

**APLICACIÓN COMPUTACIONAL PARA CÁLCULO DE VELOCIDADES
CRÍTICAS, MODOS DE VIBRACIÓN Y ESFUERZOS EN SISTEMAS
ROTATIVOS**

JOHN JAIRO BUENO ORTIZ

DIEGO ARMANDO RINCÓN ARENAS

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍA MECÁNICA**

2013

**APLICACIÓN COMPUTACIONAL PARA CÁLCULO DE VELOCIDADES
CRÍTICAS, MODOS DE VIBRACIÓN Y ESFUERZOS EN SISTEMAS
ROTATIVOS**

JOHN JAIRO BUENO ORTIZ

DIEGO ARMANDO RINCÓN ARENAS

**TRABAJO PARA OPTAR POR EL TÍTULO DE
INGENIERO MECÁNICO**

DIRECTORES

Jabid Quiroga

MAGISTER, INGENIERO MECÁNICO

Alfonso García Castro

MAGISTER, INGENIERO MECÁNICO

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍA MECÁNICA**

2013

A Luz Marina mi madre por su inquebrantable fe en la superación de sus hijos, a mi padre y a mis dos hermanos.

JOHN JAIRO BUENO ORTIZ

A los trabajadores del mundo y en especial a los de la UIS pues son ellos los que
con su esfuerzo silencioso hacen todo funcionar.

Para todos, todo.

DIEGO ARMANDO RINCÓN ARENAS.

Muchas han sido las personas vinculadas en este proyecto, pero debo agradecer al profesor Alfonso García castro y a mis compañeros, Mauricio barrios, Mardel Obredor y Diego Gómez, quienes siempre nos brindaron su apoyo.

JOHN JAIRO BUENO ORTIZ

A mi familia que a pesar de todo siempre está junto a mí.

DIEGO ARMANDO RINCÓN ARENAS

CONTENIDO

	pág.
INTRODUCCIÓN	18
1. CONCEPTOS BÁSICOS.....	21
1.1 FENÓMENO VIBRATORIO	21
1.1.1 La excitación.....	21
1.1.2 El sistema mecánico	22
1.1.3 La respuesta	23
1.2 VIBRACIÓN	23
1.2.1 Vibración libre.....	24
1.2.2 Vibración forzada	25
1.3 RESONANCIA MECÁNICA	28
1.4 VELOCIDAD CRÍTICA EN EJES.....	29
1.4.1 Modos de vibración.....	30
2. MARCO TEÓRICO.....	31
2.1 MÉTODO PARA HALLAR VELOCIDADES CRÍTICAS EN EJES	31
2.1.1 Matriz de transferencia	31
3.1.1.1 Matriz de transferencia para apoyo simple-simple.....	37
3.1.1.2 Matriz de transferencia para apoyo empotrado-libre.....	39
2.2 USANDO EL MÉTODO PARA HALLAR LOS MODOS DE VIBRACIÓN	40
2.3 DE LA DISCRETIZACIÓN	41
3. PROGRAMACIÓN DEL MÉTODO.....	42
3.1 DIAGRAMA DE FLUJO	42
3.2 INTERFAZ GRÁFICA	46

3.2.1 Datos geométricos (Zona 1)	46
3.2.2 Condiciones de frontera (Zona 2)	50
3.2.3 Botón calcular (Zona 3).....	51
3.2.4 Selección de modos a graficar (Zona 4).....	51
4. PRUEBAS EXPERIMENTALES	53
4.1 PRUEBAS REALIZADAS	53
4.2 PRUEBA DE IMPACTO	53
4.2.1 Montaje del experimento	53
4.2.2 Toma de datos	55
5.2.3 Resultados obtenidos.....	57
4.3 PRUEBA POR VARIACIÓN DE VELOCIDAD.....	58
4.3.1 Montaje del experimento.....	58
4.3.2 Toma de datos	59
4.3.3 Resultados obtenidos	62
4.4 VERIFICACIÓN DE LOS DATOS DEL SOFTWARE	62
CONCLUSIONES	65
BIBLIOGRAFÍA	66
ANEXOS.....	67

LISTA DE TABLAS

	pág.
Tabla 1.Especificaciones del módulo de toma de datos.....	55
Tabla 2.Especificaciones del chasis.....	56
Tabla 3.Especificaciones del martillo de impacto.....	56
Tabla 4.Especificaciones del acelerómetro.....	56
Tabla 5 Resumen de resultados obtenidos en la prueba de impacto.....	57
Tabla 6. Especificaciones del acelerómetro.....	59
Tabla 7. Resultado prueba de variación de velocidad.....	62
Tabla 8. Resumen de resultados de los software.....	63
Tabla 9. Resumen de resultados experimentales.....	63

LISTA DE FIGURAS

	pág.
Figura 1. Partes de un fenómeno vibratorio	21
Figura 2. Sistemas mecánicos típicos	22
Figura 3. Diagrama de cuerpo libre de un sistema en vibración libre.....	24
Figura 4. Diagrama de cuerpo libre de un sistema en vibración forzada	27
Figura 5. Caída del puente Tacoma Narrow.....	29
Figura 6. Modos de vibración en una cuerda	30
Figura 7. Discretización clásica.....	32
Figura 8. Diagrama de cuerpo libre de la i-ésima estación	32
Figura 9. Diagrama de cuerpo libre alterado de la i-ésima estación	34
Figura 10. Diagrama de cuerpo libre del i-ésimo campo... ..	35
Figura 11. Modelo discreto de un sistema con apoyo simple-simple	37
Figura 12. Modelo discreto de un sistema simple-simple	39
Figura 13. Diagrama de flujo del código programado	42
Figura 14. Zonas de interfaz gráfica.....	46
Figura 15. Detalle zona 1 (datos geométricos).....	47
Figura 16. Detalle zona 1 (tabla generada).....	48
Figura 17. Detalle del dx.....	49
Figura 18. Cuadro de alarma de error en el ingreso de datos.....	50
Figura 19. Visualización de las condiciones de frontera.....	50
Figura 20. Botón calcular y su despliegue	51
Figura 21. Salida gráfica de la deformación, momento y cortante	52
Figura 22. Montaje sobre cuerdas	54
Figura 23. Montaje del experimento	54
Figura 24. Elementos de adquisición de datos	57
Figura 25. Motor y variador de velocidad	58
Figura 26. Puntos de toma de datos del experimento.	59
Figura 27. foto-tacómetro utilizado durante las pruebas.	60

Figura 28. Toma de datos.....61
Figura 29. Ejemplo de datos adquiridos en el experimento.....61

LISTA DE ANEXOS

	Pág.
Anexo A.Ejemplo de calculo de velocidad crítica de un eje a través del método de matriz de transferencia.....	67
ANEXO B.Código en matlab.....	72
ANEXO C.Reporte del software ansys	88

RESUMEN

TITULO: APLICACIÓN COMPUTACIONAL PARA CÁLCULO DE VELOCIDADES CRÍTICAS, MODOS DE VIBRACIÓN Y ESFUERZOS EN SISTEMAS ROTATIVOS

AUTORES: JOHN JAIRO BUENO ORTIZ

DIEGO ARMANDO RINCÓN ARENAS**

PALABRAS CLAVES: frecuencia natural, matriz de transferencia, modos de vibración.

DESCRIPCIÓN:

El presente trabajo es el desarrollo de la metodología de cálculo ideada en principio por Myklestad y Prohl ⁹ para determinar las frecuencias naturales de alas de avión y alabes de turbinas, este procedimiento puede ser extrapolado a sistemas de ejes rotativos de elementos mecánicos en condiciones de flexión y llevarse al campo informático para ser programado de manera que se logre reducir significativamente el tiempo de determinación de este fenómeno, esto con el fin de que los ingenieros calculistas puedan usar su tiempo en actividades más productivas tanto mental como económicamente.

En un primer paso se explica el modelo teórico de discretización del elemento real a calcular, posteriormente se pasa al tratamiento matemático que permite a través de una matriz de transferencia asemejar el modelo matemático al físico, luego se obtiene los valores de velocidad crítica de acuerdo a las condiciones de apoyo elegidas y por último con este mismo método pero agregando asunciones lógicas se puede señalar la manera como se comporta el elemento en términos de deformación, momento y cortante.

Tras programar este procedimiento se realizan pruebas de impacto y de medición de vibraciones a través de distintas velocidades de rotación de manera que se logra verificar los resultados obtenidos del software.

*Trabajo de grado

**Facultad de Ingenierías Físico-Mecánicas, Escuela de Ingeniería Mecánica. Directores: Jabid Quiroga. Alfonso García Castro.

ABSTRACT

TITLE: COMPUTATIONAL APLICATION TO CALCULATE CRITICAL SPEEDS, MODAL VIBRATIONS AND STRESS IN ROTARY SYSTEMS

AUTHORS: JOHN JAIRO BUENO ORTIZ

DIEGO ARMANDO RINCÓN ARENAS**

KEYWORDS: Natural frequency, transfer matrix, modal vibrations.

DESCRIPTION:

The objective of this project is the development of a methodology designed by Myklestad and Prohl⁹ to determinate the natural frequencies of airplane wings and turbine blades, this method can be extrapolated to systems with rotary shafts in mechanical systems with bending that was taken to the field of computation to be programmed in a way that can decrease significantly the time in the determination of this phenomenon, the goal is that the engineer use his time in more productive activities for his brain and his pocket.

In a first step the theoretical model of distributed parameter system is discussed for a real system to calculate, later it passes to the mathematical treatment that through a transfer matrix resembles the mathematical and physical model, then the values of critical speed are obtained according to the supports chosen and finally with the same method but adding some logical assumptions is possible to indicate the behavior of deformation, moment and shear.

After programming this procedure some impact test and measurement of vibration through different rotation speeds are done in a way that is verified the answers from the software.

*Undergraduated degree work

**Physical-Mechanical Engineering Department, School of mechanical engineering. Directors: Jaid Quiroga. Alfonso García Castro.

INTRODUCCIÓN

El desarrollo de las tecnologías por parte del hombre ha permitido que este supere poco a poco los límites físicos impuestos por las leyes fundamentales del universo, de manera que le es necesario profundizar en el conocimiento de los fenómenos naturales para lograr entenderlos de manera consciente y no como resultado de la simple experiencia, esta comprensión es especialmente importante cuando se puede causar daños tanto a los elementos materiales como a las personas, un ejemplo típico de este hecho es el avance en la determinación de las causas y efectos de la fatiga sobre las máquinas y sus componentes como respuesta humana a un fenómeno que siempre ha existido pero que el hombre no conoció hasta romper ciertas limitaciones naturales en la construcción de elementos mecánicos.

Teniendo en cuenta estos elementos se decidió ahondar en un campo bastante explorado pero poco divulgado durante el pregrado de los estudiantes de ingeniería, referente al fenómeno de la resonancia mecánica de los elementos cuando se acercan a su frecuencia natural.

Para ello se ha investigado amplia bibliografía y se han realizado distintas consultas con el fin de poner en práctica un método que permita determinar de manera ágil las frecuencias naturales de quizás el elemento más común de la ingeniería mecánica como lo son los ejes, de allí se determinó que dada la capacidad computacional moderna el método de Myklestad-Prohl era el más eficiente para solucionar el problema determinado durante la etapa de proposición del proyecto.

Así pues se logra obtener un producto (software) que permite elevar el conocimiento de quienes obtén por seguir este campo de trabajo, además se consigue disminuir el tiempo de diseño de un elemento fundamental de la vida de cualquier calculista, así como entender de manera más cercana posibles causas

del excesivo mantenimiento de un elemento rotativo, permitiéndole al ingeniero ejercer labores que sean más rentables tanto mental como económicamente hablando.

Durante la lectura de este texto se encuentra en principio un capítulo correspondiente a conceptos básicos, añadidos con el fin de que una persona con escasos conocimientos de ingeniería logre entender varios términos utilizados posteriormente, además también le sirve a aquellos que ya han pasado por cátedras de vibraciones para tener nociones de lo aquí trabajado. A continuación en el capítulo referente al marco teórico se explica con el mayor detalle posible el método escogido para solucionar el problema determinado en los objetivos. Ya entendido el modelo físico y matemático asumido se pasa al siguiente capítulo donde se indica la manera como se programó la aplicación computacional y se da detalles de como el usuario debe hacerla funcionar.

Superada la sección teórica pasamos a la parte práctica en el capítulo quinto, donde se detalla las pruebas realizadas, tres para el caso, para verificar los datos obtenidos de la aplicación y la precisión obtenida, por último y tras analizar el cumplimiento de los objetivos propuestos se encuentran las conclusiones extraídas del trabajo realizado.

OBJETIVOS

1.1 OBJETIVO GENERAL

Contribuir con la misión de la escuela de ingeniería mecánica de la UIS en lo correspondiente a la construcción, aplicación y divulgación del conocimiento al diseñar una aplicación para el cálculo de velocidades críticas y modos de vibración de ejes, y verificar las respuestas a través del montaje en los bancos existentes en el laboratorio de vibraciones mecánicas.

1.2 OBJETIVOS ESPECÍFICOS

- Implementar un algoritmo en ambiente Matlab que permita el cálculo de las velocidades críticas, modos de vibración, diagramas de momento y cortante, con configuración geométrica, distribución de apoyos y dimensiones definidas por el usuario, de acuerdo con el método de Myklestad-Prohl, para flexión en sistemas rotativos.
- Verificar las velocidades críticas y modos de vibración obtenidos de la aplicación usando un prototipo de máquina rotativa del laboratorio de vibraciones.
- Elaborar una guía de uso de la aplicación desarrollada.
- Documentar los resultados del proyecto en un artículo en formato científico.

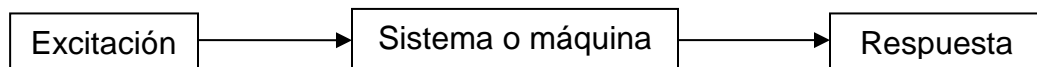
1. CONCEPTOS BÁSICOS

1.1 FENÓMENO VIBRATORIO

Se considera que un sistema vibra cuando él o una de sus propiedades oscila alrededor de una posición de equilibrio.

Un fenómeno vibratorio tiene 3 elementos principales que son el sistema mecánico, la excitación y la respuesta (vibración). En la figura 1 los elementos típicos de un sistema vibratorio.

Figura 1. Partes de un fenómeno vibratorio



Fuente: Los autores.

1.1.1 La excitación

Es aquella acción que produce la vibración del sistema y puede representarse como una fuerza variable que mueve el sistema alrededor de su posición de equilibrio. Esta puede suceder por causas externas como impactos o sismos, o por causas internas como desequilibrio de piezas rotativas o como parte de la función misional de la máquina.

La excitación puede clasificarse así⁴:

- Condiciones iniciales.
- Excitación armónica.

- Excitación periódica no armónica.
- Excitación no periódica (determinístico).
- Excitación aleatoria.

1.1.2 El sistema mecánico

Se considera a un conjunto de elementos interconectados de manera que puedan cumplir una función predispuesta (trabajo)¹⁰, ejemplo de esto es una máquina, una estructura metálica, etc. En la figura 2 se encuentran dos ejemplos típicos de un sistema mecánico, el sistema de amortiguamiento de un carro y el estadio nido de pájaro en China.

Figura 2. Sistemas mecánicos típicos



Fuente:<http://nissansierrapvr.blogspot.com/2012/02/conozca-la-importancia-de-los.html>.<http://www.fbbva.es/TLFU/tlfu/micros/esp/atlas/asiapacifico/galeria.jsp#5>.

Es necesario conocer las partes de un sistema y su relacionamiento de manera que se pueda desarrollar un modelo físico-matemático que tenga en cuenta la geometría, las restricciones, las fuerzas, las suposiciones, etc., para que se puedan establecer las ecuaciones que gobiernan el movimiento de forma que se

logre determinar cómo se comporta el sistema ante distintas alteraciones de sus condiciones de entrada⁴.

1.1.3 La respuesta

Es el resultado de la aplicación de la excitación sobre el sistema y las fuerzas recuperadoras del mismo. Puede clasificarse de acuerdo al momento en que se aplica la excitación así⁴:

- Vibración libre: Sucede cuando la excitación solo se aplica en el momento inicial del movimiento.
- Vibración forzada: Sucede cuando la excitación se aplica durante todo el transcurso del movimiento.
- Vibración paramétrica: Sucede cuando la vibración es causada por la variación de un parámetro del sistema, tales como la inercia o la rigidez.

1.2 VIBRACIÓN

Es la oscilación de un sistema o una de sus propiedades alrededor de una posición de equilibrio o referencia⁴, sus características vienen determinadas por los siguientes elementos:

Amplitud: Representa la intensidad del movimiento.

Frecuencia: Es el número de veces que el movimiento se repite por unidad de tiempo, indica el ritmo en que sucede el fenómeno.

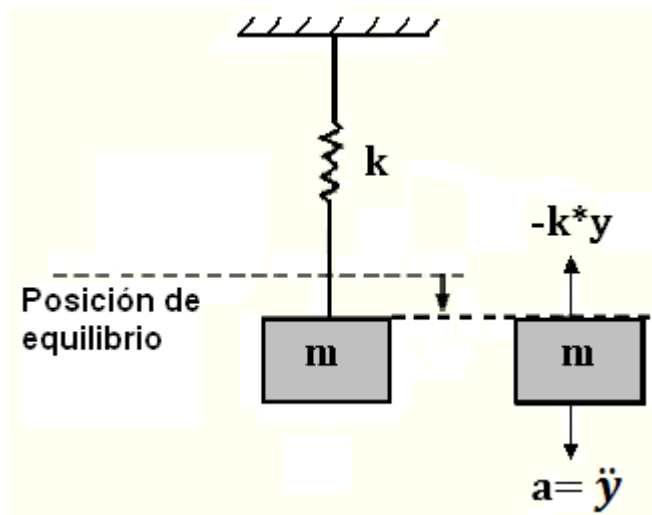
Fase: Magnitud que indica la posición angular o temporal respecto a una posición de referencia.

1.2.1 Vibración libre

Como ya hemos dicho esta se presenta cuando damos condiciones iniciales al sistema que lo saquen de su condición de equilibrio, y la excitación no se aplica durante el movimiento. Para hacer el modelamiento necesitamos conocer la inercia de rigidez y amortiguamiento, de manera que se logre hallar la frecuencia natural que establezca la ecuación que describa el movimiento vibratorio.

Para realizar el modelamiento nos ayudaremos de la segunda ley de newton y un ejemplo clásico como se muestra en la figura 3 continuación.

Figura 3. Diagrama de cuerpo libre de un sistema en vibración libre



Fuente: Los autores.

$$\sum F = \text{masa} * \text{aceleración}$$

$$-k * y = m * \ddot{y}$$

$$m * \ddot{y} + k * y = 0$$

$$\ddot{y} + \frac{k}{m} * y = 0 \quad (1)$$

Si se supone que la respuesta es periódica y armónica, se puede expresar **y** de la forma.

$$y = A * \cos(w * t) \quad (2)$$

$$\dot{y} = -A * w * \dot{\text{sen}}(w * t) \quad (3)$$

$$\ddot{y} = -A * w^2 * \cos(w * t) \quad (4)$$

Si se reemplaza las ecuaciones (2) y (4) en (1) se obtiene:

$$-m * A * w^2 * \cos(w * t) + k * A * \cos(w * t) = 0 \quad (5)$$

Al eliminar las componentes **A** y **cos(w*t)** obtenemos la siguiente solución:

$$w = \sqrt{\frac{k}{m}} \quad (6)$$

La ecuación (6) se denomina la frecuencia natural del sistema, finalmente la ecuación que modela el movimiento del sistema es¹⁰:

$$Y = A * \cos\left(\sqrt{\frac{k}{m}} * t\right) \quad (7)$$

1.2.2 Vibración forzada

Esta se presenta cuando la excitación de aplica durante todo el transcurso del movimiento vibratorio, este caso es común en máquinas rotativas cuyo centro de geométrico no coincide con el centro de masa.

En general para estos casos al comienzo de la oscilación hay una mezcla entre la frecuencia natural del sistema y la forzada, posteriormente la segunda frecuencia sobrevive de manera que se presenta un estado estable en la operación de la máquina¹⁰. A continuación se hará un modelamiento clásico con el fin de mostrar la razón matemática de lo expuesto a partir de las leyes de newton.

$$F_o * \cos(wf * t) - k * x = m * a \quad (8)$$

Haciendo una manipulación adecuada de los términos y teniendo en cuenta que la aceleración es la segunda derivada de la posición se obtiene (10).

$$\ddot{x} = a \quad (9)$$

$$\ddot{x} + \frac{k}{m} * x = \frac{F_o}{m} * \cos(wf * t) \quad (10)$$

Suponiendo respuesta periódica y armónica se puede expresar **x** de la forma.

$$x = A * \cos(wf * t) \quad (11)$$

$$\dot{x} = -A * wf * \text{sen}(w * t) \quad (12)$$

$$\ddot{x} = -A * wf^2 * \cos(w * t) \quad (13)$$

Si se reemplaza (11) y (13) en (10) se obtiene:

$$-A * wf^2 * \cos(wf * t) + \frac{k}{m} * A * \cos(wf * t) = \frac{F_o}{m} * \cos(wf * t) \quad (14)$$

Al eliminar la componente **cos(w*t)**, se reemplaza con la ecuación (6) y se despeja la amplitud **A** se obtiene lo siguiente:

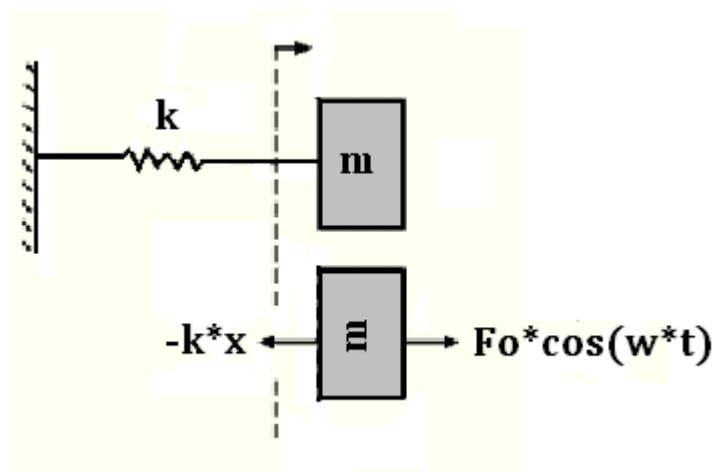
$$A = \frac{F_0/m}{(\omega^2 - \omega_f^2)} \quad (15)$$

Si se analiza detalladamente la ecuación (15) es posible realizar algunas observaciones que pueden en general ser aplicadas a todos los sistemas forzados.

- Si la frecuencia de la excitación ω_f es bastante menor a la frecuencia natural del sistema ω el sistema tenderá a ser constante de manera que la amplitud se exprese como $A = \frac{F_0}{k}$.
- Si la frecuencia de la excitación ω_f es bastante mayor a la frecuencia natural del sistema ω , el divisor tendería a ser infinito, haciendo que la amplitud tienda a cero $A \cong 0$.
- Si la frecuencia de excitación ω_f es aproximadamente igual a la frecuencia natural del sistema ω , el divisor tendería a cero, y la amplitud tendería a infinito $A \rightarrow \infty$, ocurriendo en ese instante resonancia en el sistema.

En la figura 4 se encuentra el ejemplo clásico de un sistema con vibración forzada y su modelamiento.

Figura 4. Diagrama de cuerpo libre de un sistema en vibración forzada



Fuente: Los autores.

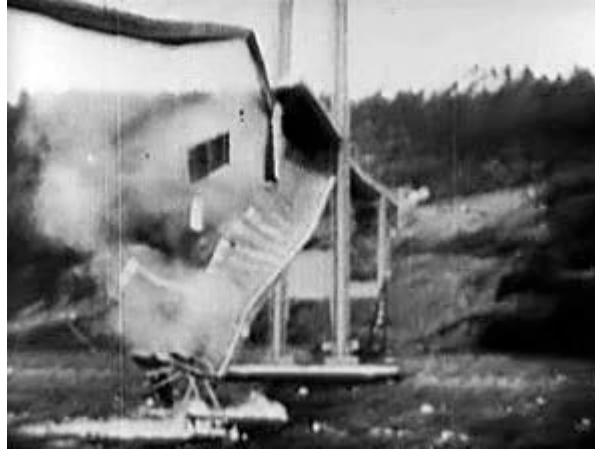
Si bien en la práctica la amplitud no va hacia infinito debido a efectos de fricción y amortiguamiento, es importante tener en cuenta esto a la hora de diseñar sistemas para evitar otros fenómenos que traiga consigo la resonancia como el desgaste excesivo o inclusive daños severos sobre los elementos del sistema.

1.3 RESONANCIA MECÁNICA

Fenómeno en el que hay incremento progresivo de la amplitud de vibración que acontece en todos los sistemas mecánicos como resultado de la excitación de los mismos en una frecuencia bastante cercana o igual a una de las frecuencias naturales del sistema³.

Su importancia radica en que puede generar problemas de integridad en el sistema, un ejemplo clásico de este fenómeno se encuentra en la figura 5, que es la caída del puente Tacoma Narrow en Estados Unidos donde los ingenieros no tuvieron en cuenta este aspecto durante el diseño de la estructura y su interacción con las corrientes de aire, otro caso bastante conocido es cuando los cantantes de ópera hacen sonidos que logran entrar en resonancia con un vaso de vidrio haciendo que este se rompa en varios pedazos.

Figura 5. Caída del puente Tacoma Narrow



Fuente: Wikipedia. http://es.wikipedia.org/wiki/Puente_de_Tacoma_Narrows.

1.4 VELOCIDAD CRÍTICA EN EJES

Se llama así a aquellas velocidades de rotación en las cuales el eje entra en resonancia. Esta velocidad es dependiente de la rigidez del eje y sus soportes, la masa del eje y el amortiguamiento del sistema, hay infinito número de velocidades críticas, sin embargo casi siempre las tres primeras son las más importantes ya que las otras son tan altas que prácticamente nunca un sistema mecánico alcanza ese nivel de operación¹.

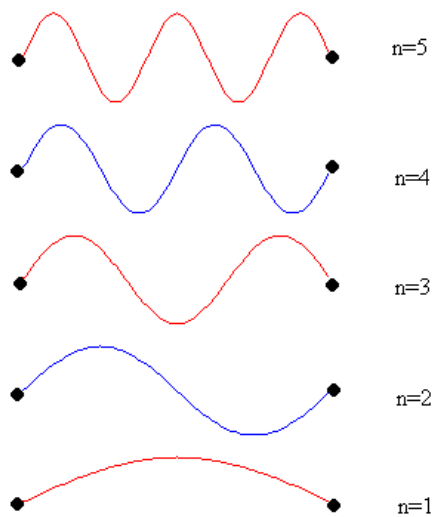
Determinar esta característica es fundamental para ciertas aplicaciones en las cuales la velocidad de rotación supera la primera o la segunda velocidad crítica puesto que mantenerse cerca a estos puntos genera daños graves que pueden inclusive generar paradas de la máquina o su cambio total, de forma que es necesario cambiar o alterar por completo el diseño del eje para evitar esta problemática².

1.4.1 Modos de vibración

Durante la operación cercana a las velocidades críticas se hace evidente los llamados modos de vibración que indica la forma característica que adquiere el elemento durante este estado de excitación. Los modos al igual que las velocidades son infinitos y el movimiento de un sistema es resultado de la combinación de los mismos, pero no todos excitados en el mismo grado³.

En la figura 6 se encuentra los modos de uno de los elementos que permite evidenciar este fenómeno con mayor facilidad como lo es una cuerda apoyada en sus extremos finales.

Figura 6. Modos de vibración en una cuerda



Fuente: <http://www.sc.ehu.es/sbweb/fisica/ondas/estacionarias/estacionarias.html>

2. MARCO TEÓRICO

2.1 MÉTODO PARA HALLAR VELOCIDADES CRÍTICAS EN EJES

Para resolver este problema se han creado una serie de métodos que de acuerdo a las capacidades de quien los utilice y la complejidad del elemento a modelar pueden ser utilizados. Los métodos de Dunkerley y Rayleigh pueden realizarse con cálculos sencillos y calculadora en mano pero se limitan a ejes con poca complejidad geométrica, en cambio el método de la matriz de transferencia requiere el uso de elementos computacionales pero permite ser aplicado a ejes complejos⁶.

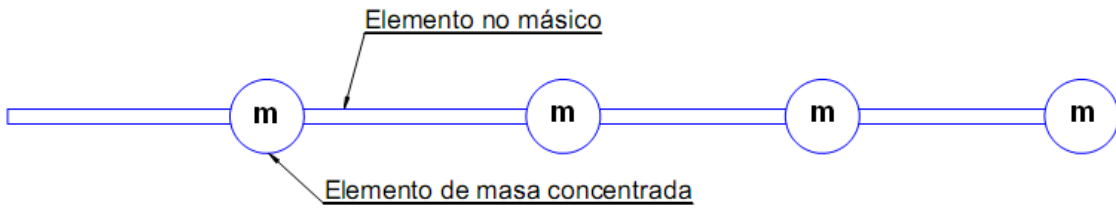
2.1.1 Matriz de transferencia

Este método fue desarrollado por Myklestad y Prohl de forma independiente, uno lo hizo para alas de avión y el otro para aspas de turbinas de vapor, en la década de los 40's del siglo XX⁷, sin embargo su modelo ha trascendido a otros espacios y con el advenimiento de la computación se facilitó su aplicación a figuras más complejas como ejes de turbo maquinaria o en general ejes escalonados.

La base del método consiste en el principio de los elementos de parámetros concentrados, donde una discretización adecuada puede asemejar un sistema continuo. Así pues se puede considerar el eje como una viga hecha de pequeñas masas concentradas unidas a través de elementos no másicos.

A las masas concentradas se les llamará *estaciones* y a los segmentos no másicos *campos*, y los vectores de estación representarán la deformación que se identificará con la letra Y , el desplazamiento angular con la letra ϕ , el momento de flexión con la letra M y la fuerza de corte Q , de manera que es un vector de 4 dimensiones. La figura 7 muestra los elementos que hacen parte de una discretización.

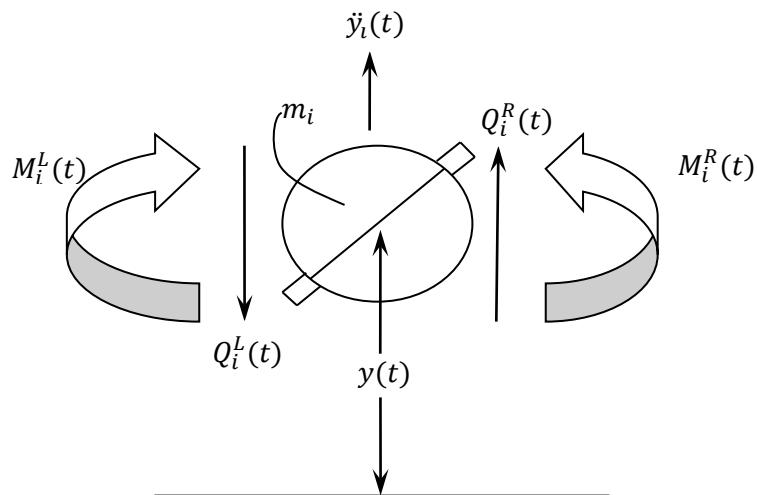
Figura 7. Discretización clásica.



Fuente: Los autores

Si a la i -ésima estación se le hace el diagrama de cuerpo libre se obtiene lo siguiente en la figura 8:

Figura 8. Diagrama de cuerpo libre de la i -ésima estación



Fuente: Los autores

De acuerdo con la segunda ley de Newton para el movimiento se tiene:

$$Q_i^R(t) - Q_i^L(t) = m_i * \ddot{y}_i * (t) \quad (16)$$

Si se supone que el desplazamiento $y_i(t)$ es armónico, y los cortantes $Q_i^R(t)$ y $Q_i^L(t)$ también, se puede obtener las siguientes ecuaciones:

$$y_i(t) = Y_i * \cos(w * t - \emptyset) \quad (17)$$

$$Q_i^R(t) = Q_i^R * \cos(w * t - \emptyset) \quad (18)$$

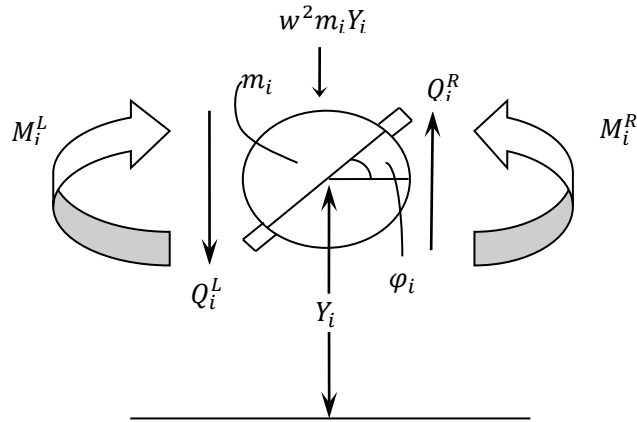
$$Q_i^L(t) = Q_i^L * \cos(w * t - \emptyset) \quad (19)$$

Si reemplazamos la segunda derivada de la ecuación (17), y las ecuaciones (18) y (19) en la ecuación (16), y dividimos sobre $\cos(w * t - \emptyset)$ se obtiene:

$$Q_i^R = Q_i^L - w^2 * m_i * Y_i \quad (20)$$

Esta ecuación permite expresar la fuerza cortante en el lado derecho de la estación i en términos de la fuerza cortante en el lado izquierdo y la fuerza inercial. Dado esto se puede reemplazar el diagrama de cuerpo libre de manera que se represente M_i^L y M_i^R que son las amplitudes de los momentos los cuales se suponen como elementos armónicos, inclusive se puede hacer lo mismo con el desplazamiento angular φ_i . La figura 9 muestra estos reemplazos en el diagrama de cuerpo libre.

Figura 9. Diagrama de cuerpo libre alterado de la i -ésima estación



Fuente: Los autores.

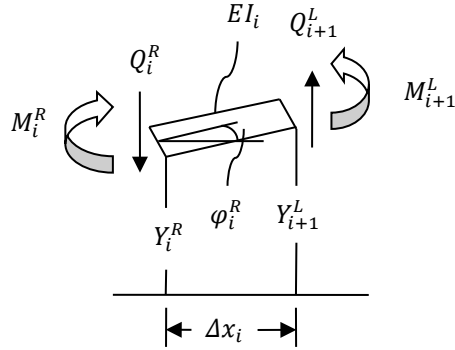
Si se tiene en cuenta que m_i es un punto concentrado de masa de forma que su momento de inercia másico es cero se puede concluir que $M_i^R = M_i^L$ y entonces por continuidad también se obtiene que $Y_i^R = Y_i^L = Y_i$ y $\varphi_i^R = \varphi_i^L = \varphi_i$, con estas suposiciones se puede construir la siguiente ecuación.

$$\begin{bmatrix} Y \\ \varphi \\ M \\ Q \end{bmatrix}_i^R = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -w^2 m_i & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} Y \\ \varphi \\ M \\ Q \end{bmatrix}_i^L = T_{S,i} * \begin{bmatrix} Y \\ \varphi \\ M \\ Q \end{bmatrix}_i^L \quad (21)$$

De manera que podemos llamar $T_{S,i}$ la estación de matriz de transferencia de la estación i .

Ahora lo que se hace es hallar la matriz de transferencia del campo, donde los superíndices R y L se referirán al lado derecho e izquierdo de la estación y no del campo. La figura 10 indica visualmente los elementos del diagrama de cuerpo libre del campo

Figura 10. Diagrama de cuerpo libre del i-ésimo campo



Fuente: Los autores.

Pero antes de encontrar las relaciones de carga-deformación se puede suponer la estación i como si estuviera fijada para introducir las definiciones de coeficiente influencia de flexibilidad así:

a_i^{YM} = desplazamiento en $i+1$ debido a un momento unitario en $i+1$, $M_{i+1}^L=1$

a_i^{YQ} = desplazamiento en $i+1$ debido a una fuerza unitaria en $i+1$, $Q_{i+1}^L=1$

$a_i^{\phi M}$ = pendiente en $i+1$ debido a un momento unitario en $i+1$, $M_{i+1}^L=1$

$a_i^{\phi Q}$ = pendiente en $i+1$ debido a una fuerza unitaria en $i+1$, $Q_{i+1}^L=1$

Se puede demostrar que los coeficientes de influencia de flexibilidad representan el recíproco de los correspondientes coeficientes de rigidez para una viga en voladizo, luego tendríamos.

$$a_i^{YM} = \frac{\Delta x_i^2}{2EI}$$

$$a_i^{YQ} = \frac{\Delta x_i^3}{3EI}$$

$$a_i^{\varphi M} = \frac{\Delta x_i}{EI}$$

$$a_i^{\varphi Q} = \frac{\Delta x_i^2}{2EI} \quad (22)$$

Con estas deducciones y la figura 10 se puede obtener lo siguiente:

$$Y_{i+1}^L = Y_{i+1}^R + (\Delta x_i)(\psi_{i+1}^R) + (a_i^{YM})(M_{i+1}^L) + (a_i^{YQ})(Q_{i+1}^L) = Y_{i+1}^R + (\Delta x_i)(\psi_{i+1}^R) + \frac{\Delta x_i^2}{2EI} M_{i+1}^L + \frac{\Delta x_i^3}{3EI} Q_{i+1}^L$$

$$\psi_{i+1}^L = \psi_{i+1}^R + a_i^{\varphi M} M_{i+1}^L + a_i^{\varphi Q} Q_{i+1}^L = \psi_{i+1}^R + \frac{\Delta x_i}{EI} M_{i+1}^L + \frac{\Delta x_i^2}{2EI} Q_{i+1}^L \quad (23)$$

Y dado que asumimos los elementos como no másicos podemos obtener lo siguiente:

$$M_{i+1}^L = M_i^R - \Delta x_i Q_i^R$$

$$Q_{i+1}^L = Q_i^R \quad (24)$$

Reemplazando (23) en (22) obtenemos:

$$Y_{i+1}^L = Y_{i+1}^R + (\Delta x_i)(\psi_{i+1}^R) + \frac{\Delta x_i^2}{2EI} M_i^R - \frac{\Delta x_i^3}{6EI} Q_i^R$$

$$\psi_{i+1}^L = \psi_{i+1}^R + \frac{\Delta x_i}{EI} M_i^R - \frac{\Delta x_i^2}{2EI} Q_i^R \quad (24)$$

Con esto se puede construir la siguiente ecuación:

$$\begin{bmatrix} Y \\ \varphi \\ M \\ Q \end{bmatrix}_{i+1}^L = \begin{bmatrix} 1 & \Delta x_i & \frac{(\Delta x_i^2)}{2EI} & -\frac{(\Delta x_i)^3}{6EI} \\ 0 & 1 & \frac{\Delta x_i}{EI} & -\frac{(\Delta x_i^2)}{2EI} \\ 0 & 0 & 1 & -\Delta x_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} Y \\ \varphi \\ M \\ Q \end{bmatrix}_i^R = T_{F,i} \begin{bmatrix} Y \\ \varphi \\ M \\ Q \end{bmatrix}_i^R \quad (25)$$

Donde $T_{F,i}$ es la matriz de transferencia del campo para el campo i . Juntando la ecuación (21) y la (25) se puede obtener.

$$\begin{bmatrix} Y \\ \varphi \\ M \\ Q \end{bmatrix}_{i+1}^L = T_i \begin{bmatrix} Y \\ \varphi \\ M \\ Q \end{bmatrix}_i^L \quad (26)$$

Donde $T_i = T_{F,i}T_{S,i}$ y es la matriz de transferencia que relaciona el vector de la estación en el lado izquierdo de la estación $i+1$ y el vector de la estación al lado izquierdo de la estación i .

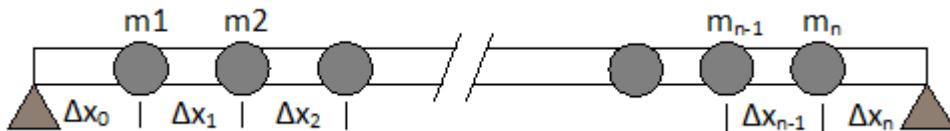
$$T_i = \begin{bmatrix} 1 + \frac{w^2 m_i (\Delta x_i^3)}{6EI} & \Delta x_i & \frac{(\Delta x_i^2)}{2EI} & -\frac{(\Delta x_i)^3}{6EI} \\ \frac{w^2 m_i (\Delta x_i^2)}{2EI} & 1 & \frac{\Delta x_i}{EI} & -\frac{(\Delta x_i^2)}{2EI} \\ w^2 m_i \Delta x_i & 0 & 1 & -\Delta x_i \\ -w^2 m_i & 0 & 0 & 1 \end{bmatrix} \quad (27)$$

Usando esto se puede hallar la ecuación de frecuencia del sistema hallando la matriz de transferencia total T .

3.1.1.1 Matriz de transferencia para apoyo simple-simple.

Para este caso la matriz es multiplicada de manera que se obtiene el valor de T total y el primer multiplicador es T_n de acuerdo la discretización mostrada en la figura 11.

Figura 11. Modelo discreto de un sistema con apoyo simple-simple



Fuente: Los autores.

$$T = T_n T_{n-1} T_{n-2} \dots T_2 T_1 T_{F,0}$$

$$T = \begin{bmatrix} t_{11}(w^2) & t_{12}(w^2) & t_{13}(w^2) & t_{14}(w^2) \\ t_{21}(w^2) & t_{22}(w^2) & t_{23}(w^2) & t_{24}(w^2) \\ t_{31}(w^2) & t_{32}(w^2) & t_{33}(w^2) & t_{34}(w^2) \\ t_{41}(w^2) & t_{42}(w^2) & t_{43}(w^2) & t_{44}(w^2) \end{bmatrix} \quad (28)$$

De acuerdo a las condiciones finales se halla el determinante.

Si el sistema se puede suponer como una viga con dos apoyos simples en los extremos las condiciones de frontera son:

$$Y_0 = 0 \quad M_0 = 0 \quad Y_{n+1} = 0 \quad M_{n+1} = 0$$

Con estas definiciones las siguientes ecuaciones se pueden determinar

$$0 = t_{12}(w^2) * \varphi_0 + t_{14}(w^2) * Q_0$$

$$\varphi_{n+1} = t_{22}(w^2) * \varphi_0 + t_{24}(w^2) * Q_0 \quad (29)$$

$$0 = t_{32}(w^2) * \varphi_0 + t_{34}(w^2) * Q_0$$

$$Q_{n+1} = t_{42}(w^2) * \varphi_0 + t_{44}(w^2) * Q_0$$

Se puede hallar la ecuación de la frecuencia con el siguiente determinante igualado a cero.

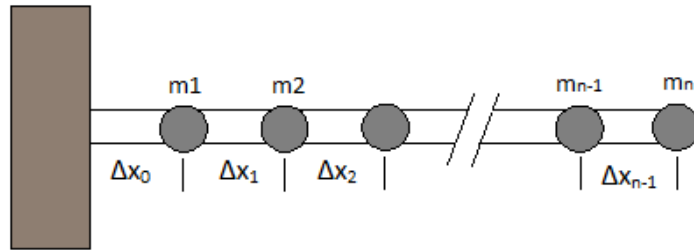
$$\det \begin{bmatrix} t_{12}(w^2) & t_{14}(w^2) \\ t_{32}(w^2) & t_{34}(w^2) \end{bmatrix} = t_{12}(w^2) * t_{34}(w^2) - t_{14}(w^2) * t_{32}(w^2) = 0 \quad (30)$$

De acuerdo a la cantidad de particiones nos va a dar la misma cantidad de valores de w que cumplen la condición de igualar a cero la ecuación, el valor más pequeño nos da el primer modo de vibración, el segundo valor nos da el siguiente modo de vibración y así sucesivamente.

3.1.1.2 Matriz de transferencia para apoyo empotrado-libre.

Para este caso la matriz es multiplicada de manera que se obtiene el valor de T total y el primer multiplicador es T_s de acuerdo la discretizacion mostrada en la figura siguiente.

Figura 12. Modelo discreto de un sistema simple-simple



Fuente: Los autores.

$$T = T_{S,n} T_{n-1} T_{n-2} \dots T_2 T_1 T_{F,0}$$

$$T = \begin{bmatrix} t_{11}(w^2) & t_{12}(w^2) & t_{13}(w^2) & t_{14}(w^2) \\ t_{21}(w^2) & t_{22}(w^2) & t_{23}(w^2) & t_{24}(w^2) \\ t_{31}(w^2) & t_{32}(w^2) & t_{33}(w^2) & t_{34}(w^2) \\ t_{41}(w^2) & t_{42}(w^2) & t_{43}(w^2) & t_{44}(w^2) \end{bmatrix} \quad (28)$$

De acuerdo a las condiciones finales se halla el determinante.

Si el sistema se puede suponer como una viga con un apoyo empotrado y el otro libre las condiciones de fronteras son:

$$Y_0 = 0 \quad \varphi_0 = 0 \quad M_n^R = 0 \quad Q_n^R = 0$$

Con estas definiciones las siguientes ecuaciones se pueden determinar

$$\begin{aligned} Y_n &= t_{13}(w^2) * M_0 + t_{14}(w^2) * Q_0 \\ \varphi_n &= t_{23}(w^2) * M_0 + t_{24}(w^2) * Q_0 \end{aligned} \quad (31)$$

$$0 = t_{33}(w^2) * M_0 + t_{34}(w^2) * Q_0$$

$$0 = t_{43}(w^2) * M_0 + t_{44}(w^2) * Q_0$$

Se puede hallar la ecuación de la frecuencia con el siguiente determinante igualado a cero.

$$\det \begin{bmatrix} t_{33}(w^2) & t_{34}(w^2) \\ t_{43}(w^2) & t_{44}(w^2) \end{bmatrix} = t_{33}(w^2) * t_{44}(w^2) - t_{34}(w^2) * t_{43}(w^2) = 0 \quad (32)$$

De acuerdo a la cantidad de particiones nos va a dar la misma cantidad de valores de w que cumplen la condición de igualar a cero la ecuación, el valor más pequeño nos da el primer modo de vibración, el segundo valor nos da el siguiente modo de vibración y así sucesivamente.

2.2 USANDO EL MÉTODO PARA HALLAR LOS MODOS DE VIBRACIÓN

Para hallar los modos de vibración de la primera de las ecuaciones de (29) se puede obtener $Q_0 = - \left[\frac{t_{12}(w^2)}{t_{14}(w^2)} \right] * \varphi_0$ tal que si se supone $\varphi_0 = 1$, y teniendo en cuenta las condiciones de frontera se puede usar el extremo izquierdo del vector estación para hallar los n -ésimos modos.

$$\begin{bmatrix} Y \\ \varphi \\ M \\ Q \end{bmatrix}_0 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ - \frac{t_{12}(w^2)}{t_{14}(w^2)} \end{bmatrix} \quad (33)$$

Para hallar los modos es necesario encontrar las soluciones a la ecuación (30) y reemplazarlo en (33), e ir multiplicando las matrices de cada estación para ir hallando la deformación en cada elemento de masa concentrada, a mayor cantidad de divisiones mayor precisión en el dibujo. Así con la menor solución que

sería la primera velocidad crítica se halla la forma del primer modo de vibración, y así sucesivamente para todos los demás modos⁵.

Para el caso de apoyo empotrado-libre se reemplaza $Q_0 = -\left[\frac{t_{33}(w^2)}{t_{34}(w^2)}\right] * M_0$, se supone $M_0 = 1$ y aplicando las respectivas condiciones de frontera se obtiene.

$$\begin{bmatrix} Y \\ \varphi \\ M \\ Q \end{bmatrix}_0 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ -\frac{t_{33}(w^2)}{t_{34}(w^2)} \end{bmatrix} \quad (34)$$

Para una mayor aclaración de cómo aplicar el método se puede ver el Anexo A donde hay un ejemplo desarrollado.

2.3 DE LA DISCRETIZACIÓN

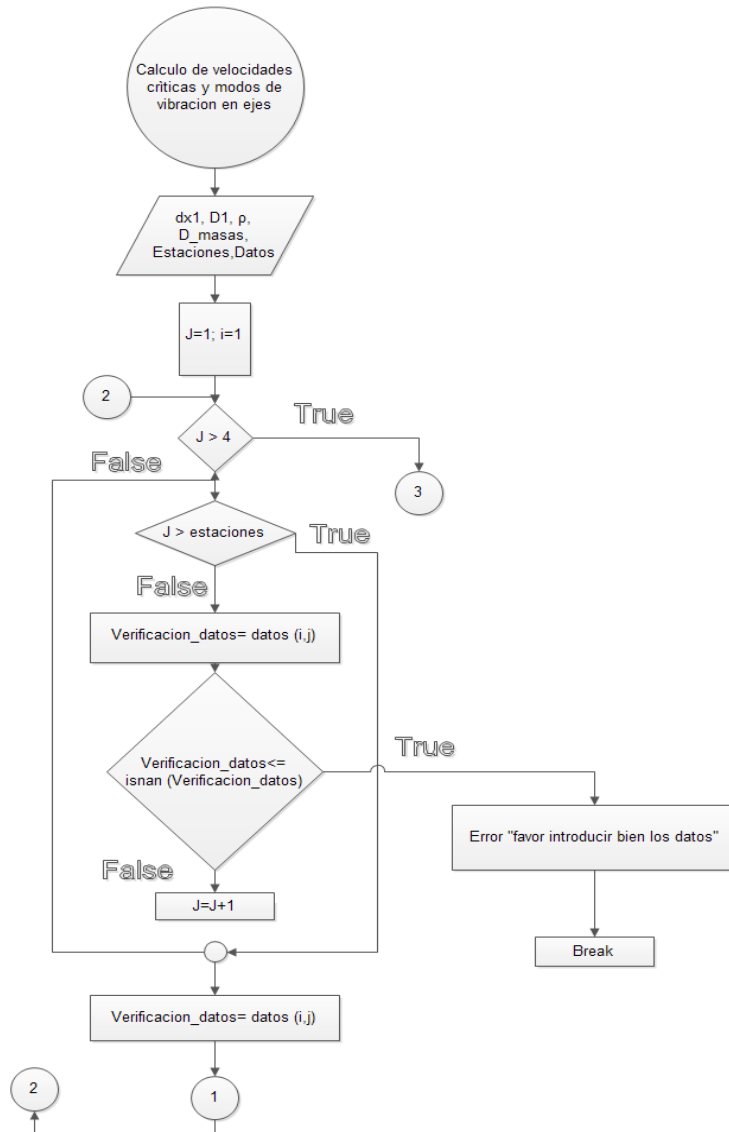
Si bien acá se usa un método de discretización en particular, existen por lo menos tres maneras distintas de hacerlo de acuerdo a que elemento se coloque primero, si el elemento másico o el elemento no másico, esto implica que a la hora de hacer las matrices de estación y de campo se debe ser cuidadoso para no cometer errores que impliquen que el modelo matemático no interpreta correctamente el modelo físico que se ha construido, lo que genera que el resultado se aleje de manera ostensible del valor buscado.

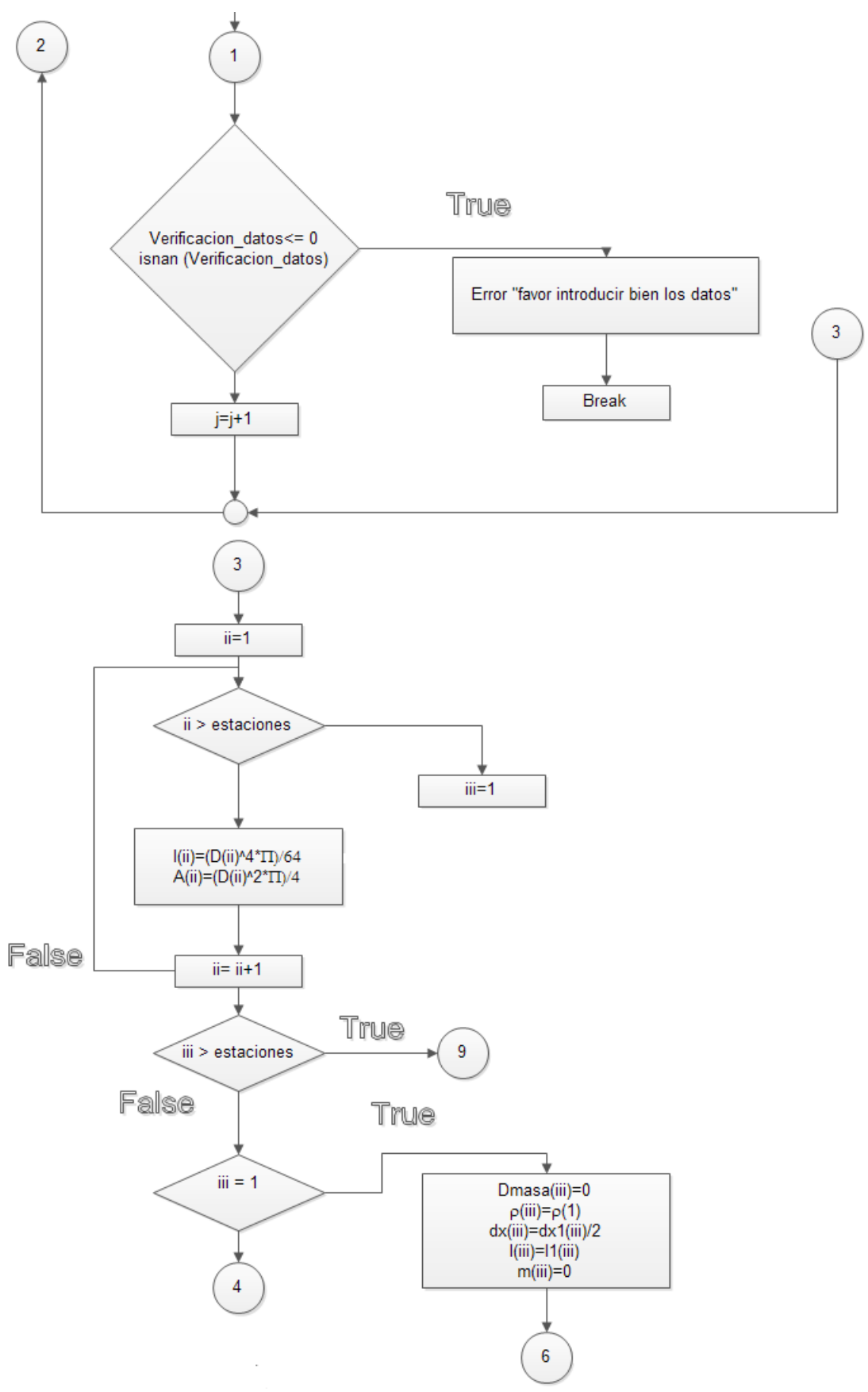
3. PROGRAMACIÓN DEL MÉTODO

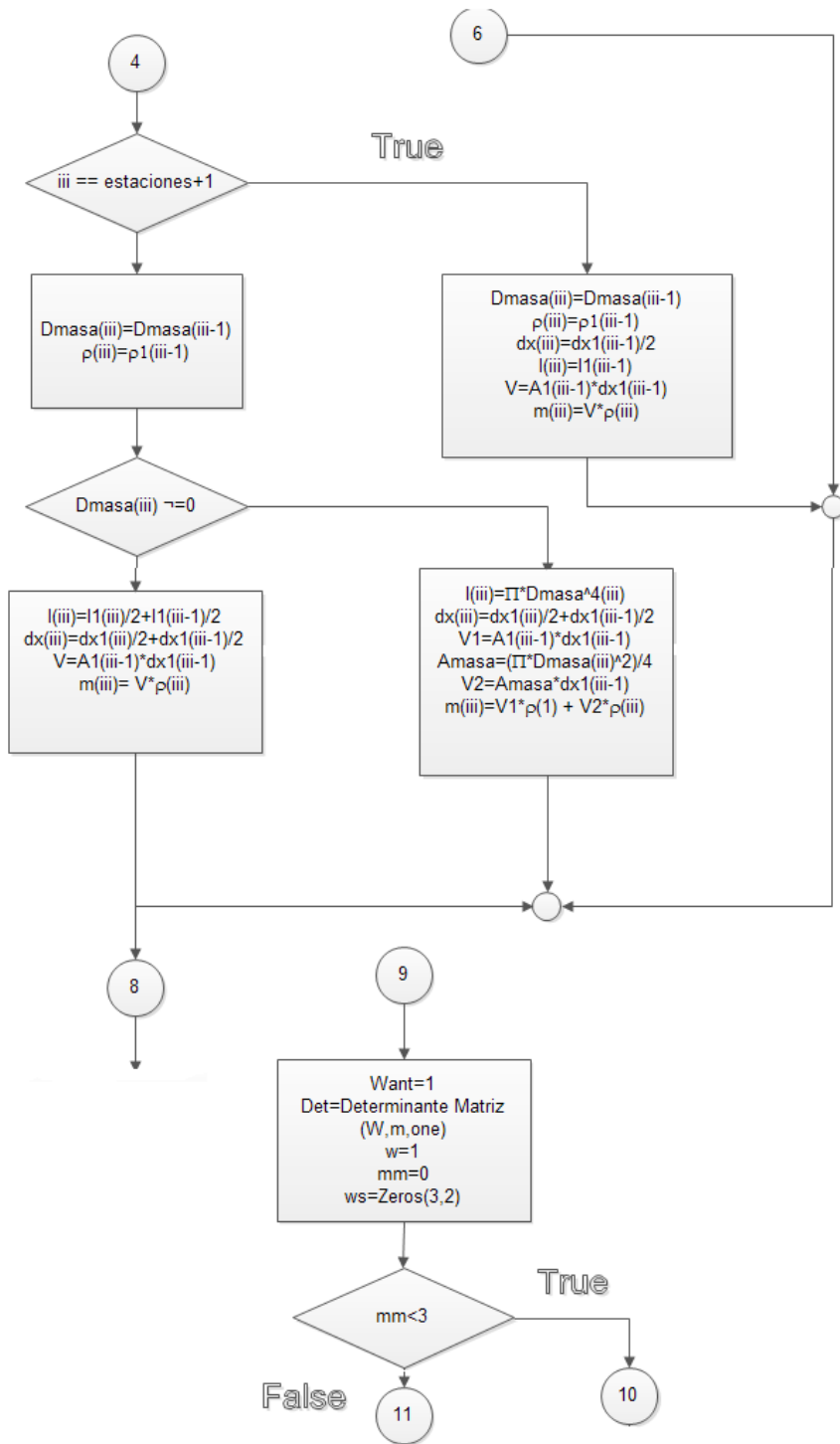
3.1 DIAGRAMA DE FLUJO

A manera de resumen se coloca el diagrama de flujo en la figura 13 que permite ver de manera rápida el método y el código, además que permite que en trabajos futuros sea usado para programar en otra plataforma si es necesario.

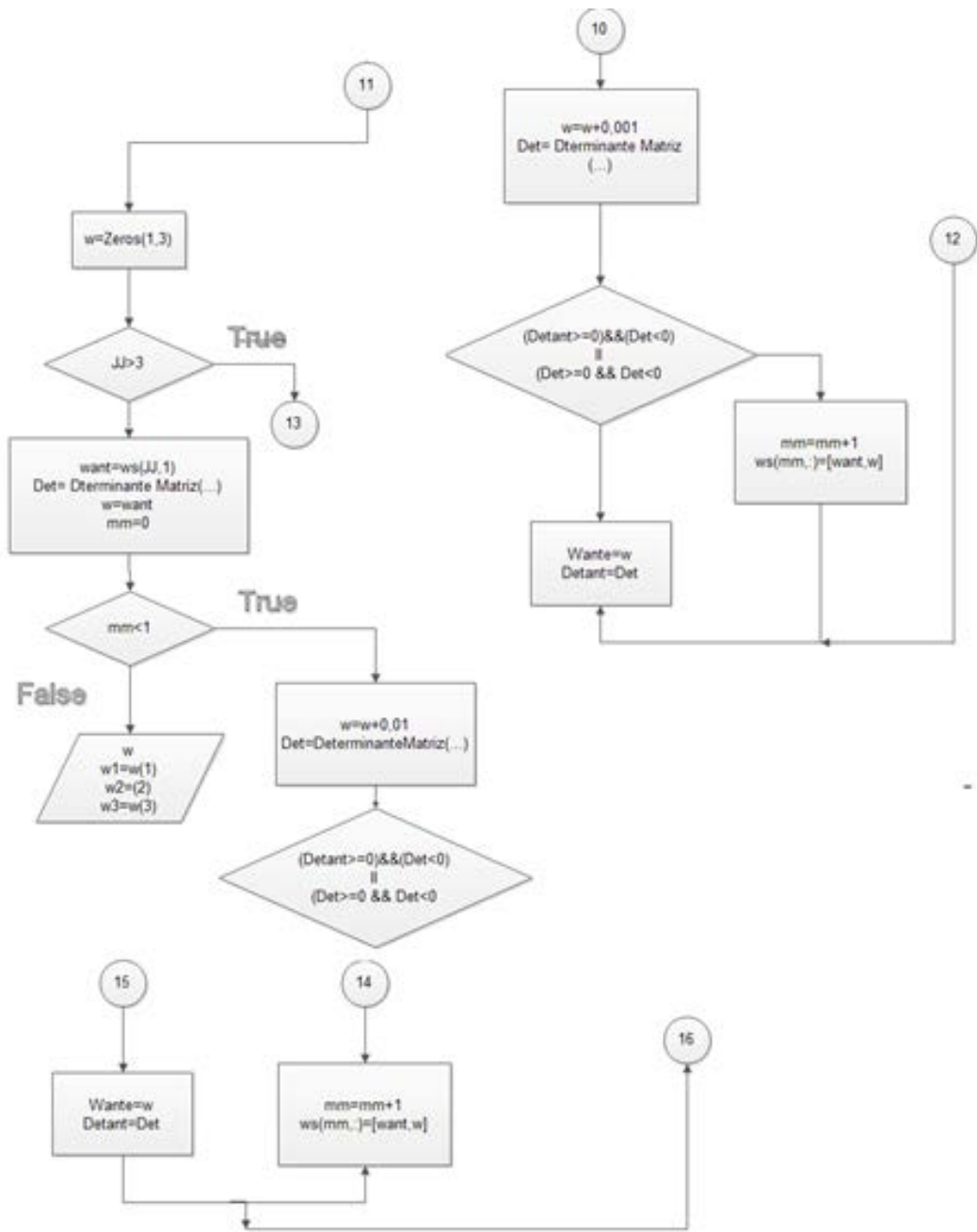
Figura 13. Diagrama de flujo del código programa







12

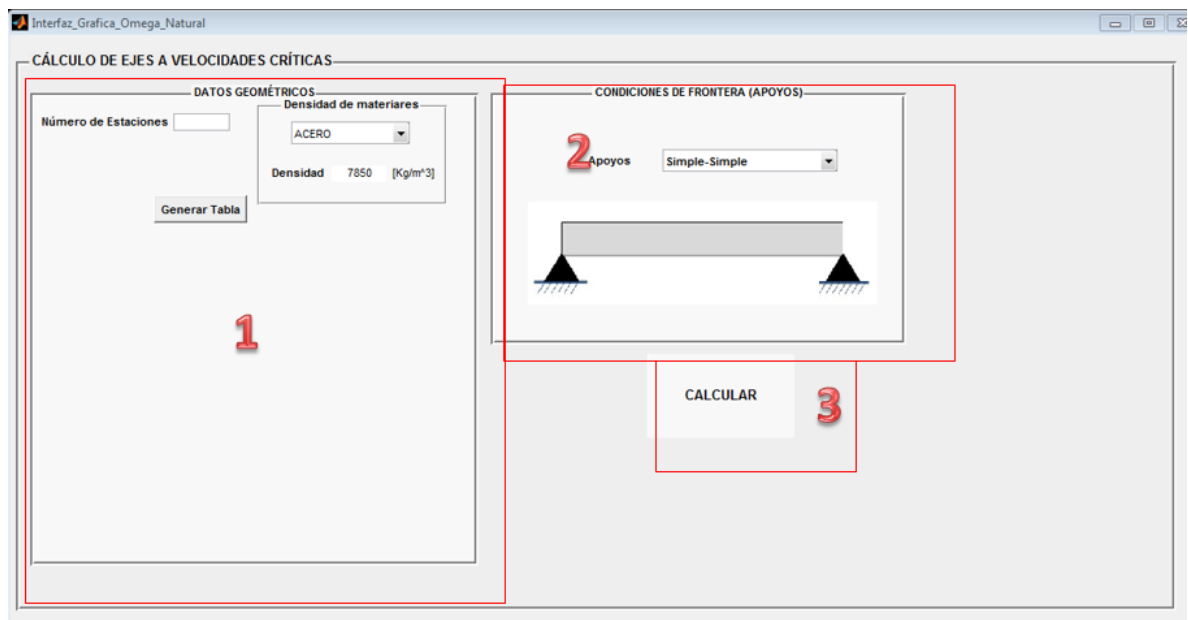


3.2 INTERFAZ GRÁFICA

Se realizó una interfaz gráfica amigable utilizando el GUIDE de Matlab la cual se explicará a continuación y que servirá como guía de uso del software:

Al correr el programa aparecerá una ventana como la figura 14, donde se puede observar dos zonas claramente definidas, el botón calcular y una última zona (zona 4), que aparecerá después presionar el botón calcular.

Figura 14. Zonas de interfaz gráfica



Fuente: Los autores

3.2.1 Datos geométricos (Zona 1)

En esta zona se piden los siguientes datos:

- **Número de Estaciones:** El usuario debe ingresar el número de estaciones deseadas, las cuales indican el número de nodos en que se va a partir el eje. (Ver Anexo A).
- **Densidad de Materiales:** El menú desplegable muestra las densidades de los materiales más comunes de los ejes. Se incluye como medio de ayuda para el usuario.
- **Generar Tabla:** Al presionar el botón se generará una tabla con el número de filas igual al número de estaciones deseadas y con 4 columnas (dx, Diámetro, Densidad, Diámetro Masa Puntual). Como se puede observar en la figura 15 y 16.

Figura 15. Detalle zona 1 (datos geométricos)

The image shows a software window titled "DATOS GEOMÉTRICOS". Inside the window, there are three main elements: a text input field labeled "Número de Estaciones" which is currently empty; a dropdown menu labeled "Densidad de materiales" with "ACERO" selected; and a label "Densidad 7850 [Kg/m³]" positioned below the dropdown. A button labeled "Generar Tabla" is located to the left of the dropdown menu.

Fuente: Los autores

Figura 16. Detalle zona 1 (tabla generada)

DATOS GEOMÉTRICOS

Número de Estaciones

Densidad de materiales

ACERO

Densidad 7850 [Kg/m³]

Generar Tabla

	dx [m]	Diámetro [m]	Densidad [Kg/m ³]	Diámetro Masa Puntual [m]
1				
2				
3				
4				

Fuente: Los autores.

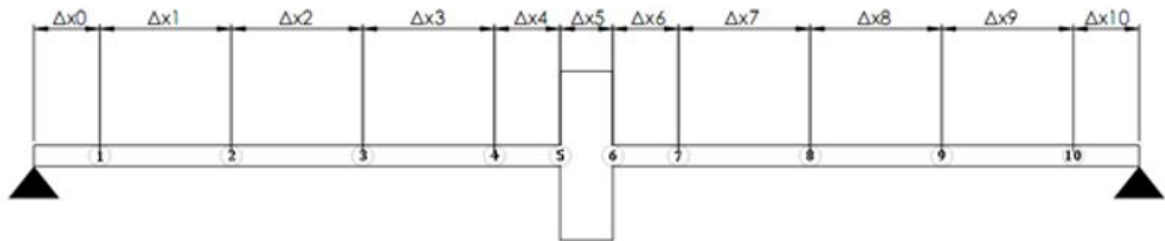
En la tabla generada se piden los siguientes datos:

- **dx:** Se define como el paso entre cada una de las estaciones. ($dx = \Delta x$). Se recomienda usar una relación de paso sobre diámetro ($\frac{dx}{\theta}$) menor a 1⁸.
- **Diámetro:** Hace referencia al diámetro base del eje en cada una de las estaciones.
- **Densidad:** Corresponde a la densidad del eje en cada estación.

• **Diámetro Masa Puntual:** Se refiere al diámetro de cada masa puntual soportada por el eje (poleas, engranajes, volantes etc.).

La figura 17 muestra con mayor claridad a que se refiere el texto con el valor Δx solicitado para el software.

Figura 17. Detalle del Δx



Fuente: Los autores.

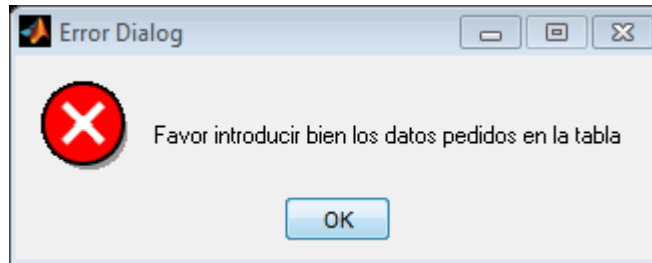
3.2.1.1 Recomendaciones durante la entrada de datos.

Dada la manera como se programó la aplicación se han creado ciertas alarmas para el usuario de manera que no genere errores en los resultados, aquí se anuncian los cuidados a tener en cuenta.

- Las medidas se deben ingresarse en unidades del Sistema Internacional.
- Dado el caso de que no existan masas puntuales en el eje, los espacios deben ser llenados con cero.
- Después del ingreso del último dato de la tabla generada, el cursor debe ser retirado de la última posición, de lo contrario la matriz de datos no será reconocida.
- Ninguna variable diferente al Diámetro de Masa Puntual puede ser cero.

Cometer un error al momento de ingresar los datos para las tres primeras recomendaciones se generará un cuadro de error como se puede apreciar en la figura 18.

Figura 18. Cuadro de alarma de error en el ingreso de datos

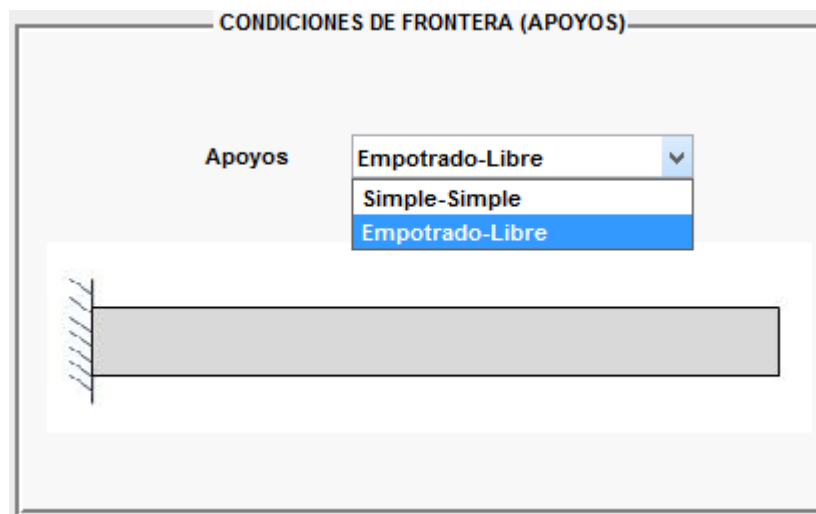


Fuente: Los autores.

3.2.2 Condiciones de frontera (Zona 2)

Para facilitar el entendimiento visual de las condiciones de frontera el menú desplegable permite decidir las condiciones de apoyo de acuerdo a los soportes supuestos por el usuario, una imagen de la zona se encuentra en la figura 19.

Figura 19. Visualización de las condiciones de frontera

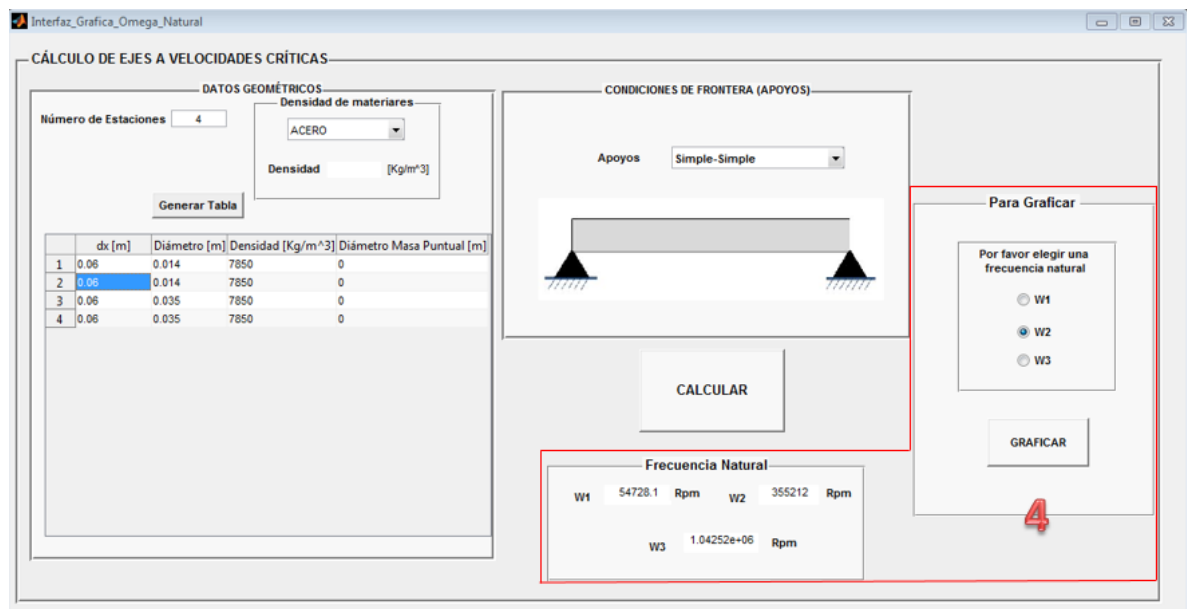


Fuente: Los autores.

3.2.3 Botón calcular (Zona 3).

Con esta opción y teniendo correctamente los datos el usuario puede obtener las tres primeras frecuencias naturales del sistema modelado, además aparecerá un panel donde puede seleccionar el modo a visualizar de acuerdo a la velocidad seleccionada. La figura 20 da un ejemplo del despliegue del botón calcular.

Figura 20. Botón calcular y su despliegue



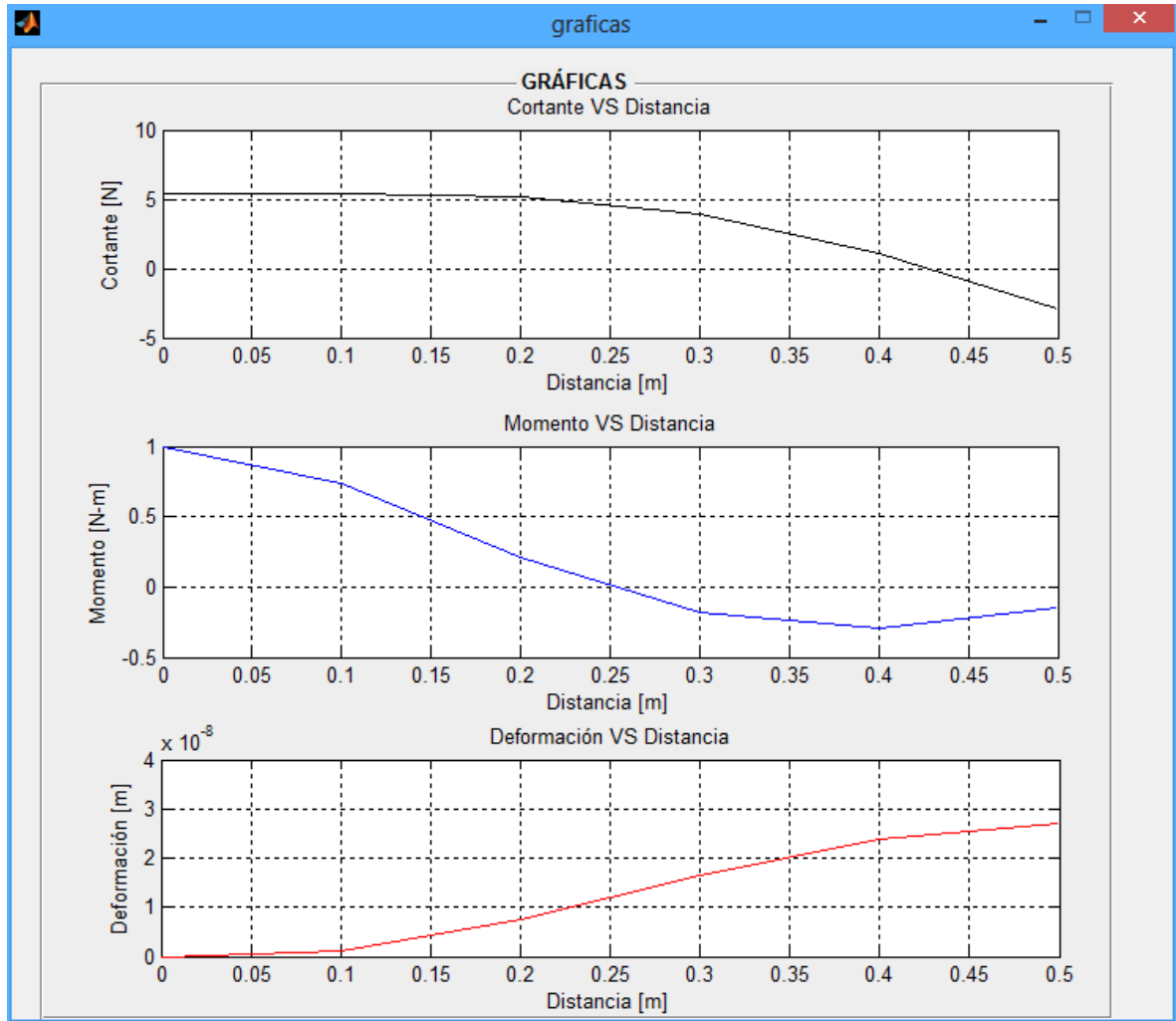
Fuente: Los autores.

3.2.4 Selección de modos a graficar (Zona 4).

La zona 4 comprende de 2 paneles, uno llamado *Frecuencia Natural*, que contiene los valores de las tres primeras frecuencia naturales calculadas por el programa y el otro llamado *Para Graficar*, en el cual el usuario puede seleccionar uno de los

modos a graficar, obteniendo el comportamiento de la deformación, el cortante y el momento. En la figura 21 se muestra un ejemplo de esta zona.

Figura 21. Salida gráfica de la deformación, momento y cortante



Fuente: Los autores

4. PRUEBAS EXPERIMENTALES

4.1 PRUEBAS REALIZADAS

Con el fin de validar las respuestas obtenidas de la programación se van a realizar dos de pruebas en el laboratorio de vibraciones de la escuela de ingeniería mecánica, por un lado con un banco se va a llevar un prototipo de eje a distintas velocidades con el fin de encontrar sus velocidades críticas, además se va a implementar otra prueba referente a excitar el sistema a través de un impacto con el fin de determinar por ese método sus frecuencias naturales.

4.2 PRUEBA DE IMPACTO

4.2.1 Montaje del experimento

En un primer momento lo que se hizo fue colocar el eje sobre unas cuerdas amarradas a una viga de manera que los apoyos sean lo suficientemente elásticos como para que el sistema se pudiera suponer libre, tal y como se muestra en la figura 22, con este montaje el eje es excitado a través de un golpe del martillo con el fin de excitar sus frecuencias naturales.

Tras obtener los datos se hizo una segunda toma con el sistema montado sobre los rodamientos y acoplado al motor de manera que se obtuviera valores del sistema que incluyeran la rigidez de los soportes, esto se puede ver en la figura 23.

Figura 22. Montaje sobre cuerdas



Fuente: Los autores.

Figura 23. Montaje del experimento



Fuente: Los autores.

4.2.2 Toma de datos

Para realizar la toma de datos se contó con los elementos que hacen parte del laboratorio de vibraciones mecánicas de la Universidad Industrial de Santander. El módulo de toma de datos es de tipo NI 9234 de la serie C de cuatro canales cuyas características se listan a continuación en la tabla 1.

Tabla 1. Especificaciones del módulo de toma de datos.

Especificaciones del módulo NI 9234 C	
Número de canales	4
Resolución	24 bits
Velocidad de muestreo	51.2 kS/s
Rango de voltaje máximo	-5V a 5V
Modo de muestreo	Simultaneo
Impedancia de entrada	305 kOhm
Rango dinámico	102 dB
Excitación de corriente	2mA

Fuente: <http://sine.ni.com/nips/cds/print/p/lang/es/nid/208802>

Junto al módulo se debe usar el chasis que lo contiene y que transmite los datos hacia el computador, sus especificaciones se encuentran en la tabla 2.

Tabla 2.Especificaciones del chasis.

Especificaciones del chasis NI USB-9162	
Interface	USB 2.0 High Speed
Corriente de consumo	500 mA, máx.
Peso	101 g

Fuente: NI USB-9162 user guide.

Para realizar la toma de datos se utilizó un martillo de marca *National instruments* cuyas características se colocan en la tabla 3 a continuación.

Tabla 3.Especificaciones del martillo de impacto.

Especificaciones técnicas del martillo de impacto	
Sensibilidad	2.25 [mV/N]
Rango de medición	2200 [N pk]
Masa	0.16 [Kg]

Fuente: <http://sine.ni.com/ds/app/doc/p/id/ds-364/lang/es>

Para la adquisición de datos posterior al golpe se utiliza un acelerómetro cerámico de alta sensibilidad cuyas especificaciones se dan a continuación en la tabla 4.

Tabla 4.Especificaciones del acelerómetro.

Especificaciones técnicas del acelerómetro	
Sensibilidad	100 [mV/g]
Rango de medición	50 g pk
Resolución de banda	1 a 10000 Hz
Rango de frecuencia	0.5 a 10000 Hz
Peso	5.8 gramos

Fuente: <http://sine.ni.com/ds/app/doc/p/id/ds-364/lang/es>

El software utilizado es Labview que es el perteneciente a la casa matriz que ofrece los sensores, de manera que se asegure una total compatibilidad. La figura 24 muestra el sistema de adquisición de datos y el martillo usado en la toma de los mismos.

Figura 24. Elementos de adquisición de datos



Fuente: Los autores.

5.2.3 Resultados obtenidos

Se hicieron varias tomas y los resultados se resumen en la tabla 5.

Tabla 5 Resumen de resultados obtenidos en la prueba de impacto.

Modelo	Valor de frecuencia (Hz)	Tipo de apoyo
Eje con tres masas	10.8	Con cuerdas
Eje con tres masas	42.66	Con rodamientos

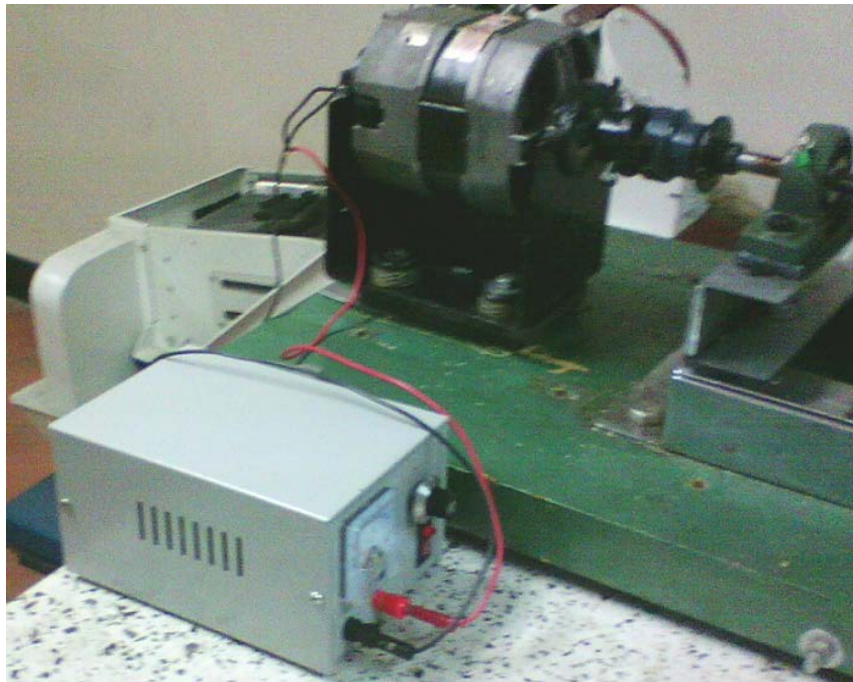
Fuente: Los autores.

4.3 PRUEBA POR VARIACIÓN DE VELOCIDAD

4.3.1 Montaje del experimento

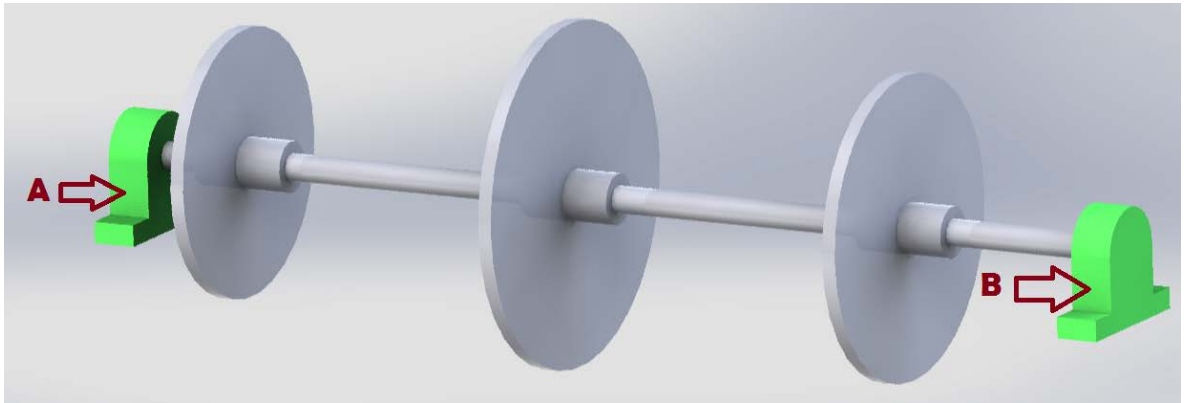
Para este caso lo que se hizo fue obtener un motor que lograra altas revoluciones y que se pudiera variar su velocidad para acoplarlo al eje que simulara un modelo definido, posteriormente se mide la vibración en los rodamientos o soportes (A y B) para determinar la variación de la vibración a través de las distintas frecuencias para encontrar en cuál de estas la excitación aumentaba de manera sustancial. En la figura 25 se encuentra el motor y variador usados en la prueba, y en la figura 26 se muestra los respectivos rodamientos donde se tomó datos.

Figura 25. Motor y variador de velocidad



Fuente: Los autores

Figura 26. Puntos de toma de datos del experimento.



Fuente: Los autores.

4.3.2 Toma de datos

Para la medición de la variación de la medición se usaron los mismos instrumentos del ítem anterior (acelerómetro, chasis, módulo de toma de datos) excepto el martillo, además se necesitó un foto-tacómetro de las características colocadas en la tabla 6 y la figura 27 es una foto del mismo.

Tabla 6. Especificaciones del acelerómetro

Especificaciones técnicas del foto-tacómetro	
Precisión	+/- 0.01% ó +/- 1 dígito
Velocidad medible	5 a 99,999 r.p.m.
Resolución	0.001 a 1 r.p.m. rango automático
Rango	5 a 90 cm.
Estándar	IEC 1010-1

Fuente:<http://www.instrumentation2000.com/monarch-instruments-pt99-tachometer.aspx>

Figura 27. foto-tacómetro utilizado durante las pruebas.



Fuente: Los autores.

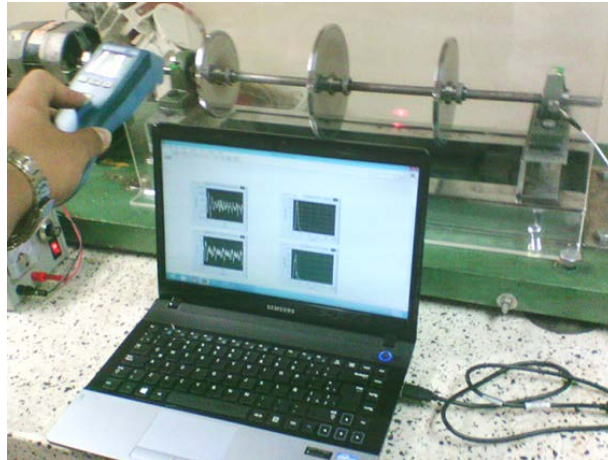
Para hacer más sencilla la prueba se decidió empezar en 400 r.p.m. e ir variando la velocidad del motor en intervalos de 50 r.p.m., con estos datos se obtuvo gráficas que permitieron determinar un intervalo donde se presentaba particularmente un aumento de la magnitud de la vibración.

Cabe decir que también se presentaba un fenómeno de desprendimiento del acople motor-eje durante el paso por esta velocidad, hecho que permite dilucidar un comportamiento de flexión excesiva en ese instante.

Por otro lado se paró la prueba al llegar a los 4000 r.p.m. debido a que se evidenciaba un aumento excesivo de la vibración debido al parecer a que el motor eléctrico mecánicamente no soportaba el esfuerzo mecánico exigido.

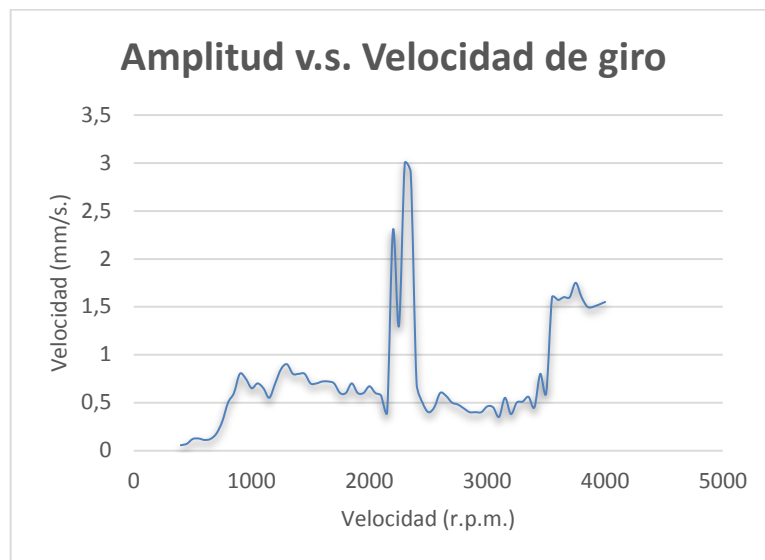
En la figura 28 se encuentra una imagen de uno de los momentos en que se estaba tomando medidas y la figura 29 es una imagen de los resultados obtenidos.

Figura 28. Toma de datos



Fuente: Los autores

Figura 29. Ejemplo de datos adquiridos en el experimento



Fuente: Los autores

4.3.3 Resultados obtenidos

De acuerdo a los datos obtenidos se determinó un intervalo donde se prevé se encuentra la velocidad crítica del eje, la tabla 7 muestra el intervalo seleccionado.

Tabla 7. Resultado prueba de variación de velocidad

Modelo	Intervalo de velocidad (r.p.m.)
Eje con tres masas	2200 a 2400

Fuente: Los autores.

4.4 VERIFICACIÓN DE LOS DATOS DEL SOFTWARE

Se resumirán los valores obtenidos para el modelo probado, se asumirá como valor teórico el resultado del software y experimental los valores obtenidos de los ensayos realizados.

Además se va a hacer una simulación con el software ansys, pero usaremos otro tipo de soportes con el fin de probar el software acá diseñado en sistema empotrado-libre, en el anexo C se puede ver el reporte entregado.

En la tabla 8 se resumen los resultados y el error obtenido en la simulación con ansys, y en la tabla 9 se hace el mismo ejercicio pero esta vez con los datos de las pruebas experimentales.

Tabla 8. Resumen de resultados de los software

Obtención del dato	Resultado	%Error
Software diseñado	16.09 Hz-101.78Hz-414.05Hz	--
Software ANSYS workbench	18.142Hz-105Hz-265Hz	12.75%,3.16%,35.9%

Fuente: Los autores

Además se va a hacer una simulación con el software ansys, pero usaremos otro tipo de soportes con el fin de probar el software acá diseñado en sistema empotrado-libre, en el anexo C se puede ver el reporte entregado.

Tabla 9. Resumen de resultados experimentales

Obtención del dato	Resultado	%Error
Software	2506.76 r.p.m.	--
Ensayo impacto	2559.6 r.p.m.	2.1%
Ensayo variación velocidad	2200 a 2400 r.p.m.	12.23% a 4.26%

Fuente: Los autores

De acuerdo con los resultados de desviación de error obtenidos se considera que el software tiene suficiente precisión, sin embargo no se puede ignorar el error obtenido y a continuación se consideran algunos causantes de estas pequeñas incongruencias.

El modelo desarrollado en el software no tiene en cuenta el acople al motor ni las afectaciones que este conjunto mecánico puede generar, que como ya se explicó obligó a detener en un momento las mediciones.

Las condiciones de frontera asumidas en el modelo son de rigidez total, caso que no es absolutamente cierto ya que todos los sistemas tienen algún tipo de amortiguación implícita, sin embargo cabe aclarar que de acuerdo a investigaciones ya referenciadas para el caso de la primera velocidad crítica esta asunción no altera de manera ostensible el resultado.

Para el caso de la prueba de impacto el hecho de que el eje no rote implica de inmediato incertidumbre, además del brazo de la persona que golpea, la correcta realización de la prueba e inclusive el montaje del acelerómetro.

Con respecto a las simulaciones realizadas en el software ansys se evidencia altos niveles de error en el tercer caso de frecuencia hallado, sin embargo es de tener en cuenta que el modelo que se asumió en la generación del software de este proyecto no considera los efectos giroscópicos para sistemas de un apoyo, lo que a altas velocidades de giro puede alterar de manera ostensible la frecuencia del sistema. Es válido decir que Ansys no solo entrega valores de frecuencia en flexión si no en torsión, sin embargo dado el alcance de este proyecto estos valores se simularon pero no se tuvieron en cuenta.

Por último es bueno decir que si bien se obtuvo el dato del eje en un sistema libre como lo fue el del ensayo de impacto sobre cuerdas, este no fue posible compararlo con el software dado que no fue diseñado para ignorar las condiciones de frontera.

CONCLUSIONES

- Se logró diseñar un software que permite determinar con suficiente precisión la velocidad crítica de sistemas rotativos con variación de diámetros a través de su geometría, además se puede también obtener el modo de vibración, el momento y el cortante.
- A través de dos pruebas se logró verificar con suficiente precisión el resultado obtenido del software con respecto a la realidad.
- Queda estipulado en este trabajo la manera como se debe utilizar la interfaz diseñada de manera que cualquier persona pueda manipularla leyéndolo.

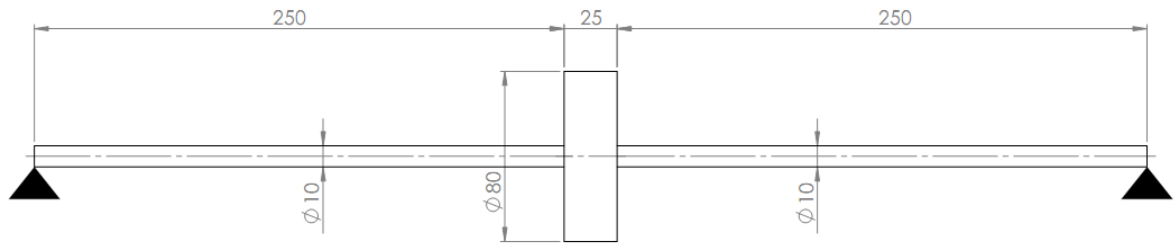
BIBLIOGRAFÍA

- 1) AKPOBI, J.A.OVUWORIE,G.C. Computer Aided design of the critical speed of shafts. En Journal of applied sciences and enviromental management. Nigeria.2008. Vol. 12.p.79-86.
- 2) CRAWFORD, Barry. MYRICK, Ted. SCHUMPERT, Gerald. Changing and controlling the “first critical speed” of overhung centrifugal fans. Proceedings of the thirty third turbo machinery symposium.2004.
- 3) FORERO, Jonathan. PÉREZ , Néstor. Desarrollo de una metodología basada en resonancia para ensayos de fatiga a flexión en probetas metálicas. Trabajo de grado para optar por el título de ingeniero mecánico. Bucaramanga: Universidad Pontificia Bolivariana. Escuela de ingeniería y administración. Facultad de ingeniería mecánica.2009.p.23.
- 4) GARCÍA C., Alfonso. Vibraciones mecánicas. Módulo 1. Bucaramanga. Universidad Industrial de Santander. 2001. p. 5-30.
- 5) MEIROVITCH, Leonard. Fundamentals of vibrations. New York. Mc Graw Hill. 2001. p.464-492.
- 6) MORSE, Ivan. HINKLE, Rolland.TSE, Francis. Mechanical vibrations, theory and applications. Second edition. Boston, Massachusetts.1978.p.449.
- 7) MURPHY, B. VANCE, J. TRIPP, H. Critical speed of turbomachinery: Computer prediction vs. Experimental measurements- Part 1: the rotor mass-elastic model. En ASME. Journal of vibration, acoustics, stress, and reliability in design. Enero 1987.Vol 109.
- 8) MURPHY, B. VANCE, J. ZEIDAN, F. Machinery vibration and rotordynamics. New Jersey, United States. John Wiley & Sons, Inc. Hoboken.2010.p.135-137.
- 9) MYKLESTAD, N. O. “A new method of calculating natural modes of uncoupled bending vibration of airplane wings and other times of beams”. En ASME. Journal of aeronautical sciences. Abril 1944. p. 153-162.
- 10)QUIROGA, Jabid. Cátedra de vibraciones mecánicas. Apuntes de clase.2013

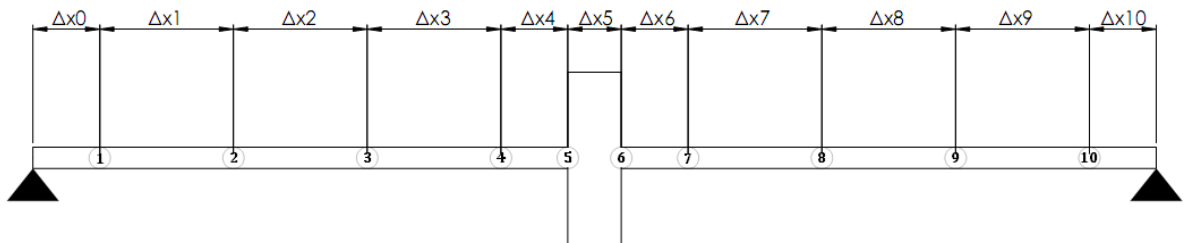
ANEXOS

ANEXO A. EJEMPLO DE CÁLCULO DE VELOCIDAD CRÍTICA DE UN EJE A TRAVÉS DEL MÉTODO DE MATRIZ DE TRANSFERENCIA

Para ejemplificar el método vamos a hacer un cálculo de un eje sencillo cuyas dimensiones se encuentran en la siguiente figura.



El material es acero de módulo de elasticidad $E=2,2 \cdot 10^{11}$ (N/m) y cuya densidad es $\rho=7850$ (kg/m³), para discretizar se divide en 10 partes el sistema, se puede dividir de las formas que se desee pero por simplicidad se hace de la forma descrita.



Se tabula con el fin de ver de manera ordenada los datos que se necesitan para los cálculos posteriores.

Estación	Longitud (m)	Inercia (m ⁴)	Masa (kg)
0	0,03125	$4,9087 \cdot 10^{-10}$	0
1	0,06250	$4,9087 \cdot 10^{-10}$	0,03853

2	0,06250	$4,9087 \cdot 10^{-10}$	0,03853
3	0,06250	$4,9087 \cdot 10^{-10}$	0,03853
4	0,03125	$4,9087 \cdot 10^{-10}$	0,01927
5	0,02500	$2,0106 \cdot 10^{-6}$	0,98650
6	0,03125	$4,9087 \cdot 10^{-10}$	0,01927
7	0,06250	$4,9087 \cdot 10^{-10}$	0,03853
8	0,06250	$4,9087 \cdot 10^{-10}$	0,03853
9	0,06250	$4,9087 \cdot 10^{-10}$	0,03853
10	0,03125	$4,9087 \cdot 10^{-10}$	0,01927

Ahora se construye las matrices correspondientes.

Para la estación 0 se tiene la siguiente multiplicación de matrices cuyo resultado se llama A, para en un cálculo posterior.

$$\begin{matrix} Y \\ \varphi \\ M \\ Q \end{matrix} = \begin{bmatrix} 1 & 0,03125 & 4,52 \cdot 10^{-6} & -4,71 \cdot 10^{-8} \\ 0 & 1 & 2,89 \cdot 10^{-4} & -4,52 \cdot 10^{-6} \\ 0 & 0 & 1 & -0,03125 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = A$$

Para las estaciones 1, 2, 3, 7, 8 y 9 se tiene la siguiente multiplicación de matrices cuyo resultado se llama B, para un cálculo posterior.

$$\begin{matrix} Y \\ \varphi \\ M \\ Q \end{matrix} = \begin{bmatrix} 1 & 0,0625 & 1,81 \cdot 10^{-5} & -3,77 \cdot 10^{-7} \\ 0 & 1 & 5,787 \cdot 10^{-4} & -1,81 \cdot 10^{-5} \\ 0 & 0 & 1 & -0,0625 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -0,0385 \cdot w^2 & 0 & 0 & 1 \end{bmatrix} = B$$

Para las estaciones 4,6 y 10 se tiene la siguiente multiplicación de matrices cuyo resultado se llama C, para un cálculo posterior.

$$\begin{matrix} Y \\ \varphi \\ M \\ Q \end{matrix} = \begin{bmatrix} 1 & 0,03125 & 4,52 \cdot 10^{-6} & -4,71 \cdot 10^{-8} \\ 0 & 1 & 2,89 \cdot 10^{-4} & -4,52 \cdot 10^{-6} \\ 0 & 0 & 1 & -0,03125 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -0,1927 \cdot w^2 & 0 & 0 & 1 \end{bmatrix} = C$$

Para la estación 5 se tiene la siguiente multiplicación de matrices cuyo resultado se llama D, para un cálculo posterior.

$$\begin{matrix} Y \\ \varphi \\ M \\ Q \end{matrix} = \begin{bmatrix} 1 & 0,025 & 2,89 * 10^{-6} & -2,41 * 10^{-8} \\ 0 & 1 & 231,5 * 10^{-6} & -2,89 * 10^{-6} \\ 0 & 0 & 1 & -0,025 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -0,9865 * w^2 & 0 & 0 & 1 \end{bmatrix} = D$$

Ahora se obtiene una matriz que se llamará E, y que es resultado de la multiplicación de matrices y que indica la unificación de los subsistemas en un solo sistema total.

$$E = C * B * B * B * C * D * C * B * B * B * A$$

Por cuestiones de espacio la matriz E no se coloca aquí, pero se aclara que esto no es problemático para resolución a través de software especializado en este tratamiento matemático como lo es matlab.

Dado que de acuerdo a la figura el eje esta simplemente apoyado para encontrar la incógnita w se iguala a cero la siguiente multiplicación de elementos de la matriz.

$$E(1,2) * E(3,4) - E(1,4) * E(3,2) = 0$$

La solución de la ecuación anterior da como resultado en los tres primeros valores más bajos así:

$$w_1 = 198.69 \left(\frac{rad}{seg} \right) \quad w_2 = 1699.13 \left(\frac{rad}{seg} \right) \quad w_3 = 3183.204 \left(\frac{rad}{seg} \right)$$

Con estas soluciones se va mostrar cómo se debe hallar el primer modo de vibración, que es la guía para hallar los siguientes:

De acuerdo a las ecuaciones de la sección 3.1.1 se puede obtener para el caso ejemplo lo siguiente:

$$\begin{bmatrix} Y \\ \varphi \\ M \\ Q \end{bmatrix}_0 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \frac{-E_{12}(w_1^2)}{E_{14}(w_1^2)} \end{bmatrix}_0$$

Del cual se obtiene que $Q=4992.4$

Con este valor se construye de nuevo las matrices de las estaciones, para el caso de la primera se tiene

$$\begin{matrix} Y \\ \varphi \\ M \\ Q \end{matrix} = \begin{bmatrix} 1 & 0,03125 & 4,52 * 10^{-6} & -4,71 * 10^{-8} \\ 0 & 1 & 2,89 * 10^{-4} & -4,52 * 10^{-6} \\ 0 & 0 & 1 & -0,03125 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 3322,2 \end{bmatrix}_0 = \begin{bmatrix} 0,031094 \\ 0,97743 \\ -156,012 \\ 4992,4 \end{bmatrix}$$

La respuesta de esta multiplicación de matrices sirve para el siguiente cálculo y se debe guardar para la gráfica respectiva

$$\begin{matrix} Y \\ \varphi \\ M \\ Q \end{matrix} = \begin{bmatrix} 1 & 0,0625 & 1,81 * 10^{-5} & -3,77 * 10^{-7} \\ 1,7193 * 10^{-2} & 1 & 5,787 * 10^{-4} & -1,81 * 10^{-5} \\ 59,183 & 0 & 1 & -0,0625 \\ -946,93 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0,031094 \\ 0,97743 \\ -156,012 \\ 4992,4 \end{bmatrix} = \begin{bmatrix} 0,0874 \\ 0,7976 \\ -310,55 \\ 4945,26 \end{bmatrix}$$

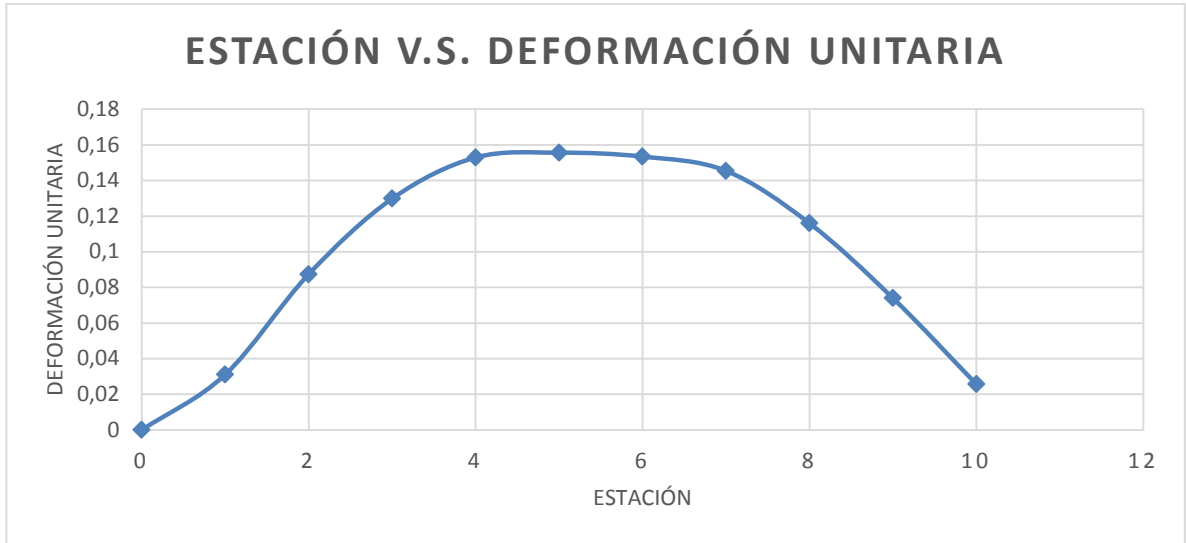
De nuevo se guarda los datos necesarios y se hace el siguiente cálculo.

$$\begin{matrix} Y \\ \varphi \\ M \\ Q \end{matrix} = \begin{bmatrix} 1 & 0,0625 & 1,81 * 10^{-5} & -3,77 * 10^{-7} \\ 1,7193 * 10^{-2} & 1 & 5,787 * 10^{-4} & -1,81 * 10^{-5} \\ 59,183 & 0 & 1 & -0,0625 \\ -946,93 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0,0874 \\ 0,7976 \\ -310,55 \\ 4945,26 \end{bmatrix} = \begin{bmatrix} 0,1298 \\ 0,5308 \\ -460,94 \\ 4812,39 \end{bmatrix}$$

Realizando este paso con todas las estaciones se obtiene en deformación unitaria los siguientes datos:

Estación	Deformación unitaria
0	0
1	0,031094
2	0,0874
3	0,1298
4	0,1529
5	0,1556
6	0,1534
7	0,1454
8	0,116
9	0,074
10	0,0257

Y si se grafican se obtiene:



ANEXO B. CÓDIGO EN MATLAB

```
function varargout = Interfaz_Grafica_Omega_Natural(varargin)
% INTERFAZ_GRAFICA_OMEGA_NATURAL MATLAB code for
Interfaz_Grafica_Omega_Natural.fig
%     INTERFAZ_GRAFICA_OMEGA_NATURAL, by itself, creates a new
INTERFAZ_GRAFICA_OMEGA_NATURAL or raises the existing
%     singleton*.
%
%     H = INTERFAZ_GRAFICA_OMEGA_NATURAL returns the handle to a new
INTERFAZ_GRAFICA_OMEGA_NATURAL or the handle to
%     the existing singleton*.
%
%
INTERFAZ_GRAFICA_OMEGA_NATURAL('CALLBACK', hObject, eventData, handles,...)
calls the local
%     function named CALLBACK in INTERFAZ_GRAFICA_OMEGA_NATURAL.M with
the given input arguments.
%
%     INTERFAZ_GRAFICA_OMEGA_NATURAL('Property','Value',...) creates a
new INTERFAZ_GRAFICA_OMEGA_NATURAL or raises the
%     existing singleton*. Starting from the left, property value pairs
are
%     applied to the GUI before
Interfaz_Grafica_Omega_Natural_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to
Interfaz_Grafica_Omega_Natural_OpeningFcn via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only
one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
Interfaz_Grafica_Omega_Natural

% Last Modified by GUIDE v2.5 02-Jul-2013 11:24:46

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @Interfaz_Grafica_Omega_Natural_OpeningFcn, ...
                  'gui_OutputFcn',  @Interfaz_Grafica_Omega_Natural_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
```

```

    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Interfaz_Grafica_Omega_Natural is made
% visible.
function Interfaz_Grafica_Omega_Natural_OpeningFcn(hObject, eventdata,
handles, varargin)
global apoyos Term1 Term2 Term3

axes(handles.axes1)
apoyo=imread('simple-simple.jpg');
axis off;
imshow(apoyo);%%% para ponerlo de fondo
set(handles.uitable1,'visible','off');
set(handles.uipanel7,'visible','off');
set(handles.uipanel8,'visible','off'); %%%%%% para ponerlo invisible

% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Interfaz_Grafica_Omega_Natural
% (see VARARGIN)

% Choose default command line output for Interfaz_Grafica_Omega_Natural
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Interfaz_Grafica_Omega_Natural wait for user response (see
UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Interfaz_Grafica_Omega_Natural_OutputFcn(hObject,
eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure

```

```

varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
global estaciones columnas filas datos apoyos Cond opcion W1 W2 W3 dx I E
m matrizdet Term1 Term2 Term3

if isnan(apoyos)
    errordlg('Favor seleccionar un tipo de apoyo');
end

if (isnan(estaciones))
    errordlg('Favor introducir el Número de Estaciones ');
end

if estaciones==0
    errordlg('El Número de estaciones no puede ser 0 ');
end

if estaciones<0
    errordlg('El Número de estaciones no puede ser negativo')
end

%%%%%%%%% tomando los datos suministrados por el usuario
dx1=datos(:,1);%%%%%%%%divisiones
D1=datos(:,2);%%%%%%%%diámetro de cada división
rho1=datos(:,3);%%%%%%%%densidad del material de la división para el
respectivo cálculo de masa
D_masa1=datos(:,4);%%%%%%%%diámetro de la amsa puntual diferente al diámetro
del eje
E=2.2*10^11; %[N/m] Modulo de young
m=zeros(estaciones+1,1);%%%%%%%%inicializando el vector masa
I1=zeros(estaciones+1,1);%%%%%%%%inicializando el vector de referencia de
inerencia
I=zeros(estaciones+1,1);%%%%%%%%inicializando el vector momento de inercia
rho=zeros(estaciones+1,1);%%%%%%%%inicializando el vector densidad
dx=zeros(estaciones+1,1);%%%%%%%%inicializando el vector dx
D_masa=zeros(estaciones+1,1);%%%%%%%%inicializando el vector de masa
Term1=zeros(estaciones+1,1);%%%%%%%%inicializando término 1
Term2=zeros(estaciones+1,1);%%%%%%%%inicializando término 2
Term3=zeros(estaciones+1,1);%%%%%%%%inicializando término 3

%%%%%%%%para saber si hay un error en la introducción en los datos%%%%%%%%%
for j=1:4 %%%%%barrido por las columnas
    for i=1:estaciones %%%%%barrido por las filas
        verificacion_datos=datos(i,j);
        if isnan(verificacion_datos) || (verificacion_datos<0) %%%%%condición
de parada del bucle for de filas
            errordlg('Favor introducir bien los datos pedidos en la
tabla'); %%%%%mensaje de error para que se introuzcan los datos de una
buena manera
            break
        end
    end
end

```

```

        end

    end
    verificacion_datos=datos(i,j);
    if isnan(verificacion_datos)|| (verificacion_datos<0)%%%%condición de
parada del bucle for de columnas
        errorldg('Favor introducir bien los datos pedidos en la tabla')
        break;%%%%parada bucle for de columnas

    end
end

for ii=1:estaciones

    I1(ii)=(pi*(D1(ii)^4))/64;
    A1(ii)=pi*D1(ii)^2/4;
end

%%%%%%para la determinación de los momentos de inercia y de las
%%%%%%verdaderas particiones
for iii=1:estaciones+1 %%%mas 1 poque el primer término la masa=0
    if iii==1
        D_masa(iii)=0;
        rho(iii)=rho1(1);
        dx(iii)=dx1(iii)/2;
        I(iii)=I1(iii);
        m(iii)=0;

    elseif iii==estaciones+1
        D_masa(iii)=D_masal(iii-1);
        rho(iii)=rho1(iii-1);
        dx(iii)=dx1(iii-1)/2;
        I(iii)=I1(iii-1);
        V=A1(iii-1)*dx1(iii-1);
        m(iii)=V*rho(iii);

    else

        D_masa(iii)=D_masal(iii-1);
        rho(iii)=rho1(iii-1);

        if D_masa(iii)~=0%%%%condición de si hay masa puntual
%           I_eje=I1(iii)/2+I1(iii-1)/2;
%           I(iii)=pi*(D_masa(iii)-D1(iii-1))^4/64+I_eje;%%%%sería una
opción
%           %%%la otra opción sería solo con al inercia del disco ya
que
%           %%% ya que se comportaría como un disco relleno totalmente
I(iii)=pi*D_masa(iii)^4/64;
%           %%%pero da diferente la inercia entonces no se cual
utilizar
%           %%%la que me mejor se acerque a la frecuencia natural real

```

```

        dx(iii)=dx1(iii)/2+dx1(iii-1)/2;
        V1=A1(iii-1)*dx1(iii-1);
        A_masa=pi*D_masa(iii)^2/4;
        V2=A_masa*dx1(iii-1);
        m(iii)=V1*rho(1)+V2*rho(iii);

    else
        dx(iii)=dx1(iii)/2+dx1(iii-1)/2;
        I(iii)=I1(iii)/2+I1(iii-1)/2;
        V=A1(iii-1)*dx1(iii-1);
        m(iii)=V*rho(iii);
    end
end

Term1(iii)=(dx(iii)/I(iii))*(1/E);
Term2(iii)=((dx(iii)^2)/I(iii))*(1/(2*E));
Term3(iii)=((dx(iii)^3)/I(iii))*(1/(6*E));

end

%Determinacion de una matriz que obtenga los rangos en los que se
%encuentran las 3 frecuencias naturales dentro de un rango de 1 rad/seg
w_ant=1; %[rad/s] Primera frecuencia a 1 [rad/s]
matriz1=zeros(4);
matriz2=zeros(4);
matriz3=zeros(4);

Det_ant=DeterminanteMatriz(w_ant,m,estaciones,dx,Term1,Term2,Term3,Cond,a
poyos); %Determinante a la primera frecuencia de prueba
w=1; %[rad/s] Segunda frecuencia de prueba
mm=0; %Numero creado para que se pueda entrar al "while" determina la
cantidad de frecuencias naturales que se quiere hallar
Ws=zeros(3,2); %Matriz que almacenara los rangos de las 3 frecuencias
naturales como [w1 w1;w2 w2;w3 w3]

while mm<3 %Halla 3 frecuencias naturales

    w=w+1; %[rad/s] Nueva frecuencia natural

    [Det,matrizdet]=DeterminanteMatriz(w,m,estaciones,dx,Term1,Term2,Term3,Co
nd,apoyos); %Determinante de la frecuencia natural analizada

    if (Det_ant>=0 && Det<0) || (Det>=0 && Det_ant<0) %Determina si cambia
de signo el denterminate
        mm=mm+1; %Recetea la variable cada vez que se encuentra una
frecuencia natural
        Ws(mm,:)= [w_ant w]; %Guarda el rango en el cual esta la frecuencia
natural
    end
end

```

```

        w_ant=w; %[rad/s] Para cada iteracion recetea la frecuencia anterior
        Det_ant=Det; %Determinante anterior se recetea por el determinante
nuevo

end

%Determinacion de las frecuencias naturales con una sensibilidad de
%0.01 [rad/s]
W=zeros(1,3); %[rad/s] Vector que obtiene las 3 frecuencias naturales
for jj=1:3%Determina las frecuencias naturales
    w_ant=Ws(jj,1); %[rad/s] primera frecuencia a analizar en cada caso

    [Det_ant,matrizdet]=DeterminanteMatriz(w_ant,m,estaciones,dx,Term1,Term2,
Term3,Cond,apoyos); %Primer determinante a analizar en cada caso
particular
    w=w_ant; %[rad/s] frecuencia natural a analizar en cada caso
    mm=0; % Numero para determinar si se encontro o no la frecuencia
natural
    while mm<1% Para cuando se halla la frecuencia natura

        w=w+0.001; %[rad/s] frecuencia en cada iteracion

        [Det,matrizdet]=DeterminanteMatriz(w,m,estaciones,dx,Term1,Term2,Term3,Co
nd,apoyos); %Determinante de la frecuencia natural analizada

        if (Det_ant>=0 && Det<0) || (Det>=0 && Det_ant<0)%Determina si
cambia de signo el denterminate

            mm=mm+1; %Recetea la variable cada vez que se encuentra una
frecuencia natural
            W(jj)=w_ant; %Guarda valor en el cual esta la frecuencia
natural

        end

        w_ant=w; %[rad/s] Para cada iteracion recetea la frecuencia
anterior
        Det_ant=Det; %Determinante anterior se recetea por el determinante
nuevo

    end
end
W;
W1=W(1); %%%%% frecuencia natural 1 en rad/s
W2=W(2); %%%%% frecuencia natural 2 en rad /s
W3=W(3); %%%%% frecuencia natural 3 en rad /S
%%%%%%%% 3 primeras frecuencia naturales en RPM%%%%%%%%
W1rpm=W1*60/(2*pi);
W2rpm=W2*60/(2*pi);
W3rpm=W3*60/(2*pi);
%%%%%%%%envio la tres primeras frecuencias naturales calculadas a la
interfaz
%%%%%%%%gráfica

```

```

set(handles.text12, 'string', W1rpm);
set(handles.text13, 'string', W2rpm);
set(handles.text14, 'string', W3rpm);
set(handles.uipanel7, 'visible', 'on');
set(handles.uipanel8, 'visible', 'on');

% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes on selection change in popupmenu1.
function popupmenu1_Callback(hObject, eventdata, handles)
global estaciones columnas filas datos apoyos Cond opcion
apoyos=get(handles.popupmenu1, 'value');
Cond=ones(4,2);%%%inicializando la condición de frontera
%%%%%%%% COLUMNA 1=INICIO   COLUMNA 2=FINAL %%%%%%%%%
%%%%%%%% FILA 1= FLECHA    FILA 2=THETA(DEFORMACIÓN ANGULAR) %%%%%%%%%
%%%%%%%% FILA 3= MOMENTO   FILA 3=CORTANTE %%%%%%%%%

if apoyos==1

axes(handles.axes1)
apoyo=imread('simple-simple.jpg');
axis off;
imshow(apoyo);
Cond(1,1)=0;%Condición de frontera simplemente apoyado en el inicio
flecha=0
Cond(3,1)=0;%Condición de frontera simplemente apoyado en el inicio
Momento=0
Cond(1,2)=0;%Condición de frontera simplemente apoyado en el final
flecha=0
Cond(3,2)=0;%Condición de frontera simplemente apoyado en el final
Momento=0

end
    if apoyos== 2
axes(handles.axes1)
apoyo=imread('empotrado-libre.jpg');
axis off;
imshow(apoyo);

Cond(1,1)=0;%Condición de frontera empotrado en el inicio flecha=0
Cond(2,1)=0;%Condición de frontera empotrado en el inicio theta =0
Cond(3,2)=0;%Condición de frontera libre en final momento=0
Cond(4,2)=0;%Condición de frontera libre en el final cortante=0
end

```

```

% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu1
contents as cell array
%         contents{get(hObject,'Value')} returns selected item from
popupmenu1

% --- Executes during object creation, after setting all properties.
function popupmenu1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%         str2double(get(hObject,'String')) returns contents of edit1 as a
double

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton2.

```

```

function pushbutton2_Callback(hObject, eventdata, handles)
global estaciones columnas filas datos apoyos cond opcion

%%%%%%%% para crear la tabla para ingresar los datos
estaciones=str2double(get(handles.edit1,'string'));
if (isnan(estaciones))
    errordlg('Favor introducir el Número de Estaciones');

elseif estaciones==0
    errordlg('El Número de estaciones no puede ser 0');

elseif estaciones<0
    errordlg('El Número de estaciones no puede ser negativo');
else

columnas=4;
filas=estaciones;
tabla=cell(filas,columnas);
tabla(:,:)={' '};
set(handles.uitable1,'Data',tabla)
set(handles.uitable1,'ColumnEditable',true(1,columnas))
set(handles.uitable1,'visible','on')
end

% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes when entered data in editable cell(s) in uitable1.

function uitable1_CellEditCallback(hObject, eventdata, handles)

% hObject    handle to uitable1 (see GCBO)
% eventdata  structure with the following fields (see Uitable)
%   Indices: row and column indices of the cell(s) edited
%   PreviousData: previous data for the cell(s) edited
%   EditData: string(s) entered by the user
%   NewData: EditData or its converted form set on the Data property.
Empty if Data was not changed
%   Error: error string when failed to convert EditData to appropriate
value for Data
% handles    structure with handles and user data (see GUIDATA)

% --- Executes when selected cell(s) is changed in uitable1.
function uitable1_CellSelectionCallback(hObject, eventdata, handles)

```

```

global estaciones columnas filas datos apoyos
datos=str2double(get(handles.uitable1,'Data'));%%%%% tomar los datos
suministrados pro el usuario

% hObject    handle to uitable1 (see GCBO)
% eventdata  structure with the following fields (see UITABLE)
%   Indices: row and column indices of the cell(s) currently selecteds
% handles     structure with handles and user data (see GUIDATA)

% --- Executes on selection change in popupmenu2.
function popupmenu2_Callback(hObject, eventdata, handles)
%%%propiedades de algunos materiales%%%
material=get(handles.popupmenu2,'value');
if material==1%%%%Acero
    rho=7850;
    %   E=210000;
    set(handles.edit2,'string',rho);

end
if material==2%%% Aluminio
    rho=2700;
    %   E=66500;
    set(handles.edit2,'string',rho);

end
if material==3%%%%%Hierro Forjado
    rho=7874;
    %   E=175000;
    set(handles.edit2,'string',rho);

end
% hObject    handle to popupmenu2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu2
contents as cell array
%   contents{get(hObject,'Value')} returns selected item from
popupmenu2

% --- Executes during object creation, after setting all properties.
function popupmenu2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns
called

% Hint: popupmenu controls usually have a white background on Windows.

```

```

%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%       str2double(get(hObject,'String')) returns contents of edit2 as a
double

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text
%       str2double(get(hObject,'String')) returns contents of edit3 as a
double

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.

```

```

%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% -----
function uitable1_ButtonDownFcn(hObject, eventdata, handles)
% hObject    handle to uitable1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)

global estaciones columnas filas datos apoyos cond opcion cond2 W1 W2 W3
dx I E m matriz_condicion flecha momento cortante distancia_eje matrizdet
Term1 Term2 Term3

if opcion==1
    Wnatural=W1;%%%para graficar con la primera frecuencia natural
elseif opcion==2
    Wnatural=W2;%%%para graficar con la segunda frecuencia natural
elseif opcion==3
    Wnatural=W3;%%%para graficar con la tercer frecuencia natural
else
    errordlg('favor seleccionar una frecuencia natural')
end

%%%%%%%%% apoyos
cond2=ones(4,1);%%%inicializamos la 2nda condición inicial
matriz_condicion=zeros(4,estaciones+1);%%%inicializamos la matriz
condición
if (apoyos==1)%%%simple-simple
    cond2(1,1)=0;%%%condición de flecha=0 para el apoyo simple
    cond2(2,1)=1;%%%suposición de condición de deformación angular=1
para apoyo simple
    cond2(3,1)=0;%%%condición de momento=0 para el apoyo simple

Tf=eye(4);%%%inicializando tf
Ts=eye(4);%%%inicializando ts
matriz_ant=zeros(4);%%%inicializando la matriz anterior
Tsn=eye(4);%%%inicializando tsn
Tf1=eye(4);%%%inicializando tfn

for jj=estaciones+1:-1:2

Tf(1,2)=dx(jj);
Tf(1,3)=Term2(jj);

```

```

Tf(1,4)=-Term3(jj);
Tf(2,3)=Term1(jj);
Tf(2,4)=-Term2(jj);
Tf(3,4)=-dx(jj);

Ts(4,1)=-Wnatural^2*m(jj);
Tf;
Ts;
Ti=Tf*Ts;
if jj==estaciones+1
    T=Ti;
else
    T=T*Ti;
end

end

Tf1(1,2)=dx(1);
Tf1(1,3)=Term2(1);
Tf1(1,4)=-Term3(1);
Tf1(2,3)=Term1(1);
Tf1(2,4)=-Term2(1);
Tf1(3,4)=-dx(1);
T=T*Tf1; %%%matriz T para hallar las condiciones
cond2(4,1)=-(T(1,2))/(T(1,4)); %%%condición inicial para el cortante

matriz_condicion=zeros(4,estaciones+1); %%%inicializamos la matriz
condición
matriz_condicion(:,1)=cond2; %%%para la condición inicial

Tf=eye(4); %%%inicializando tf por segunda vez para regresarnos
Ts=eye(4); %%%inicializando ts por segunda vez para regresarnos

for jj=2:estaciones+1
    if jj==2

        cond_siguiete=Tf1*cond2;
    else
        Tf(1,2)=dx(jj);
        Tf(1,3)=Term2(jj);
        Tf(1,4)=-Term3(jj);
        Tf(2,3)=Term1(jj);
        Tf(2,4)=-Term2(jj);
        Tf(3,4)=-dx(jj);

        Ts(4,1)=-Wnatural^2*m(jj);
        Tf;
        Ts;
        Ti=Tf*Ts;
        cond_siguiete=Ti*cond_siguiete;

    end
end

```

```

matriz_condicion(:,jj)=cond_siguiente;%%%ordenando en la matriz de
puntos a graficar

end
matriz_condicion;
flecha=matriz_condicion(1,:);
momento=matriz_condicion(3,:);
cortante=matriz_condicion(4,:);

%%% para calcular la distancia en el eje correspondiente a la división
longitud_eje=sum(dx);
distancia_eje=0:longitud_eje/(estaciones):longitud_eje;
graficas %%%nombre de la interfaz que muestra las gráficas

end

if (apoyos==2)%%%% empotrado-libre
    %%%primero inicializamos T,Ts,Tf,Tf1,Tsn

    T=eye(4);
    Ts=eye(4);
    Tsn=eye(4);
    Tf=eye(4);
    Tf1=eye(4);

    for jj=estaciones:-1:2

        %%%calculamos Tf en la posición jj
        Tf(1,2)=dx(jj);
        Tf(1,3)=Term2(jj);
        Tf(1,4)=-Term3(jj);
        Tf(2,3)=Term1(jj);
        Tf(2,4)=-Term2(jj);
        Tf(3,4)=-dx(jj);
        %%%calculamos Ts en la posición jj
        Ts(4,1)=-Wnatural^2*m(jj);
        %%%calculamos la matriz Ti
        Ti=Tf*Ts;

        if jj==estaciones
            T=Ti;
        else
            T=T*Ti;
        end

    end

    %%%ahora calculamos la matriz T, hallando Tsn y Tf1 o Tfo como la llama
    %%%el libro
    Tsn(4,1)=-Wnatural^2*m(estaciones+1);
    Tf1(1,2)=dx(1);
    Tf1(1,3)=Term2(1);

```

```

Tf1(1,4)=-Term3(1);
Tf1(2,3)=Term1(1);
Tf1(2,4)=-Term2(1);
Tf1(3,4)=-dx(1);
T=Tsn*T*Tf1;
%%%%%ahora teniendo T podemos definir las condiciones iniciales
cond2(1,1)=0;
cond2(2,1)=0;
cond2(3,1)=1;
cond2(4,1)=-T(3,3)/T(3,4);

%%la primera condicion en la matriz
matriz_condicion(:,1)=cond2;%%%%%%%%para la primera condición(la inicial)
%%for necesario para calcular las condiciones en cada punto del eje

for jj=2:estaciones+1
    if jj==2%%condición para la condición inicial
        cond_siguiente=Tf1*cond2;
        matriz_condicion(:,jj)=cond_siguiente;
    else

        %%%calculamos Tf en la posición jj
        Tf(1,2)=dx(jj);
        Tf(1,3)=Term2(jj);
        Tf(1,4)=-Term3(jj);
        Tf(2,3)=Term1(jj);
        Tf(2,4)=-Term2(jj);
        Tf(3,4)=-dx(jj);
        % %%%calculamos Ts en la posición jj
        Ts(4,1)=-Wnatural^2*m(jj);
        % %%%calculamos la matrizT
        Ti=Tf*Ts;
        cond_siguiente=Ti*cond_siguiente;
        matriz_condicion(:,jj)=cond_siguiente;
    end
end
matriz_condicion;
flecha=matriz_condicion(1,:);
momento=matriz_condicion(3,:);
cortante=matriz_condicion(4,:);
end

%%% para calcular la distancia en el eje correspondiente a la división
longitud_eje=sum(dx);
distancia_eje=0:longitud_eje/(estaciones):longitud_eje;
graficas %%%nombre de la interfaz que muestra las gráficas

% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB

```

```

% handles      structure with handles and user data (see GUIDATA)

% --- Executes when selected object is changed in uipanel9.
function uipanel9_SelectionChangeFcn(hObject, eventdata, handles)
global opcion

%%%% Para elegir la opción de frecuencia natural
if hObject==handles.w1
    opcion=1; %%%%para w1

elseif hObject==handles.w2

    opcion=2; %%%%para w2
else

    opcion=3; %%%%para w3
end

% hObject      handle to the selected object in uipanel9
% eventdata    structure with the following fields (see UIBUTTONGROUP)
%   EventName: string 'SelectionChanged' (read only)
%   OldValue:  handle of the previously selected object or empty if none
was selected
%   NewValue:  handle of the currently selected object
% handles      structure with handles and user data (see GUIDATA)

```

ANEXO C. REPORTE DEL SOFTWARE ANSYS

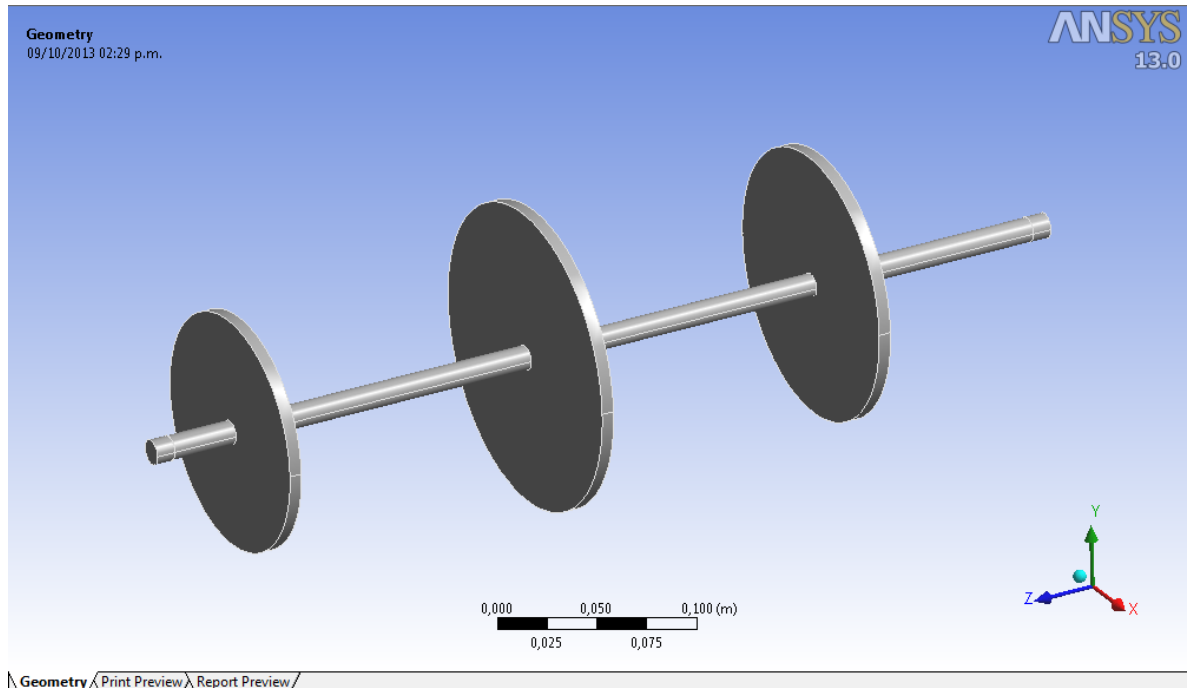


TABLE 1

Unit System	Metric (m, kg, N, s, V, A) Degrees rad/s Celsius
Angle	Degrees
Rotational Velocity	rad/s
Temperature	Celsius

TABLE 2
Model (A4) > Geometry

Object Name	Geometry
State	Fully Defined
Definition	
Source	C:\Users\deptechnikobu\Desktop\Trabajo PG\ansys\Pieza3.IGS
Type	Iges
Length Unit	Meters
Element Control	Program Controlled
Display Style	Part Color
Bounding Box	
Length X	0,16 m
Length Y	0,16 m
Length Z	0,5113 m
Properties	
Volume	3,9521e-004 m ³
Mass	3,1024 kg
Scale Factor Value	1,
Statistics	
Bodies	1
Active Bodies	1
Nodes	3767
Elements	1724
Mesh Metric	None
Preferences	
Import Solid Bodies	Yes
Import Surface Bodies	Yes
Import Line Bodies	No
Parameter Processing	Yes
Personal Parameter Key	DS
CAD Attribute Transfer	No
Named Selection Processing	No
Material Properties Transfer	No
CAD Associativity	Yes
Import Coordinate Systems	No
Reader Save Part File	No
Import Using Instances	Yes
Do Smart Update	No
Attach File Via Temp File	Yes
Temporary Directory	C:\Users\deptechnikobu\AppData\Local\Temp
Analysis Type	3-D
Mixed Import Resolution	None
Enclosure and Symmetry Processing	Yes

TABLE 3
Model (A4) > Geometry > Parts

Object Name	<i>Part 1</i>
State	Meshed
Graphics Properties	
Visible	Yes
Transparency	1
Definition	
Suppressed	No
Stiffness Behavior	Flexible
Coordinate System	Default Coordinate System
Reference Temperature	By Environment
Material	
Assignment	Structural Steel
Nonlinear Effects	Yes
Thermal Strain Effects	Yes
Bounding Box	
Length X	0,16 m
Length Y	0,16 m
Length Z	0,5113 m
Properties	
Volume	3,9521e-004 m ³
Mass	3,1024 kg
Centroid X	4,5916e-015 m
Centroid Y	-1,2744e-016 m
Centroid Z	-0,18752 m
Moment of Inertia Ip1	5,6032e-002 kg·m ²
Moment of Inertia Ip2	5,6036e-002 kg·m ²
Moment of Inertia Ip3	6,466e-003 kg·m ²
Statistics	

TABLE 4
Model (A4) > Coordinate Systems > Coordinate System

Object Name	<i>Global Coordinate System</i>
State	Fully Defined
Definition	
Type	Cartesian
Coordinate System ID	0,
Origin	
Origin X	0, m
Origin Y	0, m
Origin Z	0, m
Directional Vectors	
X Axis Data	[1, 0, 0,]
Y Axis Data	[0, 1, 0,]
Z Axis Data	[0, 0, 1,]

TABLE 5
Model (A4) > Mesh

Object Name	<i>Mesh</i>
State	Solved
Defaults	
Physics Preference	Mechanical
Relevance	0
Sizing	
Use Advanced Size Function	Off
Relevance Center	Coarse
Element Size	Default
Initial Size Seed	Active Assembly
Smoothing	Medium
Transition	Fast
Span Angle Center	Coarse
Minimum Edge Length	6,5e-003 m
Inflation	
Use Automatic Inflation	None
Inflation Option	Smooth Transition
Transition Ratio	0,272
Maximum Layers	5
Growth Rate	1,2
Inflation Algorithm	Pre
View Advanced Options	No
Advanced	
Shape Checking	Standard Mechanical
Element Midside Nodes	Program Controlled
Straight Sided Elements	No
Number of Retries	Default (4)
Extra Retries For Assembly	Yes
Rigid Body Behavior	Dimensionally Reduced
Mesh Morphing	Disabled
Defeaturing	
Pinch Tolerance	Please Define
Generate Pinch on Refresh	No
Automatic Mesh Based Defeaturing	On
Defeaturing Tolerance	Default
Statistics	
Nodes	3767
Elements	1724
Mesh Metric	None

TABLE 6
Model (A4) > Analysis

Object Name	<i>Modal (A5)</i>
State	Solved
Definition	
Physics Type	Structural
Analysis Type	Modal
Solver Target	Mechanical APDL
Options	
Environment Temperature	22, °C
Generate Input Only	No

TABLE 7
Model (A4) > Modal (A5) > Initial Condition

Object Name	<i>Pre-Stress (None)</i>
State	Fully Defined
Definition	
Pre-Stress Environment	None

TABLE 8
Model (A4) > Modal (A5) > Analysis Settings

Object Name	<i>Analysis Settings</i>
State	Fully Defined
Options	
Max Modes to Find	6
Limit Search to Range	No
Solver Controls	
Solver Type	Program Controlled
Output Controls	
Calculate Stress	No
Calculate Strain	No
Analysis Data Management	
Solver Files Directory	C:\Users\deptechnikobu\Desktop\Trabajo PG\ansys\fecuencia natural Ansys_files\dp0\SYS\MECH\
Future Analysis	None
Scratch Solver Files Directory	
Save MAPDL db	No
Delete Unneeded Files	Yes
Solver Units	Active System
Solver Unit System	mks

TABLE 9
Model (A4) > Modal (A5) > Loads

Object Name	<i>Fixed Support</i>
State	Fully Defined
Scope	
Scoping Method	Geometry Selection
Geometry	1 Face
Definition	
Type	Fixed Support
Suppressed	No

TABLE 10
Model (A4) > Modal (A5) > Solution

Object Name	<i>Solution (A6)</i>
State	Solved
Adaptive Mesh Refinement	
Max Refinement Loops	1,
Refinement Depth	2,
Information	
Status	Done

FIGURE 1
Model (A4) > Modal (A5) > Solution (A6)

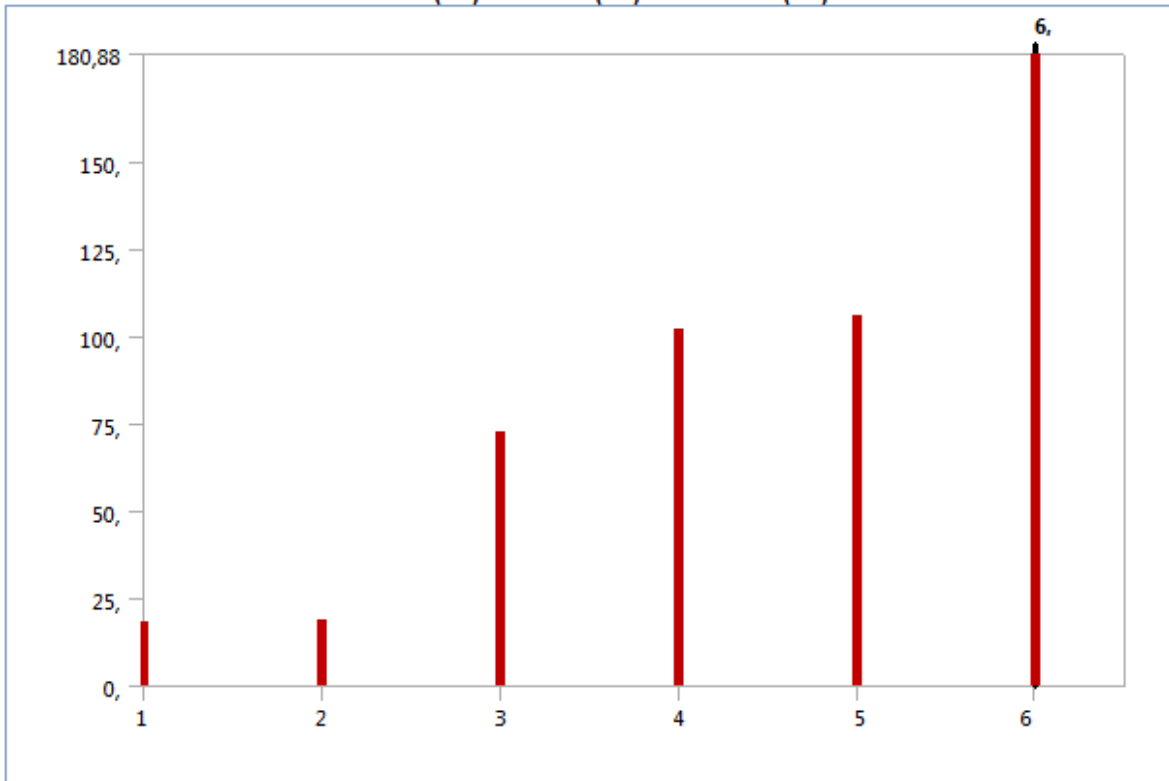


TABLE 11
Model (A4) > Modal (A5) > Solution (A6)

Mode	Frequency [Hz]
1,	18,142
2,	18,852
3,	72,413
4,	101,98
5,	105,9
6,	180,88