

Universidad Central “Marta Abreu” de Las Villas

Facultad de Construcciones

Departamento de Ingeniería Civil



## **TRABAJO DE DIPLOMA**

Calibración de modelos estructurales de torres de telecomunicaciones autosoportadas  
usando la OAPI SAP2000 – MATLAB

Autor: Alejandro Fuentes Díaz.

Tutor:

Ing. Leonardo Rodríguez González.

Cotutor:

Dr. Ing. Ernesto Chagoyén Méndez.

Santa Clara

2017

"Año 58 de la Revolución"

# DEDICATORIA

A mis padres Domingo Fuentes y María Esther Díaz.

A mi segunda madre, Ángela María Díaz.

# AGRADECIMIENTOS

A mi padre, por ser mi mano derecha, por amarme y apoyarme en todos los momentos.

A mi madre, por su amor infinito.

A mi novia, Lenay Rodríguez Alonso, por su amor sincero y por alegrarme, incluso, hasta en los días más difíciles. Por su ayuda en la elaboración de este trabajo. Por compartir los últimos 8 años de su vida conmigo y por haber sido en ese tiempo la otra gran parte que completa mi felicidad.

A mi tía, por su dedicación y muestras de inmenso cariño.

A la familia de mi novia, por acogerme como uno más de su familia y brindarme cariño y apoyo en todo momento. A ellos mi eterno agradecimiento.

A toda mi familia, por estar a mi lado siempre.

A mis amigos, gracias por su sinceridad y apoyo incondicional.

Me gustaría ofrecer un agradecimiento muy especial a:

A mi tutor el Ing. Leonardo Rodríguez González, por su atención, consejos y gran ayuda en el desarrollo de este trabajo el cual se ha ganado mi respeto y admiración en este tiempo trabajando juntos.

A mi cotutor el Dr. Ernesto L. Chagoyén Méndez, por su asesoría y conocimientos brindados y por ser un ejemplo de profesional a seguir por las nuevas generaciones.

## **Resumen**

Esta investigación está dirigida a la calibración de una torre de telecomunicaciones auto soportada modelada en SAP2000, utilizando, también, los métodos de optimización que posee MATLAB. Para ello se utilizan las posibilidades que brinda la interface OAPI SAP2000 entre SAP2000 y MATLAB.

Este proceso se realiza a partir de los parámetros dinámicos obtenidos de un modelo al que se le ha aplicado daño conocido en posiciones arbitrarias, se identifican los parámetros del comportamiento dinámico “medido”, las frecuencias de oscilación natural y las formas modales principales de oscilaciones propias, necesarios para la formulación de la función objetivo del vector compuesto por los residuos de los parámetros modales entre la respuesta modelada y la “medida”. Esta función se minimiza empleando técnicas de optimización programadas en MATLAB, proporcionando el modelo estructural calibrado y permite estimar parámetros estructurales del mismo.

Al contar con el modelo calibrado en SAP 2000, entonces se le pueden variar diferentes estados de cargas obteniéndose una respuesta del modelo mucho más cercana a la respuesta real de la estructura.

**Palabras claves:** Torres autosoportadas, modelación, calibración, parámetros dinámicos, optimización.

## **Abstract**

This research aims to update mechanical models of free-standing towers (known as calibration or model updating) modeled in SAP2000, taking advantage of the implementation of the optimization methods that has MATLAB. In order to do this, the possibilities offered by the SAP2000 OAPI interface between SAP2000 and MATLAB are used.

The dynamic parameters were obtained from a model that a known damage at arbitrary positions has been inflicted, the Eigen frequencies and the main mode shapes are identified as the parameters of the dynamic "measured" behavior necessities for the formulation of the objective function of the vector composed by residues of these two modal parameters between the modeled response and the "measured". This function is minimized using optimization techniques programmed in MATLAB (nonlinear least squares), which at the end, provide the structural model calibrated for estimating structural parameters.

Having the calibrated model in SAP2000, different load states can be changed and a model response closer to the response of the real structure would be obtained.

**Key words:** Free-standing towers, modeling, update, dynamic parameters, optimization.

# INDICE

Introducción .....	1
Objeto de estudio.....	2
Campo de acción.....	2
Problema científico .....	3
Pregunta de investigación .....	3
Objetivo general.....	3
Objetivos específicos.....	3
Hipótesis .....	4
Novedad científica .....	4
Justificación de la investigación.....	4
Métodos de investigación .....	5
Organización del informe.....	6
CAPÍTULO I: Estado del conocimiento sobre calibración de modelos estructurales de torres de telecomunicaciones autosoportadas .....	7
1.1    Introducción.....	7
1.2    Torres de telecomunicaciones .....	8
1.2.1    Clasificación de las torres de telecomunicaciones.....	9
1.2.1.1    Torres arriostradas o atirantadas.....	9
1.2.1.2    Torres Monopolo.....	10
1.2.1.3    Torres autosoportadas.....	11
1.2.2    Elementos constructivos .....	12
1.2.2.1    Las antenas .....	13
1.3    Calibración de modelos estructurales de elementos finitos .....	14
1.3.1    Obtención de los datos experimentales .....	16
1.3.2    Instrumentos para Medición de Vibraciones .....	17
1.3.3    Métodos experimentales para la medición de vibraciones .....	17
1.3.4    Procesamiento de las señales obtenidas en ensayos modales.....	19
1.3.5    Extracción de parámetros modales.....	19
1.3.6    Modelos de elementos finitos de las estructuras (FEM) .....	20
1.3.7    Validación.....	23
1.3.7.1    Model Assurance Criterion (MAC) .....	23
1.3.8    Optimización en el proceso de calibración de modelos estructurales ....	26
1.3.8.1    Análisis de sensibilidad.....	26
1.3.8.2    Conceptos básicos sobre optimización.....	27
1.3.8.3    Función objetivo.....	28
1.3.8.4    Técnicas de optimización en ingeniería.....	29
1.3.8.5    Problemas de mínimos cuadrados .....	30
1.4    Empleo de softwares computacionales para la calibración de modelos .....	31
1.4.1    MATLAB.....	31
1.4.1.1    Toolbox de Optimización .....	32

1.4.2	SAP2000 .....	34
1.4.2.1	OAPI SAP2000 .....	34
1.4.2.2	Uso de la OAPI SAP2000 .....	35
1.4.2.3	Funciones de la OAPI SAP2000.....	36
1.5	Conclusiones parciales del capítulo: .....	36
Capítulo II: Procedimiento automatizado para la calibración de modelos de torres autosoportadas utilizando la interface OAPI SAP2000 y los recursos de MATLAB, basado en resultados de pruebas modales.....		
2.1	Introducción.....	38
2.2	Modelación de torres de telecomunicaciones autosoportadas empleando SAP2000v16. ....	39
2.2.1	Variantes de modelación en Sap2000v16 .....	40
2.3	Optimización para la calibración de MEF.....	44
2.3.1	Dos estrategias: Búsqueda Lineal (Line Search) y Región de Confianza (Trust Region).....	44
2.3.2	Clasificación de los algoritmos locales.....	45
2.3.3	Problema de mínimos cuadrados .....	46
2.3.4	Función lsqnonlin en MATLAB.....	48
2.3.4.1	Ecuación, sintaxis y resultados.....	50
2.4	Procedimiento automatizado de optimización para la calibración implementado en MATLAB.....	53
2.4.1	Parámetros que intervienen en la optimización .....	54
2.4.2	Script (main_mdof_torre.m) .....	55
2.4.3	Script (data_mdof_torre.m) .....	57
2.4.4	Script (upd_mdof_torre.m) .....	58
2.4.5	Script (eigfem_mdof_torre.m) .....	61
2.4.6	Script (sens_anal_torre.m).....	64
2.5	Conclusiones parciales del capítulo .....	66
CAPITULO III: Aplicación del procedimiento automatizado para la calibración a un caso de estudio de una torre autosoportada en Cuba. ....		
3.1	Introducción.....	67
3.2	Modelación de la torre objeto de estudio .....	70
3.3	Localización de los sensores en la torre .....	72
3.4	Declaración de los parámetros que intervienen en la optimización .....	74
3.5	Obtención de los parámetros dinámicos experimentales.....	76
3.6	Métodos de optimización empleados. ....	77
3.7	Obtención y análisis de los resultados. Validación de los modelos calibrados obtenidos.....	78
3.7.1	Resultados (Caso de estudio 1).....	79
3.7.2	Resultados (Caso de estudio 2).....	82
3.7.3	Resultados (Caso de estudio 3).....	84
3.7.4	Resultados (Caso de estudio 4).....	87

3.7.5	Análisis de los resultados.....	91
3.7.6	Validación de los modelos .....	92
3.8	Conclusiones parciales del capítulo .....	94
	Conclusiones .....	95
	Recomendaciones .....	96
	BIBLIOGRAFIA Y REFERENCIAS.....	97
	ANEXOS.....	104
	Programación del script (main_mdof_torre):.....	104
	Programación del script (data_mdof_torre):.....	110
	Programación del script (upd_mdof_torre):.....	111
	Programación del script (eigfem_mdof_torre):.....	114
	Programación del script (sens_anal_torre): .....	119



## Introducción

El presente trabajo está enfocado hacia la profundización en los métodos y procedimientos necesarios para la combinación de la modelación de torres de telecomunicaciones autosoportadas y el procesamiento de las señales derivadas de pruebas modales, con el propósito de identificar la composición de frecuencias y las ordenadas de las formas principales de oscilaciones propias de la estructura de la torre, que resultan la base para la determinación de los parámetros estructurales que definen su comportamiento utilizando, para ello, la OAPI SAP2000-MATLAB.

Las torres de telecomunicaciones son de vital importancia en el desarrollo de un país, estas son el soporte de grandes y pesadas antenas que transmiten señales tanto para la televisión, como para la telefonía celular. El creciente avance de las comunicaciones ha conllevado al desarrollo de estas torres con el objetivo de buscar nuevos modelos y de lograr torres más seguras, resistentes y capaces de soportar mayores esfuerzos sin llegar a perder su integridad estructural debido a los grandes costos de las mismas y a la infraestructura necesaria para su construcción.

Sus principales vulnerabilidades son la carga de viento, especialmente en el caso de ciclones, los sismos y la colocación de un mayor número de antenas, sustitución de las existentes por otras más grandes y pesadas o el cambio de la disposición de las mismas lo que provoca la modificación de las fuerzas interiores de los elementos de la torre, ya que por una parte las antenas actúan como pantalla frente al viento haciendo que se generen fuerzas adicionales sobre la estructura y por otra la concentración de la masa en diferentes partes de la estructura modifica su respuesta dinámica. (Parnás et al., 2012)

Existen dos tipos fundamentales de torres de telecomunicaciones según su tipología estructural, las atirantadas y las autosoportadas, estas últimas objeto de estudio en este trabajo. Las torres autosoportadas tienen el agravante de presentar características estructurales de bajo amortiguamiento y poca rigidez ante la acción de las cargas transversales. (Parnás et al., 2012)

También cabe resaltar que en los últimos años se han desarrollado numerosas investigaciones acerca de las torres autosoportadas en Cuba, todas estas tratan temas como, diseño de algunos de sus componentes, estudios de los efectos de las cargas de viento y sismo, propuestas de normas para su diseño, entre otros.

El empleo de la OAPI SAP2000-MATLAB para la calibración de modelos FE en el caso de torres autosoportadas utilizando métodos de optimización es totalmente novedoso en el país, con valor agregado pues evita la necesidad de realizar ensayos destructivos que se examinan posteriormente en un laboratorio, además de permitir obtener un modelo calibrado en Sap 2000 que permite variarle estados de cargas y obtener respuestas mucho más parecidas a las experimentadas por la estructura real lo que sería importante ,por ejemplo, para el monitoreo de daño en este tipo de estructuras, su mantenimiento, y estimar cuál sería su respuesta ante la colocación de nuevas antenas o cambio en la disposición de las ya existentes.

Utilizando la interacción entre estos dos poderosos programas, es posible estimar o ajustar los parámetros de un modelo mecánico de una estructura que es, en general, la calibración del modelo real. Este proceso minimiza las diferencias entre los valores de las respuestas reales de una función objetivo de la estructura, obtenidos de mediciones experimentales, y las respuestas del modelo, lo que facilita que la toma de decisiones ante situaciones de cargas o estado de la estructura que no puedan ser experimentalmente ensayados esté basada en modelos más acertados. (Aspiro, 2016)

Para este caso la función objetivo adoptada es el vector residual en función de los parámetros de rigidez de la estructura. Este vector está compuesto por los residuos por frecuencias naturales y los residuos por formas modales. Su sintaxis se programa al final de la función de optimización creada en MATLAB para interactuar con el modelo mecánico de la torre en SAP2000 gracias al empleo de la OAPI SAP2000. El procedimiento termina cuando se haya alcanzado un alto grado de correlación entre ambas respuestas, la medida y la modelada.

## **Objeto de estudio**

Los métodos para la calibración y validación de un modelo de torre auto soportada a la que se le ha realizado ensayos modales.

## **Campo de acción**

La instrumentación combinada con la modelación mecánica, el análisis estructural de torres autosoportadas y la calibración de los modelos mecánicos de estas torres.

## **Problema científico**

En Cuba, se han producido fallos estructurales de torres de telecomunicaciones autosoportadas, algunos podrían ser debido al uso de métodos equivalentes a la hora de realizar la modelación de las cargas, como por ejemplo la del viento, que se hace una equivalencia a carga estática cuando en realidad no lo es, falta de mantenimiento o bien por no contar con equipamientos o procedimientos modernos para estimar correctamente lo que sucede verdaderamente en la estructura, entre otros.

En los últimos años se han realizado algunos estudios, principalmente en la CUJAE, en La Habana, con el objetivo de entender y aportar, un poco más con cada uno de ellos, sobre este tipo de estructuras. Estos estudios han tratado sobre el análisis del viento, diseño de elementos y conexiones, análisis dinámico entre otros.

Por lo que existe la necesidad de seguir investigando en la búsqueda de procedimientos que sean factibles de implementar y que permitan estimar con mayor precisión la respuesta de estas estructuras a partir de contar con modelos para estudiar mejor su comportamiento y monitoreo de los daños en las mismas que son de gran importancia económica y social para el país.

## **Pregunta de investigación**

¿En qué medida un procedimiento automatizado con uso de la interacción OAPI SAP2000 – MATLAB proporciona modelos calibrados de torres de telecomunicaciones autosoportadas?

## **Objetivo general**

Aplicar un procedimiento automatizado para la calibración de modelos mecánicos de torres autosoportadas, a través de la biblioteca de funciones de optimización de MATLAB y la OAPI SAP2000, con variantes de variables a calibrar.

## **Objetivos específicos**

1. Determinar el estado del conocimiento sobre calibración de modelos de torres autosoportadas basado en ensayos modales, empleando la OAPI SAP2000 y las potencialidades de MATLAB.

2. Explicar un procedimiento automatizado, con variantes de variables de diseño, para la calibración de modelos mecánicos de torres auto-soportadas, empleando la OAPI SAP2000-MATLAB.
3. Aplicar el procedimiento a un caso de estudio de una torre en Cuba, interpretando los resultados obtenidos.

## **Hipótesis**

Es posible aplicar un procedimiento automatizado, con distintas variantes de variables de diseño, que permita calibrar un modelo en SAP2000 de la estructura de una torre auto-soportada, logrando que esté presente la misma respuesta que la estructura real.

## **Novedad científica**

El empleo de la OAPI SAP2000 para realizar la calibración de torres de telecomunicaciones en Cuba es totalmente novedoso, pues hasta el momento los investigadores del país efectúan el proceso de calibración de los modelos de forma manual, convirtiéndose en un proceso prolongado y agotador, que muchas veces no garantiza la fiabilidad estructural requerida, es decir, que los resultados obtenidos no sean los más ajustados al comportamiento estructural real.

## **Justificación de la investigación**

Con el empleo de la OAPI Sap2000 para la calibración de modelos de torres auto-soportadas se logra evitar el proceso de realizarlo de forma manual, donde este último se torna muy lento y agotador y donde no siempre se logra una exactitud fiable de los resultados.

También, con la creación de un procedimiento automatizado, a través de la OAPI Sap2000 y haciendo uso de técnicas de optimización en MATLAB, se pueden llevar a cabo investigaciones posteriores donde se extienda el campo a otros tipos de estructuras, o se perfeccionen y enriquezcan los estudios ya elaborados aportando de esta manera a la rama de la ingeniería civil en Cuba y acercándonos a estos procedimientos que son muy usados en muchos países en la actualidad.

De esta manera se realiza un aporte a los conocimientos sobre torres auto-soportadas, que constituyen el soporte de tipos de antenas (radio, televisión, satelitales, comunicación en general), que son difíciles y caras de construir, de ahí que su evaluación y diagnóstico sea

imprescindible a la hora de determinar su comportamiento bajo la acción de diversas cargas.

También desde el punto de vista social se estaría contribuyendo debido a que todos los esfuerzos que se realicen estudiando nuevos métodos para la conservación, diseño, detección de daños, etcétera, en este tipo de estructuras y que sean una vía factible para su implementación son, al final, un aporte al ahorro económico del país y al bienestar general de la población.

## **Métodos de investigación**

- **Métodos del nivel teórico:**

Método histórico lógico, para el establecimiento del estado actual del conocimiento sobre los métodos para realizar la modelación de estructuras de torres autosoportadas desde el punto de vista dinámico y los métodos para el procesamiento de las señales obtenidas de pruebas de cargas dinámicas en las mismas, con vistas a la identificación de algunos de los parámetros del comportamiento dinámico de la estructura como son las frecuencias de oscilación natural, los modos o formas principales de oscilaciones propias, entre otros, mediante el cual se realizará un análisis histórico de la temática y se determinará cuáles han sido las tendencias; se definirán los conceptos claves sobre estas temáticas así como se elaborará una síntesis de las principales concepciones al respecto, identificando los aspectos positivos y realizando análisis críticos de las deficiencias.

La modelación, para estudiar las diferentes variables que inciden en el comportamiento dinámico de la torre, mediante el empleo de los softwares existentes (SAP 2000 v.16, MATLAB 2015).

- **Métodos del nivel empírico:**

Se utilizará como diseño el estudio de caso único, en una torre de telecomunicaciones auto soportada donde se realizará un monitoreo de la misma mediante ensayos modales y estáticos, así como un levantamiento de todas las características de la misma, aplicándose el método de observación participante y la técnica de notas de campo.

- **Métodos del nivel matemático estadístico:**

El análisis numérico (transformada rápida de Fourier, ajuste de curvas, etc.) de las señales obtenidas de experimentación modal de estructuras reales de torres autos portadas, se utilizará para identificar los parámetros dinámicos del sistema estructural.

Del análisis comparativo de la respuesta modelada y medida de la estructura de la torre estudiada, se pueden inferir conclusiones sobre el comportamiento dinámico real y modelado de dichas estructuras, su nivel de daño estructural, la localización de los mismos, etc.

## **Organización del informe**

- Resumen
- Abstract
- Introducción
- Capítulo I: Estado del conocimiento sobre calibración de modelos estructurales de torres de telecomunicaciones autosoportadas.
- Capítulo II: Procedimiento automatizado para la calibración de modelos de torres autosoportadas utilizando la interface OAPI SAP2000 y los recursos de MATLAB, basado en resultados de pruebas modales.
- Capítulo III: Aplicación del procedimiento automatizado para la calibración a un caso de estudio de una torre autosoportada en Cuba.
- Conclusiones
- Recomendaciones
- Bibliografía y Referencias

# **CAPÍTULO I: Estado del conocimiento sobre calibración de modelos estructurales de torres de telecomunicaciones autosoportadas**

En este capítulo se abordarán temas afines a la investigación que se lleva a cabo sobre la calibración de modelos de torres de telecomunicaciones autosoportadas. Se definen los tipos de torres de telecomunicaciones, sus principales diferencias y características más fundamentales, principalmente, en las torres autosoportadas y se explican los principales métodos empleados, procedimientos de pruebas, instrumentación implementada, así como una descripción de los softwares y sus utilidades en el proceso de calibración o actualización de los modelos mecánicos de las torres, así como la validación del mismo.

## **1.1 Introducción**

En la ingeniería civil y estructural, los modelos por elementos finitos (FE, por sus siglas en inglés) son usados para analizar, por ejemplo, la distribución de fuerzas y deformaciones en una estructura en diferentes estados límites, o las respuestas de las estructuras debido a cargas dinámicas tales como terremotos, viento y tráfico. Estos análisis también pueden ser usados para diseñar completamente la estructura, o algunos de sus componentes, o añadirle algunas ampliaciones o mejoras.

Para realizar estos propósitos, es imperativo la validación del modelo numérico empleado, aunque siempre hay, sin embargo, numerosas propiedades del sistema que son desconocidas o imposibles de saber con certeza (propiedades de los materiales, propiedades geométricas, apoyos, condiciones de cargas) por lo que hay que realizar suposiciones inevitablemente. Además, debido a la carencia de conocimientos u otras restricciones se requiere de hacer simplificaciones en la estructura del modelo o estas se cometen de manera inconsciente. Estos problemas pueden ocasionar que el modelo numérico esté lejos de su propósito y exactitud requeridos. (SIMOEN, 2013)

Todo lo anterior ha conllevado al desarrollo de las técnicas de actualización de modelos ("Model Updating"), también llamadas calibración de modelos, y en términos más generales, identificación de parámetros. Originalmente desarrolladas en las industrias aeroespacial y de control de sistemas, la identificación de parámetros es ahora comúnmente usada en las ingenierías civil, mecánica y eléctrica. El objetivo de la identificación de parámetros, en la ingeniería estructural, es identificar las características dinámicas de una estructura ya construida con gran exactitud. Estas características

dinámicas son típicamente las frecuencias naturales, las formas modales y el amortiguamiento. (Morris, 2009)

En el proceso de calibración son tomados en cuenta conceptos de desgaste, o cambios en diferentes propiedades de los materiales que componen la estructura, los cuales pudieron aparecer al pasar el tiempo. Un ejemplo de esto se puede observar en el cambio del módulo de elasticidad del material.

La herramienta computacional OAPI SAP2000, permite ampliar significativamente el uso de métodos de optimización incorporadas a MATLAB para realizar dicho proceso de calibración más allá del marco de análisis estructural establecido actualmente (Sextos and Balafas, 2011).

A partir de estas características dinámicas obtenidas se hace necesario la realización del modelo mecánico de la torre objeto de estudio empleando un software potente para ello, este puede ser SAP2000, ABAQUS, ANSYS, entre otros, que a su vez utilizan el MEF. Este modelo estará implementado desde MATLAB para realizar su calibración empleando las técnicas de optimización disponibles en su toolbox.

Con la obtención del modelo calibrado se facilita que la toma de decisiones acerca del comportamiento estructural ante situaciones de carga o estado de la estructura que no puedan ser experimentalmente ensayados, esté basada en modelos más acertados en términos de evaluación de cargas, detección de desperfectos y daños estructurales, rehabilitación de estructuras, etc.

## **1.2 Torres de telecomunicaciones**

Las estructuras utilizadas en telecomunicaciones sirven para la transmisión de energía eléctrica, así como la transmisión de señales, como en el caso de los teléfonos celulares y la televisión. Estas estructuras deben soportar diversos dispositivos, como antenas de transmisión y equipos para telecomunicaciones, entre otros.

La necesidad creciente de estructuras altas que permitan la mejor comunicación de la radio y la televisión, así como el desarrollo de la telefonía celular han producido una aceleración en la construcción de torres de celosía como soporte de antenas y han ampliado las funciones de las ya existentes. Estas estructuras son relativamente jóvenes, surgen en la segunda década del siglo XX con el nacimiento de las transmisiones de radio y a partir de la década del 50 del mismo siglo, con la invención de la televisión, su uso es extendido como soporte de antenas. (Parnás et al., 2012)



La estructura fundamental es de celosía debido a que es una estructura formada por elementos lineales unidos entre sí por sus extremos y logrando un arreglo espacial en forma de reticulado cinemáticamente invariable. Esta forma constructiva permite gran resistencia con poco consumo de material y por su permeabilidad reduce las fuerzas del viento que son generadas sobre la estructura. El material más utilizado es el acero. (González, 2010)

### 1.2.1 Clasificación de las torres de telecomunicaciones

Su estructuración puede variar según las necesidades y las condiciones del sitio en donde se vayan a colocar. Entre las más comunes se encuentran las torres arriostradas (torres con tirantes), torres autosoportadas y monopolos (figura 1.1). La altura de estas estructuras depende de la altura requerida para poder suministrar una correcta transmisión. La figura 1.1 presenta tres tipos de torres de telecomunicaciones. (Méndez, 2015)

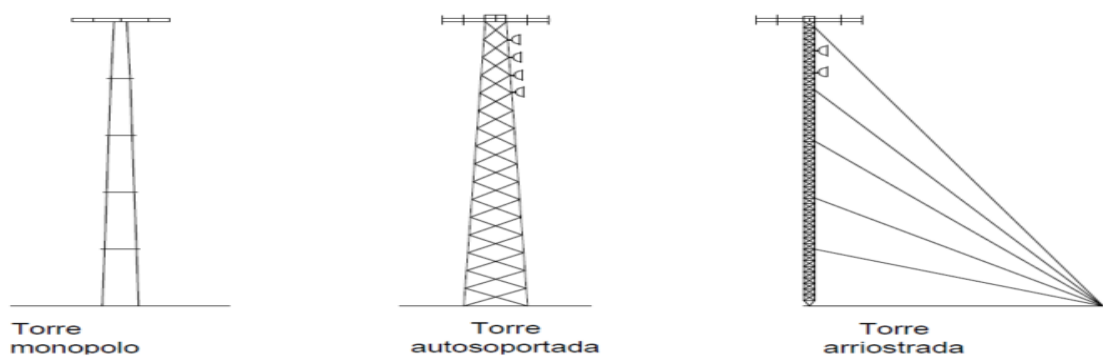


Fig. 1.1 Ejemplos de torres de telecomunicaciones. Fuente: (Méndez, 2015)

Los elementos estructurales empleados para la construcción de las torres dependen de su tipo, algunos elementos comúnmente empleados incluyen perfiles y ángulos de acero unidos por tornillos, pernos o remaches o por medio de soldadura. (Méndez, 2015)

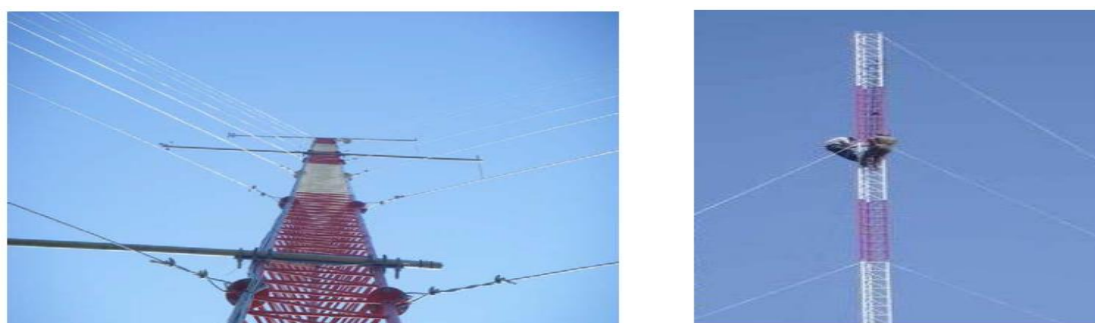
#### 1.2.1.1 Torres arriostradas o atirantadas

Las torres atirantadas, son estructuras, por lo general, formadas por una armadura espacial de sección triangular constante, y un conjunto de cables tesados en los cuales se apoyan lateralmente. Los cables son inclinados y se unen por uno de sus extremos a los vértices de la sección transversal a diferentes niveles de altura y por el otro extremo se anclan en el terreno en tres direcciones radiales espaciadas a 120 grados entre sí. (Parnás et al., 2012)

Estas torres por su economía de acero con relación a las autosoportadas cuando se requieren grandes alturas son muy usadas; sin embargo, la presencia de los cables le confiere un comportamiento estructural complejo ante las cargas ecológicas caracterizado

por la no linealidad debido a los grandes desplazamientos de la estructura atirantada y por la naturaleza fluctuante del viento que determina la respuesta dinámica del conjunto. (Parnás et al., 2012)

En este tipo de torres los tensores son generalmente de acero de alta resistencia. El empleo de tensores o tirantes permite alcanzar alturas importantes a un costo bajo de material. Estas torres se estructuran con tirantes a diferentes alturas (por lo general de 2/3 de la altura de la torre. La sección transversal más común de las torres de comunicaciones atirantadas es de celosía triangular. Los cables o arriostres se tensan para proporcionar la estabilidad adecuada a la estructura de celosía. (Méndez, 2015)



*Fig. 1.2 Torres arriostradas o atirantadas. Fuente: (Méndez, 2015)*

Este tipo de torres son ligeras y requieren mayor espacio libre para el anclaje de los cables, los cuales son anclados al terreno mediante bloques. Muchas veces se requieren instalar antenas celulares en puntos específicos o regiones, por lo que se recurre a construir torres arriostradas sobre edificaciones existentes. Estas torres cuentan generalmente de tirantes o arriostres a diferentes distancias. El peso que genera la torre sobre la estructura existente no es muy grande, por lo que no le adiciona mucho peso a la edificación; sin embargo, se debe de colocar el apoyo de las torres y sus arriostres sobre columnas y elementos resistentes. La descarga de la torre no debe hacerse directamente sobre una losa o algún otro elemento inadecuado, ya que éste podría fallar. La base de la torre transmitirá un esfuerzo de compresión en donde está apoyada, y los arriostres generalmente transmitirán esfuerzos de tensión. (Méndez, 2015)

#### **1.2.1.2 Torres Monopolo**

Este tipo de torre se emplea para antenas de radio, su altura generalmente se limita a unos 200 pies (61 m), haciéndolas inadecuadas para cumplir con las necesidades de algunos proveedores de servicios. Entre las principales ventajas de la torre de comunicación monopolo (Figura 1.3) son su peso ligero, poco requisito de espacio y bajo costo de instalación. Otra de las ventajas, es que este tipo de torres se pueden camuflajear como

árboles, lo que resulta más estético. Se deberá de construir una cimentación adecuada para resistir los efectos de la carga sobre la torre monopolo. (Méndez, 2015)



Fig. 1.3 Torres Monopolo. Fuente: (Méndez, 2015)

### 1.2.1.3 Torres autosoportadas

Las torres autosoportadas, objeto de estudio en esta investigación, son estructuras que se soportan por sí solas y pueden estar apoyadas en la tierra o sobre edificaciones, ver Figura 1.4. Estas torres se comportan como vigas en voladizo frente al viento y las cargas sísmicas. Su peso propio actúa de forma favorable frente a la acción de las cargas de viento. Las torres construidas con elementos lineales de acero en forma de celosía son las más eficientes. El uso de las celosías evita la exposición de una superficie llena y plana a la acción del viento permitiendo mayor ligereza con rigidez suficiente. (García, 2011)



Fig. 1.4 Torres autosoportadas. Fuente: (García, 2011)

a) Apoyada sobre tierra

b) Apoyada sobre edificio

Las torres de celosía más comunes son las de sección cuadrada y las de sección triangular, ver Figura 1.5. Las primeras llevan más consumo de materiales que las triangulares, pero presentan mayor rigidez a la torsión, debido a que en las cuadradas los momentos de inercia son iguales en todas las direcciones. (García, 2011)



*Fig. 1.5 Torres autosoportadas. Fuente: (García, 2011)*

a) Torre de sección triangular

b) Torre de sección cuadrada

Todas las torres autosoportadas en Cuba son de celosía y sus alturas se encuentran entre los 15m y los 125m. Las torres de más altura sobre el nivel del mar se encuentran en la provincia de Santiago de Cuba y Guantánamo con 1140 y 780 metros sobre el nivel medio del mar. (García, 2011)

Aunque las torres se comportan como elementos en voladizo al igual que la mayoría de los edificios, estudios realizados por varios autores coinciden en que las torres autosoportadas no presentan el mismo comportamiento dinámico que estos, debido a que en la mayoría de los edificios solo resulta necesario analizar el modo fundamental de vibración, mientras que, en las estructuras tipo torres, hay que analizar entre los 3 y los 5 primeros modos. La cantidad de modos que se deben analizar varía según los diferentes autores. (García, 2011)

Las principales vulnerabilidades de este tipo de torres son la carga de viento, especialmente en el caso de ciclones, los sismos y la colocación de un mayor número de antenas, sustitución de las existentes por otras más grandes y pesadas o el cambio de la disposición de las mismas lo que provoca la modificación de las fuerzas interiores de los elementos de la torre, ya que por una parte las antenas actúan como pantalla frente al viento haciendo que se generen fuerzas adicionales sobre la estructura y por otra la concentración de la masa en diferentes partes de la estructura, modifica su respuesta dinámica. (Parnás et al., 2012)

### **1.2.2 Elementos constructivos**

La mayoría de las torres se agrupan en dos tipos con relación a sus elementos. Estas pueden estar constituidas por:

- elementos angulares.

- mixtas (usan elementos angulares y elementos en forma de tubo).

La unión entre los elementos componentes de la torre se realiza por medio de pernos y planchuelas. En ocasiones se utilizan elementos tubulares tanto para las patas como para los elementos diagonales pues estas tienen un comportamiento frente al viento muy favorable. Esto permite a su vez menor consumo de material y por ende mayor economía. (Yáñez, 2010)

#### **1.2.2.1 Las antenas**

En el análisis de las fallas de torres de telecomunicaciones ocupa un lugar importante el estudio de las cargas generadas por las antenas, las que en general, se caracterizan por tener un reducido peso propio y generar una elevada carga lateral por la acción del viento sobre ellas en los puntos donde se localizan; que repercute en la respuesta estructural del sistema. Además, su presencia como masa concentrada, tiene la capacidad de modificar los períodos de oscilación. (Llanusa, 2013)

La presencia de antenas sobre el fuste conlleva a consideraciones especiales sobre sus anclajes o elementos de amarre al fuste, que incluyen la posibilidad de que las mismas se desprendan de la estructura a partir de determinados valores de la velocidad de viento para que no transmitan sus cargas a la torre. Cuando existen antenas en las torres hay que tener en cuenta que la carga de viento sobre la estructura dependerá no sólo de las condiciones climatológicas y geográficas donde se encuentra la obra, sino también del arreglo espacial de los elementos que conforman el fuste que determina la mayor o menor permeabilidad de la estructura; sino también de los diferentes tipos de antenas: UHF, VHF y parábolas, ver figura 1.6, que tienen factores de forma y de apantallamiento muy diferentes. (Llanusa, 2013)



*Fig. 1.6 Tipos de antena de izquierda a derecha UHF, VHF y Parábola. Fuente: (Llanusa, 2013)*

### **1.3 Calibración de modelos estructurales de elementos finitos**

En prácticamente todas las áreas de ciencia y la ingeniería, los modelos matemáticos o numéricos se usan para simular el comportamiento de sistemas reales. Los propósitos de estos modelos numéricos varían ampliamente, pero generalmente pueden subdividirse en tres categorías principales: Análisis, predicción, y diseño. En la ingeniería civil y estructural, los modelos de elementos finitos (FEM) están más a menudo usados para analizar, por ejemplo, los desplazamientos y fuerzas internas de estructuras en varios estados límites, o predecir respuestas de vibración debido a la carga dinámica como los terremotos, viento y tráfico. Estos análisis de modelos FE también pueden usarse para diseñar componentes estructurales o estructuras completas. Es importante decir que, para todos los propósitos anteriormente citados, la validez de los modelos numéricos adoptados es imperativa. Hay, sin embargo, siempre numerosas propiedades del sistema que son desconocidas o inciertas (por ejemplo, referente a propiedades materiales, las propiedades geométricas, las condiciones de apoyo, condiciones de cargas) para las cuales inevitablemente hay que realizar conjeturas o suposiciones. Además, debido a una falta de conocimiento u otras restricciones, a menudo suposiciones que simplifican el modelado de la estructura es requerido o implícitamente se hacen. Estos asuntos pueden desviarse de la calidad y la exactitud del modelo numérico y sus propósitos. (Simoen et al., 2013)

Esto ha conducido al desarrollo de técnicas de actualización de modelos, también llamadas calibración de modelos o, en más términos genéricos, de identificación o estimación de parámetros. Generalmente hablando, la calibración de modelos está dirigida a reconstruir o calibrar propiedades desconocidas del sistema que aparecen como parámetros en modelos numéricos, basado en el comportamiento real observado del sistema de interés. En el contexto estructural, a menudo los datos adquiridos en los experimentos de vibración (o sea las cronologías de aceleración, las funciones de respuesta en frecuencia, las frecuencias naturales y las formas modales de desplazamientos, las curvaturas o tensiones modales, flexibilidades modales, etc.) son más convenientes para los propósitos de la calibración, por proveer información detallada estimando el comportamiento global y local de la estructura de interés, y pueden estar medidos en un estado operacional de la estructura. (Simoen et al., 2013)

Muchos trabajos sobre diferentes tipos de calibración de modelos han sido llevados a cabo a través de los años. Las técnicas de calibración de modelos pueden ser clasificadas en no iterativas o también llamadas métodos directos e iterativas o también llamadas métodos de sensibilidad. (Abdullah et al., 2016)

Los métodos directos muestran resultados precisos. Por lo tanto, lo asumido sobre el modelo concuerda con los datos modales experimentales. La manera en que los datos modales medidos son reproducidos es más barata computacionalmente. No obstante, el resultado del modelo FE calibrado pueden tener pobre sentido físico porque el método viola la conectividad estructural y las matrices estructurales optimizadas son difíciles de interpretar. Esta es la razón por la que el método no ha sido generalmente usado en la práctica. (Abdullah et al., 2016)

También, el término "directo" fue utilizado para describir la identificación de un sistema sin la necesidad de calibrar un modelo de referencia. (Mottershead and Friswell, 1992)

Los métodos iterativos o sensitivos, los cuales tratan de reducir una función objetivo, la que es generalmente una función no lineal de parámetros seleccionados a calibrar, son llevados a cabo usando datos de funciones de respuesta en impulsos (IRFs) o datos en funciones de respuestas en frecuencias (FRFs). Por lo tanto, estos métodos proporcionan mayor elección de los parámetros a ser optimizados, superan las limitaciones de los métodos directos y también han sido aplicados exitosamente a problemas industriales a grandes escalas. (Abdullah et al., 2016)

Los métodos sensitivos están basados en la linearización de lo mayoría de las relaciones no lineales entre las salidas medibles tales como frecuencias naturales, formas modales o respuestas en desplazamientos de los parámetros del modelo. La cualidad más importante es definir una función de error u objetivo de los datos modales obtenidos a partir de la simulación por computadora y experimentalmente. Los parámetros estimados son obtenidos minimizando esta función con respecto a los parámetros experimentales. (Abdullah et al., 2016)

También los métodos de calibración de modelos pueden ser perfeccionados con el uso de funciones de daño, asegurando que la solución sea estable y físicamente entendible. (Teughels et al., 2002)

En conclusión, los problemas de calibración de modelos son problemas inversos, el objetivo es obtener los parámetros que producen una cierta respuesta dada. Más específicamente, la calibración de modelos persigue encontrar los parámetros óptimos de un modelo numérico que mejor igualen las respuestas del modelo con los datos observados. Esto es usualmente logrado formulando el problema como un problema constreñido de optimización, donde el objetivo es minimizar la discrepancia entre datos computados y medidos. (Simoen et al., 2013)



### **1.3.1 Obtención de los datos experimentales**

El primer paso dentro del proceso de calibración es realizar algún tipo de experimento sobre el sistema bajo estudio para obtener los datos de entrada-salida que servirán de base para la obtención del modelo final. Para la obtención de datos de entrada – salida se debe excitar el sistema mediante la aplicación de una señal de entrada y registrar la evolución de sus entradas y salidas durante un intervalo de tiempo. (Guillén, 2002)

Los datos experimentales consisten en mediciones obtenidas durante los experimentos de vibraciones, donde la estructura es equipada con a un conjunto de sensores (acelerómetros, galgas, sensores de fibras ópticas, transductores magnéticos, etc.) los cuales registran la respuesta del sistema debido a alguna excitación dinámica. Sin embargo, los datos recogidos en el dominio del tiempo, tales como las aceleraciones, no son usadas directamente para la calibración, debido a que esto requiere el conocimiento preciso de la excitación, lo cual es una condición que raramente se cumple en las aplicaciones de la ingeniería civil. Por lo tanto, en muchas más ocasiones, se utilizan las características modales, las cuales son extraídas de las mediciones de funciones de respuesta en el tiempo usando técnicas de análisis modal. Los datos modales son, en general, más ricos en contenidos de información y pueden ser obtenidos cuando la estructura se encuentra en estado de operación, lo que implica que ninguna excitación artificial se necesita aplica o conocer. (Simoen et al., 2013)

Los datos registrados pueden tener deficiencias que implican efectos devastadores en el resto del proceso de calibración, como son las siguientes (Guillén, 2002):

- Presencia de perturbaciones de alta frecuencia, por encima de las frecuencias de interés en la respuesta del sistema.
- Datos claramente erróneos, producidos por fallos en el hardware o software utilizados en el experimento de recogida de muestras.
- Desviaciones, desplazamientos o perturbaciones de baja frecuencia.

Para que el proceso de calibración sea satisfactorio, es necesario que los datos utilizados para tal fin contengan información significativa sobre el sistema. Esto implica un cuidadoso diseño del experimento de adquisición de datos, debiéndose tomar una serie de decisiones respecto a las señales que deben ser medidas, el período de muestreo a utilizar, el tipo de entrada más adecuada, el número de datos a almacenar, etc. (Guillén, 2002)

El análisis modal es un procedimiento que es usado para estudiar las propiedades dinámicas de las estructuras utilizando datos a partir de vibraciones el cual esencialmente consiste en medir y analizar la respuesta dinámica de los sistemas como estructuras o



fluidos a partir de la excitación de una fuerza. El análisis de los datos de vibraciones usualmente utiliza métodos como los análisis de Fourier y Wavelet. De estos análisis se extraen, por ejemplo, las frecuencias naturales y las formas modales. (Marwala, 2010)

### 1.3.2 Instrumentos para Medición de Vibraciones

Estos instrumentos son dispositivos que recibe la magnitud que se va a medir y la convierte en otra magnitud diferente, por ejemplo, caudal en presión o deformación en resistencia eléctrica. (Claro Duménigo, 2015). En la tabla 1.1 quedan resumidos algunos de estos y su definición.

Transductores más utilizados en la medición de vibraciones.

Acelerómetros	Transductor destinado a medir las aceleraciones. Pueden medir las aceleraciones en un único eje o en varios ejes a la vez. (Losa Miranda, 2015).
Galgas extensométricas	A partir de las deformaciones medidas se infieren los esfuerzos dinámicos. Si la selección del punto es adecuada, las mediciones pueden ser empleadas para el análisis modal o para la evaluación del comportamiento vibratorio de las estructuras. (Chan and O'Conner, 1990).
Galgas por capacitancia	Son de nuevo desarrollo y se emplean principalmente como una alternativa a los SG convencionales para ser usados en altas temperaturas, más allá incluso del límite del metal de los SG convencionales. (Gonzalez Martinez, 2007).
Transductores magnéticos	Se utilizan en la medición de desplazamientos. Las señales proporcionadas por los transductores se recogen a razón de una muestra por segundo durante la realización de las pruebas estáticas, y 70 muestras por segundo en las pruebas dinámicas.
Georadar IBIS-FS	El Georadar IBIS-FS presenta una solución totalmente novedosa en el campo de aplicación del monitoreo de vibraciones y oscilaciones en las estructuras, con muchas ventajas sobre los demás instrumentos tradicionales de monitoreo estático y dinámico. (IDS Ingegneria Dei Sistemi, 2013)

Tabla 1.1 Instrumentos de medición de vibraciones.

### 1.3.3 Métodos experimentales para la medición de vibraciones

El análisis dinámico de estructuras se refiere al análisis de las oscilaciones o vibraciones que puede sufrir una estructura alrededor de su posición de equilibrio. El análisis dinámico es importante porque ese movimiento oscilatorio produce una modificación de las tensiones y deformaciones en el tiempo, incluyendo la posibilidad de ocurrencia de resonancia.

Los análisis modales tratan de extraer las propiedades modales de una estructura (formas modales, frecuencias a las que se dan y factores de amortiguamiento) mediante la toma de medidas en la respuesta dinámica de la estructura a acciones externas. Estas acciones pueden ser inducidas, conociendo perfectamente las características de la excitación, o aleatorias, sin saber exactamente la magnitud de las acciones (OMA) (Losa Miranda, 2015).

Análisis Modal Experimental (EMA.)	Busca la extracción de las propiedades modales de una estructura mediante las medidas de la respuesta dinámica (output) a una fuerza de excitación conocida (input). (Losa Miranda, 2015).
<p>Algunas limitaciones (Zhang et al., 2010):</p> <p>Las excitaciones artificiales son normalmente llevadas a cabo para medir funciones de respuestas en frecuencias (FRFs) y funciones de respuestas en impulsos (IRFs). FRF o IRF serían muy difícil o incluso imposibles de ser medidas en pruebas en el campo o para estructuras muy grandes.</p> <p>El EMA es normalmente llevado a cabo en laboratorios. Sin embargo, en muchas aplicaciones industriales, las condiciones reales de aplicación pudieran diferir significativamente de las del laboratorio.</p> <p>En muchas ocasiones en el laboratorio, las excitaciones se les realizan a componentes, en lugar de los sistemas completos, por lo que los resultados deberían ser razonablemente considerados.</p>	
Análisis Modal Operacional (OMA)	Se plantea como una alternativa al EMA, llamando gran atención en la comunidad de ingenieros civiles a partir de los años 1990. Este trata de dar solución a las dificultades que plantea, en numerosas ocasiones, inducir una excitación artificial a estructuras de grandes dimensiones. Las excitaciones se producen de manera arbitraria por el uso habitual de la estructura (viento, tráfico).
<p>Ventajas (Zhang et al., 2010) :</p> <p>Es barato y rápido de llevar a cabo, no se necesita equipamiento para realizar las excitaciones ni simulación de condiciones.</p> <p>Se obtienen en muchos más puntos representativos las características dinámicas de los sistemas completos, en lugar de a componentes.</p> <p>Las formas modales pueden ser más fácilmente manejables y por lo tanto más conveniente para estructuras más complejas.</p> <p>El OMA puede ser utilizado no solo para el diseño dinámico, el control estructural, sino también para el monitoreo y detección de daños en las estructuras.</p>	

Tabla 1.2 Métodos experimentales para la medición de vibraciones.

#### **1.3.4 Procesamiento de las señales obtenidas en ensayos modales**

El procesamiento digital de las señales es una herramienta clave a la hora de realizar un análisis modal de un sistema concreto. El conjunto de las señales pueden clasificarse, en general, en estacionarias y no estacionarias (Losa Miranda, 2015).

Las señales estacionarias son aquéllas cuyas propiedades promedio no varían a lo largo del tiempo, pudiendo ser, a su vez, determinísticas o aleatorias. La naturaleza de las primeras permite representarlas matemáticamente y de manera explícita, de tal manera que, sus posibles valores futuros, puedan ser predichos. A este grupo pertenecen, por ejemplo, las señales periódicas. Sin embargo, el valor de las aleatorias depende del instante en el que se encuentren, resultando casi imposible prever con exactitud sus valores futuros. Las señales no estacionarias pueden dividirse en continuas o transitorias. Estas últimas pueden definirse como señales que comienzan en cero en el período de observación (Losa Miranda, 2015).

Puesto que la finalidad del análisis de las señales es tratar de obtener la máxima información de ellas, resulta óptimo contar con la capacidad de estudiarlas en sus distintos dominios. Para extraer su contenido en frecuencias, resulta más sencillo analizar las señales en el dominio de las frecuencias. Se debe tener en cuenta que los registros se realizan en el dominio del tiempo, y por tanto, se debe acudir a la transformada (e inversa) de Fourier, ya que se trata de una herramienta que permite transformar de manera sencilla una señal en el tiempo a una señal en frecuencia y viceversa (Losa Miranda, 2015).

#### **1.3.5 Extracción de parámetros modales**

La técnica más usada a nivel mundial para extraer las propiedades modales es el proceso llamado Análisis Modal. (Marwala, 2010)

Los dominios utilizados en la calibración de modelos son el dominio modal de los datos (MDD), que está expresado en las frecuencias naturales, las formas modales y los radios de amortiguamiento, y el dominio de las frecuencias de los datos donde la razón de respuesta a la excitación en este dominio es llamada funciones de respuestas en frecuencias (FRF). (Marwala, 2010)

Los datos son medidos en el dominio del tiempo y se utilizan técnicas de las transformadas de Fourier para transformar los datos al dominio de las frecuencias. Las propiedades modales son identificadas del dominio de las frecuencias (en ocasiones del dominio del tiempo). Debido a que los datos en el dominio del tiempo son complicados de entender, no

son usados ampliamente para esta identificación. Por esta razón es más conveniente el estudio de los dominios modales y de frecuencias. (Marwala, 2010)

En este punto se analizan diversas etapas de análisis que deben llevarse a cabo para conseguir el objetivo de construir un modelo. Una mayor parte de este análisis consiste en ajustar una curva teórica para cada función de respuesta en frecuencia medida. Diversos procedimientos disponibles para conseguir dicha curva, prácticamente todos se basan en el mismo fin: encontrar los coeficientes en la expresión teórica para obtener una curva lo más parecida posible a la medida experimentalmente (Anónimo, 2015).

### **1.3.6 Modelos de elementos finitos de las estructuras (FEM)**

Cuando se hace necesario conocer el comportamiento de un sistema en unas determinadas condiciones y ante unas determinadas entradas, se puede recurrir a la experimentación sobre dicho sistema y a la observación de sus salidas. Sin embargo, en muchos casos la experimentación puede resultar compleja o incluso imposible de llevar a cabo, lo que hace necesario trabajar con algún tipo de representación que se aproxime a la realidad, y a la que se conoce como *modelo*. Básicamente, un modelo es una herramienta que permite predecir el comportamiento de un sistema sin necesidad de experimentar sobre él. (Guillén, 2002)

El modelado por elementos finitos para la calibración de modelos se puede entender como el proceso numérico y matemático, a través del cual, una estructura física es traducida en un modelo matemático y a partir de ese modelo matemático un procedimiento numérico es utilizado para estimar características dinámicas como las formas modales y las frecuencias naturales. La calibración de modelos FE es un proceso a través del cual estos modelos son actualizados para reflejar mucho mejor los datos medidos. (Marwala, 2010)

Contar con el modelo confiable de una estructura es importante para realizar un adecuado análisis o diseño estructural. Sin embargo, una estructura real está sujeta a muchas variables que influyen en su comportamiento. Las variables que pueden influir en la respuesta de una estructura pueden ser el tipo de cargas, el comportamiento del material, las condiciones de frontera y las conexiones entre los diferentes elementos que la conforman. El rango de variación de estas variables influye de manera significativa en la respuesta de la estructura por lo que, encontrar los valores óptimos significa que la respuesta estructural será también confiable. (Sovero, 2014)

Existen dos métodos principales para la obtención del modelo: (Guillén, 2002):

Modelado teórico	Se trata de un método analítico, en el que se recurre a leyes básicas de la física para describir el comportamiento dinámico de un fenómeno o proceso.
Identificación del sistema	Se trata de un método experimental que permite obtener el modelo de un sistema a partir de datos reales recogidos de la planta bajo estudio.

*Tabla 1.3 Métodos para la obtención del modelo.*

El modelado teórico tiene un campo de aplicación restringido a procesos muy sencillos de modelar, o a aplicaciones en que no se requiera gran exactitud en el modelo obtenido. En muchos casos, además, la estructura del modelo obtenido a partir del conocimiento físico de la planta posee un conjunto de parámetros desconocidos y que sólo se pueden determinar experimentando sobre el sistema real. De ahí la necesidad de recurrir a los métodos de identificación de sistemas (Guillén, 2002).

En la práctica, lo ideal es recurrir a una mezcla de ambos métodos de modelado para obtener el modelo final. El uso de datos reales para identificar los parámetros del modelo provee a éste de una gran exactitud, pero el proceso de identificación se ve tanto más facilitado cuanto mayor sea el conocimiento sobre las leyes físicas que rigen el proceso (Guillén, 2002).

Para obtener el Modelo mecánico o Modelo analítico de la estructura real, se recomienda hacer un análisis integrado de las siguientes invariantes. (Figueroa Castellón and Pino Rodríguez)

1. Modelación de la forma o modelo geométrico.
2. Modelo del comportamiento de los materiales.
3. Modelo de las acciones impuestas.
4. Modelo de los apoyos y enlaces.
5. Masa.

Un aspecto fundamental tratado en (Bonilla Rocha, 2008) son los métodos de solución empleados en la modelación, pues constituyen uno de los aspectos más controversiales a la hora de enfrentar un problema, estos pueden ser analíticos o numéricos. Pero debido al creciente desarrollo informático se han creado importantes herramientas computacionales que se encuentran sobre la base de estos métodos numéricos y aplicables principalmente al análisis de problemas de tensión-deformación de sólidos. (Aspiro, 2016)

Una vez que se ha identificado el modelo y se ha programado, se necesita aplicarlo al problema concreto objeto de estudio. Para ello se necesita en primer lugar obtener los

valores de los parámetros que utiliza el modelo estos pueden medirse directamente en el campo (sería la manera de proceder en un modelo físico) u obtenerse utilizando las técnicas de optimización que a partir de unos valores conocidos de las variables de entrada produzcan los correspondientes resultados en las variables de estado y de salida (sería la manera de proceder en los modelos empíricos) (Anónimo, 2010).

Para realizar la modelación, análisis y diseño de torres en la actualidad se cuenta con numerosos softwares muy potentes, dentro de estos están:

- SAP2000
- ABAQUS
- ANSYS
- STAAD.Pro
- MYDAS

La modelación en estos programas se realiza empleando el MEF (Método de Elementos Finitos) y se destacan principalmente dos tipos de modelado: (Marwala, 2010)

- Modelado en 2 dimensiones (2D).
- Modelado en 3 dimensiones (3D).

Aunque el modelado 2D es simple y permite análisis computacionalmente eficientes, este proporciona reducida exactitud. Resultados más exactos pueden ser obtenidos a través del modelado 3D, sin embargo, este requiere mayores cálculos computacionales. Más aún, el análisis por MEF puede ser formulado tal que el sistema puede ser lineal o no lineal. Modelar un sistema lineal no es complejo y usualmente no considera deformaciones plásticas, mientras que sistemas no lineales sí toman en cuenta estas deformaciones. (Marwala, 2010)

También, el MEF es un método de solución de problemas ingenieriles por aproximación, en donde la región de integración que generalmente es continua, se discretiza, de forma que el elemento continuo se divide en un número finito de partes, cuyo comportamiento se especifica mediante un número finito de parámetros asociados a ciertos puntos característicos denominados nodos. La solución del sistema completo, en el que se involucran las ecuaciones de equilibrio, de compatibilidad de desplazamientos, las restricciones de los apoyos y constricciones, sigue entonces las reglas de los problemas discretos. El sistema completo se forma por ensamblaje de los elementos. Este método, por tanto, se basa en transformar un cuerpo de naturaleza continua en un modelo discreto aproximado, esta transformación se denomina discretización. El conocimiento de lo que sucede en el interior de este modelo de cuerpo aproximado, se obtiene mediante la

interpolación de los valores conocidos en los nodos. Es por tanto una aproximación de los valores de una función a partir del conocimiento de un número determinado y finito de puntos (Ramírez Díaz et al., 2015).

### 1.3.7 Validación

La validación es el último paso del proceso de análisis modal experimental, tal y como se describió en la introducción de este punto. La finalidad de esta fase es verificar los resultados del paso anterior (estimación de los parámetros modales). Se concreta en una serie de herramientas para evaluar la calidad del modelo modal estimado.

Este proceso de validación se puede estructurar en tres fases (Anónimo, 2015)

- 1) La primera de ellas es intuitiva, sin requisito de herramientas matemáticas. Se trata de una inspección visual de los modos de vibración y comparando las frecuencias naturales con los estimados en el problema numérico.
- 2) En el segundo nivel se requiere de ciertas herramientas matemáticas para cuantificar la calidad del modelo estimado. Se verá en 1.3.7.1 el MAC (Model Assurance Criterion), otras son, participación modal, reciprocidad, complejidad del modo...
- 3) Ya en el tercer nivel de la validación, se realiza una validación implícita, donde el modelo se usa para análisis de sensibilidad, prediciendo el efecto de cambios en la estructura, actualizando de esta forma el modelo de elementos finitos que se haya generado.

#### 1.3.7.1 Model Assurance Criterion (MAC)

El desarrollo histórico del MAC se originó a partir de la necesidad de un indicador que asegurara la calidad de los vectores modales experimentales que son estimados a partir de medir funciones de respuestas en frecuencias.

El MAC compara las formas modales medidas y las calculadas. Este puede ser matemáticamente resumido por la siguiente ecuación. (Marwala, 2010)

$$MAC_{cdr} = \frac{|\{\phi_{cr}\}\{\phi_{dr}^*\}|^2}{\{\phi_{cr}\}^T \{\phi_{cr}^*\} \{\phi_{dr}\}^T \{\phi_{dr}^*\}} \quad \{1.1\}$$

En ecuación {1.1}, donde: **MAC** - Model Assurance Criterion, **c** es para referencias, **d** son los grados de libertad, **r** es el modo, **T** es la transpuesta, **\*** es la conjugada compleja, y **{ }** es un vector.

El MAC toma valores desde cero (representando una correspondencia no consistente) hasta 1 (representando una correspondencia consistente). Este solo puede indicar consistencia, no validez ni ortogonalidad. Si los mismos errores existen en todas las estimaciones de los vectores modales, esto no es delineado por el MAC. Las suposiciones invalidas son normalmente la causa de esta clase de error potencial. A pesar de que el MAC sea la unidad, las suposiciones relacionadas con el sistema o las técnicas de estimación de los parámetros no son necesariamente correctas. Las suposiciones pueden causar errores de consistencia en todos los vectores modales sometidos todas las pruebas certificadas por el MAC. (Allemang, 2003)

Las representaciones del MAC pueden ser a partir de gráficos (figura 1.7) o numéricamente en tablas (figura 1.8), donde los valores de la diagonal principal deben ser cercanos a la unidad para que se obtenga un resultado positivo. (Allemang, 2003)

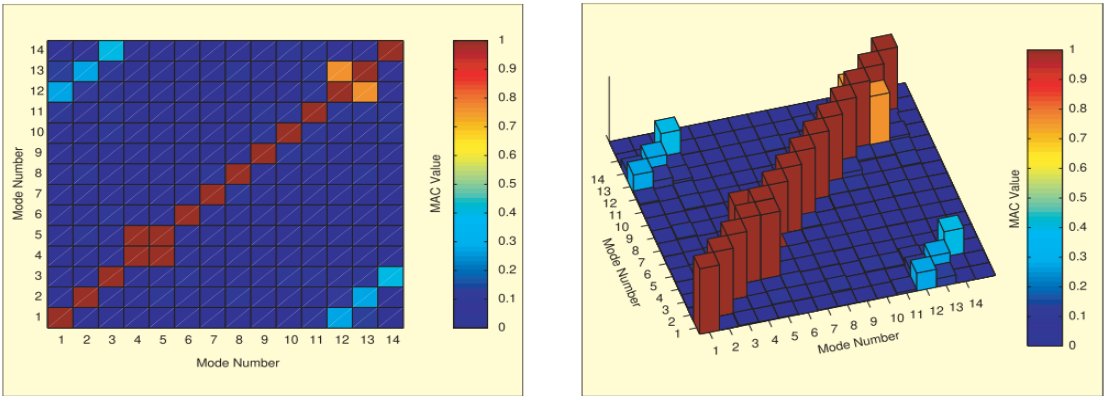


Fig. 1.7 Presentaciones 2D y 3D de los valores del MAC. Fuente: (Allemang, 2003)

Table 1. Numerical presentation of MAC values.

Mode	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	1.000	0.000	0.001	0.011	0.021	0.012	0.008	0.002	0.029	0.005	0.001	0.118	0.275	0.007
2	0.000	1.000	0.012	0.006	0.010	0.001	0.009	0.003	0.007	0.006	0.029	0.273	0.126	0.002
3	0.001	0.012	1.000	0.002	0.007	0.017	0.001	0.001	0.006	0.022	0.019	0.060	0.045	0.394
4	0.011	0.006	0.002	1.000	0.993	0.011	0.007	0.001	0.014	0.013	0.003	0.002	0.002	0.010
5	0.021	0.010	0.007	0.993	1.000	0.013	0.006	0.012	0.015	0.014	0.004	0.002	0.003	0.003
6	0.012	0.001	0.017	0.011	0.013	1.000	0.000	0.005	0.004	0.007	0.011	0.012	0.009	0.006
7	0.008	0.009	0.001	0.007	0.006	0.000	1.000	0.000	0.022	0.005	0.002	0.008	0.009	0.007
8	0.002	0.003	0.001	0.001	0.012	0.005	0.000	1.000	0.000	0.020	0.019	0.011	0.007	0.006
9	0.029	0.007	0.006	0.014	0.015	0.004	0.022	0.000	1.000	0.015	0.008	0.005	0.019	0.006
10	0.005	0.006	0.022	0.013	0.014	0.007	0.005	0.020	0.015	1.000	0.093	0.002	0.003	0.025
11	0.001	0.029	0.019	0.003	0.004	0.011	0.002	0.019	0.008	0.093	1.000	0.026	0.017	0.023
12	0.118	0.273	0.060	0.002	0.002	0.012	0.008	0.011	0.005	0.002	0.026	1.000	0.755	0.067
13	0.275	0.126	0.045	0.002	0.003	0.009	0.009	0.007	0.019	0.003	0.017	0.755	1.000	0.067
14	0.007	0.002	0.394	0.010	0.003	0.006	0.007	0.006	0.006	0.025	0.023	0.067	0.067	1.000

Fig. 1.8 Presentación numérica de los valores del MAC. Fuente: (Allemang, 2003)

Si la diagonal principal de la matriz MAC tiene valores cercanos a cero, esto es una indicación de que los vectores modales no son coherentes. Esto puede ser debido a alguna de las siguientes razones (Allemang, 2003):



- El sistema es no estacionario. Esto puede ocurrir si el sistema es no lineal y dos grupos de datos han sido adquiridos a diferentes tiempos o niveles de excitación. Las no linealidades del sistema aparecerán de forma diferente en las FRFs generadas desde una posición del excitador o señales de excitación diferentes. Los algoritmos de estimación de los parámetros modales no manejarán las diferentes características no lineales en una manera coherente.
- Hay ruido en el vector modal de referencia. Este caso es el mismo que tener ruido en la entrada de la medición de FRFs. Ninguna cantidad de procesamiento de señal puede eliminar este tipo de error.
- La estimación de los parámetros modales es invalida. Las mediciones de la FRF pueden no contener errores, pero la estimación de los parámetros modales podría no ser coherentes con los datos.
- Los vectores modales son a partir de vectores de forma modales linealmente no relacionadas. Partiendo de que los estimados del vector modal son de diferentes posiciones de excitación, esta medida de coherencia implicará que los vectores modales son ortogonales.

El MAC no es el más adecuado cuando se comparan modos que están cercanos en frecuencia o que son medidos con insuficientes posiciones del transductor. (Marwala, 2010)

Otros criterios de certidumbre similares han sido desarrollados derivados del MAC como son (Allemang, 2003):

- Weighted Modal Analysis Criterion (WMAC).
- Partial Modal Analysis Criterion (PMAC).
- Modal Assurance Criterion Square Root (MACSR).
- Scaled Modal Assurance Criterion (SMAC).
- Modal Assurance Criterion Using Reciprocal Vectors (MACRV)
- Modal Assurance Criterion with Frequency Scales (FMAC).
- Coordinate Modal Assurance Criterion (COMAC).
- The Enhanced Coordinate Modal Assurance Criterion (ECOMAC).
- Mutual Correspondence Criterion (MCC).
- Modal Correlation Coefficient (MCC).
- Inverse Modal Assurance Criterion (IMAC).
- Frequency Response Functions Assurance Criterion (FRFAC).
- Complex Correlation Coefficient (CCF).
- Frequency Domain Assurance Criterion (FDAC).
- Coordinate Orthogonality Check (CORTHOG).

La mayoría de los usos potenciales del MAC son bien conocidos pero unos pocos son más prudentes. La siguiente lista presenta los usos más típicos que han sido reportados en la literatura (Allemang, 2003):

- Validación de modelos experimentales.
- Correlación con modelos modales analíticos. (mode paring)
- Matriz de mapeo entre modelos modales analíticos y experimentales.
- Análisis de errores en vectores modales.
- Evaluaciones de la calidad del control.
- Posicionamiento óptimo de los sensores.

### **1.3.8 Optimización en el proceso de calibración de modelos estructurales**

Existen infinitudes de combinaciones de cambios que se le pueden hacer al modelo FE con el objetivo de obtener una mejor correlación entre el modelo y la estructura real, lo que es una tarea que no puede ser conseguida sin el uso de técnicas de optimización. El campo de la optimización estructural provee una manera rigurosa con la cual los parámetros son seleccionados y manipulados logrando que el modelo FE resultante es más exacto y entendible físicamente. (Morris, 2009)

Existen una gran cantidad de métodos de optimización que pueden ser utilizados para el proceso de calibración de modelos. (Marwala, 2010)

#### **1.3.8.1 Análisis de sensibilidad**

El comportamiento de una estructura depende de ciertos factores de su entorno que influyen en la respuesta óptima. El objetivo de realizar un análisis de sensibilidad es identificar cuáles de estos factores tienen una mayor influencia y modifican el comportamiento de la estructura. El análisis de sensibilidad para el diseño estructural se define como la relación que existe entre las variables disponibles y la respuesta de una estructura. (Sovero, 2014)

También, según (Morris, 2009), un análisis de sensibilidad es una prueba simple donde a través de la cual los parámetros son alterados uno a la vez y se miden los impactos que esto tiene en las frecuencias propias de oscilación. Los parámetros que tienen mayor impacto en las frecuencias del modelo deberían ser considerados en la calibración.

En función del tipo de variables, el análisis de sensibilidad puede ser de dos tipos (Sovero, 2014): a) sensibilidad de parámetros, donde las variables que más influyen en la respuesta esperada son las que corresponden al problema de equilibrio, como por ejemplo el módulo de elasticidad o el momento de inercia y, b) sensibilidad de formas, donde las variables

están relacionadas con la geometría de la estructura, la posición de las cargas o las condiciones de borde.

Si las variables no se definen correctamente el modelo numérico de la estructura no representará de manera apropiada su comportamiento estructural. Por lo tanto, para realizar adecuadamente el análisis de sensibilidad es necesario llevar a cabo un proceso parametrizado sobre la estructura, modificando paso a paso cada una de las variables y observando que pasa con la respuesta. Una vez identificados cuáles son los factores que tienen más influencia en la respuesta analizada éstos se pueden modificar a través de un proceso de optimización de tal modo que el modelo numérico represente el comportamiento real de la estructura estudiada. (Sovero, 2014)

### 1.3.8.2 Conceptos básicos sobre optimización

La optimización es un proceso matemático que consiste en encontrar una solución óptima a un problema planteado mediante la definición de una ecuación que se conoce como función objetivo. Los métodos de optimización dependen de las características de esta función y se dividen en dos grupos: a) optimización de funciones lineales, y b) optimización de funciones no lineales. Asimismo, la función objetivo depende de ciertas variables identificadas previamente en un análisis de sensibilidad y cuyo valor se desconoce. Estas variables están sujetas a restricciones que limitan la zona de búsqueda de la solución la cual está definida por un rango determinado de valores donde es posible encontrar el valor óptimo (Sovero, 2014). Consiste en obtener un resultado que satisfaga en diferentes aspectos el proceso de calibración como el tiempo de computo o la precisión del resultado después de realizada la misma, es decir, consiste en mejorar el funcionamiento y el proceso de la calibración a través de distintas metodologías. (Gutiérrez et al., 2013)

En la tabla 1.4 se presentan cuatro conceptos que son la base de la optimización.

Variables de diseño	Son variables independientes que son iteradas en orden para alcanzar el diseño óptimo. Se especifican límites superiores e inferiores como "restricciones" en las variables de diseño. Estos límites restringen el rango de variación de las Variables de Diseño (DV's, por sus siglas en inglés).(Rodríguez Calderón et al., 2007). Algunos ejemplos de estas variables pueden ser el módulo de elasticidad, la densidad o el módulo de corte de un material. Hay que tener en especial cuidado en los límites seleccionados debido a que estos son importantes en el momento de evaluar si un resultado es factible o no (Gutiérrez et al., 2013).
---------------------	--

Variables de estado	Variables de Estado (SVs por sus siglas en inglés) son cantidades que condicionan o restringen el diseño. Se conocen también como "variables dependientes", y son típicamente variables de respuesta que son funciones de las variables de diseño. Una variable de estado puede tener un límite máximo y un límite mínimo, o puede tener un "único límite".(Rodríguez Calderón et al., 2007). Algunos ejemplos de estas variables pueden ser las deflexiones en distintos puntos a lo largo de la geometría de un modelo o los modos de vibración y frecuencias naturales de una estructura (Gutiérrez et al., 2013).
Restricciones	<p>Para prevenir que el algoritmo no converja en modelo físicamente no entendible, los parámetros son constreñidos a valores reales. Las restricciones pueden ser añadidas en de dos formas: implícitamente o explícitamente. Las implícitas son las relaciones entre los parámetros y cómo impactan en el sistema de matrices y son manejadas internamente por un programa de análisis estructural. Las explícitas son aplicadas para restringir las posibles perturbaciones de los parámetros dentro de ciertos límites.</p> <p>Las restricciones en el comportamiento del diseño estructural pueden ser las limitaciones en los esfuerzos o en los desplazamientos, pero también pueden tomar la forma de restricciones en frecuencias de vibración o probabilidad al colapso (Gutiérrez et al., 2013).</p>
Función Objetivo.	<p>La función objetivo es la función que mide cuantitativamente el funcionamiento del sistema en un proceso de optimización esto quiere decir que se busca una maximización o minimización de esta, y es la base para seleccionar dentro de una gama de diseños aceptables. La función objetivo es una función vectorial de las variables de diseño. Representa la propiedad más importante de un diseño. Debido a la versatilidad de la función objetivo para adaptarse al problema propuesto, esta función puede ser continua, discreta o mezclada en aquellos casos en donde entre ciertos intervalos la función se define como discreta y en otros intervalos se define como continua (Cujia Meza, 2010).</p> <p>Es una función de las DVs, entonces, si cambian los valores de las DVs cambia el valor de la función objetivo(Gutiérrez et al., 2013).</p>

Tabla 1.4 Conceptos básicos sobre optimización.

### 1.3.8.3 Función objetivo

En este caso de calibración de modelos se emplea un algoritmo de optimización basado en el método de la región de confianza Gauss-Newton (Reynders et al., 2007) implementando en MATLAB la función: lsqnonlin (Reynders and Roeck, 2014).

El vector  $\theta$  contiene los parámetros de rigidez a actualizar. Suponiendo que el vector  $r(\theta)$  contiene las diferencias entre los parámetros del modelo numérico con las propiedades de la rigidez y los parámetros modales medidos. La forma común de resolver el problema de calibración es mediante la minimización del vector residual  $r(\theta)$  empleando mínimos cuadrados, lo cual corresponde a la minimización de la función objetivo (Reynders et al., 2007).

$$f(\theta) = \frac{1}{2} \|r(\theta)\|^2 \quad \{1.2\}$$

$$r(\theta) = \begin{bmatrix} w_f r_f(\theta) \\ w_s r_s(\theta) \end{bmatrix} \quad \{1.2.1\}$$

$\omega_f$ = factor de peso para residuos por frecuencias naturales

$\omega_s$ = factor de peso para residuos por formas modales

A su vez el vector residual consta de dos partes:

$$r_{fj}(\theta) = \frac{f_j(\theta)^2 - \tilde{f}_j^2}{\tilde{f}_j^2} \quad \text{residuos por frecuencias naturales} \quad \{1.2.1.1\}$$

$$r_{sj}(\theta) = \phi_j(\theta) \cdot MSF_j - \tilde{\phi}_j \quad \text{residuos por formas modales} \quad \{1.2.1.2\}$$

Los términos  $f_j$  y  $\tilde{f}_j$  denotan las frecuencias numéricas y experimentales del modo  $j$ , respectivamente.

Los términos  $\phi_j$  y  $\tilde{\phi}_j$  denotan las formas modales numéricas y experimentales del modo  $j$ , respectivamente.

$MSF_j$  es el factor de escala modal del modo  $j$ , definido en la ecuación {1.3}

$$MSF_j = \frac{\tilde{\phi}_j^T \phi_j}{\phi_j^T \phi_j} \quad \{1.3\}$$

#### 1.3.8.4 Técnicas de optimización en ingeniería

Las metodologías que pueden emplearse para resolver un problema de optimización pueden clasificarse en dos grandes grupos (Zaforteza, 2007):

##### Métodos Clásicos.

Un método de optimización clásico es aquel que usa las derivadas (gradientes). Se puede decir que estos métodos buscan y garantizan un óptimo local, pero sin un tiempo determinado. Los métodos con derivadas se basan en tres algoritmos fundamentales: pendiente máxima (steepest descent), el método de Newton (Newton's method) y el

método Levenberg-Marquardt. Sin embargo no son los únicos métodos y algoritmos, pero todos los demás resultan de mejoras y combinaciones de estos (Cujia Meza, 2010).

#### Métodos heurísticos y metaheurísticos:

Métodos aproximados que proporcionan una buena solución del problema, aunque no tiene por qué ser la óptima. Se trata de algoritmos sencillos e intuitivos, pero que requieren un gran esfuerzo computacional pues precisan que la función objetivo y las restricciones del problema sean evaluadas un gran número de veces. (Zaforteza, 2007)

Se fundamentan en el uso de conceptos intuitivos basados en sistemas naturales, como por ejemplo la evolución. El solo hecho de no usar derivadas que en ciertos casos pueden ser tediosas o aun casi imposibles de obtener, le da a estos métodos una flexibilidad, y una gran extensión de aplicabilidad cuando las condiciones del modelo son complejas. La manera de encontrar la solución de dichos métodos es a través de la iteración. Algunos ejemplos de estos métodos son:

Tomado de (Cujia Meza, 2010).

- Algoritmos Genéticos.
- Recocido Simulado (Simulated Annealing SA).
- Búsqueda Aleatoria (Random Search).
- Método Nelder–Mead o Método Ameba (Downhill Simplex Search).
- Optimización por Enjambres de Partículas (Particle Swarm Optimization “PSO”).
- Ascenso de montañas (Stochastic Hill-climbing).
- Algoritmos Miméticos.

#### **1.3.8.5 Problemas de mínimos cuadrados**

Los métodos para solucionar tales problemas son iterativos y cada paso de iteración usualmente requiere la solución de un problema mínimo lineal de cuadrados. El problema mínimo no lineal de cuadrados es un caso especial del problema general de optimización. El método básico de mínimos cuadrados es el Gauss-Newton, derivativo del método Newton para los problemas generales explicado en el epígrafe anterior. (Roeck, 2003)

En los problemas mínimos no lineales de cuadrados la función objetivo consta de una suma de cuadrados de funciones no lineales. El Gauss Newton y los métodos de Levenberg-Marquardt aproximan la Hessiana usando solo la matriz Jacobiana. Este acercamiento es justificado para problemas de pequeños residuales o no lineales, en los otros casos, Newton híbrido o métodos de Quasi-Newton deberían ser aplicados. (Roeck, 2003)

En nuestra investigación se utilizan los mínimos cuadrados no lineales el cual puede ser mirado como un caso especial de un problema de optimización y en el cual los métodos iterativos son aplicados para solucionarlo. En este caso se hace énfasis en el método Gauss-Newton, pero para problemas moderados o de grandes residuales o no lineales, la velocidad de convergencia del método del Gauss-Newton puede ser muy inferior al método de Newton. (Roeck, 2003)

## **1.4 Empleo de softwares computacionales para la calibración de modelos**

La disponibilidad de ordenadores de elevada potencia de cálculo y bajo coste, junto con el desarrollo de técnicas basadas en la inteligencia artificial, así como el desarrollo de potentes softwares ha permitido que en las últimas décadas haya crecido de forma importante el diseño de estructuras óptimas. Muchos de estos programas que contienen en su paquete algunas de las técnicas más avanzadas de optimización se han convertido para los ingenieros en herramientas imprescindibles de su ambiente de trabajo (González Pérez, 2015). A continuación, se referencian algunas de estas potentes herramientas:

### **1.4.1 MATLAB**

MATLAB es el nombre abreviado de “MATrix LABoratory”. MATLAB es un programa para realizar cálculos numéricos con vectores y matrices. Como caso particular puede también trabajar con números escalares –tanto reales como complejos–, con cadenas de caracteres y con otras estructuras de información más complejas. Una de las capacidades más atractivas es la de realizar una amplia variedad de gráficos en dos y tres dimensiones. MATLAB tiene también un lenguaje de programación propio. (García de Jalon et al., 2005)

MATLAB integra análisis numérico, cálculo matricial, proceso de señales y visualización gráfica en un entorno completo donde los problemas y sus soluciones son expresados del mismo modo en que se escribirían tradicionalmente, sin necesidad de hacer uso de la programación tradicional (Moore, 2007).

Como caso particular puede también trabajar con números escalares tanto reales como complejos, con cadenas de caracteres y con otras estructuras de información más complejas. Además dispone de un código básico y de varias librerías especializadas (toolboxes) (Rodríguez Pérez de Alejo, 2011).

Goza en la actualidad de un alto nivel de implantación en escuelas y centros universitarios, así como en departamentos de investigación y desarrollo de muchas compañías industriales. En el mundo industrial, MATLAB está siendo utilizado como herramienta de investigación para la resolución de complejos problemas planteados en la realización y

aplicación de modelos matemáticos en ingeniería. Los usos más característicos de la herramienta se encuentran en áreas de computación y cálculo numérico tradicional, prototipaje algorítmico, teoría de control automático, estadística, análisis de series temporales para el proceso digital de señales (Bacuilima G, 2011).

Toolboxes más importantes (Tomado de (Aspiro, 2016 #51))

- Curve fitting: Ajustes de modelos y análisis.
- Data Acquisition: Adquiere y envía datos a un instrumento electrónico conectado al computador. (sólo para Windows)
- Excel link: Permite usar MATLAB con datos leídos directamente desde planillas Excel.
- Image processing: Permite el procesamiento de imágenes, análisis y desarrollo de algoritmos.
- **MACEC**: Es un toolbox programado sobre MATLAB para el análisis modal operacional y experimental (Reynders and Roeck, 2014).
- Partial differential equation: Soluciona y analiza sistema de ecuaciones diferenciales parciales.
- **Signal Processing**: Permite el procesamiento de señales, análisis y desarrollo de algoritmos.
- Spline: Crea y manipula modelos de aproximación de datos Spline.
- **Stabil 2.0**: toolbox para el análisis por MEF de estructuras.
- Statistics: Permite aplicar modelos estadísticos y modelos de probabilidades.
- Structural Dynamics: Analiza modelos de elementos finitos y lleva a cabo análisis modales de sistemas mecánicos.
- Wavelet: Analiza, comprime y saca el ruido de señales e imágenes usando técnicas de wavelet.
- **Optimization**: Consta de un conjunto de funciones que resuelven problemas de extremos, con o sin condiciones, de funciones reales las cuales son generalmente multivariantes y no lineales.

#### 1.4.1.1 Toolbox de Optimización

El toolbox de optimización consta de un conjunto de funciones que resuelven problemas de extremos, con o sin condiciones, de funciones reales las cuales son generalmente multivariantes y no lineales. Asimismo, posee funciones para la resolución de algunos tipos de problemas matriciales en extremos. Resulta conveniente para una comprensión y mejor manejo de la toolbox poseer conocimientos básicos previos de análisis de funciones reales, matrices y teoría de extremos (Bacuilima G, 2011).



Entre las principales funciones de la caja de herramientas de optimización están (Aspiro, 2016):

- **fminbnd**: Esta función resuelve los problemas de optimización de funciones de una variable sin restricciones, se la conoce también como optimización escalar.
- **fminsearch**: Esa función en cambio resuelve los problemas de optimización de funciones de más de una variable sin restricciones.
- **fminunc**: Esta al igual que la anterior proporciona el mínimo de una función de variables sin restricciones, pero, con la diferencia que utiliza información del gradiente y el hessiano de la función objetivo.
- **fmincon**: Esta función determina el mínimo de una función multivariable con restricciones de igualdad y desigualdad, lineales y no lineales.
- **quadprog**: Realiza la minimización de una función cuadrática con restricciones de igualdad y desigualdad lineales.
- **linprog**: Esta función realiza la optimización de problemas de programación lineal.
- **lsqnonlin**: Resuelve por mínimos cuadrados problemas no lineales de funciones o de ajuste de datos.

Además, está **ga-Genetic Algorithm** que resuelve por Algoritmos Genéticos diferentes tipos de problemas con o sin restricciones.

Al analizar cada una de estas funciones y su aplicación, la función que más se ajusta para resolver el problema de Model Updating es la *lsqnonlin*, (Reynders and Roeck, 2014) pues a través de la misma se realiza la optimización de problemas no lineales por mínimos cuadrados mediante el ajuste de datos o valores obtenidos del software SAP2000 mediante la interface OAPI que se realiza con MATLAB.

MATLAB también permite la creación de interfaces con otras plataformas mediante las cuales se puede llevar a cabo una resolución conjunta del problema de calibración de un modelo estructural mediante técnicas de optimización (González Pérez, 2015).

En esta investigación precisamente se emplea la interface OAPI SAP2000-MATLAB, anteriormente puesta en práctica en investigaciones como(Bacuilima G, 2011), (Sextos and Balafas, 2011), (Aspiro, 2016) aunque también se han realizado recientemente investigaciones como(Gutiérrez et al., 2013), (Gauchía et al., 2014), en las que se emplea una interface ANSYS-MATLAB.

### **1.4.2 SAP2000**

SAP2000 es un programa estructural basado en el Método de Elementos Finitos para el análisis y diseño de las estructuras civiles. Ofrece una interfaz de usuario intuitiva y potente con muchas herramientas para ayudar en la construcción rápida y exacta de modelos, al mismo tiempo con las técnicas analíticas sofisticadas necesarias para hacer proyectos más complicados (Copyright © Computers and Structures, 2013). El elemento dividido en partes pequeñas posee las siguientes características (Bacuilima G, 2011):

- Geometría: Sistema de referencia.
- Material: Ley constitutiva.
- Condiciones de fronteras esenciales: Apoyos.
- Condiciones de fronteras naturales: Cargas.

SAP2000 está basado en objetos, queriendo decir que los modelos son creados usando miembros que representan la realidad física. Los resultados del análisis y el diseño son informados para el objeto en conjunto, y no para cada sub- elemento que conforma el objeto, suministrando información que es tanto más fácil interpretar como más consecuente con la estructura física (Copyright © Computers and Structures, 2013).

#### **1.4.2.1 OAPI SAP2000**

La Interfaz de Programación de Aplicaciones Abiertas (OAPI por sus siglas en inglés) es una poderosa herramienta que le permite al usuario apropiarse de numerosos procedimientos requeridos para construir, analizar y diseñar modelos y obtener resultados optimizados de los mismos. Además, permite al usuario crear vínculos entre SAP 2000 y softwares de última generación, suministrando una guía para el intercambio de dos flujos de información del modelo con estos programas (Computers & Structures, 2013).

Los aspectos más relevantes de las principales características que ofrece la OAPI se resumen a continuación (Sextos and Balafas, 2011):

- Directo, acoplamiento rápido y sólido con los métodos de diseño y análisis del SAP2000.
- El flujo de datos es de dos vías, ya que puede ser utilizado para facilitar ambos procedimientos pre y post-procesamiento.
- No hay necesidad de usar archivos intermedios, lo que reduce significativamente el tiempo necesario para el intercambio de datos cuando se trabaja en modelos de gran tamaño.
- Compatibilidad con la mayoría de los lenguajes de programación más importantes.

- Transferencia de datos y el control simultáneo de un modelo estructural por diferentes aplicaciones de terceros.
- Desarrollo de aplicaciones de terceros que permanecerán compatible con futuras versiones de SAP2000.
- Capacidad para desarrollar una interfaz personalizada para SAP2000, calibrado para las necesidades del usuario, o para incorporarla en una aplicación que permite la programación del usuario.

#### **1.4.2.2 Uso de la OAPI SAP2000**

Con el fin de hacer uso de las funciones de la OAPI SAP2000 se requiere un fondo de programación sólida, ya que el proceso puede resultar exigente, en términos de conocimientos de programación. Sin embargo, esto no siempre es el caso, cuando se trata de tareas de computación simples. En cualquiera de los casos, la OAPI SAP2000 ofrece una amplia gama de lenguajes de programación que se puede utilizar que cubren la gran mayoría de las modernas opciones de desarrollo de software, incluyendo Visual Basic.NET, Visual Basic para Aplicaciones (VBA), Visual C #, Visual C ++, visual Fortran y MATLAB. Para comenzar con, el acceso a la OAPI requiere nada más que una instalación SAP2000 funcional, ya que todos los componentes que lo forman están instalados de forma automática durante la instalación del programa principal. Su uso, sin embargo, depende en general del lenguaje de programación con que se esté familiarizado en el entorno de programación y que se ha seleccionado para este propósito. Los diferentes entornos de desarrollo integrado (IDE por sus siglas en inglés) ofrecen varias opciones para que el proyecto de programación tenga éxito, adicionalmente, en cada caso, la sintaxis a utilizar y la planificación potencial de la aplicación a desarrollar difieren de acuerdo con las normas del lenguaje. Se debe tener en cuenta que el estilo de programación al que se está acostumbrado tiene que ser adaptado a la forma en que la OAPI está estructurada, sin embargo, los pasos necesarios para realizar una tarea a través de la OAPI se parecen enormemente al procedimiento estándar realizado cuando se trabaja desde SAP2000 (Sextos and Balafas, 2011).

También se proporciona junto con la instalación un archivo con la documentación detallada (CSi\_OAPI\_Documentation.chm) como asistente general en la utilización de la OAPI de SAP2000, que incluye toda la información necesaria que ayudará a acostumbrarse a la programación de la OAPI. Esta guía es para ser utilizada en el desarrollo del software SAP2000, ya que contiene una lista completa de todas las funciones únicas previstas, acompañadas de su sintaxis exacta, una descripción detallada de los argumentos que maneja y un ejemplo comentado de su uso (Sextos and Balafas, 2011).

### 1.4.2.3 Funciones de la OAPI SAP2000

Las funciones de CSi OAPI:



Fig. 1.9 Funciones de la CSi OAPI. Fuente: (Chagoyén Méndez, 2015)

La convención de las funciones de la OAPI, así como las otras numerosas funciones que se encuentran incluidas dentro de las Funciones Generales de la CSi OAPI se describe en (Computers & Structures, 2013).

## 1.5 Conclusiones parciales del capítulo:

- Las torres de telecomunicaciones son estructuras que se conciben principalmente para soportar las grandes y pesadas antenas que se utilizan para la transmisión de señales de radio, televisión, etc. Se clasifican, de acuerdo a su tipología estructural, en torres atirantadas o arriostradas, torres monopolo, y torres autosoportadas.
- El proceso de validación de los datos y el modelo es fundamental para tener un indicador del grado de correlación que se alcanza entre las respuestas medidas y calculadas.
- El MAC, así como otros derivados del mismo, permite contar con un indicador para valorar el grado de correlación alcanzado en la calibración.
- Los métodos empleados para realizar la calibración del modelo son los Métodos Clásicos y los Metaheurísticos. El método más apropiado depende del tipo de problema, del número de variables del mismo (continuas o discretas), de la forma de la función objetivo y de las restricciones.
- Existen gran cantidad de métodos de optimización, dentro de los cuales en esta investigación se utilizará la resolución por mínimos cuadrados de problemas no lineales de funciones.
- En la actualidad se evidencia cada vez más el empleo de potentes softwares para asistir los problemas de optimización aplicables a la calibración de modelos estructurales mediante interfaces y funciones incorporadas a los mismos, ejemplo de estos son MATLAB y SAP2000.

- La OAPI SAP2000-MATLAB permite realizar la modelación y actualización de los modelos a partir de los resultados obtenidos en ensayos modales.

## **Capítulo II: Procedimiento automatizado para la calibración de modelos de torres autosoportadas utilizando la interface OAPI SAP2000 y los recursos de MATLAB, basado en resultados de pruebas modales.**

En este capítulo se realiza la formulación completa del problema de calibración de modelos de torres, que a su vez consiste en la elaboración del algoritmo de interface entre MATLAB y SAP2000 desarrollado en el entorno de programación MATLAB para crear el modelo estructural de la torre objeto de estudio o interactuar con el modelo ya creado en SAP2000 y obtener los parámetros de la respuesta dinámica modelada. Además, se realiza la elección e implementación de la función objetivo y el tipo de optimización apropiados para ser aplicados a este problema. También se explican todos los scripts que hacen posible la calibración, así como las funciones que los complementan.

### **2.1 Introducción**

En la actualidad, cada día aumentan los esfuerzos en aras de contar con técnicas que permitan la calibración de los modelos de estructuras. Estos modelos calibrados permiten con mayor precisión predecir lo que ocurre en las estructuras ante diferentes estados de cargas o modificaciones que se le impongan, así como la identificación de daños en las mismas. También estos procedimientos son realizados obteniéndose muy buenos resultados y a costos muy bajos lo que los hace de gran preferencia en el campo de la ingeniería civil.

Para calibrar estos modelos, como ya se expuso, es necesario contar con datos, a partir de pruebas dinámicas, de la estructura objeto de estudio (frecuencias propias de oscilación, formas modales y radios de amortiguamiento), realizar la modelación de las mismas tratando de seguir correctamente todos los invariantes de la modelación y representar lo más exacto posible la realidad, para así, poder aplicar el procedimiento automatizado que permita estimar los parámetros reales de la estructura minimizando las diferencias entre la respuesta real y la del modelo.

En este trabajo se desarrolla este procedimiento empleando la programación y optimización en MATLAB R2015a, así como la modelación en SAP2000v16 a través de la OAPI SAP2000. Para entender este procedimiento es necesario contar al menos con un nivel de entendimiento del MATLAB, su entorno de trabajo, sus funciones propias y el lenguaje a utilizar en el mismo. Todas las actividades a realizar se explican en este capítulo.

## 2.2 Modelación de torres de telecomunicaciones autosoportadas empleando SAP2000v16.

El programa SAP 2000 es una herramienta poderosa que permite realizar cálculos de estructuras complejas con una gran rapidez y facilidad. La principal característica de SAP2000 es su interfaz gráfica, herramienta potente y amigable. Dispone también de herramientas para visualización en 3D. Sin embargo, esta cualidad implica quizás un menor control y menores posibilidades que otros programas. Se puede afirmar que SAP2000 es un programa muy intuitivo y sencillo, fácil de manejar, pero que no ofrece tantas posibilidades de análisis.

Las bases generales de trabajo con SAP 2000 se pueden resumir en la figura 2.1:

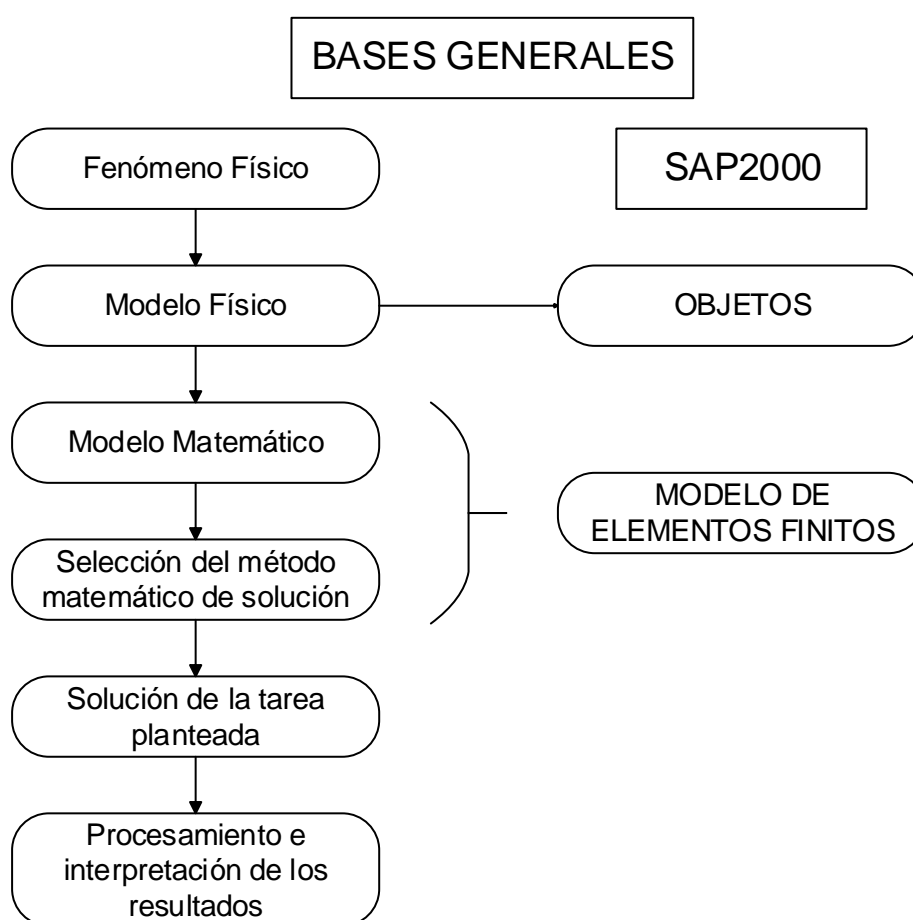


Fig. 2.1 Bases generales de trabajo con SAP 2000. Fuente: (Delgado et al., 2016)

Antes de comenzar a modelar se deben tener bien definidas todas las características de la estructura real como los tipos de perfiles utilizados, tanto en la estructura principal como en los rigidizadores, y la disposición espacial de los mismos, los tipos de uniones (articuladas o empotradas) que presentan entre los mismos. También se debe realizar una auscultación de la torre para conocer si existe algún tipo de daño o debilitamiento en alguna

sección de la misma, conocer las características de la cimentación la cual juega un papel fundamental en este tipo de torres, por último, conocer los materiales de cada elemento, las antenas existentes y la disposición de las mismas en la estructura. Se recomienda hacer un resumen de lo anterior por tramos, debido a que las torres autosoportadas, en su mayoría, la sección transversal en planta es diferente por tramos correspondientes a alturas distintas y recoger todas las características de la torre en una tabla. También es recomendable hacer grupos en Sap2000 donde se recoja los elementos de acuerdo al perfil que se utiliza y su función. (Ver tabla 2.1)

Medidas L	Grupos Correspondientes		
100 x 13	C (Torreta)		TI 1 (11-12)
			4 (1,2,5)
			3 (1,5,6)
			TP (6,7,8)
			PG (5,6,7,8)
			9 (1,2)
			6 (1,2,3,4)
			5 (6,7,8)
			C (11-20)
			C (9-10)
			D (11-20)
			4 (3)
			3 (8,9)
			TI (5)
			5 (10)
			4 (8)
			C (6,7,8)
			C (4,5)
			C (1,2,3)
			TP (5)
			TP (3,4)
			PG (1)
			TP (2)
			TP (1)
			5 (5)
			PG (2,3,4,9,10)
			TI (1)

**C- Columna**  
**D-Diagonal**  
**Números (Fuera del paréntesis) - corresponden a un tipo de elemento**  
**TI – Tranques Interiores.**  
**TP- Tranques Principales.**  
**PG – Patas de Gallo**  
**Números (Dentro del paréntesis) – corresponden al número del tramo.**

Tabla 2.1 Ejemplo de tabla de agrupación de elementos.

## 2.2.1 Variantes de modelación en Sap2000v16

Para realizar la modelación de torres autosoportadas utilizando SAP 2000 se cuentan con 3 variantes:

1. Realizar la modelación manualmente en el entorno de trabajo del SAP añadiendo cada parte de la estructura de la torre como un elemento frame. Este proceder sería muy engorroso debido a que las torres de telecomunicaciones, dependiendo de su altura, cuentan con cientos de elementos estructurales lo que abarcaría una gran cantidad de tiempo. Además, otro aspecto que dificulta este proceder es debido a que las torres son simétricas con respecto a la vertical, pero a medida que se hace en altura se hacen más estrechas lo que dificultaría notablemente la modelación en el entorno de trabajo de SAP 2000.



Para realizar la modelación por esta vía se seguirían los siguientes pasos:

- Abrir el SAP 2000v16.
- Establecer las unidades de trabajo: En SAP2000 siempre se definen primero las unidades de medida con las que se van a trabajar antes de elaborar el modelo, aunque posteriormente en el proceso de creación del mismo estas se puedan cambiar. También desde MATLAB se pueden modificar empleando el comando SetPresentUnits de la OAPI. Por lo tanto, queda a decisión del ingeniero la manera de ajustar las dimensiones.
- Definir las secciones: Se realiza empleando la herramienta (Define).
- Definir los materiales: Después de definidas las secciones de los elementos que componen el modelo se especifica el material correspondiente a cada uno de ellos. Aunque para las propiedades que serán objeto de estimación (variables independientes en el algoritmo de optimización) se puede asignar cualquier valor, se recomienda utilizar los valores de los perfiles metálicos que suministra el fabricante)
- Geometría del modelo estructural: Empleando todas las herramientas de dibujo (Draw) con que cuenta el SAP2000 se crea el modelo del reticulado estructural lo más cercano posible a la realidad y respetando las dimensiones de la estructura real.
- Definición de los apoyos: Empleando la herramienta de asignación del SAP2000 (Assign) se pueden modelar múltiples condiciones de apoyo que deben estar en correspondencia con la realidad. Los apoyos pueden ser considerado como una variable independiente más en el problema de optimización.
- Asignación de grupos: Debido a que las torres de telecomunicaciones cuentan con diferentes y gran cantidad de elementos, es aconsejable agruparlos por grupos según sus semejanzas en perfil utilizando la herramienta (Assign / Assigning to Group... / Add New Group...)
- Modelación de las cargas: Las cargas a definir en el modelo deben ser las que estaban actuando sobre la estructura en el momento que se instrumentó, para que así el modelo represente lo que se va analizar. En este caso solo se asignarán las cargas muertas impuestas por el peso propio de los elementos de la torre y de los elementos estructurales que la conforman, pues aquí hay que destacar que solamente se involucra a las cargas para asignar las masas posteriormente debido a que el caso de carga a analizar es el Modal impuesto por defecto en SAP2000 en la ventana de Define Load Cases. Esto se debe a que los valores que se van a

tomar como resultado del análisis en el SAP2000 van a ser las coordenadas modales y las frecuencias propias de los diferentes modos de oscilación analizados.

- Modelación de las masas: En SAP2000 existen diferentes formas de definir las masas. Estas están disponibles en el programa en las propiedades de los materiales (Material Properties), en las cargas verticales aplicadas (Applied Vertical Loads) o en asignación de masas a objetos (Assigned Object Masses). Para el primer caso se emplea la herramienta Define y en las propiedades de los materiales se coloca el peso por unidad de volumen del material, el programa calcula la masa por unidad de volumen dividiendo entre la gravedad. El segundo caso se emplea a través de la herramienta (Assign / Joint Loads / Forces) para las cargas en la dirección z negativa, aquí la carga es dividida igualmente por la gravedad. El tercer caso se ejecuta también desde la herramienta (Assign/ Joints / Masses). Por otra parte, para especificarle al programa donde encontrar la masa se busca en la herramienta (Define / Mass Source), después se elige si la masa proviene de los elementos y masas adicionales o si es de casos de carga o de ambas partes.
  - Definición del caso de carga: En la herramienta (Define/ Load Case) y dentro de este el Modal en el que se puede especificar el tipo de análisis: Modal, además del número máximo y mínimo de modos de vibración, el tipo de vectores a utilizar, etc.
  - Definición del tipo de análisis: El análisis a ejecutar es el Modal por lo que en la herramienta (Analyze / Set Analysis Options) se selecciona el comportamiento que tendrá, el que interesa es el análisis plano. Luego se corre el análisis en Run Analysis poniendo solo a correr el caso Modal.
  - Fichero del modelo: Se debe guardar este archivo (.sdb) en la carpeta de la función que se creará en MATLAB para interactuar con el mismo pues esta carpeta será añadida al directorio de MATLAB. El nombre colocado debe corresponderse a la ruta definida en esta función para que los comandos de la OAPI declarados abran y operen sobre el modelo.
2. Realizar la modelación a partir de los modelos de torres que SAP 2000v16v tiene incorporado. Este método de modelar es bastante sencillo y rápido, pero cuenta con el inconveniente de que la torre objeto de estudio tiene que ser del tipo del de los modelos que el SAP 2000v16 tiene incorporado porque de lo contrario el modelo estaría lejos de representar la estructura real.

Para realizar la modelación por esta variante se realizan los mismos pasos anteriores excepto que no se tiene que modelar la geometría utilizando la herramienta (Draw), sino

que se abriría el SAP 2000v16 y en la ventana que se muestra seleccionaríamos en (Select template) la plantilla de 3D Trusses y después en (3D Truss Type) se selecciona el tipo de torre de transmisión que represente a la torre objeto de estudio y luego se introducen las dimensiones de la misma.

3. Realizar la modelación en AutoCAD y luego importar el modelo a SAP 2000v16. Este método es una buena práctica y se logra el modelo en un tiempo bastante reducido, lo que se necesita tener conocimiento y habilidad en AutoCAD para poder utilizar las potencialidades de dibujo que este programa brinda, de lo contrario no sería conveniente su utilización.

Para modelar de esta manera se necesita:

- Abrir el AutoCAD.
- Dibujar la torre. Todos los elementos tienen que ser tipo líneas individuales y tienen que estar todas agrupadas en una misma capa diferente de la capa 0 que aparece por defecto en el AutoCAD.
- Después de terminado el dibujo, guardar el archivo en formato (.dxf). Para hacerlo se necesita dar clic en el icono representado por una (A) en la esquina superior izquierda del espacio de trabajo del AutoCAD, luego dar click en (Save As) y luego en la lista desplegable (Files of type) seleccionar el formato (.dxf), seleccionar la ubicación del archivo y luego clic en (Save). Es importante también que la versión del AutoCAD sea más antigua que la del SAP 2000.
- Abrir el SAP 2000v16.
- Seleccionar las unidades de trabajo.
- Seleccionar en (Select Templates) la plantilla en blanco (Blank).
- Seleccionar en la barra de herramienta (File / Import / AutoCAD .dxf File...)
- Buscar el directorio donde se encuentra el dibujo .dxf de AutoCAD y dar clic sobre él y dar clic en Abrir.
- Luego aparece una ventana pequeña pidiendo la dirección en que se importará el dibujo. Seleccionar (Z) y dejar las mismas unidades ya definidas anteriormente o cambiarlas a otras a decisión del ingeniero.
- Después aparece otra ventana donde en Frames se selecciona el layer donde se dibujaron todas las líneas, los demás se dejan en (NONE). Pulsar ok.

Después de este paso se debe revisar bien el modelo para ver si todas las líneas correspondientes a los elementos de las torres están importadas correctamente. En caso

de no estarlo hay que revisar si todas estaban en el dibujo en la misma capa que fue la que se importó y distinta de la capa 0.

Los siguientes pasos son los mismos que los explicados anteriormente exceptuando el correspondiente a (Geometría del modelo estructural).

## 2.3 Optimización para la calibración de MEF.

Una cantidad considerable de literatura ha sido publicada tratando el tema de optimización, para el caso de esta tesis, se hace un resumen de lo expuesto en (Roeck, 2003).

Los métodos de optimización pueden clasificarse en dos grandes grupos: **Optimización local** y **Optimización global**, en este trabajo solo se resumirán las principales características de los primeros, por ser los utilizados para la calibración de modelos de elementos finitos. Los algoritmos que convergen más rápidamente resultan solo en una solución local. Un *mínimo local* es un punto en el cual la función objetivo alcanza su valor más pequeño de los posibles, aunque los algoritmos de optimización local no siempre encuentran el mínimo absoluto (*el mínimo global*). El óptimo global es difícil de identificar porque el conocimiento de la función es usualmente solo local y el comportamiento de esta es desconocido en la región que no ha sido probada por el algoritmo.

También los problemas de optimización pueden clasificarse en **constreñidos** y **no constreñidos**, dependiendo si se le imponen constricciones o no.

En la optimización con constricciones, la cual es el caso de muchos problemas prácticos, las variables de diseño no pueden ser escogidas arbitrariamente, ellas más bien tienen que satisfacer algún requerimiento explícito. Estos requerimientos pueden ser limitaciones físicas en las variables formulados por constricciones o, de manera más general, relaciones entre las variables representadas por constricciones de igualdades o desigualdades lineales o no.

Los problemas de optimización sin constricciones pueden emanar de algunas aplicaciones prácticas y jugar un rol central en la teoría de optimización, pues las optimizaciones con constricciones son, muchas veces, extensiones de algoritmos sin constricciones.

### 2.3.1 Dos estrategias: Búsqueda Lineal (Line Search) y Región de Confianza (Trust Region)

Hay dos estrategias fundamentales para forzar a la convergencia. Estas son la Búsqueda Lineal y la Región de Confianza. Estas dos aproximaciones, difieren en el orden en que las cuales escogen la dirección y la distancia del paso hasta la siguiente iteración. Un algoritmo de búsqueda lineal comienza por fijar la dirección y luego identifica una distancia apropiada,

llamada la distancia del paso. Un algoritmo de región de confianza, en cambio, primero escoge una distancia máxima, llamada el radio de la región de confianza, y luego busca una dirección y una distancia de paso que contenga la mejor vía posible dentro de ese radio. Si este paso prueba ser insatisfactorio, la distancia es reducida y se prueba con otra.

La figura 2.2 muestra ambas aproximaciones en una función de dos dimensiones  $f$  con un valle curvo. En esta se entiende que la dirección que toma la aproximación lineal permite una reducción pequeña de la función  $f$  incluso si un paso óptimo de iteración fue escogido, y por el otro lado el método por la región de confianza itera hasta la minimización dentro de la región de confianza lo cual resulta en una reducción más significativa y por lo tanto en un mejor paso.

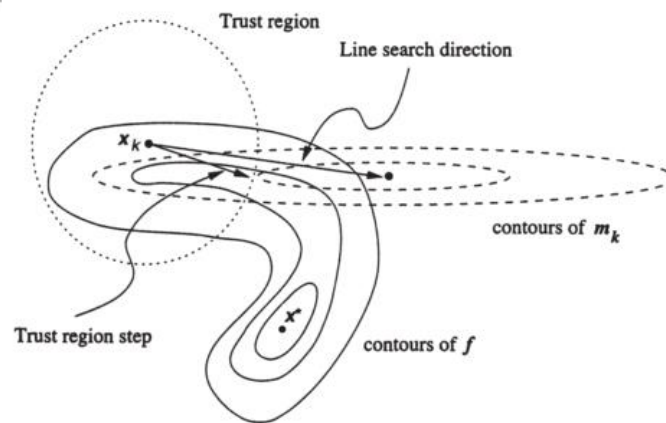


Fig. 2.2 La minimización de una función  $f$  caracterizada por un valle curvo. En el punto actual  $x_k$ , la dirección de búsqueda lineal apunta hacia el minimizador del modelo cuadrático (su contorno elíptico es mostrado con líneas discontinuas). El paso de la región de confianza se queda dentro de la región (círculo con líneas de punto) Fuente: (Roeck, 2003)

### 2.3.2 Clasificación de los algoritmos locales

Un espectro muy amplio de algoritmos existe para optimizaciones sin constricciones. Esos métodos pueden ser ampliamente clasificados en tres categorías, en términos de la información de la derivada que es usada. Se distinguen los **métodos de búsqueda directa** (direct search methods), **los métodos de primer orden** (first-order methods) y los **métodos de segundo orden** (second-order methods).

Los métodos de búsqueda directa usan solo valores de la función objetivo. Son más convenientes para problemas que son muy no lineales o tienen un número de discontinuidades.

Los métodos de primer orden, también llamados métodos de gradiente (gradient methods), requieren además de los valores de la función objetivo, las derivadas de primer orden.

Generalmente son más eficientes cuando la función objetivo es continua en su primera derivada.

Los métodos de segundo orden, como el método de Newton, son solo verdaderamente convenientes cuando la información de segundo orden está disponible y puede ser fácilmente calculada. También el uso de diferenciaciones numéricas es computacionalmente extensivo.

En la figura 2.3 se muestran algunos métodos de optimización, para cada método se presenta el algoritmo principal que se enfoca en las características de la convergencia.

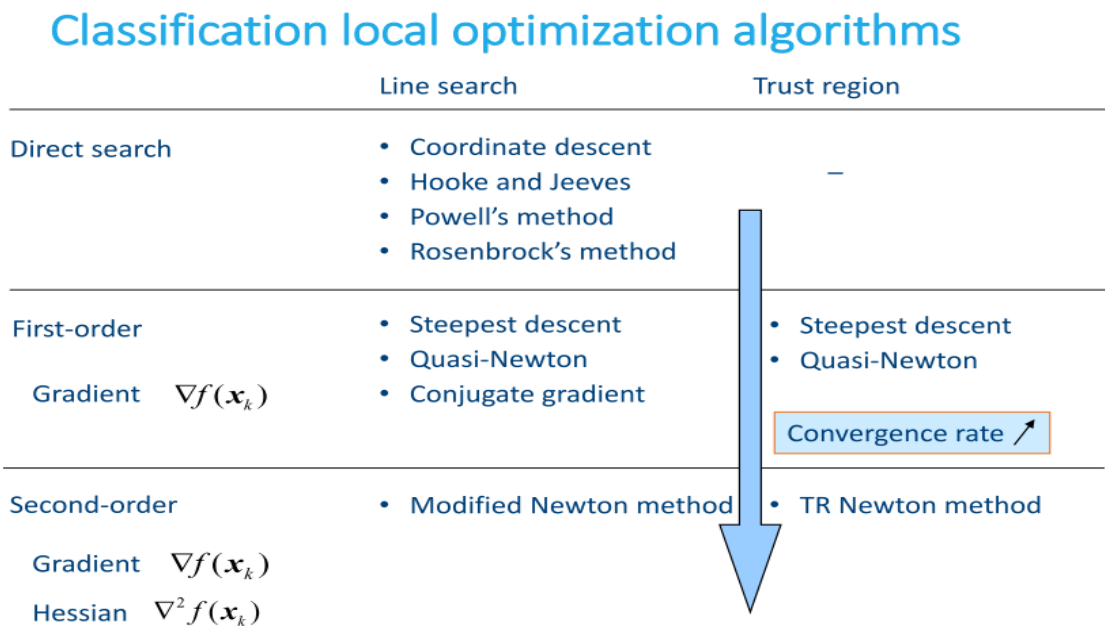


Fig. 2.3 Clasificación de los algoritmos locales. Fuente: (Maes et al., 2017)

### 2.3.3 Problema de mínimos cuadrados

Los métodos para resolver estos problemas son iterativos y cada paso de iteración usualmente requiere la solución de un problema relacionado de optimización de mínimos cuadrados. El problema de mínimos cuadrados no lineales es un caso especial del problema general de optimización.

La principal diferencia entre ellos es si la función objetivo es una suma de funciones lineales, estamos en presencia de optimización lineal por mínimos cuadrados (**linear least square problems**), en cambio, si es una suma de cuadrados de funciones no lineales, entonces conlleva a un problema de optimización por mínimos cuadrados no lineal (**nonlinear least square problems**). El conocimiento de los problemas lineales es

importante porque estos, en muchas ocasiones, emanan como sub-problemas en los algoritmos de la optimización por mínimos cuadrados no lineal.

Existen técnicas específicas para resolver estos algoritmos eficientemente. Los métodos Gauss-Newton y Levenberg-Marquardt aproximan el Hessian usando solo la matriz Jacobiana. Esta aproximación es solo justificada para residuos pequeños y problemas casi lineales. En otros casos, el método híbrido de Newton y el método quasi-Newton deberían ser aplicados.

En la figura 2.4 se muestran, de manera general, los algoritmos para problemas de mínimos cuadrados lineales y no lineales.

## Classifications least squares methods

First-order	<ul style="list-style-type: none"><li>• Gauss-Newton</li><li>• Levenberg-Marquardt</li><li>• Hybrid quasi-Newton (e.g. hybrid quasi- and Gauss-Newton)</li></ul>
Hybrid first- and second-order	<ul style="list-style-type: none"><li>• Hybrid Newton (e.g. hybrid Newton and Gauss-Newton)</li></ul>
Second-order	<ul style="list-style-type: none"><li>• Newton</li></ul>

Fig. 2.4 Clasificación de los algoritmos de mínimos cuadrados. Fuente:(Maes et al., 2017)

Los métodos Gauss-Newton y Levenberg-Marquardt son los algoritmos que MATLAB tienen incorporado con la función *lsqnonlin* y que se pueden utilizar, indistintamente, en esta investigación en los scripts de MATLAB.

El método **Gauss-Newton** excluye el término de segundo orden. Requiere solo los cálculos de solo las derivadas de primer orden. En problemas con residuos muy pequeños, el comportamiento de este método puede ser similar al del método de Newton puro. En particular, en problemas consistentes, el método de Gauss-Newton puede, últimamente, lograr un rango de convergencia cuadrático, a pesar de que solo las primeras derivadas son utilizadas. Para residuos de moderados a largos o problemas muy no lineales, la velocidad de convergencia puede ser mucho menor que el método de Newton.

El algoritmo de **Levenberg-Marquardt** puede ser imaginado como una modificación con región de confianza al método de Gauss-Newton. El uso de una región de confianza evade la principal debilidad del método de Gauss-Newton en su comportamiento cuando el Jacobiano presenta un rango deficiente. Con la inclusión de la región de confianza el método de Gauss-Newton es estabilizado. En este método el término de segundo orden de la Hessian es todavía ignorado, por lo que la convergencia de este método es muy similar al de Gauss-Newton. Para problemas con residuos pequeños o casos casi lineales se logran rápidas convergencias; pero si los residuos son grandes o los problemas son muy no lineales, este método tendrá convergencias lentas. También existen otros métodos más robustos y complejos para estos casos.

### 2.3.4 Función *lsqnonlin* en MATLAB.

La función *lsqnonlin* del toolbox de optimización de MATLAB es la función de optimización a emplear en esta investigación.

La función *lsqnonlin* resuelve por mínimos cuadrados problemas no lineales de funciones o de ajuste de datos, lo que se corresponde con el tipo de problema al que esta investigación se enfrenta. A través de la misma se realiza la optimización mediante el ajuste de datos o valores obtenidos del software SAP2000v16 empleando la interface OAPI realizada con MATLAB.

Aunque una función de mínimos cuadrados puede ser minimizada con una técnica de optimización sin constricciones, existen en MATLAB algoritmos específicos para problemas de mínimos cuadrados con el objetivo de mejorar la eficiencia de las iteraciones para la solución del procedimiento.

**Para algoritmos de mediana escala** los métodos de Gauss-Newton y Levenberg-Marquardt son suministrados por MATLAB. Los dos son implementados como métodos de búsqueda directa.

En el primero la dirección de búsqueda es calculada resolviendo:

$$(J_k^T J_k) p^{GN} = -J_k^T r_k \quad \{2.1\}$$

En el segundo una dirección regularizada:

$$(J_k^T J_k + u_k I) p^{LM} = -J_k^T r_k \quad \{2.2\}$$

$J_k$  – Matriz Jacobiana.

$r_k$  – Residuos.

$I$  – Matriz idéntica (1 en la diagonal principal, 0 fuera de la diagonal principal).

$p^{GN}$  – Distancia del paso de Gauss-Newton.



$p^{LM}$  – Distancia del paso de Levenberg-Marquardt.

$u_k$  – Parámetro que limite el tamaño de  $p$ ; si  $u_k = 0$  entonces  $p^{GN}=p^{LM}$ .

es utilizada para mejorar la condición del problema. La dirección de la búsqueda de Gauss-Newton es calculada exactamente con una descomposición QR de  $J_k$ , sin formar la ecuación 2.1 explícitamente, requiriendo solo pocas iteraciones y tiempo de cálculo para problemas a medias escalas. El método de Levenberg-Marquardt requiere un poco más de cálculos a partir de que  $u_k$  tiene que ser controlada en cada iteración.

Subsecuentemente, una búsqueda en una dirección es realizada a través de la dirección de búsqueda con un algoritmo iterativo.

El Jacobino puede ser suministrado por el usuario o aproximado con una diferencia finita de cálculos.

**Los algoritmos de gran escala** son métodos basados en la estrategia de región de confianza. Análogamente, así como en la minimización general, una función modelo cuadrática es construida en cada iteración que aproxima la función objetivo. En la aplicación de mínimos cuadrados, la Hessian es aproximada con la formulación de Gauss-Newton. Las derivadas de segundo orden son por consiguiente no usadas.

En particular, el algoritmo es implementado tal que la dirección de Gauss-Newton se usa para definir el subespacio de dos dimensiones. Esta dirección es la solución de:

$$\min_p \|J_k p + r_k\|^2 \quad \{2.3\}$$

El método de los gradientes conjugados pre-condicionados es usado para aproximadamente e iterativamente resolver la correspondiente ecuación normal 2.1, aunque formándola explícitamente.

A diferencia de los algoritmos de mínimos cuadrados de mediana escala, los de larga escala pueden resolver problemas con constricciones. En este caso una dirección modificada de Gauss-Newton es calculada, tomando las constricciones en cuenta.

También, en este caso, el Jacobiano puede ser especificado por el usuario o aproximado con una diferencia finita de cálculos.

En la calibración de modelos de elementos finitos, se resuelven problemas de mínimos cuadrados no lineales. Para esta aplicación los algoritmos a grandes escalas son usados, principalmente porque los algoritmos a medianas escalas no permiten restricciones a las variables. El hecho es que, las direcciones de búsqueda son muy sensitivas a aproximaciones y errores, y si las variables de diseño no son constreñidas en la búsqueda directa en todas las direcciones, esto resultaría en pasos iterativos y grandes que van más

allá de límites realistas y de lo que busca principalmente el problema. Como resultado, el proceso de optimización puede divergir. Incluso el método de Levenberg-Marquardt, implementado como versión de búsqueda directa, puede no ser suficiente para resolver este problema. El método de la región de confianza es, para este tipo de problemas, mucho más robusto.

Particularmente, en este método las variables son limitadas automáticamente por las constricciones de la región de confianza y por lo tanto no se toman grandes pasos iterativos que extremadamente grandes ni irrealistas. Lo que evita que el proceso no converja a una solución exacta. Adicionalmente, más constricciones pueden ser aplicadas, no solo necesariamente por lo robusto del método sino, principalmente, para imponer limitaciones específicas a las variables.

Usando el algoritmo de la región de confianza para solución de problemas de mínimos cuadrados, sin embargo, implica que los subproblemas son resueltos con un algoritmo iterativo de gradientes conjugados. Mientras que este algoritmo es muy eficiente para problemas a grandes escalas, no lo es para problemas a mediana escala para los cuales otros algoritmos existen, pero no son implementados por MATLAB. Por otro lado, usando el método de región de confianza no se tienen que realizar búsquedas directas iterativas.

Al ser que el algoritmo robusto sea más importante que la velocidad de convergencia, el método de la región de confianza a grandes escalas es escogido para la resolución de problemas de calibración de modelos de elementos finitos.

#### **2.3.4.1 Ecuación, sintaxis y resultados.**

La función *lsqnonlin* como fue definida anteriormente, resuelve por mínimos cuadrados problemas no lineales de funciones o de ajuste de datos, lo que se corresponde con el tipo de problema al que esta investigación se enfrenta. A través de la misma se realiza la optimización mediante el ajuste de datos o valores obtenidos del software SAP2000 empleando la interface OAPI realizada con MATLAB.

La ecuación queda definida de la siguiente forma:

$$\min_x \|f(x)\|_2^2 = \min_x \sum_i f_i^2(x) \quad (2.4)$$

Tal que  $lb \leq x \leq ub$ .

O replanteada en término de vectores.

$$\min_x \|f(x)\|_2^2 = \min_x (f_1(x)^2 + f_2(x)^2 + \dots + f_n(x)^2) \quad (2.4.1)$$

Con límites opcionales inferior (Tarantola) y superior (Monroy Rincón and Arrubla Tovar) sobre los componentes de  $x$ .

Sintaxis:

- $x = \text{lsqnonlin}(\text{fun}, x0)$
- $x = \text{lsqnonlin}(\text{fun}, x0, lb, ub)$
- $x = \text{lsqnonlin}(\text{fun}, x0, lb, ub, \text{options})$
- $x = \text{lsqnonlin}(\text{problem})$
- $[x, \text{resnorm}] = \text{lsqnonlin}(\dots)$
- $[x, \text{resnorm}, \text{residual}] = \text{lsqnonlin}(\dots)$
- $[x, \text{resnorm}, \text{residual}, \text{exitflag}] = \text{lsqnonlin}(\dots)$
- $[x, \text{resnorm}, \text{residual}, \text{exitflag}, \text{output}] = \text{lsqnonlin}(\dots)$
- $[x, \text{resnorm}, \text{residual}, \text{exitflag}, \text{output}, \text{lambda}] = \text{lsqnonlin}(\dots)$
- $[x, \text{resnorm}, \text{residual}, \text{exitflag}, \text{output}, \text{lambda}, \text{jacobian}] = \text{lsqnonlin}(\dots)$

Descripción:

En lugar de calcular el valor  $\|f(x)\|_2^2$  (La suma de los cuadrados), `lsqnonlin` requiere la función definida por el usuario para calcular el vector de función con valores de:

$$f(x) = \begin{bmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_n(x) \end{bmatrix} \quad (2.4.2)$$

Es decir, en términos de vectores, se puede repetir este problema de en donde  $x$  es un vector o una matriz y  $f(x)$  es una función que devuelve un valor del vector.

Explicación de la sintaxis:

- $x = \text{lsqnonlin}(\text{fun}, x0)$  comienza en el punto  $x0$  y encuentra un mínimo de la suma de los cuadrados de las funciones descritas en *fun*. Se devuelve en *fun* un vector de valores y no la suma de los cuadrados de los valores. (El algoritmo implícitamente suma y eleva al cuadrado  $\text{fun}(x)$ ).
- $x = \text{lsqnonlin}(\text{fun}, x0, lb, ub)$  define un conjunto de límites inferior y superior en las variables de diseño en  $x$ , por lo que la solución siempre está en el rango de  $lb \leq x \leq ub$ . Cuando no existan límites se pasan matrices vacías en  $lb$  y  $ub$ .
- $x = \text{lsqnonlin}(\text{fun}, x0, lb, ub, \text{options})$  reduce al mínimo con las opciones de optimización especificadas en la estructura de *options*. Se utiliza *optimset* para establecer estas opciones. Para este trabajo se va utilizar la optimización mediante el Jacobiano definido

por el usuario en *fun*. La función *fun* debe devolver, en el segundo argumento de salida, el valor del Jacobiano en una matriz J.

J = Jacobiano de la función evaluada en el último valor de x.

Si *fun* devuelve un vector (matriz) de m componentes y x tiene una longitud n, donde n es la longitud de x0, el Jacobiano J es una matriz de m -por- n en donde J (i, j) es la derivada parcial de F(i) con respecto a x(j). El Jacobiano J es la transpuesta del gradiente de F, es decir, la metodología que se puede emplear para determinar los valores óptimos de la función que se requiere minimizar, es aproximar los valores de su gradiente a cero (Afonso Aspiro, 2016, Bacuilima G, 2011).

- $x = lsqnonlin(problem)$  encuentra el mínimo del problema, donde el problema es una estructura descrita en argumentos de entrada. Crea la estructura del problema exportando un problema de Optimization APP.
- $[x, resnorm] = lsqnonlin(...)$  devuelve el valor de la potencia 2 del residuo en x: suma  $(fun(x))^2$ .
- $[x, resnorm, residual] = lsqnonlin(...)$  devuelve el valor residual de *fun*(x) en la solución x.
- $[x, resnorm, residual, exitflag] = lsqnonlin(...)$  devuelve un valor *exitflag* que describe la condición de salida. Los posibles valores de *exitflag* y las condiciones de salida correspondientes son:

1 *lsqnonlin* converge a una solución.

2 El cambio en x fue más pequeño que la tolerancia especificada.

3 El cambio en los residuos fue más pequeño que la tolerancia especificada.

4 Dirección de búsqueda de cómputo muy pequeña.

0 Demasiadas evaluaciones de funciones o iteraciones.

-1 Parado por *output/plot* de la función.

-2 Los límites son incongruentes.

-4 La búsqueda de línea no podía reducir suficientemente el residual a lo largo de la dirección de búsqueda actual.

- $[x, resnorm, residual, exitflag, output] = lsqnonlin(...)$  devuelve una estructura *output* que contiene información acerca de la optimización.
- $[x, resnorm, residual, exitflag, output, lambda] = lsqnonlin(...)$  devuelve una estructura *lambda* cuyos campos contienen los multiplicadores de Lagrange en la solución x.

- $[x, resnorm, residual, exitflag, output, lambda, jacobian] = lsqnonlin(...)$  devuelve el Jacobiano de *fun* en la solución *x*.

Si los límites de entrada especificados por un problema son incompatibles, la salida de *x* es *x0* y las salidas *resnorm* y residuales son []. Los componentes de *x0* que violan los límites  $lb \leq x \leq ub$  se restablecen en el interior del rango definido por los límites. Los componentes con respecto a los límites no se modifican. Todo lo anterior explicado se encuentra en (Afonso Aspiro, 2016) y en las ayudas que posee el MATLAB.

## 2.4 Procedimiento automatizado de optimización para la calibración implementado en MATLAB

De manera general, el procedimiento de calibración cuenta con 4 scripts de MATLAB de los cuales, sus aplicaciones principales, quedan resumidas en la tabla 2.2:

Scripts de Matlab	Aplicaciones
main_mdof_torre.m	<ul style="list-style-type: none"> <li>• Inicialización de parámetros</li> <li>• Calibración del modelo (upd_mdof_torre.m)</li> <li>• Plotea resultados finales</li> </ul>
data_mdof_torre.m	<ul style="list-style-type: none"> <li>• Define el modelo de múltiples grados de libertad (= realidad)</li> <li>• Define los parámetros dinámicos de referencia.</li> </ul>
upd_mdof_torre.m	<ul style="list-style-type: none"> <li>• Se especifican opciones para la calibración (optimset)</li> <li>• Se define la función objetivo (@objfun)</li> <li>• Calcula los parámetros modales (eigfem_mdof_torre.m)</li> <li>• Define la función de salida (@outfun)</li> <li>• Realiza la optimización no lineal (lsqnonlin.m)</li> </ul>
eigfem_mdof_torre.m	<ul style="list-style-type: none"> <li>• Comandos OAPI. Extrae los parámetros modales del modelo en Sap2000 para ser utilizados luego por MATLAB en la optimización.</li> </ul>

Tabla 2.2 Scripts de MATLAB para la calibración.

Todos estos scripts tienen que estar agrupados en una misma carpeta y se recomienda que el modelo de la torre esté ubicado en otra carpeta dentro de esta.

También es necesario introducirle al “path” de MATLAB las funciones que contiene el StaBIL 2.0 porque dentro de estos scripts, en ocasiones es necesario utilizar estas funciones para hacer posible el procedimiento de calibración. Estas funciones son:

- selectdof.m (Selecciona los grados de libertad del modelo)
- asmkm\_mdof.m (Ensambla las matrices de masa y rigidez del sistema de múltiples grados de libertad (MDOF system))

Otro script importante con que cuenta el procedimiento es (**sens\_anal\_torre.m**) el cual se encarga de presentar la matriz MAC. Este script es explicado más detalladamente en posteriores epígrafes.

#### 2.4.1 Parámetros que intervienen en la optimización

En todo problema deben declararse tanto las variables de diseño, las variables de estado, la función objetivo como las restricciones del problema. Las variables de diseño (independientes) a tener en cuenta en esta investigación son las propiedades de los materiales; en dependencia del material empleado (hormigón y/o acero). Otras variables que se pudieran tener en cuenta son:

- Propiedades de peso por unidad de volumen: Densidad.
- **Propiedades Isotrópicas:** El módulo de elasticidad ( $E$ ), el módulo de rigidez al cortante ( $G$ ), el coeficiente de Poisson ( $\mu$ ) y el coeficiente de dilatación térmica.
- Propiedades mecánicas: La resistencia específica del hormigón ( $f'c$ ) en el caso de elementos estructurales de hormigón armado o el límite de fluencia ( $F_y$ ) y la resistencia última ( $H_e$  and  $F_u$ ) para el caso de elementos metálicos.
- **Condiciones de apoyos:** para el caso de que se asignen apoyos elásticos, la rigidez  $k$  de los mismos podría determinarse al finalizar la optimización.
- Se pueden incluir variables como los espesores de los elementos componentes del modelo estructural, determinándose así la reducción de los mismos.

Las variables de estado (dependientes) a tener en cuenta en esta investigación son los parámetros modales como:

- Las frecuencias naturales de oscilación de las formas modales seleccionadas, las coordenadas modales de estas últimas en los puntos de interés, así como la curvatura modal en cada uno de ellos.

Las Restricciones se tienen en cuenta pues estas permiten proporcionar los límites que condicionan el problema, debido a que las variables independientes no pueden tomar cualquier valor ya que están conformadas con las propiedades de los materiales de los

elementos estructurales. Se definen límites inferiores, superiores y en algunos casos pudieran emplearse restricciones de desigualdad. Estas restricciones facilitan que las respuestas obtenidas durante la calibración sean lógicas y evitará la realización de corridas adicionales y gran uso de tiempo de cómputo.

#### 2.4.2 Script (main\_mdof\_torre.m)

Este script es el principal de todos, debido a que es el único que se ejecuta en MATLAB. Se encarga, automáticamente, de ir abriendo cada uno de los otros scripts e ir ejecutando cada una de las funciones existente necesarias en la calibración. Al mismo tiempo se van presentando en el Command Windows de MATLAB los números de iteraciones que se han ejecutado y los residuos para cada una de ellas. Con la ayuda de la siguiente tabla se explica cada comando programado.

<pre> Tic  close all, clear all, clc  data_mdof_torre;  E_0 = [1.95*10^8 1.95*10^8 1.95*10^8]; E_min = [1.92*10^8 1.92*10^8 1.92*10^8]; E_max = [2.4*10^8 2.4*10^8 2.4*10^8];  alpha = 1; beta = 1;  save upd_param_mdof_ern alpha beta E_0  load data_mdof_ern nparameters;  T_0 = ones(1,nparameters); T_min = E_min / E_0; T_max = E_max / E_0; </pre>	<p>Se define el comando tic, el cual, al final del script tiene que terminar con toc.</p> <p>Con estos comandos se limpia el workspace, el command Windows y cualquier figura o gráfico que se esté abierto.</p> <p>Se ejecuta el script data_mdof_torre.</p> <p>Se especifican los límites superiores e inferiores y los valores iniciales para la optimización. En el caso de utilizar el algoritmo de <i>Levenberg-Marquardt</i> no se tienen que definir constricciones al problema.</p> <p>Se definen los factores de peso para las formas y frecuencias modales.</p> <p>Se guardan los parámetros alpha, betha y E_0 en un archivo .mat de MATLAB para ser cargados en cualquier parte de la programación por otro script.</p> <p>Se carga del archivo .mat creado al ejecutar el script data_mdof_torre la variable nparameters que representa el número de parámetros a calibrar.</p> <p>Se especifican los límites superiores e inferiores y el punto de partida para la calibración, pero supuestamente sin unidades para que los tome la función de optimización.</p>
---	--

<pre> [T_upd] = upd_mdof_torre(T_0,T_min,T_max);  E_upd = E_0 * T_upd;  load data_mdof_ern DOF_ref phi_ref omega_ref [phi_0,omega_0,DOF]=eigfem_mdof_torre(E_0); [phi_upd,omega_upd,DOF]=eigfem_mdof_torre(E_upd);  modeSel=1:5;  phi_ref=phi_ref(:,modeSel); omega_ref=omega_ref(modeSel); phi_0=phi_0(:,modeSel); omega_0=omega_0(modeSel); phi_upd=phi_upd(:,modeSel); omega_upd=omega_upd(modeSel);  DOF_sensors=[[62,97,122,128,203,228,245,263]+0.01,[63,98,123,129]+0.02].';  L_ref=selectdof(DOF_ref,DOF_sensors); L_0=selectdof(DOF,DOF_sensors);  disp('Elasticity modulus initial [kN/m2]') fprintf('%15.2f',E_0) fprintf('\n') disp('Elasticity modulus updated [kN/m2]') fprintf('%15.2f',E_upd) fprintf('\n') fprintf('\n') disp('Identified natural frequencies [Hz]') fprintf('%9.2f',omega_ref/2/pi) fprintf('\n') disp('Calculated natural frequencies - (initial) [Hz]') fprintf('%9.2f',omega_0/2/pi) fprintf('\n') disp('Calculated natural frequencies - (updated) [Hz]') fprintf('%9.2f',omega_upd/2/pi) fprintf('\n') </pre>	<p>Se ejecuta en el script la función upd_mdof_torre asignándole los parámetros de la optimización.</p> <p>Se calcula al finalizar la optimización los valores de las variables de diseño calibradas. Aquí termina la calibración.</p> <p>Estos pasos de aquí en adelante solo son para presentar y plotear los resultados</p> <p>Se cargan los valores de las frecuencias y formas modales de referencia, se calculan para los valores iniciales y para los valores calibrados.</p> <p>Se seleccionan los modos a tener en cuenta para estas formas y frecuencias modales.</p> <p>Se calculan las formas modales y frecuencias iniciales y calibradas para los modos seleccionados.</p> <p>Se seleccionan las posiciones de los sensores en la torre, para realizar estos se tienen que introducir los nodos y luego +0.01 para decir que estos miden en la dirección de "x" y +0.02 para la dirección de "y". En el caso de colocar sensores en la dirección de "z" hay que agregar +0.03.</p> <p>Se definen las matrices L de referencia e inicial.</p> <p>Estos comandos son para presentar los resultados en el command windows como las variables de diseño iniciales y calibradas, las frecuencias iniciales, calibradas y de referencia.</p>
--	--



<pre> disp('Error natural frequencies - (initial) [Hz]') fprintf('%9.2f', (omega_0-omega_ref)/2/pi) fprintf('\n') disp('Error natural frequencies - (initial) [%]') fprintf('%9.2f', (omega_0- omega_ref)./omega_ref*100) fprintf('\n') disp('Error natural frequencies - (updated) [Hz]') fprintf('%9.2f', (omega_upd-omega_ref)/2/pi) fprintf('\n') disp('Error natural frequencies - (updated) [%]') fprintf('%9.2f', (omega_upd- omega_ref)./omega_ref*100) fprintf('\n') fprintf('\n') disp('MAC (initial) [-]') fprintf('%15.5f',diag(mac(L_0*phi_0,L_ref*phi_ ref))) fprintf('\n') disp('MAC (updated) [-]') fprintf('%15.5f',diag(mac(L_0*phi_upd,L_ref*ph i_ref))) fprintf('\n')  % PLOT RESULTS load results_mdof T_iter resnorm_iter  figure plot(resnorm_iter,'r-','linewidth',2) hold on ylim([0,9e-4]) xlim([0,10]) xlabel('Iteration Number') ylabel('Objective function') set(gca,'fontsize',11)  toc </pre>	<p>Se presentan los valores de la diagonal principal de las matrices MAC iniciales e calibradas.</p> <p>Se cargan los residuos del script upd_mdof_torre que este los guarda en un archivo .mat nombrado results_mdof.</p> <p>Se presenta una gráfica donde se presentan los residuos vs el número de iteración.</p> <p>Complemento de la función tic que termina con presentar el tiempo total de la calibración.</p>
--	--

En el caso de que se cambie el nombre de alguna variable, en necesario cambiarla en todos los demás scripts donde esta esté presente, de lo contrario se puede encontrar con erros en la calibración que el MATLAB presenta en ese caso.

### 2.4.3 Script (data\_mdof\_torre.m)

Este script es, principalmente, el que le aporta, al procedimiento general, las propiedades reales las cuales el modelo en Sap2000 debe presentar al final de la calibración. Se explica a partir de la siguiente tabla:

<pre> close all, clear all, clc  x_ref=[2.1*10^8 2.05*10^8 2.0*10^8 1.98*10^8 1.99*10^8 2.1*10^8 1.96*10^8];  [phi_ref,omega_ref,DOF_ref]=eigfem_mdof_torre( x_ref);  nparameters=7;  modSel=(1:5); phi_ref=phi_ref(:,modSel); omega_ref=omega_ref(modSel);  save data_mdof_ern nparameters DOF_ref phi_ref omega_ref </pre>	<p>Con estos comandos se limpia el workspace, el command Windows y cualquier figura o gráfico que se esté abierto.</p> <p>Se definen los parámetros a calibrar de referencias, los cuales son los que deben obtenerse al finalizar la calibración.</p> <p>Se evalúan estos parámetros de referencia en la función eigfem_mdof_torre definida en el script eigfem_mdof_torre y se extraen los parámetros modales de referencia. En el caso de tener las propiedades dinámicas reales solo hay que introducirlas en forma de matriz y se puede obviar este paso y el anterior.</p> <p>Se seleccionan la cantidad de variables o parámetros a calibrar.</p> <p>Se seleccionan los modos a tener en cuenta en la calibración y se calculan los parámetros modales para estos modos.</p> <p>Se guardan los parámetros modales de referencia en un archivo .mat para ser cargados luego en los demás scripts.</p>
--	---

#### 2.4.4 Script (upd\_mdof\_torre.m)

Este script es el más extenso debido a que, de manera general:

- Especifica las opciones para actualizar (optimset.m)
- Define la función objetivo (@objfun)
- Calcula los parámetros modales en interacción con (eigfem\_mdof\_torre.m)
- Definición de la función de salida (@outfun)
- Permite la optimización no lineal utilizando (lsqnonlin.m)
- Incorpora o no, en sus funciones de actualización, la matriz Jacobiana.

Toda su programación queda explicada en la siguiente tabla:

```

function [T_upd]=upd_mdof_torre(T_0)
function [T_upd]=upd_mdof_torre(T_0,T_min,T_max)
x)

TolFun=1e-16;
TolX=1e-8;

options=optimset('Algorithm','levenberg-marquardt','ScaleProblem','Jacobian','Display','iter-detailed','Jacobian','off','TolFun',TolFun,'TolX',TolX,'OutputFcn',@outfun);

load data_mdof_ern nparameters

[T_upd,resnorm,residual,exitflag,output]=lsqnonlin(@objfun,T_0,-inf*ones(1,nparameters),inf*ones(1,nparameters),options);

[T_upd,resnorm,residual,exitflag,output]=...lsqnonlin(@objfun,T_0,T_min,T_max,options);

function [res]=objfun(T)

load upd_param_mdof_ern E_0

Evec = T .* E_0;

```

Se define la función upd\_mdof\_torre la cual es evaluada en los parámetros que se definen en el script main\_mdof\_torre. Se puede presentar de dos maneras, una donde existan constricciones y otra donde no las haya. La segunda función es la variante donde existen constricciones.

Se definen las tolerancias en los residuos de la función objetivo y las tolerancias de la distancia del paso. Para la iteración que los residuos o la distancia del paso sea menor que la tolerancia especificada, entonces la optimización se detendrá.

Se definen las opciones que tendrá la optimización, el algoritmo a emplear, si no se define se tomará automáticamente el Trust Region - Reflective Algorithm, también se pueden definir algunas características propias de estos algoritmos, que para el algoritmo escogido las modificaciones que se le pueden realizar son diferentes, estas se pueden encontrar en las ayudas del MATLAB, también se define si la matriz Jacobiana será suministrada o no, si se desea ir mostrando los valores de cada iteración, se definen las tolerancias y la función de salida.

Para el caso de utilizar el algoritmo de Levenberg-Marquardt se cargan el número de parámetros a calibrar del archivo .mat creado al final del script data\_mdof\_torre.

Se define el algoritmo de optimización a emplear, la función objetivo a utilizar, el valor inicial y las constricciones y las opciones de optimización. Para el caso que no se necesiten constricciones se tiene que definir como menos infinito y más infinito. En cursiva se muestra una variante donde se definen las constricciones.

Se define la función objetivo, que es la función que obtiene los residuos por frecuencias y formas modales.

Se cargan del archivo .mat creado por el script main\_mdof\_torre los valores iniciales de las variables a calibrar.

Se define el valor de la variable de diseño para cada iteración.

```

load data_mdof_ern nparameters DOF_ref phi_ref
omega_ref

model = 3; % 1 = analytical, 2 = ANSYS, 3 =
OAPI SAP2000 MatLab

switch model
    case 1 % ANALYTICAL MODEL
        mvec=m_ref*ones(nfloor,1);
        [phi,omega]=eigfem_mdof(mvec,kvec);

    case 2 % ANSYS MODEL
        ansys_dir='../..\ansys\exercise2';
        [phi,omega]=getansys(kvec,ansys_dir);

    case 3 % OAPI SAP2000 MatLab MODEL
        [phi,omega,DOF] =
eigfem_mdof_torre(Evec);
end

DOF_sensors=[[62,97,122,128,203,228,245,263]+0.01,[63,98,123,129]+0.02].';

L_ref=selectdof(DOF_ref,DOF_sensors);
L=selectdof(DOF,DOF_sensors);

MAC=mac(L_ref*phi_ref,L*phi);
threshold=0.7;
mode_ref_sel=(1:size(MAC,1))';
mode_sel=zeros(size(MAC,1),1);
for ind1=1:length(mode_ref_sel)
    [max_,ind_max]=max(MAC(ind1,:));
    if max_>=threshold
        mode_sel(ind1)=ind_max; % Match found
    else
        disp(['No mode match found for
measured mode ' int2str(ind1) '.'])
    end
end
mode_ref_sel(mode_sel==0,:)=[];
mode_sel(mode_sel==0,:)=[];
if length(unique(mode_sel))~=length(mode_sel)
    disp('At least one calculated mode is
matched to multiple measured modes.')
end
phi_ref=phi_ref(:,mode_ref_sel);
omega_ref=omega_ref(mode_ref_sel);
phi=phi(:,mode_sel); omega=omega(mode_sel);
nm=length(omega); % number of modes

gamma=zeros(nm,1);

```

Se cargan las propiedades dinámicas de referencias guardadas en un archivo .mat salvado por el script data\_mdof\_torre.

Se selecciona el tipo de interacción de MATLAB con el programa de elementos finitos, en este caso se selecciona 3 por corresponder con la interacción SAP2000-MATLAB

Se calculan para el tipo de interacción seleccionada los parámetros dinámicos para cada iteración.

En este caso se realiza a partir de evaluar Evec en la función eigfem\_mdof\_torre creada en el script del mismo nombre.

Se definen las posiciones de los sensores de la misma manera que en el script main\_mdof\_torre.

Se calculan las matrices L de referencia y las correspondientes a cada iteración.

Se calcula la matriz MAC para las frecuencias modales de referencia y las calculados en cada iteración. En el caso que el valor más grande de la matriz MAC obtenida sea menor que el valor threshold o límite define se presenta en el command Windows que no se ha encontrado ninguna coincidencia para los modos medidos.

Se calculan las formas y las frecuencias modales de referencia y las calculadas en cada iteración para los modos seleccionados. También se definen el número de modos.

Se calculan los factores de escala para las formas modales.

<pre> for ind1=1:nm  gamma(ind1)=(L_ref*phi_ref(:,ind1)).'*L*phi(:,ind1)/norm(L_ref*phi_ref(:,ind1),2)^2; end  lambda=omega.^2; lambda_ref=omega_ref.^2;  load upd_param_mdof_ern alpha beta  res=sqrt(alpha/2)*(lambda- lambda_ref)./lambda_ref; for ind1=1:nm  wf=sqrt(beta/2)/norm(gamma(ind1)*L_ref*phi_ref(:,ind1),2);      % weighting factor res=[res;wf*(L*phi(:,ind1)- gamma(ind1)*L_ref*phi_ref(:,ind1))]; end  function stop=outfun(T,optimValues,state) stop=false; % Save intermediate results iterations if strcmp(state,'init')     T_iter=T';     resnorm_iter=optimValues.resnorm;     save results_mdof T_iter resnorm_iter elseif strcmp(state,'iter')     load results_mdof T_iter resnorm_iter     T_iter=[T_iter;T'];  resnorm_iter=[resnorm_iter;optimValues.resnorm ];     save results_mdof T_iter resnorm_iter end </pre>	<p>Se cargan los factores alpha y betha definidos en el script main_mdof_torre a partir del archivo .mat creado en el mismo.</p> <p>Se calculan los residuos de la función objetivo para las frecuencias y las formas modales en los modos seleccionados.</p> <p>Se define la función de salida de los resultados donde se guardan los residuos en un archivo .mat para cada iteración, los que se van acumulando hasta que termina la optimización.</p>
---	--

### 2.4.5 Script (eigfem\_mdof\_torre.m)

Los parámetros modales que proporciona el modelo en SAP 2000 son exportados hacia MATLAB a través de la CSI OAPI Documentation donde todas las funciones que esta posee se encuentran en (Computers & Structures, 2013). Luego de que MATLAB reconozca estas variables las mismas son utilizadas en el procedimiento de optimización.

La OAPI SAP2000 dispone de numerosas funciones para trabajar con análisis modal, dentro de estas las que más interesan están vinculadas a la extracción de parámetros modales de la estructura objeto de estudio debido que de estos parámetros depende el

proceso de calibración. La siguiente tabla resume los comandos programados en este script:

<pre>function [phi,omega,DOF] = eigfem_mdof_torre(x_moduli)  feature('COM_SafeArraySingleDim', 1);  feature('COM_PassSafeArrayByRef', 1);  SapObject =actxserver('sap2000v16.SapObject');  SapObject.ApplicationStart; % Open SAP  ret = SapObject.SapModel.File.OpenFile('C:\Users\Dr\ Desktop\TODO ALE\Tesis 2016\Matlab files\Tower model E calibrated\Najasa Modal continua.sdb');  SapModel = SapObject.SapModel;  ret = SapModel.SetModelIsLocked(Copyright © Computers and Structures);  ret = SapModel.Analyze.CreateAnalysisModel;  MATERIAL_STEEL = 1; % steel = 1, concrete = 2 ret = SapModel.PropMaterial.SetMaterial('H1', MATERIAL_STEEL);  x1 = x_moduli(Copyright © Computers and Structures); ret = SapModel.PropMaterial.SetMPIsotropic('H1', x1,0.3,0.000007);  Object = 0;  group=xlsread('Tramos dañados.xlsx');  for iElem=group.'     ret = SapModel.FrameObj.SetMaterialOverwrite(int2str (iElem), 'H1',Object); end  ret =SapObject.SapModel.Analyze.RunAnalysis();</pre>	<p>Se define la función eigfem_mdof_torre.</p> <p>Permite pasar los datos a SAP2000 como matrices de dimensión 1.</p> <p>Permite pasar las matrices no escalares por referencia.</p> <p>Se crea el objeto SAP2000v16.</p> <p>Se abre SAP2000</p> <p>Se define la ubicación del modelo de la estructura en SAP2000.</p> <p>Se crea el objeto SapModel.</p> <p>Se desbloquea el modelo en SAP para poder ser modificado desde MATLAB.</p> <p>Se crea el modelo de análisis.</p> <p>Se definen las propiedades de los materiales a crear. Este comando se puede repetir tantas veces como materiales se deseen crear.</p> <p>Se le asigna a x1 el primer valor de la variable x_moduli y después se le declaran las propiedades al material creado anteriormente H1. En caso de haber creado otros materiales, hay que repetir este comando para cada uno de ellos</p> <p>Comandos que sobrescriben a los elementos seleccionados, las propiedades del material. En este caso, a los elementos que se encuentran dentro del archivo Excel “Tramos dañados” se le asigna la propiedad del material H1. Para asignar otra propiedad a otros elementos hay que repetir este comando.</p> <p>Se corre el análisis en SAP2000.</p>
---	--

<pre> ret = SapModel.Results.Setup.DeselectAllCasesAndComb osForOutput;  ret = SapModel.Results.Setup.SetCaseSelectedForOutpu t('MODAL');  NumberResults = 0; Obj = cellstr(' '); Elm = cellstr(' '); LoadCase = cellstr('MODAL'); StepType = cellstr('Mode'); StepNum = zeros(1,1,'double'); Period = zeros(1,1,'double'); Frequency = zeros(1,1,'double'); CircFreq = zeros(1,1,'double'); EigenValue = zeros(1,1,'double');  [ret, NumberResults, LoadCase, StepType, StepNum, Period, Frequency, CircFreq, EigenValue] = ... SapModel.Results.ModalPeriod(NumberResults, LoadCase, StepType, StepNum, Period, Frequency, CircFreq, EigenValue);  omega = CircFreq.'; [rad/s]  NumberResults = 0; Obj = cellstr(' '); Elm = cellstr(' '); ACase = cellstr('MODAL'); StepType = cellstr('Mode'); StepNum = zeros(1,1,'double'); U1 = zeros(1,1,'double'); U2 = zeros(1,1,'double'); U3 = zeros(1,1,'double'); R1 = zeros(1,1,'double'); R2 = zeros(1,1,'double'); R3 = zeros(1,1,'double'); GroupElm = 2; [ret, NumberResults, Obj, Elm, ACase, StepType, StepNum, U1, U2, U3, R1, R2, R3] = ... SapModel.Results.ModeShape('ALL', GroupElm, NumberResults, Obj, Elm, ACase, StepType, StepNum, U1, U2, U3, R1, R2, R3);  phix=U1; phiy=U2; phiz=U3;  n_m=length(omega); n_node=length(phix)/n_m; </pre>	<p>Se deseleccionan todos los casos de análisis y combinaciones.</p> <p>Se selecciona el caso modal.</p> <p>Comandos para obtener las frecuencias para cada modo de oscilación.</p> <p>Se define la variable omega como las frecuencias naturales.</p> <p>Comandos para obtener las coordenadas modales en “x”, “y” y “z”</p> <p>Se definen las coordenadas modales en “x”, “y” y “z” como phix, phiy y phiz respectivamente.</p> <p>Se calculan el número de modos. Se calculan el número de nodos de la estructura.</p>
--	---

<pre> phix_=zeros(n_node,n_m); phiy_=zeros(n_node,n_m); phiz_=zeros(n_node,n_m);  for ind1=1:n_m     phix(:,ind1)= phix(:,1:n_node);     phix(:,1:n_node)=[];     phiy(:,ind1)= phiy(:,1:n_node);     phiy(:,1:n_node)=[];     phiz(:,ind1)= phiz(:,1:n_node);     phiz(:,1:n_node)=[]; end  phi=[phix_;phiy_;phiz_]; nodes=zeros(n_node,1); for ind1=1:n_node     nodes(ind1)=str2double(Obj{ind1}); end  DOF=[nodes+0.01;... % 0.01 = x      nodes+0.02;... % 0.02 = y      nodes+0.03]; % 0.03 = z  ret = SapObject.ApplicationExit(false());     SapModel = 0;     SapObject = 0; </pre>	<p>Se definen matrices intermedias para obtener las coordenadas modales definitivas.</p> <p>Se calculan las coordenadas modales definitivas.</p> <p>Se crea una matriz phi que contiene las coordenadas modales en las tres direcciones principales.</p> <p>Se definen los grados de libertad de la estructura en las tres direcciones.</p> <p>Se cierra la aplicación SAP2000.</p>
--	---

Todos estos parámetros extraídos desde SAP2000 son los que permiten realizar la calibración debido a que en cada iteración este script se ejecuta varias veces hasta que los parámetros que se extraigan sean muy parecidos a los de la estructura real y la función objetivo de residuos muy pequeños.

#### 2.4.6 Script (sens\_anal\_torre.m)

Este script realiza el análisis de sensibilidad, con la obtención de la matriz MAC. Esta matriz tiene varios usos como que a partir de la cual se puede tener una valoración de cuanto la ubicación de los sensores escogida es correcta cuando en su diagonal principal se obtienen valores de casi 1 y fuera de esta menores o iguales que 0.5. También se puede utilizar para cuando se cuenten con valores reales de mediciones, poder numerar los modos de la estructura real al ser comparados con los modos del modelo, si dos modos presentan un valor cercano a 1 en la matriz MAC, se puede concluir que son modos semejantes.

La matriz MAC que se presenta cuenta de colores y una escala para esos colores desde 0 hasta 1. Cuando se analizan los resultados finales, para obtener una buena calibración, se deberían tener valores de uno o cercanos a uno en la diagonal principal de esta matriz, mientras que si nos acercamos a los extremos se deben ir disminuyendo estos valores hasta llegar a valores cercanos a cero.



En la figura 2.5 se presenta un ejemplo de esta matriz. También, al analizarla, se aprecia que se tuvo en cuenta ocho modos de la estructura y se obtuvo un gran grado de calibración puesto que presenta valores de uno en la diagonal principal, otro detalle es que cuando se comparan los grados de semejanzas entre otros modos, se aprecia que hay modos cercanos que presentan grados de calibración muy bajo, por ejemplo, el modo uno y dos presentan un valor MAC de cero lo cual no es muy bueno, pues al ser dos modos consecutivos deberían presentar grados de calibración más elevados.

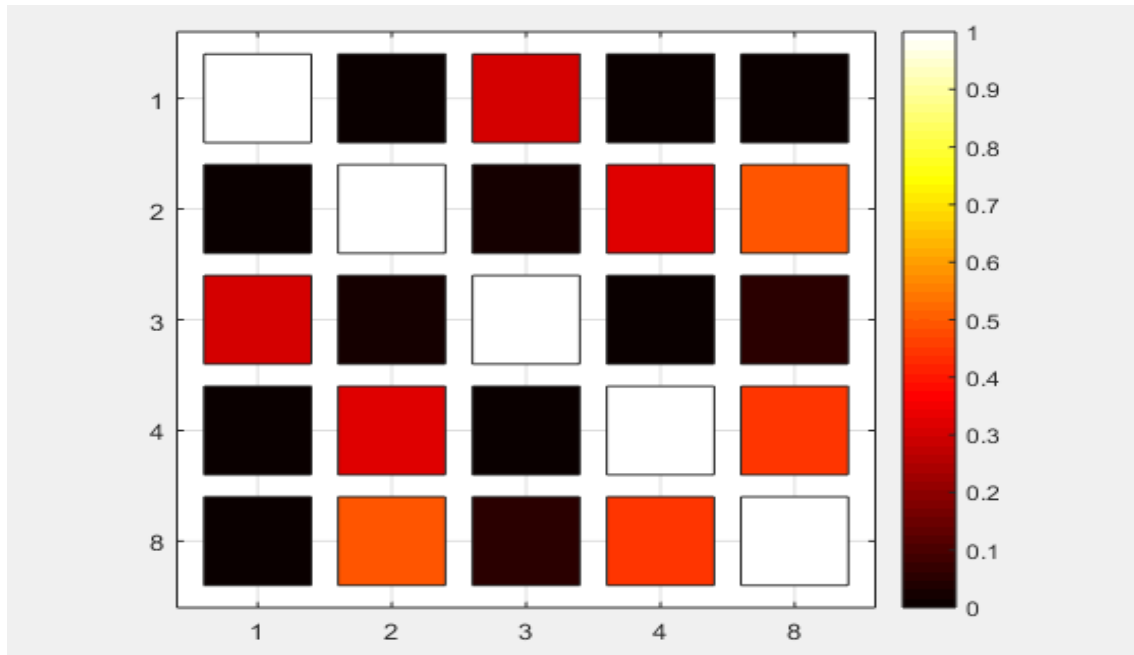


Fig. 2.5 Ejemplo de matriz MAC obtenida.

El script contiene fundamentalmente:

- Se indica la propiedad del modelo de referencia.  
`E_ref=0.96*2.1e8;`
- Se indica la propiedad de la estructura real.  
`E_0=2.1e8;`
- Se seleccionan los modos para calcular los parámetros modales.  
`modeSel=1:5;`
- Se escogen las posiciones de los sensores.  
`DOF_sensors=[[62,97,122,128,203,228,245,263]+0.01,[63,98,123,129]+0.02].';`
- Se seleccionan los modos a tener en cuenta en el cálculo de la matriz MAC.  
`modSel=[1:4,8];`  
`autoMAC=mac(L_0*phi_ref(:,modSel),L_0*phi_ref(:,modSel))`

- Se grafican los resultados y se configura el gráfico (colores, x eje x, y, etiquetas de los ejes, nombre general del gráfico).

## 2.5 Conclusiones parciales del capítulo

- Existen diferentes maneras de obtener el modelo de una torre en Sap2000, como son: modelarlas manualmente, utilizar los modelos que tiene incorporado Sap2000 por defecto o modelarlas en AutoCAD; este último siendo el preferible debido a las grandes posibilidades de dibujo que nos brinda AutoCAD y la reducción de tiempo y trabajo que esto implica.
- Existen una gran cantidad algoritmos y métodos de optimización, la elección del más adecuado depende del problema al que se enfrente el ingeniero. En este caso, el algoritmo escogido es el que tiene MATLAB incorporado con la función *lsqnonlin*, el cual está basado en los métodos de Gauss-Newton y Levenberg-Marquardt, los cuales, para problemas de calibración de modelos de elementos finitos, logran la convergencia en un buen tiempo y exactitud de los resultados.
- En todo problema de optimización es importante escoger de manera correcta las variables de diseño, las variables de estado y las restricciones, puesto que dependen de ellas la correcta estimación de los parámetros calibrados del modelo. Para esta investigación, el Módulo de Deformación del Acero o Módulo de Young, se seleccionó como variable de diseño, las propiedades dinámicas de la estructura, como las frecuencias propias de oscilación y las formas modales como variables de estado.
- Los scripts de MATLAB confeccionados brindan una plataforma interactiva para ser usado por los ingenieros en la calibración de torres autosoportadas u otro tipo de estructura y la incorporación de la matriz MAC en este proceso permite contar con una valoración de cuán exacto fue el proceso y de cuán calibrado resultó el modelo en Sap2000.

## **CAPITULO III: Aplicación del procedimiento automatizado para la calibración a un caso de estudio de una torre autosoportada en Cuba.**

En este capítulo se aplica el algoritmo de calibración antes explicado a un caso de torre autosoportadas en Cuba, se explica de manera general la torre seleccionada, la elaboración de su modelo, así como las particularidades en los scripts de Matlab que se tuvieron en cuenta a la hora de prepararlos para la calibración. Es muy importante tener conocimiento de la interface OAPI SAP2000-MATLAB, pues a través de la misma se hace posible la aplicación de estas potentes técnicas de optimización y se puede seleccionar varios parámetros a calibrar como el módulo de deformación del material, sea acero o hormigón, las inercias de las secciones, las rigideces de los apoyos, entre otros.

### **3.1 Introducción**

La calibración de torres de telecomunicaciones, como se ha explicado previamente, es de gran importancia, puesto que son estructuras muy sensibles a las cargas dinámicas como huracanes y terremotos, y al calibrarlas se puede contar con modelos más precisos de estas que permitan realizar valoraciones o la toma de decisiones en las mismas con respecto a ampliaciones de antenas, efectos de huracanes, terremotos, daños que presente la torre, entre otros o también contar con estos modelos calibrados que sirvan como bases para otras investigaciones.

El presente capítulo de la investigación se encarga, a partir de un modelo de torre real, de explicar el procedimiento aplicado al modelo de torre objeto de estudio y analizar los resultados para evaluar si es efectivo o no la aplicación del mismo a casos de torres de telecomunicaciones autosoportadas.

El modelo de torre objeto de estudio seleccionado es Najasa, de fabricación nacional, este se encuentra distribuido en varias localidades del país. Esta torre autosoportada tiene forma troncopiramidal desde la base. Su sección transversal es cuadrada de 6x6m hasta una altura de 45m. A partir de este punto se levanta la torreta de sección constante con una altura de 15m y una sección de 1.17x1.17m. La altura total de la estructura de 60m y está compuesta por columnas, diagonales principales, tranques secundarios, diagonales secundarias (estas últimas junto con los tranques en diagonales conforman el sistema de arriostramiento de las diagonales principales), tranques y tranques interiores. Para una mejor comprensión en la figura 3.1 se muestra

los elementos componentes de la torre, los perfiles son todos L de varias dimensiones, la resistencia del acero es  $f_y=250\text{MPa}$  y el módulo de deformación del material  $E = 2,1 \times 10^5 \text{ MPa}$ .

En la figura 3.2 se muestra la disposición espacial y los tipos de perfiles metálicos que presenta la torre para cada elemento en los distintos niveles.

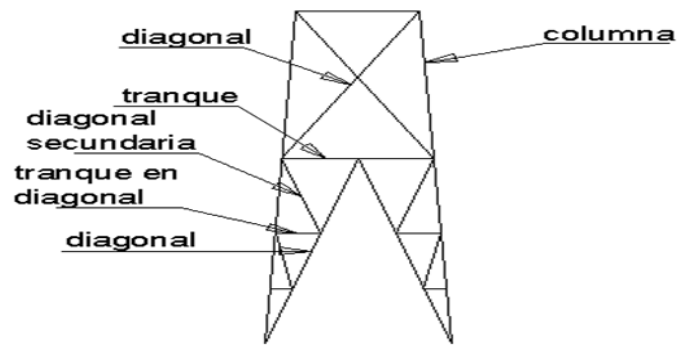


Fig. 3.1 Elementos componentes de la torre.

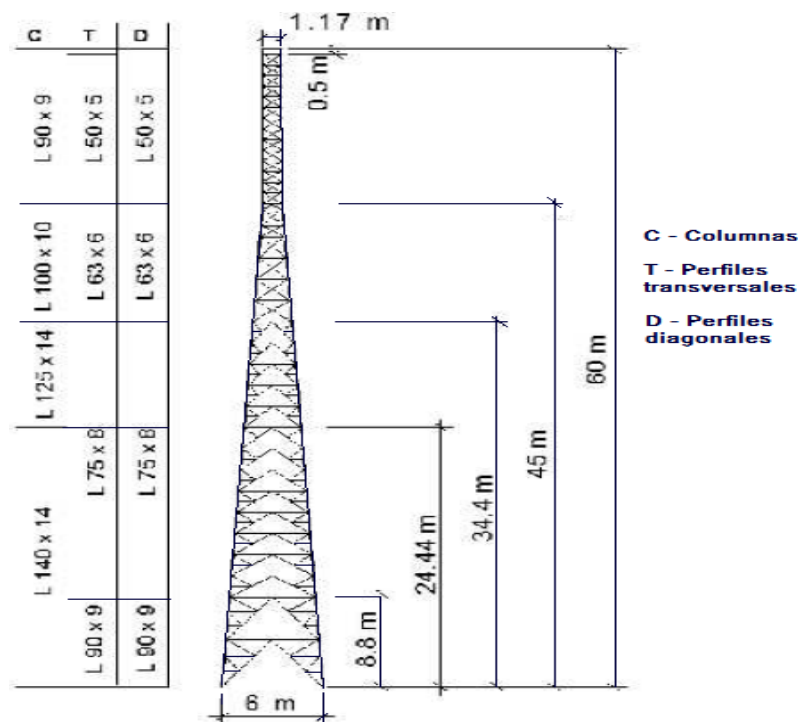
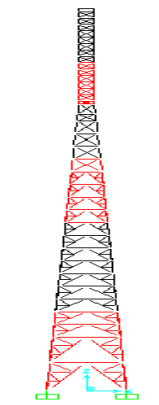
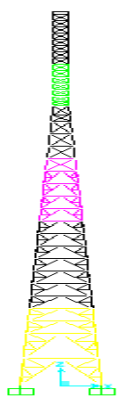
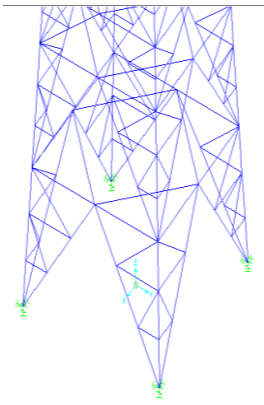
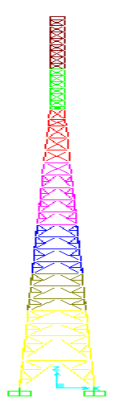


Fig. 3.2 Tipos de perfiles por tramos en la torre autosoportada objeto de estudio.

De manera general la calibración se les realiza a 4 casos de estudio de la torre autosoportada. Para el primero o **Caso 1** se programaron los scripts de MATLAB para calibrar el módulo de deformación del material (E) en 3 tramos de la torre, simulando que estos están dañados en la misma medida, o sea que se obtendrá un único módulo para las tres secciones de la torre. En el segundo **Caso 2** las variables a calibrar es el módulo de

deformación en los mismos tramos, pero suponiendo que cada tramo tiene un daño distinto, o sea se obtendrán 3 valores del módulo, uno para cada sección. Para el tercero o **Caso 3** las propiedades a calibrar son las rigideces al desplazamiento de los apoyos  $k_z$ ,  $k_y$  y  $k_x$ , donde se reajustaron los scripts de MATLAB para mantener el módulo de deformación del material constante e igual a  $2,1 \times 10^5$  MPa en todos los elementos y sean  $k_z$ ,  $k_y$  y  $k_x$  las iteradas hasta encontrar una solución, por lo que se obtendrán 2 valores de rigideces, uno para la rigidez en “z” y otro para las rigideces en “x” y “y”, estas últimas tienen el mismo valor. Para el último caso o **Caso 4**, el cual es el más complejo, se calibrarán 7 módulos de deformación del material correspondientes a los elementos de 7 tramos distintos en toda la torre. Todo lo expuesto anteriormente queda resumido en la tabla 3.1.

Caso 1	Caso 2	Caso 3	Caso 4
 <p>La figura muestra de color rojo los tramos que serán calibrados para la primera variante. Los elementos de estos tramos se encuentran en un archivo Excel nombrado “Tramos dañados”, el cual es el que lee MATLAB para asignarles el módulo de deformación <math>E1=x1</math> igual para todos.</p>	 <p>La figura muestra de 3 colores diferentes los 3 tramos que serán calibrados en la segunda variante. Los elementos de estos tramos se encuentran en 3 archivos Excel distintos nombrados (en la figura de abajo hacia arriba); “Tramo inferior”, “Tramo intermedio” y “Tramo superior”, los cuales son los que lee MATLAB y le asigna 3</p>	 <p>La figura muestra los springs que le asigna MATLAB a las tres direcciones principales “x”, “y”, y “z” a los nodos 3, 163, 266 y 4 en cada corrida del modelo. Se le asignan <math>k_z=x1</math> para los cuatro nodos y <math>k_x=k_y=x2</math> para los cuatro nodos también por lo que serían 2 variables a calibrar.</p>	 <p>La figura muestra los 7 tramos que serán calibrados cada uno con un color diferente. Los elementos que componen estos tramos se encuentran en 7 archivos Excel nombrados (en la figura de abajo hacia arriba) GROUP1, GROUP2, GROUP3, GROUP4, GROUP5, GROUP6 y GROUP7. Para cada uno de estos tramos se</p>

	módulos de deformación diferentes para cada uno $E1=x1$ , $E2=x2$ , $E3=x3$ .		obtendrá un módulo de deformación calibrado $E1=x1$ , $E2=x2$ , $E3=x3$ , $E4=x4$ , $E5=x5$ , $E6=x6$ y $E7=x7$ .
--	---	--	---

Tabla 3.1 Resumen de los tramos a calibrar para cada uno de los casos de estudios implementados.

Es importante también aclarar que no se cuentan con mediciones reales de la estructura de la torre por lo que los parámetros modales experimentales tendrán que ser supuestos, el procedimiento para la obtención de los parámetros modales que se supondrán medidos experimentalmente se explica detalladamente en el epígrafe 3.5.

### 3.2 Modelación de la torre objeto de estudio

La modelación de la torre se realizó en AutoCAD 2015 debido a la complejidad de la misma y la cantidad de elementos con que cuenta, en total 1076 perfiles, los cuales están representados en el modelo por elementos tipo frame. La sección transversal de la torre es cuadrada en toda su altitud como se puede apreciar en la figura 3.2.

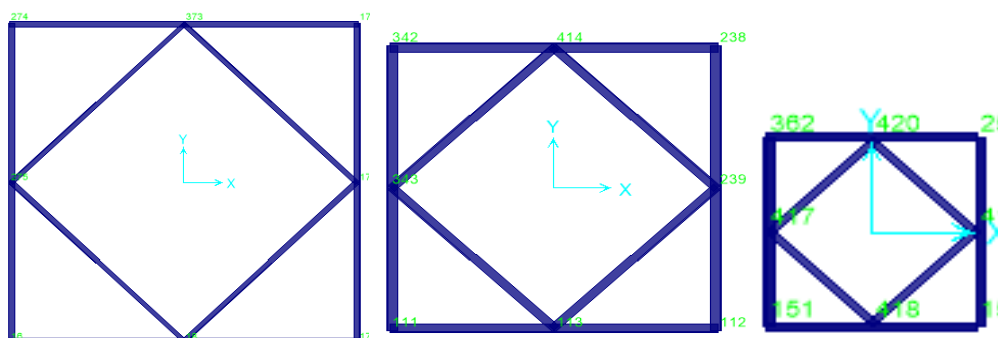
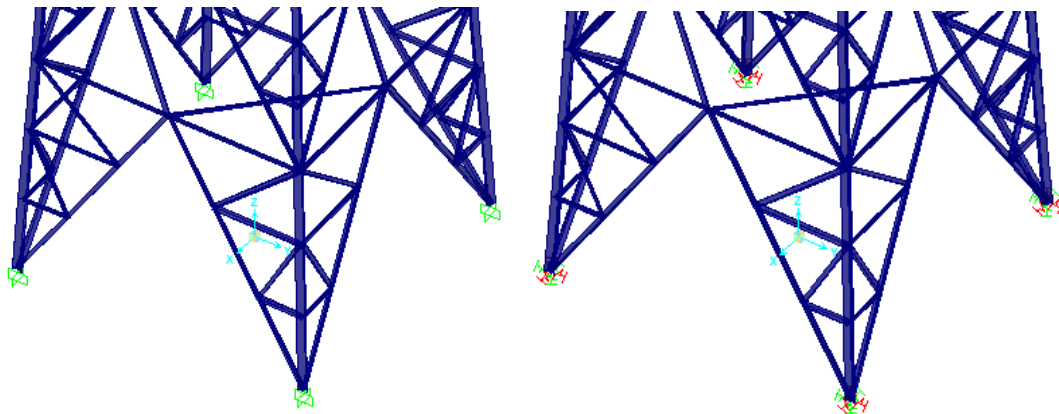


Fig. 3.3 Secciones transversales a diferentes niveles de la torre autosoportada objeto de estudio.

La modelación, de manera general, se recogen en la siguiente serie de pasos:

1. En AutoCAD 2015 todas las líneas se dibujaron en una capa distinta de la capa cero, la cual fue nombrada, en este caso, "elementos frame".
2. Luego se exportó este archivo .dxf a SAP2000v16 mediante el procedimiento explicado en el epígrafe 2.2.1, para el caso de la tercera variante.
3. Se definió el material para todos los elementos, en este caso acero A36, a partir del comando Define/ Materials. Este paso es importante porque si en MATLAB se definen los módulos del material para los elementos que se van a calibrar

- solamente, entonces los demás no tendrían un valor de  $E$  y entonces las características dinámicas del modelo no serían una representación de la torre real.
4. Se definieron en SAP2000v16 las secciones con que cuenta la torre objeto de estudio, distribuidos a distintos niveles, a partir de los comandos Define/ Section Properties/ Frame Sections y se les asignaron a esas secciones el acero A36 creado anteriormente.
  5. Luego se asignaron los perfiles a partir del comando assign/ Frame sections, seleccionando todos los elementos frame de perfiles iguales y presionando el botón ok. Esto se repite para los otros elementos frame hasta haber asignado las secciones a todos.
  6. Se definieron las condiciones de apoyo como articulaciones para el caso de la calibración del módulo de deformación del material. En el caso de que las variables a calibrar sean las rigideces de los apoyos hay que modelarlas como springs o muelles en SAP2000 o dejar estos nudos sin apoyos para luego en MATLAB programar a partir de la OAPI la asignación de springs a estos nudos con valores en la dirección de  $x$ ,  $y$ ,  $z$ . Esto se puede apreciar en la figura 3.4.



*Fig. 3.4 Casos de apoyos en la torre autosoportada objeto de estudio. (A la izquierda modelos para calibrarle el módulo de deformación del material, a la derecha modelo a calibrarle las rigideces de los apoyos).*

7. Se definieron grupos en toda la estructura para los casos 1,2 y 4 con el objetivo de exportarlos a un documento Excel que contenga los labels de los elementos frames correspondientes a cada tramo que se supone dañado en toda la estructura.  
Para el caso de estudio donde se calibran los apoyos, no se definieron grupos, todos los elementos tienen el mismo material, acero A36.
8. Al final se guardó este archivo (.sdb) con el nombre de -Najasa Modal continua- en los casos de calibrar el módulo de deformación, y con el nombre de -Najasa springs-

para el caso de calibrar las rigideces de los apoyos, dentro de 4 carpetas diferentes dentro de la carpeta donde se encuentran los scripts de MATLAB.

En la figura 3.5 se muestra el contenido de la carpeta general para la calibración.

Nombre	Fecha de modificación	Tipo	Tamaño
Scripts 3 E a calibrar	20/05/2017 20:48	Carpeta de archivos	
Scripts 7 E a calibrar	20/05/2017 20:20	Carpeta de archivos	
Scripts E a calibrar	12/05/2017 19:12	Carpeta de archivos	
Scripts Springs a calibrar	19/05/2017 20:57	Carpeta de archivos	
toolbox	30/03/2017 17:28	Carpeta de archivos	
Tower model 3 E calibrated	20/05/2017 20:49	Carpeta de archivos	
Tower model 7 E calibrated	20/05/2017 20:12	Carpeta de archivos	
Tower model E calibrated	12/05/2017 19:49	Carpeta de archivos	
Tower model springs calibrated	20/05/2017 12:10	Carpeta de archivos	
asmkm_mdof.m	10/01/2017 9:07	Archivo M	1 KB
bar3color.m	13/01/2017 13:07	Archivo M	4 KB
mac.p	04/11/2016 9:30	Archivo P	1 KB
selectdof.m	29/03/2017 10:21	Archivo M	3 KB

Fig. 3.5 Carpeta general de la calibración.

### 3.3 Localización de los sensores en la torre

La localización de los sensores para la medición de vibraciones en la torre de telecomunicaciones es muy importante, debido a que una correcta localización de los mismos permite obtener con mayor precisión las características dinámicas de la torre como las frecuencias y las formas modales. Las mediciones de los sensores se recogen en función del tiempo para luego a partir de transformadas de Fourier llevarlas al dominio de las frecuencias, este proceso se denomina Procesamiento de Señales (Signal Processing), el cual si se contaran con mediciones reales de la torre habría aportado las formas y frecuencias modales que servirían como variables de estado en el proceso de optimización.

Al no contar con esta información, fue necesario encontrar una correcta ubicación de los sensores lo cual se realizó a partir del script (**sens\_anal\_torre.m**). Vale aclarar que los sensores corresponden a puntos (joints) en el modelo en SAP2000v16 de los cuales se extraen las coordenadas modales en cada iteración hasta que se calibre la torre. También el objetivo es tratar de colocar la mayor cantidad de sensores posibles debido a que sería más información de entrada para la optimización lo que daría mayor precisión en los resultados. Este procedimiento realizado se describe en los siguientes pasos:

1. Se definen los niveles donde se colocan los sensores, esto corresponde a distintas alturas en el modelo en SAP2000.



2. Luego se decide si se colocan 2, 3 o 4 sensores por nivel en dependencia de la precisión que se desea. Se deberían colocar al menos 1 sensor en cada dirección (x y) por nivel.
3. Después se programa el script de MATLAB (**sens\_anal\_torre.m**) con los sensores en las posiciones escogidas.
4. Por último, se corre este script y se analiza la matriz MAC que este presenta. Si esta matriz presenta valores en la diagonal principal muy cerca de 1 y los restante valores todos son menores que 0.5, entonces se puede decir que las posiciones de los sensores escogidas son correctas o brindan suficiente información de la estructura para realizar la calibración.

Para el caso de esta investigación se realizaron estos pasos y los resultados que se obtuvieron fueron los siguientes:

Como se puede apreciar en la figura 3.6 se decidió colocar sensores en 8 niveles distintos en la torre, a 10m, 20m, 30m, 40m, 45m, 50m 55m y 60m.

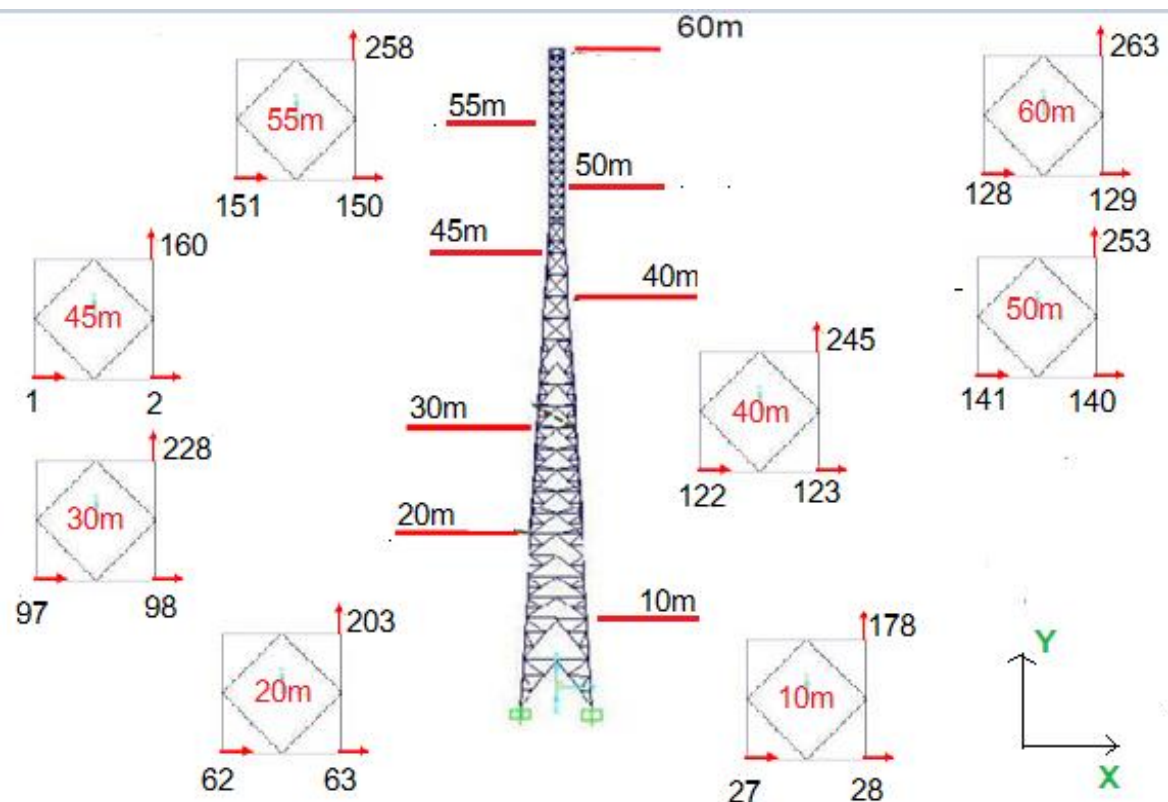


Fig. 3.6 Arreglo de los sensores en la torre objeto de estudio. (Las flechas hacia la derecha representan la dirección de "x" y hacia arriba la dirección de "y")

Luego se decidió también colocar 3 sensores por niveles: 2 en la dirección de “x” y uno en la dirección de “y” como se puede apreciar en la figura 3.6 donde en resumen se tienen 24 sensores en toda la estructura, 16 en la dirección de x y 8 en la dirección de y.

Como último paso se programó el script (**sens\_anal\_torre.m**) con la ubicación de los sensores propuesta y analizando los 5 primeros modos, puesto que con más modos la información que se obtendría sería muy grande e irrelevante, y se obtuvo la matriz MAC que se presenta en la figura 3.7.

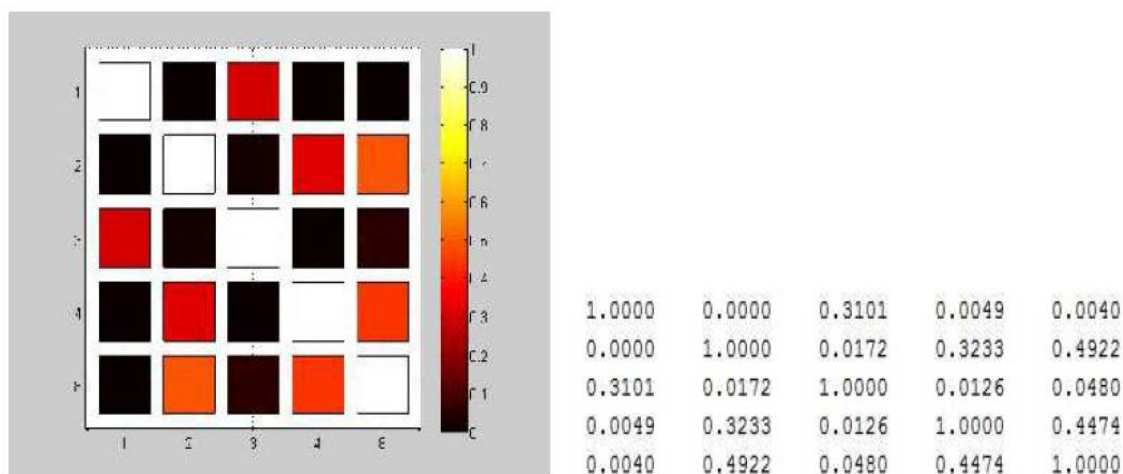


Fig. 3.7 Matriz MAC obtenida y valores de la misma.

Al analizar estos valores de la matriz **se acepta** la posición de los sensores propuesta debido a que la diagonal principal tiene valores de 1 y los restantes valores son menores que 0,5 con lo que concluye este procedimiento.

### 3.4 Declaración de los parámetros que intervienen en la optimización

Después de realizar un análisis minucioso se seleccionaron los componentes del problema de optimización para la calibración del modelo estructural de la torre analizada, quedando de la siguiente forma:

#### Variables de diseño:

Son variables independientes que son iteradas en orden para alcanzar los valores óptimos, la siguiente tabla resume como quedan definidas estas variables en los 4 casos de calibración.

Caso 1	Caso 2	Caso 3	Caso 4
X1 = E1	X1 = E1 X2 = E2	X1 = kz X2 = kx = ky	X1 = E1 X2 = E2

La única variable que se calibra es el módulo de deformación de los tres tramos de la torre con un valores único e igual para los tres tramos.	$X3 = E3$  Se calibran 3 valores de módulos de deformación, cada valor corresponde a cada uno de los tramos de la torre.	Se calibran dos variables correspondientes a las rigideces de los apoyos. Las rigideces en "x" y "y" son iguales por lo que se les asignarán un solo valor.	$X3 = E3$ $X4 = E4$ $X5 = E5$ $X6 = E6$ $X7 = E7$  Se calibran 7 valores de módulos de deformación, cada valor corresponde a cada uno de los tramos de la torre.
--	--	---	--

#### Variables de Estado:

Se definen como variables de estado las frecuencias naturales de los 5 primeros modos de oscilación flexionales simétricos de la torre, además las coordenadas medidas en los puntos que corresponden a los sensores ubicados, también para los 5 primeros modos. Estos parámetros son dependientes pues dependen de los valores que toma cada variable de diseño para cada una de las iteraciones.

#### Función Objetivo:

La función objetivo es la función del vector residual entre los resultados obtenidos en las mediciones y los obtenidos en el modelo. Esta se divide en 2 residuos explicados en los capítulos anteriores. En la determinación de la forma modal solo las coordenadas modales en (x y) en los puntos especificados anteriormente por los sensores fueron utilizados. Por otra parte, los factores de peso o ponderación ( $\alpha$  y  $\beta$ ) se eligieron según la magnitud de cada tipo de residuos con el objetivo de que todos tuviesen la misma influencia en el vector residual total a la hora de realizarse la optimización.

#### Restricciones:

Se establecen como restricciones los límites inferior y superior que deben cumplir las variables de diseño. Esto queda resumido en la siguiente tabla.

<b>Caso 1</b>	<b>Caso 2</b>	<b>Caso 3</b>	<b>Caso 4</b>
$Lb = Emin = 199 \text{ GPa}$  $Ub = Emax = 220 \text{ GPa}$	$Lb = Emin = -inf$  $Ub = Emax = +inf$	$Lb = Emin = -inf$  $Ub = Emax = +inf$	$Lb = Emin = -inf$  $Ub = Emax = +inf$

Se definieron los límites inferiores y superiores que pueden tomar las variables de diseño para que el algoritmo de optimización no se hiciera muy extenso o que no pudiera encontrar la convergencia. Cabe destacar que el algoritmo escogido para esta variante en MATLAB fue el <b>Trust-Region-Reflective Least Squares Algorithm</b> el cual permite la incorporación de restricciones.	Para este caso de calibración el algoritmo utilizado en MATLAB fue el <b>Levenberg-Marquardt Method</b> el cual no permite la incorporación de restricciones.	Para este caso de calibración el algoritmo utilizado en MATLAB fue el <b>Levenberg-Marquardt Method</b> el cual no permite la incorporación de restricciones.	Para este caso de calibración el algoritmo utilizado en MATLAB fue el <b>Levenberg-Marquardt Method</b> el cual no permite la incorporación de restricciones.
--	---	---	---

Es importante aclarar que estos dos límites, en el caso de proveerlos, deben comprender en su intervalo al valor real al que la calibración debe arribar, sino el algoritmo tardará mucho o los valores que encontrará serán irreales por completo. También si se invierten los límites, o sea  $lb > ub$ , MATLAB los reajustaría automáticamente a  $-\inf$  y  $+\inf$ .

### 3.5 Obtención de los parámetros dinámicos experimentales

Como ya se explicó anteriormente, no se dispone de mediciones experimentales para la identificación de los parámetros modales necesarios para realizar la calibración apropiada del modelo, sujeta a la función objetivo que se aplica al tipo de problema de optimización correspondiente para estos casos.

Debido a este inconveniente se escogió la estrategia de crear dos modelos para la calibración en cada uno de los casos de estudio, excepto en el caso de la calibración de las rigideces de los apoyos, uno que sería el modelo a calibrar y el otro sería el modelo dañado del cual se extraen los parámetros modales que sirven como mediciones experimentales. Con el modelo dañado la estrategia fue disminuirle las áreas, las inercias en “x” y “y” y el módulo de deformación del material para los casos 1, 2 y 4 a todos los

elementos que estén dentro de los tramos a calibrar en cada caso correspondiente. Esto se hizo posible a partir de en SAP2000 asignarle modificadores de las propiedades a las secciones de estos elementos. Cuando en SAP2000 se asignan modificadores a las inercias y las áreas, el programa multiplica los modificadores por las inercias y áreas, es decir que, si se le asignan modificadores menores de 1, entonces se le estaría reduciendo estas propiedades a los elementos, lo que puede nombrarse “provocar un daño”. Los valores de estas reducciones fueron 0,8 en todos los casos para las áreas y las inercias en las dos direcciones. Para reducir el módulo de deformación del material se programó a partir de MATLAB en el script (data\_mdof\_torre).

Para el caso del modelo a calibrar no se aplicaron modificadores a ninguno de sus elementos por lo que el objetivo final para estos casos de estudio es encontrar una combinación de las variables de diseño (módulo de deformación del material) que produzcan características dinámicas (frecuencias naturales de oscilación y formas modales) en el modelo a calibrar que igualen o sean muy parecidas a las características dinámicas obtenidas del modelo dañado.

Cuando se calibrar las rigideces de los apoyos, la estrategia definida fue reducirle las rigideces en “x” y “y” produciendo así un daño, para luego iterar las mismas hasta encontrar un valor de rigideces que iguale las características dinámicas del modelo de la estructura con las dañadas. Para este caso el resultado a esperar serían las mismas rigideces dañadas que se introdujeron de referencia.

### 3.6 Métodos de optimización empleados.

La optimización se realiza de manera automática con uso de la función de optimización lsqnonlin que tiene MATLAB incorporada, lo que se explicó detalladamente en el Capítulo 2, por lo que el objetivo de este epígrafe es brindar la información sobre cómo se desarrolló este proceso.

Para todos los casos de calibración el algoritmo de optimización empleado fue el **Levenberg-Marquardt Method**, debido a que con el otro algoritmo la calibración no encontraba el mínimo correctamente o llegaba un punto donde los residuos se mantenían constantes y la optimización se detenía debido a que la distancia del paso era muy pequeña, por debajo de la tolerancia permitida. Esto queda programado en MATLAB de siguiente manera:

```
% Specify options
options=optimset('Algorithm','levenberg-marquardt','ScaleProblem','Jacobian','Display','iter-detailed',...
    'Jacobian','off','TolFun',TolFun,'TolX',TolX,'OutputFcn',@outfun);
```

```
load data_mdof_ern nparameters

% Nonlinear optimization
[T_upd, resnorm, residual, exitflag, output]=...
lsqnonlin(@objfun,T_0,-
inf*ones(1,nparameters),inf*ones(1,nparameters),options);
```

Se puede notar que el Jacobiano se define en off y también se definen límites a las variables de diseño de -infinito y +infinito, debido a que este algoritmo de optimización no permite que se le definan restricciones al problema.

### **3.7 Obtención y análisis de los resultados. Validación de los modelos calibrados obtenidos.**

Después de tener todos los scripts programados correctamente y los nodos donde se colocan los sensores definidos se procedió a ejecutar los 4 casos de calibración. Para validar los modelos calibrados en cada caso la estrategia escogida fue aplicar cargas de 400kN en algunos nodos y obtener los desplazamientos en una columna de la torre para los modelos inicial, calibrado y de referencia o dañado. A continuación, se exponen los resultados para cada una de las variantes y al final del epígrafe el análisis de los mismos.

En las figuras 3.8, 3.11, 3.14 y 3.17 se presentan los valores de la función objetivo, o sea, los residuos para cada iteración a manera de una curva para que sea mucho más fácil su interpretación, así mismo, se presenta una tabla para cada una de estas figuras en la que quedan definidos los valores de manera exacta.

En las tablas 3.2, 3.4, 3.6 y 3.8 se hace un resumen para cada variante de las frecuencias propias de oscilación para cada modo debido a que estas son variables de estado e influyen en que se un resultado positivo. Como la calibración se efectuó solo teniendo en cuenta los 5 primeros modos, estos son los que se analizan en este caso, estas frecuencias se muestran para estos modos, pero para tres casos distintos, el caso del modelo inicial, o sea los valores de las frecuencias calculados para la primera iteración la cual tiene los valores de las variables de diseño iniciales, el caso de los valores experimentales y el caso del modelo calibrado, o sea, las frecuencias para cada modo calculados a partir de asignarle al modelo los valores de las variables de diseño que se obtuvieron al terminar la calibración. Otro aspecto que se muestra es el porcentaje de error en las frecuencias para el modelo inicial y el calibrado y los valores MAC en cada uno de los modos para el modelo inicial y el calibrado.

En las figuras 3.9, 3.12, 3.15 y 3,18 se muestran, al igual que las tablas anteriores, los valores de las frecuencias propias de oscilación para cada modo, pero a manera de gráfico para que sea mucho más fácil de interpretar.

En las figuras 3.10, 3.13, 3.17 y 3,20 se grafican los valores de las coordenadas modales para los cinco primeros modos las cuales son variables de estado en la calibración por lo que, si se obtienen coordenadas casi iguales para los valores de referencia y los valores del modelo calibrado, entonces la calibración sería correcta. Esto se realiza para una de las cuatro columnas del modelo de la torre que comprende los nodos [4 5 16 27 34 41 48 55 62 69 76 83 90 97 104 111 118 120 122 124 126 1 131 133 135 137 139 141 143 145 147 149 151 153 155 157 159 128] con alturas [0 4.49 8.48 10.47 12.46 14.46 16.45 18.4469 20.44 22.44 24.43 26.42 28.42 30.41 32.41 34.4 36.4 38.39 40.38 42.38 43.38 45 45.5 46.5 47.5 48.5 49.5 50.5 51.5 52.5 53.5 54.5 55.5 56.5 57.5 58.5 59.5 60] respectivamente. Todos los resultados están dados para la dirección de “x” y la dirección de “y”.

Al final de los resultados de cada variante se presentan las tablas 3.3, 3.5, 3.7 y 3.9 en las cuales se muestran los valores de las variables de diseño que se supusieron, de acuerdo con lo explicado en el epígrafe 3.5, y las que se obtuvieron al final de la calibración.

### 3.7.1 Resultados (Caso de estudio 1)

Para el caso del caso 1 donde se calibra 1 módulo de deformación para tres tramos en la torre objeto de estudio la calibración terminó en 1h y 40min con un total de 24 iteraciones debido a que la distancia del paso fue menor que la tolerancia permitida.

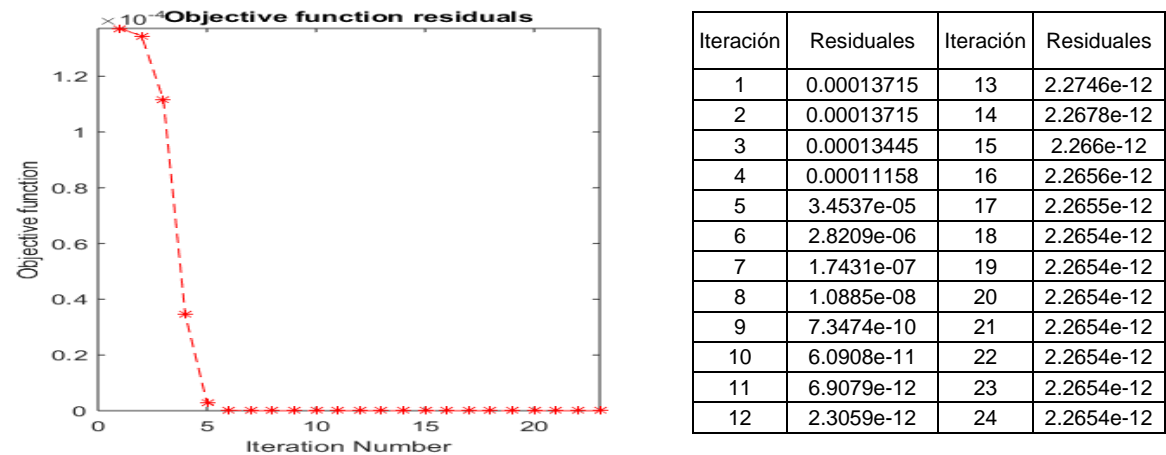


Fig. 3.8 Residuos de la función objetivo para cada iteración

Modos	Frecuencias propias de oscilación			Error en las frecuencias (%)		Valores MAC (%)	
	Valor Experimental	Modelo Inicial	Modelo Calibrado	Modelo Inicial	Modelo Calibrado	Modelo Inicial	Modelo Calibrado
1	1,3001	1,3849	1,301	6,52	0,07	99,99522	99,99993
2	1,3027	1,3875	1,3035	6,51	0,06	99,99639	99,99998
3	3,6044	3,872	3,6072	7,42	0,08	99,97641	99,99987
4	3,6077	3,8754	3,6104	7,42	0,07	99,97659	99,99996
5	4,8058	5,3603	4,8013	11,54	0,09	99,72779	99,99941

Tabla 3.2 Frecuencias propias de oscilación, errores en las mismas y valores de la matriz MAC para los valores experimentales y los modelos inicial y calibrado

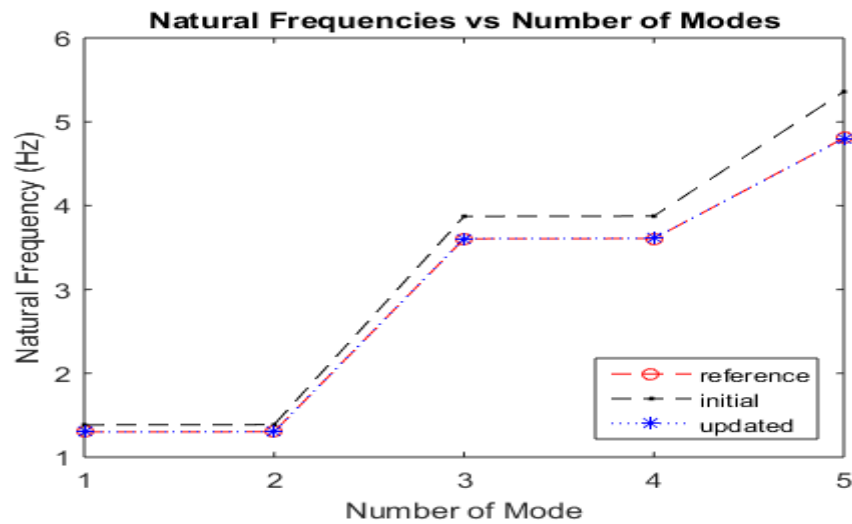
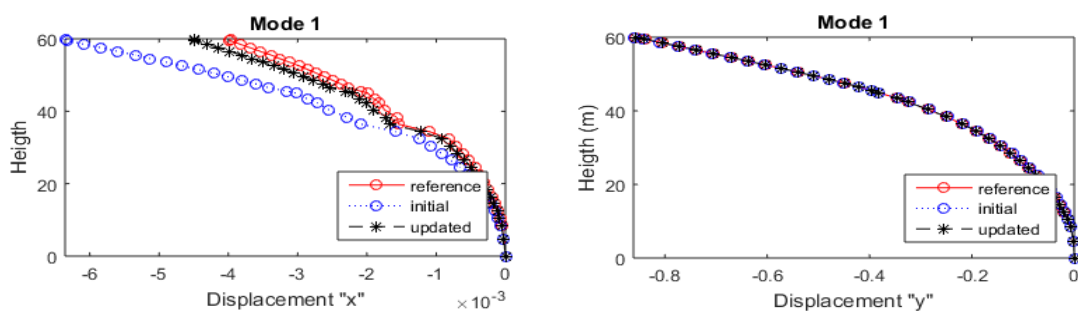


Fig. 3.9 Frecuencias propias de oscilación para los 5 primeros modos correspondientes a los valores experimentales y a los modelos inicial y calibrado.

La figura 3.10 muestra los valores de los desplazamientos modales para las dos direcciones en que se desplaza la torre "x" y "y" para los valores de referencia, del modelo inicial y del modelo calibrado. Estos valores corresponden a los nodos de la columna de la torre definida anteriormente.





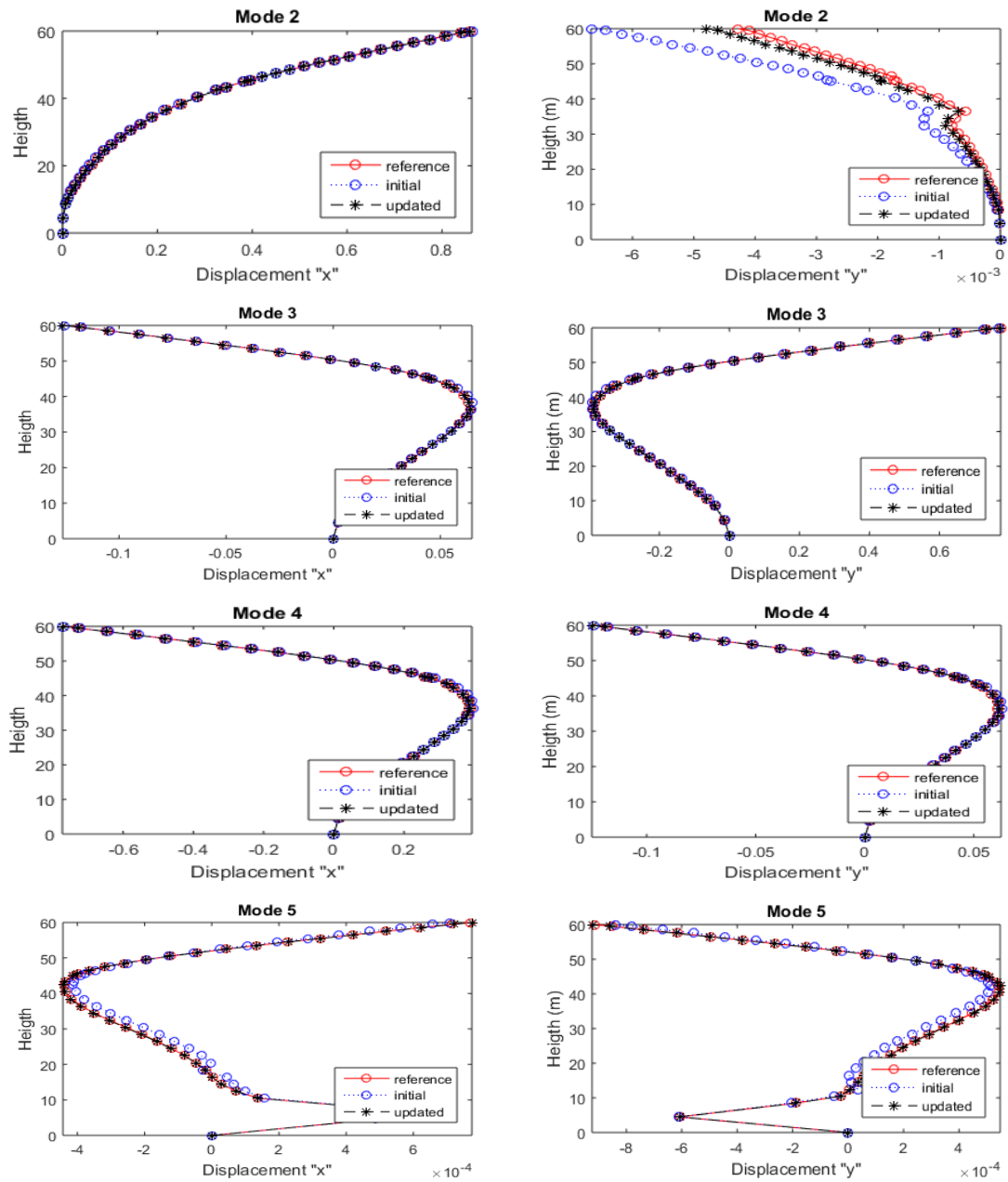


Fig. 3.10 Valores de las coordenadas modales para cada modo de los valores experimentales y los modelos inicial y calibrado. (A la izquierda coordenadas en "x" y a la derecha coordenadas en "y")

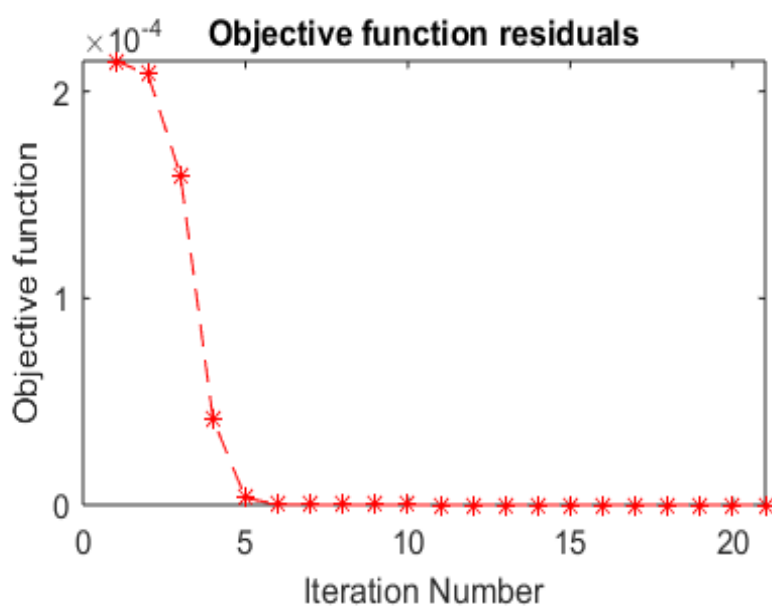
También, como los valores experimentales o de referencia son calculados a partir de lo explicado en el epígrafe 3.5, se presentan en la tabla 3.4 los valores del módulo de deformación que se supuso que la torre real tuviera y los calculados al terminar la calibración.

Modelo con inercias y áreas reducidas en 0,8	Modelo con inercias y áreas no reducidas	
Valores de E de referencia (kPa)	Valores de E Iniciales (kPa)	Valores de E calculados (kPa)
210000000	210000000	168405260,55

Tabla 3.3 Valores de E supuestos y los obtenidos al finalizar la calibración.

### 3.7.2 Resultados (Caso de estudio 2)

Para el caso de estudio 2 donde se calibran 3 módulos de deformación para tres tramos en la torre objeto de estudio la convergencia se logró en 2h y 3 min con un total de 21 iteraciones. En la figura 3.11 se presentan los valores de la función objetivo para cada iteración. En la tabla 3.4 se resumen los valores de las frecuencias propias de oscilación para los 5 primeros modos, los cuales fueron los tomados en cuenta a la hora de la calibración.



Iteración	Residuales
1	0.00021475
2	0.00021475
3	0.00020837
4	0.00015934
5	4.1502e-05
6	3.3324e-06
7	2.0543e-07
8	1.2563e-08
9	7.8064e-10
10	5.1074e-11
11	4.7625e-12
12	1.4963e-12
13	1.1167e-12
14	1.11e-12
15	1.0436e-12
16	1.0435e-12
17	1.0432e-12
18	1.0432e-12
19	1.0432e-12
20	1.0431e-12
21	1.0431e-12

Fig. 3.11 Residuos de la función objetivo para cada iteración.

Modos	Frecuencias propias de oscilación			Error en las frecuencias (%)		Valores MAC (%)	
	Valor Experimental	Modelo Inicial	Modelo Calibrado	Modelo Inicial	Modelo Calibrado	Modelo Inicial	Modelo Calibrado
1	1,2926	1,3849	1,2925	7,14	0,01	99.9932	99.9998
2	1,2952	1,3875	1,295	7,13	0,02	99.9956	99.9999
3	3,5706	3,872	3,5717	8,44	0,03	99.9741	99.9998

4	3,5739	3,8754	3,5749	8,44	0,03	99.9751	99.9999
5	4,7484	5,3603	4,7466	12,89	0,04	99.717	99.9995

Tabla 3.4 Frecuencias propias de oscilación, errores en las mismas y valores de la matriz MAC para los valores experimentales y los modelos inicial y calibrado.

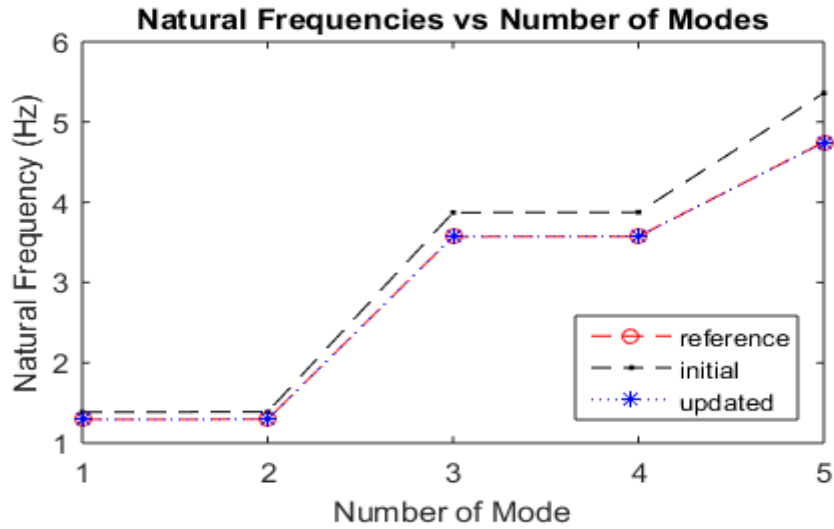
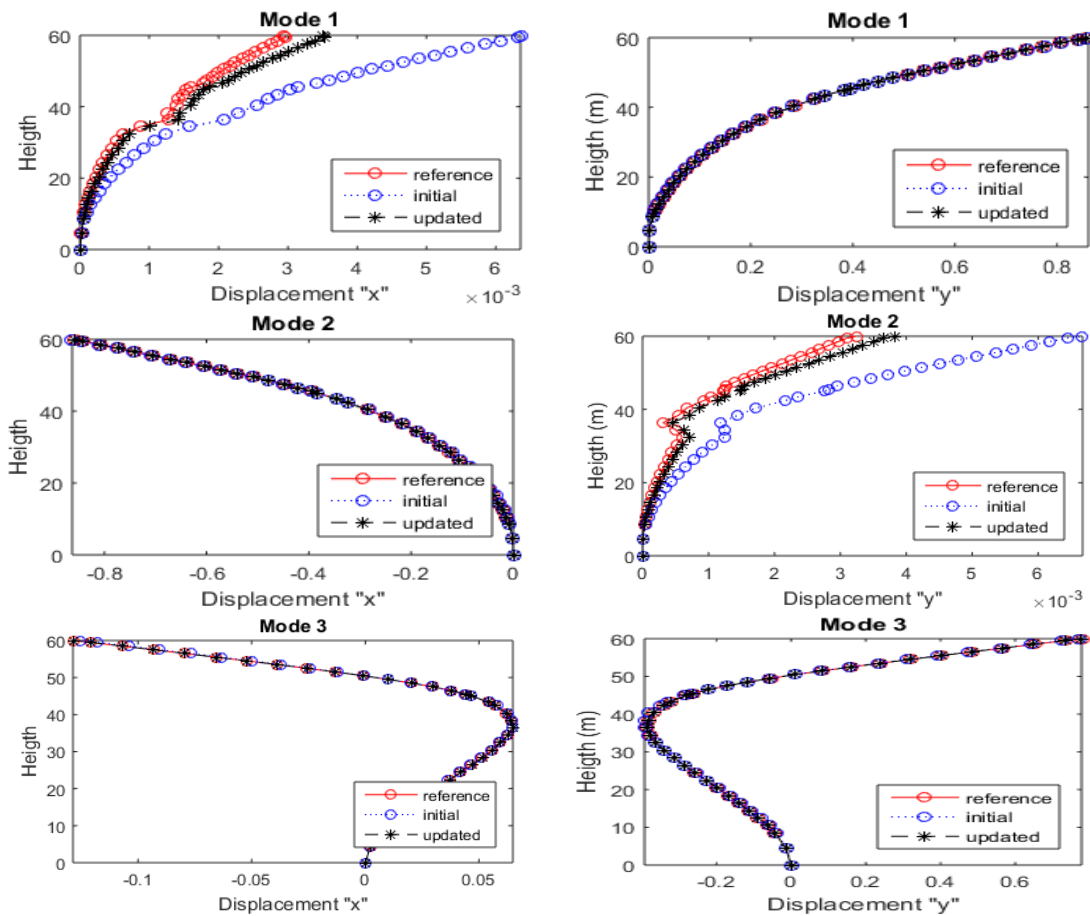


Fig. 3.12 Frecuencias propias de oscilación para los 5 primeros modos correspondientes a los valores experimentales y a los modelos inicial y calibrado.



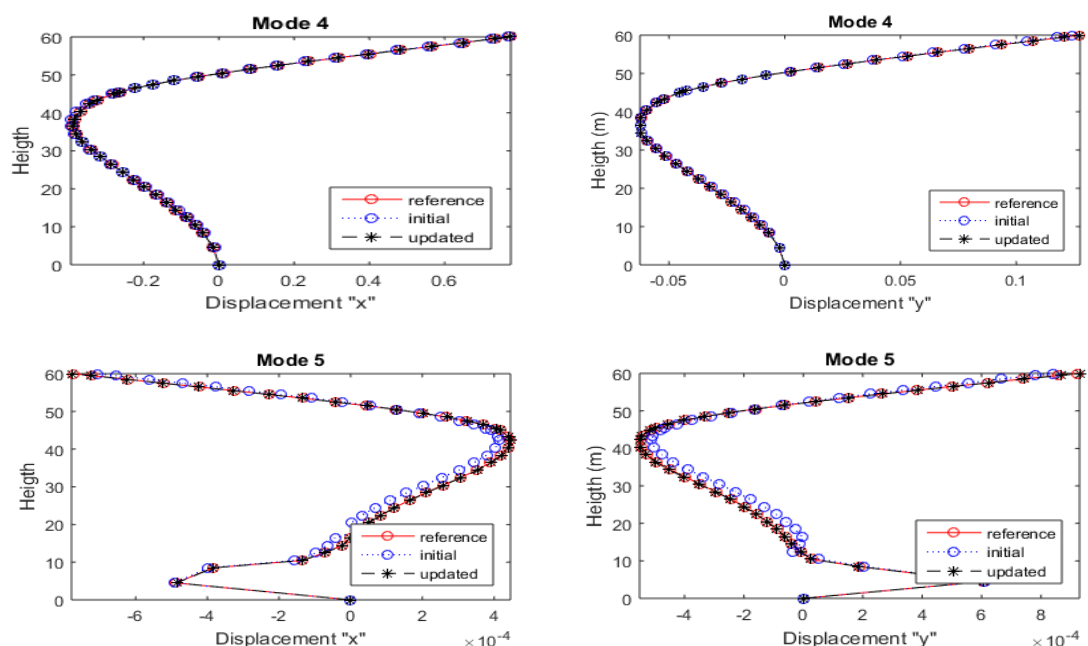


Fig. 3.13 Valores de las coordenadas modales para cada modo de los valores experimentales y los modelos inicial y calibrado. (A la izquierda coordenadas en "x" y a la derecha coordenadas en "y")

También, como los valores experimentales o de referencia son calculados a partir de lo explicado en el epígrafe 3.5, se presentan en la tabla 3.5 los valores del módulo de deformación que se supuso que la torre real tuviera y los calculados al terminar la calibración.

Modelo con inercias y áreas reducidas en 0,8	Modelo con inercias y áreas no reducidas	
Valores de E de referencia (kPa)	Valores de E Iniciales (kPa)	Valores de E calculados (kPa)
205000000	210000000	164582078,80
210000000	210000000	166718621,33
200000000	210000000	160094506,84

Tabla 3.5 Valores de E supuestos y los obtenidos al finalizar la calibración.

### 3.7.3 Resultados (Caso de estudio 3)

Para el caso de estudio 3 donde se calibran 2 rigideces de los apoyos  $k_z$  y  $k_y=k_x$  la convergencia se logró en 18min y 09 segundos con un total de 8 iteraciones. En la figura 3.14 se presentan los valores de la función objetivo para cada iteración donde se aprecia la reducción significativa de la misma entre las iteraciones 1 y 2 donde luego se mantiene muy cercana a 0 hasta alcanzar un valor menor que la tolerancia especificada de  $1,0E-16$ . En la tabla 3.6 se resumen los valores de las frecuencias propias de oscilación para los 5

primeros modos, los cuales fueron los tomados en cuenta a la hora de la calibración. En esta tabla se observa como los errores en las frecuencias iniciales son reducidos hasta 0 cuando se obtiene el modelo calibrado y se muestran los valores de la matriz MAC inicial y del modelo calibrado lográndose obtener una coincidencia del 100%.

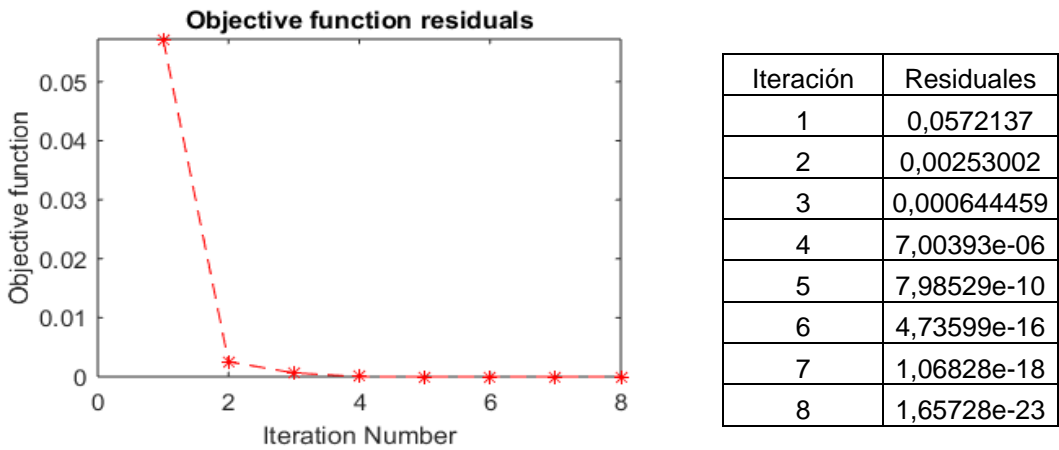


Fig. 3.14 Residuos de la función objetivo para cada iteración.

Modos	Valor Experimental	Modelo Inicial	Modelo Calibrado	Error en las frecuencias (%)		Valores MAC (%)	
				Modelo Inicial	Modelo Calibrado	Modelo Inicial	Modelo Calibrado
1	1.14	1.02	1.14	9.9	0	99.7179	100
2	1.14	1.02	1.14	9.92	0	99.7153	100
3	3.15	2.97	3.15	5.51	0	99.7608	100
4	3.15	2.98	3.15	5.50	0	99.6637	100
5	4.52	4.52	4.52	0	0	99.0998	100

Tabla 3.6 Frecuencias propias de oscilación, errores en las mismas y valores de la matriz MAC para los valores experimentales y los modelos inicial y calibrado.

En la figura 3.15 se muestra a manera de gráfico los valores de las frecuencias para cada modo para el modelo inicial, calibrado y las frecuencias de mediciones experimentales.

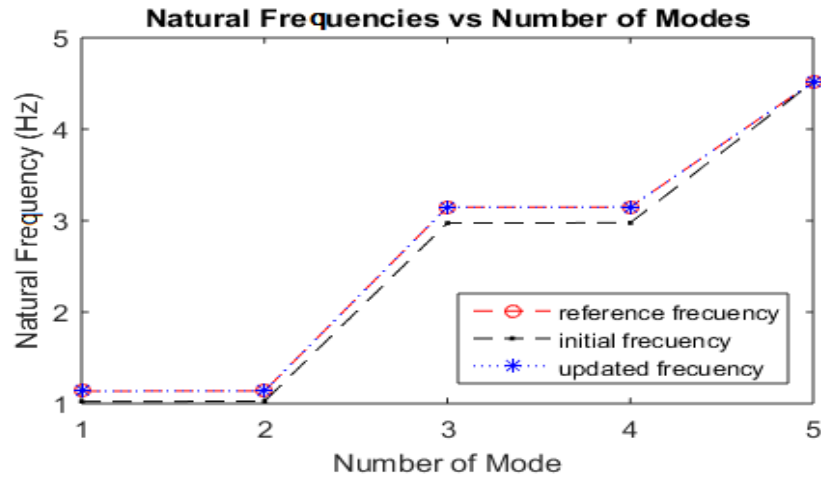
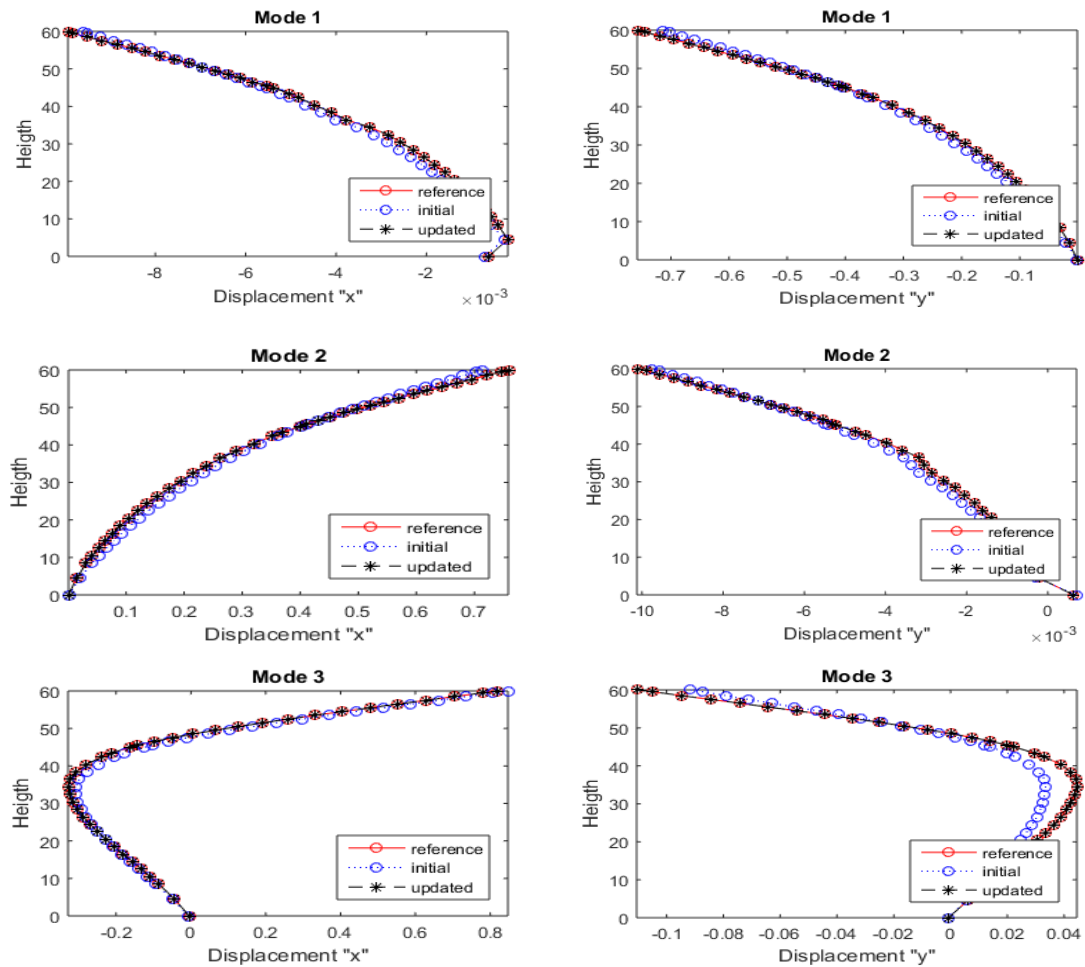


Fig. 3.15 Frecuencias propias de oscilación para los 5 primeros modos correspondientes a los valores experimentales y a los modelos inicial y calibrado.

La figura 3.16 muestra los valores de los desplazamientos modales para las dos direcciones en que se desplaza la torre “x” y “y” para los valores de referencia, del modelo inicial y del modelo calibrado. Estos valores corresponden a los nodos de una columna de la torre.



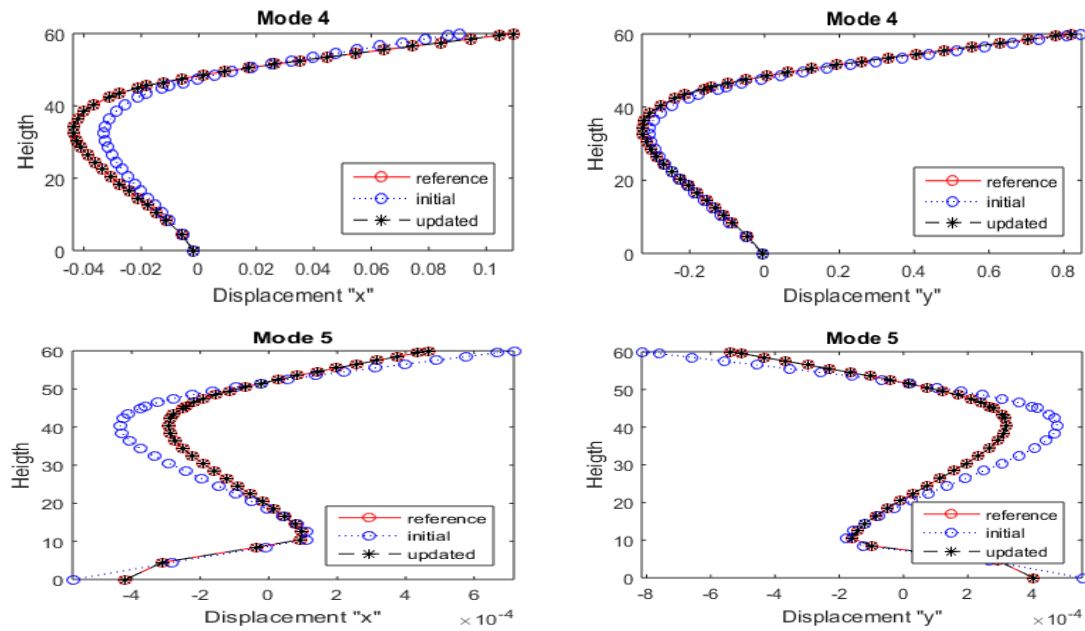


Fig. 3.16 Valores de las coordenadas modales para cada modo de los valores experimentales y los modelos inicial y calibrado. (A la izquierda coordenadas en "x" y a la derecha coordenadas en "y")

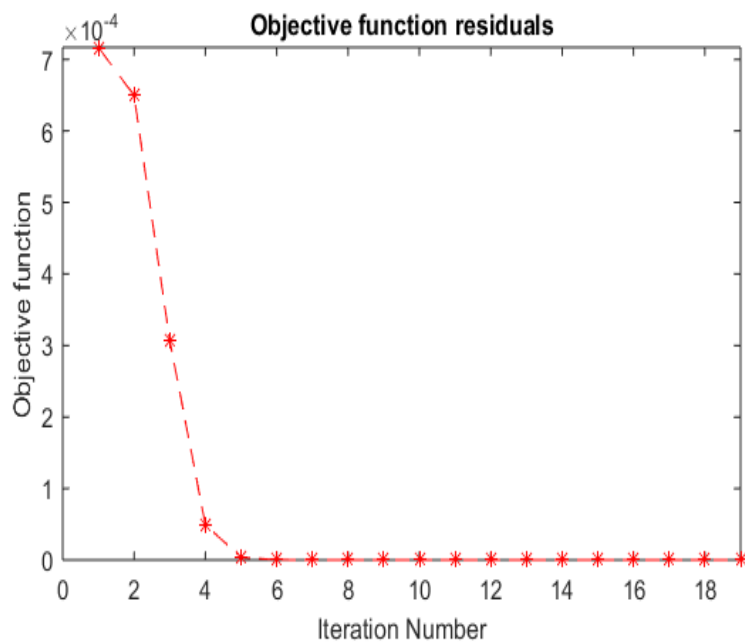
Como los valores de referencia de las rigideces de los apoyos eran a partir de reducciones de las mismas, es lógico que los valores a los que arribó la calibración fueran los mismos que los dañados lo que se puede apreciar en la tabla 3.8.

Valores de k reales del modelo (kPa)	Valores de k Iniciales (kPa)	Valores de k Calculados (kPa)	Error (%)
$k_z = 51200$	30000	51200	0
$k_x = k_y = 41100$	30000	41100	0

Tabla 3.7 Valores de k supuestos y los obtenidos al finalizar la calibración.

### 3.7.4 Resultados (Caso de estudio 4)

Para el caso de estudio 4 donde se calibran 7 módulos de deformación para 7 tramos en la torre objeto de estudio la convergencia se logró en 2 horas y 46 min con un total de 20 iteraciones. En la figura 3.17 se presentan los valores de la función objetivo para cada iteración. En la tabla 3.8 se resumen los valores de las frecuencias propias de oscilación para los 5 primeros modos, los cuales fueron los tomados en cuenta a la hora de la calibración.



Iteración	Residuales
1	0.00071693
2	0.00071693
3	0.00064962
4	0.00030725
5	4.9119e-05
6	3.5898e-06
7	2.2583e-07
8	1.4098e-08
9	1.398e-09
10	4.5737e-10
11	2.7049e-10
12	2.5696e-10
13	2.5467e-10
14	2.5435e-10
15	2.5416e-10
16	2.5416e-10
17	2.5416e-10
18	2.5415e-10
19	2.5414e-10
20	2.5414e-10

Fig. 3.17 Residuos de la función objetivo para cada iteración

Modos	Frecuencias propias de oscilación			Error en las frecuencias (%)		Valores MAC (%)	
	Valor Experimental	Modelo Inicial	Modelo Calibrado	Modelo Inicial	Modelo Calibrado	Modelo Inicial	Modelo Calibrado
1	1,2339	1,4005	1,2348	13,50	0,07	99.9989	99.9988
2	1,2363	1,4031	1,237	13,49	0,06	99.9989	99.9990
3	3,4547	3,9077	3,4617	13,11	0,20	99.9954	99.9942
4	3,4579	3,9114	3,4647	13,11	0,20	99.9955	99.9957
5	4,8048	5,3606	4,7935	11,57	0,24	99.9755	99.9946

Tabla 3.8 Frecuencias propias de oscilación, errores en las mismas y valores de la matriz MAC para los valores experimentales y los modelos inicial y calibrado.

En la figura 3.18 se muestra a manera de gráfico los valores de las frecuencias para cada modo para el modelo inicial, calibrado y las frecuencias de mediciones experimentales lo que posibilita su interpretación de una manera mucha más rápida y fácil.



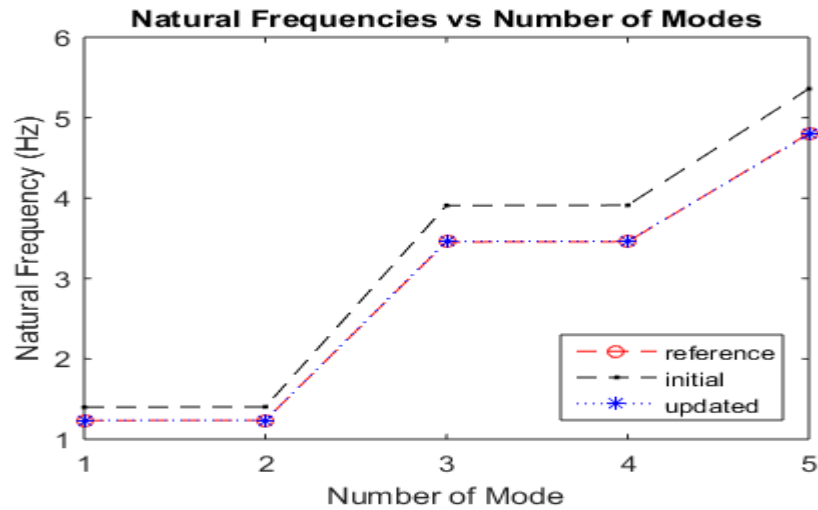
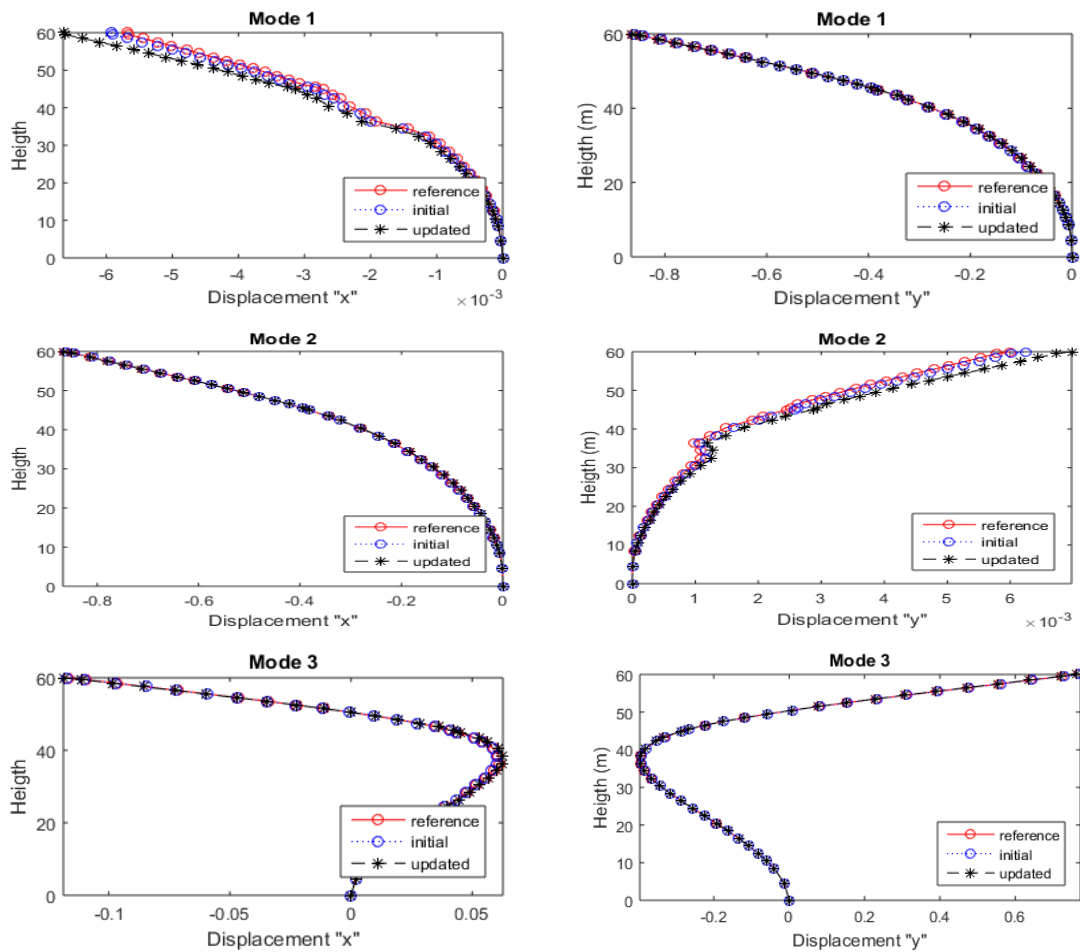


Fig. 3.18 Valores de las frecuencias propias de oscilación para cada modelo y los valores experimentales para cada modo.

La figura 3.19 muestra los valores de los desplazamientos modales para las dos direcciones en que se desplaza la torre “x” y “y” para los valores de referencia, del modelo inicial y del modelo calibrado. Estos valores corresponden a los nodos de la columna de la torre explicado al inicio de este epígrafe.



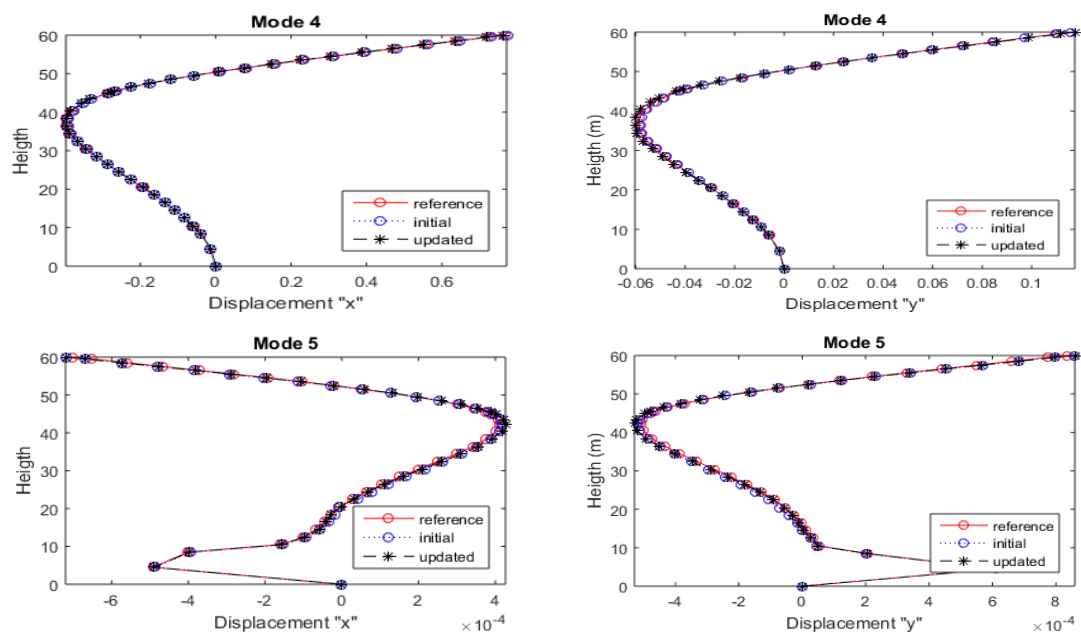


Fig. 3.19 Valores de las coordenadas modales para cada modo de los valores experimentales y los modelos inicial y calibrado. (A la izquierda coordenadas en "x" y a la derecha coordenadas en "y").

También, como los valores experimentales o de referencia son calculados a partir de lo explicado en el epígrafe 3.5, se presentan en la tabla 3.9 los valores del módulo de deformación que se supuso que la torre real tuviera y los calculados al terminar la calibración.

Modelo con inercias y áreas reducidas en 0,8	Modelo con inercias y áreas no reducidas	
Valores de E de referencia (kPa)	Valores de E Iniciales (kPa)	Valores de E calculados (kPa)
210000000	210000000	167866421,44
205000000	205000000	193658517,35
200000000	200000000	143859773,58
198000000	198000000	162264447,28
199000000	199000000	153248835,33
210000000	210000000	172517828,78
196000000	196000000	154555044,53

Tabla 3.9 Valores de E supuestos y los obtenidos al finalizar la calibración.

### 3.7.5 Análisis de los resultados

Los residuales en la función objetivo (figuras 3.8, 3.11, 3.14 y 3.17) muestran para todas las variantes una correcta disminución a medida que aumentan las iteraciones lo cual demuestra que el algoritmo de optimización implementado fue encontrando la convergencia sin pérdidas de tiempo o iteraciones innecesarias, también todos los casos de estudio arribaron a residuos más pequeños que  $1,0e-10$  lo que demuestra una gran precisión en la calibración. Se puede notar que siempre, en los casos 1, 2 y 4, a partir de una iteración específica los residuos disminuyen muy poco o se mantienen casi iguales y es debido a que el algoritmo ya encontró un arreglo de variables de diseño que aproximen las respuestas modales a las de referencias y los residuos correspondientes a este arreglo son más grandes que la tolerancia permitida  $1,0e-16$ , por lo que la optimización se siguió ejecutando hasta que la distancia del paso fue la que sobrepasó la tolerancia  $1,0e-8$ .

Al analizar las tablas 3.2, 3.4, 3.6 y 3.8 y las figuras 3.9, 3.12, 3.15 y 3.18 se puede observar como para el modelo inicial, o sea los valores de las variables de diseño iniciales, en la primera iteración, las frecuencias de oscilación muestran una notable diferencias para algunos modos en algunas variantes logrando luego al finalizar y obtener los valores calibrados errores muy pequeños siempre menores que el 0,24%. También los valores MAC iniciales muestran valores muy cercanos al 100%, mientras que al finalizar son aún más grandes lo que muestra una perfecta coincidencia entre los modos experimentales y los obtenidos al final de la calibración.

En las figuras 3.10, 3.13, 3.17 y 3.20 se puede apreciar como las formas modales para los modelos calibrados y de referencia muestran valores muy exactos, exceptuando las formas modales en “x” para el primer modo y las formas modales en “y” para el segundo modo en los casos de estudio 1,2 y 4 donde hay pequeñas diferencias lo que provoca que los residuos en la función objetivo se hayan mantenido en el orden de  $1,0e-10$  a  $1,0e-12$  para estos casos.

En las tablas 3.3, 3.5, 3.7 y 3.9 se presentan los valores de las variables de diseño para los modelos dañados y los calibrados donde exceptuando la variante 3, los valores que se obtienen de E son mucho más pequeños que los supuestos dañados, esta reducción es debida a que se necesita igualar las reducciones que provocan las disminuciones en las inercias y las áreas.

En la figura 3.21 se puede observar el tiempo de calibración para cada caso de estudio donde es importante aclarar que todas las calibraciones fueron realizadas en un mismo ordenador, este ordenador no cuenta con especificaciones muy altas (2GB de RAM y un

microprocesador Celeron 2.6Hz.) por lo que es importante distinguir, más que el tiempo total, las variaciones de cada tiempo en función de la cantidad de variables a calibrar.

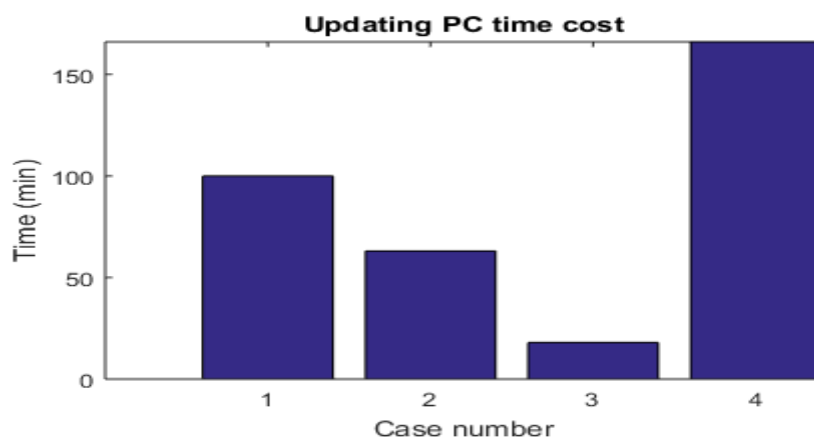


Fig. 3.20 Tiempo de calibración consumido para los distintos casos de estudios implementados.

### 3.7.6 Validación de los modelos

La validación de los modelos se realiza a partir de aplicarle un conjunto de cargas puntuales en un conjunto de nodos de la torre a los modelos, el dañado y el calibrado, y extraer del SAP2000 los desplazamientos en los todos los nodos de una columna de la torre, la misma columna a la que se le extrajeron las formas modales, para analizar el comportamiento de los modelos. Para el caso de estudio 3 no se realizó el procedimiento debido a que los valores de rigideces para el modelo dañado fue las mismas obtenidas al final de la calibración por lo que es de esperar que los desplazamientos en la estructura sean de la misma magnitud.

Las figuras muestran los desplazamientos ploteados en las dos direcciones “x” y “y” debido a las cargas impuestas para los modelos inicial, de referencia y calibrados.

#### Caso de estudio 1:

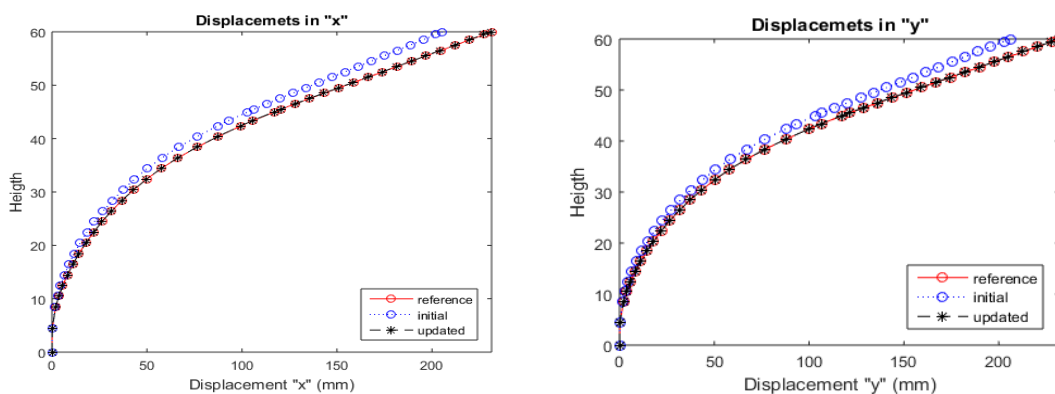


Fig. 3.21 Desplazamientos en las direcciones de “x” y “y” debido a las cargas impuestas en los modelos. Caso de estudio 1.

#### Caso de estudio 2:

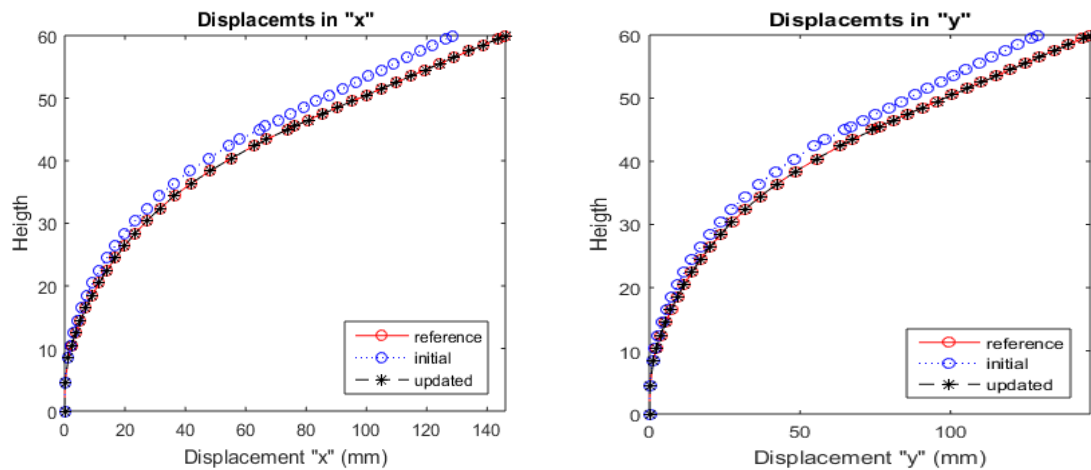


Fig. 3.22 Desplazamientos en las direcciones de "x" y "y" debido a las cargas impuestas en los modelos.

#### Caso de estudio 4:

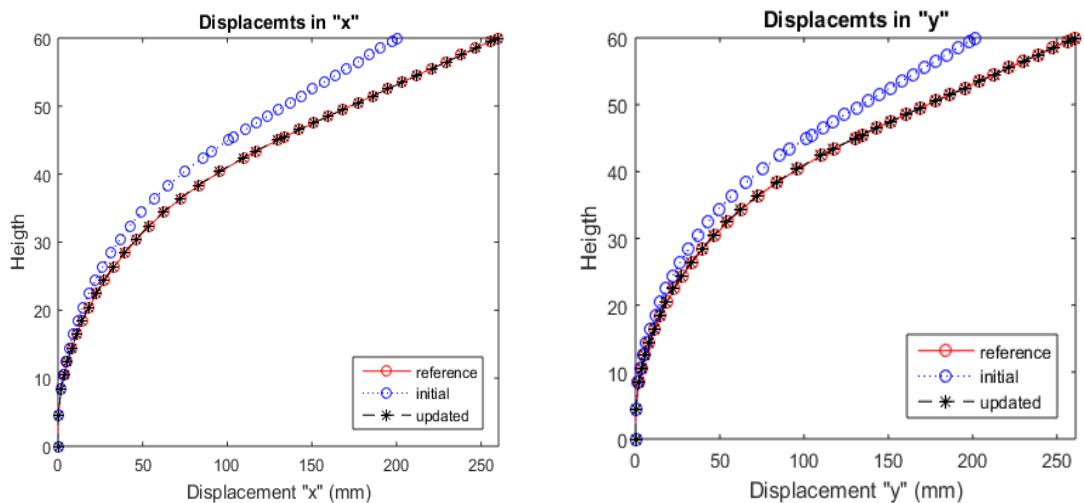


Fig. 3.23 Desplazamientos en las direcciones de "x" y "y" debido a las cargas impuestas en los modelos.

Al analizar estas figuras para estos casos, se distinguen valores favorables debido a que los desplazamientos para los modelos calibrado y de referencia son los mismos lo que aprueba que los resultados alcanzados al terminar la calibración representan en comportamiento del modelo de referencia o dañado por lo que cualquier adición de cargas o modificaciones en estos modelos tiene que provocar resultados muy parecidos o casi iguales.

### 3.8 Conclusiones parciales del capítulo

- La modelación de la torre objeto de estudio se realizó, primero, utilizando el programa AUTOCad lo que permitió mayores posibilidades y facilidades de dibujo, y luego, el dibujo fue importado por SAP2000v16.
- La localización de los sensores en una torre de telecomunicaciones es de gran importancia debido a que pocos sensores o ubicaciones incorrectos de los mismos darían una calibración con ineficiencias y errores. La matriz AUTOMAC programada en el script (sens\_anal\_torre) fue utilizada para este propósito.
- La calibración de la torre objeto de estudio se realizó para 4 variantes de variables de diseño lo que permitió una mejor evaluación del procedimiento automatizado de calibración implementado.
- Como los parámetros dinámicos medidos o experimentales de la torre no se obtuvieron de mediciones experimentales, su cálculo se procedió a partir de lo explicado en el epígrafe 3.5, donde se aplican daños a diversos tramos de la torre, lo que permite obtenerlos de una manera razonable para ser utilizados en la calibración.
- El método empleado en la optimización fue uno de los que MATLAB cuenta para la función lsqnonlin: el **Levenberg-Marquardt Method**.
- La aplicación de los algoritmos de optimización empleados permite contar con modelos con un alto grado de calibración, residuos siempre menores que  $1,0e-10$  para las variantes implementadas.
- El procedimiento implementado permitió contar con modelos calibrados con residuos menores de  $1,0e-10$  y errores en las frecuencias propias de oscilación y las formas modales menores de 0,5% lo que lo hace una herramienta muy poderosa en este tipo de trabajos.

## Conclusiones

- La interface OAPI SAP2000 garantiza la interacción entre SAP2000 y MATLAB, permitiendo que la optimización en la calibración de los modelos mecánicos de torres autosoportadas sea posible;
- La calibración de los modelos mecánicos de torres autosoportadas de telecomunicaciones tiene como objetivo fundamental acercar estos modelos lo más posible a la realidad, permitiendo evaluar el comportamiento de la estructura bajo la acción de diversos estados de carga muy difíciles de ensayar en condiciones reales.
- Una gran ventaja de esta investigación es que el modelo mecánico de la torre, aunque se puede construir en MATLAB mediante los comandos OAPI, se realiza en SAP2000, facilitando la programación al ingeniero.
- A medida que aumenta la cantidad de parámetros modales y la cantidad de variables de diseño incluidos en el análisis el procedimiento se complejiza. Esto mejora el porcentaje de calibración, pero a costa de un mayor tiempo para la programación y ejecución del mismo. Por lo tanto, se debe elegir la estrategia adecuada para obtener resultados satisfactorios en un tiempo aceptable.
- El procedimiento implementado permitió contar con modelos calibrados con residuos menores de  $1,0e-10$  y errores en las frecuencias propias de oscilación y las formas modales menores de 0,5% lo que lo hace una herramienta muy poderosa en este tipo de trabajos.

## Recomendaciones

- 1) Aplicar esta investigación no solo a torres autosoportadas sino a torres arriostradas o atirantadas donde se tengan resultados experimentales de mediciones reales efectuadas.
- 2) Utilizar además otros tipos de programas como:
  - ANSYS-MATLAB
  - ANSYS con su propio toolbox de optimización
  - ABAQUS-MATLAB
  - SAP2000-Microsoft Excel
  - ETABS-MATLAB
- 3) Extender el procedimiento implementado a la calibración de otros parámetros como, la resistencia del acero, las inercias en los perfiles, rigideces de las uniones, etc.



## BIBLIOGRAFIA Y REFERENCIAS

- ABDEL WAHAB, M. & DE ROECK, G. 1999. Damage detection in bridges using modal curvatures: Application to a real damage scenario. *J. Sound Vib.*, 2, 217–235.
- ABDULLAH, N. A. Z., SANI, M. S. M., RAHMAN, M. M. & ZAMAN, I. 2016. A review on model updating in structural dynamics. [Accessed 25/11/2016 at 17:05].
- AFONSO ASPIRO, Y. 2016. Estimación de parámetros estructurales en puentes usando la OAPI SAP2000 - MATLAB. Santa Clara.
- ALONSO, E. 1989. Instrumentación de obras, Universidad Politécnica de Cataluña. Barcelona.
- ÁLVAREZ, M. R. 2011. *Estudio de la no linealidad geométrica en torres atirantadas de telecomunicaciones bajo carga de viento extremo*. Tesis de grado, CUJAE.
- ALLEMANG, R. J. 2003. The Modal Assurance Criterion – Twenty Years of Use and Abuse. *Sound and vibration* [Online].
- ANÓNIMO 2010. Estimación de parámetros, validación de modelos y análisis de sensibilidad.
- ANÓNIMO 2015. Análisis modal experimental. *Análisis modal experimental*.
- ARAGÓN MIRANDA, Y. 2010. *Modelación estructural de un puente metálico de ferrocarril de viga, tablero superior*. Trabajo de Diploma, Universidad Central “Marta Abreu” de Las Villas
- ASPIRO, Y. A. 2016. *Estimación de parámetros estructurales en puentes usando la OAPI SAP2000 - MATLAB*. Pregrado, Universidad Central “Marta Abreu” de Las Villas.
- BACUILIMA G, F. L. 2011. *Técnicas de optimización para la identificación de parámetros mecánicos de estructuras* [Online]. Cuenca.
- BONILLA ROCHA, J. D. 2008. Estudio del comportamiento de conectores tipo perno de estructuras compuestas de hormigón y acero mediante modelación numérica. Santa Clara.
- CANTIENI, R. 1984. A Laser System for the Measurement of Displacements of Structures Under Static and Dynamic Loading. *Industrial Applications of Lasers*, Ch., 231-235.

- CARRIÓN VIRAMONTES, F. J., LOZANO GUZMÁN, A., FABELA GALLEGOS, M. D. J., VÁZQUEZ VEGA, D. & ROMERO NAVARRETE, J. A. 1999. Evaluación de puentes mediante el análisis de vibraciones. Investigaciones recientes. *Publicación Técnica No. 132*, 98.
- CLARO DUMÉNIGO, A. 2015. *Métodos para la modelación y el análisis experimental de puentes frente a cargas dinámicas*. Tesis de grado.
- COMPUTERS & STRUCTURES, I. 2013. CSI OAPI Documentation. In: COMPUTERS & STRUCTURES, I. (ed.).
- COPYRIGHT © COMPUTERS AND STRUCTURES, I., 1978-2013. 2013. Introductory Tutorial for SAP2000®. In: COMPUTERS & STRUCTURES, I. (ed.) *Linear and Nonlinear Static and Dynamic Analysis and Design of Three-Dimensional Structures*. United States of America.
- CUJIA MEZA, Y. D. 2010. COMPUTACIÓN EN PARALELO APLICADA A LA OPTIMIZACIÓN DEL DISEÑO ESTRUCTURAL: ESTADO DEL ARTE. Bucaramanga.
- CHAGOYÉN MÉNDEZ, D. E. L. OAPI SAP 2000 - Matlab. Introducción a OAPI SAP 2000 - Matlab., Mayo-Junio 2015 2015 Facultad de Construcciones, Departamento de Ing Civil, UCLV., 46.
- CHAN, T. H. T. & O'CONNER, C. 1990. Wheel Loads from Highway Bridge Strains: Field Studies. *ASCE Journal of Structural Engineering*, 116.
- DELGADO, A. A., O'REILLY, E. G. & DÍAZ, J. C. 2016. Multimedia para la modelación de torres en SAP 2000.
- EYRE, R. & TILLY, G. P. 1977. Damping Measurement on Steel and Composite Bridges. U.K.: Crwthorne.
- FENG, U. Z., KIM, Y. H. & YANG, B.-S. 2006. Applications of hybrid optimization techniques for model updating of rotor shafts. *Struct Multidisc Optim*, 32, 65-75.
- FIGUEROA CASTELLÓN, O. & PINO RODRÍGUEZ, M. L. 2010. *MODELACION MECANICA DE ESTRUCTURAS ISOSTATICAS*. Apuntes., Santa Clara.
- FRITZEN, C.-P., D.JENNEWEIN & KIEFER, T. 1997. DAMAGE DETECTION BASED ON MODEL UPDATING METHODS.
- GARCÍA, C. G. 2011. *ANÁLISIS ESTRUCTURAL DE TORRES AUTOSOPORTADAS BAJO CARGAS DE VIENTO Y SISMO.*, Instituto Politécnico Jose Antonio Echeverría
- GARCIA DE JALON, J., RODRIGUEZ, J. I. & VIDAL, J. 2005. "Aprenda MATLAB 7.0 como si estuviera en primero". Universidad Politécnica de Madrid.

- GAUCHÍA, A., BOADA, B. L., BOADA, M. J. L. & DÍAZ, V. 2014. Integration of MATLAB and ANSYS for Advanced Analysis of Vehicle Structures. INTECH.
- GERO, M. B. P. 2002. *Técnica y construcción de edificios metálicos en base a algoritmos genéticos*. Doctoral, Universidad de Oviedo.
- GONZÁLES PÉREZ, A. G. 2015. Calibración y validación de modelos de puentes. Santa Clara.
- GONZÁLEZ, D. B. 2010. *Sistema de evaluación de la condición física para el mantenimiento de torres Autosoportadas de telecomunicaciones*. Tesis de grado, CUJAE.
- GONZALEZ MARTINEZ, D. 2007. *Modelación del puente metálico del ferrocarril km 9.568 del Ramal Santa Clara-Cenizas mediante el Método de los Elementos Finitos*. Universidad Central "Marta Abreu" de Las Villas.
- GUILLÉN, E. L. 2002. Identificación de Sistemas. Aplicación al modelado de un motor de continua.
- GUSTAFSSON, B. 2011. *Fundamentals of Scientific Computing*, London, England, Springer Heidelberg Dordrecht London New York.
- GUTIÉRREZ, S., ALEJANDRO, D., RODRÍGUEZ, S. & FELIPE, L. 2013. ALGORITMO PARA LA CALIBRACIÓN DE MODELOS ESTRUCTURALES EN ELEMENTOS FINITOS DE PUENTES USANDO ANSYS. Bogotá.
- HE, J. & FANG FU, Z. 2001. *Modal Analysis*, Linacre House, Jordan Hill, Oxford OX2 8DP, 225 Wildwood Avenue, Woburn, MA 01801-2041, Butterworth-Heinemann.
- IDS INGEGNERIA DEI SISTEMI, S. P. A. 2013. IBIS-FS DYNAMIC MONITORING GEORADAR.
- ISON, M., SITT, J. & TREVISAN, M. 2005. Algoritmos genéticos: aplicación en MATLAB. *genetico.tar.gz* [Online]. Available: [www.df.uba.ar](http://www.df.uba.ar) [Accessed Noviembre 25].
- JASIM, N. A. & GALEB, A. C. 2002. Optimum design of square free-standing communication towers. *Journal of Constructional Steel Research*, 58 (2002) 413–425, 13.
- KANEV, S., WEBER, F. & VERHAEGEN, M. 2007. Experimental validation of a finite-element model updating procedure. *Journal of Sound and Vibration*, 20.
- KATIA 2016. Presentación taller las villas.

- LÓPEZ, F. C. 2010. *Optimización global con algoritmos genéticos*. Tesis de grado, Universidad Politécnica de Cataluña.
- LORENTE, A. G. 2012. *Catálogo de deterioros para torres autosoportadas de telecomunicación*. Trabajo de diploma, Instituto Superior José Antonio Hecheverría.
- LOSA MIRANDA, G. 2015. *Tecnologías de sensorización e identificación modal. Aplicación a la determinación del amortiguamiento estructural*. Grado en Ingeniería Mecánica, UNIVERSIDAD DE VALLADOLID ESCUELA DE INGENIERÍAS INDUSTRIALES.
- LLANUSA, A. L. 2013. *Influencia de la presencia de antenas en torres de telecomunicaciones en la vulnerabilidad estructural ante cargas de viento*. Maestría, Instituto Superior Politécnico José Antonio Echeverría.
- MAES, K., LOMBAERT, G., NIMMEN, K. V., PAPADOPOULOS, M. & ROECK, G. D. 2017. Model updating in structural dynamics. *Post-graduate course – Model updating in Structural Dynamics TEAM-VLIR PROJECT “VIBRAS”*. Santa Clara, Villa Clara, Cuba.
- MAGALHÃES, F. M. R. L. D. 2012. *OPERATIONAL MODAL ANALYSIS FOR TESTING AND MONITORING OF BRIDGES AND SPECIAL STRUCTURES*. Doctoral, Universidade do Porto.
- MALEKZEHTAB, H., GOLAFSHANI, A. A. & MOHAJERI, S. 2011. Application of Genetic Algorithm Based Finite Element Model Updating in Damage Detection of Offshore Jacket Platforms.
- MARWALA, T. 2010. Finite-element-model Updating Using Computational Intelligence Techniques. *In: YORK, S. L. D. H. N. (ed.)*. South Africa.
- MÉNDEZ, K. C. G. 2015. *ANÁLISIS DE UNA TORRE DE TELECOMUNICACIONES ATIRANTADA ANTE LOS EFECTOS DE VIENTO: RELACIÓN DEL COSTO DE LA TORRE EN FUNCIÓN DE LA VELOCIDAD DEL VIENTO*. Pregrado, UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO.
- MONROY RINCÓN, J. C. & ARRUBLA TOVAR, M. 2013. Estimación de cargas dinámicas de camiones pesados que circulan por un puente instrumentado con acelerómetros. . Bogotá D.C.
- MOORE, H. 2007. *MATLAB® para ingenieros*, México, PEARSON EDUCACIÓN.
- MORRIS, B. 2009. *AN INVESTIGATION OF HUMAN INDUCED VIBRATIONS ON CLEMSON MEMORIAL STADIUM*. Pregrado, Clemson University.
- MOTTERSHEAD, J. E. & FRISWELL, M. I. 1992. Model updating in structural dynamics: A survey.

- NASH, J. E. & SUTCLIFFE, I. V. 1970. RIVER FLOW FORECASTING THROUGH CONCEPTUAL MODELS. *Journal of Hydrology*, PART I - A DISCUSSION OF PRINCIPLES\*.
- PARNÁS, D. I. V. E., RODRÍGUEZ, M. I. P. M., LUZARDO, I. A. C., LORENZO, I. I. F. & LLANUSA, I. A. L. 2012. METODOLOGÍA DE CÁLCULO DE TORRES ATIRANTADAS DE TELECOMUNICACIONES BAJO LA ACCIÓN DEL VIENTO.
- QUARTERONI, A. 2010. *Scientific Computing with MATLAB and Octave*, Springer Heidelberg Dordrecht London New York.
- RAMÍREZ DÍAZ, R., GÁMEZ BRETO, Y., CHAGOYÉN MÉNDEZ, E. & PÉREZ LECUSAY, H. 2015. Evaluación de un puente de vías férrea mediante ensayos estructurales y modelación computacional. *Obras y Proyectos*, 18, 32-43.
- RAMOS, L. S. F., AGUILAR, R., LOURENCO, P. B. & MOREIRA, S. 2013. Dynamic structural health monitoring of Saint Torcato church. *Mechanical Systems and Signal Processing*, 35 (2013) 1–15, 15.
- RECAREY, C. A. 1999. Modelación del terreno y las estructuras en el dominio del tiempo. Santa Clara.
- RECIO, H. C. P. 2011. *Diseño de cimentaciones para torres atirantadas de telecomunicaciones*. Tesis de grado, INSTITUTO SUPERIOR POLITÉCNICO "JOSÉ ANTONIO ECHEVERRÍA" FACULTAD DE INGENIERÍA CIVIL.
- REYNDERS, E., DE ROECK, G., GUNDES BAKIR, P. & SAUVAGE, C. 2007. Damage Identification on the Tiff Bridge by Vibration Monitoring Using Optical Fiber Strain Sensors. *Journal of Engineering Mechanics*, 133 No. 2.
- REYNDERS, E. & ROECK, G. D. 2014. System identification and parameter estimation in civil engineering. *VLIR-UOS Computational techniques for engineering applications* Santa Clara, Cuba.
- RODRÍGUEZ CALDERÓN, W., MUÑOZ DÍAS, E. & NUÑEZ MORENO, F. 2007. Optimización aplicada a la calibración y validación de modelos de elementos finitos de puentes. *Ingeniería*, 17, 43-59.
- RODRÍGUEZ CASTELLANOS, M. 2010. Modelación estructural de un puente metálico ferroviario de viga continua. Santa Clara.
- RODRÍGUEZ CASTELLANOS, M. 2014. *Influencia del comportamiento dinámico en la respuesta de los puentes metálicos de ferrocarril de Cuba para su conservación y/o evaluación*. Tesis de Maestría.
- RODRIGUEZ PÉREZ DE ALEJO, Y. Introducción a la Informática. Conferencia #4, 2011.

- ROECK, G. D. 2003. *INVERSE MODELLING OF CIVIL ENGINEERING STRUCTURES BASED ON OPERATIONAL MODAL DATA*. KATHOLIEKE UNIVERSITEIT te LEUVEN.
- RUIZ ROJAS, G. A. 2005. *Diseño, construcción y caracterización de un sensor para medir cargas a tensión por extensometría*. Tesis de grado, UNIVERSIDAD TECNOLÓGICA DE LA MIXTECA.
- S.A., E. C. 2011. Diccionario Español. *In: CALPE, E. (ed.) Diccionario Español*. Espasa Calpe.
- SEXTOS, A. G. & BALAFAS, G. K. THE NEW SAP2000 OPEN APPLICATION PROGRAMMING INTERFACE TO DEVELOP AN INTERACTIVE FRONT-END FOR THE MODAL PUSHOVER ANALYSIS OF BRIDGES. 3rd ECCOMAS Thematic Conference on Computational Methods in Structural Dynamics and Earthquake Engineering 2011 Corfu, Greece. COMPDYN, 14.
- SILVA, J. G. S. D., S.VELLASCO, P. C. G. D., ANDRADE, S. A. L. D. & OLIVEIRA, M. I. R. D. 2005. Structural assessment of current steel design models for transmission and telecommunication towers. *Journal of Constructional Steel Research*, 61 (2005) 1108–1134, 27.
- SIMOEN, E. 2013. *UNCERTAINTY QUANTIFICATION IN FINITE ELEMENT MODEL UPDATING*. Doctoral, KU-LEUVEN ARENBERG DOCTORAL SCHOOL.
- SIMOEN, E., ROECK, G. D. & LOMBAERT, G. 2013. Dealing with uncertainty in model updating for damage assessment: A review.
- SMITH, D. M. 2010. *ENGINEERING COMPUTATION with MATLAB*, Library of Congress Cataloging-in-Publication Data.
- SOVERO, M. 2014. *Herramienta para el proceso de calibración de modelos numéricos*. PONTIFICIA UNIVERSIDAD CATOLICA DEL PERÚ.
- TARANTOLA, A. 2005. Inverse Problem Theory and Methods for Model Parameter Estimation. United States of America: Library of Congress Cataloging-in-Publication Data.
- TEAM, T. 2017. Najasa.
- TEUGHEL, A., MAECK, J. & ROECK, G. D. 2002. Damage assessment by FE model updating using damage functions. *Computer & Structures* [Online].
- WU, J. R. & LI, Q. S. 2003. Finite element model updating for a high-rise structure based on ambient vibration measurements.

YÁÑEZ, Y. N. 2010. *Análisis de fallas en torres de telecomunicaciones. Caso de estudio: Modelo Najasa*. Trabajo de Diploma Tesis de grado, Instituto Superior Politécnico José Antonio Echeverría.

ZAFORTEZA, I. P. 2007. *OPTIMIZACIÓN HEURÍSTICA DE PÓRTICOS DE EDIFICACIÓN DE HORMIGÓN ARMADO*. Doctoral, Universidad Politécnica de Valencia.

ZHANG, L., BRINCKER, R. & ANDERSEN, P. 2010. An Overview of Operational Modal Analysis: Major Development and Issues.

## ANEXOS

En el anexo se presenta todo el contenido de los scripts que sirvieron para la calibración.

### Programación del script (main\_mdof\_torre):

#### MASTER FILE UPDATING MDOF SYSTEM

- JULY 2017
- Kristof Maes
- Modified by Alejandro Fuentes Diaz
- Units: kN, m, s

```
tic  
close all; clear all; clc
```

#### Defining damaged and updated model locations

1. Define model paths in the PC, these model paths correspond to the reference model path, named **file\_damaged**, and the model to be updated path named **file\_upd**.
2. Save those paths in a **.mat** file.

```
file_damaged='G:\ING CIVIL\Tesis 2016\Matlab files\Tower model 7 E calibrated\Damaged  
model\Najasa Modal continua.sdb';  
file_upd='G:\ING CIVIL\Tesis 2016\Matlab files\Tower model 7 E calibrated\Updated  
model\Najasa Modal continua.sdb';  
  
save upd_param_mdof_ern file_damaged file_upd
```

#### Reference model execution = Computing the reference dynamic parameters

1. Computing the reference dynamic parameters. These are the field measurements in this case.

```
data_mdof_torre;
```

#### Perform model updating

2. Enter the starting values for all variables

```
x_0 = [2.1*10^8 2.1*10^8 2.1*10^8 2.1*10^8 2.1*10^8 2.1*10^8 2.1*10^8];
```

#### Define weighting factors



```
alpha = 1;      % weight assigned to error on eigenvalues
beta  = 1;      % weight assigned to error on mode shapes
```

### Saving the inputs in a .mat file

Saving the defined parameters in a .mat file to be used for others scripts

```
save upd_param_mdof_ern alpha beta X_0 file_damaged file_upd
```

### Performing nonlinear optimization

3. Loading the number of parameter to be updated.
4. Defining the starting point as a non-unit value.
5. Performing optimization.
6. Defining the updated parameters.

```
load data_mdof_ern nparameters;

T_0 = ones(1,nparameters);

[T_upd] = upd_mdof_torre(T_0);
X_upd = X_0 .* T_upd;
```

### Loading intermediate optimization results

1. Loading the initial, reference and updated frequencies and mode shapes.
2. Select the modes to take account.

```
load data_mdof_ern DOF_ref phi_ref omega_ref
[phi_0,omega_0,DOF_0]=eigfem_mdof_torre(X_0,file_upd);
[phi_upd,omega_upd,DOF]=eigfem_mdof_torre(X_upd,file_upd);
modSel=1:5;
phi_ref=phi_ref(:,modSel);
omega_ref=omega_ref(modSel);
phi_0=phi_0(:,modSel);
omega_0=omega_0(modSel);
phi_upd=phi_upd(:,modSel);
omega_upd=omega_upd(modSel);
```

### Rearranging data for plotting

Rearranging the initial and updated modal shapes to match the sign of the reference modal shape.

```
for i=1:length(phi_ref) % Rearrenging the initial mode shapes
    for k=modSel
        if phi_0(i,k)<=0 && phi_ref(i,k)<=0 || phi_0(i,k)>=0 && phi_ref(i,k)>=0
            phi_0(i,k)=phi_0(i,k);
        else
```

```

        phi_0(i,k)=-phi_0(i,k);
    end
end
end

for i=1:length(phi_ref) % Rearrenging the updated mode shapes
    for k=modSel
        if phi_upd(i,k)<=0 && phi_ref(i,k)<=0 || phi_upd(i,k)>=0 && phi_ref(i,k)>=0
            phi_upd(i,k)=phi_upd(i,k);
        else
            phi_upd(i,k)=-phi_upd(i,k);
        end
    end
end
end

```

### Choose the sensor position

Choose the sensor positions in SAP2000 model. These sensor positions correspond to nodes in the SAP2000 model.

```

DOF_sensors=[ [62,97,122,128,203,228,245,263,27,28,141,140,1,2,151,150,...
               ]+0.01, [63,98,123,129,178,253,160,258]+0.02] .';

```

### Selecting L matrixes

```

L_ref=selectdof(DOF_ref,DOF_sensors);
L_0=selectdof(DOF_0,DOF_sensors);

```

### Presenting updating results

Presenting results as the updated parameter and the Eigen frequencies for the initial, updated and reference models.

```

disp('Elasticity modulus initial [kN/m2]')
fprintf('%15.2f',X_0)
fprintf('\n')
disp('Elasticity modulus updated [kN/m2]')
fprintf('%15.2f',X_upd)
fprintf('\n')
fprintf('\n')
disp('Identified natural frequencies [Hz]')
fprintf('%9.2f',omega_ref/2/pi)
fprintf('\n')
disp('Calculated natural frequencies - (initial) [Hz]')
fprintf('%9.2f',omega_0/2/pi)
fprintf('\n')
disp('Calculated natural frequencies - (updated) [Hz]')
fprintf('%9.2f',omega_upd/2/pi)
fprintf('\n')
disp('Error natural frequencies - (initial) [Hz]')

```

```

fprintf('%9.2f',(omega_0-omega_ref)/2/pi)
fprintf('\n')
disp('Error natural frequencies - (initial) [%]')
fprintf('%9.2f',(omega_0-omega_ref)./omega_ref*100)
fprintf('\n')
disp('Error natural frequencies - (updated) [Hz]')
fprintf('%9.2f',(omega_upd-omega_ref)/2/pi)
fprintf('\n')
disp('Error natural frequencies - (updated) [%]')
fprintf('%9.2f',(omega_upd-omega_ref)./omega_ref*100)
fprintf('\n')
fprintf('\n')
disp('MAC (initial %) [-]')
fprintf('%15.5f',diag(mac(L_0*phi_0,L_ref*phi_ref))*100)
fprintf('\n')
disp('MAC (updated %) [-]')
fprintf('%15.5f',diag(mac(L_0*phi_upd,L_ref*phi_ref))*100)
fprintf('\n')

```

## Plotting results

```

load results_mdof T_iter resnorm_iter

figure %Plotting the norm of residuals of the objective function
plot(resnorm_iter(2:length(resnorm_iter)),'r--*')
hold on
ylim([0,inf])
xlim([0,inf])
xlabel('Iteration Number')
ylabel('Objective function')
title('Objective function residuals')
set(gca,'fontsize',11)

%Plotting the natural frequencies of the reference, initial and updated
%model
figure
plot(modSel,omega_ref/2/pi,'r--o',modSel,omega_0/2/pi,'k--.',modSel,...
     omega_upd/2/pi,'b:*')
hold on
title('Natural Frequencies vs Number of Modes')
xlabel('Number of Mode')
ylabel('Natural Frequency (Hz)')
legend('reference','initial','updated','Location','southeast')
set(gca,'fontsize',11)

%Defining the height and labels of a column nodes for plotting the mode
%shapes
z=[0 4.49 8.48 10.47 12.46 14.46 16.45 18.4469 20.44 22.44 24.43 26.42 ...
   28.42 30.41 32.41 34.4 36.4 38.39 40.38 42.38 43.38 45 45.5 46.5 ...
   47.5 48.5 49.5 50.5 51.5 52.5 53.5 54.5 55.5 56.5 57.5 58.5 59.5 60];
column_nodes=[4 5 16 27 34 41 48 55 62 69 76 83 90 97 104 111 118 120 ...
   122 124 126 1 131 133 135 137 139 141 143 145 147 149 151 153 155 ...
   157 159 128];

%Plotting the mode shapes of the column defined

```

```

for i=modSel % the x mode shape
    figure
    plot(phi_ref(column_nodes,i),z,'r-o')
    hold on
    plot(phi_0(column_nodes,i),z,'b:o')
    plot(phi_upd(column_nodes,i),z,'k--*')
    hold off
    xlim([-inf,inf])
    ylim([-inf,inf])
    ylabel('Height')
    xlabel('Displacement "x"')
    title(['Mode ' int2str(i)])
    legend('reference','initial','updated','Location','southeast')
end

for i=modSel % the y mode shape
    figure
    plot(phi_ref(504.+column_nodes,i),z,'r-o')
    hold on
    plot(phi_0(504.+column_nodes,i),z,'b:o')
    plot(phi_upd(504.+column_nodes,i),z,'k--*')
    hold off
    xlim([-inf,inf])
    ylim([-inf,inf])
    ylabel('Height (m)')
    xlabel('Displacement "y"')
    title(['Mode ' int2str(i)])
    legend('reference','initial','updated','Location','southeast')
end

```

## Computing displacements

Computing displacements due to a set of loads applied at the top of the tower

```

file_damaged_disp='G:\ING CIVIL\Tesis 2016\Matlab files\Tower model 7 E calibrated\Damaged
for displacement\Najasa Modal continua.sdb';
file_updated_disp='G:\ING CIVIL\Tesis 2016\Matlab files\Tower model 7 E calibrated\Updated
for displacement\Najasa Modal continua.sdb';

load data_mdof_ern x_ref

[dis_damaged] = eigfem_mdof_torre_for_displacements(x_ref,file_damaged_disp);
[dis_initial] = eigfem_mdof_torre_for_displacements(x_0,file_updated_disp);
[dis_updated] = eigfem_mdof_torre_for_displacements(x_upd,file_updated_disp);

```

## Plotting displacements

```

figure %displacements in x
plot(dis_damaged(column_nodes,1),z,'r-o')
hold on
plot(dis_initial(column_nodes,1),z,'b:o')
plot(dis_updated(column_nodes,1),z,'k--*')
hold off

```

```

xlim([-inf,inf])
ylim([-inf,inf])
ylabel('Height')
xlabel('Displacement "x" (mm)')
title('Displacements in "x"')
legend('reference','initial','updated','Location','southeast')

figure %displacements in y
plot(dis_damaged(504.+column_nodes,1),z,'r-o')
hold on
plot(dis_initial(504.+column_nodes,1),z,'b:o')
plot(dis_updated(504.+column_nodes,1),z,'k--*')
hold off
xlim([-inf,inf])
ylim([-inf,inf])
ylabel('Height')
xlabel('Displacement "y" (mm)')
title('Displacements in "y"')
legend('reference','initial','updated','Location','southeast')

```

## **Saving the optimization results and figures**

By saving the optimization results can be ensured that MATLAB can be shut down and in other session these results can be loaded.

```

save all_result_optimization x* modSel o* p* x_ref z column_nodes T* r* d* L*
savefig((1:((max(modSel)*2)+4)),'All updating figures')
toc

```

## Programación del script (data\_mdof\_torre):

### DEFINE REFERENCE MDOF MODEL (REALITY)

- JULY 2017
- Kristof Maes
- Modified by Alejandro Fuentes Diaz
- Units: kN, m, s

```
close all; clc
```

### Calculating modal parameters using OAPI SAP 2000 MATLAB

1. Define the reference parameters.
2. Loading the damaged model path.
3. Calculating the reference Eigen frequencies, DOF and mode shapes.
4. Define the number of updating variables.
5. Define the mode to take account.
6. Rearranging the Eigen frequencies and mode shapes for the defined modes.
7. Saving data to be used for other scripts.

```
x_ref=[2.1*10^8 2.05*10^8 2.0*10^8 1.98*10^8 1.99*10^8 2.1*10^8 1.96*10^8];

load upd_param_mdof_ern file_damaged
[phi_ref,omega_ref,DOF_ref]=eigfem_mdof_torre(x_ref,file_damaged);
nparameters=7; % number of parameters to be updated

modSel=1:5;
phi_ref=phi_ref(:,modSel);
omega_ref=omega_ref(modSel);

% Save modal data for updating
save data_mdof_ern nparameters DOF_ref phi_ref omega_ref x_ref
```

## Programación del script (upd\_mdof\_torre):

### MODEL UPDATING FOR MDOF SYSTEM

- JULY 2017
- Kristof Maes
- Modified by Alejandro Fuentes Diaz

#### Defining the updating lsqnonlin optimization

1. Defining optimization function.
2. Define objective function and step size tolerances and the levenberg-marquardt method initial damping.
3. Specifying optimization options.
4. Calling the lsqnonlin optimization.

```
function [T_upd]=upd_mdof_torre(T_0)

TolFun=1e-16;      % Tolerance on objective function
TolX=1e-8;        % Tolerance on parameters
initdamp = 1e2;    %Initial damping of serch direction lamda parameter

% Specify options
options=optimoptions(@lsqnonlin,'Algorithm','levenberg-
marquardt','ScaleProblem','Jacobian','InitDamping',initdamp,'Display','iter-detailed',...
    'Jacobian','off','TolFun',TolFun,'TolX',TolX,'OutputFcn',@outfun);

load data_mdof_ern nparameters

% Nonlinear optimization
[T_upd]=...
    lsqnonlin(@objfun,T_0,-inf*ones(1,nparameters)...
    ,+inf*ones(1,nparameters),options);
```

#### Defining the objective function.

Objective function where Eigen frequencies and mode shape residuals are computed.

```
function [res]=objfun(T)

load upd_param_mdof_ern X_0 file_upd
Tvec = T .* X_0;

% Load experimental modal data
load data_mdof_ern DOF_ref phi_ref omega_ref

model = 3; % 1 = analytical, 2 = ANSYS, 3 = OAPI SAP2000 MatLab

switch model
case 1 % ANALYTICAL MODEL
    mvec=m_ref*ones(nfloor,1);
    [phi,omega]=eigfem_mdof(mvec,kvec);
```

```

case 2 % ANSYS MODEL
    ansys_dir='..\..\ansys\exercise2';
    [phi,omega]=getansys(kvec,ansys_dir);

case 3 % OAPI SAP2000 MatLab MODEL
    [phi,omega,DOF] = eigfem_mdof_torre(Tvec,file_upd);
end

DOF_sensors=[ [62,97,122,128,203,228,245,263,27,28,141,140,1,2,151,150,...
    ]+0.01, [63,98,123,129,178,253,160,258]+0.02].';

% DOF selection matrix
L_ref=selectdof(DOF_ref,DOF_sensors);
L=selectdof(DOF,DOF_sensors);

% Mode matching
MAC=mac(L_ref*phi_ref,L*phi); % Calculate MAC-matrix
threshold=0.7; % Threshold for mode matching
mode_ref_sel=(1:size(MAC,1))'; % No. of measured modes
mode_sel=zeros(size(MAC,1),1); % No. of corresponding calculated modes
for ind1=1:length(mode_ref_sel)
    [max_,ind_max]=max(MAC(ind1,:));
    if max_>=threshold
        mode_sel(ind1)=ind_max; % Match found
    else
        disp(['No mode match found for measured mode ' int2str(ind1) '.'])
    end
end
mode_ref_sel(mode_sel==0,:)=[];
mode_sel(mode_sel==0,:)=[];
if length(unique(mode_sel))~=length(mode_sel)
    disp('At least one calculated mode is matched to multiple measured modes.')
end
phi_ref=phi_ref(:,mode_ref_sel); omega_ref=omega_ref(mode_ref_sel);
phi=phi(:,mode_sel); omega=omega(mode_sel);
nm=length(omega); % number of modes

% Calculate scaling factors mode shapes
gamma=zeros(nm,1);
for ind1=1:nm
    gamma(ind1)=((L_ref*phi_ref(:,ind1)).'*L*phi(:,ind1))/norm(L_ref*phi_ref(:,ind1),2)^2;
end

% Calculate eigenvalues
lambda=omega;
lambda_ref=omega_ref;

% Calculate residuals
load upd_param_mdof_ern alpha beta
res=(sqrt(alpha/2).*((lambda-lambda_ref)./lambda_ref).^2);
for ind1=1:nm
    wf=sqrt(beta/2)/(norm(gamma(ind1)*L_ref*phi_ref(:,ind1),2)); % weighting factor
    res=[res; (wf.*(L*phi(:,ind1)-(gamma(ind1)*L_ref*phi_ref(:,ind1))))).^2];
end

```



## Defining an output function

Output function that is called on every lsqnonlin iteration.

```
function stop=outfun(T,optimValues,state)
stop=false;
% Save intermediate results iterations
if strcmp(state,'init')
    T_iter=T;
    resnorm_iter=optimValues.resnorm;
    save results_mdof T_iter resnorm_iter
elseif strcmp(state,'iter')
    load results_mdof T_iter resnorm_iter
    T_iter=[T_iter;T];
    resnorm_iter=[resnorm_iter;optimValues.resnorm];
    save results_mdof T_iter resnorm_iter
end
```

## Programación del script (eigfem\_mdof\_torre):

### OAPI FUNCTION BETWEEN MATLAB AND SAP2000

The functions that allow MATLAB to interact with SAP2000 are written in the script.

```
function [phi,omega,DOF] = eigfem_mdof_torre(x_moduli,file)

feature('COM_SafeArraySingleDim', 1);
feature('COM_PassSafeArrayByRef', 1);
SapObject = actxserver('sap2000v16.SapObject');
```

### Open SAP

```
SapObject.ApplicationStart;
```

### Hiding SAP2000

```
SapObject.Hide;
```

### Opening the model in the path named file

```
ret = SapObject.SapModel.File.OpenFile(file);
```

### Creating the SAP object model

```
SapModel = SapObject.SapModel;
```

### Unblocking the model

```
ret = SapModel.SetModelIsLocked(Copyright © Computers and Structures);
```

### Creating the analysis model

```
ret = SapModel.Analyze.CreateAnalysisModel;
```

### Defining the material properties

Creating materials called H1, H2 and H3.

```
MATERIAL_STEEL = 1; % steel = 1, concrete = 2

ret = SapModel.PropMaterial.SetMaterial('H1', MATERIAL_STEEL);
```

```
ret = SapModel.PropMaterial.SetMaterial('H2', MATERIAL_STEEL);
ret = SapModel.PropMaterial.SetMaterial('H3', MATERIAL_STEEL);
```

### Assigning the isotropic material properties

1. Create a x variable that correspond to the property that is assigned to x\_moduli in the specified vector column.
2. Assign the property of the created variable to the material created before.

```
x1 = x_moduli(Copyright @ Computers and Structures); x2 = x_moduli(Copyright @ Computers and Structures); x3 = x_moduli(Copyright @ Computers and Structures);
ret = SapModel.PropMaterial.SetMPIsotropic('H1', x1,0.3,0.000007);
ret = SapModel.PropMaterial.SetMPIsotropic('H2', x2,0.3,0.000007);
ret = SapModel.PropMaterial.SetMPIsotropic('H3', x3,0.3,0.000007);
```

### Assigning properties to the SAP model.

Three ways of assigning materials or springs or inertia reduction factors are presented in this section.

Case 1: Overwrite material properties.

```
% Reading the element labels contained on the Excel files and assigning
% them to variables named groups.
Object = 0;
group=xlsread('tramo inferior.xlsx');
group1=xlsread('tramo intermedio.xlsx');
group2=xlsread('tramo superior.xlsx');

% Assigning the material created to the elements on the groups
for iElem=group.'
    ret = SapModel.FrameObj.SetMaterialOverwrite(int2str(iElem),'H1',Object);
end

for iElem=group1.'
    ret = SapModel.FrameObj.SetMaterialOverwrite(int2str(iElem),'H2',Object);
end

for iElem=group2.'
    ret = SapModel.FrameObj.SetMaterialOverwrite(int2str(iElem),'H3',Object);
end

% Assigning springs to the ponts 163 3 266 4. These point correspond to
% the structure supports.

Object = 0;
IsLocalCSys = 1;
Replace = 1;
```

Case 2: Assign springs to points.

```
% Obtaining the spring values from the x_moduli vector.
x7 = x_moduli(Copyright © Computers and Structures); x8 = x_moduli(Copyright © Computers
and Structures);
k = [x8; x8; x7; 0; 0; 0];

% Assigning the springs to the points.
ret = SapModel.PointObj.SetSpring('3', k, Object, IsLocalCSys, Replace);
ret = SapModel.PointObj.SetSpring('4', k, Object, IsLocalCSys, Replace);
ret = SapModel.PointObj.SetSpring('163', k, Object, IsLocalCSys, Replace);
ret = SapModel.PointObj.SetSpring('266', k, Object, IsLocalCSys, Replace);
```

Case 3: Assign moment of inertia reduction factors to elements in the SAP model.

```
x4=x_moduli(4); x5=x_moduli(5); x6=x_moduli(6);
I_mod1 = [1; 1; 1; 1;1 ;x4 ;1; 1];
I_mod2 = [1; 1; 1; 1;1 ;x5 ;1; 1];
I_mod3 = [1; 1; 1; 1;1 ;x6 ;1; 1];

Object=0;

for iElem=group.'
    ret = SapObject.SapModel.FrameObj.SetModifiers(int2str(iElem), I_mod1,Object);
end

for iElem=group1.'
    ret = SapObject.SapModel.FrameObj.SetModifiers(int2str(iElem), I_mod2,Object);
end

for iElem=group2.'
    ret = SapObject.SapModel.FrameObj.SetModifiers(int2str(iElem), I_mod3,Object);
end
```

## Running analysis

```
ret = SapObject.SapModel.Analyze.RunAnalysis();
```

## Deselecting all the cases to execute

```
ret = SapModel.Results.Setup.DeselectAllCasesAndCombosForOutput;
```

## Selecting the case or combination for results

```
ret = SapModel.Results.Setup.SetCaseSelectedForOutput('MODAL');
```

## Obtaining the modal period

```
NumberResults = 0;
Obj = cellstr(' ');
```

```

Elm = cellstr(' ');
LoadCase = cellstr('MODAL');
StepType = cellstr('Mode');
StepNum = zeros(1,1,'double');
Period = zeros(1,1,'double');
Frequency = zeros(1,1,'double');
CircFreq = zeros(1,1,'double');
EigenValue = zeros(1,1,'double');

[ret, NumberResults, LoadCase, StepType, StepNum, Period, Frequency, CircFreq, EigenValue]
= ...
SapModel.Results.ModalPeriod(NumberResults, LoadCase, StepType, StepNum, Period, Frequency,
CircFreq, EigenValue);

omega = CircFreq.'; % Natural frequency [rad/s]

```

## Obtaining the modal shapes (modal coordinates)

```

NumberResults = 0;
Obj = cellstr(' ');
Elm = cellstr(' ');
ACase = cellstr('MODAL');
StepType = cellstr('Mode');
StepNum = zeros(1,1,'double');
U1 = zeros(1,1,'double');
U2 = zeros(1,1,'double');
U3 = zeros(1,1,'double');
R1 = zeros(1,1,'double');
R2 = zeros(1,1,'double');
R3 = zeros(1,1,'double');
GroupElm = 2;
[ret, NumberResults, Obj, Elm, ACase, StepType, StepNum, U1, U2, U3, R1, R2, R3] = ...
SapModel.Results.ModeShape('ALL', GroupElm, NumberResults, Obj, Elm, ACase, StepType,
StepNum, U1, U2, U3, R1, R2, R3);

```

## Arranging the results and calculating the DOF.

```

phix=U1;
phiy=U2;
phiz=U3;

n_m=length(omega);           % number of modes
n_node=length(phix)/n_m;     % number of nodes

phix_=zeros(n_node,n_m);     % intermediate matrix
phiy_=zeros(n_node,n_m);     % intermediate matrix
phiz_=zeros(n_node,n_m);     % intermediate matrix

for ind1=1:n_m
    phix(:,ind1)= phix(:,1:n_node);
    phix(:,1:n_node)=[];
    phiy(:,ind1)= phiy(:,1:n_node);
    phiy(:,1:n_node)=[];

```

```

        phiz_(:,ind1)= phiz(:,1:n_node);
        phiz(:,1:n_node)=[];
    end
    phi=[phix_;phiy_;phiz_];
    nodes=zeros(n_node,1);
    for ind1=1:n_node
        nodes(ind1)=str2double(Obj{ind1});
    end
    DOF=[nodes+0.01;... % 0.01 = x
        nodes+0.02;... % 0.02 = y
        nodes+0.03]; % 0.03 = z

```

### Shutting down the SAP application.

```

ret = SapObject.ApplicationExit(false());
SapModel = 0;
SapObject = 0;

```

## Programación del script (sens\_anal\_torre):

### Sensitivity analysis

```
tic  
close all, clear all, clc
```

### Reference model

```
E_ref=0.90*2.1e8; % damaged
```

### Healthy structure

```
E_0=2.1e8
```

### Number of updating parameters

```
nparameters=1  
  
save data_mdof_ern nparameters
```

### Calculate modal parameters

```
[phi_ref,omega_ref,DOF_ref]=eigfem_mdof_torre(E_ref,file);  
[phi_0,omega_0,DOF_0]=eigfem_mdof_torre(E_0,file);  
modSel=1:5;  
phi_ref=phi_ref(:,modSel);  
omega_ref=omega_ref(modSel);  
phi_0=phi_0(:,modSel);  
omega_0=omega_0(modSel);
```

### Choose sensor positions

```
DOF_sensors=[[62,97,122,128,203,228,245,263]+0.01,[63,98,123,129]+0.02].';
```

### Selection matrices L

```
L_ref=selectdof(DOF_ref,DOF_sensors);  
L_0=selectdof(DOF_0,DOF_sensors);
```

### Calculate AUTO-MAC

```
modSel=[1:5];  
autoMAC=mac(L_0*phi_ref(:,modSel),L_0*phi_ref(:,modSel))
```

## Presenting results

```
thisFig=figure;
gcfPos=get(gcf,'position');
bar3color(autoMAC)
view([0 90])
axis equal
hold on
xlim([0.4 size(autoMAC,1)+0.6])
ylim([0.4 size(autoMAC,1)+0.6])
set(gca,'clim',[0 1])
colorbar;
box on
colormap(hot))

labels=[];
for ind1=1:length(modSel)
    labels=strvcat(labels,int2str(modSel(ind1)))
end
set(gca,'xtick',1:length(modSel),'XTickLabel',labels)
set(gca,'ytick',1:length(modSel),'YTickLabel',labels)

toc
```