado de tal forma que las partículas están distribuidas aleatoriamente en todo el espacio de diseño, cada una con un vector de velocidad inicial aleatoria. En el presente trabajo se utilizó el enfoque realizado por *Venter y Sobieszcanski-Sobiesk (2003)*, donde las siguientes ecuaciones se utilizan para obtener la posición inicial aleatoria y vectores de velocidad:

$$x_0^i = x_{min} + r_3(x_{max} - x_{min}) \quad (4.7)$$

$$v_0^i = \frac{x_{min} + r_4(x_{max} - x_{min})}{\Delta t} \quad (4.8)$$

En las ecuaciones (4.7) y (4.8), x_0^i es la posición inicial del vector y v_0^i es el vector de velocidad inicial de la partícula i , r_3 y r_4 son números aleatorios independientes entre 0 y 1, x_{min} es el vector de límite inferior y x_{max} es el vector de límite superior para las variables de diseño. Un paso de unidad de tiempo Δt se utiliza a lo largo del presente trabajo.

Por otro lado se considero la generación aleatoria bien distribuida de las partículas por medio del muestreo por hipercubos latinos, más conocido como Latin Hypercube Sampling (LHS). El método estadístico de muestreo por hipercubos latinos fue diseñado para una generación de colecciones plausibles de valores de una distribución multidimensional. En el contexto del muestreo estadístico, una malla cuadrada que contiene posiciones muestreadas, es un cuadrado latino si y solo si existe una muestra en cada fila y en cada columna, por lo tanto un hipercubo latino es una generalización de este concepto a un número arbitrario de dimensiones (*apud Serrano Víctor, 2007*).

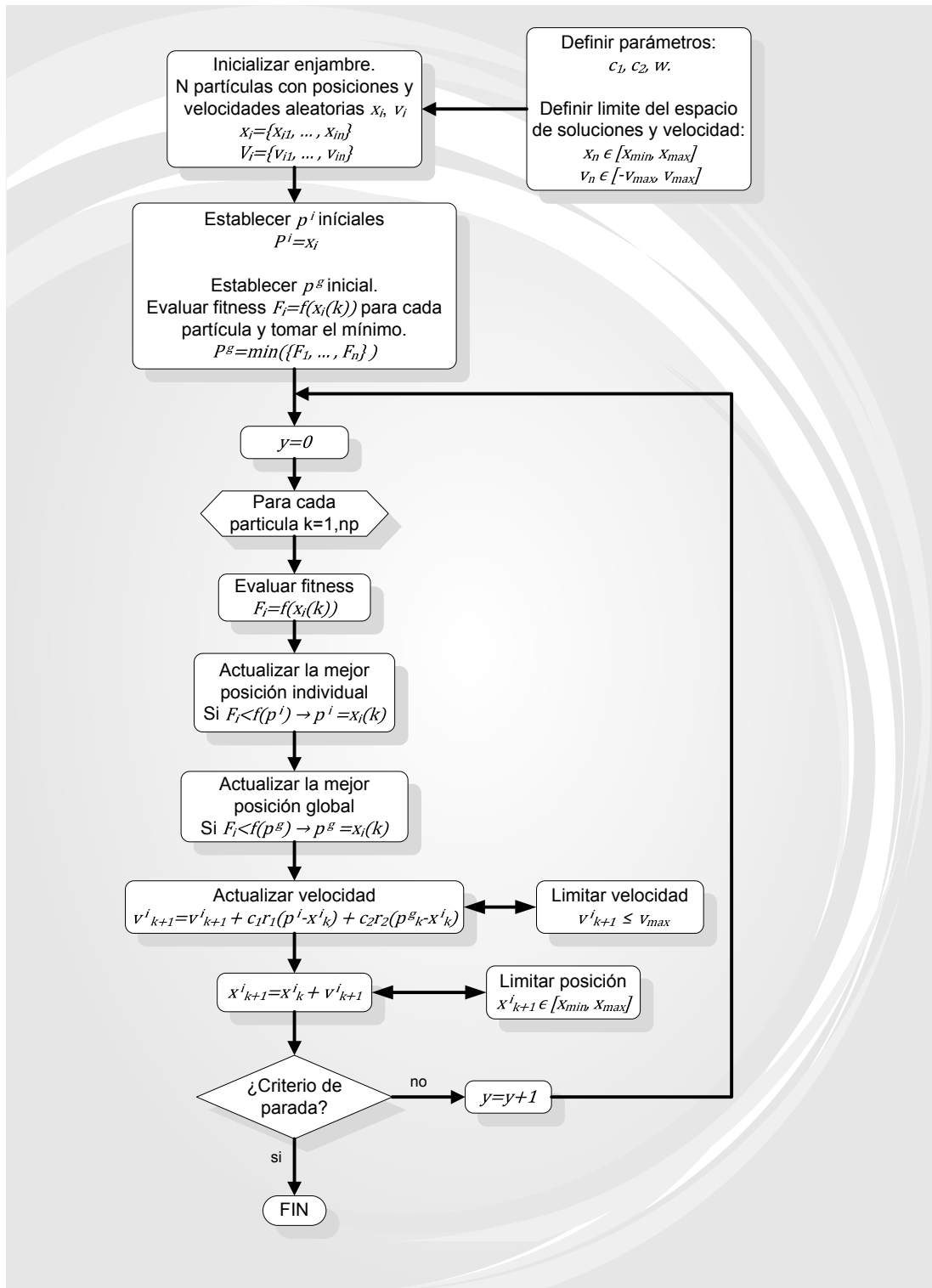
4.5. CRITERIO DE CONVERGENCIA

Un sólido criterio de convergencia es importante para cualquier optimizador, generalmente para evitar la evaluación de funciones adicionales después de que una solución óptima es encontrada. El criterio de convergencia considerado en este trabajo está basado en la determinación de la desviación estándar (DevStd) de la función objetivo en un número definido de iteraciones consecutivas.

La DevStd de los valores de la función objetivo para el subconjunto de las mejores partículas fue determinada, si la DevStd fuera menor que un valor umbral determinado, se asumía la convergencia del algoritmo.

En el presente trabajo, un subconjunto de las mejores partículas del enjambre de 20 iteraciones consecutivas fue considerado, y una DevStd umbral de 1E-5 fue utilizada. Un valor de partida de $w = 1.0$ es utilizado para realizar una búsqueda más global inicialmente y fue dinámicamente reducido durante todo el proceso de optimización.

Figura 23. Diagrama de flujo PSO básico



Fuente: Los autores

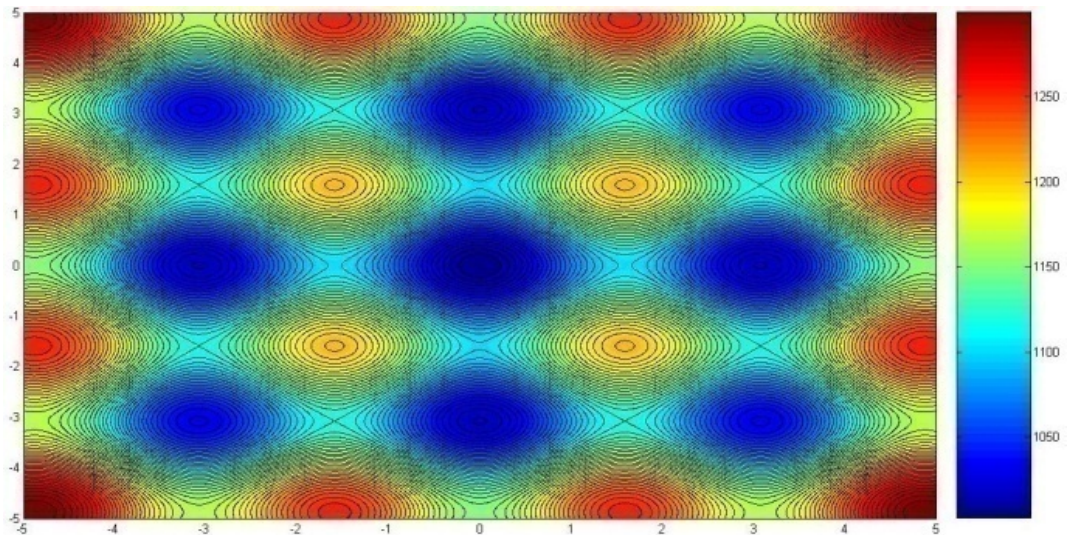
4.6. EJEMPLO DE APLICACIÓN

El ejemplo de aplicación mostrado a continuación tiene como finalidad proporcionarle al lector una guía básica sobre el funcionamiento del PSO, de esta forma el ejemplo se realizó únicamente con dos partículas por lo que no se garantiza que el algoritmo obtenga el valor óptimo global de la función. Para este ejemplo, se consideró la siguiente función en dos dimensiones sin restricciones en un intervalo de $[-5,5]$:

$$F(X_1, X_2) = X_1^2 - 100\text{Cos}(X_1) - 100\text{Cos}(X_1^2/30) + X_2^2 - 100\text{Cos}(X_2)^2 - 100\text{Cos}(X_2^2/30) + 1400 \quad (4.9)$$

Donde la función objetivo considerada debe minimizarse. Este problema tiene muchos mínimos locales, como se muestra en la figura 24, que proporciona un gráfico de contorno de la función sobre el espacio de diseño. El mínimo global tiene un valor de 1000 y está localizado en $x_1 = x_2 = 0$.

Figura 24. Función del ejemplo de aplicación



Fuente: Los autores

El procedimiento del algoritmo básico PSO implementado según el diagrama de flujo mostrado en la figura 23 es el siguiente:

- 1) Definición de parámetros, valores máximos y mínimos permitidos para la velocidad de cada partícula.
- 2) Comenzar con un conjunto inicial de partículas, generalmente distribuidas aleatoriamente en todo el espacio de diseño.
- 3) Calcular el vector de velocidad para cada partícula en el enjambre.
- 4) Actualización de la posición de cada partícula, utilizando su posición anterior y la actualización de los vectores de velocidad.
- 5) Ir al paso 2) y repetir hasta la convergencia.

Los parámetros del algoritmo se seleccionaron en la sección 4.3 donde los parámetros de confianza son $c_1 = c_2 = 2$ y el peso de inercia es $w = 1$ reducido linealmente en cada iteración como $w_{i+1} = w_i * 0.98$.

La velocidad de cada partícula deberá encontrarse dentro de un rango que es establecido por el “paso” el cual está definido en función del intervalo de búsqueda de las partículas como:

$$P = \frac{|x_{min} - x_{max}|}{10} \quad \therefore \quad P = \frac{|-5 - 5|}{10} \quad \therefore \quad P = 1 \quad (4.10)$$

Con lo cual, la velocidad para cada partícula debe encontrarse dentro del intervalo $(-1,1)$.

ENJAMBRE INICIAL

La distribución del enjambre inicial se realiza por medio de una distribución aleatoria en todo el espacio de diseño, de acuerdo con las ecuaciones (4.7) y (4.8) de la sección 4.4. Para la posición y la velocidad se debe tener en cuenta, para este ejemplo que: $x_{min} = -5$ y $x_{max} = 5$.

➤ Partícula 1

La coordenada x_1 se obtiene así: utilizando Excel o una calculadora común en la opción Aleatorio o rand se obtienen los valores necesarios para el ejercicio, en este caso se obtuvo:

$$r_3 = 0.8147 \quad r_4 = 0.9058$$

Introduciendo estos valores en la ecuación (4.7) y (4.8) se obtiene:

$$x_1^1 = -5 + 0.8147(5 - (-5)) \quad \therefore \quad x_1^1 = 3.1472$$

$$v_1^1 = \frac{-5 + 0.9058(5 - (-5))}{1} \quad \therefore \quad v_1^1 = 4.0579$$

Se realiza el mismo procedimiento para obtener la coordenada x_2 , donde se deben obtener valores aleatorios de r_3 y r_4 distintos:

$$r_3 = 0.1270 \quad r_4 = 0.9134$$

Introduciendo estos valores en la ecuación (4.7) y (4.8) se obtiene:

$$x_2^1 = -5 + 0.1270(5 - (-5)) \quad \therefore \quad x_2^1 = -3.7301$$

$$v_2^1 = \frac{-5 + 0.9134(5 - (-5))}{1} \quad \therefore \quad v_2^1 = 4.1338$$

Por lo tanto para la partícula 1 obtenemos:

$$\text{Coordenadas: } x^1 = (3.1472, -3.7301)$$

$$\text{Velocidad: } v^1 = (4.0579, 4.1338)$$

Debido a que el espacio de búsqueda está restringido al intervalo $[-5,5]$, es necesario verificar que cada una de las coordenadas x^1 se encuentre dentro del intervalo de lo

contrario se modifica la coordenada correspondiente que viole esta limitación. Para la partícula 1 esta condición es satisfecha.

Por otro lado la velocidad debe encontrarse dentro del rango “p” [-1,1] establecido en la ecuación (4.10) de lo contrario debe ajustarse. Para la partícula 1 esta limitación no es cumplida por lo que v^1 es ajustada a: $v^1 = (1, 1)$

➤ Partícula 2

El procedimiento de obtención y verificación de datos es exactamente el mismo que para la partícula 1, por lo tanto se obtienen los valores aleatorios y se continúa con los mismos pasos

$$r_3 = 0.6324 \quad r_4 = 0.0975$$

Introduciendo estos valores en la ecuación (4.7) y (4.8) se obtiene:

$$x_1^2 = -5 + 0.6324(5 - (-5)) \quad \therefore \quad x_1^2 = 1.3236$$

$$v_1^2 = \frac{-5 + 0.0975(5 - (-5))}{1} \quad \therefore \quad v_1^2 = -4.0246$$

Se realiza el mismo procedimiento para obtener la coordenada x_2 , donde se deben obtener valores aleatorios de r_3 y r_4 distintos:

$$r_3 = 0.2785 \quad r_4 = 0.5469$$

$$x_{min} = -5 \quad x_{max} = 5$$

Introduciendo estos valores en la ecuación (4.7) y (4.8) se obtiene:

$$x_2^2 = -5 + 0.2785(5 - (-5)) \quad \therefore \quad x_2^2 = -2.2150$$

$$v_2^2 = \frac{-5 + 0.5469(5 - (-5))}{1} \quad \therefore \quad v_2^2 = 0.4688$$

Por la tanto para la partícula 2 obtenemos:

$$\text{Coordenadas: } x^2 = (1.3236, -2.2150)$$

$$\text{Velocidad: } v^2 = (-4.0246, 0.4688)$$

Como las dos componentes de la partícula 2 se encuentran dentro del intervalo [-5,5], no se le realiza modificación alguna.

Por otro lado, la primera componente de la velocidad no se encuentra en el intervalo [-1,1], debe ser ajustada a: $v^2 = (-1, 0.4688)$.

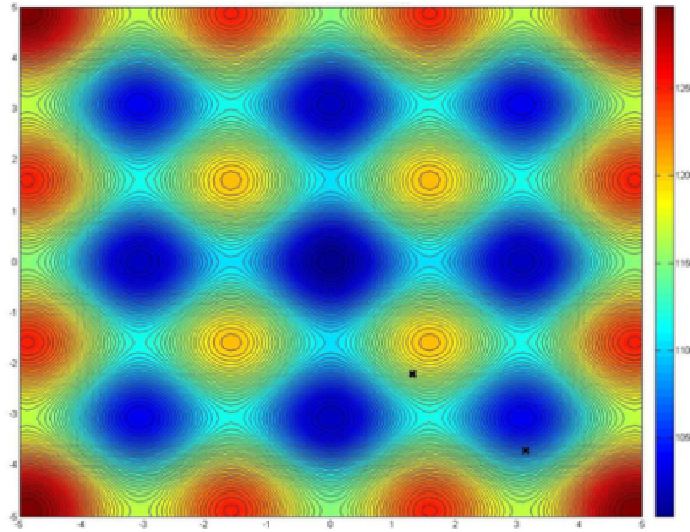
La ubicación de las dos partículas distribuidas aleatoriamente dentro de la función se puede observar en la figura 25.

Para continuar con el procedimiento se debe obtener la mejor coordenada local de cada partícula. Debido a que esta es la primera iteración las mejores partículas locales serán las que se calcularon anteriormente en el enjambre inicial, es decir:

$$p^L = (3.1472, -3.7301) \Rightarrow \text{Particula 1}$$

$$(1.3236, -2.2150) \Rightarrow \text{Particula 2}$$

Figura 25. Distribución de las partículas iniciales dentro de la función



Fuente: Los autores

Ahora se evalúa la función objetivo en los mejores puntos locales determinados anteriormente, esto se realiza para cada partícula de la siguiente forma:

evaluando la ecuación (4.9) para la partícula 1 obtenemos:

luego evaluando la ecuación (4.9) para la partícula 2 obtenemos:

Seguidamente, se determina el mejor punto global, el cual corresponde al punto local que genera el valor mínimo de la ecuación (4.9), es decir 1070.6061, por lo tanto:

ITERACIÓN 1

A continuación se actualiza la velocidad para cada partícula de acuerdo a la ecuación (4.3) descrita en la sección 4.2, por lo tanto:

➤ Partícula 1

Para

Para determinar la coordenada obtenemos los siguientes números aleatorios:

Introduciendo estos valores en la ecuación (4.3), y recordando que:

y obtenemos:

Como la componente de la velocidad se encuentra en el intervalo $[-1,1]$, no es ajustada.

Teniendo la velocidad se determina la nueva posición para la componente x_1 de acuerdo a la ecuación (4.2) de la sección 4.2, por lo tanto:

$$x_1^1 = 3.1472 + 1 \quad \therefore \quad x_1^1 = 4.1472$$

Para x_2

Realizando el mismo procedimiento que para x_1 obtenemos:

$$r_1 = 0.1576 \quad r_2 = 0.9706$$

Introduciendo estos valores en la ecuación (4.3), resulta:

$$v_2^1 = 1 * 1 + 2 * 0.1576((-3.7301) - (-3.7301)) + 2 * 0.9706((-3.7301) - (-3.7301))$$
$$v_2^1 = 1$$

Como la componente de la velocidad se encuentra en el intervalo $[-1,1]$, no es ajustada.

Ahora se determina la nueva posición para la componente x_2 de acuerdo a la ecuación (4.2), por lo tanto:

$$x_2^1 = -3.7301 + 1 \quad \therefore \quad x_2^1 = -2.7301$$

Por lo tanto para la partícula 1 obtenemos:

$$\text{Coordenadas: } x^1 = (4.1472, -2.7301)$$

$$\text{Velocidad: } v^1 = (1, 1)$$

Como las dos componentes de la partícula 1 se encuentran dentro del intervalo $[-5,5]$, no se le realiza modificación alguna.

➤ Partícula 2

Para x_1

Realizando el mismo procedimiento que para la partícula 1 obtenemos:

$$r_1 = 0.9572 \quad r_2 = 0.4854$$

Introduciendo estos valores en la ecuación (4.3), sabiendo que $v^2 = (-1, 0.4688)$, resulta:

$$v_1^2 = 1 * (-1) + 2 * 0.9572(1.3236 - 1.3236) + 2 * 0.4854(3.1472 - 1.3236)$$
$$v_1^2 = 0.7703$$

Como la componente de la velocidad se encuentra en el intervalo $[-1,1]$, no es ajustada.

Teniendo la velocidad se puede determinar la nueva posición para la componente x_1 de acuerdo a la ecuación (4.2), por lo tanto:

$$x_1^2 = 1.3236 + 0.7703 \quad \therefore \quad x_1^2 = 2.0939$$

Para x_2

Realizando el mismo procedimiento que para obtenemos:

Introduciendo estos valores en la ecuación (4.3), resulta:

Como la componente de la velocidad se encuentra en el intervalo $[-1,1]$, no es ajustada.

Teniendo la velocidad se determina la nueva posición para la componente de acuerdo a la ecuación (4.2), por lo tanto:

Por lo tanto para la partícula 2 obtenemos:

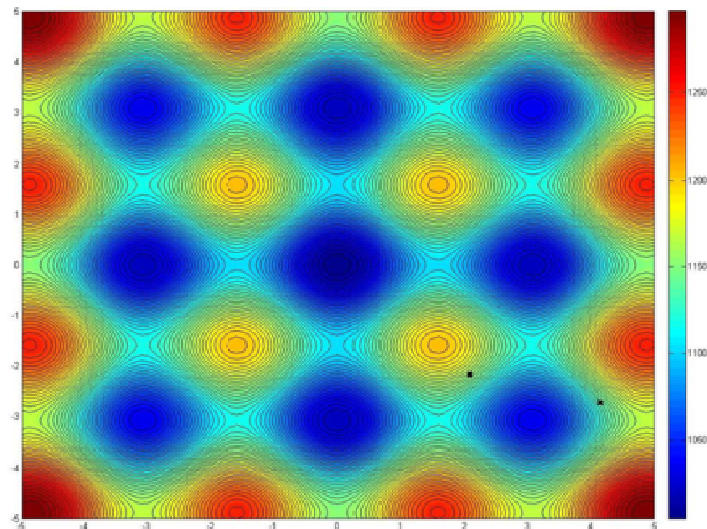
Coordenadas:

Velocidad:

Como las dos componentes de la partícula 2 se encuentran dentro del intervalo $[-5,5]$, no se le realiza ninguna modificación.

La nueva ubicación de las dos partículas dentro de la función se puede observar en la siguiente figura:

Figura 26. Primera actualización de las partículas



Fuente: Los autores

Ahora se realiza la evaluación la función (ecuación (4.9)) para cada una de las partículas, por lo tanto:

para la partícula 1 obtenemos:

$$F(4.1472, -2.7301) = 1131.0284$$

luego para la partícula 2 obtenemos:

$$F(2.0939, -2.1762) = 1154.0903$$

Para determinar los mejores puntos locales es necesario realizar la comparación de los resultados de las dos últimas evaluaciones de la función y tomar los puntos para los cuales la función de cómo resultado los valores mínimos, esto se realiza para cada partícula, de la siguiente forma:

$$P^L = \begin{matrix} (3.1472, -3.7301) & \Rightarrow & \text{Particula 1} \\ (1.3236, -2.2150) & \Rightarrow & \text{Particula 2} \end{matrix}$$

Teniendo en cuenta que la evaluación de la función para cada partícula es:

para la partícula 1 obtenemos:

$$F(3.1472, -3.7301) = 1070.6061$$

y para la partícula 2 obtenemos:

$$F(1.3236, -2.2150) = 1166.1063$$

Seguidamente se determina el mejor punto global el cual corresponde al punto local que nos genera el valor mínimo de la ecuación (4.9), es decir 1070.6061, por lo tanto:

$$P^G = (3.1472, -3.7301)$$

ITERACIÓN 2

A continuación se actualiza la velocidad para cada partícula de acuerdo a la ecuación (4.3), por lo tanto, en esta etapa se obtiene:

➤ Partícula 1

Para x_1

Para determinar la coordenada se calculan los siguientes números aleatorios:

$$r_1 = 0.9575 \quad r_2 = 0.9649$$

Introduciendo estos valores en la ecuación (4.3), y recordando que: $c_1 = 2$, $c_2 = 2$, $w = 0.98$ y $v^1 = (1, 1)$ se obtiene:

$$v_1^1 = 0.98 * 1 + 2 * 0.4218(3.1472 - 4.1472) + 2 * 0.9157(3.1472 - 4.1472)$$

$$v_1^1 = -1.6950$$

Como la componente de la velocidad no se encuentra en el intervalo $[-1, 1]$, es necesario ajustarla, por lo tanto $v_1^1 = -1$.

Teniendo la velocidad se procede a determinar la nueva posición para la componente x_1 de acuerdo a la ecuación (4.2), por lo tanto:

$$x_1^1 = 4.1472 + (-1) \quad \therefore \quad x_1^1 = 3.1472$$

Para x_2

Realizando el mismo procedimiento que para x_1 obtenemos:

$$r_1 = 0.7922 \quad r_2 = 0.9595$$

Introduciendo estos valores en la ecuación (4.3), resulta:

$$v_2^1 = 0.98 * 1 + 2 * 0.7922((-3.7301) - (-2.7301)) + 2 * 0.9595((-3.7301) - (-2.7301))$$

$$v_2^1 = -2.5234$$

Como la componente de la velocidad no se encuentra en el intervalo $[-1,1]$, es necesario ajustarla, por lo tanto $v_2^1 = -1$.

Teniendo la velocidad se calcula la nueva posición para la componente x_2 de acuerdo a la ecuación (4.2), por lo tanto:

$$x_2^1 = -2.7301 + (-1) \quad \therefore \quad x_2^1 = -3.7301$$

Por lo tanto para la partícula 1 obtenemos:

$$\text{Coordenadas: } x^1 = (3.1472, -3.7301)$$

$$\text{Velocidad: } v^1 = (-1, -1)$$

Como las dos componentes de la partícula 1 se encuentran dentro del intervalo $[-5,5]$, no se le realiza modificación alguna.

➤ Partícula 2

Para x_1

Realizando el mismo procedimiento que para la partícula 1 obtenemos:

$$r_1 = 0,6557 \quad r_2 = 0,0357$$

Introduciendo estos valores en la ecuación (4.3), sabiendo que $v^2 = (0.7703, 0.0389)$, resulta:

$$v_1^2 = 0.98 * 0.7703 + 2 * 0.6557(2,0939 - 2,0939) + 2 * 0.0357(3.1472 - 2,0939)$$

$$v_1^2 = 0,8301$$

Como la componente de la velocidad se encuentra en el intervalo $[-1,1]$, no es ajustada.

Teniendo la velocidad, se calcula la nueva posición, como fue explicada anteriormente, obteniendo:

$$x_1^2 = 2,0939 + 0.8301 \quad \therefore \quad x_1^2 = 2.9240$$

Para x_2

Realizando el mismo procedimiento que para x_1 obtenemos:

$$r_1 = 0,8491 \quad r_2 = 0,9340$$

Introduciendo estos valores en la ecuación (4.3), resulta:

$$v_2^2 = 0.98 * 0.0389 + 2 * 0.8491((-2.1762) - (-2.1762)) + 2 * 0,9340((-3.7301) - (-2.1762))$$

$$v_2^2 = -2.8647$$

Como la componente de la velocidad no se encuentra en el intervalo $[-1,1]$, es necesario ajustarla, por lo tanto $v_2^2 = -1$.

La nueva posición para la componente de acuerdo a la ecuación (4.2), es:

De esta forma para la partícula 2 se obtiene:

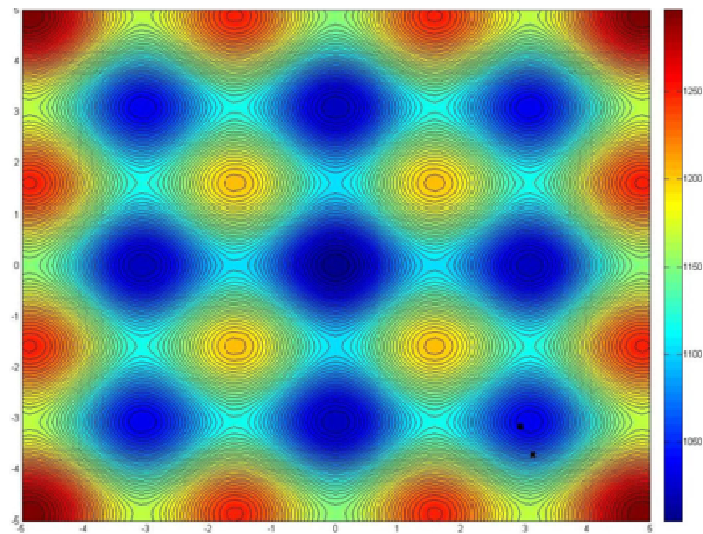
Coordenadas:

Velocidad:

Como las dos componentes de la partícula 1 se encuentran dentro del intervalo $[-5,5]$, no se le realiza ningún cambio.

La nueva ubicación de las dos partículas dentro de la función se puede observar en la siguiente figura:

Figura 27. Segunda actualización de las partículas



Fuente: Los autores

Ahora se realiza la evaluación la función (ecuación (4.9)) para cada una de las partículas, por lo tanto:

para la partícula 1 obtenemos:

plego para la partícula 2 obtenemos:

Realizando el proceso indicado en las anteriores iteraciones, obtenemos:

Teniendo en cuenta que la evaluación de la función para cada partícula es:

para la partícula 1:

y para la partícula 2:

Seguidamente se continua con la determinación del mejor punto global el cual corresponde al punto local que nos genera el valor mínimo de la ecuación (4.9), es decir 1070.6061, por lo tanto:

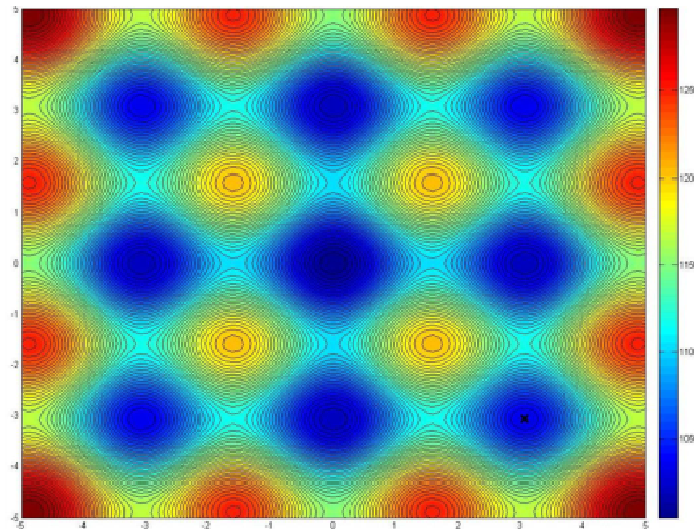
En este punto es conveniente detenerse debido a que para el resto de las iteraciones se realiza el mismo procedimiento, los datos de todo el proceso se encuentran consignados en el anexo A, lo que se muestra a continuación es la culminación del proceso según el criterio de convergencia descrito en la sección 4.5

Mejor coordenada global

Con un valor de la función de:

Finalmente se presenta la grafica de la última iteración, donde se puede observar el punto óptimo encontrado por el algoritmo.

Figura 28. Última actualización de las partículas



Fuente: Los autores

4.7. EVALUACIÓN DEL ALGORITMO PSO CON FUNCIONES DE LA LITERATURA INTERNACIONAL

Antes de la implementación del algoritmo de optimización con enjambre de partículas (PSO) en la identificación de daños en un pórtico de 2D, se considero la implementación del PSO en varias funciones matemáticas multidimensionales reportadas en la literatura internacional.

Esta implementación del PSO en las funciones matemáticas tiene como finalidad conocer el comportamiento, debilidades y fortalezas del algoritmo frente a diversos planteamientos matemáticos multidimensionales. Para tal fin se realizó la programación del algoritmo en el lenguaje de programación conocido como MATLAB® (ver Anexo B)

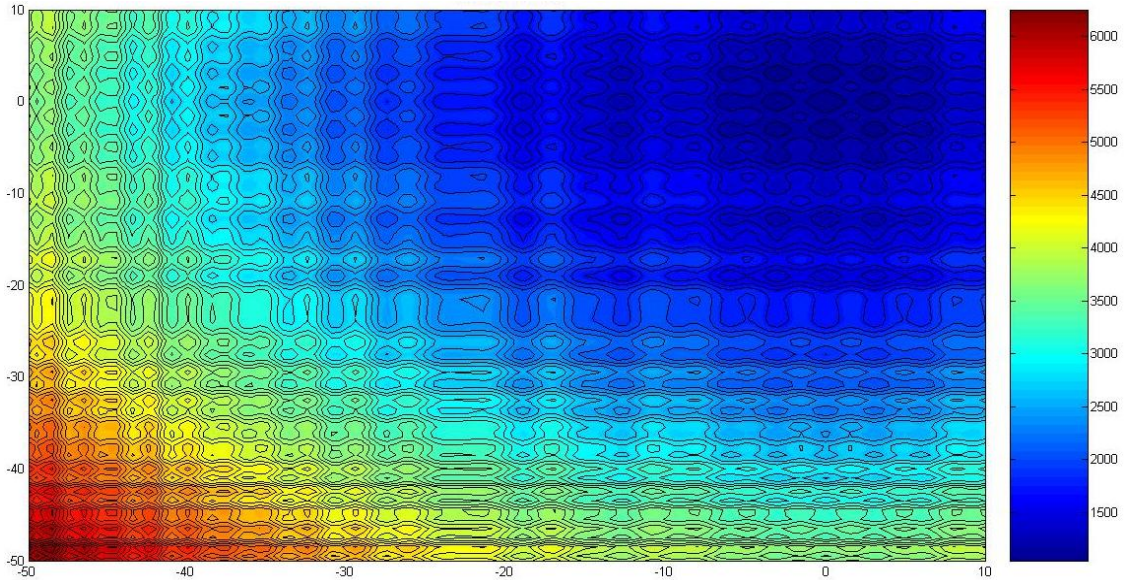
Con el fin de determinar el desempeño del algoritmo se consideraron 4 funciones matemáticas: Venter, N-Dimensional, Schwefel y Rosenbrock mostradas en las ecuaciones 4.11 a 4.14 y en las figuras 29 a 32. A continuación se presentan las diferentes funciones en las cuales se realizó la implementación del PSO. Para cada una de ellas se proporciona un diagrama de contorno en dos dimensiones dentro del intervalo establecido para realizar el análisis.

➤ Función VENTER

$$F(X_1, X_2) = X_1^2 - 100\cos(X_1) - 100\cos(X_1^2/30) + X_2^2 - 100\cos(X_2)^2 - 100\cos(X_2^2/30) + 1400 \quad (4.11)$$

$$\text{Limites } [-50,10] \quad , \quad F_{\text{optimo}}(0,0) = 1000$$

Figura 29. Función VENTER 2D



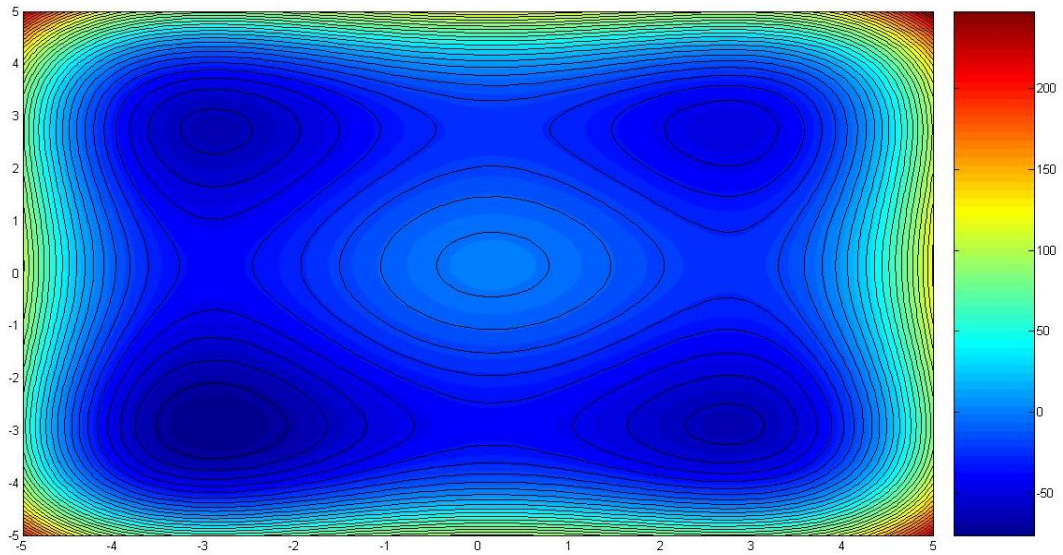
Fuente: Los autores

➤ N-DIMENSIONAL TEST FUNCTION

$$F(x) = \frac{1}{N} \sum_{i=1}^N (X_i^4 - 16X_i^2 + 5X_i) \quad (4.12)$$

$$\text{Limites } [-5,5] \quad , \quad F_{\text{optimo}}(-2.9, \dots, -2.9) = -78.332$$

Figura 30. Función N-DIMENSIONAL 2D



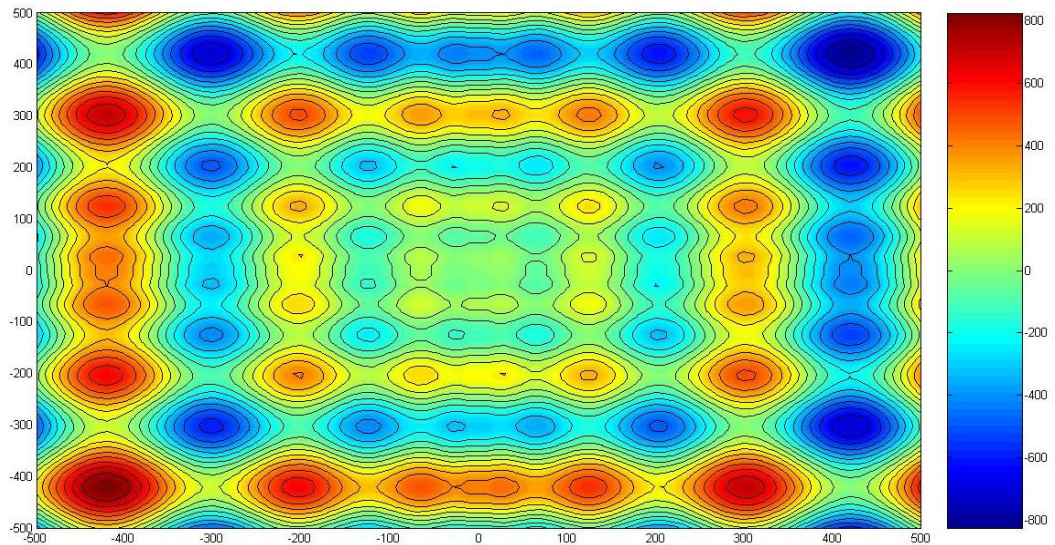
Fuente: Los autores

➤ Función SCHWEFEL

$$F(x) = \sum_{i=1}^N -X_i \sin(\sqrt{|X_i|}) \quad (4.13)$$

Limites $[-500,500]$, $F_x(420.968, \dots, 420.968) = -2094.914$

Figura 31. Función Schwefel 2D



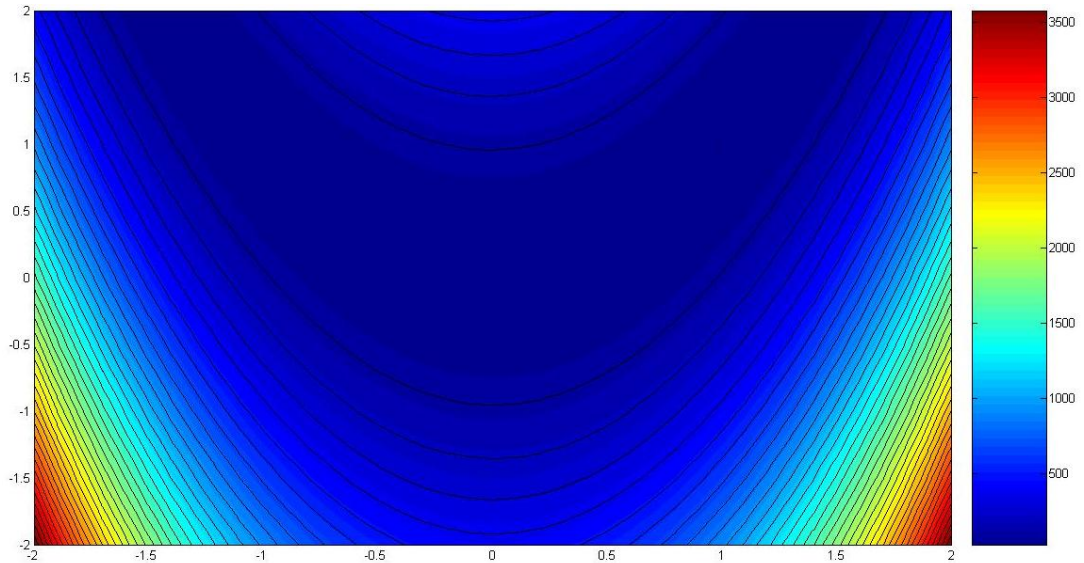
Fuente: Los autores

➤ Función ROSENBROCK

$$F(x) = \sum_{i=1}^{N-1} [100(X_{i+1} - X_i^2)^2 + (X_i - 1)^2] \quad (4.14)$$

$$\text{Limites } [-2,2] \quad , \quad F_{\text{optimo}}(1, \dots, 1) = 0$$

Figura 32. Función Rosenbrock 2D



Fuente: Los autores

4.7.1. Resultados

Para realizar el proceso de evaluación del algoritmo se realizaron 100 pruebas a cada función en diferentes dimensiones y con 20, 50, 100 y 200 partículas respectivamente. Los parámetros del algoritmo y el criterio de convergencia se describen en la sección 4.3 y 4.5 respectivamente.

Además teniendo en cuenta la naturaleza estocástica del algoritmo se realizaron dos tipos de distribuciones iniciales de las partículas como ya se mencionó anteriormente, la primera será una distribución aleatoria y la segunda se realizara por medio del Latin Hypercube Sampling (LHS).

A continuación podemos apreciar los resultados obtenidos mediante la aplicación del algoritmo PSO a las respectivas funciones matemáticas en diferentes dimensiones (ver tablas 2 a 6), resaltando que la función objetivo considerada debe ser la minimización. En la ejecución del programa para cada función se registró el promedio de la función objetivo, el promedio de iteraciones y el promedio del número de evaluaciones realizadas a la función matemática, además se consideraron otros parámetros tales como la desviación estándar y la confiabilidad del algoritmo teniendo en cuenta un error absoluto menor del 5%.

4.7.1.1. Función VENTER

A continuación, en la tabla 2, se presentan los resultados obtenidos mediante la aplicación del algoritmo a la función Venter en 2 dimensiones (2D), la cual se puede observar en la figura 29.

Tabla 2. Resultados función Venter 2D

VENTER 2D		
PARTICULAS	20	
DISTRIBUCION	ALEAT.	LHS
VALOR OPTIMO PROMEDIO DE LA F.O.	1000.001	1000.032
DESVIACION ESTANDAR	9.55E-03	0.196
CONFIABILIDAD	99%	95%
VALOR PROMEDIO DE ITERACIONES	73	71
VALOR PROMEDIO DE EVALUACIONES DE LA FUNCION	1460	1420
VALOR OPTIMO = 1000		
*ALEAT = DISTRIBUCION ALEATORIA *LHS = DISTRIBUCION POR LATIN HYPERCUBE SAMPLING		

Fuente: Los autores

De acuerdo con los resultados obtenidos de la aplicación del algoritmo a la función Venter 2D que se encuentran registrados en la tabla 2, podemos apreciar claramente que el algoritmo es capaz de resolver este problema en dos dimensiones con gran precisión, requiriendo solamente un enjambre de 20 partículas para generar una buena confiabilidad, resaltando que el algoritmo tubo un mejor desempeño con la distribución aleatoria generando una confiabilidad del 99%, comparada con la confiabilidad del 95% que resulto de la distribución por medio del Latin Hypercube Sampling.

4.7.1.2. Función N-DIMENSIONAL (5D)

En la figura 30 se observa el correspondiente grafico de contorno en dos dimensiones de la función. En la tabla 3 se presentan los resultados obtenidos mediante la aplicación del algoritmo a la función N-Dimensional en 5 dimensiones (5D).

Conforme con los resultados obtenidos de la aplicación del algoritmo a la función N-Dimensional 5D que se encuentran registrados en la tabla 3, podemos observar como la capacidad del algoritmo para resolver este problema en cinco dimensiones depende principalmente del tamaño del enjambre. Para un tamaño de enjambre grande se realiza un mejor muestreo mejorando su confiabilidad y por lo tanto aumentando el coste computacional. Por otro lado, el desempeño del algoritmo con respecto al tipo de

distribución inicial no posee alguna influencia notable en los resultados ya que ambos tipos de distribución tanto la aleatoria como la del Latin Hypercube Sampling nos generan el valor promedio de la función objetivo similar y ninguna es dominante con referencia a la confiabilidad.

Tabla 3. Resultados función N-Dimensional 5D

N-DIMENSIONAL 5D										
PARTICULAS	20		50		100		200		2000	
DISTRIBUCION	ALEAT.	LHS	ALEAT.	LHS	ALEAT.	LHS	ALEAT.	LHS	ALEAT.	LHS
VALOR OPTIMO PROMEDIO DE LA F.O.	-71.83	-70.70	-73.24	-73.64	-75.39	-75.62	-77.09	-76.75	-78.33	-78.28
CONFIABILIDAD	26%	21%	34%	42%	55%	58%	78%	73%	100%	99%
DESVIACION ESTANDAR	5.04	5.36	4.44	4.76	3.55	3.46	2.35	2.68	1.99E-9	0.57
VALOR PROMEDIO DE ITERACIONES	81	80	76	75	73	72	70	70	62	62
VALOR PROMEDIO DE EVALUACIONES DE LA F. O.	1620	1600	3800	3750	7300	7200	14000	14000	124000	124000
VALOR OPTIMO = -78.332	*ALEAT = DISTRIBUCION ALEATORIA *LHS = DISTRIBUCION POR LATIN HYPERCUBE SAMPLING									

Fuente: Los autores

4.7.1.3. Función SCHWEFEL (5D)

En la figura 31 se observa el correspondiente grafico de contorno en dos dimensiones de la función Schwefel, donde se puede observar la gran dificultad que presenta ya que posee muchos mínimos locales.

Los resultados obtenidos de la aplicación del algoritmo a la función Schwefel 5D se encuentran registrados en la tabla 4, en la que se puede observar como la capacidad del algoritmo para resolver este problema en cinco dimensiones es muy limitada comparado con la función N-Dimensional, ya que no basta con aumentar el tamaño del enjambre para obtener buenos resultados, resaltando que el mejor desempeño encontrado se logro con un enjambre de 2000 partículas que genera una confiabilidad hasta el 17% comparada con los resultados de los demás tamaños de enjambres que fue una confiabilidad máxima del 1%, independiente de la clase de distribución inicial utilizada.

A continuación se presentan los resultados obtenidos mediante la aplicación del algoritmo a la función Schwefel en 5 dimensiones (5D):

Tabla 4. Resultados función Schwefel 5D

SCHWEFEL 5D										
PARTICULAS	20		50		100		200		2000	
DISTRIBUCION	ALEAT.	LHS	ALEAT.	LHS	ALEAT.	LHS	ALEAT.	LHS	ALEAT.	LHS
VALOR OPTIMO PROMEDIO DE LA F.O.	-1484.52	-1474.38	-1554.83	-1572.97	-1689.06	-1664.78	-1763.88	-1742.55	-1905.41	-1924.55
CONFIABILIDAD	0%	1%	0%	0%	0%	1%	1%	0%	17%	16%
DESVIACION ESTANDAR	183.03	190.35	176.80	165.02	154.73	151.41	137.19	128.83	128.2	105.23
VALOR PROMEDIO DE ITERACIONES	103	101	95	94	91	90	87	87	75	76
VALOR PROMEDIO DE EVALUACIONES DE LA F. O.	2060	2020	4750	4700	9100	9000	17400	17400	150000	152000
VALOR OPTIMO = -2094.914	*ALEAT = DISTRIBUCION ALEATORIA *LHS = DISTRIBUCION POR LATIN HYPERCUBE SAMPLING									

Fuente: Los autores

4.7.1.4. Función ROSENBROCK (5D)

Los resultados obtenidos de la aplicación del PSO a esta función en 5 dimensiones son:

Tabla 5. Resultados función Rosenbrock 5D

ROSENBROCK 5D										
PARTICULAS	20		50		100		200		2000	
DISTRIBUCION	ALEAT.	LHS	ALEAT.	LHS	ALEAT.	LHS	ALEAT.	LHS	ALEAT.	LHS
VALOR OPTIMO PROMEDIO DE LA F.O.	1.42	1.47	0.39	0.64	0.28	0.49	0.32	0.16	0.08	0.08
CONFIABILIDAD	0%	0%	3%	2%	28%	23%	75%	76%	98%	98%
DESVIACION ESTANDAR	1.22	1.22	0.89	1.22	0.87	1.18	1.04	0.67	0.55	0.55
VALOR PROMEDIO DE ITERACIONES	409	390	1185	1090	1384	1235	1376	1380	584	616
VALOR PROMEDIO DE EVALUACIONES DE LA F. O.	8180	7800	59250	54500	138400	123500	275200	276000	1168000	1232000
VALOR OPTIMO = 0	*ALEAT = DISTRIBUCION ALEATORIA *LHS = DISTRIBUCION POR LATIN HYPERCUBE SAMPLING									

Fuente: Los autores

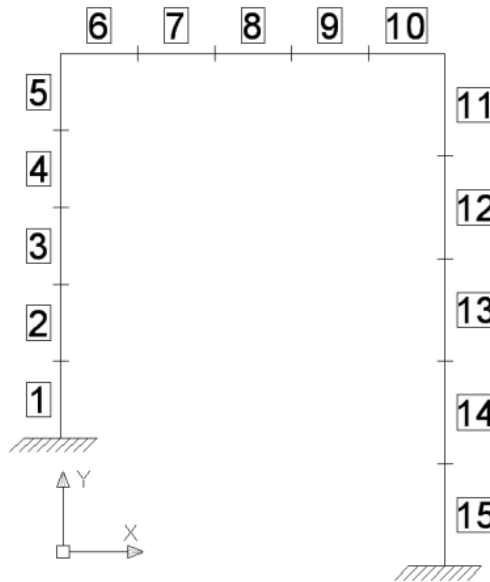
De acuerdo con los resultados obtenidos de la aplicación del algoritmo a la función Rosenbrock 5D que se encuentran registrados en la tabla 5, podemos observar como con el aumento del tamaño del enjambre, mejora la capacidad del algoritmo para resolver este problema en cinco dimensiones, donde para un tamaño de enjambre grande se realiza un mejor muestreo mejorando su confiabilidad y por lo tanto aumentando el coste computacional. Por otro lado, el desempeño del algoritmo con respecto al tipo de distribución inicial no posee alguna influencia notable en los resultados ya que ambos tipos de distribución tanto la aleatoria como la del Latin Hypercube Sampling nos generan valores de confiabilidad muy similares.

Por otro lado en ensayos realizados no reportados aquí, con funciones en 20 dimensiones la confiabilidad obtenida fue nula. Esto se debe a que a mayor número de dimensiones de la función, aumenta la complejidad del problema, generando una mayor inestabilidad del algoritmo con resultados de baja confiabilidad.

5. IMPLEMENTACIÓN DEL ALGORITMO PSO PARA LA DETECCIÓN DE DAÑO

En este trabajo se presenta la evaluación de desempeño del algoritmo PSO en la detección de daños en un pórtico 2D, descrito en la sección 3.1, el cual fue dividido en 15 elementos finitos, como se puede observar en la figura 33.

Figura 33. Pórtico para la detección de daños



Fuente: Los autores

Basándose en la descripción teórica del capítulo 3, la modelación del pórtico en estudio se realizó por medio rutina creada en MATLAB[®] (ver Anexo C), en la cual, a partir de las propiedades geométricas y de los materiales de la estructura se realiza el ensamble tanto para la matriz de rigidez como para la matriz de masa de la estructura con el fin de poder extraer propiedades dinámicas como frecuencias y modos de vibración, la validación del modelo se efectuó por medio del programa de modelación de elementos finitos ANSYS[®] en el cual se considero el primer modo y la primera frecuencia (ver Anexo D), por otro lado en MATLAB[®] se verificaron las propiedades de ortogonalidad.

Como este trabajo es una implementación numérica y no experimental, se dividirá en dos etapas. La primera, es la definición y simulación de daño en el pórtico con su respectiva modelación empleando la rutina propuesta en MATLAB[®]. La segunda etapa comprende la selección de la función objetivo a emplear, para que el algoritmo aproxime el daño presente en la estructura. A continuación se dan a conocer los aspectos más importantes para cada etapa.

5.1. SIMULACION DEL DAÑO

Como se pudo observar anteriormente en la sección 3.4, la determinación de frecuencias y modos de vibración de una estructura depende de la masa y la rigidez del sistema. En

caso de presentarse daño en la estructura, asumiendo que la matriz de masa es constante, entonces el daño se presentaría con un cambio en la matriz de rigidez, por lo tanto la matriz de rigidez dañada de un elemento i estará dado por:

$$[K]_{di} = \alpha_i [K]_i$$

Donde:

$[K]_{di}$ = Matriz de rigidez dañada del elemento i

α_i = Factor de daño del elemento i

$[K]_i$ = Matriz de rigidez sin daño del elemento i

En el que α_i es la variable que indica la severidad y posición del daño, simulando el cambio en las propiedades del elemento, para este trabajo consideramos la variable de daño como la disminución en el modulo de elasticidad. El daño α_i estará dado por valores entre cero y uno. Un valor bajo valor de α_i indicaría un daño severo, mientras que un valor alto indicaría daños leves.

5.1.1. Escenarios de daño

Para observar el desempeño del PSO en el problema planteado se consideraron dos escenarios de daño. El primer escenario es de daño simple y el segundo es un escenario de daño múltiple. Para cada escenario se generaron 5 casos diferentes.

5.1.1.1. Daño simple

En el escenario de daño simple (D.S.) se seleccionó aleatoriamente un elemento para ser afectado por pérdida de rigidez, de igual forma el factor de daño del elemento fue generado aleatoriamente, esto se realizó para cada uno de los casos de daño. En la tabla 6 se puede observar la configuración de cada caso de daño simple estudiado.

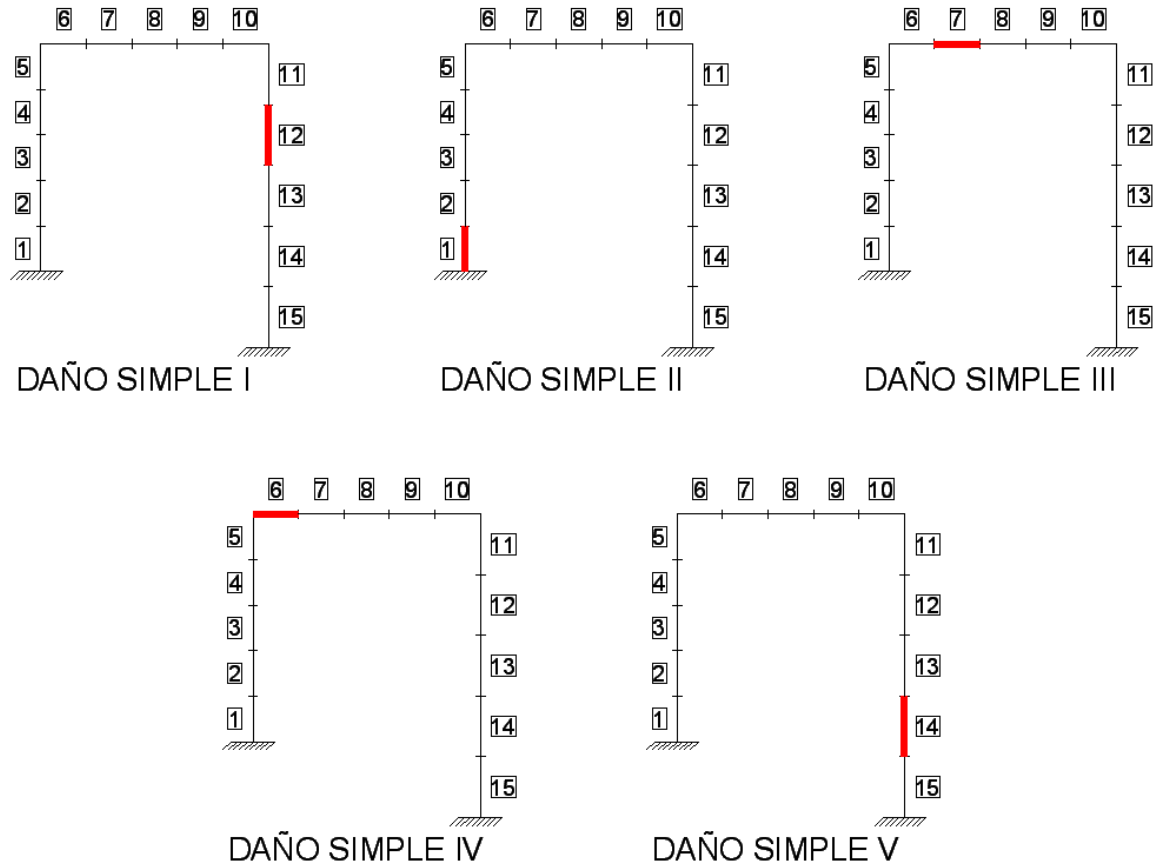
Tabla 6. Casos de daño simple

CASO	ELEMENTO	FACTOR DE DAÑO (α)
DAÑO SIMPLE I	12	0.22
DAÑO SIMPLE II	1	0.85
DAÑO SIMPLE III	7	0.64
DAÑO SIMPLE IV	6	0.51
DAÑO SIMPLE V	14	0.9

Fuente: Los autores

A continuación se realiza un esquema grafico de los resultados obtenidos en la tabla 6, con el fin de dar al lector una mejor comprensión de la ubicación del daño dentro del pórtico a estudiar. De esta forma, para cada caso de daño se realizó su respectiva ubicación dentro del pórtico como se muestra en la figura 34.

Figura 34. Casos de daño simple



Fuente: Los autores

Cada simulación de daño puede representar el deterioro o defectos, que se podrían presentar en algún elemento de la estructura, por ejemplo: presencia de fisuras o defectos en los apoyos. En el caso de fisuras se podría pensar que cada una de las situaciones estudiadas (figura 34) representan este tipo de daño. Por otro lado, el deterioro o los defectos en el se podría considerar en caso de DAÑO SIMPLE II ya que el elemento dañado es que está en contacto con el apoyo.

5.1.1.2. Daño múltiple

En el escenario de daño múltiple (D.M.) a diferencia del anterior, se seleccionaron cuatro elementos aleatoriamente tanto para ser afectados por pérdida de rigidez (en forma simultánea), como para generar el factor de daño para cada elemento. En la tabla 7 se pueden observar para cada caso propuesto los elementos afectados y su respectivo factor de daño.

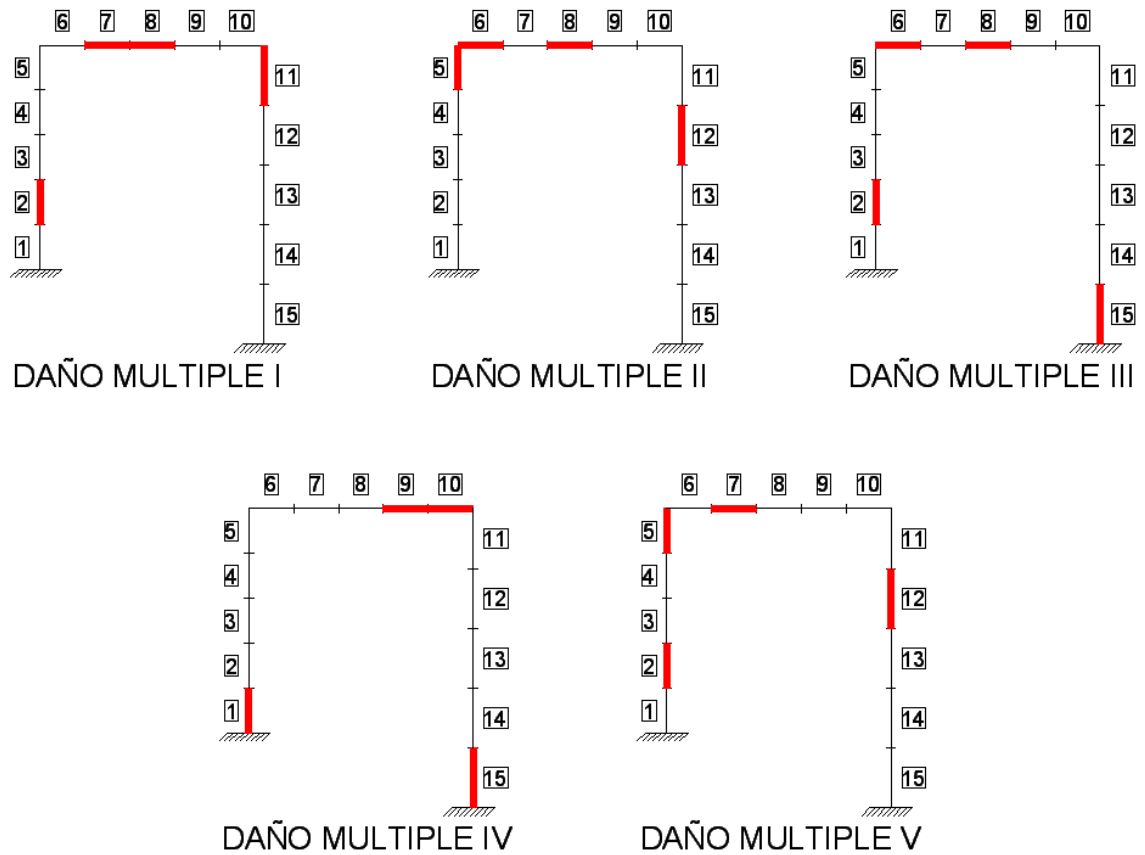
Tabla 7. Casos de daño múltiple

CASO	ELEMENTO	FACTOR DE DAÑO (α)
DAÑO MULTIPLE I	2	0.78
	7	0.69
	8	0.15
	11	0.81
DAÑO MULTIPLE II	5	0.56
	6	0.88
	8	0.90
	12	0.15
DAÑO MULTIPLE III	2	0.65
	6	0.39
	8	0.42
	15	0.75
DAÑO MULTIPLE IV	1	0.37
	9	0.90
	10	0.80
	15	0.78
DAÑO MULTIPLE V	2	0.90
	5	0.80
	7	0.70
	12	0.50

Fuente: Los autores

Al igual que para el daño simple, se presenta un esquema con los resultados de la tabla 10, para así dar al lector una mejor comprensión de la ubicación del daño múltiple dentro del pórtico.

Figura 35. Casos de daño múltiple



Fuente: Los autores

Al igual que en el daño simple, cada simulación puede representar el deterioro o defectos presentes en algunos elementos de la estructura, pero en este escenario de daño múltiple, podemos combinar las consideraciones presentadas en el daño simple, incluyendo defectos en las uniones. Para todos los casos planteados podríamos considerar presencia de grietas en los elementos dañados, por otro lado se puede considerar en el caso de daño múltiple II el defecto en la unión que se podría ser representada por los elementos 5 y 6, además en los casos de daño múltiple III y IV podríamos suponer defectos en los apoyos, para el caso III en el elemento 15 y en el caso IV, en los elementos 1 y 15.

5.2. FUNCIONES OBJETIVO

Para la evaluación la función objetivo se realizó una combinación entre la rutina del algoritmo y la del pórtico con el fin de poder realizar la aplicación. En esta rutina el algoritmo ajusta los factores de daño para cada elemento que compone el pórtico, por lo tanto las funciones consideradas para la implementación se consideran de 15 dimensiones de acuerdo con el pórtico de la figura 33. Inicialmente se realiza la simulación del pórtico dañado para extraer los valores experimentales que quedan fijos a

durante el proceso, mientras los valores analíticos son los calculados mediante la implementación algoritmo y van cambiando en durante la optimización.

Una primera función considerada reportada en la literatura, se basa principalmente en la minimización de cambios de la frecuencia y cambios en los modos de vibración. La función es definida como:

$$f = \min \left[\sum_{i=1}^L \sum_{j=1}^C (\phi_{ij}^e - \phi_{ij}^a)^2 + \sum_{i=1}^L \left| 1 - \frac{\omega_i^e}{\omega_i^a} \right| \right] \quad (5.1)$$

Donde:

L = Número de vectores propios medidos (modos de vibración).

C = Número de elementos de los vectores propios medidos.

ϕ_{ij}^e = Elemento i del vector propio j experimental.

ϕ_{ij}^a = Elemento i del vector propio j analítico.

La segunda función implementada es propuesta por los autores, al igual que la primera esta se basa en cambios de frecuencia y cambios en los modos de vibración, pero utilizando un enfoque diferente, como se muestra a continuación:

$$f_{obj} = \min \left[\sum_{i=1}^L \sqrt{\sum_{j=1}^C (\phi_{ij}^e - \phi_{ij}^a)^2} + \sqrt{\sum_{i=1}^L (\omega_i^e - \omega_i^a)^2} \right] \quad (5.2)$$

Donde:

L = Número de vectores propios medidos (modos de vibración).

C = Número de elementos de los vectores propios medidos.

ϕ_{ij}^e = Elemento i del vector propio j experimental.

ϕ_{ij}^a = Elemento i del vector propio j analítico.

La tercera función implementada se fundamenta en el método del vector de fuerza residual descrita en la sección 2.6, la cual está definida por:

$$\{fr\}_i = [K]_{da} * \{\phi\}_{ei} - \omega_{ei}^2 * [M] * \{\phi\}_{ei} \quad (5.4)$$

Donde:

$\{fr\}_i$ = Vector de fuerza residual del modo i .

$[K]_{da}$ = Matriz de la estructura con daño analítico.

$\{\phi\}_{ei}$ = Modo de vibración i experimental.

ω_{ei}^2 = Frecuencia de vibración al cuadrado del modo i experimental.

$[M]$ = Matriz de masa de la estructura.

Esta función a diferencia de las dos anteriores se basa principalmente en la matriz dañada de la estructura, por lo cual no se hace necesario conocer los modos y

frecuencias del pórtico a partir del daño analítico ajustado por el algoritmo. Cabe resaltar que cada modo proporciona un solo vector de fuerza residual, por lo tanto si consideramos varios modos de vibración nos resultara una matriz $[FR]$, donde cada columna representa el vector de fuerza residual para el respectivo modo de vibración. El enfoque utilizado en este trabajo se basa en la determinación de la norma de dicha matriz, es decir:

$$f_{obj} = \min[\| [FR] \|_2] \quad (5.5)$$

Donde:

$\| [FR] \|_2$ = Norma inducida de orden 2 de la matriz $[FR]$

Dado que cada función objetivo seleccionada para la detección de daños depende fundamentalmente del número de frecuencias y el número de modos, en este trabajo se consideraran solamente las primeras cinco frecuencias con sus respectivos modos de vibración. A continuación se presentan los resultados obtenidos de la implementación del algoritmo PSO a la detección de daños en el pórtico 2D.

5.3. RESULTADOS NUMÉRICOS

Como ya se mencionado anteriormente, la selección de los parámetros y criterio de convergencia del algoritmo PSO utilizados a lo largo del presente trabajo se definen en la sección 4.3 y 4.5 respectivamente.

De acuerdo a la aplicación del algoritmo a las funciones matemáticas multidimensionales se decidió utilizar la distribución inicial aleatoria ya que no se encontraron diferencias significativas con respecto a la distribución por medio del Latin Hypercube Sampling (LHS).

Inicialmente se realizaron pruebas con cada una de las funciones anteriormente establecidas para la detección de daños en el pórtico 2D, y así poder determinar la factibilidad de la aplicación de cada una al problema propuesto, obteniendo como resultado que las dos primeras funciones propuestas: comparación directa de frecuencias y modos de vibración, tanto la reportada en la literatura como la propuesta por los autores, no identificaron correctamente el daño dentro de la estructura. Respecto a la función basada en el vector de fuerza residual presento mejores resultados, por lo tanto esta última función se considero para la implementación en la detección de daños y las dos restantes se descartaron. Cabe resaltar que la función seleccionada es muy sensible a pequeños cambios, de tal forma que puede detectar el daño con bastante precisión pero la función objetivo resultante no es un valor nulo, por lo cual los resultados presentados se basan en el error resultante en cada elemento del pórtico, entre el daño simulado y el daño ajustado por el algoritmo.

Para realizar el proceso de evaluación del algoritmo en la detección de daños, se realizaron 100 pruebas a cada caso de daño con un enjambre inicial de 50 partículas.

A continuación se pueden apreciar los resultados obtenidos mediante la aplicación del algoritmo PSO a los diferentes casos de daños propuestos. En la ejecución del programa se registró la mejor y la peor identificación de daño realizada por el PSO, además de registrar datos estadísticos como lo son: media, desviación estándar y la confiabilidad del algoritmo donde se tuvo en cuenta un error relativo menor del 3%, considerando la

dificultad de la obtención de los datos que se pueden presentar en un modelo experimental propiamente dicho.

5.3.1. DAÑO SIMPLE I

Los resultados de la implementación del PSO al caso de DAÑO SIMPLE I se encuentran en la tabla 8 y en la figura 36, donde se presentan las graficas del porcentaje de error vs. número de iteraciones para cada uno de los 15 elementos que componen el pórtico. En este primer caso, de acuerdo con la tabla 6 y como se puede observar en la figura 34, el elemento 12 tiene un factor de daño de 0.22, lo cual genera una pérdida de rigidez del 78%.

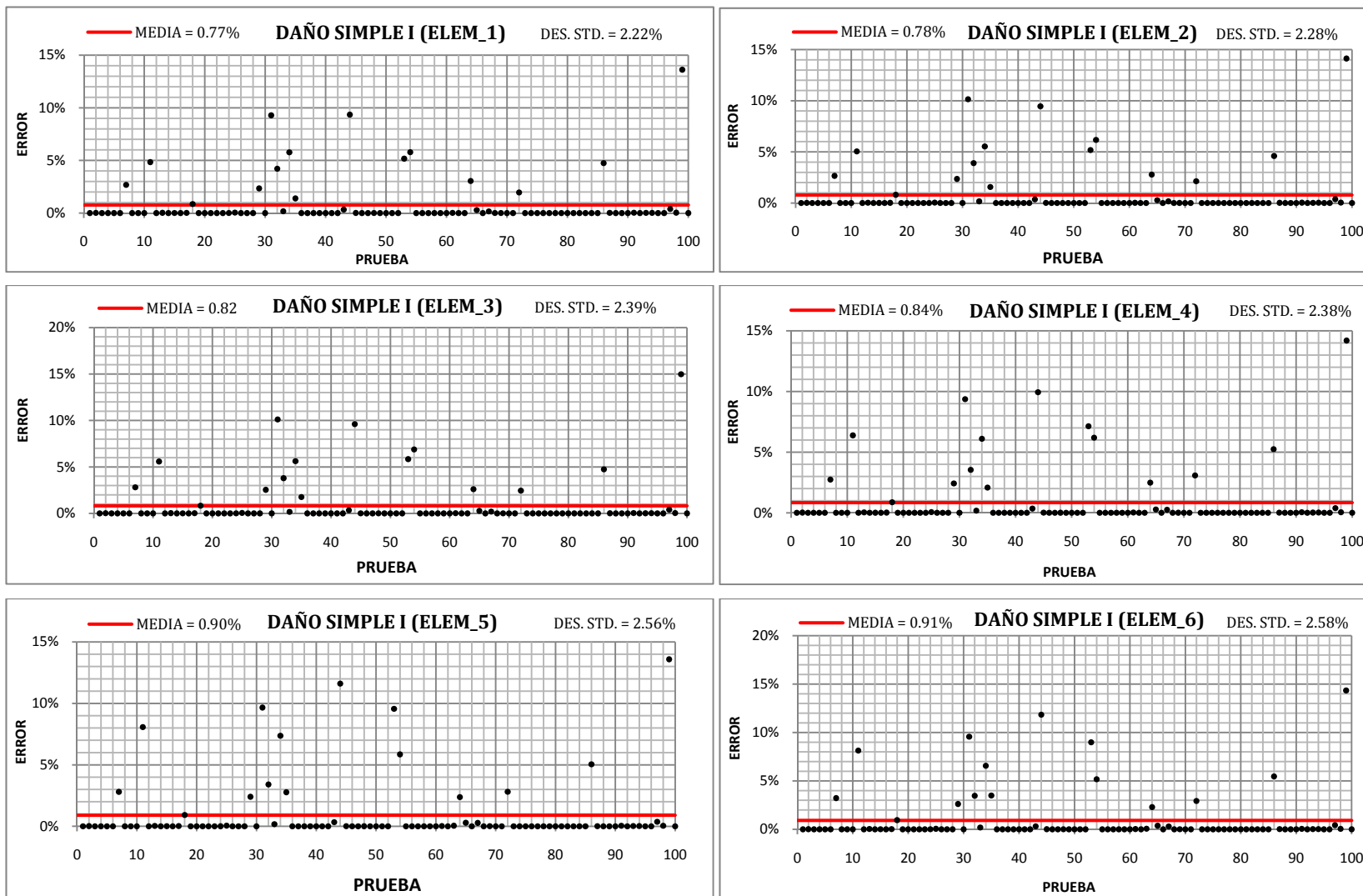
Tabla 8. Resultados DAÑO SIMPLE I

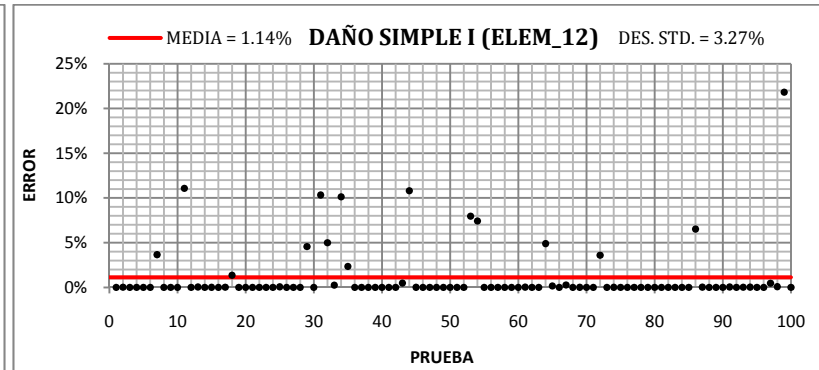
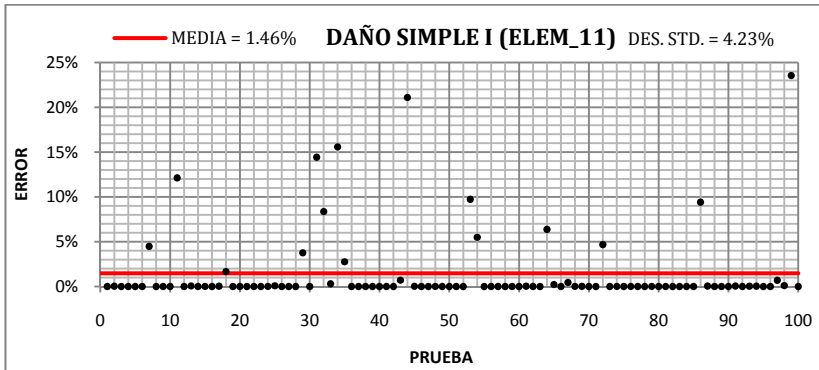
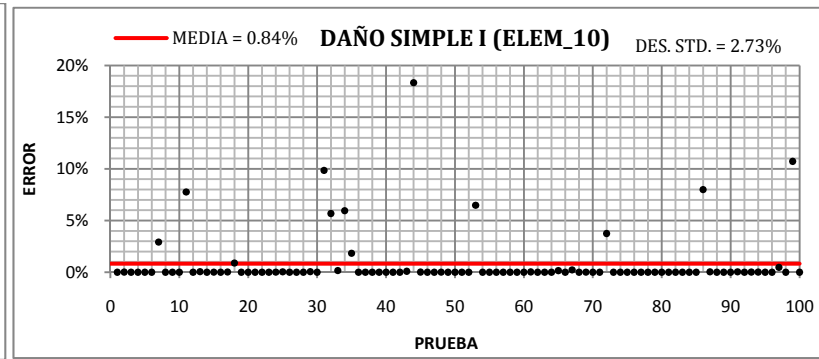
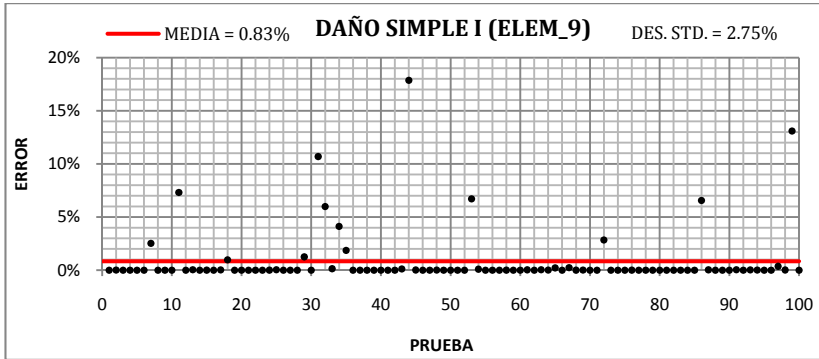
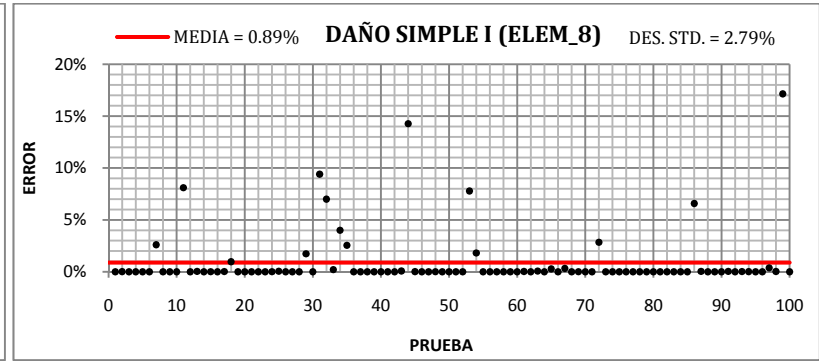
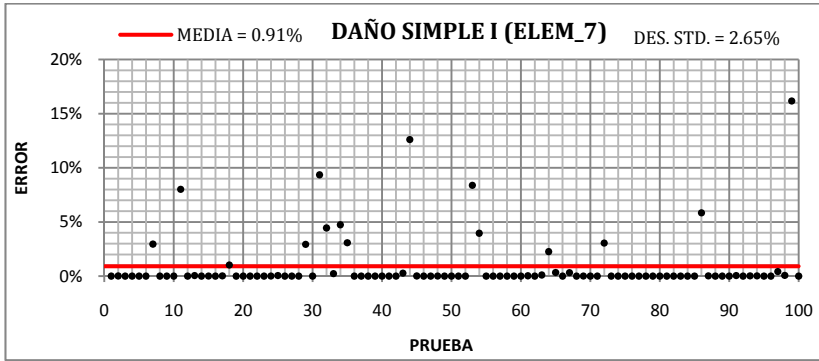
DAÑO SIMPLE I								
ELEMENTO	MEJOR PRUEBA		PEOR PRUEBA		ESTADISTICA			
	F.D.	ERROR	F.D.	ERROR	DES. EST.	MEDIA	CONFIABILIDAD	
1	1	0.00%	0.86382	13.62%	2.22%	0.77%	90%	
2	1	0.00%	0.85871	14.13%	2.28%	0.78%	91%	
3	1	0.00%	0.85023	14.98%	2.39%	0.82%	91%	
4	1	0.00%	0.85802	14.20%	2.38%	0.84%	90%	
5	1	0.00%	0.86409	13.59%	2.56%	0.90%	91%	
6	1	0.00%	0.85658	14.34%	2.58%	0.91%	89%	
7	1	0.00%	0.83833	16.17%	2.65%	0.91%	89%	
8	1	0.00%	0.82858	17.14%	2.79%	0.89%	92%	
9	1	0.00%	0.86906	13.09%	2.75%	0.83%	92%	
10	1	0.00%	0.89277	10.72%	2.73%	0.84%	91%	
11	1	0.00%	0.7646	23.54%	4.23%	1.46%	87%	
12	0.22	0.00%	0.17201	21.81%	3.27%	1.14%	87%	
13	1	0.00%	0.80314	19.69%	3.12%	1.09%	88%	
14	1	0.00%	0.7955	20.45%	3.07%	1.05%	88%	
15	1	0.00%	0.80264	19.74%	2.94%	1.00%	88%	

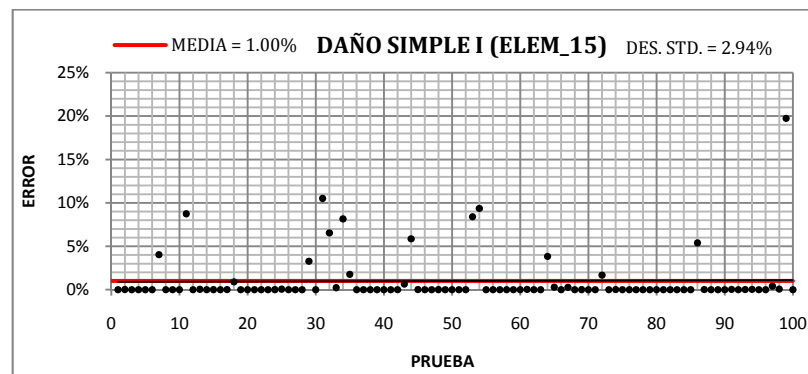
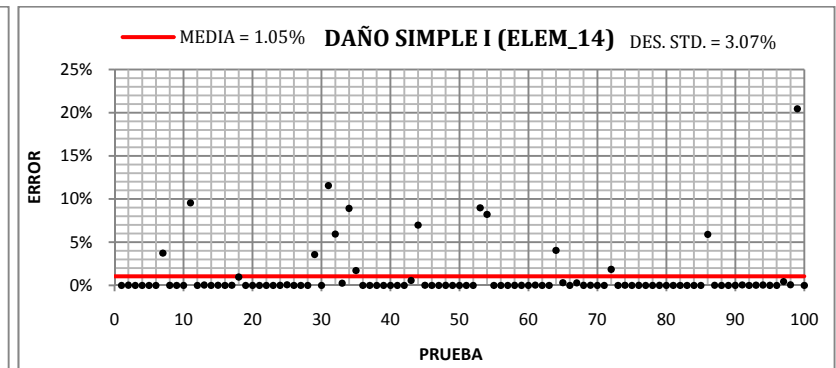
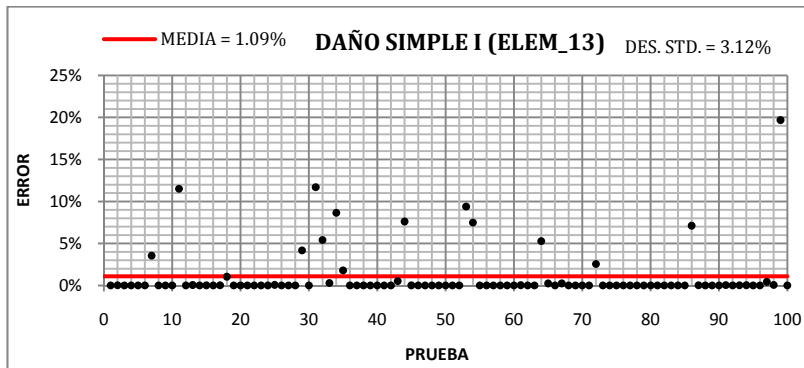
*F.D. = FACTOR DE DAÑO *DES. EST. = DESVIACION ESTANDAR

Fuente: Los autores

Figura 36. Resultados DAÑO SIMPLE I







Fuente: Los autores

En general de acuerdo a los resultados obtenidos, podemos apreciar el buen desempeño del algoritmo, obteniendo un error promedio menor que 1.46% y una desviación estándar menor de 4,23%, generando una confiabilidad entre el 87% y el 90% por lo que podemos considerar al algoritmo estable. Como podemos observar en la figura 36 la mayoría de las pruebas en las que se detecto el daño, se obtuvo un error nulo.

5.3.2. DAÑO SIMPLE II

En este segundo caso, de acuerdo con la tabla 6 y como se puede observar en la figura 34, el elemento 1 tiene un factor de daño de 0.85, lo cual genera una pérdida de rigidez del 15%. En la tabla 9 se presentan los resultados obtenidos.

Tabla 9. Resultados DAÑO SIMPLE II

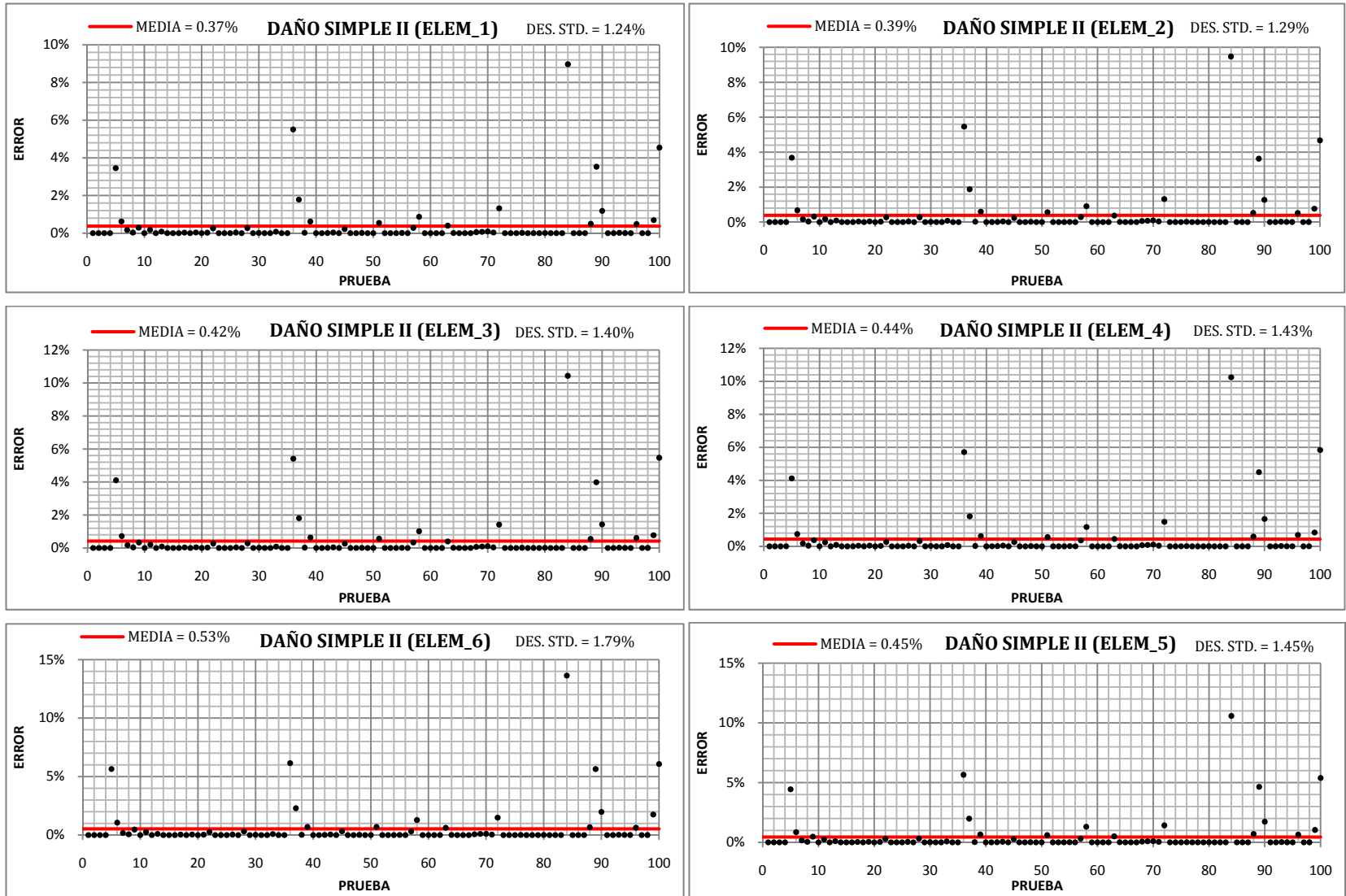
DAÑO SIMPLE II								
ELEMENTO	MEJOR PRUEBA		PEOR PRUEBA		ESTADISTICA			
	F.D.	ERROR	F.D.	ERROR	DES. EST.	MEDIA	CONFIABILIDAD	
1	0.85	0.00%	0.77376	8.97%	1.24%	0.37%	95%	
2	1	0.00%	0.90527	9.47%	1.29%	0.39%	95%	
3	1	0.00%	0.89563	10.44%	1.40%	0.42%	95%	
4	1	0.00%	0.89754	10.25%	1.43%	0.44%	95%	
5	1	0.00%	0.89419	10.58%	1.45%	0.45%	95%	
6	1	0.00%	0.86366	13.63%	1.79%	0.53%	95%	
7	1	0.00%	0.89372	10.63%	1.60%	0.51%	95%	
8	1	0.00%	0.96645	3.36%	1.17%	0.39%	96%	
9	1	0.00%	0.94948	5.05%	1.19%	0.37%	97%	
10	1	0.00%	0.92333	7.67%	1.56%	0.50%	95%	
11	1	0.00%	0.86664	13.34%	1.91%	0.60%	94%	
12	1	0.00%	0.8859	11.41%	1.60%	0.48%	96%	
13	1	0.00%	0.88589	11.41%	1.53%	0.43%	96%	
14	1	0.00%	0.89806	10.19%	1.37%	0.39%	96%	
15	1	0.00%	0.91992	8.01%	1.16%	0.35%	96%	

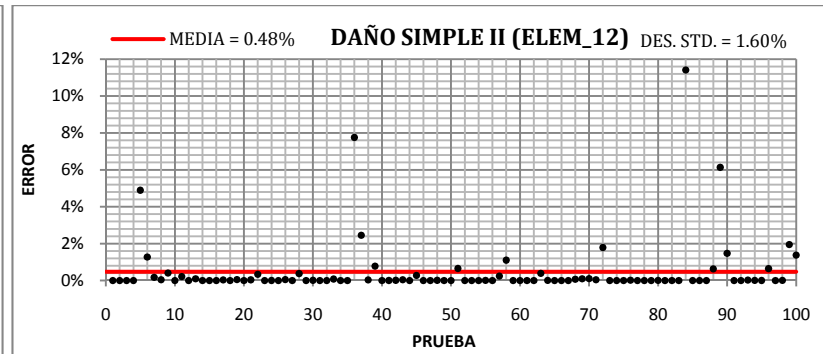
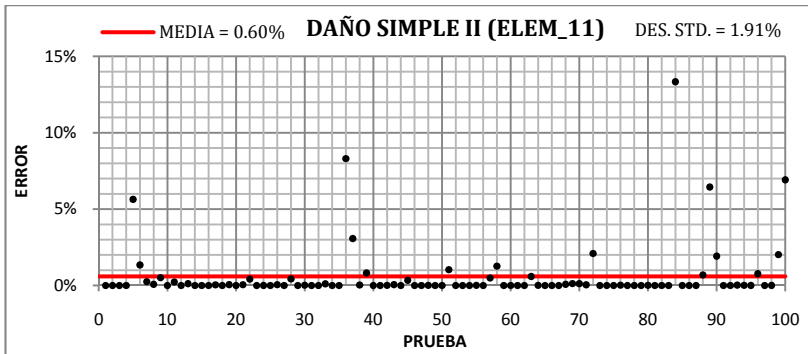
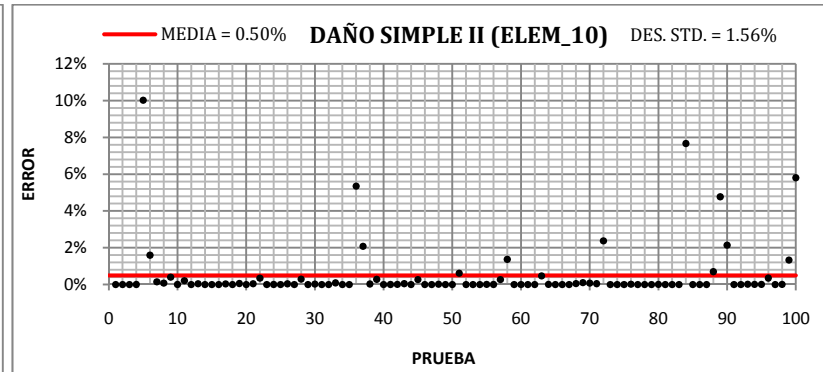
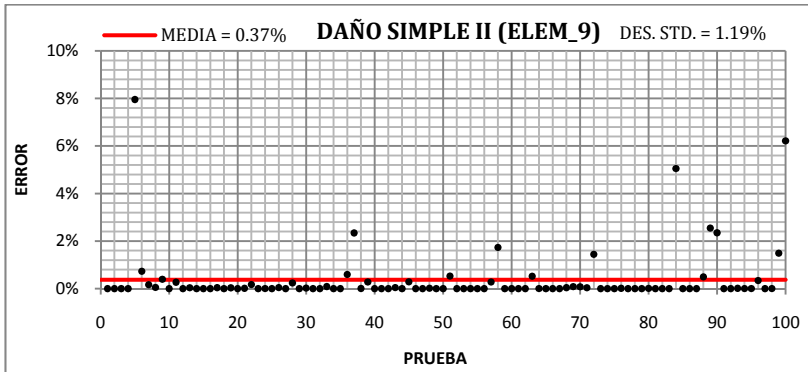
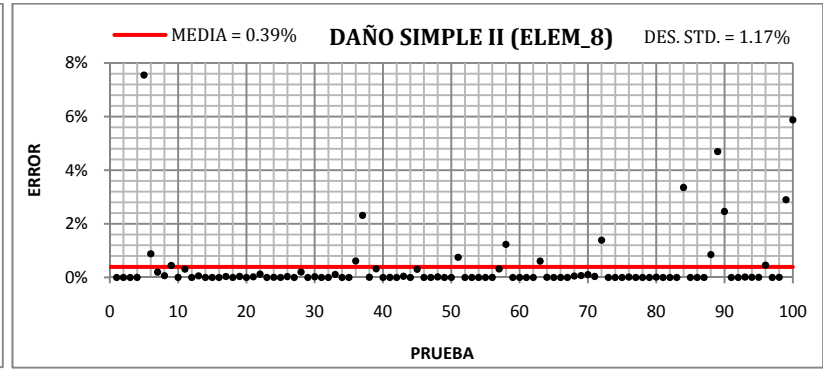
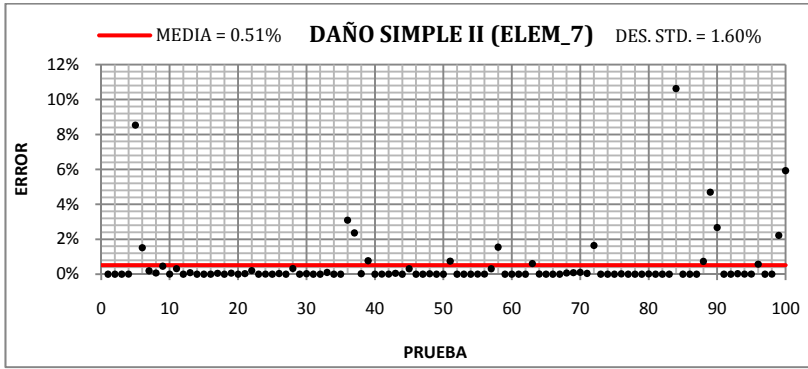
*F.D. = FACTOR DE DAÑO *DES. EST. = DESVIACION ESTANDAR

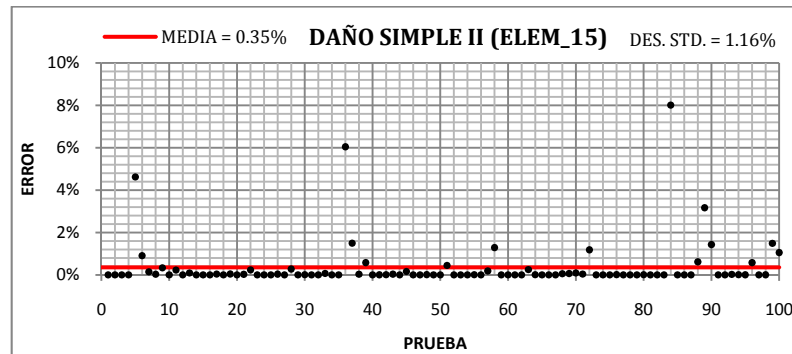
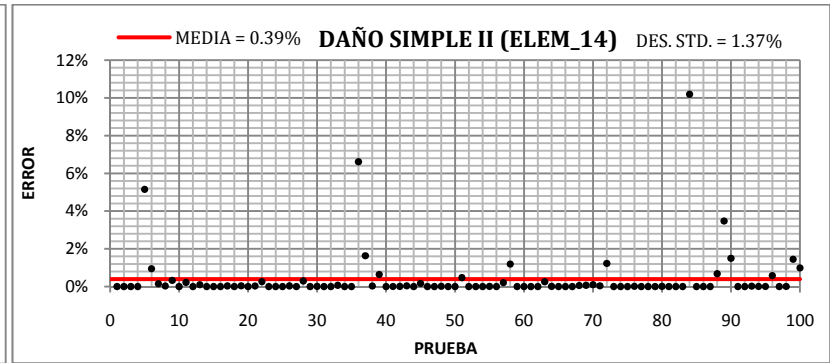
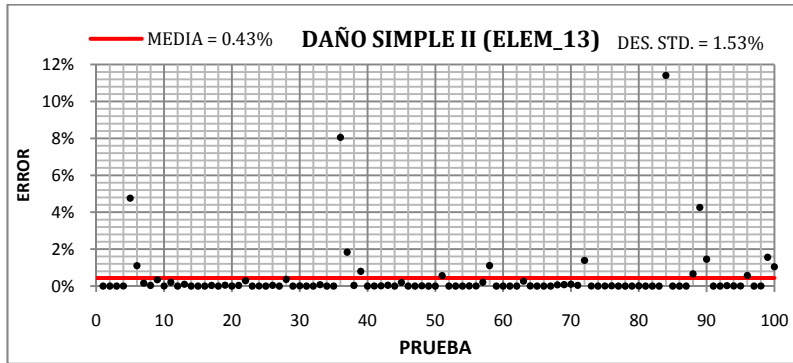
Fuente: Los autores

En este caso de daño el algoritmo tuvo un mejor desempeño, comparado con el primer caso estudiado. En la figura 37 se puede observar el error relativo obtenido de cada uno de los elementos del pórtico en cada prueba realizada.

Figura 37. Resultados DAÑO SIMPLE II







Fuente: Los autores

De acuerdo a los resultados obtenidos, podemos apreciar que en este caso de daño mejoró el desempeño del algoritmo con respecto al anterior caso estudiado, obteniendo un error promedio menor que 0.60% y una desviación estándar menor de 1.91%, generando una confiabilidad entre el 95% y el 96%.

5.3.3. DAÑO SIMPLE III

Los resultados de la implementación del PSO al caso de DAÑO SIMPLE III se encuentran en la tabla 8. En este caso, de acuerdo con la tabla 6 y como se puede observar en la figura 34, el elemento 7 tiene un factor de daño de 0.64, lo cual genera una pérdida de rigidez del 36%.

Tabla 10. Resultados DAÑO SIMPLE III

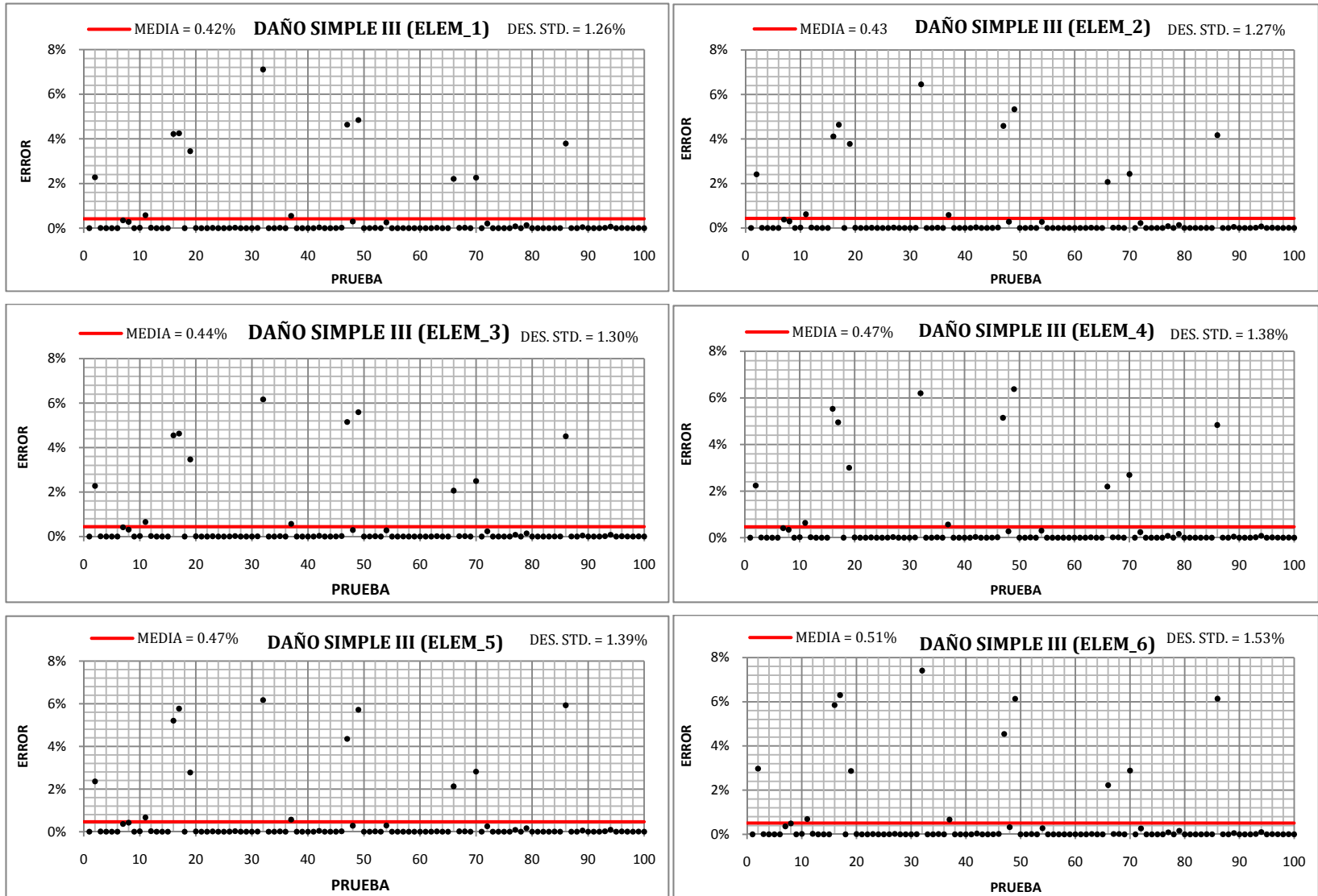
DAÑO SIMPLE III								
ELEMENTO	MEJOR PRUEBA		PEOR PRUEBA		ESTADISTICA			
	F.D.	ERROR	F.D.	ERROR	DES. EST.	MEDIA	CONFIABILIDAD	
1	1	0.00%	0.92892	7.11%	1.26%	0.42%	93%	
2	1	0.00%	0.93547	6.45%	1.27%	0.43%	93%	
3	1	0.00%	0.93836	6.16%	1.30%	0.44%	93%	
4	1	0.00%	0.938	6.20%	1.38%	0.47%	93%	
5	1	0.00%	0.93831	6.17%	1.39%	0.47%	94%	
6	1	0.00%	0.92591	7.41%	1.53%	0.51%	94%	
7	0.64	0.00%	0.59413	7.17%	1.63%	0.53%	93%	
8	1	0.00%	0.92781	7.22%	1.78%	0.54%	93%	
9	1	0.00%	0.93971	6.03%	1.51%	0.46%	92%	
10	1	0.00%	0.94469	5.53%	1.50%	0.49%	91%	
11	1	0.00%	0.93208	6.79%	1.97%	0.67%	91%	
12	1	0.00%	0.94294	5.71%	1.23%	0.44%	92%	
13	1	0.00%	0.94372	5.63%	1.13%	0.40%	93%	
14	1	0.00%	0.94012	5.99%	1.09%	0.38%	93%	
15	1	0.00%	0.9325	6.75%	1.08%	0.36%	95%	

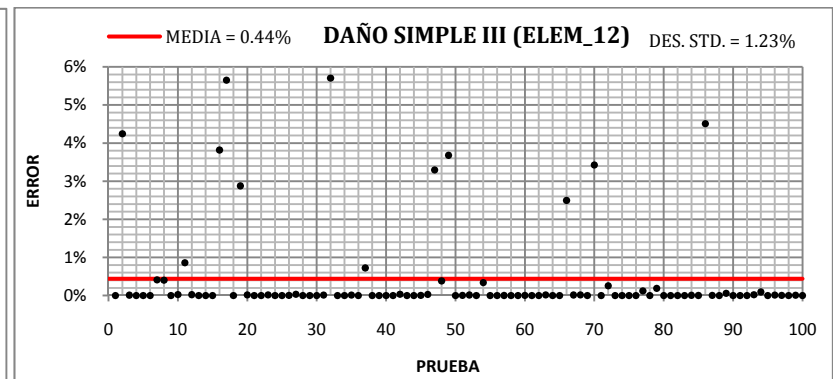
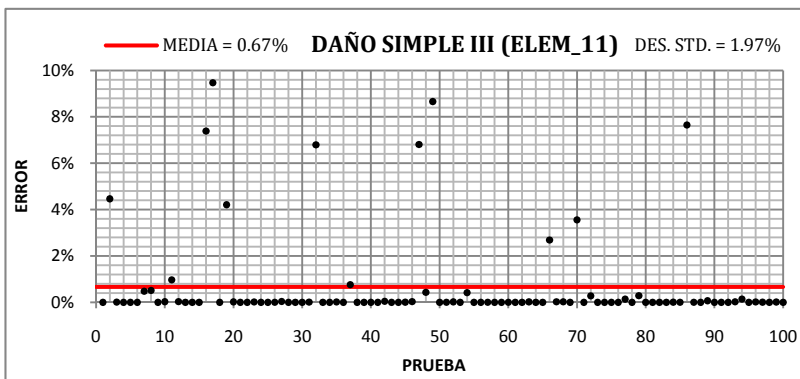
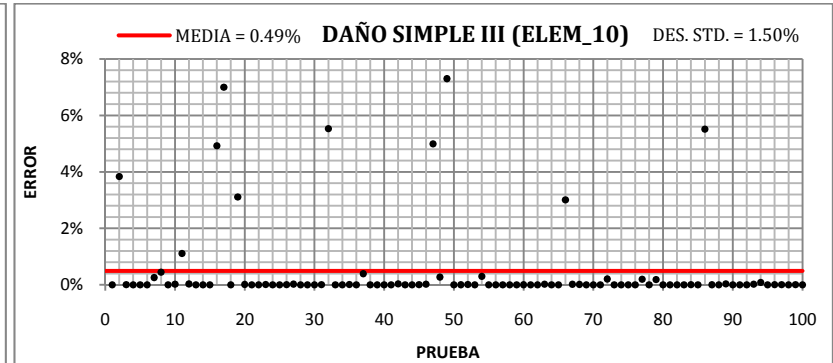
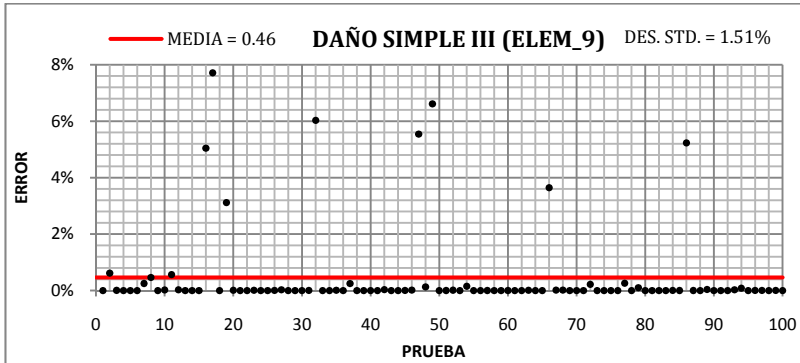
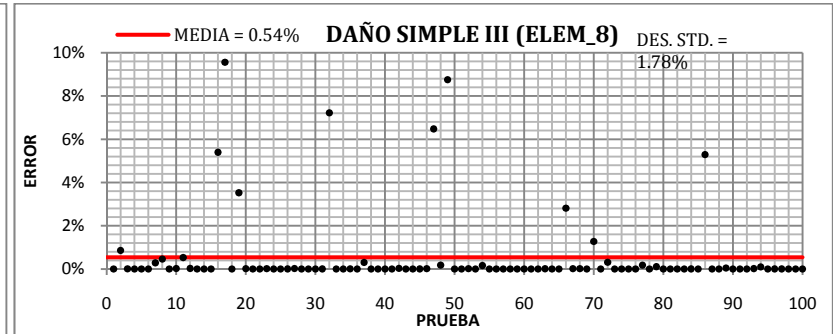
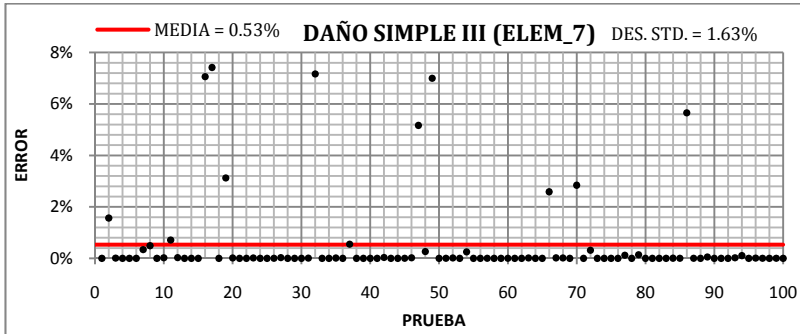
*F.D. = FACTOR DE DAÑO *DES. EST. = DESVIACION ESTANDAR

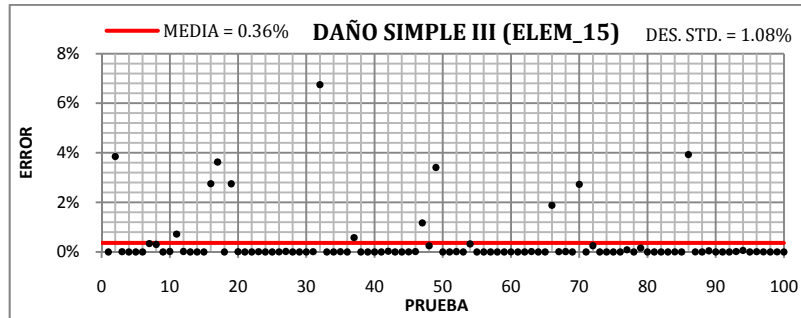
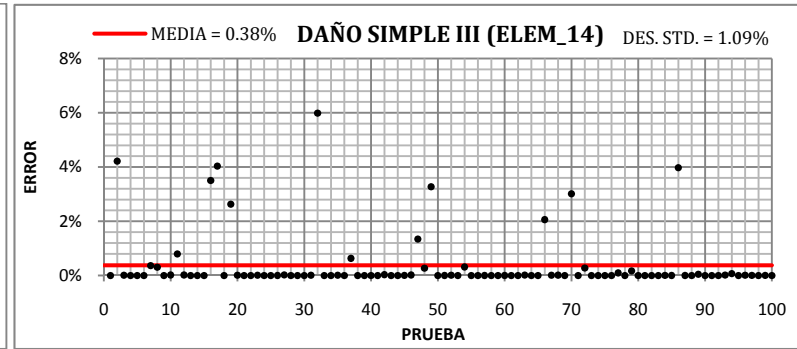
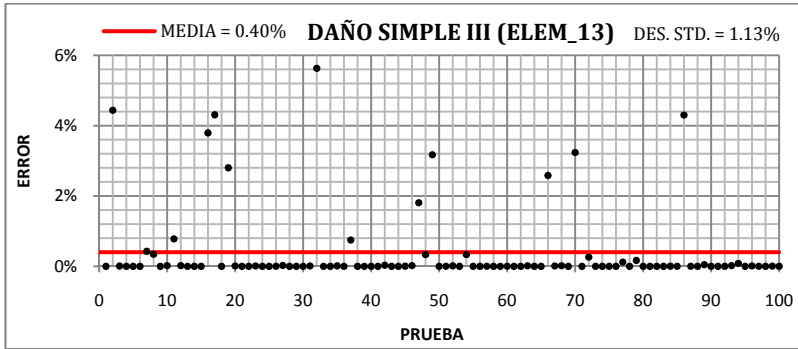
Fuente: Los autores

Al igual que los dos casos anteriores, se encontró un buen desempeño del algoritmo en la detección del daño. A continuación se presentan las graficas de los resultados obtenidos, donde se puede observar el error relativo del ajuste del daño hecho por el algoritmo en cada uno de los 15 elementos que componen el pórtico.

Figura 38. Resultados DAÑO SIMPLE III







Fuente: Los autores

En las graficas anteriores, podemos apreciar el buen desempeño del algoritmo, obteniendo un error promedio menor que 0.59% y una desviación estándar menor de 1,97%, valores muy cercanos a los obtenido en caso de DAÑO SIMPLE II, pero en este caso se tiene una confiabilidad entre el 91% y el 95% por lo que podemos considerar al algoritmo estable.

5.3.4. DAÑO SIMPLE IV

En la tabla 6 y en la figura 34, se puede observar que el elemento 6 tiene un factor de daño de 0.51, lo cual genera una pérdida de rigidez del 49%. En la tabla 11 se presentan los resultados obtenidos.

Tabla 11. Resultados DAÑO SIMPE IV

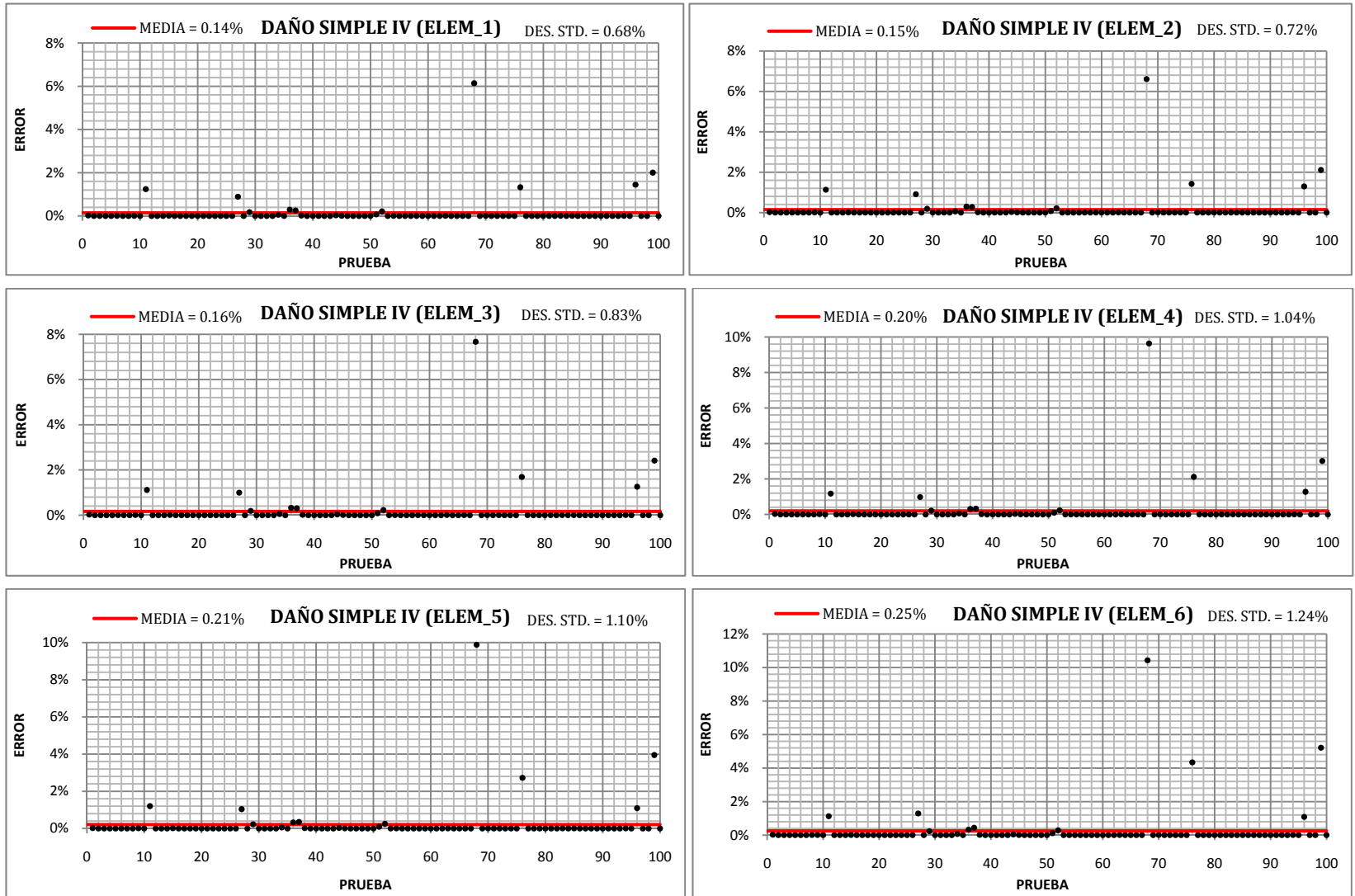
DAÑO SIMPLE IV								
ELEMENTO	MEJOR PRUEBA		PEOR PRUEBA		ESTADISTICA			
	F.D.	ERROR	F.D.	ERROR	DES. EST.	MEDIA	CONFIABILIDAD	
1	1	0.00%	0.93862	6.14%	0.68%	0.14%	99%	
2	1	0.00%	0.93392	6.61%	0.72%	0.15%	99%	
3	1	0.00%	0.92329	7.67%	0.83%	0.16%	99%	
4	1	0.00%	0.90368	9.63%	1.04%	0.20%	98%	
5	1	0.00%	0.9012	9.88%	1.10%	0.21%	98%	
6	0.51	0.00%	0.4568	10.43%	1.24%	0.25%	97%	
7	1	0.00%	0.90021	9.98%	1.16%	0.23%	97%	
8	1	0.00%	0.90645	9.36%	1.06%	0.21%	98%	
9	1	0.00%	0.92243	7.76%	0.87%	0.17%	98%	
10	1	0.00%	0.88821	11.18%	1.17%	0.20%	99%	
11	1	0.00%	0.83404	16.60%	1.74%	0.30%	97%	
12	1	0.00%	0.91748	8.25%	0.93%	0.19%	98%	
13	1	0.00%	0.94343	5.66%	0.70%	0.16%	98%	
14	1	0.00%	0.95441	4.56%	0.60%	0.14%	99%	
15	1	0.00%	0.96096	3.90%	0.54%	0.13%	99%	

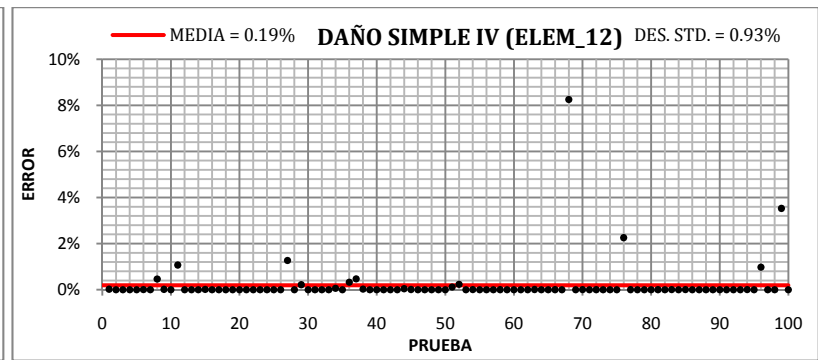
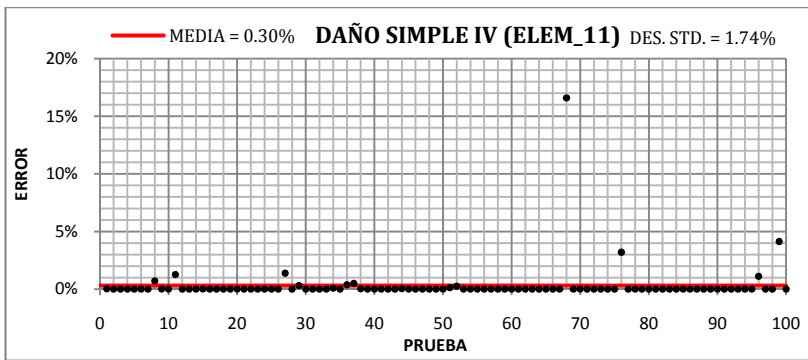
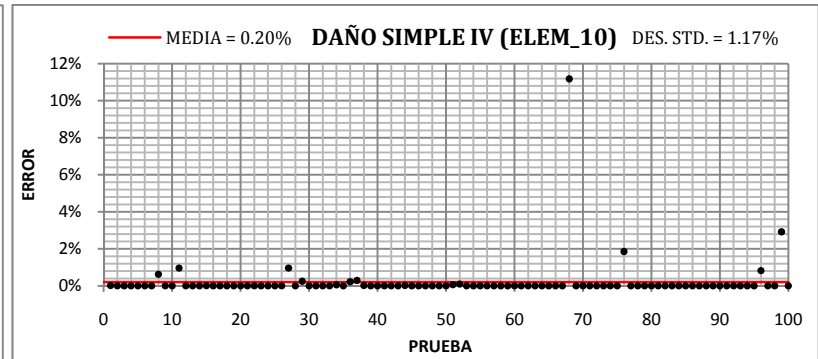
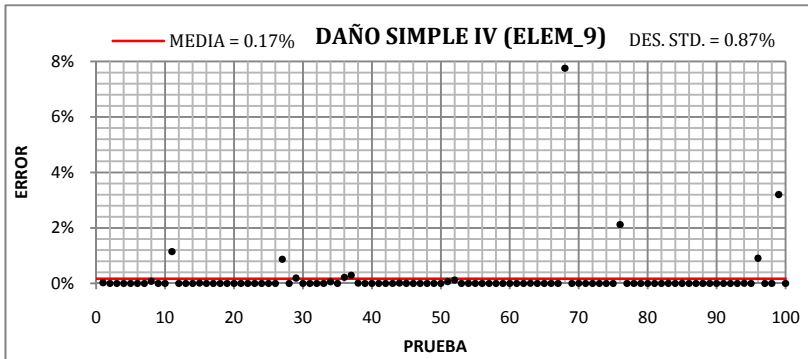
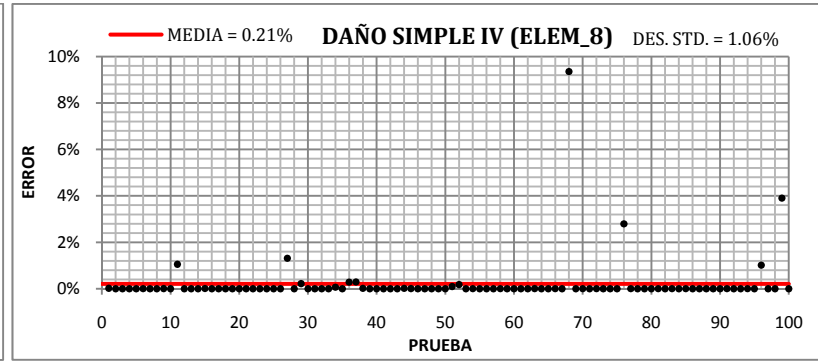
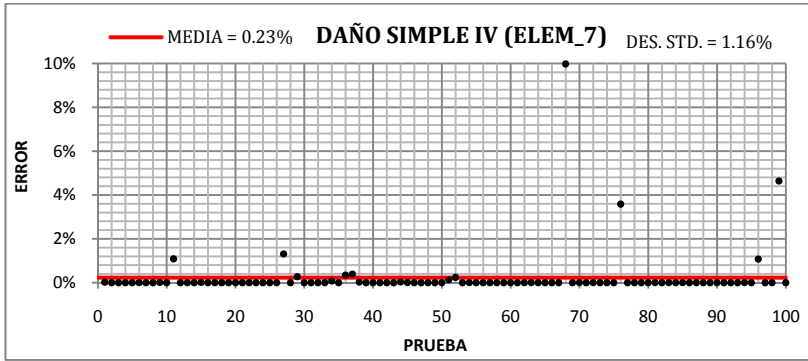
*F.D. = FACTOR DE DAÑO *DES. EST. = DESVIACION ESTANDAR

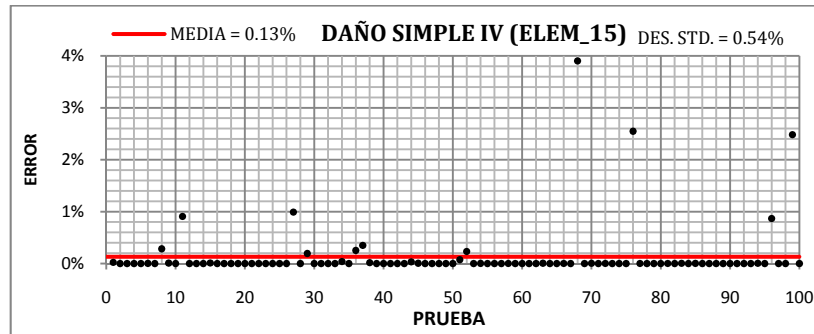
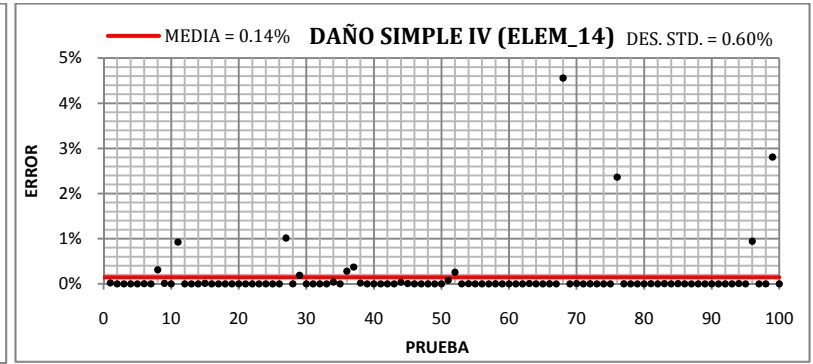
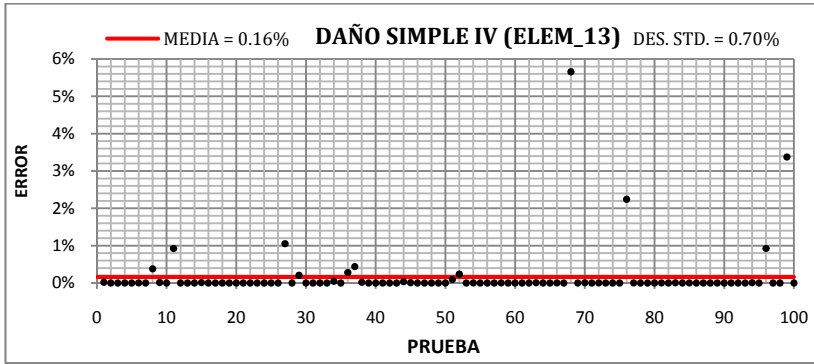
Fuente: Los autores

En este cuarto caso de daño el algoritmo tuvo un mejor desempeño, comparado con los tres casos anteriores. En la figura 37 se puede observar el error relativo obtenido de cada uno de los elementos del pórtico en cada prueba realizada.

Figura 39. Resultados DAÑO SIMPE IV







Fuente: Los autores

De acuerdo a los resultados obtenidos, podemos apreciar que en este caso de daño, mejoró el desempeño del algoritmo, obteniendo un error promedio menor que 0.30% y una desviación estándar menor de 1.74%, generando una confiabilidad entre el 97% y el 99% para la detección correcta del daño dentro de la estructura.

5.3.5. DAÑO SIMPLE V

En la tabla 6 y en la figura 35 se encuentra especificado este último caso de daño simple considerado, donde el elemento 14 tiene un factor de daño de 0.90, el cual da como resultado una pérdida de rigidez del 10%. En la tabla 12 se presentan los resultados de la aplicación del PSO a este caso de daño.

Tabla 12. Resultados DAÑO SIMPLE V

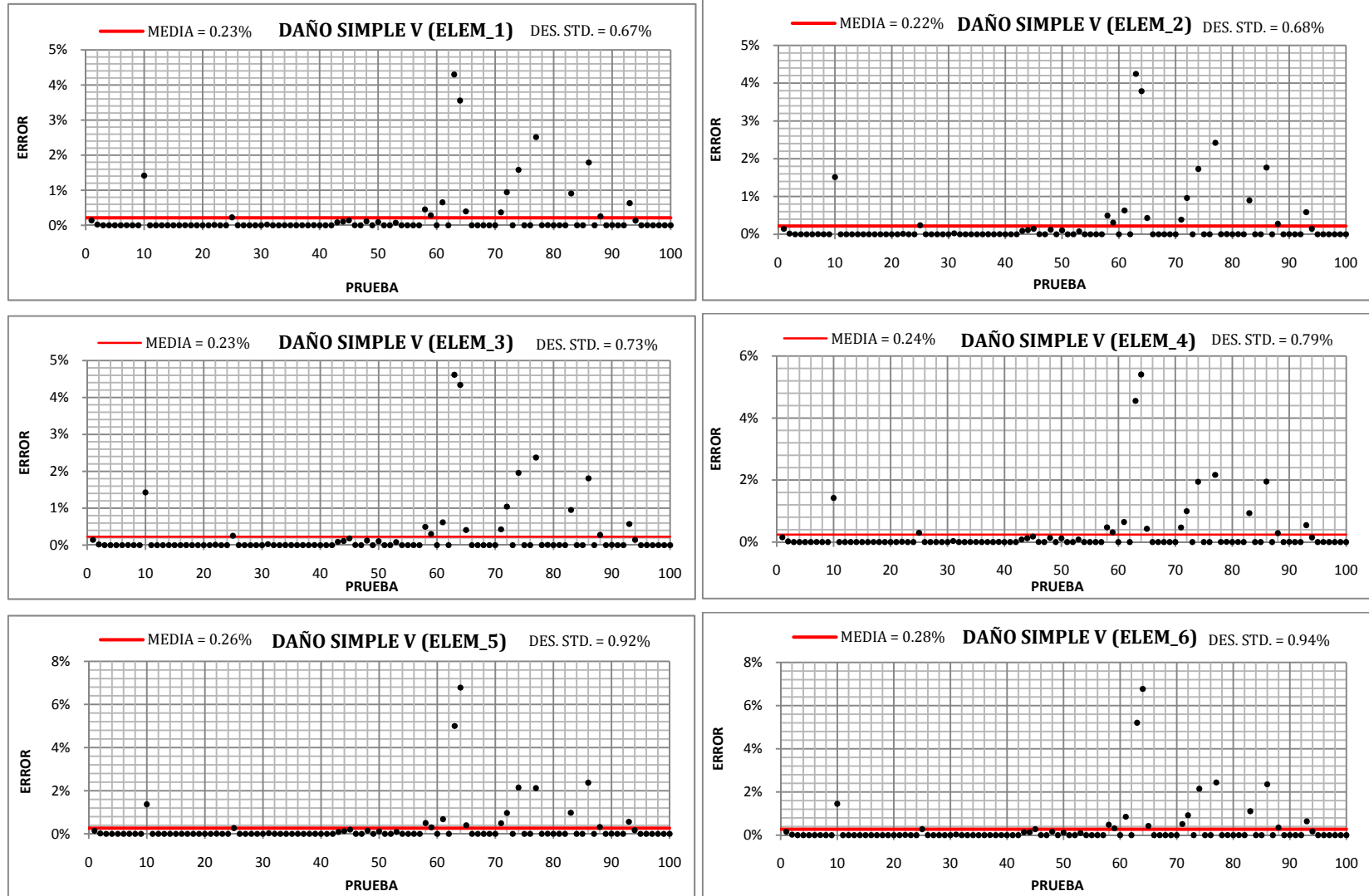
DAÑO SIMPLE V								
ELEMENTO	MEJOR PRUEBA		PEOR PRUEBA		ESTADISTICA			
	F.D.	ERROR	F.D.	ERROR	DES. EST.	MEDIA	CONFIABILIDAD	
1	1	0.00%	0.96444	3.56%	0.67%	0.21%	98%	
2	1	0.00%	0.96212	3.79%	0.68%	0.22%	98%	
3	1	0.00%	0.95662	4.34%	0.73%	0.23%	98%	
4	1	0.00%	0.94596	5.40%	0.79%	0.24%	98%	
5	1	0.00%	0.93218	6.78%	0.92%	0.26%	98%	
6	1	0.00%	0.93229	6.77%	0.94%	0.28%	98%	
7	1	0.00%	0.93473	6.53%	0.90%	0.27%	98%	
8	1	0.00%	0.94115	5.89%	0.81%	0.23%	98%	
9	1	0.00%	0.95376	4.62%	0.69%	0.20%	98%	
10	1	0.00%	0.96077	3.92%	0.65%	0.21%	98%	
11	1	0.00%	0.9314	6.86%	1.16%	0.36%	97%	
12	1	0.00%	0.94388	5.61%	0.88%	0.28%	98%	
13	1	0.00%	0.94408	5.59%	0.84%	0.26%	98%	
14	0.9	0.00%	0.84793	5.79%	0.83%	0.25%	98%	
15	1	0.00%	0.94757	5.24%	0.79%	0.24%	98%	

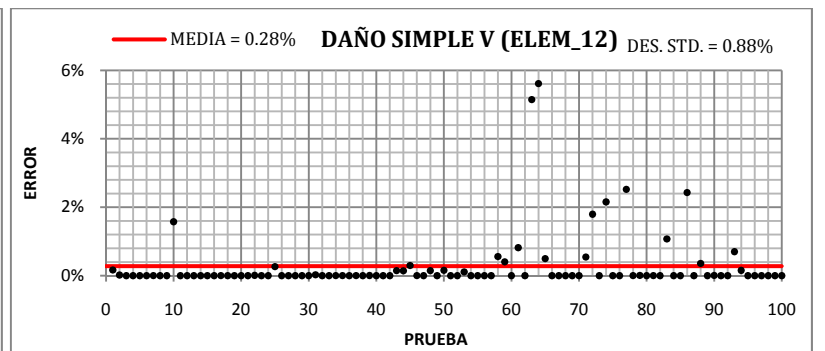
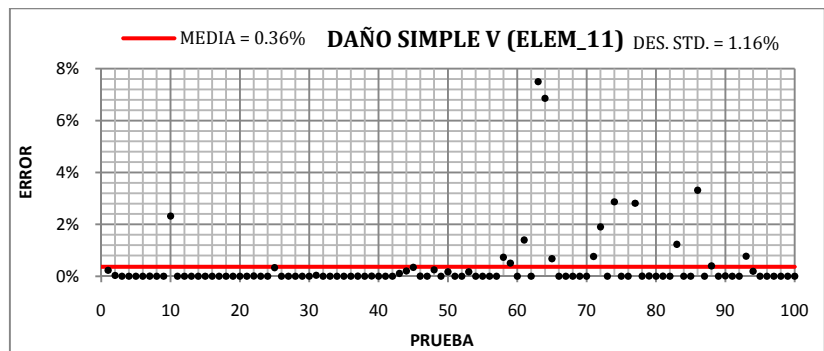
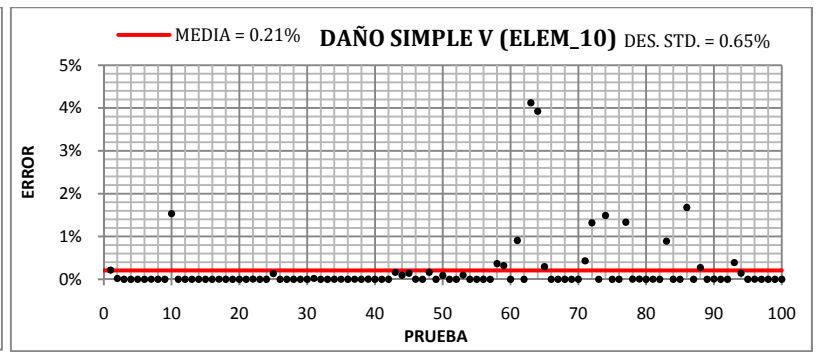
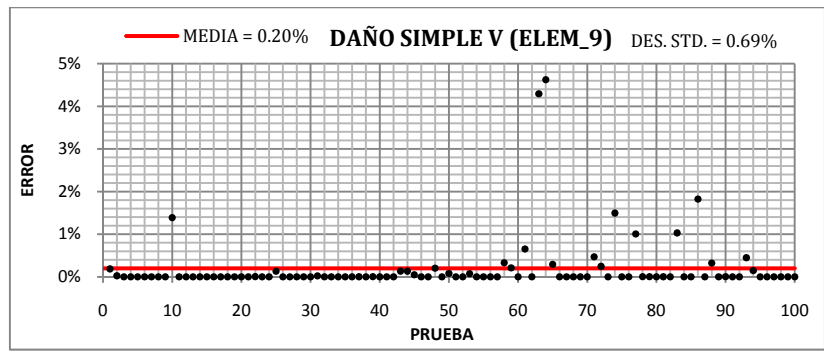
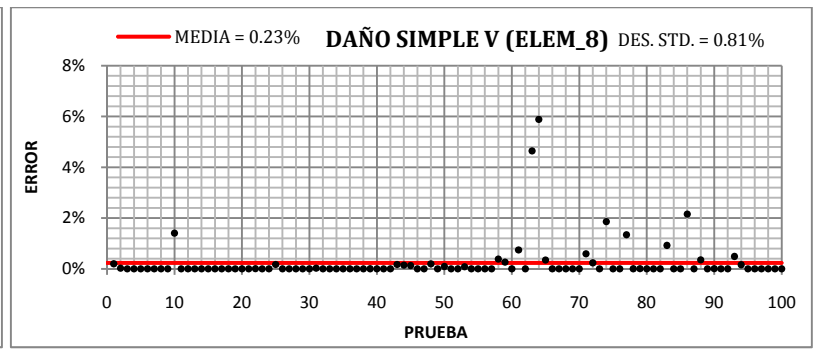
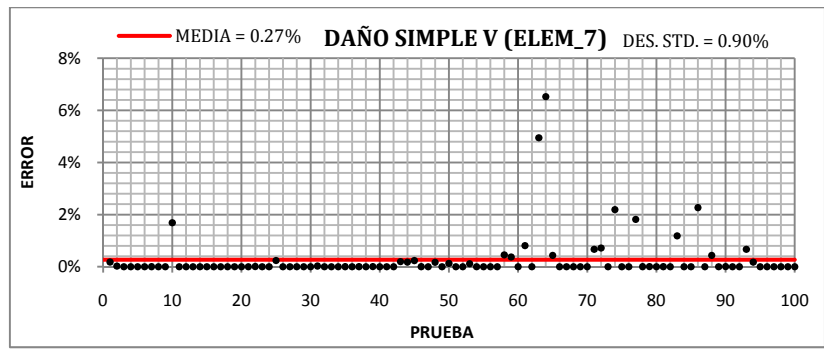
*F.D. = FACTOR DE DAÑO *DES. EST. = DESVIACION ESTANDAR

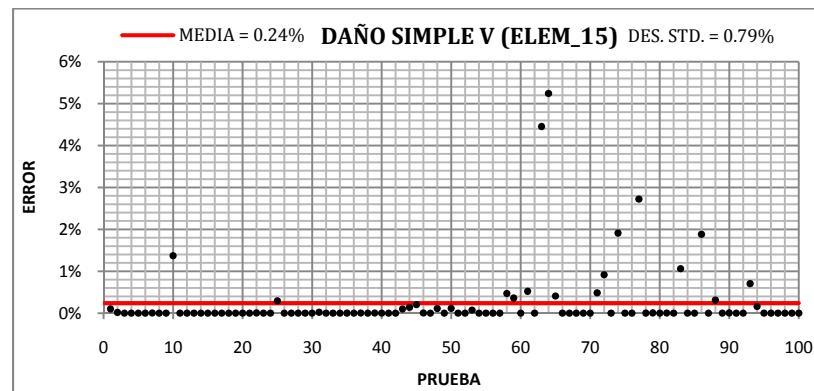
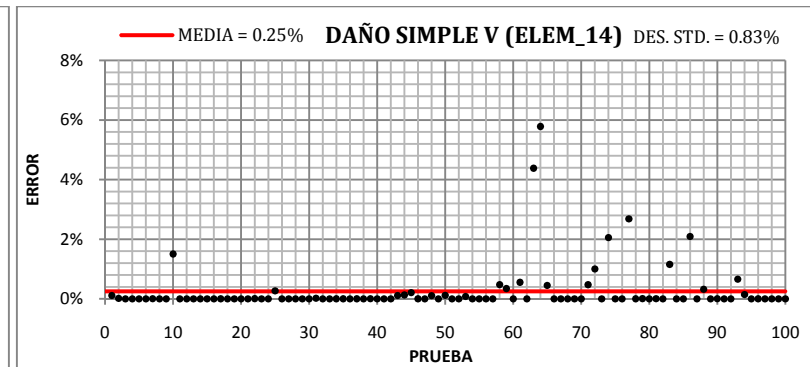
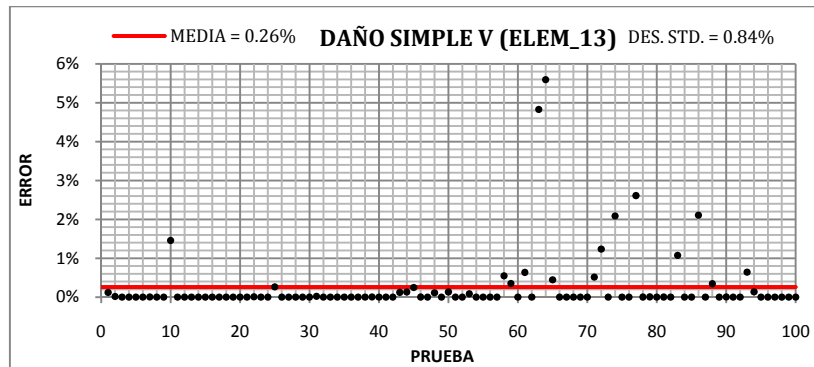
Fuente: Los autores

En la figura 40 se presentan los resultados de las pruebas en cada uno de los elementos del pórtico. En cada grafica se encuentran registrados los errores relativos obtenidos en las respectivas pruebas realizadas.

Figura 40. Resultados DAÑO SIMPLE V







Fuente: Los autores

En este último caso de daño simple se obtuvo un error relativo promedio menor de 0.36% una desviación estándar menor de 1.16% con lo que podemos considerar un buen desempeño del algoritmo, además la confiabilidad que se encuentra entre 97% y 98% nos permiten concluir que el algoritmo es estable.

En general, de los resultados obtenidos de la implementación del algoritmo PSO al caso de daño simple, podemos considerar el buen desempeño que realizó el algoritmo en este tipo de problema planteado, generando una alta confiabilidad con desviación y con un promedio de error bajo. Cabe resaltar que el problema en estudio es de 15 dimensiones.

5.3.6. DAÑO MULTIPLE I

Los resultados de la implementación del PSO al caso de DAÑO MULTIPLE I se encuentran en la tabla 13. En este primer caso, de acuerdo con la tabla 7 y como se puede observar en la figura 35, los elementos 2, 7, 8 y 11 tienen un factor de daño, lo cual genera una pérdida de rigidez del 22%, 31%, 85% y 19% respectivamente.

Tabla 13. Resultados DAÑO MULTIPLE I

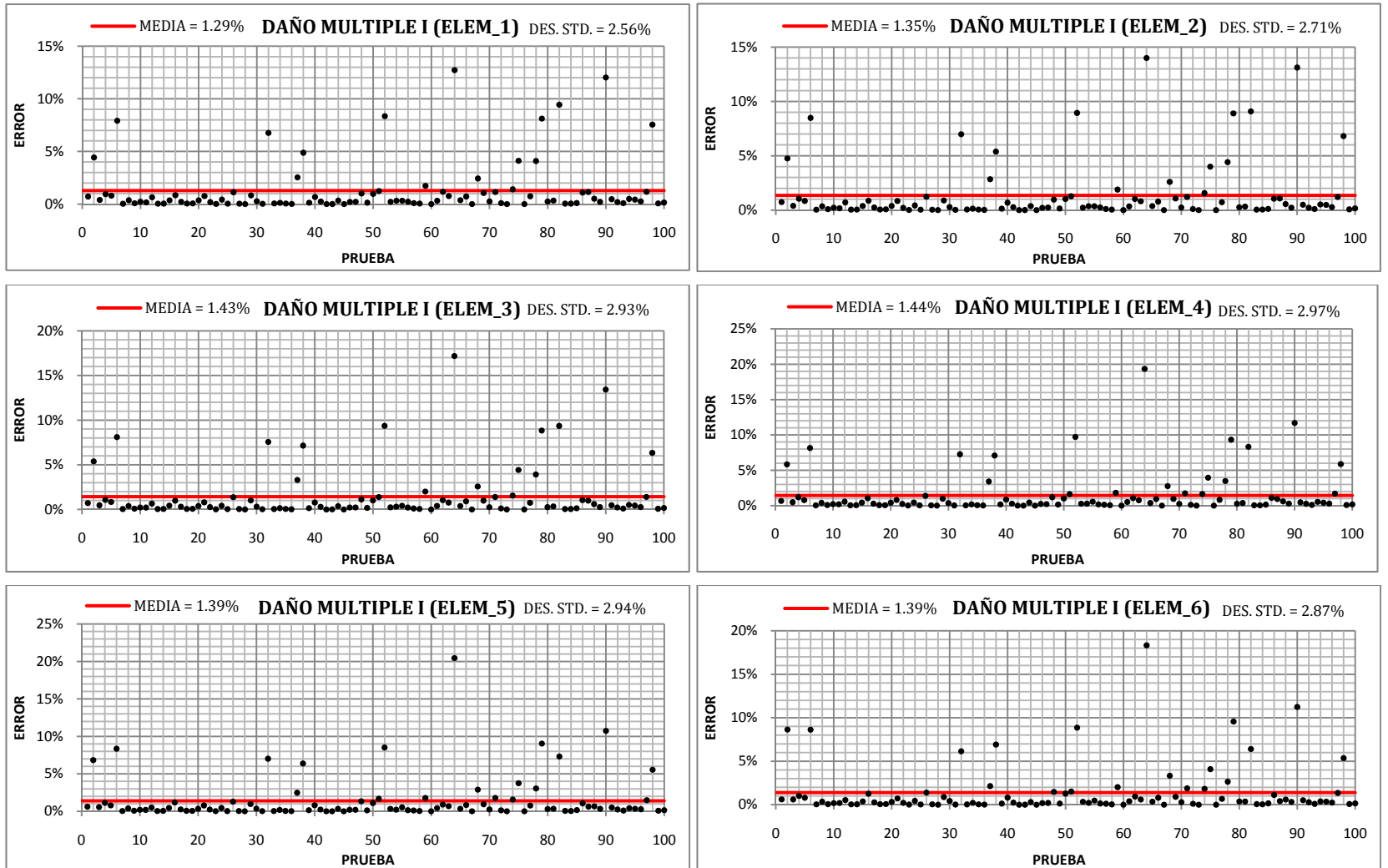
DAÑO MULTIPLE I								
ELEMENTO	MEJOR PRUEBA		PEOR PRUEBA		ESTADISTICA			
	F.D.	ERROR	F.D.	ERROR	DES. EST.	MEDIA	CONFIABILIDAD	
1	1	0.00%	0.87277	12.72%	2.56%	1.29%	88%	
2	0.78	0.00%	0.67079	14.00%	2.71%	1.35%	88%	
3	1	0.00%	0.82824	17.18%	2.93%	1.43%	87%	
4	1	0.00%	0.80652	19.35%	2.97%	1.44%	87%	
5	1	0.00%	0.79554	20.45%	2.94%	1.39%	88%	
6	1	0.00%	0.81676	18.32%	2.87%	1.39%	88%	
7	0.69	0.00%	0.57461	16.72%	2.62%	1.24%	88%	
8	0.15	0.00%	0.12754	14.97%	2.28%	1.07%	89%	
9	1	0.00%	0.84846	15.15%	2.28%	1.08%	89%	
10	1	0.00%	0.81658	18.34%	2.69%	1.36%	87%	
11	0.81	0.00%	0.66954	17.34%	3.15%	1.62%	86%	
12	1	0.00%	0.84553	15.45%	2.71%	1.28%	89%	
13	1	0.00%	0.85983	14.02%	2.39%	1.15%	89%	
14	1	0.00%	0.85726	14.27%	2.30%	1.09%	90%	
15	1	0.00%	0.86639	13.36%	2.21%	1.05%	90%	

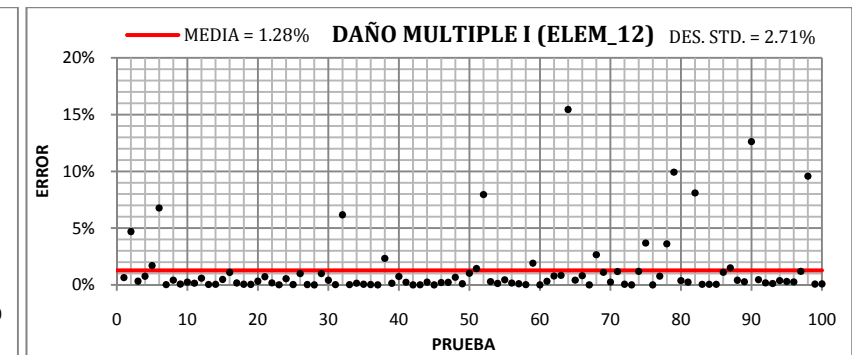
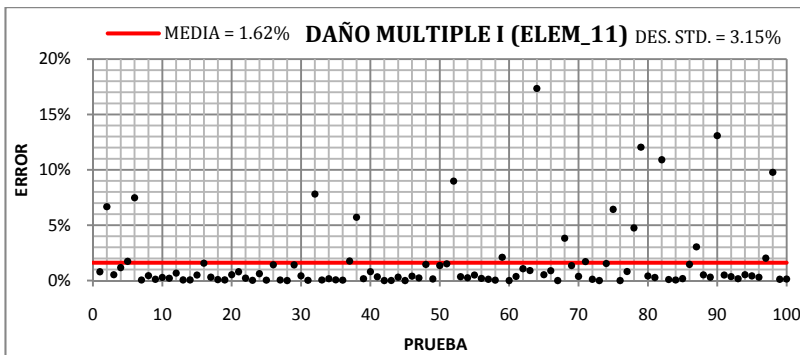
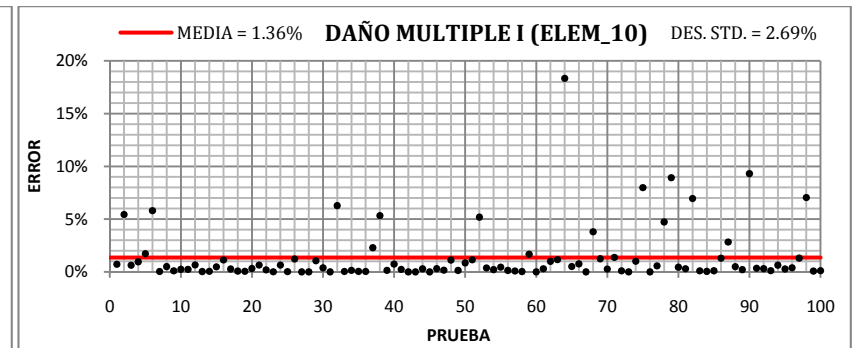
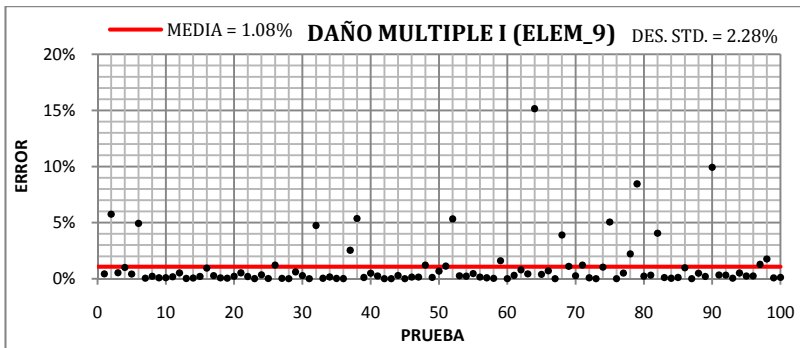
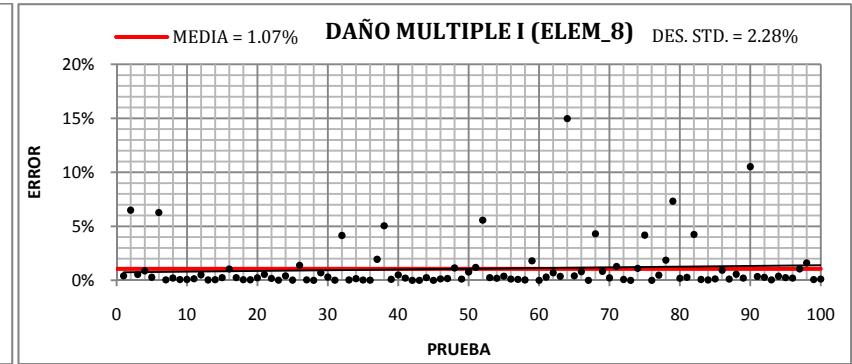
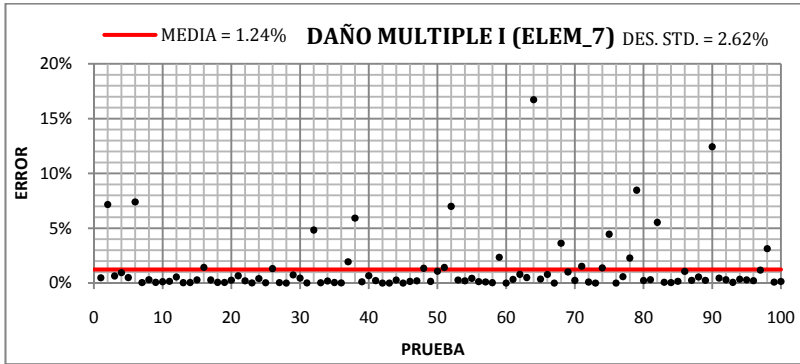
*F.D. = FACTOR DE DAÑO *DES. EST. = DESVIACION ESTANDAR

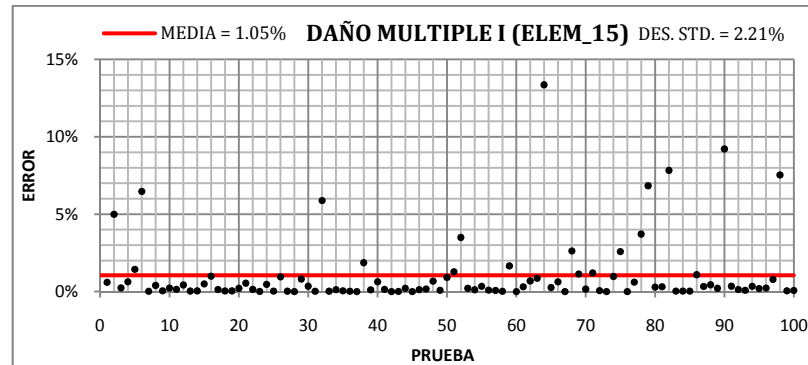
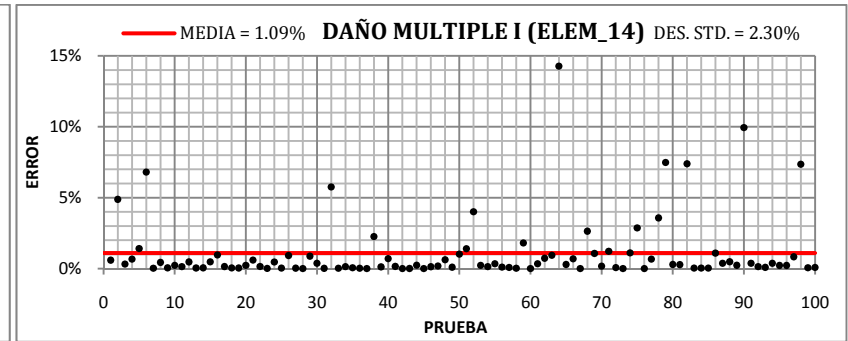
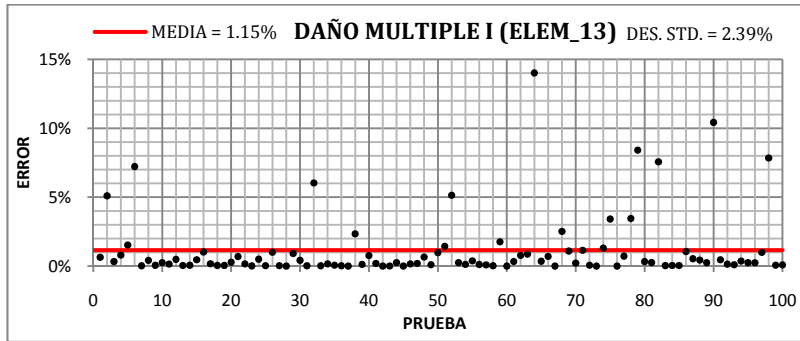
Fuente: Los autores

En la figura 41, se presentan las graficas del porcentaje de error vs número de pruebas para cada uno de los 15 elementos que componen el pórtico.

Figura 41. Resultados DAÑO MULTIPLE I







Fuente: Los autores

De acuerdo a los resultados mostrados, podemos apreciar el buen desempeño del algoritmo, obteniendo un error promedio menor que 1.62% y una desviación estándar menor de 3.15%, generando una confiabilidad entre el 87% y el 90% por lo que podemos considerar al algoritmo estable. Como podemos observar en la figura 41 la mayoría de las pruebas en las que se detectó el daño, se obtuvo un error entre el rango establecido (error relativo menor del 3%) para considerar la prueba como exitosa.

5.3.7. DAÑO MULTIPLE II

En este segundo caso de daño múltiple, de acuerdo a la tabla 7 y como se puede observar en la figura 35, los elementos 5, 6, 8 y 12 tienen un factor de daño, lo cual genera una pérdida de rigidez del 44%, 12%, 10% y 85% respectivamente.

Tabla 14. Resultados DAÑO MULTIPLE II

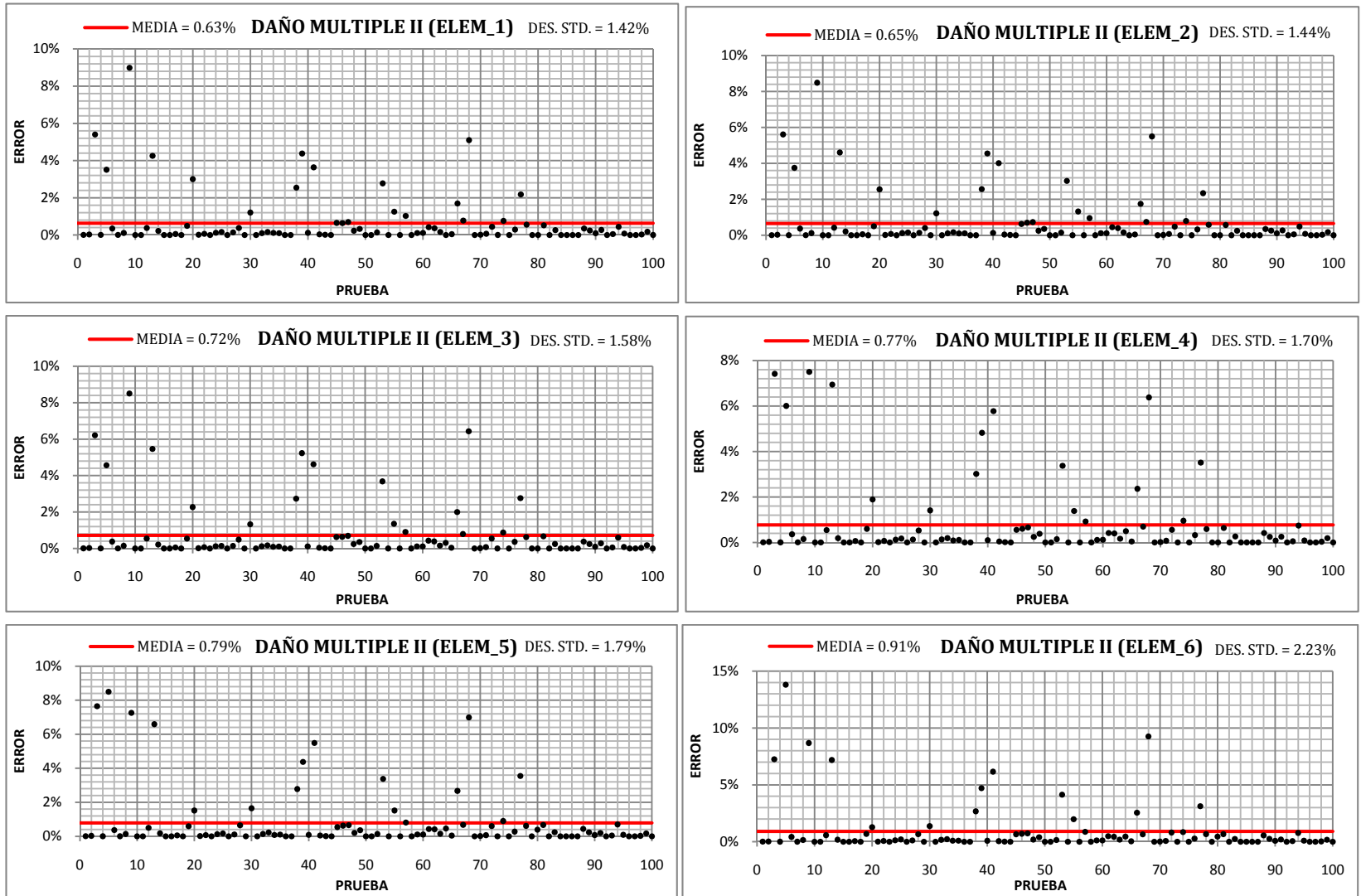
DAÑO MULTIPLE II								
ELEMENTO	MEJOR PRUEBA		PEOR PRUEBA		ESTADISTICA			
	F.D.	ERROR	F.D.	ERROR	DES. EST.	MEDIA	CONFIABILIDAD	
1	1	0.00%	0.91014	8.99%	1.42%	0.63%	92%	
2	1	0.00%	0.9151	8.49%	1.44%	0.65%	92%	
3	1	0.00%	0.91494	8.51%	1.58%	0.72%	92%	
4	1	0.00%	0.92496	7.50%	1.70%	0.77%	90%	
5	0.56	0.00%	0.51937	7.26%	1.79%	0.79%	91%	
6	0.88	0.00%	0.80364	8.68%	2.23%	0.91%	91%	
7	1	0.00%	0.91474	8.53%	2.30%	0.94%	91%	
8	0.9	0.00%	0.82199	8.67%	2.26%	0.91%	92%	
9	1	0.00%	0.92484	7.52%	2.21%	0.86%	92%	
10	1	0.00%	0.92162	7.84%	2.51%	0.94%	92%	
11	1	0.00%	0.85218	14.78%	3.10%	1.37%	88%	
12	0.15	0.00%	0.13384	10.77%	1.85%	0.90%	89%	
13	1	0.00%	0.88011	11.99%	1.97%	0.96%	87%	
14	1	0.00%	0.87793	12.21%	2.10%	1.01%	88%	
15	1	0.00%	0.87599	12.40%	2.23%	1.04%	88%	

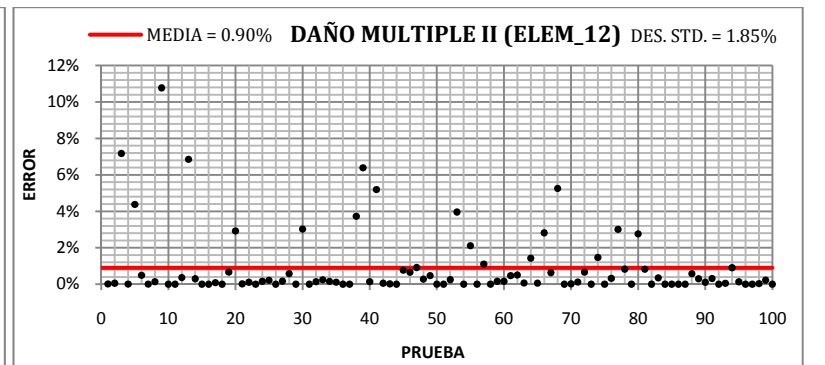
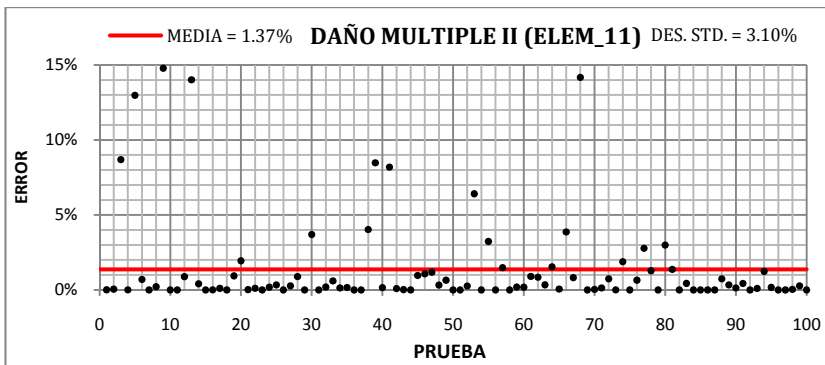
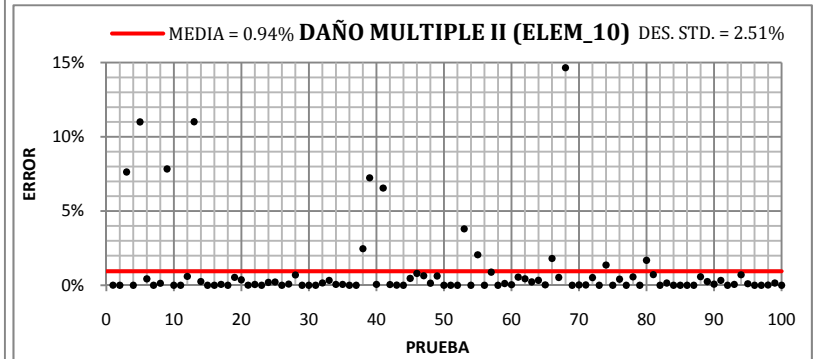
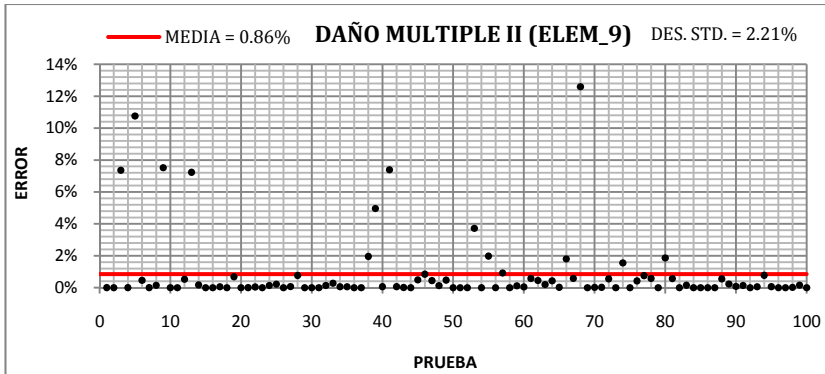
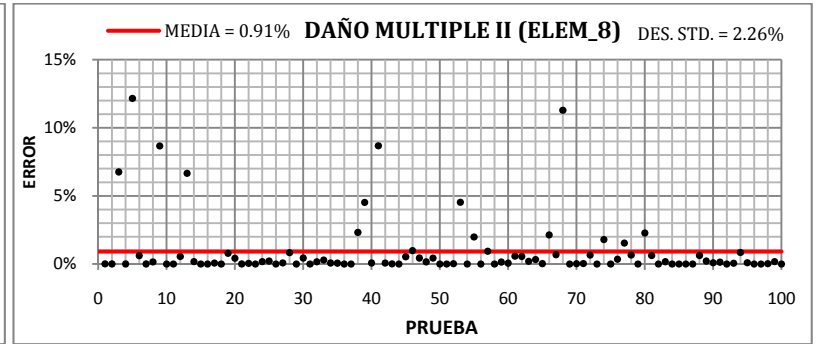
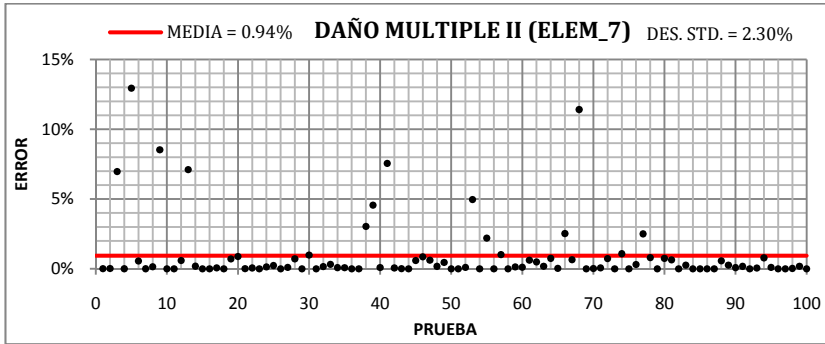
*F.D. = FACTOR DE DAÑO *DES. EST. = DESVIACION ESTANDAR

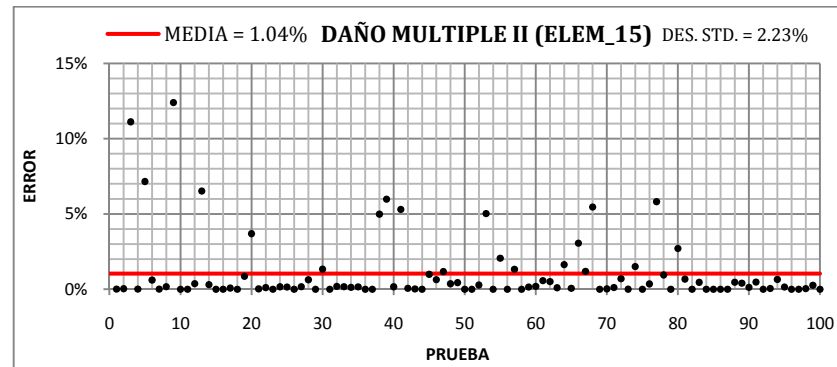
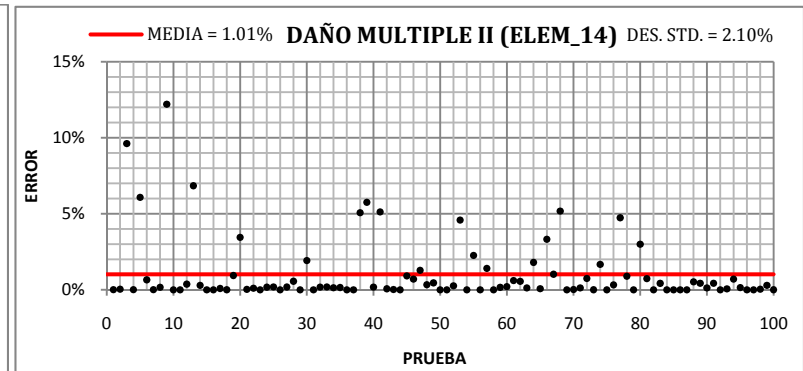
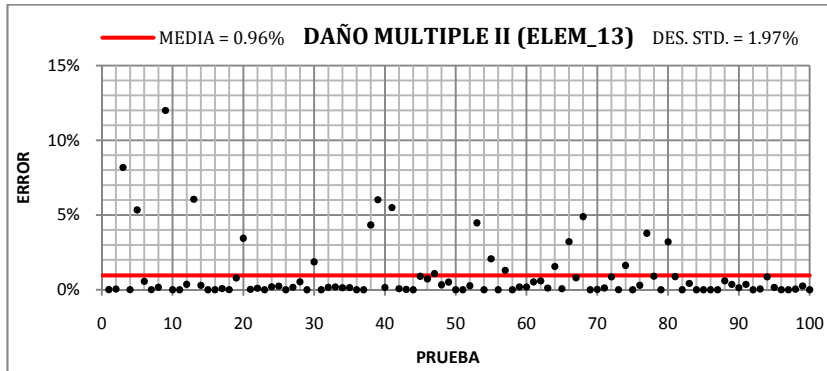
Fuente: Los autores

En este caso de daño el desempeño del algoritmo es similar comparado con el primer caso estudiado. En la figura 37 se puede observar el error relativo obtenido de cada uno de los elementos del pórtico en cada prueba realizada.

Figura 42. Resultados DAÑO SIMPLE II







Fuente: Los autores

De acuerdo a los resultados obtenidos, podemos apreciar que en este caso de daño el desempeño del algoritmo fue similar al anterior caso estudiado, obteniendo un error promedio menor que 1.37% y una desviación estándar menor de 3.10%, generando una confiabilidad entre el 88% y el 92%.

5.3.8. DAÑO MULTIPLE III

Los resultados de la implementación del paso al caso de DAÑO MULTIPLE III se encuentran registrados en la tabla 16. En este caso de daño, los elementos 2, 6, 8 y 15 tienen un factor de daño como se muestra en la tabla 7 y se puede observar en la figura 35, lo cual genera una pérdida de rigidez del 35%, 61%, 58% y 25% para cada elemento respectivamente.

Tabla 15. Resultados DAÑO MULTIPLE III

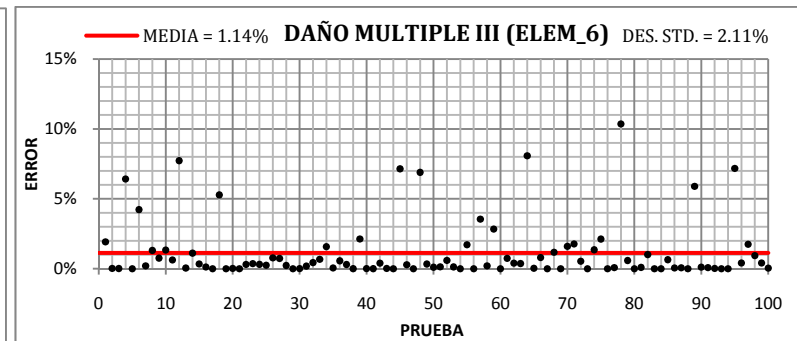
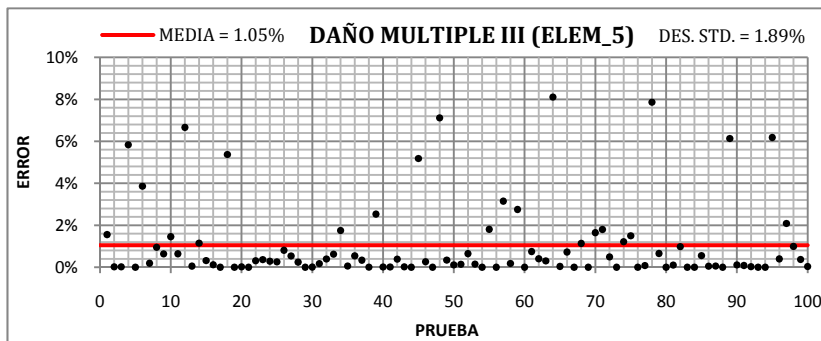
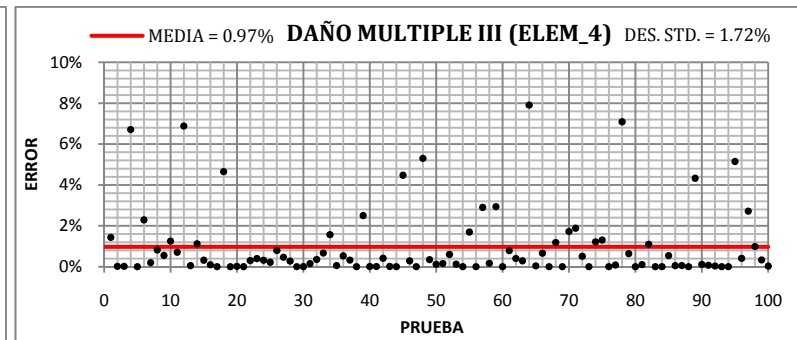
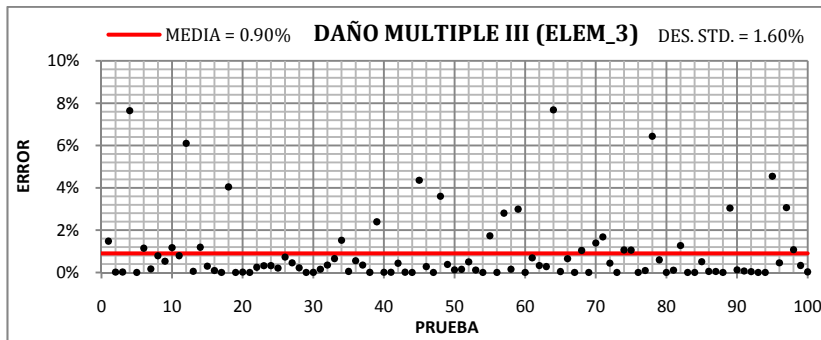
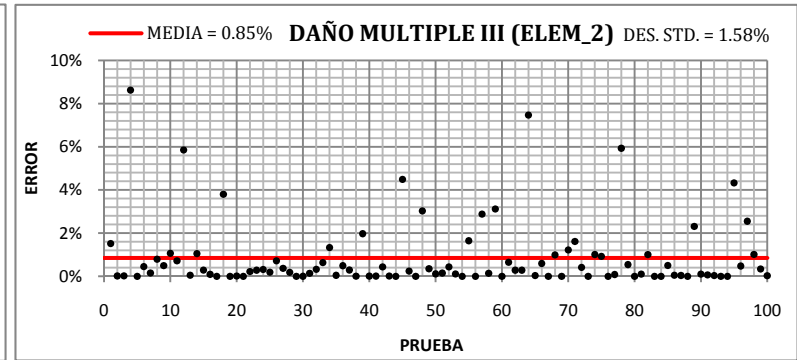
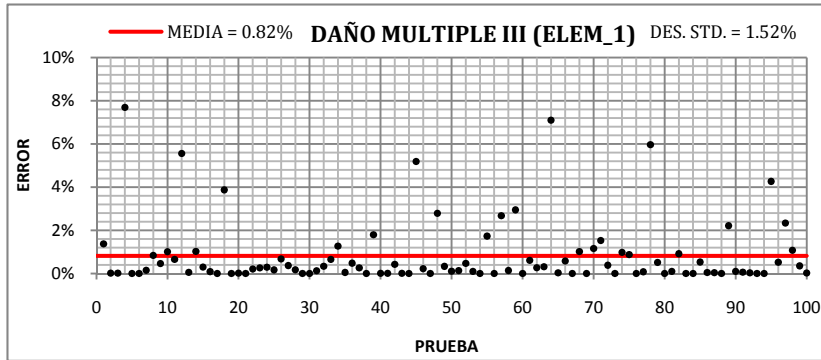
DAÑO MULTIPLE III							
ELEMENTO	MEJOR PRUEBA		PEOR PRUEBA		ESTADISTICA		
	F.D.	ERROR	F.D.	ERROR	DES. EST.	MEDIA	CONFIABILIDAD
1	1	0.00%	0.94034	5.97%	1.52%	0.82%	93%
2	0.65	0.00%	0.61145	5.93%	1.58%	0.85%	91%
3	1	0.00%	0.93565	6.44%	1.60%	0.90%	90%
4	1	0.00%	0.92911	7.09%	1.72%	0.97%	91%
5	1	0.00%	0.92137	7.86%	1.89%	1.05%	89%
6	0.39	0.00%	0.3496	10.36%	2.11%	1.14%	89%
7	1	0.00%	0.85535	14.47%	2.44%	1.20%	89%
8	0.42	0.00%	0.38475	8.39%	1.97%	1.01%	88%
9	1	0.00%	0.94113	5.89%	1.66%	0.84%	90%
10	1	0.00%	0.93065	6.94%	1.97%	1.06%	88%
11	1	0.00%	0.89501	10.50%	2.69%	1.51%	86%
12	1	0.00%	0.90222	9.78%	2.01%	1.05%	89%
13	1	0.00%	0.89927	10.07%	1.82%	0.94%	90%
14	1	0.00%	0.89134	10.87%	1.74%	0.87%	91%
15	0.75	0.00%	0.67737	9.68%	1.59%	0.80%	91%

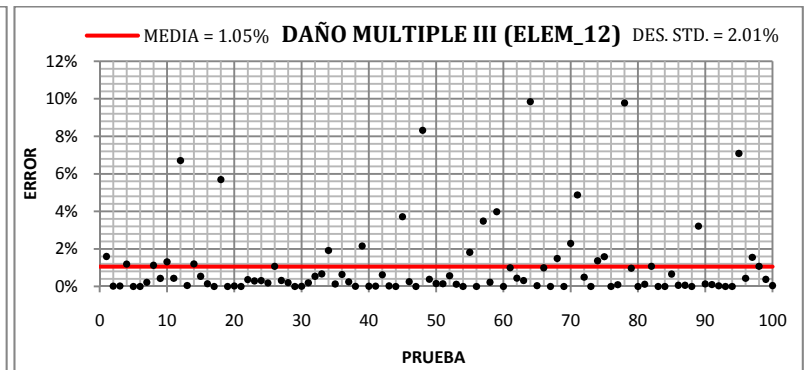
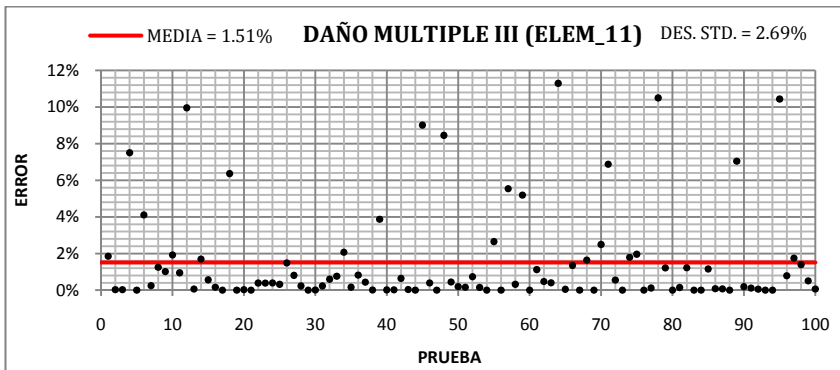
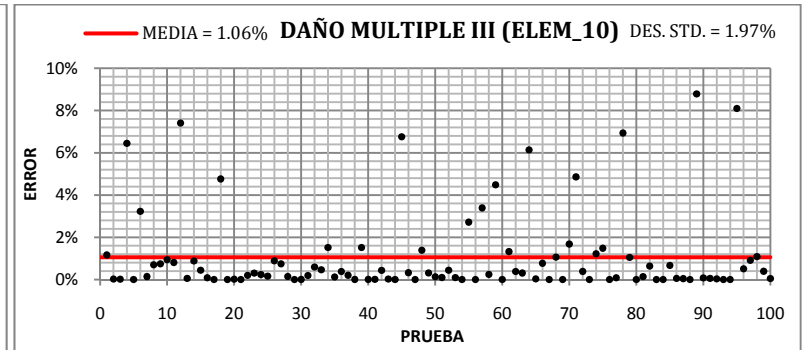
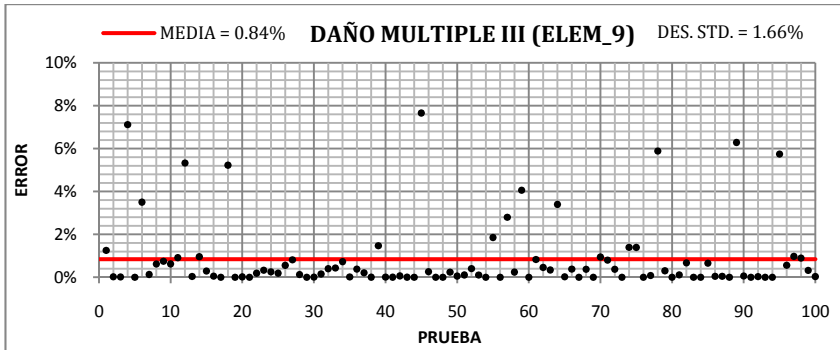
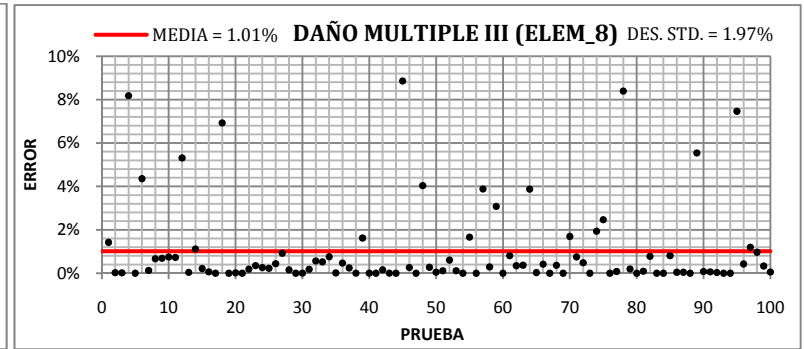
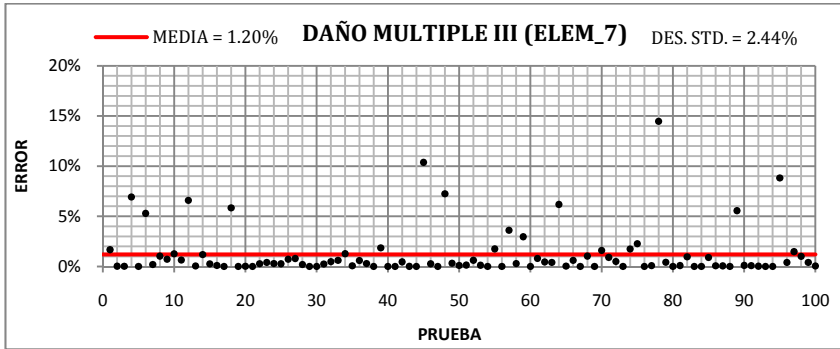
*F.D. = FACTOR DE DAÑO *DES. EST. = DESVIACION ESTANDAR

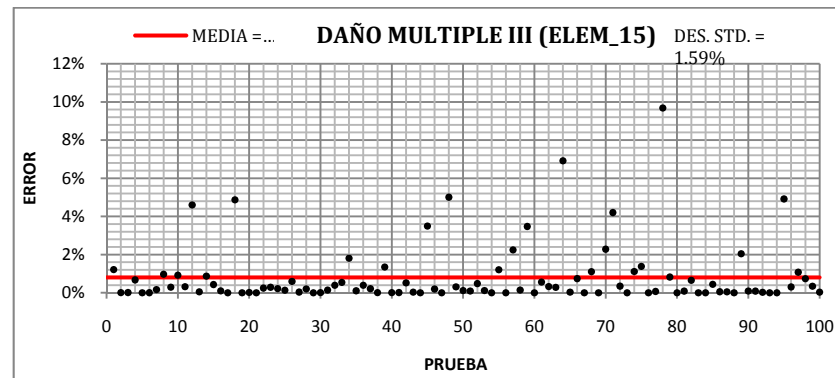
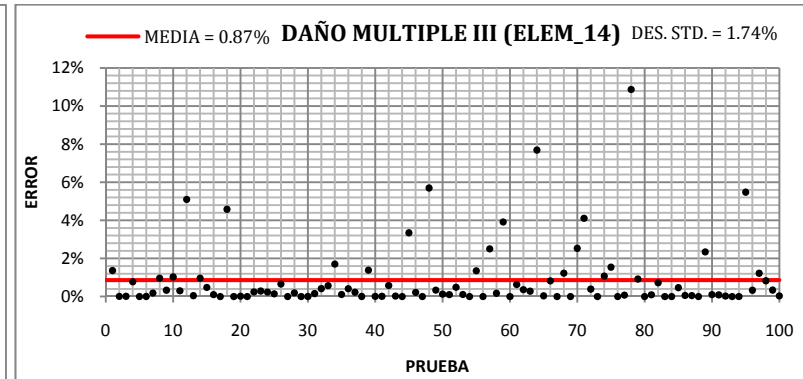
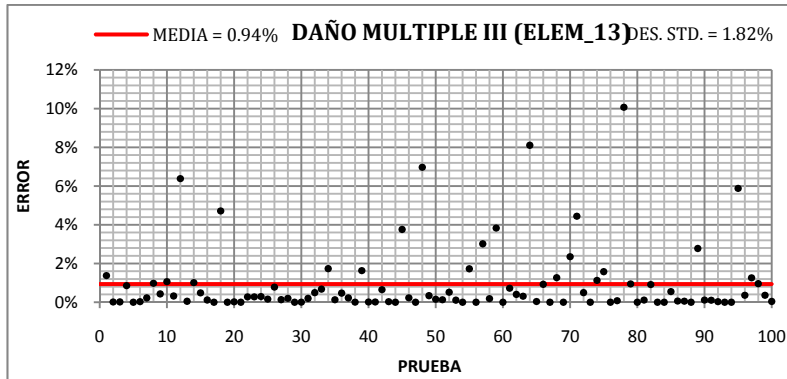
Fuente: Los autores

A continuación se presentan las graficas de los resultados obtenidos, donde se puede observar el error relativo del ajuste del daño hecho por el algoritmo en cada uno de los 15 elementos que componen el pórtico.

Figura 43. Resultados DAÑO MULTIPLE III







Fuente: Los autores

En general de acuerdo a los resultados obtenidos, podemos apreciar el buen desempeño del algoritmo, obteniendo un error promedio menor que 1,37% y una desviación estándar menor de 3,10%, valores muy cercanos a los obtenidos en los casos anteriores, pero en este caso se tiene una confiabilidad entre el 88% y el 92% por lo que podemos considerar al algoritmo estable. Al igual que los dos casos anteriores, se encontró un buen desempeño del algoritmo en la detección del daño.

5.3.9. DAÑO MULTIPLE IV

En la tabla 7 y en la figura 35, se puede observar que, los elementos 1, 9, 10 y 15 tienen un factor de daño, lo cual genera una pérdida de rigidez del 63%, 10%, 20% y 22% respectivamente. En la tabla 17 se presentan los resultados obtenidos.

Tabla 16. Resultados DAÑO MULTIPLE IV

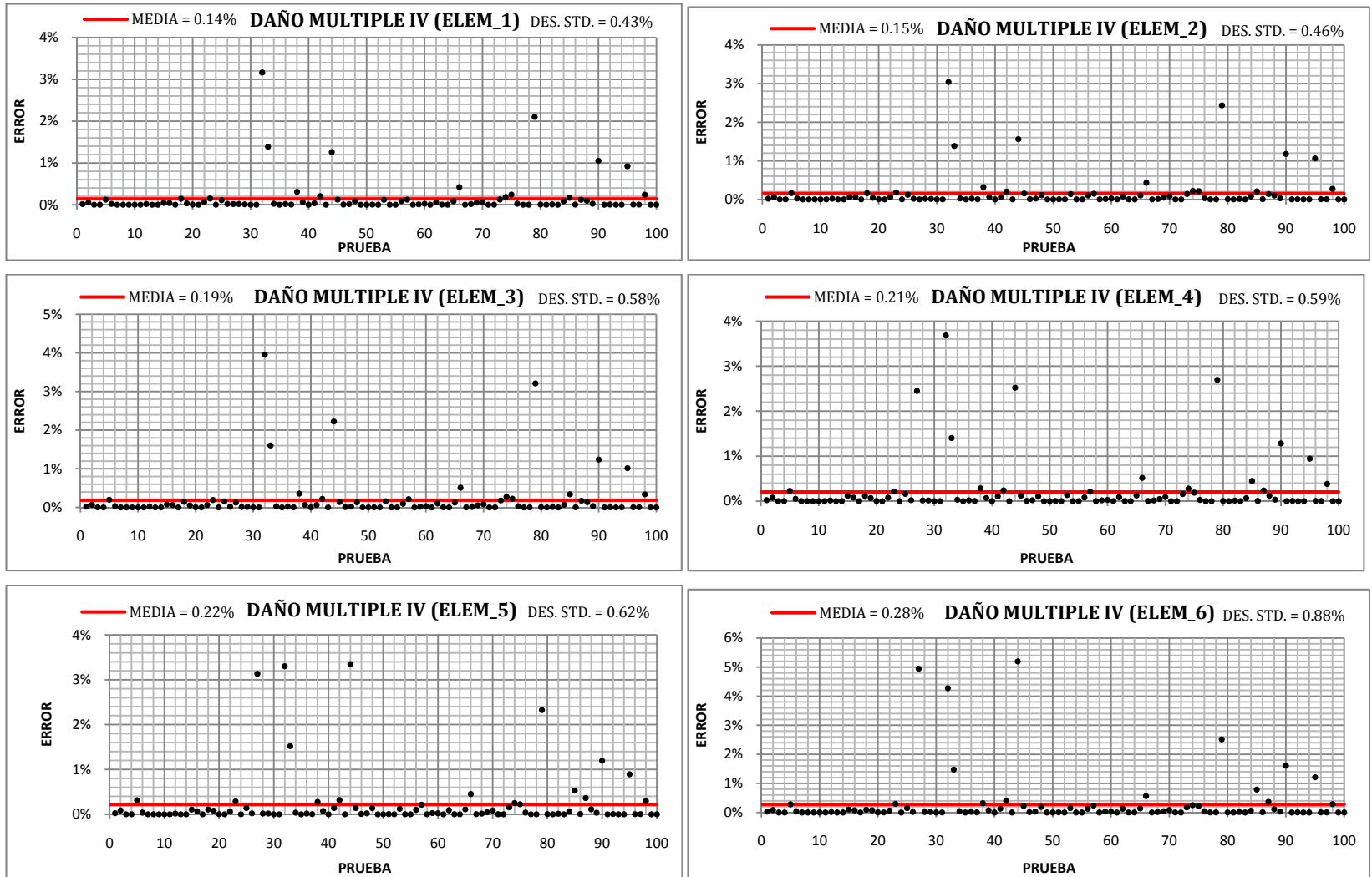
DAÑO MULTIPLE IV								
ELEMENTO	MEJOR PRUEBA		PEOR PRUEBA		ESTADISTICA			
	F.D.	ERROR	F.D.	ERROR	DES. EST.	MEDIA	CONFIABILIDAD	
1	0.37	0.00%	0.36222	2.10%	0.43%	0.14%	99%	
2	1	0.00%	0.97563	2.44%	0.46%	0.15%	99%	
3	1	0.00%	0.96792	3.21%	0.58%	0.19%	98%	
4	1	0.00%	0.97305	2.70%	0.59%	0.21%	99%	
5	1	0.00%	0.97675	2.33%	0.62%	0.22%	97%	
6	1	0.00%	0.97483	2.52%	0.88%	0.28%	97%	
7	1	0.00%	0.97211	2.79%	0.86%	0.28%	97%	
8	1	0.00%	0.98458	1.54%	0.70%	0.23%	97%	
9	0.9	0.00%	0.89189	0.90%	0.60%	0.20%	99%	
10	0.8	0.00%	0.75578	5.53%	0.91%	0.32%	97%	
11	1	0.00%	0.95625	4.38%	0.74%	0.26%	97%	
12	1	0.00%	0.95896	4.10%	0.58%	0.20%	99%	
13	1	0.00%	0.95763	4.24%	0.58%	0.19%	99%	
14	1	0.00%	0.95787	4.21%	0.57%	0.19%	99%	
15	0.78	0.00%	0.75013	3.83%	0.56%	0.18%	98%	

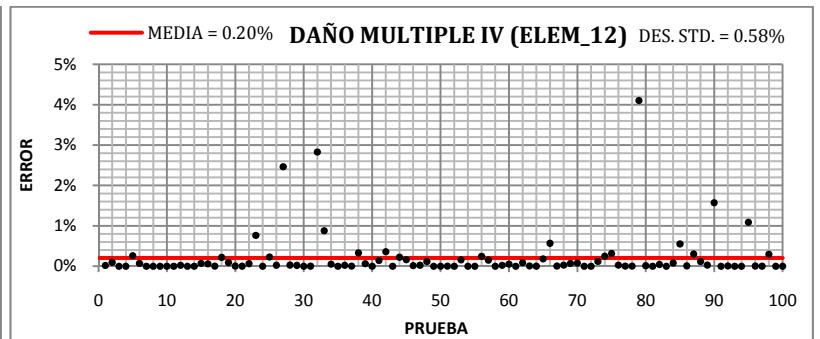
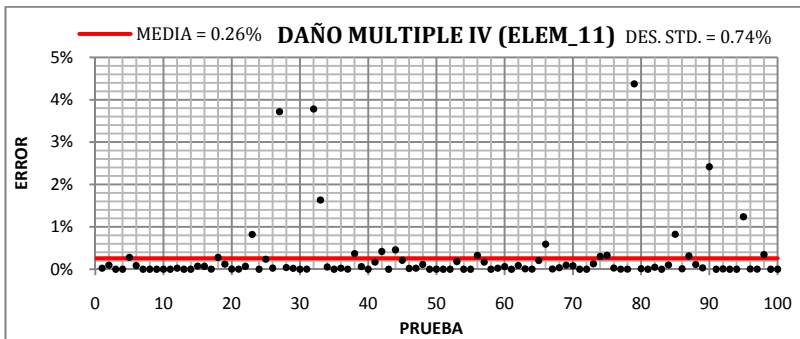
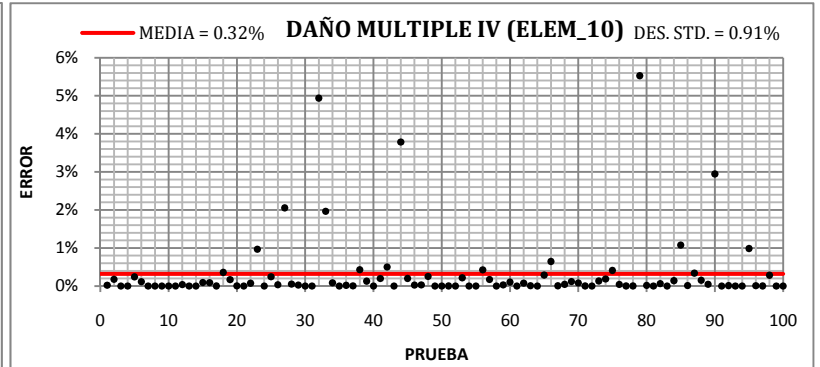
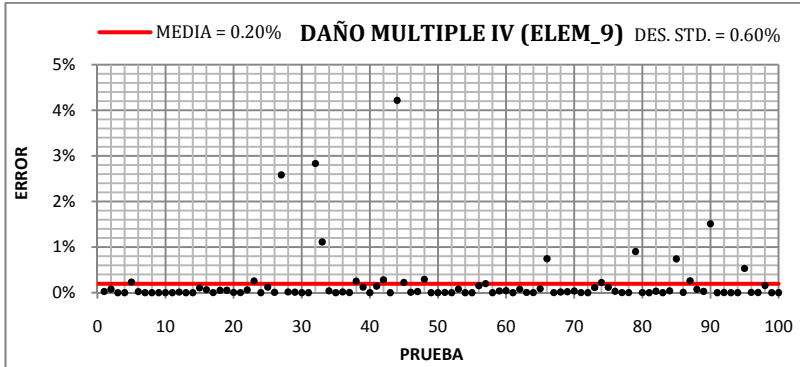
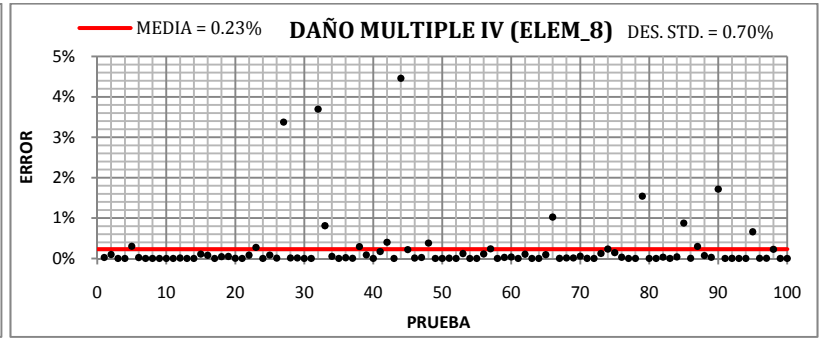
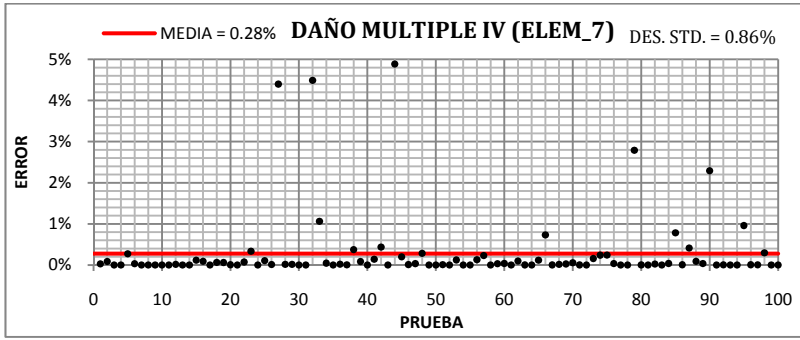
*F.D. = FACTOR DE DAÑO *DES. EST. = DESVIACION ESTANDAR

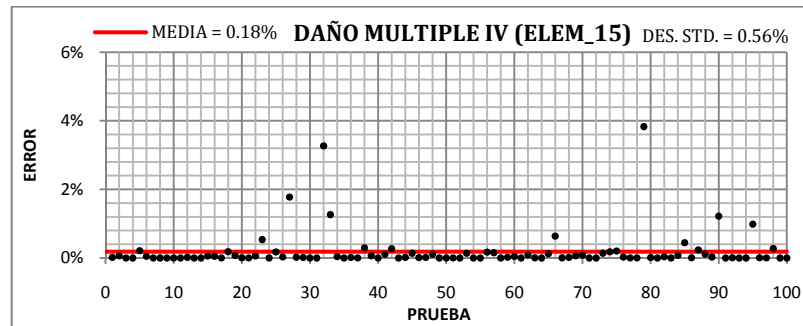
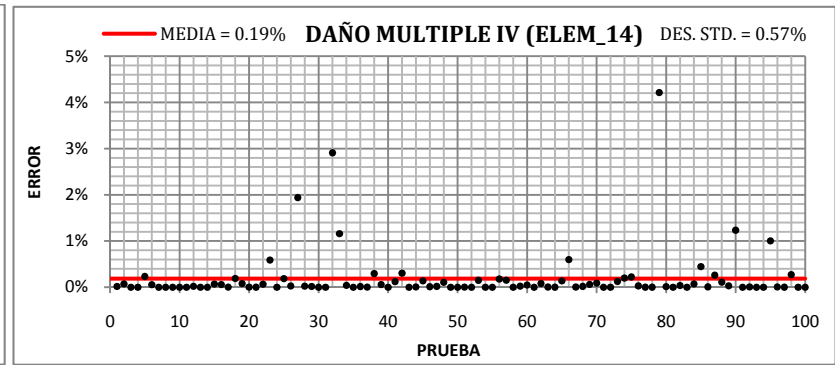
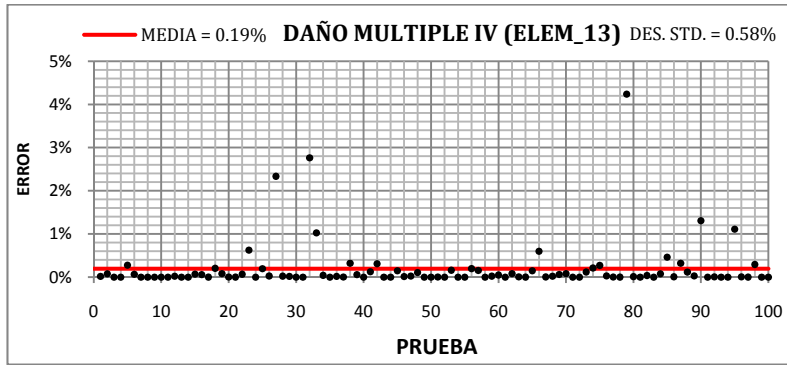
Fuente: Los autores

En este cuarto caso de daño el algoritmo tuvo un mejor desempeño, comparado con los tres casos anteriores. En la figura 37 se puede observar el error relativo obtenido de cada uno de los elementos del pórtico en cada prueba realizada.

Figura 44. Resultados DAÑO MULTIPLE IV







Fuente: Los autores

De acuerdo a los resultados obtenidos, podemos apreciar que en este caso de daño mejoro el desempeño del algoritmo, obteniendo un error promedio menor que 0.32% y una desviación estándar menor de 0,91%, generando una confiabilidad entre el 97% y el 99% para la detección correcta del daño dentro de la estructura.

5.3.10. DAÑO MULTIPLE V

En la tabla 7 y en la figura 35 se encuentra especificado este último caso de daño simple considerado, donde los elementos 2, 5, 7 y 12 tiene un factor de daño, el cual da como resultado una perdida re rigidez del 10%, 20%, 30% y 50% respectivamente. En la tabla 18 se presentan los resultados de la aplicación del PSO a este caso de daño.

Tabla 17. Resultados DAÑO MULTIPLE V

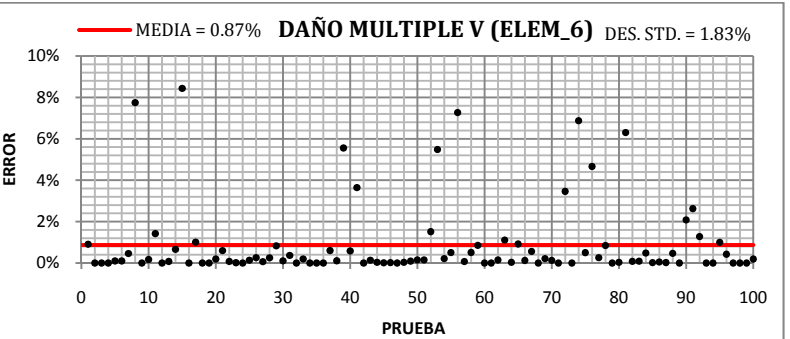
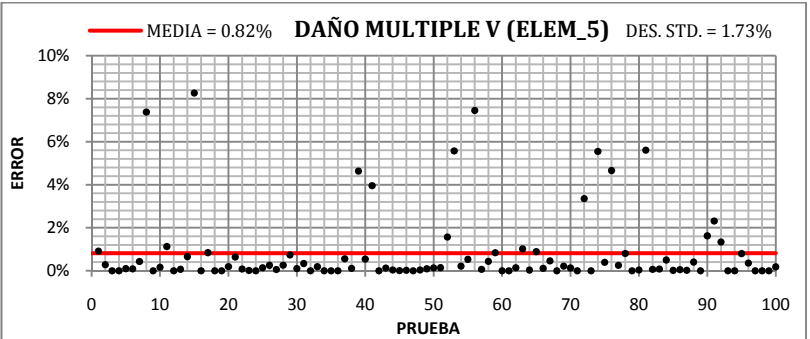
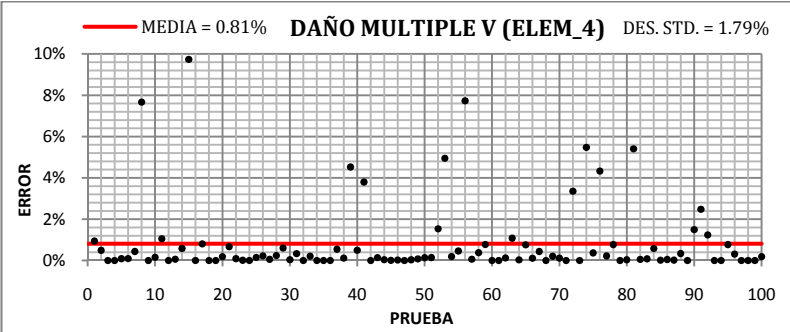
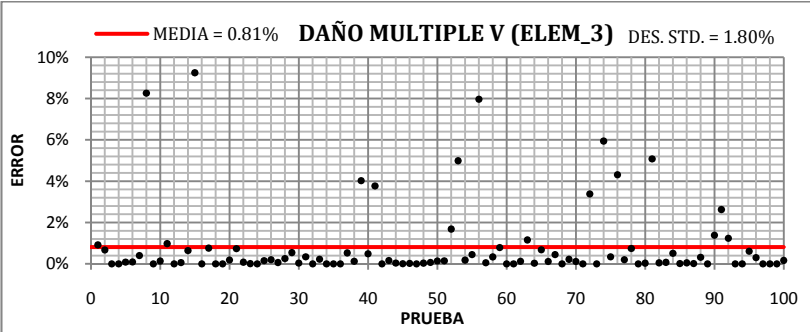
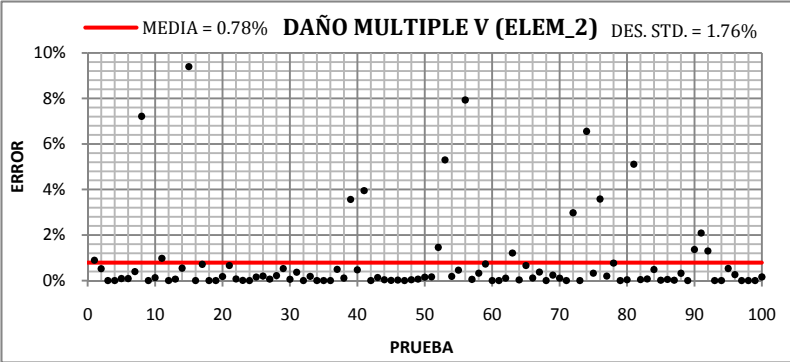
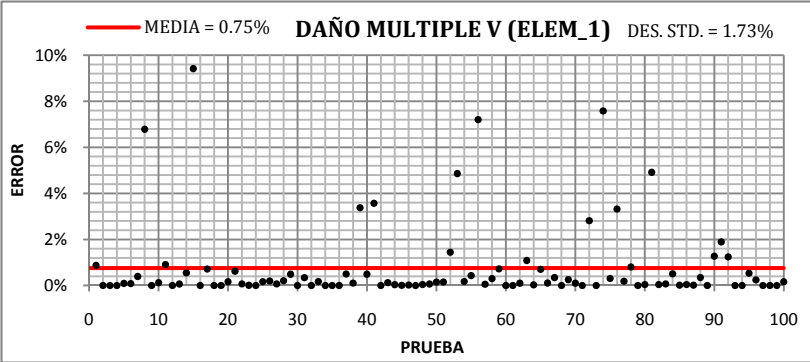
DAÑO MULTIPLE V								
ELEMENTO	MEJOR PRUEBA		PEOR PRUEBA		ESTADISTICA			
	F.D.	ERROR	F.D.	ERROR	DES. EST.	MEDIA	CONFIABILIDAD	
1	1	0.00%	0.90583	9.42%	1.73%	0.75%	91%	
2	0.9	0.00%	0.81545	9.39%	1.76%	0.78%	91%	
3	1	0.00%	0.90754	9.25%	1.80%	0.81%	90%	
4	1	0.00%	0.90265	9.74%	1.79%	0.81%	90%	
5	0.8	0.00%	0.73391	8.26%	1.73%	0.82%	90%	
6	1	0.00%	0.91566	8.43%	1.83%	0.87%	90%	
7	0.7	0.00%	0.63689	9.02%	1.93%	0.89%	90%	
8	1	0.00%	0.91144	8.86%	1.66%	0.75%	92%	
9	1	0.00%	0.90808	9.19%	1.69%	0.74%	91%	
10	1	0.00%	0.90493	9.51%	1.86%	0.85%	90%	
11	1	0.00%	0.89224	10.78%	2.57%	1.20%	89%	
12	0.5	0.00%	0.47295	5.41%	2.03%	0.88%	91%	
13	1	0.00%	0.94298	5.70%	1.79%	0.81%	91%	
14	1	0.00%	0.94432	5.57%	1.66%	0.74%	92%	
15	1	0.00%	0.95109	4.89%	1.65%	0.71%	92%	

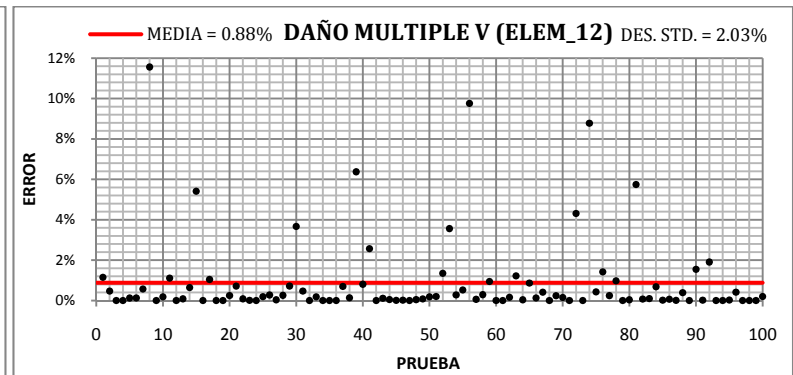
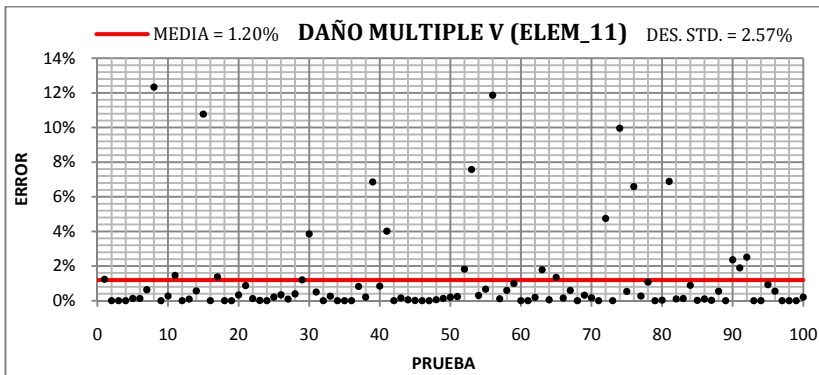
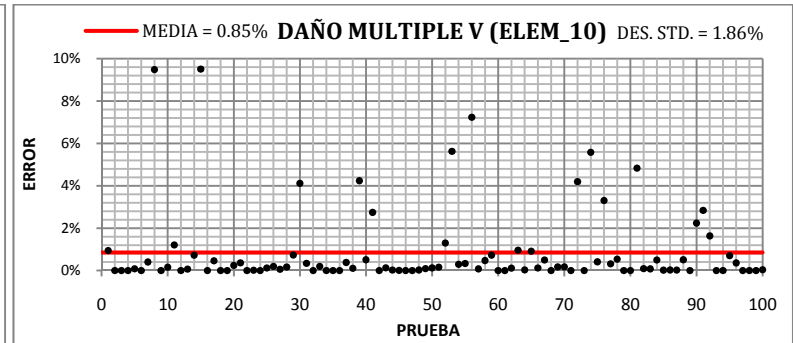
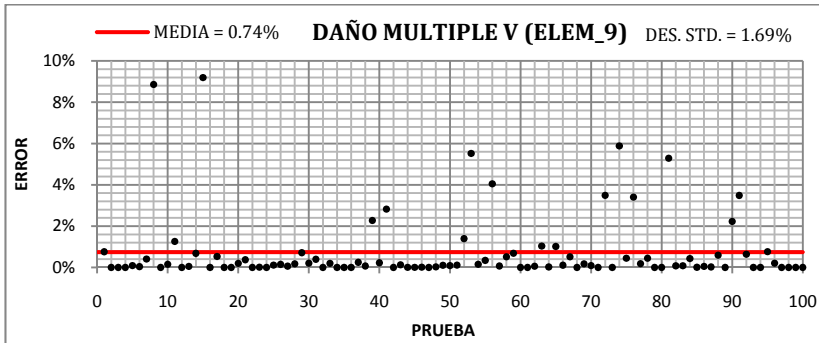
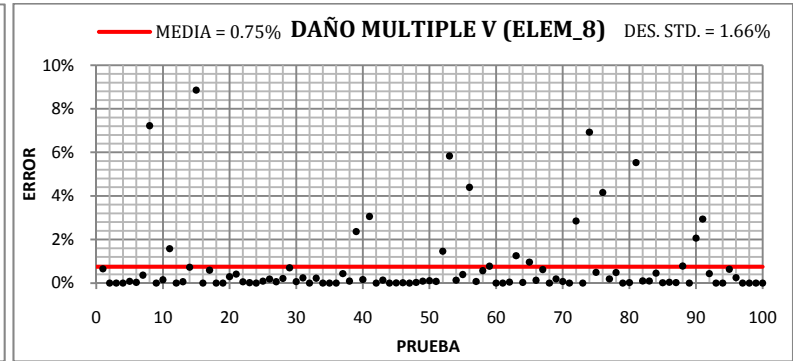
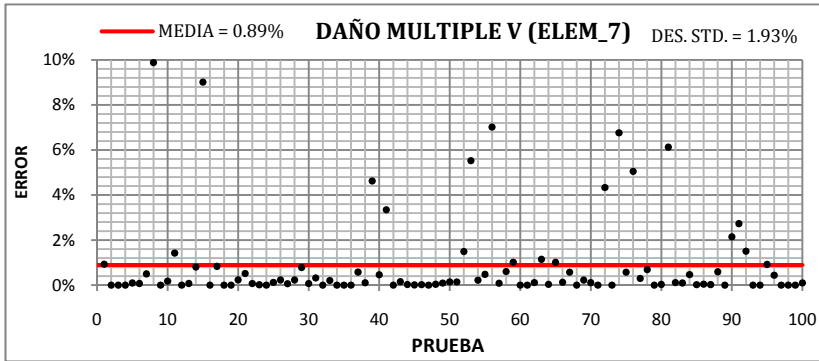
*F.D. = FACTOR DE DAÑO *DES. EST. = DESVIACION ESTANDAR

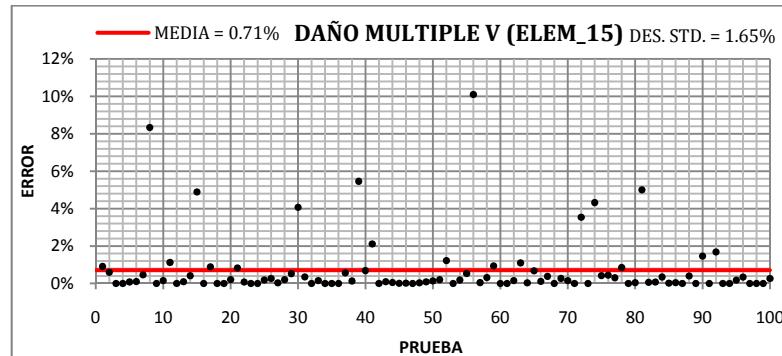
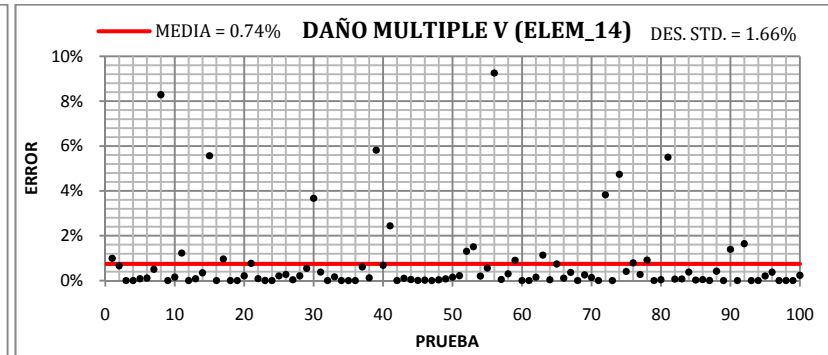
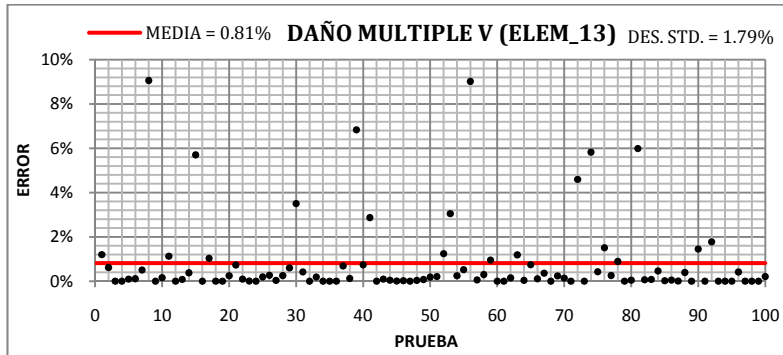
Fuente: Los autores

En la figura 40 se presentan los resultados de las pruebas en cada uno de los elementos del pórtico. En cada grafica se encuentran registrados los errores relativos obtenidos en las respectivas pruebas realizadas.

Figura 45. Resultados DAÑO MULTIPLE V







Fuente: Los autores

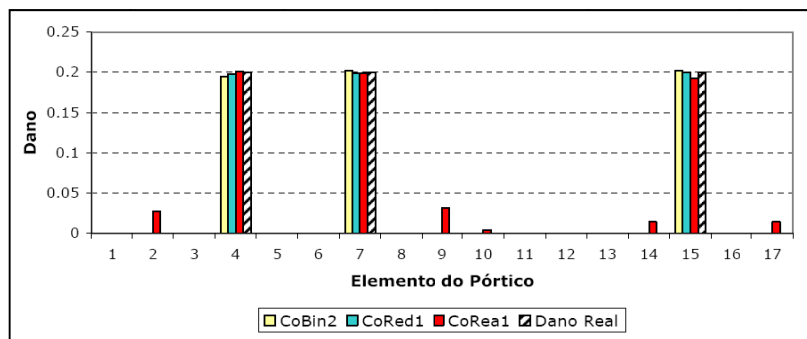
En este último caso de daño multiple se obtuvo un error relativo promedio menor de 1,20% una desviación estándar menor de 2,57% lo que permite concluir que el algoritmo es estable, además la confiabilidad que se encuentra entre 97% y 98% se puede considerar un buen desempeño del algoritmo

En general, de los resultados obtenidos de la implementación del algoritmo PSO al caso de daño multiple, podemos considerar el buen desempeño que realizó el algoritmo en este tipo de problema planteado, generando una alta confiabilidad con desviación y con un promedio de error bajo. Cabe resaltar que el problema en estudio es de 15 dimensiones.

5.3.11. Comparación PSO con los AG CoBin2, CoRed1 y CoRea1

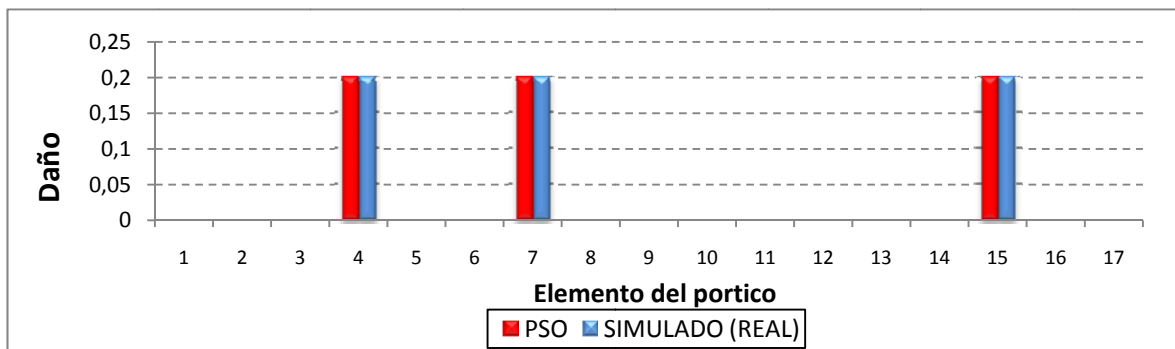
En un estudio reportado en la literatura especializada, *Villalba (2009)* realizó la implementación de los algoritmos genéticos (AG) utilizando parámetros dinámicos, al problema de detección de daños en estructuras tipo viga, cercha y pórtico, donde la simulación del daño se realizó por medio de la disminución del módulo de elasticidad. En este trabajo, el enfoque se centra en la comparación de resultados para el pórtico entre el PSO y los AG estudiados por *Villalba (2009)*, resaltando que esta estructura se compone de 17 elementos. El escenario de daño aquí estudiado es el P3C en el cual los elementos 4, 7 y 15 son dañados con una disminución de en el módulo de elasticidad de 0.2 para cada uno respectivamente, por lo tanto para el caso de la implementación del PSO consideramos un factor de daño de 0.8. La detección de daños mediante la implementación de los AG se realiza en dos etapas. La primera consta de identificar los elementos que están probablemente dañados, esto se realiza por medio del algoritmo genético de código binario con metodología de localización de elementos probablemente dañados (CoBin2). La segunda etapa consiste en la cuantificación del nivel del daño, esto se realiza con dos algoritmos y así poder observar su desempeño en el problema, el primero es un algoritmo genético de código real con proceso de reinicialización de la población de individuos (CoRea1), el segundo es un algoritmo genético de código binario con representación redundante implícita (CoRed1), en la figura 46 se presentan los resultados, donde las mediciones se consideran sin presencia de ruido.

Figura 46. Aplicación de los AG al escenario de daño P3C



Fuente: Villalba (2009)

Figura 47. Aplicación del PSO al escenario de daño P3C



Fuente: Los autores

En la figura 47 y la tabla 18 se encuentran registrados los resultados obtenidos en la implementación del PSO al escenario de daño P3C.

Tabla 18. Resultados escenario de daño P3C

Escenario de daño P3C								
ELEMENTO	MEJOR PRUEBA		PEOR PRUEBA		ESTADISTICA			
	F.D.	ERROR	F.D.	ERROR	DES. EST.	MEDIA	CONFIABILIDAD	
1	1	0.00%	0.96782	3.22%	0.58%	0.23%	98%	
2	1	0.00%	0.96344	3.66%	0.65%	0.26%	98%	
3	1	0.00%	0.95687	4.31%	0.75%	0.28%	98%	
4	0.8	0.00%	0.7749	3.14%	0.68%	0.27%	98%	
5	1	0.00%	0.9763	2.37%	0.66%	0.27%	98%	
6	1	0.00%	0.98064	1.94%	0.64%	0.25%	98%	
7	0.8	0.00%	0.78624	1.72%	0.57%	0.23%	100%	
8	1	0.00%	0.98958	1.04%	0.38%	0.16%	100%	
9	1	0.00%	0.99063	0.94%	0.32%	0.13%	100%	
10	1	0.00%	0.97575	2.43%	0.63%	0.22%	100%	
11	1	0.00%	0.97096	2.90%	0.61%	0.25%	100%	
12	1	0.00%	0.96317	3.68%	0.72%	0.27%	98%	
13	1	0.00%	0.95724	4.28%	0.84%	0.31%	96%	
14	1	0.00%	0.95152	4.85%	0.88%	0.32%	96%	
15	0.8	0.00%	0.75167	6.04%	1.03%	0.36%	96%	
16	1	0.00%	0.9428	5.72%	1.01%	0.36%	98%	
17	1	0.00%	0.94921	5.08%	0.94%	0.34%	96%	

*F.D. = FACTOR DE DAÑO *DES. EST. = DESVIACION ESTANDAR

Fuente: Los autores

De acuerdo a los resultados obtenidos por el PSO, en el cual se realizaron 50 pruebas, se puede asegurar su estabilidad en este problema de 17 dimensiones, debido a sus bajos valores del promedio del error y la desviación estándar. La figura 47 se realiza con el promedio de los resultados obtenidos para la reducción del modulo de elasticidad para cada elemento.

Comparando los resultados del PSO con los algoritmos genéticos (AG), se puede observar que el algoritmo genético implementado por *Villalba (2009)*, inicialmente consigue detectar de forma efectiva los elementos que probablemente se encuentran dañados. En cuanto a la cuantificación del daño el algoritmo CoRed1 presenta un mejor desempeño que el CoRea1, detectando el elemento afectado y la magnitud del daño acertadamente, mientras que el algoritmo CoRea1 presenta dificultades en la detección de los elementos dañados, a pesar que el factor de reducción de rigidez en los elementos fue acertado con errores muy bajos. Con respecto al PSO se encuentra que realizó una buena identificación al detectar, ubicar y cuantificar el daño presente, además de generar una confiabilidad alta, según los resultados de la tabla 18.

6. CONCLUSIONES

- Los resultados del PSO dan a conocer, que el algoritmo tiene la capacidad para resolver el problema de detección de daños en pórticos 2D, basándose en sus características dinámicas. En los casos tanto de daño simple como de daño múltiple que fueron simulados, el algoritmo logro identificar correctamente el daño, obteniendo una confiabilidad aproximada del 90%, con desviación estándar y un error promedio bajo.
- Se documento el estado del arte reportando los estudios más relevantes para la detección de daños. Se encontró que no existe una metodología única estandarizada para resolver este problema.
- Se programo una versión del PSO (Shi y Eberhart (2001)) en MATLAB®, la cual fue probada en varias funciones matemáticas reportadas en la literatura internacional, alcanzando los resultados esperados. Adicionalmente, se presentó un ejemplo ilustrativo del proceso realizado por el PSO para el cálculo de mínimos.
- Se realizo en MATLAB® la herramienta para la implementación del PSO para la detección de daños, donde el usuario especifica: geometría, propiedades de los materiales, discretización del pórtico y los elementos con su respectivo factor de daño para realizar la simulación. A partir de la teoría de MEF se determina la matriz de masa $[M]$ y la matriz de rigidez $[K]$ de la estructura, en esta ultima teniendo en cuenta los elementos dañados, y mediante el análisis dinámico se determina los parámetros usados en la función objetivo para la detección daños (datos experimentales). Seguidamente, mediante un proceso iterativo el PSO ajusta los factores de daño de cada elemento que compone el pórtico, en cada iteración se realiza la determinación de $[K]$ de acuerdo a los factores de daño de cada elemento y mediante el análisis dinámico se determina los parámetros usados en la función objetivo (datos analíticos). Finalmente, la herramienta realizada arroja como resultados, los elementos que componen el pórtico con su respectivo factor de daño ajustados por el PSO cumpliendo con el criterio de convergencia especificado.
- Se comprobó la influencia que presenta la función objetivo en el desempeño del PSO para el problema de detección de daños. En este trabajo presentó muy buen desempeño el VFR (ver ecuacion 5.4 y 5.5).
- La comparación del PSO con los AG permiten evidenciar la variabilidad de los resultados según el enfoque realizado para solucionar el problema planteado, como se puede observar con el algoritmo CoRea1 y el PSO, donde este último presentó un mejor desempeño.

7. RECOMENDACIONES

- Continuar con la investigación de la implementación y modificación del algoritmo de optimización PSO en la detección de daños, buscando mejorar su confiabilidad.
- Determinar la viabilidad de la implementación de un PSO modificado en una estructura en 3D.
- Comparar el desempeño del PSO modificado con otras técnicas reportadas en la literatura.

BIBLIOGRAFIA

- ARROS, Jorma. Dynamics of Structures. In: CHEN, Wai-Fah and SCAWTHORN, Charles. In: Earthquake Engineering Handbook. CRC Press, 2003. p. 3.1-3.40.
- ATIENZA PASCUAL, Ricardo. Técnicas de Actualización Aplicadas a la Detección de Daños. Tesis Doctoral. Madrid: Universidad Politécnica de Madrid. Escuela Técnica Superior de Ingenieros Aeronáuticos, 2004. 172p.
- BAVIRISETTY, Rambabu; VINAYAGAMOORTHY, Murugesu and DUAN, Lian. Dynamic Analysis. In: CHEN, Wai-Fah and and DUAN, Lian. Bridge Engineering: Seismic Design. CRC Press, 2003. p. 3.1-3.35.
- BEER, Ferdinand, JOHNSTON Russell y DEWLOF John. Mecánica de Materiales. 3 ed. México D.F.: McGraw-Hill Interamericana, 2004. 790p.
- BEGAMBRE, Oscar J. Detecção de Dano a Partir da Resposta Dinâmica da Estrutura: Estudo Analítico com Aplicação a Estruturas do Tipo Viga. Tesis para a obtenção do Título de Mestre em Engenharia de Estruturas. São Paulo: Universidade de São Paulo. Escola de Engenharia de São Carlos, 2004. 112p.
- CARDEN, Peter E. and FANNING, Paul. Vibration Based Condition Monitoring: A Review. In: Sage Publications. December, 2004, vol. 3, no. 4, p. 355-377
- CELIZ MELO, Camilo Eduardo y MILLAN PARAMO, Carlos Andrés. Evaluación del Desempeño de Algoritmos de Evolución Paramétrica para optimización. Trabajo de Grado. Bucaramanga: Universidad Industrial de Santander. Escuela de Ingeniería Civil, 2009, 200p.
- CUI, Zhihua ; CAI, Xingjuan, ZENG, Jianchao and SUN, Guoji. Particle Swarm Optimization With Fuss And Rws For High Dimensional Functions. In: Applied Mathematics and Computation. November, 2008. vol. 205, no.1, p. 98-108.
- ESCAMILLA, Jairo Uribe. Análisis de Estructuras. 3 ed. Bogotá D. C.: Escuela Colombiana de Ingeniería, 2002. 790p.
- Fonseca, Jun. Ferramentas de Simulação em Mecânica: Elementos Finitos, 2002.
- HERNANDEZ, Miguel. Detección de Daño a Partir de Cambios en las Características Vibracionales Usando Redes Neuronales. Universidad de los Andes [online], Junio, 2005 [citado 2009-05-25], p. 10. Disponible en internet: <http://hdl.handle.net/1992/484>.
- HIBBELER RUSSELL, Charles. Análisis Estructural. 3 ed. México: Pearson Educación, 1997. 752p.

JIAO, Wei; LIU, Guangbin and LIU, Dong. Elite Particle Swarm Optimization with Mutation. 7th International Conference on System Simulation and Scientific Computing (10/10/2008 - 12/10/2008: Beijing, China), p. 800-803.

KENNEDY, James and EBERHART Russell. Particle Swarm Optimization. Proceedings of the IEEE International Conference on Neural (December 1995: Perth, Australia), vol.4, pp. 1942-1948.

KUMAR TRIPATHI, Praveen; BANDYOPADHYAY, Sanghamitra and KUMAR PAL, Sankar. Multi-Objective Particle Swarm Optimization With Time Variant Inertia And Acceleration Coefficients. In: Information Sciences. November, 2007. vol. 177, no.22, p. 5033-5049.

KUNERT, Hernán y AMBROSINI, Daniel. Determinación de la Distribución de Daño en Vigas Mediante el Cambio en sus Frecuencias Naturales. En: Asociación Argentina de Mecánica Computacional. Octubre, 2007, vol. XXVI, p. 2623-2645.

LEE, Sangwook; SOAK, Sangmoon; OH, Sanghoun; PEDRYCZ, Witold and JEON, Moongu. Modified Binary Particle Swarm Optimization. In: Progress in Natural Science. September, 2008. vol. 18, no. 9, p. 1161-1166.

LI, Hong-Nan, LI, Dong-Sheng and SONG, Gang-Bing. Recent Applications of Fiber Optic Sensors to Health Monitoring in Civil Engineering. In: Engineering Structures. September, 2004, vol. 26, no. 11, p. 1647-1657.

MALDONADO RONDON, Esperanza y CHIO CHO, Gustavo. Análisis Sísmico de Edificaciones. Bucaramanga: Universidad Industrial de Santander, 2004. 297p.

MARULANDA, Johannio, THOMSON, Peter y MARULANDA A., Johannio. Monitoreo de Salud Estructural. En: Ingeniería y Competitividad. Diciembre, 2000, vol. 2, no. 2, p. 40-46.

MARULANDA, Johannio. Sistema de Monitoreo del Comportamiento Dinámico del Viaducto Pereira-Dosquebradas. Tesis Maestría en Ingeniería Civil. Cali: Universidad del Valle, 2005.

MERCADO ORTEGA, Álvaro Enrique y SEÑA MALO, Cristian Rodrigo. El PSOS (Particle Swarm Optimization + Simplex) en la optimización del calentamiento de un dieléctrico con microondas. Trabajo de Grado. Bucaramanga: Universidad Industrial de Santander. Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones, 2008. 145p.

NAGY, E. ET.AL. (2003). 3D Analysis of Tomographic Images. 16th ASCE Engineering Mechanics Conference, July 16-18, University of Washington, Seattle.

PARSOPOULOS, K.E. and VRAHATIS, M.N. Parameter selection and adaptation in Unified Particle Swarm Optimization. In: Mathematical and Computer Modelling. July, 2007. vol. 46, no. 1-2, p. 198-213.

PATJAWIT, Anun and KANOK-NUKULCHAI, Worsak. Health Monitoring of Highway Bridges Based on a Global Flexibility Index. In: Engineering Structures. August, 2005, vol. 27, no. 9, p. 1385-1391.

PAZ, Mario. Dinámica Estructural Teoría y Cálculo. 3 ed. España: Reverte, 2002. 648p.

PEREZ LOPEZ, Jesús Ramón. Contribución a los métodos de optimización basados en procesos naturales y su aplicación a la medida de antenas de campo próximo. Tesis

Doctoral. Santander: Universidad de Cantabria. Departamento de Ingeniería de Comunicaciones, 2005. 238p.

RICHARD LIEW, J. Y. and SHANMUGAM, N.E. Structural Analysis. In: CHEN, W. F. and LUI, E. M. Handbook of Structural Engineering. 2 ed. New York: CRC Press, 2005. p. 2.1-2.172

RIVEROS, Carlos. Structural Health Monitoring Methodology for Simply Supported Bridges: Numerical Implementation. En: Revista Facultad de Ingeniería (Universidad de Antioquia). Marzo, 2007, vol. 39, p.42-55.

SERRANO HERNÁNDEZ, Víctor Antonio. Métodos para reducir evaluaciones en algoritmos evolutivos Multi-Objetivo, basados en aproximación de funciones. Tesis de Maestría en Ciencias. México: Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional. Departamento de Computación, 2007. 102p.

SHI, Yuhui and EBERHART Russell. Particle Swarm Optimization: Developments, Applications and Resources. Congress on Evolutionary Computation (27/05/2001 - 30/05/2001: Seoul, South Korea), vol. 1, p. 81-86.

VENTER, Gerhard and SOBIESZCZANSKI-SOBIESKI, Jaroslaw. Particle Swarm Optimization. In: Journal AIAA. August, 2003, vol. 41, no. 8, p. 1583-1589.

VILLALBA MORALES, Jesús Daniel. Detecção de Dano em Estruturas Utilizando Algoritmos Geneticos e Parâmetros Dinâmicos. Tesis para a obtenção do Título de Mestre em Engenharia de Estruturas. São Paulo: Universidade de São Paulo. Escola de Engenharia de São Carlos, 2009. 215p.

XI-HUAI, Wang and JUN-JUN, Li. Hybrid Particle Swarm Optimization With Simulated Annealing. Third International Conference on Machine Learning and Cybernetics (26/08/2004-29/08/2004: Shanghai, China), p. 2402-2405.

ANEXOS

ANEXO A. RESULTADOS PSO DEL EJEMPLO DE APLICACIÓN

PARTICULA 1					PARTICULA 2											
	r_3	r_4	x_0		L		r_3	r_4	x_0		L	G	Y_{min}	D_{stad}		
INICIO	0,8147	0,9058	3,1472	-3,7301	3,1472	-3,7301	0,6324	0,0975	1,3236	-2,2150	1,3236	-2,2150	3,1472	4,1472	1070,606	NA
	0,1270	0,9134					0,2785	0,5469								

ITER.	r_1	r_2	x_1		L		r_1	r_2	x_1		L	G	Y_{min}	D_{stad}		
1	0,9575	0,9649	4,1472	-2,7301	3,1472	-3,7301	0,9572	0,4854	2,0939	-2,1762	2,0939	-2,1762	3,1472	-3,7301	1070,606	NA
	0,1576	0,9706					0,8003	0,1419								
2	0,4218	0,9157	3,1472	-3,7301	3,1472	-3,7301	0,6557	0,0357	2,9240	-3,1762	2,9240	-3,1762	2,9240	-3,1762	1033,050	NA
	0,7922	0,9595					0,8491	0,9340								
3	0,6787	0,7577	2,1472	-4,2560	3,1472	-3,7301	0,6555	0,1712	3,7213	-4,1366	2,9240	-3,1762	2,9240	-3,1762	1033,050	NA
	0,7431	0,3922					0,7060	0,0318								
4	0,2769	0,0462	1,8316	-3,2560	3,1472	-3,7301	0,6948	0,3171	2,8581	-3,1491	2,9240	-3,1762	2,9240	-3,1762	1033,050	NA
	0,0971	0,8235					0,9502	0,0344								
5	0,4387	0,3816	2,8316	-2,9326	2,8316	-2,9326	0,1869	0,4898	2,1512	-2,2974	2,9240	-3,1762	2,9240	-3,1762	1033,050	NA
	0,7655	0,7952					0,4456	0,6463								
6	0,5412	0,0428	3,7435	-2,7787	2,8316	-2,9326	0,2687	0,9239	3,1512	-3,1253	3,1512	-3,1253	3,1512	-3,1253	1030,414	NA
	0,5891	0,2840					0,4689	0,4403								
7	0,1515	0,3447	3,8666	-2,7469	2,8316	-2,9326	0,9678	0,5996	4,0370	-3,8588	3,1512	-3,1253	3,1512	-3,1253	1030,414	NA
	0,0594	0,1245					0,5060	0,1229								
8	0,5061	0,7519	2,8666	-3,1593	2,8666	-3,1593	0,4666	0,2896	3,4662	-3,0626	3,1512	-3,1253	3,1512	-3,1253	1030,414	NA

	0,1452	0,5101					0,1991	0,7777								
9	0,5001	0,1735	2,1146	-3,4979	2,8666	-3,1593	0,2097	0,1068	2,7811	-2,5411	3,1512	-3,1253	3,1512	-3,1253	1030,414	NA
	0,2074	0,1815					0,4313	0,8106								
10	0,1921	0,9695	3,1146	-3,7478	2,8666	-3,1593	0,9112	0,9866	3,6145	-3,5411	3,1512	-3,1253	3,1512	-3,1253	1030,414	NA
	0,0472	0,0004					0,4079	0,8738								
11	0,8087	0,2442	3,5485	-2,7478	2,8666	-3,1593	0,6261	0,3043	3,4333	-3,4567	3,1512	-3,1253	3,1512	-3,1253	1030,414	NA
	0,6455	0,5115					0,3873	0,6969								
12	0,1870	0,5374	3,2138	-2,6764	2,8666	-3,1593	0,3966	0,3221	2,8826	-2,5135	3,1512	-3,1253	3,1512	-3,1253	1030,414	NA
	0,2140	0,7328					0,5539	0,7674								
13	0,9767	0,5842	2,2138	-3,6764	2,8666	-3,1593	0,5593	0,5473	3,0448	-3,5135	3,1512	-3,1253	3,1512	-3,1253	1030,414	NA
	0,3836	0,9335					0,8201	0,8555								
14	0,9108	0,1465	2,9086	-3,4745	2,8666	-3,1593	0,4551	0,1124	3,2902	-3,5887	3,1512	-3,1253	3,1512	-3,1253	1030,414	NA
	0,3860	0,5187					0,5742	0,3195								
15	0,1770	0,4517	3,6365	-2,5697	2,8666	-3,1593	0,3759	0,1483	3,3294	-2,5887	3,1512	-3,1253	3,1512	-3,1253	1030,414	NA
	0,1271	0,9630					0,5982	0,8534								
16	0,9619	0,1969	2,6365	-3,1079	2,8666	-3,1593	0,5285	0,5924	2,9588	-3,0089	3,1512	-3,1253	3,1512	-3,1253	1030,414	NA
	0,4815	0,5747					0,2130	0,8667								
17	0,2965	0,6421	2,7101	-3,6130	2,8666	-3,1593	0,1430	0,3359	2,8748	-3,3736	3,1512	-3,1253	3,1512	-3,1253	1030,414	NA
	0,8373	0,8446					0,1666	0,0937								
18	0,7511	0,4887	3,4285	-2,8864	2,8666	-3,1593	0,1840	0,4995	3,1930	-3,1350	3,1512	-3,1253	3,1512	-3,1253	1030,414	NA
	0,3171	0,8173					0,1653	0,8362								
19	0,7597	0,0897	3,0243	-2,9422	3,0243	-2,9422	0,2519	0,5644	3,3459	-2,9377	3,1512	-3,1253	3,1512	-3,1253	1030,414	12,2369
	0,6908	0,3845					0,9496	0,6725								
20	0,0086	0,3727	2,8435	-3,3387	3,0243	-2,9422	0,1243	0,9398	3,0356	-3,2305	3,1512	-3,1253	3,1512	-3,1253	1030,414	8,9279
	0,6563	0,9787					0,9119	0,2267								
21	0,3954	0,8238	3,3727	-2,8857	3,0243	-2,9422	0,9327	0,6357	3,1910	-3,2861	3,1512	-3,1253	3,1512	-3,1253	1030,414	1,0818
	0,3923	0,9529					0,4121	0,2529								

22	0,4242	0,4212	3,2367	-3,0941	3,0243	-2,9422	0,6545	0,2484	3,2207	-2,8579	3,1512	-3,1253	3,1512	-3,1253	1030,414	0,9657
	0,8668	0,8489					0,8855	0,5594								
23	0,9203	0,5981	2,6562	-3,2469	3,0243	-2,9422	0,0125	0,5619	3,1599	-3,1722	3,1512	-3,1253	3,1512	-3,1253	1030,414	0,8113
	0,1261	0,9199					0,8912	0,2097								
24	0,2731	0,9415	3,4245	-3,0242	3,0243	-2,9422	0,5433	0,7318	3,0994	-3,2888	3,1512	-3,1253	3,1512	-3,1253	1030,414	0,5894
	0,2622	0,6535					0,2984	0,5644								
25	0,8162	0,4795	2,9822	-2,9709	2,9822	-2,9709	0,9806	0,5729	3,2230	-2,9390	3,1512	-3,1253	3,1512	-3,1253	1030,414	0,0000
	0,1527	0,5384					0,6457	0,6436								

ANEXO B. CODIGO PSO

```
clc;
clear all;

disp('*****')
disp('***** PSO *****')
disp('*****')
disp('_____')
disp('FUNCIONES A EVALUAR ')
disp('VENTER..... (1) ')
disp('N_DIMENSIONAL..... (2) ')
disp('SCHUEFEL..... (3) ')
disp('ROSENBROCK..... (4) ')
disp('_____')

%DATOS DE ENTRADA
funcion=input('Ingrese la funcion a evaluar = ');
d=input('Numero de dimensiones de la funcion = ');
a=input('Ingrese el limite inferior = ');
b=input('Ingrese el limite superior = ');
np=input('Ingrese el numero de particulas = ');
rd=input('Numero de datos para determinar la Desviación Standar = ');
E=input('Desviación Standar admisible = ');

to=clock;

%PARAMETROS DEL ALGORITMO
p=abs((b-a))/10;
c1=2;
c2=2;
w=1;
x0=[];
v0=[];

%DISTRIBUCION INICIAL DE POSICIONES X0 Y VELOCIDADES V0
for k=1:1:np
    for h=1:1:d
        a1=rand;
        x0(k,h)=a+a1*(b-a);
        if x0(k,h)>b
            x0(k,h)=b;
        elseif x0(k,h)<a
            x0(k,h)=a;
        end
        b1=rand;
        v0(k,h)=(a+b1*(b-a))/1;
        if v0(k,h)>p
            v0(k,h)=p;
        elseif v0(k,h)<-p
            v0(k,h)=-p;
        end
    end
end

e=0;
des=10;

while des>E
```

```

e=e+1;
if e==1

    %EVALUACION DE LA FUNCION EN LOS PUNTOS X0
    switch funcion
        case 1 %FUNCION VENTER
            for k=1:1:np
                L(k,:)=x0(k,:);
                ym_local(k,1)=L(k,1)^2-100*cos(L(k,1))^2-100*cos(L(k,1)^2/30)...
                +L(k,2)^2-100*cos(L(k,2))^2-100*cos(L(k,2)^2/30)+1400;
            end
        case 2 %FUNCION N_DIMENSIONAL
            for k=1:1:np
                L(k,:)=x0(k,:);
                ev=0;
                for h=1:1:d
                    ev=L(k,h)^4-16*L(k,h)^2+5*L(k,h)+ev;
                end
                ym_local(k,1)=(1/d)*ev;
            end
        case 3 %FUNCION SCHUEFEL
            for k=1:1:np
                L(k,:)=x0(k,:);
                ev=0;
                for h=1:1:d
                    ev=-L(k,h)*sin(sqrt(abs(L(k,h))))+ev;
                end
                ym_local(k,1)=ev;
            end
        case 4 %FUNCION ROSENBROCK
            for k=1:1:np
                L(k,:)=x0(k,:);
                ev=0;
                for h=1:1:(d-1)
                    ev=100*(L(k,h+1)-L(k,h)^2)^2+(L(k,h)-1)^2+ev;
                end
                ym_local(k,1)=ev;
            end
        end

    end

    %COMPARACION DE LA FUNCION EVALUADA EN X0 Y X1 PARA LA ASIGNACION DE LOS
    %MEJORES PUNTOS Y EVALUACIONES LOCALES
    else
        for k=1:1:np
            if yx1(k,1)<y_m_local(k,1)
                L(k,:)=x1(k,:);
                ym_local(k,1)=yx1(k,1);
            else
                L(k,:)=L(k,:);
                ym_local(k,1)=ym_local(k,1);
            end
        end
    end

    end

    %VALOR MINIMO DE LA FUNCION
    ymin=min(ym_local);

    %PUNTO DONDE LA FUNCION ES MINIMA
    for k=1:1:np
        if ymin==ym_local(k,1)

```

```

        G=L(k, :);
    end
end

%ACTUALIZACION DE X0 Y V0
if e>1
    x0=x1;
    v0=v1;
end

%DETERMINACION DE X1 Y V1
for k=1:1:np
    for h=1:1:d
        r1=rand;
        r2=rand;
        v1(k,h)=w*v0(k,h)+(c1*r1*(L(k,h)-x0(k,h))/1)+(c2*r2*(G(1,h)...
        -x0(k,h))/1);
        if v1(k,h)>p
            v1(k,h)=p;
        elseif v1(k,h)<-p
            v1(k,h)=-p;
        end
        x1(k,h)=x0(k,h)+v1(k,h)*1;
        if x1(k,h)>b
            x1(k,h)=b;
        elseif x1(k,h)<a
            x1(k,h)=a;
        end
    end
end

%EVALUACION DE LA FUNCION EN LOS PUNTOS X1
switch funcion
case 1 %FUNCION VENTER
    for k=1:1:np
        yx1(k,1)=x1(k,1)^2-100*cos(x1(k,1))^2-100*cos(x1(k,1)^2/30)+...
        x1(k,2)^2-100*cos(x1(k,2))^2-100*cos(x1(k,2)^2/30)+1400;
    end
case 2 %FUNCION N_DIMENSIONAL
    for k=1:1:np
        ev=0;
        for h=1:1:d
            ev=x1(k,h)^4-16*x1(k,h)^2+5*x1(k,h)+ev;
        end
        yx1(k,1)=(1/d)*ev;
    end
case 3 %FUNCION SCHUEFEL
    for k=1:1:np
        ev=0;
        for h=1:1:d
            ev=-x1(k,h)*sin(sqrt(abs(x1(k,h))))+ev;
        end
        yx1(k,1)=ev;
    end
case 4 %FUNCION ROSENBROCK
    for k=1:1:np
        ev=0;
        for h=1:1:(d-1)
            ev=100*(x1(k,h+1)-x1(k,h)^2)^2+(x1(k,h)-1)^2+ev;
        end
    end
end

```

```

        yx1(k,1)=ev;
    end
end

%REDUCCION LINEAL DE w
w=w*0.98;

%ALMACENAMIENTO DE DATOS PARA DETERMINAR EL CRITERIO DE PARADA
ds(1,e)=ymin;

%CALCULO DEL CRITERIO DE PARADA
if e>rd
    des=std(ds(1,e-rd:e));
end
end

duracion=etime(clock,to)
iteraciones=e
valor_min=ymin
puntos=G'

```

ANEXO C. CODIGO PARA LA SIMULACION DEL PORTICO

```
clc;
clear all;

%INGRESO DE DATOS GENERALES DEL PORTICO
disp('DATOS GENERALES DEL PORTICO')
H1=input('Altura de la columna 1 [m] = ');
H2=input('Altura de la columna 2 [m] = ');
L=input('Longitud del portico [m] = ');
E=input('Modulo de elasticidad del material [N/m2] = ');
d=input('Densidad del material [kg/m3] = ');
clc;

%INGRESO DE DATOS DEL ELEMENTO 1 (COLUMNA 1)
disp('DATOS ELEMENTO 1')
A1=input('Ingrese el area de la seccion [m2] = ');
I1=input('Ingrese la inercia de la seccion [m4] = ');
n1=input('Ingrese el numero de divisiones del elemento = ');

L1=H1/n1;

C1=(A1*E)/L1;   C2=(12*E*I1)/L1^3;   C3=(6*E*I1)/L1^2;   C4=(4*E*I1)/L1;
C5=(2*E*I1)/L1;

%MATRIZ DE RIGIDEZ DEL ELEMENTO 1 EN COORDENADAS LOCALES
K1(1,1)=C1;   K1(1,2)=0;   K1(1,3)=0;   K1(1,4)=-C1;   K1(1,5)=0;   K1(1,6)=0;
K1(2,1)=0;   K1(2,2)=C2;   K1(2,3)=C3;   K1(2,4)=0;   K1(2,5)=-C2;   K1(2,6)=C3;
K1(3,1)=0;   K1(3,2)=C3;   K1(3,3)=C4;   K1(3,4)=0;   K1(3,5)=-C3;   K1(3,6)=C5;
K1(4,1)=-C1;   K1(4,2)=0;   K1(4,3)=0;   K1(4,4)=C1;   K1(4,5)=0;   K1(4,6)=0;
K1(5,1)=0;   K1(5,2)=-C2;   K1(5,3)=-C3;   K1(5,4)=0;   K1(5,5)=C2;   K1(5,6)=-C3;
K1(6,1)=0;   K1(6,2)=C3;   K1(6,3)=C5;   K1(6,4)=0;   K1(6,5)=-C3;   K1(6,6)=C4;

D1=22*L1;   D2=13*L1;   D3=4*L1^2;   D4=3*L1^2;

%MATRIZ DE MASA DEL ELEMENTO 1 EN COORDENADAS LOCALES
M1(1,1)=140;   M1(1,2)=0;   M1(1,3)=0;   M1(1,4)=70;   M1(1,5)=0;   M1(1,6)=0;
M1(2,1)=0;   M1(2,2)=156;   M1(2,3)=D1;   M1(2,4)=0;   M1(2,5)=54;   M1(2,6)=-D2;
M1(3,1)=0;   M1(3,2)=D1;   M1(3,3)=D3;   M1(3,4)=0;   M1(3,5)=D2;   M1(3,6)=-D4;
M1(4,1)=70;   M1(4,2)=0;   M1(4,3)=0;   M1(4,4)=140;   M1(4,5)=0;   M1(4,6)=0;
M1(5,1)=0;   M1(5,2)=54;   M1(5,3)=D2;   M1(5,4)=0;   M1(5,5)=156;   M1(5,6)=-D1;
M1(6,1)=0;   M1(6,2)=-D2;   M1(6,3)=-D4;   M1(6,4)=0;   M1(6,5)=-D1;   M1(6,6)=D3;

M1=M1*((d*A1*L1)/420);

%MATRIZ DE TRANSFORMACION PARA EL ELEMENTO 1
KT1(1,1)=cosd(90);   KT1(1,2)=sind(90);   KT1(1,3)=0;
KT1(1,4)=0;   KT1(1,5)=0;   KT1(1,6)=0;
KT1(2,1)=-sind(90);   KT1(2,2)=cosd(90);   KT1(2,3)=0;
KT1(2,4)=0;   KT1(2,5)=0;   KT1(2,6)=0;
KT1(3,1)=0;   KT1(3,2)=0;   KT1(3,3)=1;
KT1(3,4)=0;   KT1(3,5)=0;   KT1(3,6)=0;
KT1(4,1)=0;   KT1(4,2)=0;   KT1(4,3)=0;
KT1(4,4)=cosd(90);   KT1(4,5)=sind(90);   KT1(4,6)=0;
KT1(5,1)=0;   KT1(5,2)=0;   KT1(5,3)=0;
KT1(5,4)=-sind(90);   KT1(5,5)=cosd(90);   KT1(5,6)=0;
KT1(6,1)=0;   KT1(6,2)=0;   KT1(6,3)=0;
KT1(6,4)=0;   KT1(6,5)=0;   KT1(6,6)=1;

%MATRIZ DE RIGIDEZ Y MASA EN COORDENADAS GLOBALES
K1=KT1'*K1*KT1;
```

```

M1=KT1'*M1*KT1;
clc;

%INGRESO DE DATOS DEL ELEMENTO 2 (VIGA)
disp('DATOS ELEMENTO 2')
A2=input('Ingrese el area de la seccion [m2] = ');
I2=input('Ingrese la inercia de la seccion [m4] = ');
n2=input('Ingrese el numero de divisiones del elemento = ');

L2=L/n2;

C1=(A2*E)/L2; C2=(12*E*I2)/L2^3; C3=(6*E*I2)/L2^2; C4=(4*E*I2)/L2;
C5=(2*E*I2)/L2;

%MATRIZ DE RIGIDEZ DEL ELEMENTO 2 EN COORDENADAS GLOBALES
K2(1,1)=C1; K2(1,2)=0; K2(1,3)=0; K2(1,4)=-C1; K2(1,5)=0; K2(1,6)=0;
K2(2,1)=0; K2(2,2)=C2; K2(2,3)=C3; K2(2,4)=0; K2(2,5)=-C2; K2(2,6)=C3;
K2(3,1)=0; K2(3,2)=C3; K2(3,3)=C4; K2(3,4)=0; K2(3,5)=-C3; K2(3,6)=C5;
K2(4,1)=-C1; K2(4,2)=0; K2(4,3)=0; K2(4,4)=C1; K2(4,5)=0; K2(4,6)=0;
K2(5,1)=0; K2(5,2)=-C2; K2(5,3)=-C3; K2(5,4)=0; K2(5,5)=C2; K2(5,6)=-C3;
K2(6,1)=0; K2(6,2)=C3; K2(6,3)=C5; K2(6,4)=0; K2(6,5)=-C3; K2(6,6)=C4;

D1=22*L2; D2=13*L2; D3=4*L2^2; D4=3*L2^2;

%MATRIZ DE MASA DEL ELEMENTO 2 EN COORDENADAS GLOBALES
M2(1,1)=140; M2(1,2)=0; M2(1,3)=0; M2(1,4)=70; M2(1,5)=0; M2(1,6)=0;
M2(2,1)=0; M2(2,2)=156; M2(2,3)=D1; M2(2,4)=0; M2(2,5)=54; M2(2,6)=-D2;
M2(3,1)=0; M2(3,2)=D1; M2(3,3)=D3; M2(3,4)=0; M2(3,5)=D2; M2(3,6)=-D4;
M2(4,1)=70; M2(4,2)=0; M2(4,3)=0; M2(4,4)=140; M2(4,5)=0; M2(4,6)=0;
M2(5,1)=0; M2(5,2)=54; M2(5,3)=D2; M2(5,4)=0; M2(5,5)=156; M2(5,6)=-D1;
M2(6,1)=0; M2(6,2)=-D2; M2(6,3)=-D4; M2(6,4)=0; M2(6,5)=-D1; M2(6,6)=D3;

M2=M2*((d*A2*L2)/420);
clc;

%INGRESO DE DATOS DEL ELEMENTO 3 (COLUMNA 2)
disp('DATOS ELEMENTO 3')
A3=input('Ingrese el area de la seccion [m2] = ');
I3=input('Ingrese la inercia de la seccion [m4] = ');
n3=input('Ingrese el numero de divisiones del elemento = ');

L3=H2/n3;

C1=(A3*E)/L3; C2=(12*E*I3)/L3^3; C3=(6*E*I3)/L3^2; C4=(4*E*I3)/L3;
C5=(2*E*I3)/L3;

%MATRIZ DE RIGIDEZ DEL ELEMENTO 3 EN COORDENADAS LOCALES
K3(1,1)=C1; K3(1,2)=0; K3(1,3)=0; K3(1,4)=-C1; K3(1,5)=0; K3(1,6)=0;
K3(2,1)=0; K3(2,2)=C2; K3(2,3)=C3; K3(2,4)=0; K3(2,5)=-C2; K3(2,6)=C3;
K3(3,1)=0; K3(3,2)=C3; K3(3,3)=C4; K3(3,4)=0; K3(3,5)=-C3; K3(3,6)=C5;
K3(4,1)=-C1; K3(4,2)=0; K3(4,3)=0; K3(4,4)=C1; K3(4,5)=0; K3(4,6)=0;
K3(5,1)=0; K3(5,2)=-C2; K3(5,3)=-C3; K3(5,4)=0; K3(5,5)=C2; K3(5,6)=-C3;
K3(6,1)=0; K3(6,2)=C3; K3(6,3)=C5; K3(6,4)=0; K3(6,5)=-C3; K3(6,6)=C4;

D1=22*L3; D2=13*L3; D3=4*L3^2; D4=3*L3^2;

%MATRIZ DE MASA DEL ELEMENTO 3 EN COORDENADAS LOCALES
M3(1,1)=140; M3(1,2)=0; M3(1,3)=0; M3(1,4)=70; M3(1,5)=0; M3(1,6)=0;
M3(2,1)=0; M3(2,2)=156; M3(2,3)=D1; M3(2,4)=0; M3(2,5)=54; M3(2,6)=-D2;
M3(3,1)=0; M3(3,2)=D1; M3(3,3)=D3; M3(3,4)=0; M3(3,5)=D2; M3(3,6)=-D4;

```

```

M3(4,1)=70; M3(4,2)=0; M3(4,3)=0; M3(4,4)=140; M3(4,5)=0; M3(4,6)=0;
M3(5,1)=0; M3(5,2)=54; M3(5,3)=D2; M3(5,4)=0; M3(5,5)=156; M3(5,6)=-D1;
M3(6,1)=0; M3(6,2)=-D2; M3(6,3)=-D4; M3(6,4)=0; M3(6,5)=-D1; M3(6,6)=D3;

```

```

M3=M3*((d*A3*L3)/420);

```

```

%MATRIZ DE TRANSFORMACION PARA EL ELEMENTO 3

```

```

KT3(1,1)=cosd(-90); KT3(1,2)=sind(-90); KT3(1,3)=0;
KT3(1,4)=0; KT3(1,5)=0; KT3(1,6)=0;
KT3(2,1)=-sind(-90); KT3(2,2)=cosd(-90); KT3(2,3)=0;
KT3(2,4)=0; KT3(2,5)=0; KT3(2,6)=0;
KT3(3,1)=0; KT3(3,2)=0; KT3(3,3)=1;
KT3(3,4)=0; KT3(3,5)=0; KT3(3,6)=0;
KT3(4,1)=0; KT3(4,2)=0; KT3(4,3)=0;
KT3(4,4)=cosd(-90); KT3(4,5)=sind(-90); KT3(4,6)=0;
KT3(5,1)=0; KT3(5,2)=0; KT3(5,3)=0;
KT3(5,4)=-sind(-90); KT3(5,5)=cosd(-90); KT3(5,6)=0;
KT3(6,1)=0; KT3(6,2)=0; KT3(6,3)=0;
KT3(6,4)=0; KT3(6,5)=0; KT3(6,6)=1;

```

```

%MATRIZ DE RIGIDEZ Y MASA EN COORDENADAS GLOBALES

```

```

K3=KT3'*K3*KT3;
M3=KT3'*M3*KT3;

```

```

%TAMAÑO DE LA MATRIZ DE MASA Y RIGIDEZ DEL PORTICO

```

```

N=n1+n2+n3;
KG=zeros((N+1)*3);
MG=zeros((N+1)*3);

```

```

%ENSAMBLE DE LA MATRIZ MASA Y RIGIDEZ DEL PORTICO

```

```

for u=1:1:N
    if u<=n1
        M=M1;
        K=K1;
    else
        if u<=(n1+n2)
            M=M2;
            K=K2;
        else
            if u<=(n1+n2+n3)
                M=M3;
                K=K3;
            end
        end
    end
    end
    KG(3*u-2:3*u+3,3*u-2:3*u+3)=KG(3*u-2:3*u+3,3*u-2:3*u+3)+K;
    MG(3*u-2:3*u+3,3*u-2:3*u+3)=MG(3*u-2:3*u+3,3*u-2:3*u+3)+M;
end

```

```

%ELIMINACION DE LOS GRADOS DE LIBERTAD RESTRINGIDOS EN EL PORTICO

```

```

KG(1:3,:)=[];
KG(:,1:3)=[];
MG(1:3,:)=[];
MG(:,1:3)=[];
t=size(KG);
KG(t(1,1)-2:t(1,1),:)=[];
KG(:,t(1,1)-2:t(1,1))=[];
MG(t(1,1)-2:t(1,1),:)=[];
MG(:,t(1,1)-2:t(1,1))=[];

```

```
%DETERMINACION DE LAS FRECUENCIAS Y MODOS DE VIBRACION
[mv,fr]=eig(KG,MG);
fr=diag(fr)
mv
```

ANEXO D. VALIDACIÓN DE LA SIMULACION DEL PÓRTICO CON ANSYS

G. D. L.	modo_1 MATLAB	modo_1 ANSYS	ERROR
1	0.0060719	0.0060547	0.28%
2	0.00027924	0.00027873	0.18%
3	-0.03774	-0.037639	0.27%
4	0.021008	0.020954	0.26%
5	0.00055818	0.00055717	0.18%
6	-0.059137	-0.059	0.23%
7	0.039966	0.03987	0.24%
8	0.00083654	0.00083502	0.18%
9	-0.06467	-0.064535	0.21%
10	0.058329	0.058197	0.23%
11	0.001114	0.001112	0.18%
12	-0.055418	-0.055312	0.19%
13	0.07191	0.071753	0.22%
14	0.0013903	0.0013878	0.18%
15	-0.033121	-0.033067	0.16%
16	0.072191	0.072032	0.22%
17	-0.0047977	-0.0047917	0.13%
18	-0.0096867	-0.0096772	0.10%
19	0.072397	0.072237	0.22%
20	-0.0053604	-0.0053545	0.11%
21	0.004356	0.0043489	0.16%

G. D. L.	modo_1 MATLAB	modo_1 ANSYS	ERROR
22	0.072526	0.072366	0.22%
23	-0.0031414	-0.0031382	0.10%
24	0.0088276	0.0088182	0.11%
25	0.07258	0.072419	0.22%
26	-0.0010363	-0.001035	0.13%
27	0.0035777	0.0035765	0.03%
28	0.072557	0.072397	0.22%
29	-0.0019744	-0.0019704	0.20%
30	-0.011468	-0.011444	0.21%
31	0.062991	0.062849	0.23%
32	-0.001584	-0.0015807	0.21%
33	-0.035268	-0.035202	0.19%
34	0.045587	0.045478	0.24%
35	-0.0011906	-0.0011881	0.21%
36	-0.04978	-0.049683	0.20%
37	0.02502	0.024954	0.26%
38	-0.00079494	-0.00079329	0.21%
39	-0.050448	-0.050334	0.23%
40	0.0074764	0.0074541	0.30%
41	-0.00039784	-0.00039701	0.21%
42	-0.034309	-0.034215	0.27%

fr_hz MATLAB	fr_hz ANSYS	ERROR
86.691	86.496	0.23%

ANEXO E: CODIGO PORTICO-PSO

```
clc;
clear all;
%INGRESO DE DATOS GENERALES DEL PORTICO
disp('DATOS GENERALES DEL PORTICO')
H1=input('Altura de la columna 1 [m] = ');
H2=input('Altura de la columna 2 [m] = ');
L=input('Longitud del portico [m] = ');
E=input('Modulo de elasticidad del material [N/m2] = ');
d=input('Densidad del material [kg/m3] = ');
clc;

%INGRESO DE DATOS DEL ELEMENTO 1 (COLUMNA 1)
disp('DATOS ELEMENTO 1')
A1=input('Ingrese el area de la seccion [m2] = ');
I1=input('Ingrese la inercia de la seccion [m4] = ');
n1=input('Ingrese el numero de divisiones del elemento = ');

L1=H1/n1;

C1=(A1*E)/L1;   C2=(12*E*I1)/L1^3;   C3=(6*E*I1)/L1^2;   C4=(4*E*I1)/L1;
C5=(2*E*I1)/L1;

%MATRIZ DE RIGIDEZ DEL ELEMENTO 1 EN COORDENADAS LOCALES
K1(1,1)=C1;   K1(1,2)=0;   K1(1,3)=0;   K1(1,4)=-C1;   K1(1,5)=0;   K1(1,6)=0;
K1(2,1)=0;   K1(2,2)=C2;   K1(2,3)=C3;   K1(2,4)=0;   K1(2,5)=-C2;   K1(2,6)=C3;
K1(3,1)=0;   K1(3,2)=C3;   K1(3,3)=C4;   K1(3,4)=0;   K1(3,5)=-C3;   K1(3,6)=C5;
K1(4,1)=-C1;   K1(4,2)=0;   K1(4,3)=0;   K1(4,4)=C1;   K1(4,5)=0;   K1(4,6)=0;
K1(5,1)=0;   K1(5,2)=-C2;   K1(5,3)=-C3;   K1(5,4)=0;   K1(5,5)=C2;   K1(5,6)=-C3;
K1(6,1)=0;   K1(6,2)=C3;   K1(6,3)=C5;   K1(6,4)=0;   K1(6,5)=-C3;   K1(6,6)=C4;

D1=22*L1;   D2=13*L1;   D3=4*L1^2;   D4=3*L1^2;

%MATRIZ DE MASA DEL ELEMENTO 1 EN COORDENADAS LOCALES
M1(1,1)=140;   M1(1,2)=0;   M1(1,3)=0;   M1(1,4)=70;   M1(1,5)=0;   M1(1,6)=0;
M1(2,1)=0;   M1(2,2)=156;   M1(2,3)=D1;   M1(2,4)=0;   M1(2,5)=54;   M1(2,6)=-D2;
M1(3,1)=0;   M1(3,2)=D1;   M1(3,3)=D3;   M1(3,4)=0;   M1(3,5)=D2;   M1(3,6)=-D4;
M1(4,1)=70;   M1(4,2)=0;   M1(4,3)=0;   M1(4,4)=140;   M1(4,5)=0;   M1(4,6)=0;
M1(5,1)=0;   M1(5,2)=54;   M1(5,3)=D2;   M1(5,4)=0;   M1(5,5)=156;   M1(5,6)=-D1;
M1(6,1)=0;   M1(6,2)=-D2;   M1(6,3)=-D4;   M1(6,4)=0;   M1(6,5)=-D1;   M1(6,6)=D3;

M1=M1*((d*A1*L1)/420);

%MATRIZ DE TRANSFORMACION PARA EL ELEMENTO 1
KT1(1,1)=cosd(90);   KT1(1,2)=sind(90);   KT1(1,3)=0;
KT1(1,4)=0;   KT1(1,5)=0;   KT1(1,6)=0;
KT1(2,1)=-sind(90);   KT1(2,2)=cosd(90);   KT1(2,3)=0;
KT1(2,4)=0;   KT1(2,5)=0;   KT1(2,6)=0;
KT1(3,1)=0;   KT1(3,2)=0;   KT1(3,3)=1;
KT1(3,4)=0;   KT1(3,5)=0;   KT1(3,6)=0;
KT1(4,1)=0;   KT1(4,2)=0;   KT1(4,3)=0;
KT1(4,4)=cosd(90);   KT1(4,5)=sind(90);   KT1(4,6)=0;
KT1(5,1)=0;   KT1(5,2)=0;   KT1(5,3)=0;
KT1(5,4)=-sind(90);   KT1(5,5)=cosd(90);   KT1(5,6)=0;
KT1(6,1)=0;   KT1(6,2)=0;   KT1(6,3)=0;
KT1(6,4)=0;   KT1(6,5)=0;   KT1(6,6)=1;

%MATRIZ DE RIGUIDEZ Y MASA EN COORDENADAS GLOBALES
K1=KT1'*K1*KT1;
M1=KT1'*M1*KT1;
```

```

clc;

%INGRESO DE DATOS DEL ELEMENTO 2 (VIGA)
disp('DATOS ELEMENTO 2')
A2=input('Ingrese el area de la seccion [m2] = ');
I2=input('Ingrese la inercia de la seccion [m4] = ');
n2=input('Ingrese el numero de divisiones del elemento = ');

L2=L/n2;

C1=(A2*E)/L2; C2=(12*E*I2)/L2^3; C3=(6*E*I2)/L2^2; C4=(4*E*I2)/L2;
C5=(2*E*I2)/L2;

%MATRIZ DE RIGIDEZ DEL ELEMENTO 2 EN COORDENADAS GLOBALES
K2(1,1)=C1; K2(1,2)=0; K2(1,3)=0; K2(1,4)=-C1; K2(1,5)=0; K2(1,6)=0;
K2(2,1)=0; K2(2,2)=C2; K2(2,3)=C3; K2(2,4)=0; K2(2,5)=-C2; K2(2,6)=C3;
K2(3,1)=0; K2(3,2)=C3; K2(3,3)=C4; K2(3,4)=0; K2(3,5)=-C3; K2(3,6)=C5;
K2(4,1)=-C1; K2(4,2)=0; K2(4,3)=0; K2(4,4)=C1; K2(4,5)=0; K2(4,6)=0;
K2(5,1)=0; K2(5,2)=-C2; K2(5,3)=-C3; K2(5,4)=0; K2(5,5)=C2; K2(5,6)=-C3;
K2(6,1)=0; K2(6,2)=C3; K2(6,3)=C5; K2(6,4)=0; K2(6,5)=-C3; K2(6,6)=C4;

D1=22*L2; D2=13*L2; D3=4*L2^2; D4=3*L2^2;

%MATRIZ DE MASA DEL ELEMENTO 2 EN COORDENADAS GLOBALES
M2(1,1)=140; M2(1,2)=0; M2(1,3)=0; M2(1,4)=70; M2(1,5)=0; M2(1,6)=0;
M2(2,1)=0; M2(2,2)=156; M2(2,3)=D1; M2(2,4)=0; M2(2,5)=54; M2(2,6)=-D2;
M2(3,1)=0; M2(3,2)=D1; M2(3,3)=D3; M2(3,4)=0; M2(3,5)=D2; M2(3,6)=-D4;
M2(4,1)=70; M2(4,2)=0; M2(4,3)=0; M2(4,4)=140; M2(4,5)=0; M2(4,6)=0;
M2(5,1)=0; M2(5,2)=54; M2(5,3)=D2; M2(5,4)=0; M2(5,5)=156; M2(5,6)=-D1;
M2(6,1)=0; M2(6,2)=-D2; M2(6,3)=-D4; M2(6,4)=0; M2(6,5)=-D1; M2(6,6)=D3;

M2=M2*((d*A2*L2)/420);
clc;

%INGRESO DE DATOS DEL ELEMENTO 3 (COLUMNA 2)
disp('DATOS ELEMENTO 3')
A3=input('Ingrese el area de la seccion [m2] = ');
I3=input('Ingrese la inercia de la seccion [m4] = ');
n3=input('Ingrese el numero de divisiones del elemento = ');

L3=H2/n3;

C1=(A3*E)/L3; C2=(12*E*I3)/L3^3; C3=(6*E*I3)/L3^2; C4=(4*E*I3)/L3;
C5=(2*E*I3)/L3;

%MATRIZ DE RIGIDEZ DEL ELEMENTO 3 EN COORDENADAS LOCALES
K3(1,1)=C1; K3(1,2)=0; K3(1,3)=0; K3(1,4)=-C1; K3(1,5)=0; K3(1,6)=0;
K3(2,1)=0; K3(2,2)=C2; K3(2,3)=C3; K3(2,4)=0; K3(2,5)=-C2; K3(2,6)=C3;
K3(3,1)=0; K3(3,2)=C3; K3(3,3)=C4; K3(3,4)=0; K3(3,5)=-C3; K3(3,6)=C5;
K3(4,1)=-C1; K3(4,2)=0; K3(4,3)=0; K3(4,4)=C1; K3(4,5)=0; K3(4,6)=0;
K3(5,1)=0; K3(5,2)=-C2; K3(5,3)=-C3; K3(5,4)=0; K3(5,5)=C2; K3(5,6)=-C3;
K3(6,1)=0; K3(6,2)=C3; K3(6,3)=C5; K3(6,4)=0; K3(6,5)=-C3; K3(6,6)=C4;

D1=22*L3; D2=13*L3; D3=4*L3^2; D4=3*L3^2;

%MATRIZ DE MASA DEL ELEMENTO 3 EN COORDENADAS LOCALES
M3(1,1)=140; M3(1,2)=0; M3(1,3)=0; M3(1,4)=70; M3(1,5)=0; M3(1,6)=0;
M3(2,1)=0; M3(2,2)=156; M3(2,3)=D1; M3(2,4)=0; M3(2,5)=54; M3(2,6)=-D2;
M3(3,1)=0; M3(3,2)=D1; M3(3,3)=D3; M3(3,4)=0; M3(3,5)=D2; M3(3,6)=-D4;
M3(4,1)=70; M3(4,2)=0; M3(4,3)=0; M3(4,4)=140; M3(4,5)=0; M3(4,6)=0;

```

```

M3(5,1)=0;    M3(5,2)=54;    M3(5,3)=D2;    M3(5,4)=0;    M3(5,5)=156;    M3(5,6)=-D1;
M3(6,1)=0;    M3(6,2)=-D2;    M3(6,3)=-D4;    M3(6,4)=0;    M3(6,5)=-D1;    M3(6,6)=D3;

M3=M3*((d*A3*L3)/420);

%MATRIZ DE TRANSFORMACION PARA EL ELEMENTO 3
KT3(1,1)=cosd(-90);    KT3(1,2)=sind(-90);    KT3(1,3)=0;
KT3(1,4)=0;            KT3(1,5)=0;            KT3(1,6)=0;
KT3(2,1)=-sind(-90);   KT3(2,2)=cosd(-90);   KT3(2,3)=0;
KT3(2,4)=0;            KT3(2,5)=0;            KT3(2,6)=0;
KT3(3,1)=0;            KT3(3,2)=0;            KT3(3,3)=1;
KT3(3,4)=0;            KT3(3,5)=0;            KT3(3,6)=0;
KT3(4,1)=0;            KT3(4,2)=0;            KT3(4,3)=0;
KT3(4,4)=cosd(-90);    KT3(4,5)=sind(-90);    KT3(4,6)=0;
KT3(5,1)=0;            KT3(5,2)=0;            KT3(5,3)=0;
KT3(5,4)=-sind(-90);   KT3(5,5)=cosd(-90);   KT3(5,6)=0;
KT3(6,1)=0;            KT3(6,2)=0;            KT3(6,3)=0;
KT3(6,4)=0;            KT3(6,5)=0;            KT3(6,6)=1;

%MATRIZ DE RIGIDEZ Y MASA EN COORDENADAS GLOBALES
K3=KT3'*K3*KT3;
M3=KT3'*M3*KT3;

%TAMAÑO DE LA MATRIZ DE MASA Y RIGIDEZ DEL PORTICO
N=n1+n2+n3;
KG=zeros((N+1)*3);
MG=zeros((N+1)*3);

ss=0;
Z=[];
clc;

%INGRESO DE ELEMETOS CON DAÑO
X=input('Desea reducir la rigidez de algun elemento? (si) (no) ','s');
while X=='si'
    ss=ss+1;
    z=input('Elemento que desea reducir la rigidez = ');
    p=input('Porcentaje de reduccion = ');
    Z(ss,1)=z;
    clc;
    Z(ss,2)=1-p/100
    X=input('Desea reducir la rigidez de algun otro elemento? (si) (no) ','s');
end

clc;
tz=size(Z);

%ENSAMBLE DE LA MATRIZ MASA Y RIGIDEZ DEL PORTICO
for u=1:1:N
    if u<=n1
        M=M1;
        K=K1;
        for ss=1:1:tz(1,1)
            if u==Z(ss,1)
                K=K*Z(ss,2);
            end
        end
    else
        if u<=(n1+n2)
            M=M2;

```



```

p=(b-a)/10;
c1=2;
c2=2;
w=1;
x0=[];
v0=[];

%DISTRIBUCION INICIAL DE DAÑOS X0 Y VELOCIDADES V0
for k=1:1:np
    for h=1:1:d
        a1=rand;
        x0(k,h)=a+a1*(b-a);
        if x0(k,h)>b
            x0(k,h)=b;
        elseif x0(k,h)<a
            x0(k,h)=a;
        end
        b1=rand;
        v0(k,h)=(a+b1*(b-a))/1;
        if v0(k,h)>p
            v0(k,h)=p;
        elseif v0(k,h)<-p
            v0(k,h)=-p;
        end
    end
end

e=0;
des=10;

while des>E
    e=e+1;
    for k=1:1:np
        if e==1
            L(k,1:d)=x0(k,1:d);
            KG=zeros((N+1)*3);
            MG=zeros((N+1)*3);

            %ENSAMBLE DE LA MATRIZ MASA Y RIGIDEZ DEL PORTICO CONSIDERANDO
            %LOS DAÑOS INICIALES XO
            for u=1:1:N
                if u<=n1
                    M=M1;
                    K=K1*L(k,u);
                else
                    if u<=(n1+n2)
                        M=M2;
                        K=K2*L(k,u);
                    else
                        if u<=(n1+n2+n3)
                            M=M3;
                            K=K3*L(k,u);
                        end
                    end
                end
            end
            KG(3*u-2:3*u+3,3*u-2:3*u+3)=KG(3*u-2:3*u+3,3*u-2:3*u+3)+K;
            MG(3*u-2:3*u+3,3*u-2:3*u+3)=MG(3*u-2:3*u+3,3*u-2:3*u+3)+M;
        end
    end

    %ELIMINACION DE LOS GRADOS DE LIBERTAD RESTRINGIDOS EN EL

```

```

%PORTICO
KG(1:3,:)=[];KG(:,1:3)=[];
MG(1:3,:)=[];MG(:,1:3)=[];
t=size(KG);
KG(t(1,1)-2:t(1,1),:)=[];
KG(:,t(1,1)-2:t(1,1))=[];
MG(t(1,1)-2:t(1,1),:)=[];
MG(:,t(1,1)-2:t(1,1))=[];

%CALCULO DE LA FUNCION OBJETIVO PARA LOS DAÑOS X0
FO=[];
for u=1:1:nm
    FO(:,u)=KG*MV(:,u)-FR(u,1)*MG*MV(:,u);
end
FO_TOT=norm(FO);
ym_local(k,1)=FO_TOT;

%COMPARACION DE LA FUNCION OBJETIVO CONSIDERANDO LOS DAÑOS X0 Y X1
%PARA LA ASIGNACION DE LOS MEJORES PUNTOS Y EVALUACIONES LOCALES
else
    if yx1(k,1)<ylocal(k,1)
        L(k,1:d)=x1(k,1:d);
        ylocal(k,1)=yx1(k,1);
    else
        L(k,1:d)=L(k,1:d);
        ylocal(k,1)=ym_local(k,1);
    end
end
end

%VALOR MINIMO DE LA FUNCION
ymin=min(ylocal);

%PUNTO DONDE LA FUNCION ES MINIMA
for k=1:1:np
    if ymin==ym_local(k,1)
        G=L(k,:);
    end
end

%ACTUALIZACION DE X0 Y V0
if e>1
    x0=x1;
    v0=v1;
end

%DETERMINACION DE X1 Y V1
for k=1:1:np
    for h=1:1:d
        r1=rand;
        r2=rand;
        v1(k,h)=w*v0(k,h)+(c1*r1*(L(k,h)-x0(k,h))/1)+(c2*r2*(G(1,h)-...
        x0(k,h))/1);
        if v1(k,h)>p
            v1(k,h)=p;
        elseif v1(k,h)<-p
            v1(k,h)=-p;
        end
        x1(k,h)=x0(k,h)+v1(k,h)*1;
        if x1(k,h)>b

```

```

        x1(k,h)=b;
    elseif x1(k,h)<a
        x1(k,h)=a;
    end
end
end
for k=1:1:np
    KG=zeros((N+1)*3);
    MG=zeros((N+1)*3);

%ENSAMBLE DE LA MATRIZ MASA Y RIGIDEZ DEL PORTICO CONSIDERANDO LOS DAÑOS X1
    for u=1:1:N
        if u<=n1
            M=M1;
            K=K1*x1(k,u);
        else
            if u<=(n1+n2)
                M=M2;
                K=K2*x1(k,u);
            else
                if u<=(n1+n2+n3)
                    M=M3;
                    K=K3*x1(k,u);
                end
            end
        end
        KG(3*u-2:3*u+3,3*u-2:3*u+3)=KG(3*u-2:3*u+3,3*u-2:3*u+3)+K;
        MG(3*u-2:3*u+3,3*u-2:3*u+3)=MG(3*u-2:3*u+3,3*u-2:3*u+3)+M;
    end

%ELIMINACION DE LOS GRADOS DE LIBERTAD RESTRINGIDOS EN EL PORTICO
    KG(1:3,:)=[];
    KG(:,1:3)=[];
    MG(1:3,:)=[];
    MG(:,1:3)=[];
    t=size(KG);
    KG(t(1,1)-2:t(1,1),:)=[];
    KG(:,t(1,1)-2:t(1,1))=[];
    MG(t(1,1)-2:t(1,1),:)=[];
    MG(:,t(1,1)-2:t(1,1))=[];

%CALCULO DE LA FUNCION OBJETIVO PARA LOS DAÑOS X1
    FO=[];
    for u=1:1:nm
        FO(:,u)=KG*MV(:,u)-FR(u,1)*MG*MV(:,u);
    end
    FO_TOT=norm(FO);
    yx1(k,1)=FO_TOT;
end

%REDUCCION LINEAL DE w
w=w*0.98;

%ALMACENAMIENTO DE DATOS PARA DETERMINAR EL CRITERIO DE PARADA
ds(1,e)=ymin;

%CALCULO DEL CRITERIO DE PARADA
if e>rd
    des=std(ds(1,e-rd:e));
end
end

```

```
end

duracion=etime(clock,to)
iteraciones=e
ymin
elementos_danio(:,1)=(1:N);
elementos_danio(:,2)=G'
ym_local
```