

ANÁLISIS DINÁMICO DE UN DIAPASÓN USANDO DMD (DYNAMIC MODE
DECOMPOSITION)

DANIEL ANDRÉS LEÓN ROMERO
LUIS DAVID FONTECHA GARCÍA

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS
ESCUELA DE INGENIERÍA MECÁNICA
BUCARAMANGA

2021

ANÁLISIS DINÁMICO DE UN DIAPASÓN USANDO DMD (DYNAMIC MODE
DECOMPOSITION)

DANIEL ANDRÉS LEÓN ROMERO
LUIS DAVID FONTECHA GARCÍA

Trabajo de grado para optar al título de Ingeniero Mecánico

Director

Jabid Eduardo Quiroga Méndez

PhD. en Ingeniería Civil

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS
ESCUELA DE INGENIERÍA MECÁNICA
BUCARAMANGA

2021

DEDICATORIA

A Dios por brindarme sabiduría, fortaleza y perseverancia para enfrentar cada instante de mi proceso académico.

A mi madre Martha Romero, padre Daniel Alberto León y hermana Leidy Johana León, por ser ese apoyo incondicional e impulsarme a alcanzar este gran logro.

A mi novia Diana Vivas por su apoyo, comprensión y paciencia en los momentos difíciles.

A mis demás familiares y amigos que creyeron en mí y aportaron en gran medida para que esto fuera posible.

Daniel Andrés León Romero

DEDICATORIA

A Dios por darme salud, fortaleza y permitirme salir adelante en todo momento.

A mi madre por sus consejos y por su acompañamiento constante,

A mi padre y a mis tres hermanas por su cariño y apoyo incondicional.

*A mi hermano Erik Fontecha el cual fue mi motivación más grande durante mi
proceso de formación*

Luis David Fontecha García

AGRADECIMIENTOS

A nuestro director, profesor PhD. Jabid Eduardo Quiroga Méndez por brindarnos su conocimiento, gran disposición, tiempo y valiosa orientación, lo cual fue fundamental para llevar a cabo este proyecto académico.

A los profesores de la Escuela de Ingeniería Mecánica de la Universidad Industrial de Santander, por ser nuestros guías en este proceso de formación y sus consejos para enfrentar los nuevos retos en la vida profesional.

A nuestros familiares y amigos que siempre nos apoyaron estando allí cuando más los necesitamos.

Daniel Andrés León Romero

Luis David Fontecha García

CONTENIDO

	Pág.
INTRODUCCIÓN	20
1. FORMULACIÓN DEL PROBLEMA	22
1.1 IDENTIFICACIÓN DEL PROBLEMA	22
1.2 JUSTIFICACIÓN PARA SOLUCIONAR EL PROBLEMA	23
2. OBJETIVOS DEL PROYECTO	25
2.1 OBJETIVO GENERAL	25
2.2 OBJETIVOS ESPECÍFICOS.....	25
3. MARCO REFERENCIAL	26
3.1 GENERALIDADES	26
3.2 DIAPASÓN ACÚSTICO	27
3.3 SINGULAR VALUE DECOMPOSITION (SVD).....	30
3.4 DYNAMIC MODE DECOMPOSITION (DMD).....	33
3.4.1 Usos e interpretaciones del DMD	34
3.4.2 Ventajas del DMD.	35
3.4.3 Formulación de la técnica DMD	35
3.4.4 Características de la organización de la información	37
3.4.5 Desarrollo de la técnica DMD	38
3.4.6 Algoritmo DMD	38
3.4.7 Matriz de Hankel	42
3.5 ESTADO DEL ARTE.....	43
4. MODELADO ESTRUCTURAL Y SIMULACIÓN	46
4.1 MODELO 3D DEL DIAPASÓN.	46
4.2 ANÁLISIS MODAL	47
4.2.1 Modelo y propiedades	47
4.2.2 Mallado para el análisis modal	48

4.2.3	Función solución	48
4.3	ANÁLISIS TRANSITORIO	49
4.3.1	Modelo y propiedades	49
4.3.2	Mallado para el análisis transitorio	49
4.3.3	Condiciones iniciales	50
4.3.4	Función solución	52
4.4	LINEALIDAD Y NO LINEALIDAD DEL DIAPASÓN.	52
5.	IMPLEMENTACIÓN DEL ALGORITMO DMD AL DIAPASÓN	57
5.1	DMD NODOS INDIVIDUALES.....	57
5.1.1	DMD estándar	58
5.1.2	DMD usando la matriz de Hankel.....	59
5.1.3	DMD implementando error cuadrático medio	60
5.2	DMD DIAPASÓN COMPLETO	63
5.2.1	Datos de entrada al algoritmo	63
5.2.2	Funciones características DMD.....	63
5.2.3	Algoritmo DMD diapason completo	65
6.	RESULTADOS Y VALIDACIÓN	69
6.1	ANÁLISIS MODAL	69
6.2	ANÁLISIS TRANSITORIO.	71
6.3	DMD NODOS INDIVIDUALES.....	73
6.3.1	Algoritmos característicos para la reconstrucción.	73
6.3.2	DMD estándar	75
6.3.3	DMD usando la matriz de Hankel.....	77
6.3.4	DMD implementando error cuadrático medio	80
6.4	DMD diapason completo.....	87
6.4.1	Frecuencia de vibración del diapason	89
7.	CONCLUSIONES.....	96
	BIBLIOGRAFÍA	98
	ANEXOS	101

LISTA DE FIGURAS

	Pág.
Figura 1. Cantidad de datos reales y previstos en el mundo por año.....	23
Figura 2. Diapasón de sección circular.	27
Figura 3. Movimiento simétrico de las puntas sobre su mismo plano al oscilar en la frecuencia fundamental.	28
Figura 4. Desplazamiento transversal de las puntas para el modo fundamental. .	29
Figura 5. Representación de desplazamiento axial en el diapasón.....	29
Figura 6. Esquema de las matrices SVD sin y con solución económica.	33
Figura 7. Breve descripción paso a paso de la técnica DMD.	42
Figura 8. Análisis bibliométrico para palabras clave dynamic mode decomposition AND dynamic systems.	44
Figura 9. Tendencia de citas que ha recibido J. Nathan Kutz.....	44
Figura 10. Tendencia de citas que ha recibido Steven L. Brunton.....	45
Figura 11. Análisis bibliométrico para palabras clave dynamic mode decomposition AND mechanical vibrations.	45
Figura 12. Modelo 3D del diapasón creado como un archivo de extensión “.stl”.	46
Figura 13. Mallado para el análisis modal.	48
Figura 14. Mallado para el análisis transitorio.	50
Figura 15. Ubicación de una de las dos caras fijadas para las condiciones de contorno.	51
Figura 16. Cara sobre la cual se aplica la carga de excitación.....	52
Figura 17. Principio de superposición.	53
Figura 18. Principio de homogeneidad.....	53
Figura 19. Comparación de desplazamientos para el nodo 24.	54
Figura 20. Comparación de desplazamientos, nodo 136.	55
Figura 21. Comparación de desplazamientos, nodo 11.	56

Figura 22. Representación gráfica de los modos flexibles determinados.....	70
Figura 23. Desplazamientos en el eje y para todos los nodos del diapasón y 6,000 instantes de tiempo.	72
Figura 24. Acercamiento de la figura 23.....	72
Figura 25. Desplazamientos en el eje x para todos los nodos del diapasón.....	73
Figura 26. Representación datos originales y datos DMD para la cara 11.....	75
Figura 27. Representación de los valores propios en un círculo unitario para la cara 11.	76
Figura 28. Desplazamientos datos originales y datos DMD nodo 166.	77
Figura 29. Desplazamientos datos originales y datos DMD nodo 43.	77
Figura 30. Acercamiento de la figura 29.....	78
Figura 31. Desplazamientos datos originales y datos DMD nodo 24.	78
Figura 32. FFT datos originales y datos DMD nodo 166.	79
Figura 33. FFT datos originales y datos DMD nodo 43.	80
Figura 34. FFT datos originales y datos DMD nodo 24.	80
Figura 35. Error calculado para el nodo 11.	81
Figura 36. Error calculado para el nodo 158.	81
Figura 37. Error calculado para el nodo 234.	82
Figura 38. Desplazamientos datos originales y datos DMD nodo 11.	82
Figura 39. Acercamiento de la figura 38 para un intervalo de tiempo de 0.01 a 0.02 segundos.....	83
Figura 40. Acercamiento de la figura 38 para un intervalo de tiempo de 0.055 a 0.065 segundos.....	83
Figura 41. Desplazamientos datos originales y datos DMD nodo 158.	84
Figura 42. Desplazamientos datos originales y datos DMD nodo 234.	84
Figura 43. Acercamiento de la figura 42.....	85
Figura 44. FFT datos originales y datos DMD nodo 11.	86
Figura 45. FFT datos originales y datos DMD nodo 158.	86
Figura 46. FFT datos originales y datos DMD nodo 234.	87

Figura 47. Representación 3D del diapasón construido con datos originales y con datos DMD.	88
Figura 48. Valor propio en el plano complejo con magnitud limitada por una circunferencia unitaria.	91
Figura 49. Valores propios para el nodo 1,450 ubicados sobre una circunferencia unitaria.	92
Figura 50. Dispersión de la frecuencia para cada nodo elegida con el criterio de círculo unitario con mínimo y máximo de Hankel de 10 a 200 respectivamente. ...	93
Figura 51. Valores atípicos.....	94
Figura 52. Nueva dispersión de la frecuencia para cada nodo elegida con el criterio de círculo unitario con mínimo y máximo de Hankel de 200 a 400 respectivamente.	95

LISTA DE CUADROS

	Pág.
Cuadro 1. Propiedades mecánicas generalmente usadas para el acero.	47
Cuadro 2. Nodos seleccionados para implementar la técnica DMD.....	57
Cuadro 3. Datos de entrada algoritmo DMD completo.....	63
Cuadro 4. Modos de frecuencia obtenidos por el análisis modal y sus valores aproximados.....	69

LISTA DE ANEXOS

	Pág.
Anexo A. Funciones complementarias para los algoritmos DMD.....	101
Anexo B. Detalle dimensional del diapasón.	107
Anexo C. Algoritmo para el análisis modal del diapasón.....	108
Anexo D. Algoritmo para el análisis transitorio del diapasón.....	109
Anexo E. Algoritmo DMD con entrada de un valor n_{ajuste}	111
Anexo F. Algoritmo DMD con error cuadrático medio.	112
Anexo G. Algoritmo DMD diapasón completo.	114
Anexo H. Algoritmo para la reconstrucción 3D del diapasón con datos originales y con datos DMD.....	116

GLOSARIO

ARREGLO MATRICIAL: está dado por una matriz M , y consiste en una organización rectangular o cuadrada de datos, dados por m cantidad de vectores fila o n cantidad de vectores columna, dando lugar a un arreglo $m \times n$.

COMBINACIÓN LINEAL: se presenta como la suma de pares de elementos (escalar y un vector) multiplicados entre sí que producen un vector. Para el caso de dos matrices, se obtiene que cada columna de la matriz resultante es una combinación lineal de las columnas de M_A . Entonces, cualquier vector puede reescribirse como una combinación lineal de otros vectores que tengan dirección diferente.

DEPENDENCIA LINEAL: dada una cantidad n de vectores, se dicen que son linealmente dependientes si existe una combinacional lineal además de la trivial, que dé como resultado el vector 0.

ECUACIONES DIFERENCIALES PARCIALES (EDP): tienen como objetivo realizar la búsqueda de una función desconocida f o y , que cumpla las características de la ecuación, es decir, al derivar la función encontrada respecto a su variable independiente, se obtenga la igualdad de la ecuación diferencial. Son escritas a modo de derivadas parciales, donde también su solución consiste en encontrar una función desconocida u o y , la cual es dependiente de más de una variable independiente.

ESPACIO VECTORIAL: es un conjunto de objetos conocidos como vectores u (n componentes), que traen consigo dos operaciones conocidas como suma y multiplicación por un escalar y que además cumplen con las propiedades o axiomas

ya establecidos. A todos los elementos que conforman un espacio vectorial se les conoce como vectores y a los elementos que conforman el cuerpo, escalares.

INDEPENDENCIA LINEAL: para una cantidad n de vectores, estos no son dependientes si la única combinación lineal que da como resultado al vector nulo ($\vec{0}$), es la solución trivial y estos no pueden ser reescritos como combinación lineal de los restantes. Estos vectores tienen distinta dirección y las componentes no son proporcionales. Si M_A es una matriz invertible, entonces las columnas y filas forman un conjunto de vectores linealmente independientes.

MATRIZ COMPLEJA: es aquella matriz en la que por lo menos alguno de sus datos es un número complejo ($\alpha + \beta i$), es decir, la matriz pertenece a \mathbb{C} .

MATRIZ CONJUGADA: es una matriz en la que sus elementos dados por números complejos, se reescribe la parte imaginaria usando el signo contrario. Generalmente se denota como $\overline{M_A}$.

MATRIZ DIAGONAL: corresponde a una matriz cuadrada en la cual todos los valores exceptuando los ubicados en la diagonal principal, son nulos. Algunos de los datos en la diagonal principal pueden ser nulos. Se diferencia de la matriz identidad en que los valores en la diagonal generalmente son diferentes de 1. Una matriz diagonal es también una matriz simétrica y triangular.

MATRIZ IDENTIDAD: es una matriz de dimensión $m \times m$, en la cual los elementos de la diagonal principal son iguales a 1 y todos los demás (triángulo superior e inferior) son iguales a 0. Conocida como el elemento neutro del producto matricial se cumple la propiedad conmutativa $M_A * I = I * M_A$.

MATRIZ INVERSA: corresponde a una matriz no singular $m \times m$ que al ser multiplicada por la matriz original (también cuadrada), se obtiene como resultado la

matriz identidad I_m , la cual cuenta con la misma dimensión ($M_A * M_A^{-1} = M_A^{-1} * M_A = I$). Para una matriz que es invertible, se cumple el concepto de unicidad, es decir, la inversa (M_A^{-1}) de una matriz es única.

MATRIZ ORTOGONAL: Una matriz es ortogonal si esta es una matriz cuadrada y su inversa coincide con la transpuesta y los vectores fila o vectores columna forman un conjunto de vectores ortonormales (definido por producto interno). Además, se puede afirmar que dos matrices son ortogonales si se cumple que su producto interno es igual a 0.

MATRIZ ORTONORMAL: se dice que dos matrices son ortonormales si estas, además de tener un producto interno igual a 0 (ortogonalidad), la norma o módulo de cada vector que compone a la matriz es igual a 1.

MATRIZ POR UN ESCALAR: se multiplica una matriz M_A por un escalar α , el cual incrementa en α -veces cada dato de la matriz inicial.

MATRIZ PSEUDOINVERSA: generalmente denotada como M_A^\dagger , la pseudoinversa de Moore-Penrose, es una matriz M_A que satisface las llamadas condiciones de Penrose.

MATRIZ TRANSPUESTA: es el resultado de transformar los vectores fila de una matriz $m \times n$ en vectores columna, formando así una matriz de $n \times m$. Se denota como M_A^T o M_A' .

MATRIZ TRANSPUESTA CONJUGADA COMPLEJA: conocida también como matriz adjunta (M_A^*), se produce al obtener la transpuesta de la matriz y luego proceder a desarrollar su conjugada. Si los elementos de la matriz fueran reales, su resultado sería sólo la matriz transpuesta.

MATRIZ UNITARIA: hace referencia a una matriz cuadrada compleja que al ser multiplicada por su matriz transpuesta conjugada da como resultado la matriz identidad I , cumpliendo la propiedad conmutativa del producto matricial ($M_A * M_A^* = M_A^* * M_A = I$). Por lo tanto, la inversa de una matriz unitaria, es la transpuesta conjugada.

PRODUCTO MATRICIAL: una matriz M_A de $m \times n$ puede multiplicarse por otra matriz M_B de $p \times q$, si y sólo si $n = p$, dando como resultado un arreglo $m \times p$ al operar filas de M_A con columnas de M_B cumpliendo una serie de propiedades.

RANGO DE UNA MATRIZ: corresponde al número de filas o columnas que son linealmente independientes. El rango (r) otorga la dimensión de la matriz una vez eliminadas todas las filas o columnas linealmente dependientes de la matriz original. El máximo rango posible de una matriz, corresponde a la dimensión más pequeña de la misma.

SISTEMAS DINÁMICOS: puede definirse un sistema como el “conjunto de componentes interconectados que presenta ciertas relaciones de causa y efecto, las cuales actúan con un determinado objetivo. Los sistemas dinámicos hacen referencia al análisis, predicción y comprensión del comportamiento de sistemas de ecuaciones diferenciales o mapeos iterativos que representan la evolución del estado de un sistema en el tiempo.

SISTEMAS DINÁMICOS DISCRETOS: en los sistemas discretos el tiempo t sólo toma valores discretos en ciertos puntos de la escala numérica real, a menudo estos instantes de tiempo se encuentran espaciados uniformemente con una separación Δt . El modelado computacional discretiza sistemas continuos para convertirlos en su lenguaje de operación.

TRANSFORMACION LINEAL: cualquier matriz representa una transformación lineal entre dos espacios vectoriales (dominio y co-dominio) de dimensión finita. La transformación T es una función que asigna a cada vector del dominio \mathbb{R}^n un vector único del rango \mathbb{R}^m que hace parte del co-dominio. Al ser lineal cumple con las propiedades de superposición.

TRANSFORMADA RÁPIDA DE FOURIER (FFT): es un algoritmo que permite simplificar el cálculo de la (transformada discreta de Fourier) DFT y su inversa. Generalmente la DFT representa una transformación o función matemática, mientras FFT es una familia específica de algoritmos para calcular las DFTs.

VALORES PROPIOS: conocidos también como valores característicos o eigenvalores, corresponden a un número real o complejo λ si existe un vector no nulo (diferente de 0) en \mathbb{C}^n , que satisfaga la ecuación $M_A * x = \lambda * x$, donde M_A , es una matriz cuadrada y que corresponde a una transformación lineal T , tal que $T * x$ y x son paralelos.

VECTOR COLUMNA: corresponde a un conjunto de n datos organizados y escritos de manera vertical (uno debajo de otro) y donde n corresponde a su número de componentes.

VECTOR FILA: corresponde a un conjunto de m datos organizados y escritos en forma de renglón y donde m corresponde a su número de componentes.

VECTORES PROPIOS: al igual que la definición de los valores característicos, un vector propio es aquél vector x que satisface la igualdad $M_A * x = \lambda * x$, Corresponde a un vector que se queda en su propio subespacio generado durante la transformación. Si el valor propio es un número complejo, el vector propio será un vector complejo.

RESUMEN

TÍTULO: ANÁLISIS DINÁMICO DE UN DIAPASÓN USANDO DMD (DYNAMIC MODE DECOMPOSITION)*

AUTOR: DANIEL ANDRÉS LEÓN ROMERO, LUIS DAVID FONTECHA GARCÍA**

PALABRAS CLAVE: DMD, VIBRACIONES MECÁNICAS, DIAPASÓN ACÚSTICO, MATRIZ DE HANKEL, ANÁLISIS BASADO EN DATOS, SISTEMAS DINÁMICOS.

DESCRIPCIÓN: El presente proyecto describe la metodología para la implementación del algoritmo dynamic mode decomposition (DMD) en un diapasón acústico que es excitado en un instante de tiempo corto, con el propósito de validar la adaptabilidad del algoritmo en el campo de las vibraciones mecánicas y de generar información útil para posteriores investigaciones que se fundamenten en análisis basados en datos, que tengan como objeto identificar la dinámica de sistemas pertenecientes a esta área.

Inicialmente se presentan conceptos teóricos que son fundamentales para la interpretación del comportamiento del diapasón como sistema dinámico, así como la descripción, formulación (estándar y variante usada para el desarrollo del presente proyecto), interpretación y usos habituales del algoritmo DMD.

En segundo lugar, se explica de manera detallada el proceso para la creación del modelo, el cual es necesario para realizar el análisis modal y transitorio en MATLAB. Se presenta también el método de verificación del comportamiento lineal o no lineal en el diapasón. Posteriormente, se realiza la implementación del algoritmo DMD para diferentes casos establecidos, que van desde la aplicación en nodos individuales, así como en la totalidad de los nodos del diapasón que presentan libertad para moverse.

Finalmente, se presentan los resultados obtenidos y sus respectivas validaciones para cada caso en el que fue implementado el algoritmo. Además, se muestran las respectivas conclusiones producto del cumplimiento de los objetivos planteados en este proyecto de investigación.

* Proyecto de grado

** Facultad de Ingenierías Fisicomecánicas. Escuela de Ingeniería Mecánica. Director: Jabid Eduardo Quiroga Méndez, PhD en Ingeniería Civil.

ABSTRACT

TITLE: DYNAMIC ANALYSIS OF A TUNING FORK USING DMD (DYNAMIC MODE DECOMPOSITION)*

AUTHOR: DANIEL ANDRÉS LEÓN ROMERO, LUIS DAVID FONTECHA GARCÍA**

KEY WORDS: DMD, MECHANICAL VIBRATIONS, TUNING FORK, HANKEL MATRIX, DATA-BASED ANALYSIS, DYNAMIC SYSTEMS.

DESCRIPTION: The present project describes the methodology for the implementation of the dynamic mode decomposition (DMD) algorithm in an acoustic tuning fork that is excited in a short time, in order to validate the adaptability of the algorithm in the field of mechanical vibrations and to generate useful information for subsequent investigations that are based on data-based analysis, whose objective is to identify the dynamics of systems belonging to this area.

Initially, theoretical concepts are presented that are fundamental for the interpretation of the behavior of the tuning fork as a dynamic system, as well as the description, formulation (standard and variant used for the development of this project), interpretation and common uses of the DMD algorithm.

Second, the process for creating the model, which is necessary to perform modal and transient analysis in MATLAB, is explained in detail. The method of verification of linear or non-linear behavior on the tuning fork is also presented. Subsequently, the implementation of the DMD algorithm is carried out for different established cases, ranging from the application in individual nodes, as well as in all the nodes of the tuning fork that present freedom to move.

Finally, the results obtained and their respective validations are presented for each case in which the algorithm was implemented. In addition, the respective conclusions resulting from the fulfillment of the objectives set out in this research project are shown.

* Bachelor Thesis

** Physicomechanical Faculty of Engineering. School of Mechanical Engineering. Director: Javid Quiroga Méndez, PhD in Civil Engineering.

INTRODUCCIÓN

IBM¹ pronostica que para el año 2025, habrán más de 150 zettabytes de datos entre estructurados y no estructurados que requerirán ser analizados.

Con el gran auge de la ciencia de datos y debido a la gran necesidad de analizar y comprender los fenómenos reales que evolucionan en el tiempo, poco a poco en la última década se ha dejado a un lado el paradigma de que son necesarios modelos, ecuaciones o grandes algoritmos para extraer conocimiento de estos sistemas.

Gran parte de los sistemas dinámicos que representan fenómenos reales, presentan alta complejidad como: no linealidades, difícil modelado computacional y/o desconocimiento de las ecuaciones subyacentes de movimiento. Por tal razón y a raíz de esta problemática, investigadores han estado en la ardua tarea de encontrar alternativas de solución, siendo una de ellas la técnica dynamic mode decomposition (DMD), debido a su poder para reconstruir la dinámica de sistemas altamente complejos al aplicar un análisis puramente basado en datos dependientes del tiempo.

La importancia de esta técnica ha escalado de tal forma que han surgido una serie de variaciones implementadas para diferentes aplicaciones según el grado de complejidad, como: climatología, neurociencia, fluidos turbulentos, robótica, finanzas; entre otros. Sin embargo, en el campo de las vibraciones mecánicas es poco lo que se ha investigado a la fecha.

En consecuencia y con el propósito de un primer acercamiento a esta técnica y al área de las vibraciones mecánicas, se plantea el objetivo principal de este proyecto académico como la implementación del algoritmo DMD a un diapasón acústico, que

¹ IBM. How to manage complexity and realize the value of big data [Sitio web]. [Consulta: 09 de abril 2021]. Disponible en: <https://cutt.ly/KvqQlr5>

es golpeado de tal forma que vibre en su modo fundamental. Lo anterior con el propósito de reconstruir su dinámica, a partir del análisis de los datos de deflexión obtenidos con el software MATLAB v.2019b, de lo cual se espera obtener y validar que la flexibilidad de la técnica DMD efectivamente permite que sea aplicada al campo de las vibraciones mecánicas.

1. FORMULACIÓN DEL PROBLEMA

1.1 IDENTIFICACIÓN DEL PROBLEMA

Actualmente, analizar o evaluar el estado en el tiempo de gran parte de los sistemas dinámicos resulta ser un problema de consideración, debido a la alta complejidad de las ecuaciones que los gobiernan o porque su dinámica es desconocida. Otro factor que obstaculiza el análisis, es la no linealidad al no cumplirse el principio de superposición, lo que dificulta su modelado o definición de ecuaciones de gobierno, a raíz del caos o los comportamientos impredecibles que puedan presentar.

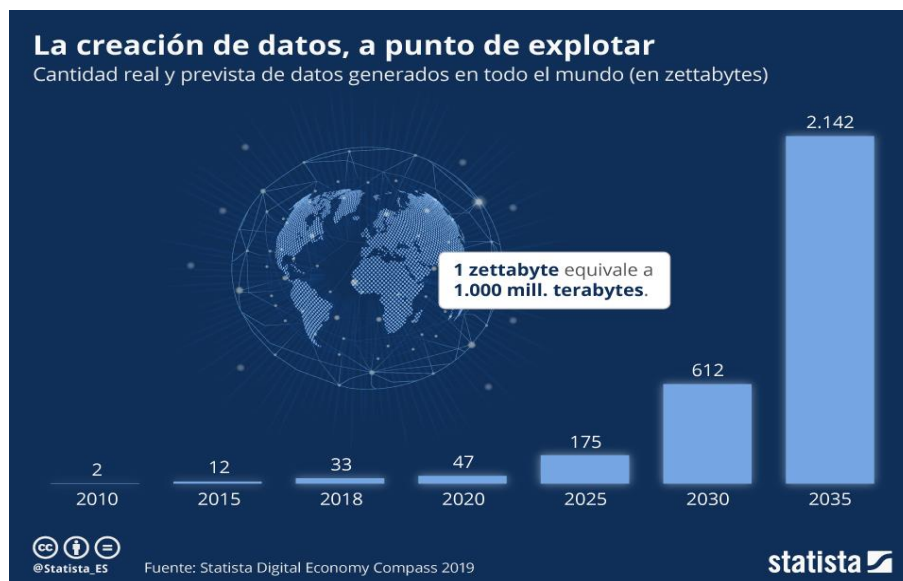
Debido a que gran parte de los sistemas dinámicos presentan una alta dimensionalidad a causa de la gran cantidad de variables dependientes del tiempo, estos son analizados a través de dinámica experimental, la cual puede llevarse a cabo por medio de experimentos en laboratorios y/o simulaciones computacionales. Al realizar la simulación para definir la dinámica de un sistema físico (generalmente de tiempo continuo), se pueden generar gran cantidad de mediciones para cada instante de tiempo resultando en millones de datos en total, lo cual es un problema debido al alto consumo de recursos computacionales y de tiempo de trabajo requeridos.

El algoritmo para la descomposición en modos dinámicos (DMD) nace y generalmente es aplicado en el área de la mecánica de fluidos, sin embargo, la información presente en las principales bases de datos científicas para el campo de las vibraciones mecánicas es muy reducida o casi nula, por esta razón se considera adecuado tomar como punto de partida el implementar dicho algoritmo en un diapasón acústico con fines académicos generando información útil para posteriores investigaciones.

1.2 JUSTIFICACIÓN PARA SOLUCIONAR EL PROBLEMA

En el afán por lograr describir e interpretar la evolución del estado de un sistema, el análisis basado en datos ha sido una gran alternativa que ha puesto a nuestra disposición la ciencia e ingeniería, gracias a profundas bases matemáticas y a los claros avances tecnológicos en materia computacional y de experimentación. Este cambio de paradigma se presenta, dado que los datos son cada vez más abundantes, mientras que las ecuaciones o leyes físicas en áreas como: la mecánica de fluidos, la óptica, la combustión, las finanzas, neurociencia, epidemiología, predicción del clima; entre otros, siguen siendo esquivas a medida que su estudio se vuelve más realista.

Figura 1. Cantidad de datos reales y previstos en el mundo por año.



Fuente. STATISTA, A la espera de un Big Bang de datos. [sitio web]. Fecha de publicación: 17 de abril del 2019. [Consulta: 10 de enero del 2021]. Disponible en: <https://bit.ly/3shYhIT>.

La descomposición en modos dinámicos (DMD), es un algoritmo de reducción de dimensionalidad novedoso, del cual de su aplicación en Colombia existe muy poca

información. Se desarrolló para el campo de la mecánica de fluidos, pero ha tomado importancia para el estudio de diferentes tipos de sistemas dinámicos lineales y no lineales. Al encontrarse con grandes cantidades de datos medidos por experimentación o producto de simulaciones computacionales, la DMD permite obtener modos espacio-temporales al reducir de forma significativa estos datos, conservando sólo los más dominantes (mayor contenido energético espacialmente correlacionados) que a diferencia de métodos como el POD o SVD, permite determinar la evolución en el tiempo del estado de un sistema.

En las bases de datos más reconocidas a nivel mundial, es casi nula la información que se encuentra sobre la implementación del algoritmo DMD en el estudio de las vibraciones mecánicas, en el cual pueden obtenerse gran cantidad de mediciones para cada instante de tiempo. En ese sentido, el implementar la técnica DMD en un diapasón acústico, cuyas mediciones se obtienen a través de simulación computacional, representa un punto de partida para mostrar las ventajas del algoritmo DMD en el campo de los fenómenos vibratorios.

2. OBJETIVOS DEL PROYECTO

2.1 OBJETIVO GENERAL

Realizar un estudio basado en datos para las mediciones de deflexión obtenidas por la simulación computacional de un diapasón acústico, con el fin de verificar la técnica de Descomposición en Modos Dinámicos (DMD) al determinar patrones de actividad dinámica mediante el software MATLAB.

2.2 OBJETIVOS ESPECÍFICOS

- Generar un modelado estructural haciendo uso del módulo Partial Differential Equation (PDE) que incluye el Toolbox de MATLAB para el análisis modal y transitorio de un diapasón acústico, que permita simular su comportamiento al ser golpeado en una de sus caras, con el fin de obtener grandes cantidades de datos para cada instante de tiempo establecido y proceder a su posterior organización, según los criterios del algoritmo DMD.
- Aplicar la técnica de descomposición en modos dinámicos (DMD), con el fin de obtener del diapasón los modos dinámicos traducidos en frecuencias naturales, a partir de la extracción de patrones dominantes, representados por los valores y los vectores propios de las matrices generadas.
- Validar la implementación del algoritmo DMD, al comparar la frecuencia natural dominante obtenida por medio de la simulación y la frecuencia determinada a través del análisis basado en datos aplicando la Transformada Rápida de Fourier (FFT).

3. MARCO REFERENCIAL

El marco referencial se caracteriza por recopilar la información consultada, la cual gira en torno al proyecto y es necesaria para el desarrollo del mismo, con el propósito de dar cumplimiento a los objetivos planteados. Por lo tanto, se incluye en este proyecto haciendo especial énfasis en el algoritmo de DMD (Dynamic Mode Decomposition), mostrando a detalle las teorías y operaciones matemáticas necesarias para su funcionamiento en la aplicación al sistema dinámico elegido.

3.1 GENERALIDADES

Durante el desarrollo de este proyecto y para propósito del entendimiento del algoritmo DMD, fue necesario abordar a profundidad temas pertenecientes al álgebra lineal, los cuales son la base y fundamentan la descomposición en modos dinámicos.

Temas como los relacionados a tipos de matrices y sus operaciones, espacios vectoriales, valores y vectores propios; son claves para comprender la manera interna en la que el algoritmo consigue obtener la dinámica con datos reducidos.

En aras de simplificar el presente texto, tales conceptos principales se mencionan en el glosario de este documento, teniendo en cuenta que son términos de fácil consulta y en caso de requerir observar su procedimiento algebraico, se puede recurrir a la bibliografía pertinente²³⁴

Además de los términos mencionados anteriormente, existen otros que son clave para el desarrollo del proyecto, los cuales se verán a continuación.

² STRANG, Gilbert. Linear algebra and its applications. 3 ed. United States of America: Thomson Learning, 1988. ISBN 0-15-551005-3.

³ STANLEY, Grossman S y FLORES, José. Álgebra lineal. 7 ed. México, D.F.: McGraw-Hill, 2012. ISBN 978-607-15-0760-0.

⁴ KOLMAN, Bernard y HILL, David R.. Álgebra lineal. 8 ed. México, D.F.: Pearson Educación, 2006. ISBN 970-26-0696-9.

3.2 DIAPASÓN ACÚSTICO

Un diapasón es un resonador acústico con forma de horquilla con dos puntas en forma de “U” unidas entre sí en forma circular a la base o mango.

Generalmente se encuentran contruidos de un material elástico como el acero, el cual al ser golpeado con un objeto o contra una superficie, este vibra a una frecuencia constante o natural que produce un tono audible y puro una vez desaparecen los armónicos.

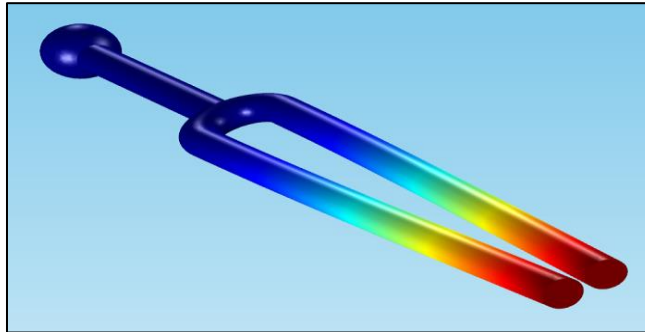
Figura 2. Diapasón de sección circular.



Fuente. AMAZON. Diapasón, Estándar A 440 Hz Afinador de Guitarra Violín Herramienta de Tono de Instrumento. [Sitio web]. Fecha de publicación: 28 de marzo del 2018. [Consulta: 15 de febrero del 2021]. Disponible en: <https://n9.cl/fv92s>

La frecuencia de oscilación a la cual vibra un diapasón, es su fundamental (primera frecuencia) y su valor dependerá del largo de las puntas, área transversal de las mismas, así como de las propiedades de su material de construcción, generalmente la densidad y rigidez. El valor de la frecuencia fundamental deseado se consigue durante su construcción al ir puliendo las puntas hasta lograr el espectro requerido.

Figura 3. Movimiento simétrico de las puntas sobre su mismo plano al oscilar en la frecuencia fundamental.



Fuente. COMSOL. The Tuning Fork Application. [Sitio web]. Fecha de publicación: 11 de noviembre del 2014. [Consulta: 16 de febrero del 2021]. Disponible en: <https://n9.cl/dcj05>

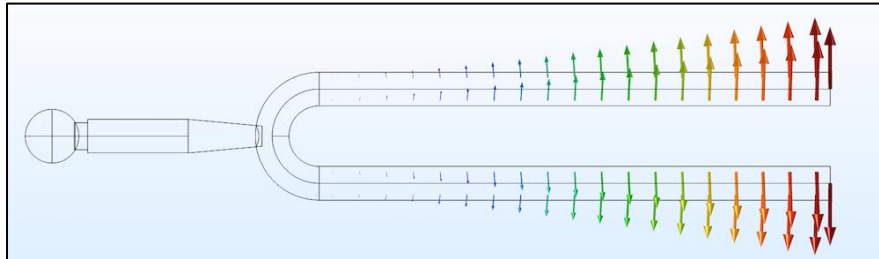
El diapasón se emplea principalmente en la afinación de instrumentos musicales y en pruebas de audiometría. De forma general, gran parte de los diapasones al ser excitados oscilan a una frecuencia propia de 440 [Hz].

Al ser excitado, el diapasón vibrará en un patrón complejo de movimiento, el cual puede ser descrito como superposición de modos resonantes o armónicos, donde cada modo está asociado a una frecuencia particular. Los armónicos más altos se amortiguarán con mayor rapidez que el fundamental, debido a que estos presentan movimientos de torsión y rotación que se transfieren al mango, energía que se disipa rápidamente al sostener el diapasón con una mano.

Para el modo fundamental, existe simetría en el movimiento de las puntas, las cuales oscilan sobre un mismo plano y se acercan o alejan entre si comportándose como vigas en voladizo. En este caso, la amortiguación es mucho más sutil, ya que las fuerzas presentes en las puntas se equilibran justo en el centro de unión entre ellas, evitando transferir mucha energía al mango y a medida que las puntas se desplazan, este se mueve de forma longitudinal en aras de mantener el centro de masa en posición fija. Este movimiento axial del mango permite amplificar el tono

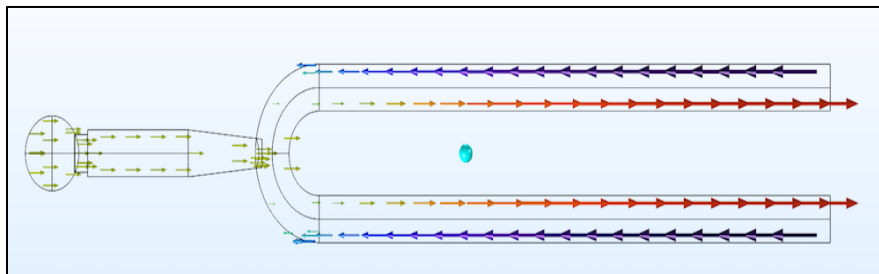
fundamental del diapasón si este se apoya sobre una superficie, como por ejemplo una mesa.

Figura 4. Desplazamiento transversal de las puntas para el modo fundamental.



Fuente. COMSOL. Finding Answers to the Tuning Fork Mystery with Simulation. [Sitio web]. Fecha de publicación: 13 de abril del 2018. [Consulta: 18 de febrero del 2021]. Disponible en: <https://n9.cl/0rgu3>

Figura 5. Representación de desplazamiento axial en el diapasón.



Fuente. COMSOL. Finding Answers to the Tuning Fork Mystery with Simulation. [Sitio web]. Fecha de publicación: 13 de abril del 2018. [Consulta: 18 de febrero del 2021]. Disponible en: <https://n9.cl/0rgu3>

Son pocos los estudios existentes sobre la dinámica del diapasón en los que se centran en la matemática de fondo que modela su comportamiento. Una vez excitado, este oscila con patrones de movimiento complejos, los cuales varían a medida que se recorre su superficie. Es posible aproximar las puntas del diapasón como vigas de flexión en voladizo, donde sus desplazamientos se aproximan por ecuaciones diferenciales parciales (lineales y no lineales).

Dentro de esos patrones complejos de movimiento, el diapasón presenta comportamientos lineales y no lineales, los cuales se ilustrarán en un capítulo posterior.

Por último, no sólo en el modo fundamental se presenta esta simetría de movimiento entre las puntas oscilantes en un mismo plano, pues existe otro modo de frecuencia mucho más elevada (aproximadamente 6.25 veces la fundamental)⁵ conocido como modo clang, en la que se presenta tal fenómeno.

Este modo es el responsable de un sonido agudo y metálico en los inicios del diapasón excitado, ya que el oído humano es más sensible para percibir frecuencias de entre 1 [KHz] a 5 [KHz]⁶. Este modo se desvanece rápidamente.

3.3 SINGULAR VALUE DECOMPOSITION (SVD)

La descomposición en valores singulares (SVD) es una técnica ampliamente utilizada para descomponer una matriz de alta dimensión (factorización matricial) en varias matrices componentes, exponiendo muchas de las propiedades útiles y más dominantes de la matriz original que representa al sistema dinámico. La SVD permite obtener de forma sistemática aproximaciones que conllevan a una eliminación o reducción de características redundantes del sistema.

La SVD contiene a modo general el concepto de transformada rápida de Fourier (FFT), pero difieren en que la FFT es usada para estudios analíticos y numéricos clásicos idealizados, mientras que la descomposición en valores singulares es una técnica que se adapta a mediciones específicas de un análisis basado en datos. Si los datos en la matriz son multidimensionales, se deben organizar en un vector columna, el cual representa el comportamiento de una variable del sistema para un instante de tiempo definido, donde gracias a la aplicación de la técnica se obtiene

⁵ HEARING HEALTH & TECHNOLOGY MATTERS. The Tuning Fork. [Sitio web]. Fecha de publicación: 1 de abril del 2012. [Consulta: 25 de febrero del 2021]. Disponible en: <https://n9.cl/jcg3n>

⁶ HEFFNER, Henry y HEFFNER, Rickye. Hearing ranges of lab-oratory animals. *J. Am. Assoc. Lab. Anim. Sci.* 2007, pp. 21-23

una matriz reducida, lo que significa que en la gran matriz de datos existen valores dominantes del sistema, demostrando así la robustez y eficiencia de SVD en la determinación de patrones, además de ser numéricamente estable y con la especial característica de organizar decrecientemente según su energía los factores más importantes del sistema en una de las matrices obtenida de la descomposición. Según Brunton y Kutz⁷ para definir a profundidad la técnica SVD, generalmente el interés se centra en analizar un gran conjunto de datos dado por $\mathbf{X} \in \mathbb{C}^{m \times n}$.

$$\mathbf{X} = \begin{bmatrix} \vdots & \vdots & \dots & \vdots \\ \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_n \\ \vdots & \vdots & \dots & \vdots \end{bmatrix} \quad (\text{Ec 1})$$

Las columnas $\mathbf{x}_k \in \mathbb{C}^m$, se pueden obtener de simulaciones o experimentos y generalmente representan el estado de un sistema físico que evoluciona en el tiempo, como las componentes de velocidad de un fluido en puntos discretos.

El subíndice k es una etiqueta que indica el k -ésimo conjunto distinto de medidas y \mathbf{X} consiste en una serie temporal de datos al igual que $\mathbf{x}_k = \mathbf{x}(k\Delta t)$. Con frecuencia la dimensión n es muy alta, en el orden de millones o billones grados de libertad.

Las columnas \mathbf{x}_k también son llamadas snapshots (instantáneas) y m es el número de instantáneas. Para muchos sistemas $m \gg n$, donde resulta una matriz alta y delgada, a diferencia de $m \ll n$ que produce una matriz corta y ancha.

La técnica SVD es una factorización matricial única que existe para cada matriz de valores complejos $\mathbf{X} \in \mathbb{C}^{m \times n}$:

$$\mathbf{X} = \mathbf{U} * \Sigma * \mathbf{V}^* \quad (\text{Ec 2})$$

⁷ BRUNTON, Steven L. y KUTZ, J. Nathan. Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control. Cambridge: Cambridge University Press, 2019. p. 15.

Donde,

- $\mathbf{U} \in \mathbb{C}^{m \times m}$: es una matriz unitaria. Para la técnica SVD \mathbf{U} representa una matriz de vectores singulares a la izquierda.
- $\mathbf{V} \in \mathbb{C}^{n \times n}$: al igual que \mathbf{U} es una matriz unitaria, con la característica que contiene los vectores singulares a la derecha.
- $\Sigma \in \mathbb{R}^{m \times n}$: es la matriz que contiene los valores singulares, devueltos como una matriz diagonal los cuales están organizados en forma decreciente según su contenido energético.
- \mathbf{V}^* : hace referencia a la matriz traspuesta conjugada o matriz adjunta.

Para facilitar su análisis el SVD se puede expresar de la siguiente manera:

$$\mathbf{X} = \mathbf{U} * \Sigma * \mathbf{V}^* = [\hat{\mathbf{U}} \quad \hat{\mathbf{U}}^\perp] \begin{bmatrix} \hat{\Sigma} \\ \mathbf{0} \end{bmatrix} \mathbf{V}^* = \hat{\mathbf{U}} \hat{\Sigma} \mathbf{V}^* \quad (\text{Ec 3})$$

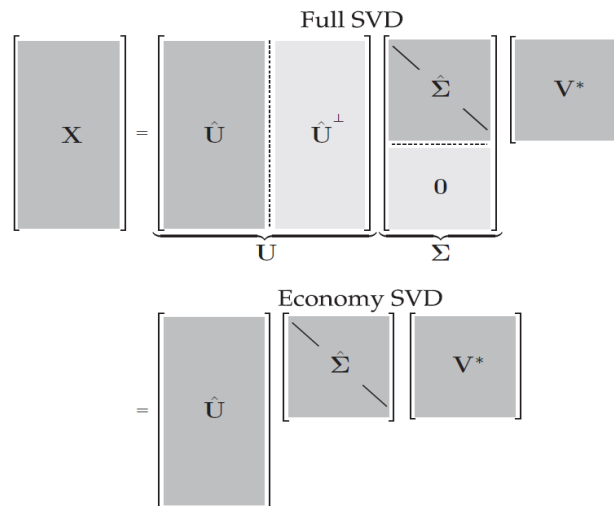
Donde:

- Las columnas de $\hat{\mathbf{U}}^\perp$ abarca un espacio vectorial complementario y ortogonal al generado por $\hat{\mathbf{U}}$.
- Las columnas de $\hat{\mathbf{U}}$ se conocen como vectores singulares izquierdos de \mathbf{X} y las columnas de \mathbf{V} corresponden a los vectores singulares derechos.
- La diagonal $\hat{\Sigma} \in \mathbb{C}^{n \times n}$ está conformada por los valores singulares quienes se encuentran organizados de mayor a menor (más dominante a menos dominante). La cantidad de valores singulares distintos de cero determinan el rango de la matriz \mathbf{X} .

En MATLAB existe una función SVD económico que como se puede ver en la figura 8, elimina las filas o columnas extra de ceros de la matriz diagonal de valores singulares $\hat{\Sigma}$, junto con las columnas en cualquiera de las dos matrices \mathbf{U} ó \mathbf{V} , con el propósito de cumplir la regla de multiplicación de matrices que indica que el número de columnas de la primera matriz debe coincidir con el número de filas de

la segunda matriz. La eliminación de estos ceros y columnas puede mejorar de manera considerable el tiempo de ejecución y reducir los requisitos de almacenamiento sin comprometer la precisión de la descomposición.

Figura 6. Esquema de las matrices SVD sin y con solución económica.



Fuente. BRUNTON, Steven L. y KUTZ, J. Nathan. Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control. Cambridge: Cambridge University Press, 2019. p. 6.

3.4 DYNAMIC MODE DECOMPOSITION (DMD)

En la dinámica de fluidos Peter Schmid⁸ diseñó un algoritmo con la capacidad de representar el comportamiento de un sistema dinámico de alta dimensión, a partir de la identificación de patrones característicos, el cual llamó DMD.

DMD tiene sus principios en la descomposición ortogonal aplicada (POD), que utiliza la descomposición en valores singulares (SVD), mientras que el SVD y POD dan como resultado valores singulares o modos jerarquizados en correlación espacial y contenido energético, ignorando la información temporal, el DMD realiza una

⁸ SCHMID, PETER J. Dynamic mode decomposition of numerical and experimental data. En: Journal of Fluid Mechanics. Aug 10,. vol. 656, no. August, 2010. p. 5-28

descomposición en modos dinámicos, dónde cada modo contiene una estructura espacialmente correlacionada que tienen el mismo comportamiento lineal en el tiempo.

3.4.1 Usos e interpretaciones del DMD. Se puede pensar que el DMD permite tres principales usos:

- a) **Diagnóstico.** En sus inicios el DMD fue utilizado como herramienta de diagnóstico para caracterizar el comportamiento de fluidos⁹. En muchos sistemas dinámicos es posible determinar su comportamiento completamente a partir de la extracción de características espaciotemporales, donde DMD se ha catalogado como una de las principales herramientas de reducción de dimensionalidad¹⁰.
- b) **Estimación del estado actual y predicción del estado futuro.** Este es un uso más desafiante del algoritmo DMD, donde el objetivo es extraer los patrones dominantes de un sistema dinámico y a partir de ellos estimar el comportamiento subsiguiente, es decir, anticipar el estado del sistema en un régimen en el que no se realizaron mediciones¹¹. Una de las claves para que el resultado sea exitoso es el muestreo inteligente de los datos, teniendo en cuenta los criterios exigidos por el algoritmo.
- c) **Control.** Teniendo en cuenta, que estamos usando un modelo dinámico lineal para predecir el estado futuro de un sistema dinámico¹², es razonable esperar que haya un tiempo limitado para que dicha acción sea exitosa. Por esta razón,

⁹ SCHMID, PETER J. Dynamic mode decomposition of numerical and experimental data. En: Journal of Fluid Mechanics. Aug 10,. vol. 656, no. August, 2010. p 3.

¹⁰ KUTZ, J. N., et al. Dynamic Mode Decomposition. Society for Industrial and Applied Mathematics, 2016. p 2.

¹¹ Ibid., p 2.

¹² Ibid., p 3.

si se tiene una predicción precisa y lo suficientemente larga, es posible la toma de decisiones que influyan en el estado futuro del sistema.

3.4.2 Ventajas del DMD.

- ✓ Una de las principales ventajas de la descomposición en modo dinámico es su formulación simple, a partir de conocimientos técnicos de álgebra lineal.
- ✓ Básicamente la descomposición en modos dinámicos puede asimilarse como una regresión lineal, la cual permite determinar el mejor comportamiento que se ajuste a los valores del sistema inicial.
- ✓ Como dice Brunton y Kutz¹³ DMD no requiere conocimientos de primeros principios o ecuaciones de gobierno y por ende su aplicación ha sido amplia, puede ser implementado casi en cualquier sistema que evolucione en el tiempo conociendo únicamente información espaciotemporal del sistema dinámico. Algunos ejemplos pueden ser: procesamiento de video, epidemiología, neurociencias, finanzas, fluidos, fenómenos vibratorios, entre otros.

3.4.3 Formulación de la técnica DMD. Uno de los principales problemas en la vida real para describir el comportamiento de un sistema dinámico es el conocimiento en detalle de las ecuaciones que lo gobiernan, donde las mismas pueden estar formuladas según la Ec 4. Por esta razón, se hace necesario implementar técnicas como DMD, capaces de encontrar un modelo que describa el comportamiento de sistemas dinámicos complejos de los cuales, generalmente, lo único que se puede obtener es datos de sus variables principales.

¹³ BRUNTON, Steven L. y KUTZ, J. Nathan. Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control. Cambridge: Cambridge University Press, 2019. p. 236.

Los sistemas dinámicos proporcionan uno de los campos de las matemáticas más completos y mejor conectados uniendo diversos temas. Como introducción matemática, se tiene un tipo de proceso dinámico dado por:

$$\frac{d}{dt}\mathbf{x}(t) = f(\mathbf{x}(t), t; \boldsymbol{\beta}), \quad (\text{Ec } 4)$$

Donde \mathbf{x} es el estado del sistema y pertenece a \mathbb{R}^n , f representa la dinámica de acuerdo al tipo de sistema y puede estar relacionada con el estado de \mathbf{x} (*), t es el tiempo y $\boldsymbol{\beta}$ es un conjunto de parámetros; representa a un sistema no autónomo o forzado, el cual se considera como variante en el tiempo

El sistema dinámico a estudiar, se puede expresar como una regresión lineal expresada de la siguiente manera¹⁴:

$$\frac{d\mathbf{x}}{dt} = \mathbf{A}\mathbf{x} \quad (\text{Ec } 5)$$

Donde \mathbf{A} es un factor que reconstruye el sistema localmente lineal aproximado.

El sistema dinámico (Ec 5) es posible describirlo como un sistema dinámico de tiempo discreto muestreado cada Δt ¹⁵.

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k \quad (\text{Ec } 6)$$

Donde al solucionar la Ec 6, tenemos que:

¹⁴ KUTZ, J. N., et al. Dynamic Mode Decomposition. Society for Industrial and Applied Mathematics, 2016. p 3.

¹⁵ BRUNTON, Steven L. y KUTZ, J. Nathan. Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control. Cambridge: Cambridge University Press, 2019. p. 237.

(*) \mathbf{f} puede ser un espacio vectorial, el cual al ser multivariable proporciona múltiples ecuaciones diferenciales, es decir, \mathbf{f} proporciona el grado de dificultad a la solución de la ecuación 1.

$$\mathbf{X}' \approx \mathbf{A} * \mathbf{X}$$

$$\mathbf{x}(t_0 + t) = e^{At} \mathbf{x}(t_0)$$

$$\mathbf{x}(t_0 + t) = \mathbf{W} * e^{\lambda t} * \mathbf{W}^{-1} * \mathbf{x}(t_0) \quad (\text{Ec 7})$$

La solución a este sistema puede ser expresada simplemente en términos de los vectores propios (\mathbf{W}_k) y valores propios (λ_k) de la matriz de tiempo discreto \mathbf{A} :

$$\mathbf{A}\mathbf{W} = \mathbf{W}\lambda \quad (\text{Ec 8})$$

Finalmente, el objetivo es reconstruir el sistema dinámico encontrando su modelo matemático:

$$\mathbf{x}_k \approx \sum_{j=1}^n \Phi_j \mathbf{b}_j e^{\lambda t} \quad (\text{Ec 9})$$

3.4.4 Características de la organización de la información. Se debe tener claro que para implementar la técnica DMD, lo único necesario es obtener información a lo largo del tiempo de un sistema dinámico. Donde, según Schmid¹⁶ la información debe ser recolectada u organizada en dos matrices (\mathbf{X} y \mathbf{X}') de tamaño $m \times n$.

$$\mathbf{X} = \begin{bmatrix} \vdots & \vdots & \dots & \vdots \\ \mathbf{x}_{(t1)} & \mathbf{x}_{(t2)} & \dots & \mathbf{x}_{(tn-1)} \\ \vdots & \vdots & \dots & \vdots \end{bmatrix} \quad (\text{Ec 10})$$

¹⁶ SCHMID, PETER J. Dynamic mode decomposition of numerical and experimental data. En: Journal of Fluid Mechanics. Aug 10,. vol. 656, no. August, 2010. p. 5-28

$$\mathbf{X}' = \begin{bmatrix} \vdots & \vdots & \dots & \vdots \\ \mathbf{x}_{(t2)} & \mathbf{x}_{(t3)} & \dots & \mathbf{x}_{(tn)} \\ \vdots & \vdots & \dots & \vdots \end{bmatrix} \quad (\text{Ec 11})$$

Donde cada vector columna $\mathbf{x}_k \in \mathbb{C}^m$, contiene m número de datos de la variable de interés del sistema dinámico en un instante de tiempo determinado, con la característica de que cada vector columna $\mathbf{x}'_k \in \mathbb{C}^m$ de la matriz \mathbf{X}' esta desplazada un intervalo de tiempo (Δt) constante en el futuro.

3.4.5 Desarrollo de la técnica DMD. El algoritmo busca la descomposición espectral principal (es decir, valores y vectores propios) donde \mathbf{A} es el operador lineal que relaciona las dos matrices \mathbf{X} y \mathbf{X}' .

$$\mathbf{X}' \approx \mathbf{A}\mathbf{X} \quad (\text{Ec 12})$$

Según Kutz et al¹⁷ matemáticamente, el mejor ajuste de \mathbf{A} está dado por:

$$\mathbf{A} = \text{arg } \|\mathbf{X}' \approx \mathbf{A}\mathbf{X}\|_F = \mathbf{X}'\mathbf{X}^\dagger \quad (\text{Ec 13})$$

Donde, $\|\cdot\|_F$ es la norma de Frobenius¹⁸ y \mathbf{X}^\dagger denota el pseudo-inverso Moore-Penrose¹⁹ que tiene como objetivo encontrar el mejor ajuste (mínimos cuadrados) y de esta manera encontrar el modelo matemático.

3.4.6 Algoritmo DMD. Una vez obtenidas las matrices \mathbf{X} y \mathbf{X}' de dimensión $m \times n$ organizadas de tal manera que cada columna almacene información de una instantánea en el tiempo (Ec 10) y (Ec 11), se procede a realizar una primera reducción por medio del método SVD y una segunda reducción de la matriz Σ

¹⁷ KUTZ, J. N., *et al.* Dynamic Mode Decomposition. Society for Industrial and Applied Mathematics, 2016. p. 6.

¹⁸ AGAPITO, Rubén. Cálculo exacto de la matriz exponencial. Pro Mathematica. 2014. p 58-60.

¹⁹ BARATA, João, *et al.* The Moore–Penrose Pseudoinverse: A Tutorial Review of the Theory. Brazilian journal of physics. 2012, vol. 42, nro. 1. pp. 146-165

truncada por el rango (r), con el fin de facilitar el cálculo entre matrices y así mismo la identificación de patrones dinámicos del sistema.

- Inicialmente se calcula la descomposición en valores singulares (SVD ver capítulo 3.3) de la matriz \mathbf{X} , dando como resultado las matrices \mathbf{U} , $\mathbf{\Sigma}$ y \mathbf{V} (Ec 3).
- Se realiza la pseudoinversa de \mathbf{X} partiendo de la descomposición en valores singulares (SVD), para obtener una matriz \mathbf{A} , sin embargo, es más eficiente computacionalmente trabajar con una matriz \mathbf{A} reducida ($\tilde{\mathbf{A}}$).

$$\mathbf{A} = \mathbf{X}'\mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^* \quad (\text{Ec 14})$$

- Se reduce a una matriz ($\tilde{\mathbf{A}}$) de tamaño $r \times r$, donde r es el rango, calculado a partir de la matriz diagonal $\mathbf{\Sigma}$, la cual contiene valores de alta energía que almacenan la mayor parte de la información del sistema dinámico en los primeros valores de la matriz diagonal, y así mismo, es posible ignorar valores muy cercanos a cero que son considerados con baja energía o poco relevantes para reconstruir el comportamiento del sistema dinámico.

$$\tilde{\mathbf{A}} = \mathbf{U}_r^* \mathbf{A} \mathbf{U}_r = \mathbf{U}_r^* \mathbf{X}' \mathbf{V}_r \mathbf{\Sigma}_r^{-1} \quad (\text{Ec 15})$$

Donde,

- $\mathbf{U}_r \in \mathbb{C}^{m \times r}$
- $\mathbf{\Sigma}_r \in \mathbb{C}^{r \times r}$
- $\mathbf{V}_r \in \mathbb{C}^{n \times r}$

- Partiendo de la matriz $\tilde{\mathbf{A}}$, se calculan los valores y vectores propios.

$$\tilde{\mathbf{A}} \mathbf{W}_r = \mathbf{W}_r \boldsymbol{\lambda} \quad (\text{Ec 16})$$

Donde, \mathbf{W}_r es una matriz con los vectores propios y λ es una matriz diagonal que contiene los valores propios.

- Se reconstruye el sistema dinámico a partir de una matriz Φ , calculada según la ecuación que se presenta, a continuación:

$$\Phi = \mathbf{X}'\mathbf{V}\Sigma^{-1}\mathbf{W}_r \quad (\text{Ec 17})$$

La Ec 17 contiene en sus columnas los vectores propios de la matriz \mathbf{A} , que corresponden a los valores propios λ como se observa:

$$\mathbf{A}\Phi = (\mathbf{X}'\mathbf{V}_r\Sigma^{-1}_r\mathbf{U}^*_r)(\mathbf{X}'\mathbf{V}_r\Sigma^{-1}_r\mathbf{W}_r)$$

$$\tilde{\mathbf{A}} = \mathbf{U}^*_r\mathbf{X}'\mathbf{V}_r\Sigma^{-1}_r$$

$$\mathbf{A}\Phi = \mathbf{X}'\mathbf{V}_r\Sigma^{-1}_r\tilde{\mathbf{A}}\mathbf{W}_r$$

$$\mathbf{A}\Phi = \mathbf{X}'\mathbf{V}_r\Sigma^{-1}_r\mathbf{W}_r\lambda$$

$$\mathbf{A}\Phi = \Phi\lambda \quad (\text{Ec 18})$$

- Para el cálculo de los modos de vibración, se utiliza la siguiente ecuación:

$$\Omega = \frac{\ln(\lambda_k)}{\Delta t} \quad (\text{Ec 19})$$

- Se considera la primera instantánea de datos \mathbf{X}_1 , en el instante de tiempo $t = 0$ para calcular el vector \mathbf{b} , a partir de la pseudoinversa de Φ , denotada como Φ^\dagger , a continuación, se muestra la ecuación que representa el cálculo mencionado.

$$\mathbf{b} = \Phi^+ \mathbf{X}_1 \quad (\text{Ec 20})$$

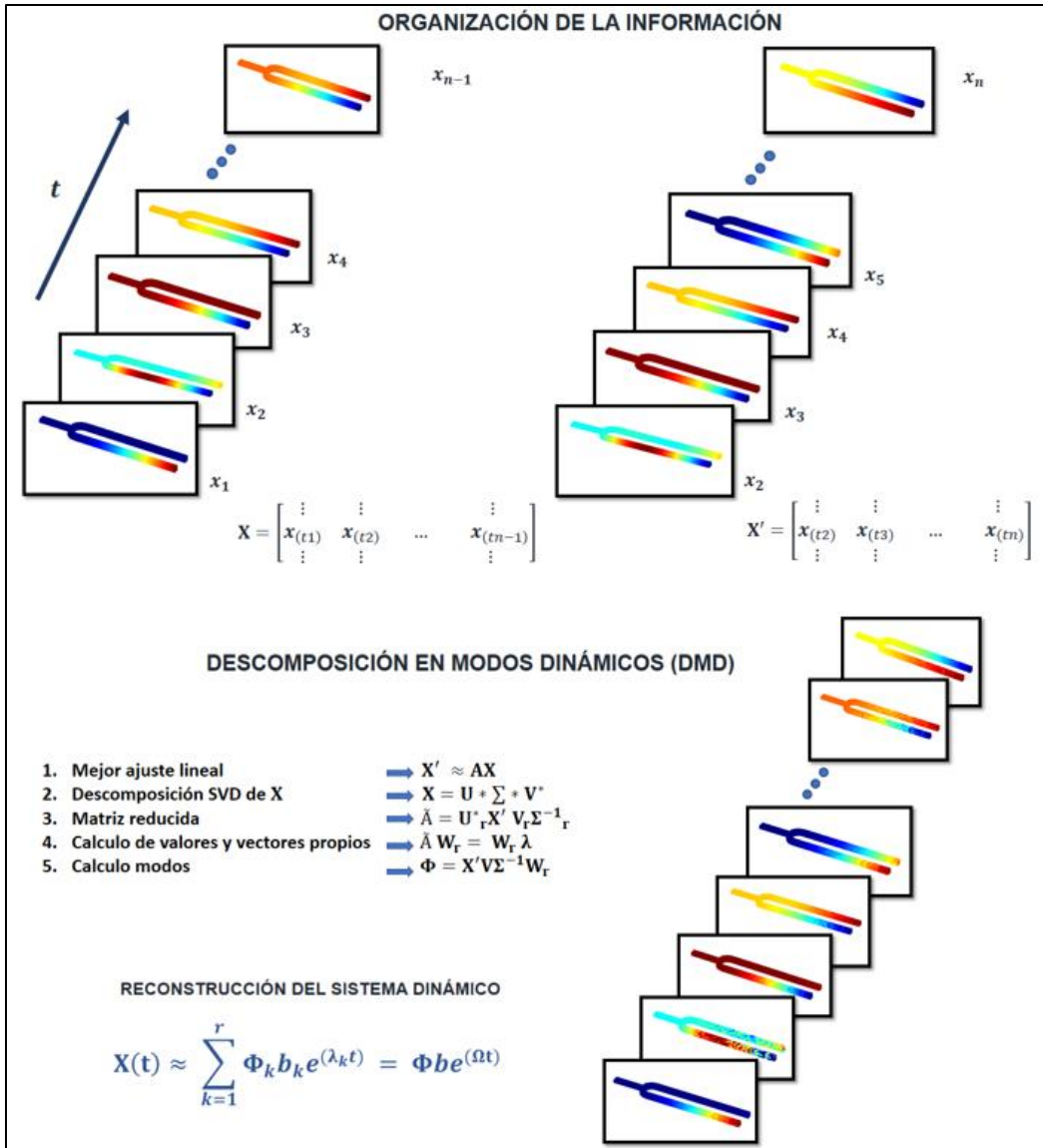
- Se construye la solución futura aproximada del sistema dinámico, dada por la ecuación:

$$\mathbf{X}(t) \approx \sum_{k=1}^r \Phi_k \mathbf{b}_k e^{(\lambda_k t)} = \Phi \mathbf{b} e^{(\Omega t)} \quad (\text{Ec 21})$$

Donde,

- \mathbf{b}_k : es la amplitud inicial de cada modo, contenidos en un vector \mathbf{b}
- Φ : es una matriz cuyas columnas son los vectores propios \mathbf{W}_k
- Ω : es una matriz diagonal cuyas entradas son los valores propios λ_k

Figura 7. Breve descripción paso a paso de la técnica DMD.



3.4.7 Matriz de Hankel. Se pretende utilizar el esquema de mapeo por coordenadas de retardo, el cual tiene como objetivo obtener una diferencia en la ubicación ($X_{(ij)}$) entre un atractor incrustado en retardo y el atractor original, esta organización específica, es posible reconstruirla a partir de la matriz de Hankel²⁰, básicamente es una matriz cuadrada con una característica única, la cual organiza

²⁰ WATANABE, Junzo. Hankel matrices and Hankel ideals. ResearchGate. 1995, vol. 32, p. 11-21

las anti-diagonales de derecha a izquierda paralelas numéricamente, representada por la siguiente estructura:

$$\mathbf{H} = \begin{bmatrix} x(t_1) & x(t_2) & \cdots & x(t_{nc}) \\ x(t_2) & x(t_3) & \cdots & x(t_{nc+1}) \\ \vdots & \vdots & \ddots & \vdots \\ x(t_{no}) & x(t_{no+1}) & \cdots & x(t_n) \end{bmatrix} \quad (\text{Ec 22})$$

En el caso del diapasón, se pretende realizar el estudio para varios nodos con desplazamiento en una sola dirección, ya sea horizontal, vertical o axial dependiendo el comportamiento del diapasón en su base o en sus puntas. Esta información se debe organizar apilando múltiples copias de los valores desfasados en el tiempo, haciendo uso de la matriz de Hankel.

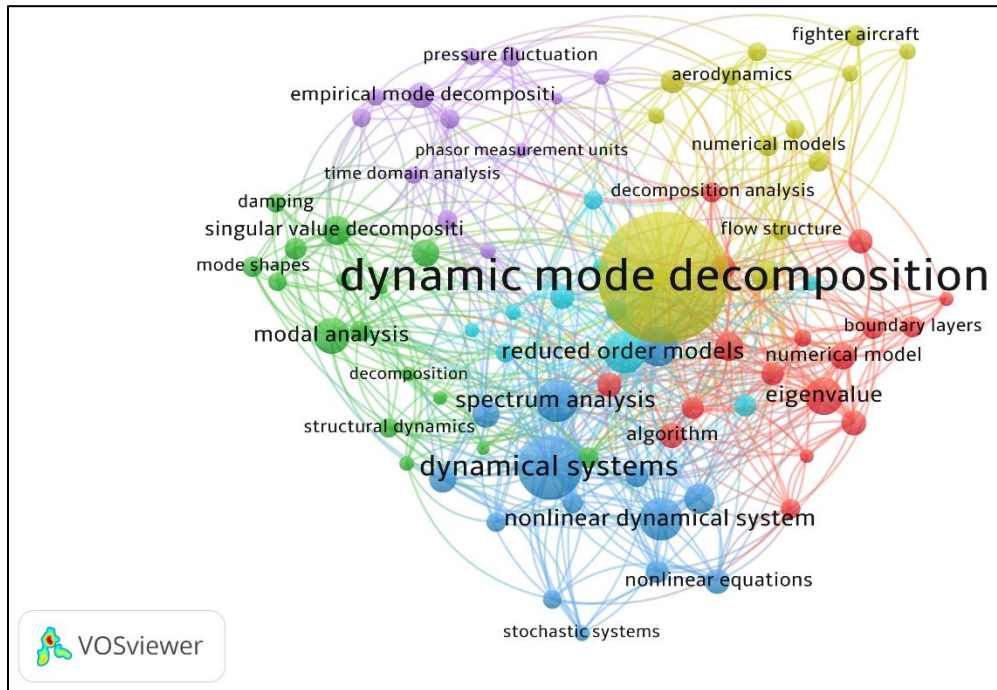
3.5 ESTADO DEL ARTE

Pese a ser una técnica poderosa, la descomposición en modos dinámicos (DMD) no generó un fuerte impacto inmediato en el mundo científico desde su creación en 2010 por el ingeniero Peter Schmid²¹, sin embargo, en años posteriores y a medida que aumenta cada vez más el auge de la ciencia de datos y aprendizaje automático en las múltiples disciplinas, la técnica ha cobrado gran relevancia gracias a su aplicabilidad en el análisis de sistemas dinámicos de alta complejidad como lo son: la turbulencia, la neurociencia, la combustión, el clima, la epidemiología, las finanzas, la óptica; entre otros.

La producción bibliográfica referente a esta técnica ha ido en aumento casi exponencial, lo cual se corroboró de acuerdo al análisis bibliométrico realizado por medio de la plataforma SCOPUS y VOSviewer.

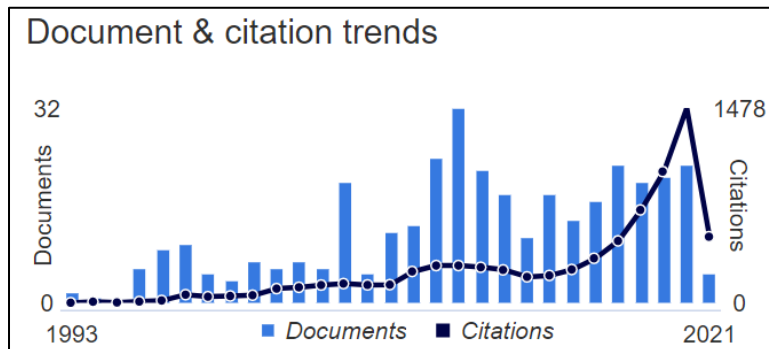
²¹ SCHMID, PETER J. Dynamic mode decomposition of numerical and experimental data. En: Journal of Fluid Mechanics. Aug 10,. vol. 656, no. August, 2010. p. 5-28

Figura 8. Análisis bibliométrico para palabras clave dynamic mode decomposition AND dynamic systems.



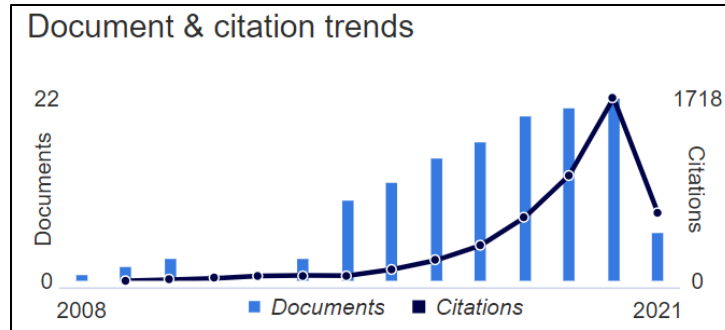
Dos de los autores quizás más importantes por las investigaciones realizadas en esta técnica son J. Nathan Kutz y Steven L. Brunton, en los que se observó que la influencia de sus publicaciones ha tomado fuerza para muchos investigadores.

Figura 9. Tendencia de citas que ha recibido J. Nathan Kutz.



Fuente. SCOPUS; Authors. [sitio web]. [Consulta: 09 de abril 2014]. Disponible en: <https://www.scopus.com/>

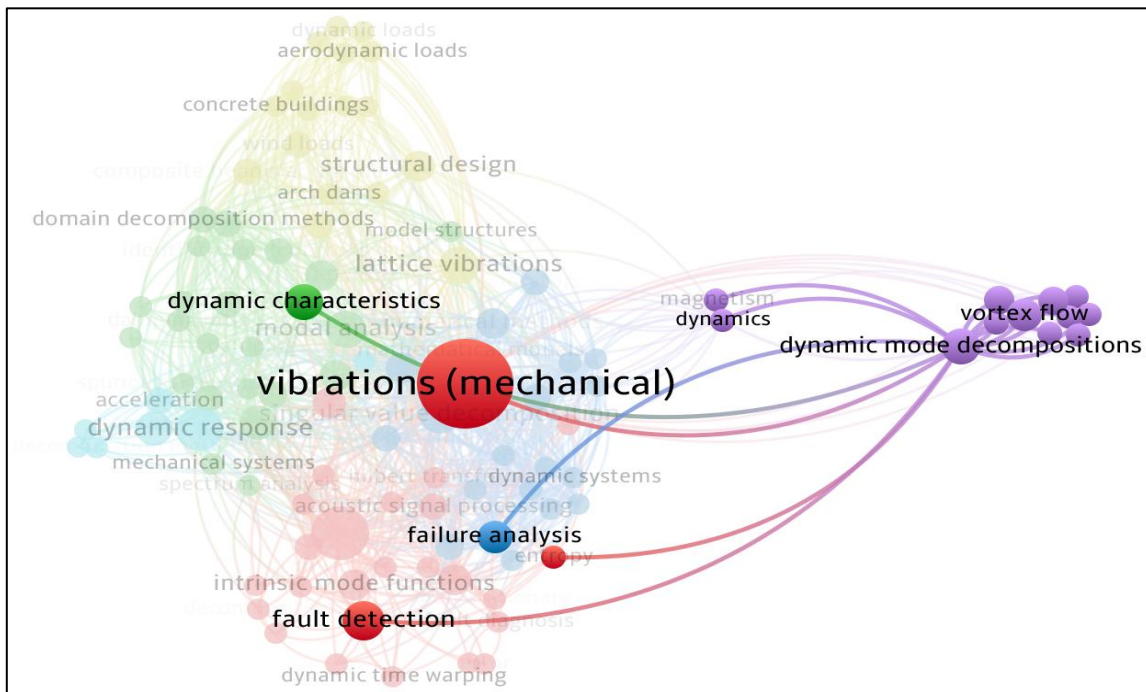
Figura 10. Tendencia de citas que ha recibido Steven L. Brunton.



Fuente. Fuente. SCOPUS; Authors. [sitio web]. [Consulta: 09 de abril 2014].
 Disponible en: <https://www.scopus.com/>

Aunque es claro el aumento del nivel de impacto de la técnica, se encontró que la aplicación en el campo de las vibraciones mecánicas ha sido poco investigado, motivo por el cual el presente proyecto centró su investigación en esta área en aras de corroborar la flexibilidad del algoritmo y como este se comporta en dicho campo.

Figura 11. Análisis bibliométrico para palabras clave dynamic mode decomposition AND mechanical vibrations.



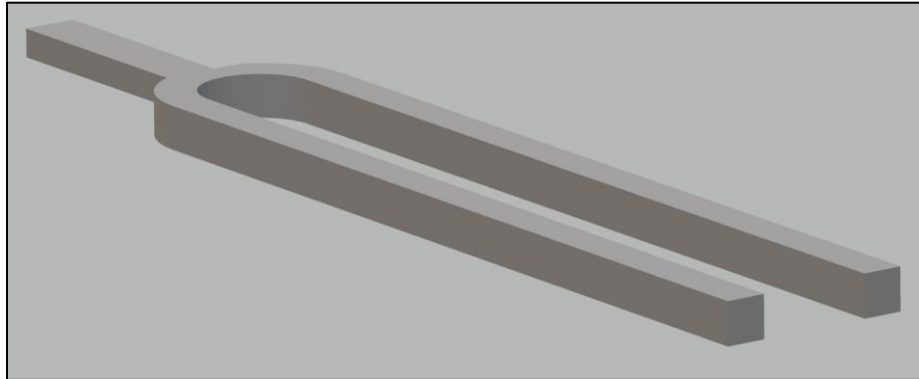
4. MODELADO ESTRUCTURAL Y SIMULACIÓN

4.1 MODELO 3D DEL DIAPASÓN.

Una vez analizado el sistema dinámico elegido, el algoritmo a implementar y su matemático de fondo, se realizó el modelado estructural del sistema y su posterior simulación; ambos procesos realizados en MATLAB.

En primer lugar, el modelo estructural que se carga al software corresponde a un archivo de imagen 3D de extensión “.*stl*”, el cual es muy usado en modelos para impresión en 3D. El plano con las dimensiones del diapasón puede ser consultado en la sección de anexos.

Figura 12. Modelo 3D del diapasón creado como un archivo de extensión “.*stl*”.



El comportamiento dinámico de este elemento se encuentra gobernado por ecuaciones diferenciales parciales, por lo que es necesario recurrir el método de elementos finitos para aproximar sus soluciones.

El diapasón que se usó, corresponde a uno con sección transversal rectangular, tanto en sus puntas como en su base.

4.2 ANÁLISIS MODAL

El análisis modal permite determinar las frecuencias naturales o modos de vibración de un modelo estructural, cuando este vibra libremente. Se lleva a cabo generalmente por medio del MEF.

4.2.1 Modelo y propiedades. Para iniciar con el análisis modal del diapasón, se creó una variable con la función interna de MATLAB *createpde*, la cual devuelve un modelo PDE (partial differential equation) de análisis estructural para resolver sistemas de elasticidad lineal con deformación pequeña. Además, se debe importar la geometría del modelo:

```
modelo_modal = createpde('structural','modal-solid');  
importGeometry(modelo_modal,'Diapason_acustico.stl');
```

Cabe resaltar que para hacer uso del modelado por medio de PDE, fue necesario contar con la instalación del módulo Partial Differential Equation Toolbox, el cual proporciona las funciones necesarias para resolver ese tipo de ecuaciones mediante el análisis del MEF.

Definiendo las propiedades mecánicas de la estructura de acero con características isotrópicas, como:

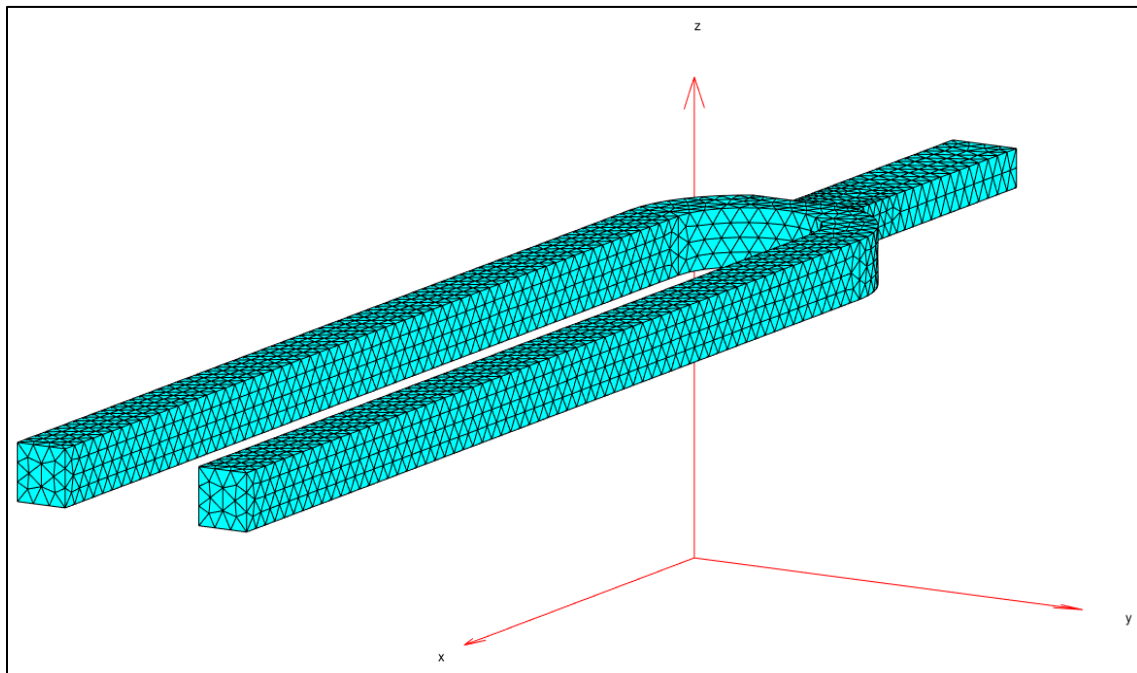
```
structuralProperties(modelo_modal,'YoungsModulus',E,'PoissonsRatio',nu,'MassDensity',rho);
```

Cuadro 1. Propiedades mecánicas generalmente usadas para el acero.

Módulo de Young	E	$210 * 10^9 \left[\frac{N}{m^2} \right]$
Densidad	ρ	$8,000 \left[\frac{kg}{m^3} \right]$
Coefficiente de Poisson	ν	0.3

4.2.2 Mallado para el análisis modal. Se establece un mallado para el modelo 3D, el cual consistió en una malla tetraédrica con longitud de borde máxima $H_{max} = 0.001$ y con tasa de crecimiento del 50 %, lo cual da lugar a 25,066 nodos.

Figura 13. Mallado para el análisis modal.



4.2.3 Función solución. Por último, para resolver el modelo, se estableció un rango de frecuencias, las cuales deben ser insertadas al código como frecuencias circulares ($\Omega = 2 * \pi * f$):

```
solucion_modal = solve(modelo_modal, 'FrequencyRange', [-1, 4000]*2*pi);
```

Resolviendo el modelo, se obtuvieron los modos de vibración, así como la frecuencia fundamental o propia del diapasón, cuyos valores se pueden apreciar en el capítulo de resultados.

4.3 ANÁLISIS TRANSITORIO

4.3.1 Modelo y propiedades. Una vez obtenidos los modos de vibración del diapasón para las condiciones dadas en el análisis modal, se realizó un análisis de manera similar al anterior, puesto que fue necesario importar nuevamente la geometría y crear el modelo PDE para la estructura, con la diferencia de que los argumentos estarán dados para el estado transitorio.

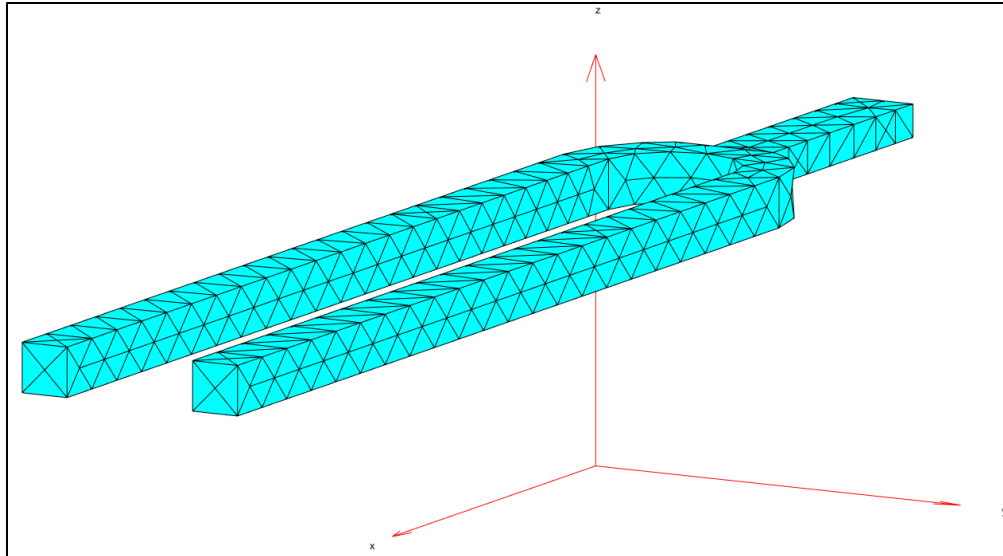
```
modelo_transitorio = createpde('structural','transient-solid');  
importGeometry(modelo_transitorio,'Diapason_acustico.stl');
```

Las propiedades mecánicas se definieron de igual manera que en el análisis modal (ver cuadro 1).

4.3.2 Mallado para el análisis transitorio. Al igual que en el análisis modal, se estableció un mallado para el modelo 3D, el cual consistió igualmente en una malla tetraédrica ahora con longitud de borde máxima $H_{max} = 0.003$ y con tasa de crecimiento del 50 %, lo cual da lugar a 2,125 nodos; una reducción significativa de elementos.

Para este análisis, el mallado es de mayor tamaño (reduce casi en doce veces la cantidad de nodos del análisis modal), debido a que el análisis transitorio requiere una mayor cantidad de recursos computacionales.

Figura 14. Mallado para el análisis transitorio.

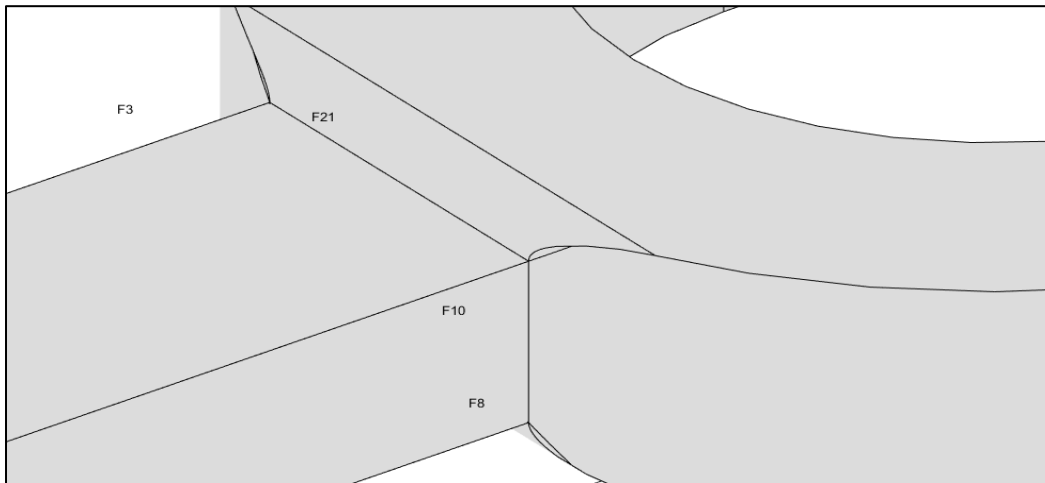


4.3.3 Condiciones iniciales, de frontera y de carga. Para que el comportamiento dinámico del diapasón fuese lo más real posible, fue necesario establecer una serie de condiciones y parámetros.

En las condiciones de contorno se establecieron restricciones de movimiento para dos caras (21 y 22) del diapasón. Lo anterior con el propósito de aproximar el comportamiento del diapasón cuando este se sostiene con una mano o se encuentra apoyado de forma vertical sobre una mesa.

```
structuralBC(modelo_transitorio, 'Face' [21,22], 'Constraint', 'fixed');
```

Figura 15. Ubicación de una de las dos caras fijadas para las condiciones de contorno.

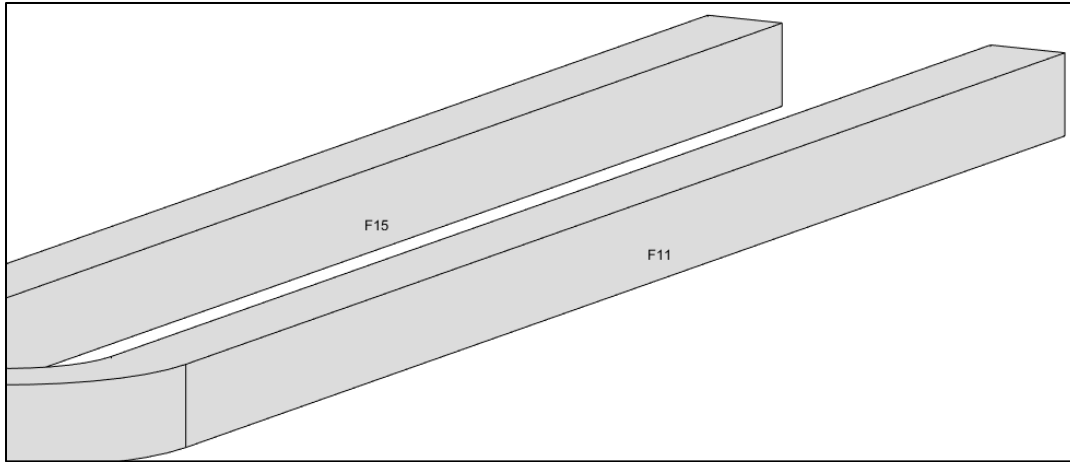


Como condiciones iniciales se establecieron parámetros de desplazamiento y velocidad iguales a 0 en todas sus componentes.

Para la carga de excitación del diapasón, esta se ejerce en forma de impulso transcurrido un tiempo t muy pequeño, donde la misma fue aplicada sobre la superficie de la cara lateral 11.

```
structuralBoundaryLoad(modelo_transitorio, 'Face', 11, 'Pressure', 5E6, 'EndTime', Periodo/300);
```

Figura 16. Cara sobre la cual se aplica la carga de excitación.



Para excitar al diapasón en su frecuencia fundamental, la carga aplicada debe garantizar la excitación del modo fundamental. Por lo anterior, esta se aplicó en una fracción muy pequeña del periodo de oscilación, la cual se calculó a partir de dicho modo de frecuencia.

$$Periodo = \frac{2 * \pi [rad]}{frec\ fundamental \left[\frac{rad}{s} \right]} [s] \quad (Ec\ 23)$$

4.3.4 Función solución. La solución del análisis transitorio permitió obtener datos de desplazamiento en tres componentes para cada nodo del mallado. La cantidad de datos arrojados es proporcional al número de nodos del sistema y al vector tiempo utilizado.

```
sol_transitorio = solve(modelo_transitorio, tiempo)
```

4.4 LINEALIDAD Y NO LINEALIDAD DEL DIAPASÓN.

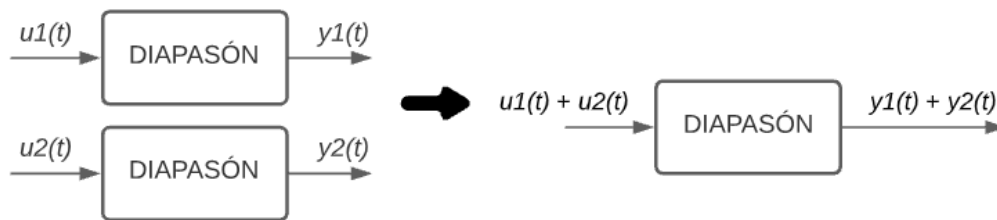
En el marco referencial se mencionó que la dinámica de un diapasón se encuentra gobernada por ecuaciones diferenciales parciales lineales y no lineales. Lo anterior

es posible corroborarlo si el principio de superposición y homogeneidad se cumplen (o no) en diferentes nodos al variar la entrada del sistema (carga de presión).

Teniendo en cuenta que los principios mencionados establecen que:

$$f(x + y) = f(x) + f(y) \quad (\text{Ec 24})$$

Figura 17. Principio de superposición.



$$f(\alpha * x) = \alpha * f(x) \quad (\text{Ec 25})$$

Figura 18. Principio de homogeneidad.



Si alguno de los dos principios no se cumple, el sistema es no lineal.

El diapasón presentó mayormente comportamiento lineal en su superficie al ser excitado en su modo fundamental.

Para verificar la linealidad fueron seleccionados tres nodos en ubicaciones alejadas entre sí, las cuales corresponden a:

- Nodo 24 ubicado en el inicio de la punta que es golpeada.

- Nodo 136 ubicado en cercanías al inicio de la curvatura de la punta antes mencionada.
- Nodo 11 ubicado en el mango del diapasón.

Para el análisis, se inició con la verificación del principio de homogeneidad, variando el valor de la entrada del sistema (magnitud del golpe), a través de un coeficiente α , cuyo valor se estableció en $\alpha = 1$ y $\alpha = 2$, es decir, la magnitud del golpe sufre un incremento del doble, con el propósito de comprobar que los desplazamientos (salida del sistema) sobre el eje y y x (en el caso del mango) sufrieron el mismo incremento α .

Figura 19. Comparación de desplazamientos para el nodo 24.

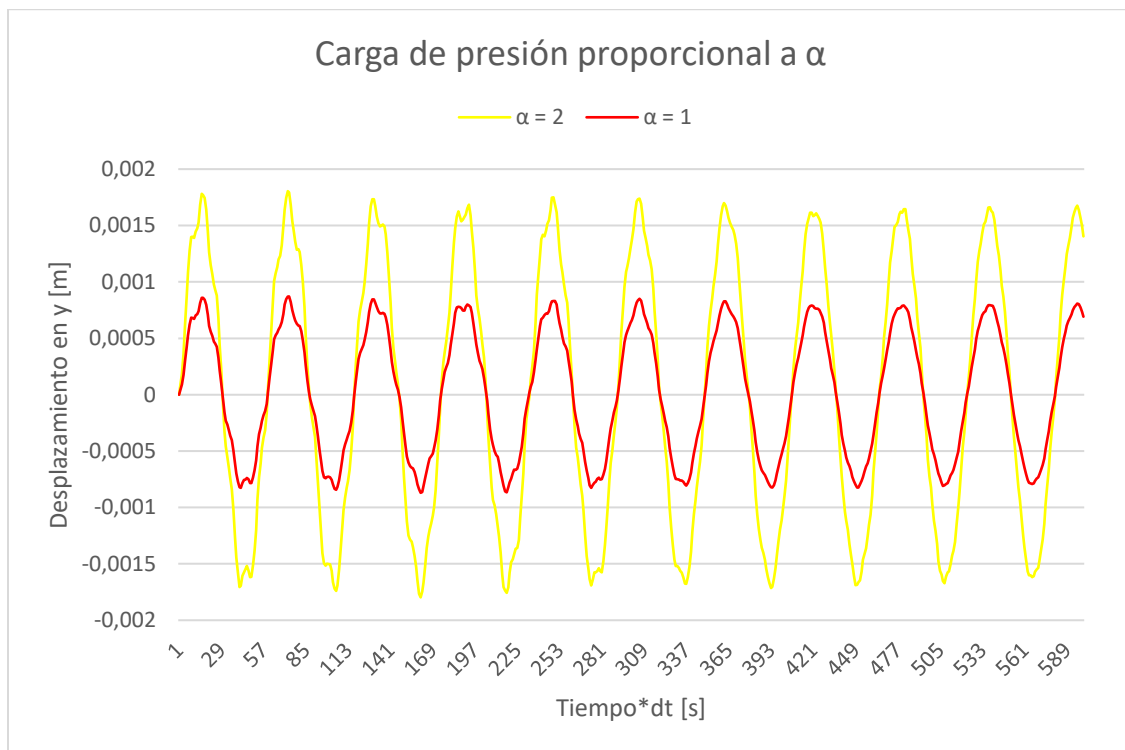
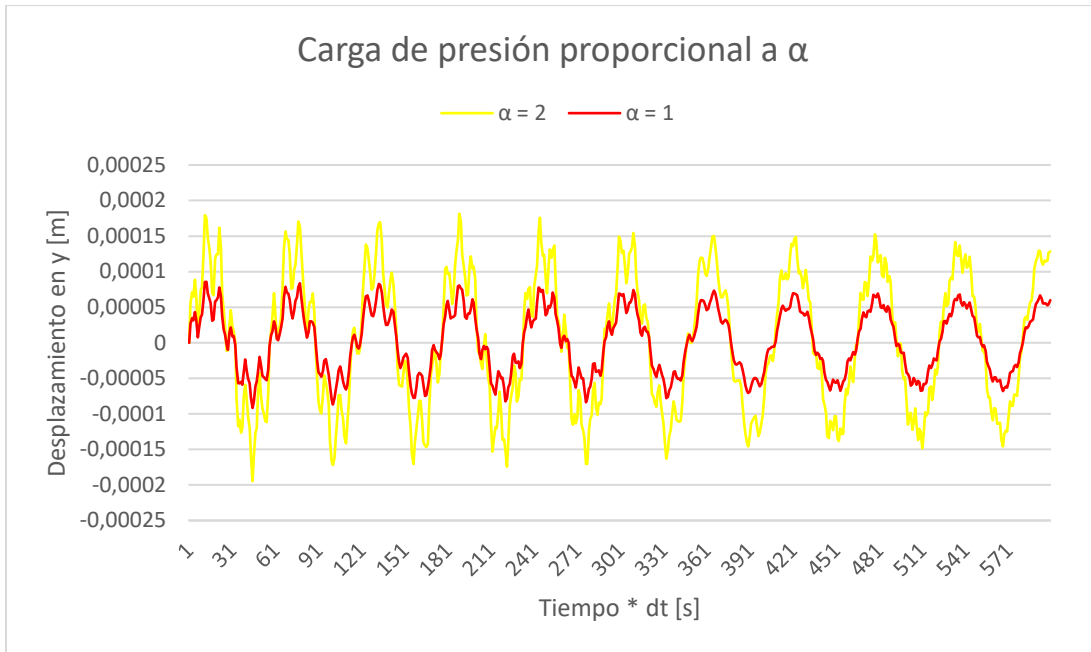


Figura 20. Comparación de desplazamientos, nodo 136.

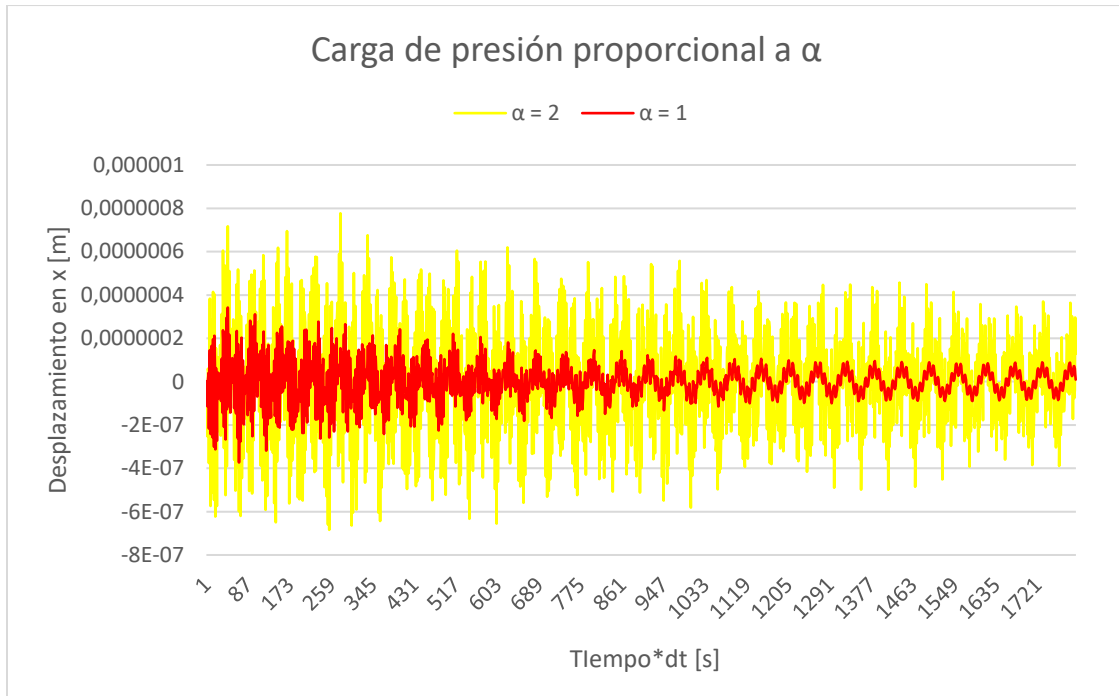


En la figura 19 y 20, es notorio el comportamiento lineal del diapasón, ya que al duplicar la magnitud del golpe ($\alpha = 2$) la magnitud del desplazamiento aumenta en aproximadamente la misma proporción, es decir, las dos puntas del diapasón presentan un comportamiento aproximadamente lineal al ser excitado en su frecuencia fundamental

Situación distinta se presenta en el mango del diapasón (nodo 11) ya que, al realizar la comparación de desplazamientos, se observa el fenómeno no lineal. Debido a la no linealidad las oscilaciones no son completamente periódicas y el desplazamiento periodo a periodo varía. La no linealidad ocurre por el movimiento que debe hacer el mango para compensar el centro de masa alterado por el movimiento de las puntas.

Por lo anterior se afirma que el diapasón contiene los dos tipos de comportamientos dinámicos (lineal y no lineal), los cuales difieren en el mango y cercanías a este y en las dos puntas del diapasón.

Figura 21. Comparación de desplazamientos, nodo 11.



Cabe anotar que se realizó el análisis con distintos valores de α , obteniendo los mismos resultados de linealidad y no linealidad.

5. IMPLEMENTACIÓN DEL ALGORITMO DMD AL DIAPASÓN

El algoritmo DMD se implementó para el estado transitorio de la simulación del diapasón. La información extraída (deflexión de los nodos) está condicionada por: la excitación en la cara 11 del diapasón (figura 16) aplicando una carga de presión durante una fracción muy pequeña del periodo de tiempo del modo fundamental, condiciones de frontera que asemejan el agarre del diapasón con una mano o su ubicación vertical sobre una mesa y finalmente condiciones iniciales restringiendo el movimiento del diapasón en el tiempo ($t = 0$).

5.1 DMD NODOS INDIVIDUALES

En primer lugar, se realizó la implementación del algoritmo DMD seleccionando nodos de forma individual en distintas zonas geométricas del diapasón, como punto de partida para un análisis posterior donde se pretende diagnosticar y predecir la dinámica del diapasón tomando la totalidad de los nodos.

Cuadro 2. Nodos seleccionados para implementar la técnica DMD

Eje de desplazamiento	Nodo	Δt	Número de ciclos	No. de instantáneas	Ubicación geométrica
y	24, 25, 317	<i>Periodo</i> 60	10	600	Cara 12
	2, 32, 369		100	6000	Cara 16
	144, 166		10	600	Cara 11 y 13
	43, 242		100	6000	Cara 1 y 15
	136, 158		10	600	Cara 11 y 13
	36, 234		100	6000	Cara 1 y 15
x	11, 13		30	1800	Cara 6

Cada uno de los nodos contiene información del desplazamiento del diapasón en una dimensión. Para realizar un análisis detallado se tomaron diferente número de instantáneas según la zona geométrica del diapasón, con la finalidad de observar el desempeño de la técnica DMD.

En el presente trabajo para obtener un resultado satisfactorio el algoritmo sufrió una serie de modificaciones partiendo del algoritmo DMD estándar, luego se implementó la matriz de Hankel (sección 3.4.7) para optimizar la organización de la información y finalmente se usó la técnica de análisis numérico error cuadrático medio²² para encontrar la dinámica del diapasón que minimiza el porcentaje de error y facilita la compilación del código, tanto de la frecuencia fundamental como de la deflexión de cada uno de los nodos comparado con la información original.

5.1.1 DMD estándar. Es el algoritmo planteado inicialmente por Smith, el cual fue estudiado a profundidad en la sección 3.4. La información es organizada en un vector directamente desde la simulación del estado transitorio del diapasón según los criterios de entrada del algoritmo DMD.

El algoritmo se desarrolló en el lenguaje técnico del entorno de trabajo de MATLAB. A continuación, se puede ver la programación de cada una de las ecuaciones que hacen parte del DMD, según lo planteado en la sección 3.4.

```
[U,S,V]= svd(X,'econ'); % Ejecución de la técnica de reducción SVD
r = rank(S); % Cálculo del rango de la matriz S
S_r = S(1:r,1:r); % Valores singulares truncados por el rango de S
U_r = U(:,1:r); % Vectores a la izquierda truncados por el rango S
V_r = V(:,1:r); % Vectores a la derecha truncados por el rango de S
A_reducida = U_r'*X_desplazada*V_r*pinv(S_r); % Reducción a una matriz reducida  $\tilde{A}$ 
[W_r,Lambda]= eig(A_reducida); % Cálculo de los valores y vectores propios de  $\tilde{A}$ 
Omega = diag(log(diag(Lambda)))/dt; % Cálculo de la frecuencia fundamental diapasón
Phi = X_desplazada*V_r*pinv(S_r)*W_r; % Modos del DMD
alpha = S_r*V_r(1,:); % Cálculo de la amplitud b de los modos de la matriz  $\Phi$ 
b = (W_r*Lambda)\alpha;

% Reconstrucción de la dinámica y generación de una ecuación de gobierno para el
comportamiento de un nodo del diapasón
for tt=1:n
```

²² LEHMANN. Erich y CASELLA. George. Theory of point estimation. 2nd ed. New York: Springer Science & Business Media, 1998. p 48-56. ISBN 0-387-98502-6

```
X_DMD(:,tt)= Phi*exp(Omega*tiempo(tt))*b;
end
```

Según la información de entrada al algoritmo debe estar organizada en dos grandes matrices \mathbf{X} y \mathbf{X}' (sección 3.4.4), donde cada vector columna $x_k \in \mathbb{C}^n$, contiene m número de datos de la variable de interés del sistema dinámico en un instante de tiempo determinado, pero ello presentó dos principales problemas: la imposibilidad de trabajar con único nodo (un vector) e igualmente se trabajó con un grupo de nodos, sin embargo, el resultado tampoco fue exitoso. Por esta razón, se recurrió a otra alternativa para solucionar dicho problema de organización, la cual nace del libro “Data driven science and engineering” de Steven L. Brunton y J. Nathan Kutz²³ (capítulo 7) presentada en la siguiente sección.

5.1.2 DMD usando la matriz de Hankel. Al no tener éxito usando el algoritmo DMD estándar, se implementó la matriz Hankel (sección 3.4.7), donde se tiene un valor de ajuste ($n_{ajuste} = \text{número de filas}$) que organiza la información de un solo nodo del diapasón eliminando valores del vector (deflexión en un instante de tiempo) y creando nuevas filas, de tal manera que se obtenga la matriz de Hankel que mejor se ajuste a los resultados esperados.

Se debe tener en cuenta, que el algoritmo mostrado en el siguiente código es solo para la organización de la información en dos matrices \mathbf{X} y \mathbf{X}' , es decir, una vez obtenidas estas matrices, se procede a aplicar la técnica DMD estándar (sección 5.1.1).

```
n_ajuste = 31; % Valor máximo de filas agregadas
X_DMD = zeros(n_ajuste,n); % n = número de instantáneas(columnas)
X = zeros(n_ajuste,(n-n_ajuste));
X_desplazada = zeros(n_ajuste,(n-n_ajuste));
for ii = 1:n_ajuste
    for jj = 1:(n-n_ajuste)
```

²³ BRUNTON, Steven L. y KUTZ, J. Nathan. Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control. Cambridge: Cambridge University Press, 2019.

```

X(ii,jj) = uy_intervalos(nodo,jj+ii-1); % Matriz de desplazamientos X
X_desplazada(ii,jj) = uy_intervalos(nodo,jj+ii); % Matriz desplazada un
                                                Δt en el futuro

end
end

```

Se puede observar en el código dos ciclos anidados (*for*), cuya función es iniciar un llenado de las matrices X y X' partiendo de un vector que contiene información de un solo nodo. Este tipo de llenado de la matriz de Hankel tiene la particularidad de ir eliminando datos del vector fila e ir agregando filas hasta el valor ajuste (n_{ajuste}) indicado al comenzar el código.

La razón de implementar la matriz de Hankel vino dada por la necesidad de aumentar la información contenida en cada nodo, esto se debe: primero a la obligación de crear una matriz de rango mayor (con características según sección 3.4.4) y segundo para el desarrollo de la técnica de reducción SVD, debido a la imposibilidad que tiene la misma para ser implementada en un vector. Sin embargo, esta forma de ejecutar el algoritmo es muy ineficiente, debido a que se debe intentar con diferentes valores de ajuste hasta encontrar el valor correcto que reconstruya la dinámica del nodo analizado.

5.1.3 DMD implementando error cuadrático medio. Es una optimización del algoritmo DMD usando la matriz de Hankel (sección 5.1.2), donde se debe definir un valor mínimo y un valor máximo, los cuales tienen como objetivo generar varias matrices de Hankel de diferentes dimensiones, donde por cada ciclo (*for*), se compara la información del nodo original con la matriz X_{DMD} reconstruida.

5.1.3.1 Funciones características del algoritmo.

- La función *FDMD_individual* corresponde al algoritmo DMD estándar, el cual es alimentado en cada ciclo (*for*) por una matriz de Hankel diferente, tomando como rango de organización un valor mínimo y un valor máximo preestablecido.

```
[X_DMD] = FDMD_individual(n_ajuste,n,nodo,dt,tiempo);
```

- La función *immse* hace parte de la documentación de MathWorks, la cual tiene como entradas dos vectores; el vector generado a partir de la primera fila (*Desplazamiento_uy*) de la matriz **X_DMD** de la salida de la función *FDMD_individual* con el vector del nodo desplazamiento introducido para el análisis. Esta función mide el promedio de los errores al cuadrado de los dos vectores comparados.

```
error(cont) = immse(Desplazamiento_uy,nodo);
```

- La función *find* que igualmente es de la documentación de MathWorks retorna la ubicación del mínimo error, calculado con la función *immse*.

```
Encontrar_error = find(error==(min(error)));
```

- La función *FDMDajustada_individual* corresponde igualmente al algoritmo DMD estándar con la característica que tiene como entrada un valor de ajuste nuevo (*n_ajuste*) calculado según el error encontrado con la función *find*.

```
[X_DMDajustada,Omega] = FDMDajustada_individual(n_ajuste,n,nodo,dt,tiempo);
```

5.1.3.2 Algoritmo. Este es el algoritmo completo que da una solución óptima a los problemas que se han presentado en los algoritmos DMD anteriores (sección 5.1.1 y 5.1.2), se puede observar claramente cada una de sus líneas programadas y su explicación según las funciones expuestas en la sección 5.1.3.1.

```
% Rango mínimo y máximo
```

```
minimo = 10;
```

```
maximo = 100;
```

```
error = zeros(1,maximo-minimo+1);
```

```

cont = 0;

%% Mejor ajuste
for i = minimo:maximo

    cont = cont+1;
    n_ajuste = i;
    % Función para crear varias matrices de Hankel y aplicar la técnica DMD
    [X_DMD] = FDMD_individual(n_ajuste,n,nodo,dt,tiempo);
    Desplazamiento_uy = real(X_DMD(1,:)); % Se toma la primera fila de la matriz
                                         reconstruida

    error(cont) = immse(Desplazamiento_uy,nodo); % Cálculo del error cuadrático
                                                medio

    Encontrar_error = find(error==(min(error))); % Encontrar la ubicación del
                                                error mínimo

    K = error(Encontrar_error); % Valor del error mínimo calculado

end

```

Luego, se parte de la ubicación del mínimo error encontrado (número máximo de filas agregadas) para generar una matriz final de Hankel con la capacidad de reconstruir la dinámica del nodo analizado de forma exitosa.

```

n_ajuste = (Encontrar_error+minimo-1); % Se genera un nuevo valor de ajuste
% Función para crear la matriz de Hankel final y reconstruir la dinámica del
diapasón
[X_DMDajustada,Omega] = FDMDajustada_individual(n_ajuste,n,nodo,dt,tiempo);
Desplazamiento_uy_ajustado = real(X_DMDajustada(1,:)); % Información de los datos
                                                         desplazamiento DMD para 1
                                                         nodo

```

Con el algoritmo DMD implementando el error medio cuadrático, se tiene un mayor rango de análisis, el cual reconstruye la dinámica del diapasón, sin necesidad de compilar para varios valores ajuste hasta encontrar el ideal con la capacidad de reconstruir el sistema.

5.2 DMD DIAPASÓN COMPLETO

La implementación de la técnica DMD al diapasón completo, es decir, a la totalidad de los nodos es basado en los algoritmos DMD de la sección 5.1, con características propias del código que facilitaron la reconstrucción de la dinámica, con base en la geometría 3D del diapasón.

5.2.1 Datos de entrada al algoritmo. Para desarrollar la técnica DMD en la totalidad de los nodos del diapasón se tuvieron en cuenta los siguientes criterios, con la finalidad de obtener la reconstrucción y el modelo matemático correcto de la dinámica del diapasón acústico.

Cuadro 3. Datos de entrada algoritmo DMD diapasón completo.

Características de entrada	Criterios del diapasón
Número de ciclos	100
Diferencial de tiempo (dt)	$\frac{\textit{Periodo}}{6}$
Número de instantáneas para el cálculo	600
Número de instantáneas máximas	6,000
Número de nodos	2,125
Valor mínimo	10
Valor máximo	200
Eje de desplazamiento	y

Para entender a detalle cada uno de los ítems de la tabla anterior, revisar el capítulo 4 y sección 5.1.

5.2.2 Funciones características DMD. La base de este algoritmo es el algoritmo DMD con error cuadrático medio, sin embargo, fue necesario implementar funciones que optimizaran el desarrollo de la técnica.

- La función *eliminar_ceros* se creó con el fin de eliminar filas dadas por vectores nulos de la matriz completa del diapasón, estos nodos no tienen información de desplazamiento debido a las restricciones establecidas en la simulación original del diapasón (capítulo 4). Estas filas de ceros fueron eliminadas para evitar inconsistencias en el cálculo del error con la función *immse* vista en la sección 5.1.3.1.

```
[Data_original, uy_intervalos] = eliminar_ceros(uy_nodos_totales, instantaneas);
```

- La función *calculo_frecuencias* determina la frecuencia fundamental en *Hz* presente en el vector Ω calculado con el algoritmo DMD estándar (sección 5.1.1). Esta función toma el valor imaginario de un dato Ω_k del vector Ω , según tres condiciones de análisis: cálculo de la frecuencia filtrado por la parte real con el valor más cercano a cero, incluyendo números positivos y negativos, cálculo de la frecuencia filtrando únicamente la parte real con el valor negativo más cercano a cero, cálculo de la frecuencia filtrado únicamente para la primera fila del vector Ω .

```
[Frec_abs, Frec_neg, Frec_1er_fila] = calculo_frecuencias(mm, Omega, nodo);
```

- Los valores mínimo y máximo tomados para el algoritmo DMD con error cuadrático medio (sección 5.1.3), donde no siempre con este rango es posible encontrar la frecuencia fundamental teniendo en cuenta las tres condiciones de selección (función *calculo_frecuencias*). Se observó que el mayor porcentaje de error es con las frecuencias obtenidas son para el mínimo valor de la parte real de datos negativos y positivos. Teniendo en cuenta lo anterior, esta función encuentra el nodo que para tales condiciones posee valores atípicos, con base en lo alejado que este valor se encuentre del patrón de datos, lo cual se lleva a cabo haciendo una comparación por desviación estándar de la mediana o por un rango de valores percentiles.

```
[nodo_atipico_abs,nodo_atipico_neg,nodo_atipico_1er_fila] =
Valor_atipico(All_frec_abs,All_frec_neg,All_frec_1er_fila);
```

5.2.3 Algoritmo DMD diapasón completo. Para un fácil entendimiento del código se van a realizar una serie de pasos que explican en detalle el funcionamiento del algoritmo.

- **Paso 1.**

Para llevar a cabo la implementación de la técnica DMD en el diapasón completo, fue necesario organizar la matriz de información de desplazamientos, donde la misma originalmente consta de 2,125 nodos con una totalidad de 6,000 instantáneas, teniendo en cuenta el alto consumo computacional que se requiere para compilar dicha información fue necesario hacer una reducción a 600 instantáneas, es decir, se acomodó el algoritmo que seleccionará un vector de desplazamiento cada 10 instantáneas hasta llegar a 6,000.

Una vez lista la matriz desplazamientos de tamaño $2,125 \times 600$, fue necesario eliminar 48 filas formadas por vectores nulos, obteniendo una matriz de tamaño $2,077 \times 600$. A continuación, se muestra la programación para dicho cálculo.

```
%% Organización de datos previa a la eliminación de ceros, con el propósito de
obtener la matriz final de trabajo
m = size(uy_nodos_totales,1);

for cont = 1:instantaneas
    uy_intervalos = zeros(m,cont);
end
for cont = 1:instantaneas
    column = cont*(instantaneas_max/instantaneas); % Organización de los datos
                                                    según las instantáneas
                                                    deseadas
    uy_intervalos(:,cont) = uy_nodos_totales(:,column); % Guardar todo en una
                                                         matriz
```

```

end

%% Eliminación de filas nulas
Data_original = uy_intervalos;
eliminar_ceros = sum(abs(uy_intervalos),1)==0;
uy_intervalos(sum(abs(uy_intervalos),2)==0,:)=[];
uy_intervalos(:,eliminar_ceros) = [];
end

```

El código anterior (función *eliminar_ceros*) también cuenta con la organización previa de los datos seleccionando 600 instantáneas en total, sin embargo, se debe tener en cuenta, que la función fue programada para que el usuario seleccionará el número de instantáneas que desee para su posterior uso en el algoritmo DMD.

- **Paso 2.**

Esta parte del desarrollo es muy similar a la solución optimizada con el algoritmo DMD con error cuadrático medio (ver sección 5.1.3), su principal diferencia radicó en la modificación del código para que adquiriera la capacidad de recorrer cada uno de los nodos del diapasón y a cada uno de ellos generarle las respectivas matrices de Hankel y guardar la primera fila de la matriz **X_{DMD}**, para encontrar el mínimo error comparando el vector original con el obtenido por el DMD estándar.

Finalmente, se desarrolló una nueva matriz de Hankel para calcular nuevamente el DMD estándar y de esta manera obtener el resultado deseado. Lo anterior con el propósito de generar un algoritmo capaz de ejecutar esta serie de procedimientos de forma automática, sin necesidad de compilar nodo por nodo.

- **Paso 3.**

Una vez encontrado el ajuste con menor error, se aplicó nuevamente el algoritmo DMD con la nueva condición de entrada, obteniendo la primera fila de la matriz reconstruida.

Para almacenar los desplazamientos de todos los nodos, se crea una matriz que guarda el primer vector fila de la matriz **X_{DMDajustada}** para cada nodo.

```

%Primera fila de la reconstrucción DMD una vez aplicado el error
Desplazamiento_uy_ajustado = real(X_DMDajustada(1,:));

% Matriz con todos los nodos del diapasón (sin las filas de ceros)...
% ...obtenida con la primer fila de cada DMD ajustado con el error
Desplazamientos_totales(nodo,:) = Desplazamiento_uy_ajustado;

```

- **Paso 4.**

En esta fase del desarrollo del código se utilizan las funciones *calculo_frecuencias* y *valor_atipico* (sección 5.2.2), donde se obtiene la información final del código algoritmo DMD diapasón completo, para su posterior reconstrucción en un código capaz de reubicar la información del desplazamiento en *y* de cada uno de los nodos en la ubicación geométrica y en el instante de tiempo adecuado e igualmente agregar las filas nulas (filas de ceros), necesarias para que el modelo de cada uno de los modos según la información de cada nodo sea correcta. Esta reconstrucción será explicada en detalle en el capítulo 6.

A continuación, se muestra en detalle la recopilación de los datos para su posterior reconstrucción y análisis.

```

[Frec_abs, Frec_neg, Frec_1er_fila] = calculo_frecuencias(mm, Omega, nodo);

%% Vectores que acumulan las frecuencias calculadas para cada uno de los
nodos en los tres casos
All_frec_abs(nodo,:) = Frec_abs;
All_frec_neg(nodo,:) = Frec_neg;
All_frec_1er_fila(nodo,:) = Frec_1er_fila;

%% Análisis de datos para las frecuencias calculadas en los tres casos
(outliers)
v_logico_abs = isoutlier(All_frec_abs(:,1), 'percentiles', [15, 100]);
nodo_atipico_abs = find(v_logico_abs);

v_logico_neg = isoutlier(All_frec_neg(:,1));

```

```
nodo_atipico_neg = find(v_logico_neg);
```

```
v_logico_1era_fila = isoutlier(All_frec_1er_fila);
```

```
nodo_atipico_1er_fila = find(v_logico_1era_fila);
```

6. RESULTADOS Y VALIDACIÓN

6.1 ANÁLISIS MODAL

Como se mencionó anteriormente, el objetivo del análisis modal es obtener los modos de frecuencias o frecuencias naturales de un objeto que vibra libremente.

Se resuelve para un rango de frecuencias de -1 [Hz] hasta $4,000$ [Hz], obteniendo ocho modos, cuyo valor se muestra a continuación:

Se parte de una frecuencia negativa con el fin de evitar error matemático en la solución del modelo PDE de usado a partir de Toolbox.

Cuadro 4. Modos de frecuencia obtenidos por el análisis modal y sus valores aproximados.

No. Modo	Frecuencia [Hz]
1	0.0077799
2	0.0034067
3	0.0020257
4	0.0031827
5	0.006891
6	0.01048
7	460.42
8	706.34
9	1911.5
10	2105.5
11	2906.5
12	3814.7

Del cuadro anterior, se observa que el valor de los seis primeros modos hallados es cercano a cero. Lo anterior se presenta debido a que estos corresponden a los modos de cuerpo rígido.

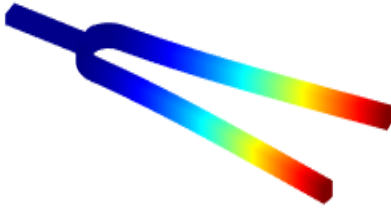
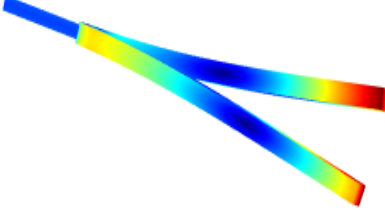
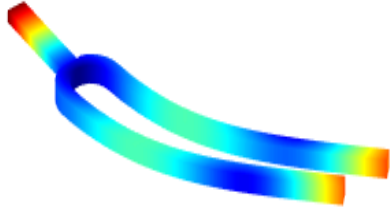

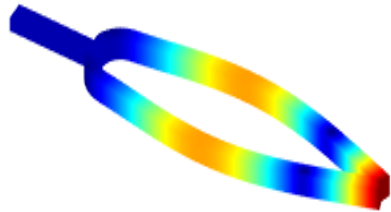
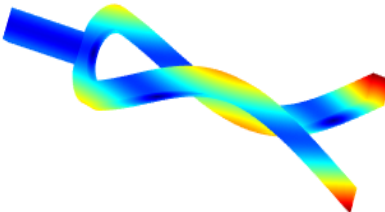
Para un análisis modal sobre un objeto que no presenta ninguna restricción de movimiento o cargas externas actuantes, habrá seis modos de cuerpo rígido, los

cuales representan la traslación y rotación libre sin sufrir algún tipo de deformación. El número de modos de cuerpo rígido equivale a los grados de libertad que presenta el objeto.

Al descartar los seis primeros modos, se obtiene que el séptimo modo hallado corresponde al primer modo flexible del diapasón y por ende a su frecuencia fundamental; modo principal del desarrollo de este proyecto.

A continuación, se ilustra gráficamente el comportamiento de los seis modos flexibles determinados:

Figura 22. Representación gráfica de los modos flexibles determinados.

	
460.42 [Hz]	706.34 [Hz]
	
1911.5 [Hz]	2105.5 [Hz]
	
2906.5 [Hz]	3814.7 [Hz]

Validando lo mencionado en el marco referencial, se observa que para el modo fundamental y para el quinto modo flexible, el desplazamiento de las puntas ocurre principalmente en el eje y (modos simétricos). Además, efectivamente el quinto modo flexible equivale a aproximadamente 6.25 veces el modo fundamental (modo clang)

Como se observó en el capítulo 3, generalmente los diapasones oscilan a una frecuencia de 440 [Hz], sin embargo, se ve en el cuadro 4 que la frecuencia fundamental corresponde a aproximadamente 460.42 [Hz], la cual es cercana, pero difiere de la general por razones dimensionales.

6.2 ANÁLISIS TRANSITORIO.

Por motivo del mallado realizado y el cual se ilustró en la sección 4.3.2, este análisis arroja como resultado los desplazamientos de estos nodos en sus tres dimensiones para intervalos de tiempo (en segundos) preestablecidos. Para el caso del intervalo de tiempo más grande, se observa que el arreglo de datos obtenido corresponde a una gran matriz de 2125×6000 , es decir 12'750,000 datos, por cada dimensión (x, y, z).

Como se mencionó en secciones anteriores, la implementación del algoritmo DMD se enfoca en el comportamiento del diapason cuando este es excitado en el modo fundamental, razón por la cual en principio sólo se tienen en cuenta principalmente los desplazamientos sobre el eje y . Sin embargo, debido a que el mango presenta desplazamientos importantes sobre el eje x que además son de comportamiento no lineal, se consideran los datos obtenidos para esta componente.

Figura 23. Desplazamientos en el eje y para todos los nodos del diapasón y 6,000 instantes de tiempo.

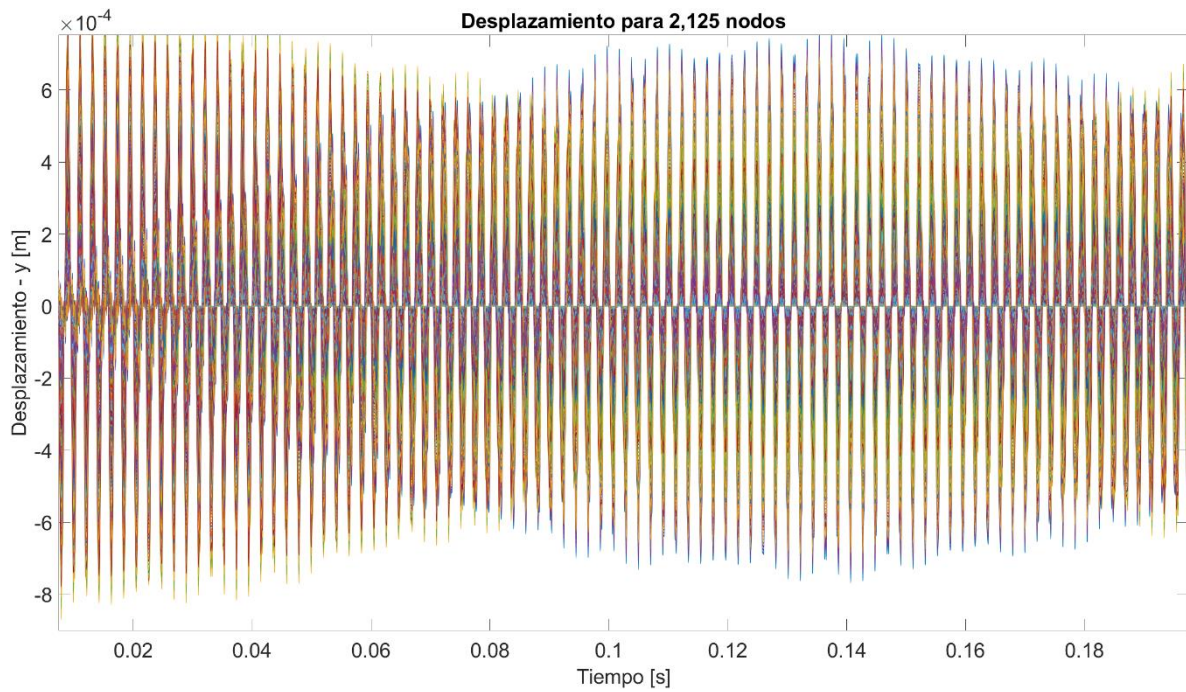


Figura 24. Acercamiento de la figura 23.

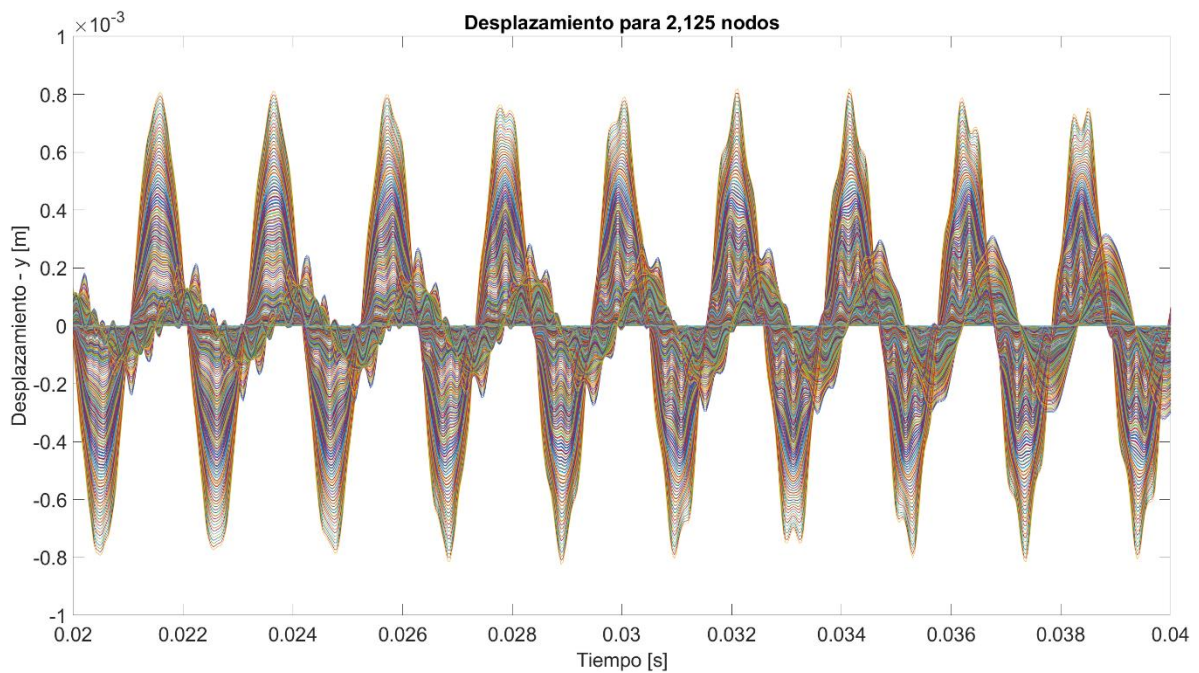
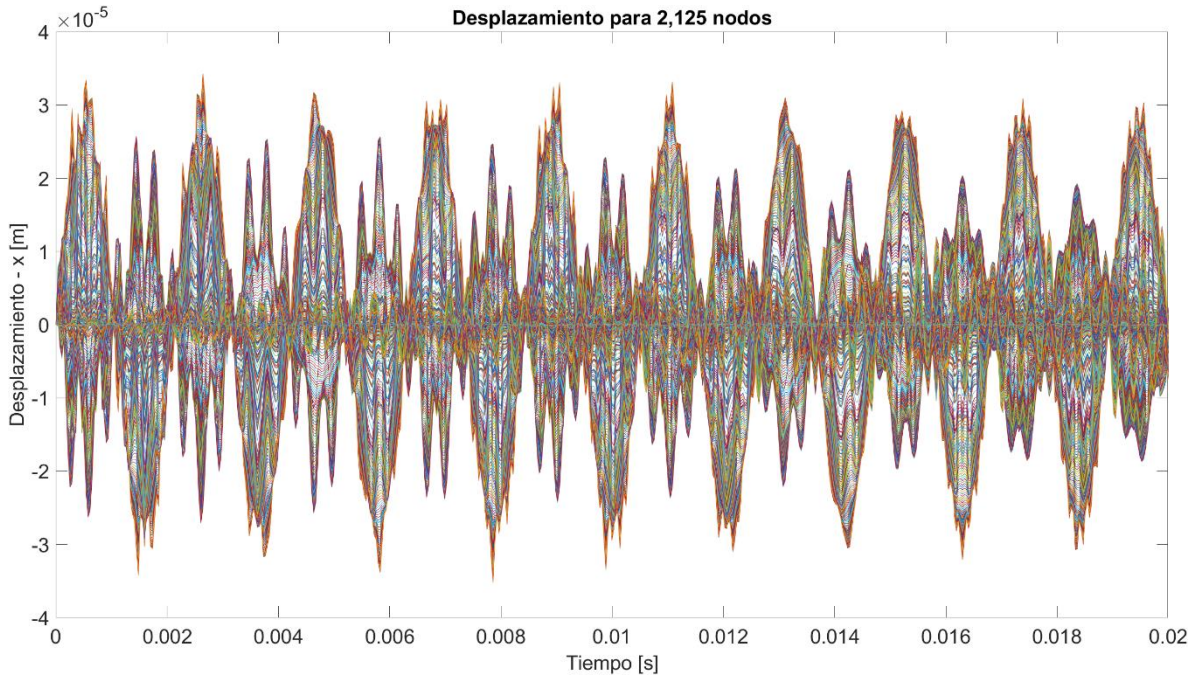


Figura 25. Desplazamientos en el eje x para todos los nodos del diapasón.



6.3 DMD NODOS INDIVIDUALES

En esta sección se discuten los resultados y la validación de la aplicación de la técnica DMD de los algoritmos DMD estándar, DMD usando la matriz de Hankel y DMD implementando error cuadrático medio (ver sección 5.1). De particular interés es la comparación de los desplazamientos (x,y) y la frecuencia fundamental con respecto a la información original según el comportamiento de cada nodo del diapasón acústico. Para la validación de los resultados se usa la transformada rápida de Fourier (FFT) para representar la señal de cada nodo en el dominio del tiempo (desplazamientos) en una señal en el dominio de la frecuencia.

6.3.1 Algoritmos característicos para la reconstrucción. Para representar de forma clara los datos dados por los algoritmos de la sección 5.1, se presenta en un gráfico de barras el error calculado de cada matriz de Hankel con respecto a los datos desplazamientos originales.

Se explica el código para graficar los desplazamientos en el dominio del tiempo y finalmente para su verificación se usa la función *Diapason_FFT*, la cual tiene como objetivo calcular la transformada rápida de Fourier.

- La siguiente fracción del código tiene como objetivo graficar el vector error arrojado por la función *immse* (sección 5.1.3.1). Sus datos de entrada son el valor máximo, el valor mínimo y el vector error que contiene el promedio de los errores al cuadrado de los dos vectores comparados (vector desplazamientos originales y la primera fila de la matriz **X_DMD**).

```
figure
bar(minimo:maximo,error)
title({'Error medio cuadrático'})
xlabel('Rango analizado'); ylabel('Error (m)')
xlim([10 100])
```

- A continuación, se presenta la forma de graficar los desplazamientos de cada nodo, donde se tiene un vector (*tiempo*) el cual contiene la evolución en el tiempo según el estado transitorio con un Δt constante en el futuro.

```
figure
plot(tiempo,nodo,'b-','linewidth',1.5);hold on; % Grafica vector datos original
plot(tiempo,Desplazamiento_uy_ajustado,'r--','LineWidth',1);grid on % Grafica
                                                                    resultados
DMD
title({'Desplazamientos original-DMD ajustado'})
legend('Original','DMDajustado');
xlabel('Tiempo (s)'); ylabel('Desplazamiento en x (m)')
```

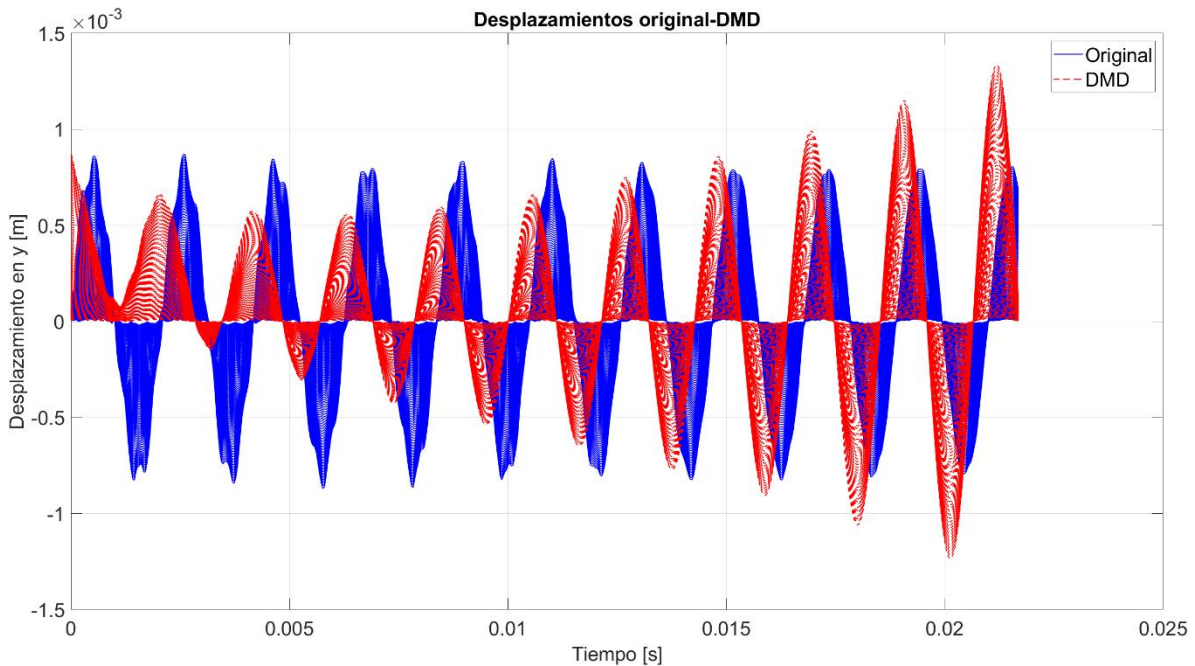
- La función *Diapason_FFT* es la encargada de calcular la transformada discreta de Fourier (DFT) con la que se identifica la frecuencia de vibración de cada uno de los nodos analizados (cuadro 2). Las entradas son la variable *Desplazamientos* (vector de la deflexión según cada nodo) y la variable

muestreo , la cual es definida como $60/Periodo$. Las salidas son f , la abscisa donde se encuentra la frecuencia fundamental y P_1 (el cambio de la presión en diferentes puntos del tiempo).

```
function [f,P_1] = Diapason_FFT(Desplazamientos,muestreo)
L = numel(Desplazamientos);
L = L + mod(L,2);
Y = fft(Desplazamientos);
P_2 = abs(Y/L);
P_1 = P_2(1:L/2+1);
P_1(2:end-1) = 2*P_1(2:end-1);
f = muestreo*(0:(L/2))/L;
end
```

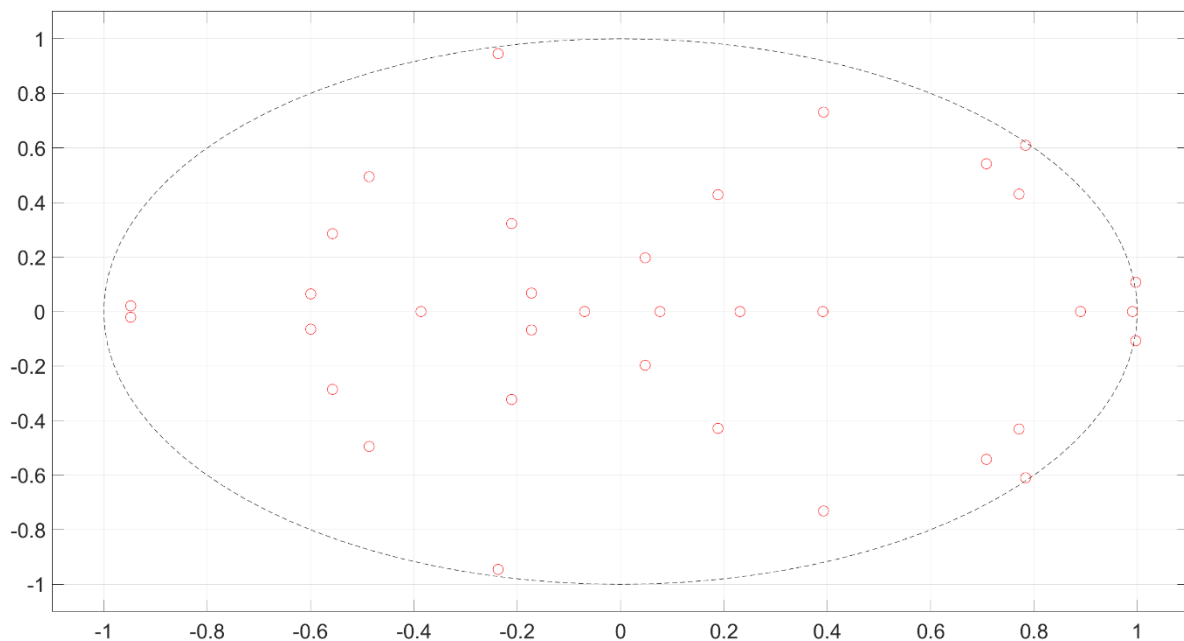
6.3.2 DMD estándar. En la sección 5.1.1 se mencionó que esta forma de solución contiene la programación general del algoritmo DMD, sin embargo, sus resultados no son exitosos debido a la organización de la información.

Figura 26. Representación datos originales y datos DMD para la cara 11.



En la gráfica anterior, se tomaron la totalidad de los nodos de la cara 11 organizándolos de la forma convencional, según las matrices X y X' , sin embargo, su resultado no fue exitoso, donde se puede observar la gráfica roja con ajuste aceptable en alguna instantánea notando un decrecimiento y crecimiento secuencial a medida que transcurre el tiempo. Por ello, se tuvo la necesidad de buscar una manera más eficiente de organizar los datos teniendo una solución óptima con la organización según los criterios de la matriz de Hankel.

Figura 27. Representación de los valores propios en un círculo unitario para la cara 11.



Esta forma de organizar los datos no fue exitosa básicamente porque la mayoría de sus valores propios se encuentran por dentro del círculo unitario (figura 27), donde dicho comportamiento es catalogado como ruido por razones que se expondrán más adelante.

6.3.3 DMD usando la matriz de Hankel

6.3.3.1 Resultados. En las figuras 28, 29 y 31 se puede observar en color azul la gráfica del vector desplazamiento original y en color verde la gráfica de la primera fila de la matriz X_{DMD} del algoritmo DMD (sección 5.1.2).

Figura 28. Desplazamientos datos originales y datos DMD nodo 166.

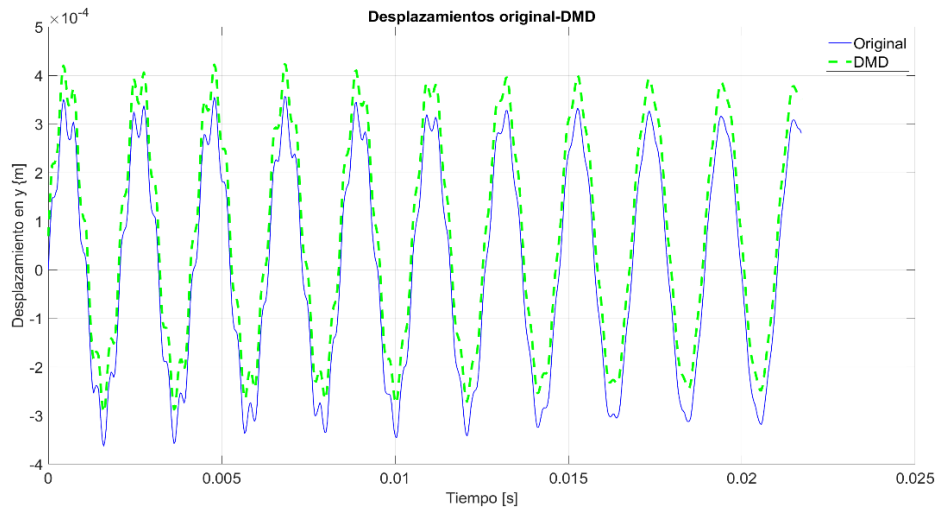


Figura 29. Desplazamientos datos originales y datos DMD nodo 43.

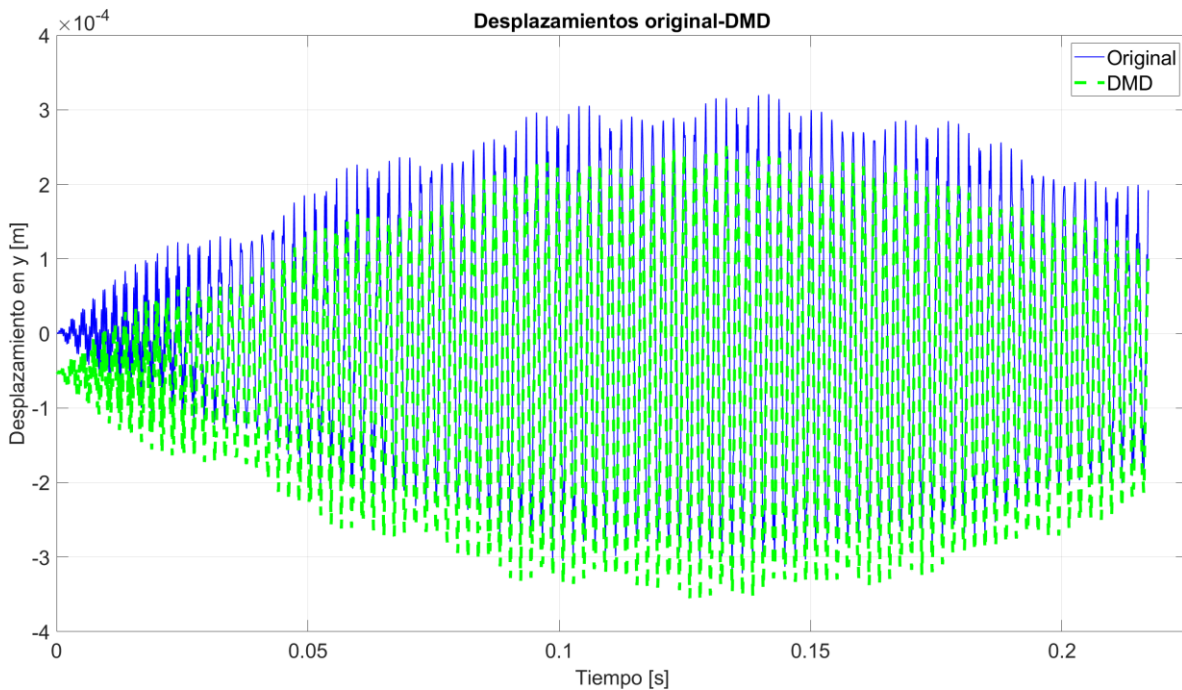
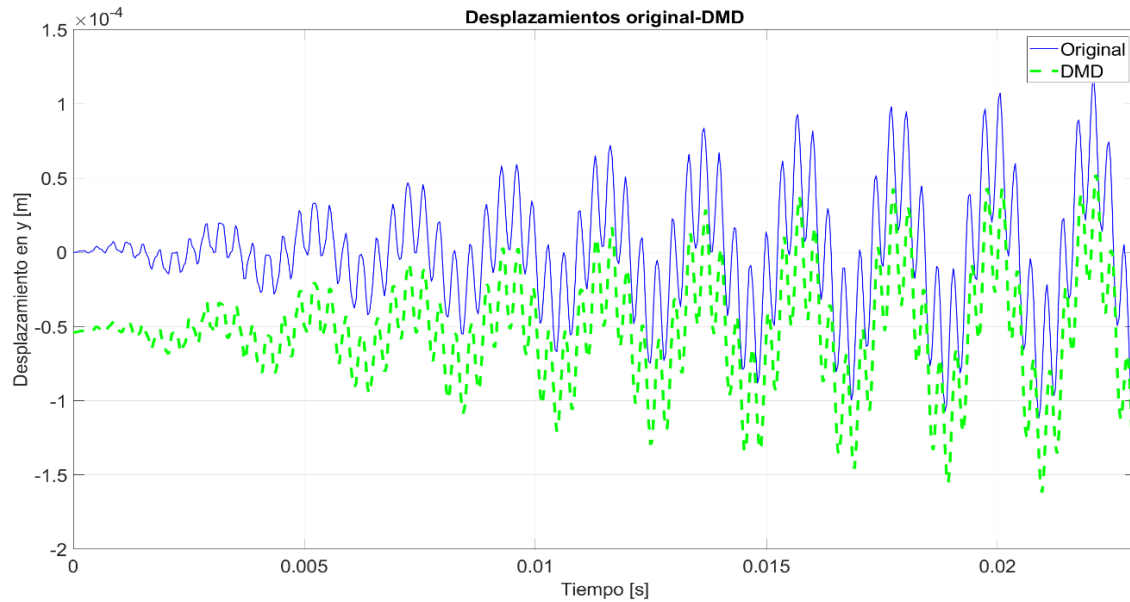
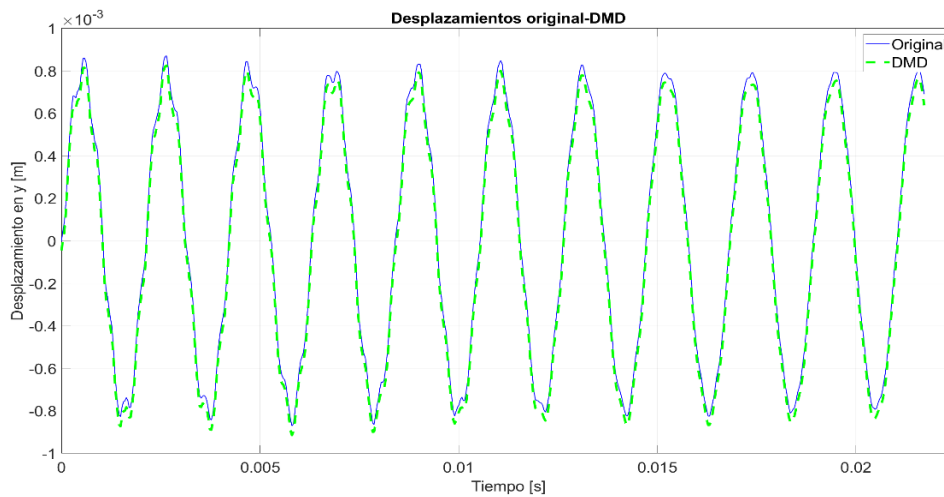


Figura 30. Acercamiento de la figura 29.



El nodo 43 presenta ese comportamiento creciente al inicio, debido a que la excitación inicialmente solo se aplica en la cara 12 y al transcurrir el tiempo se empieza a transmitir la energía a los demás nodos hasta llegar a la cara 1 y 15 (nodo 43). Por este motivo, se eligieron 6000 instantáneas con la finalidad de tener un tiempo prudente permitiendo un análisis amplio en estas secciones del diapasón.

Figura 31. Desplazamientos datos originales y datos DMD nodo 24.



En las gráficas anteriores se puede notar un leve desfase en las amplitudes. Esto sucede básicamente por el hecho de usar una sola matriz de Hankel, donde es necesario usar varios valores n_{ajuste} hasta encontrar el mínimo error con respecto a la gráfica de información original. Por esta razón, es necesario generar varias matrices de Hankel de forma automática hasta encontrar la que contenga el mínimo error entre la información original y la calculada con el algoritmo DMD.

6.3.3.2 Validación. A los nodos 166, 43 y 24 en el dominio del tiempo se les realiza la FFT (figuras 32, 33 y 34), con la finalidad de identificar la frecuencia de vibración de cada uno de los nodos. En la sección 6.1 se estableció que la frecuencia fundamental del diapason es aproximadamente **460.42 Hz**, la cual es verificada tanto para la información calculada a partir del estado transitorio, como para la información obtenida del algoritmo DMD.

Se debe tener claro que la técnica DMD calcula un vector Ω con las frecuencias predominantes del sistema dinámico, es decir, la frecuencia fundamental es verificada tanto para la información de los desplazamientos de la primera fila de la matriz X_{DMD} y obtenida a partir del vector Ω de frecuencias.

Figura 32. FFT datos originales y datos DMD nodo 166.

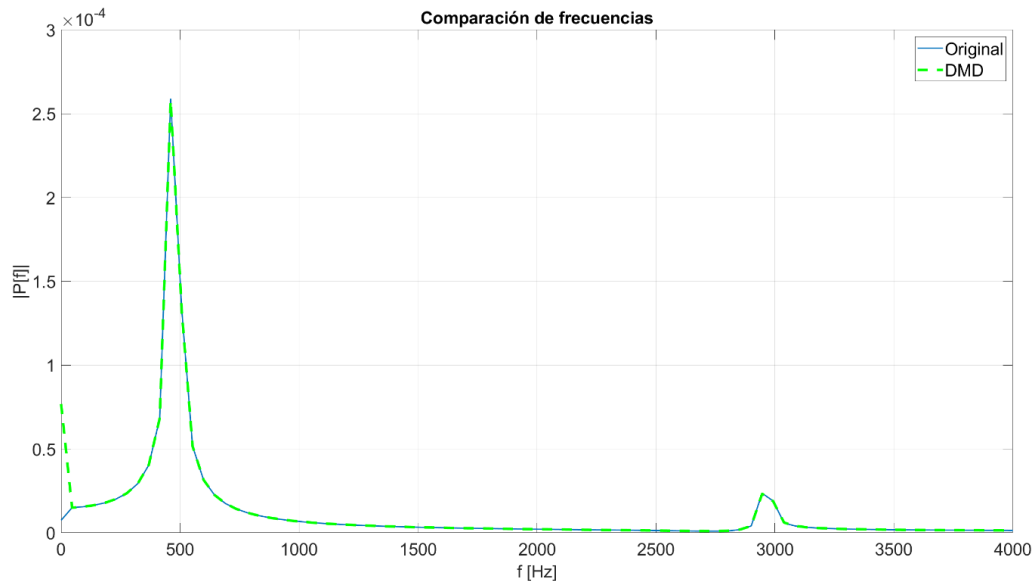


Figura 33. FFT datos originales y datos DMD nodo 43.

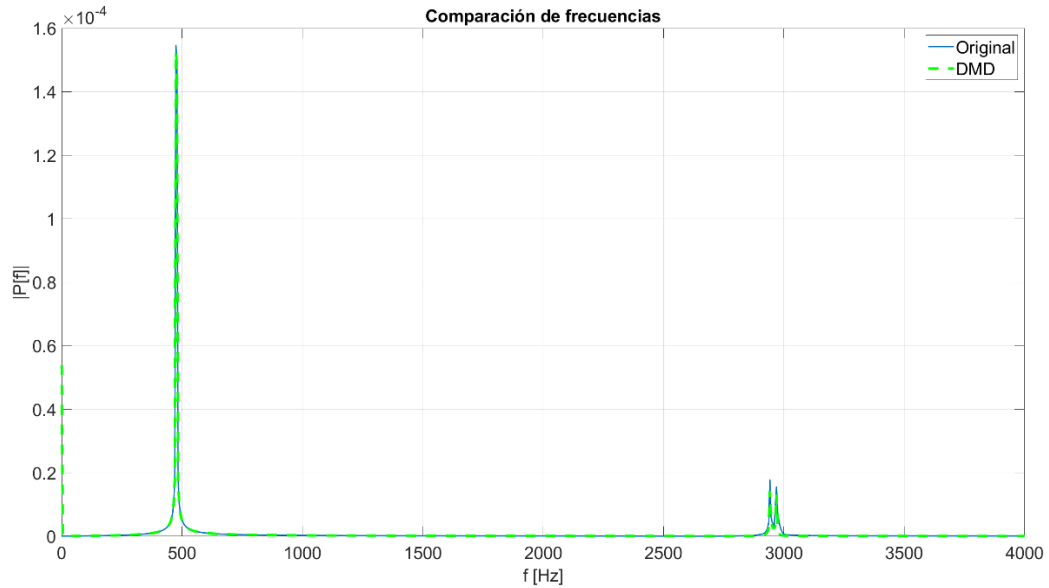
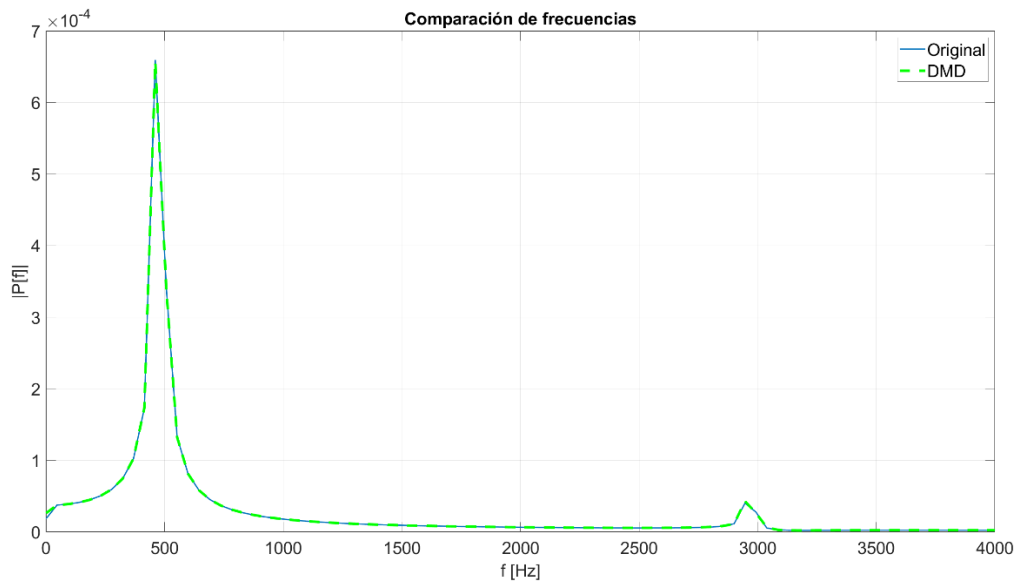


Figura 34. FFT datos originales y datos DMD nodo 24.



6.3.4 DMD implementando error cuadrático medio

6.3.4.1 Resultados. En los siguientes gráficos de barras se puede observar el ajuste, por medio del análisis matemático error cuadrático medio, donde la abscisa corresponde al valor n_{ajuste} (número de filas agregadas para la construcción de

la matriz de Hankel), mientras que la ordenada corresponde al error calculado en metros según el vector desplazamiento original y DMD respectivamente.

Figura 35. Error calculado para el nodo 11.

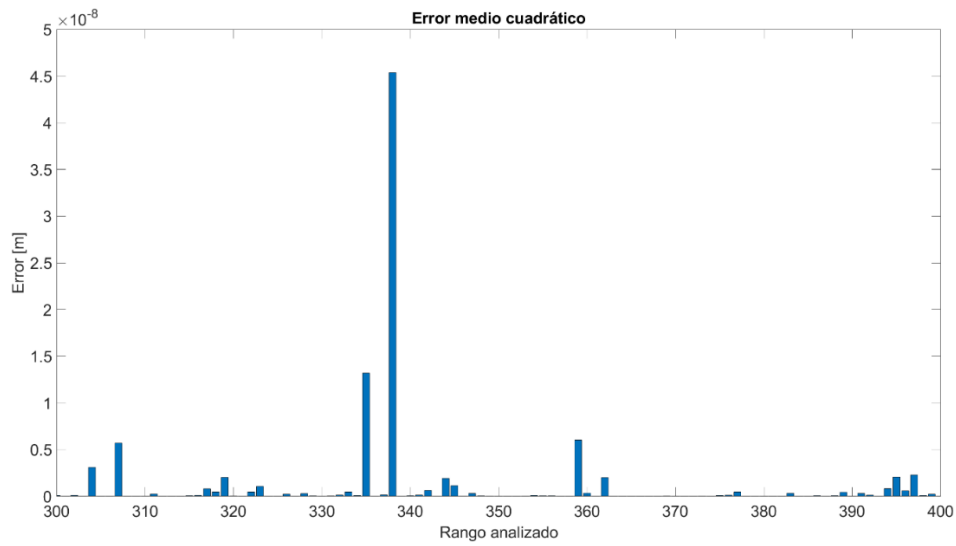


Figura 36. Error calculado para el nodo 158.

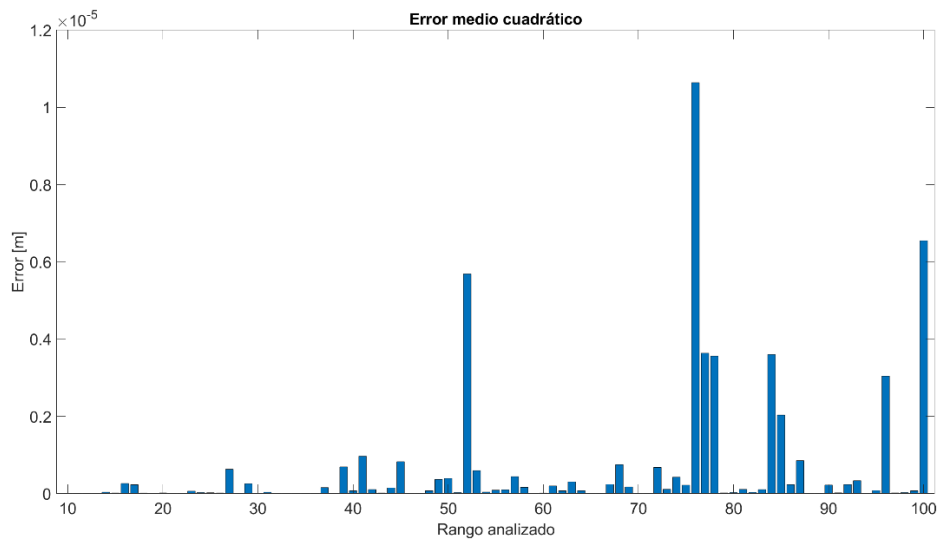
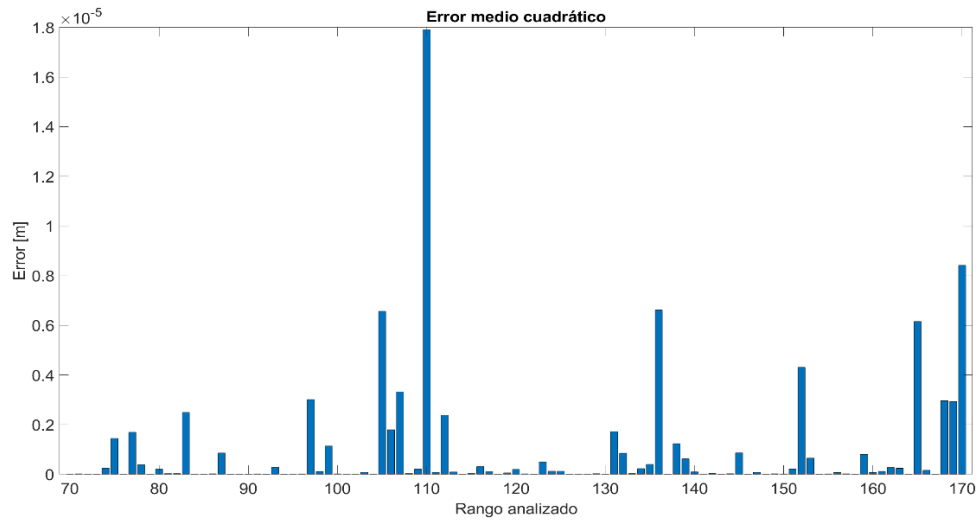


Figura 37. Error calculado para el nodo 234.



El algoritmo automáticamente recorre cada uno de los errores calculados y por medio de la función *find* encuentra la primera fila de la matriz de Hankel que presente el mínimo error, la cual es usada para la reconstrucción del vector desplazamiento, por medio de una nueva matriz de Hankel.

Figura 38. Desplazamientos datos originales y datos DMD nodo 11.

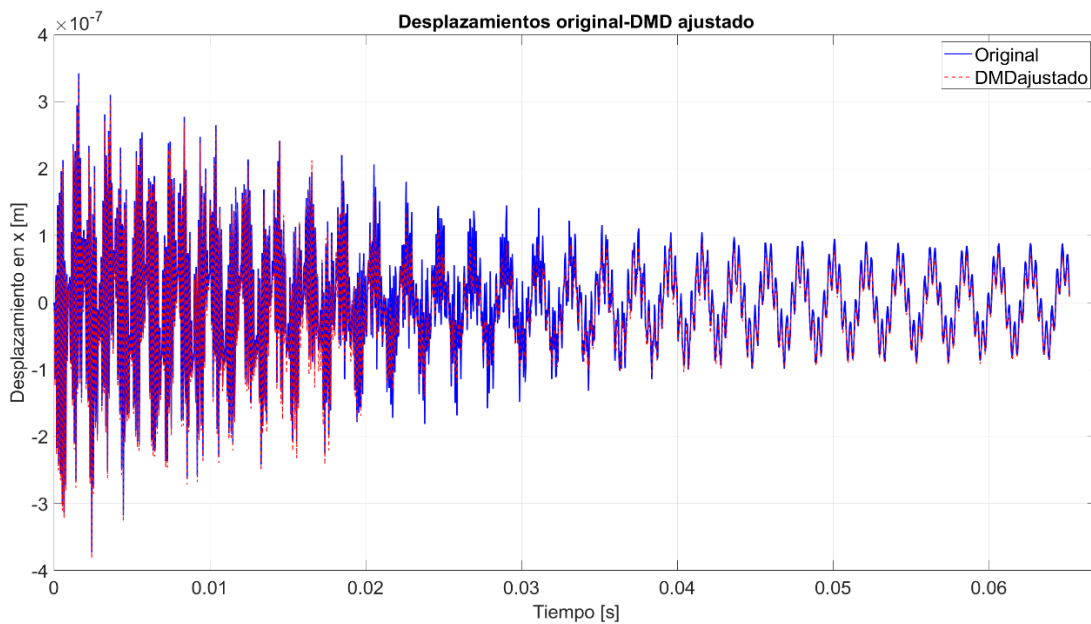


Figura 39. Acercamiento de la figura 38 para un intervalo de tiempo de 0.01 a 0.02 segundos.

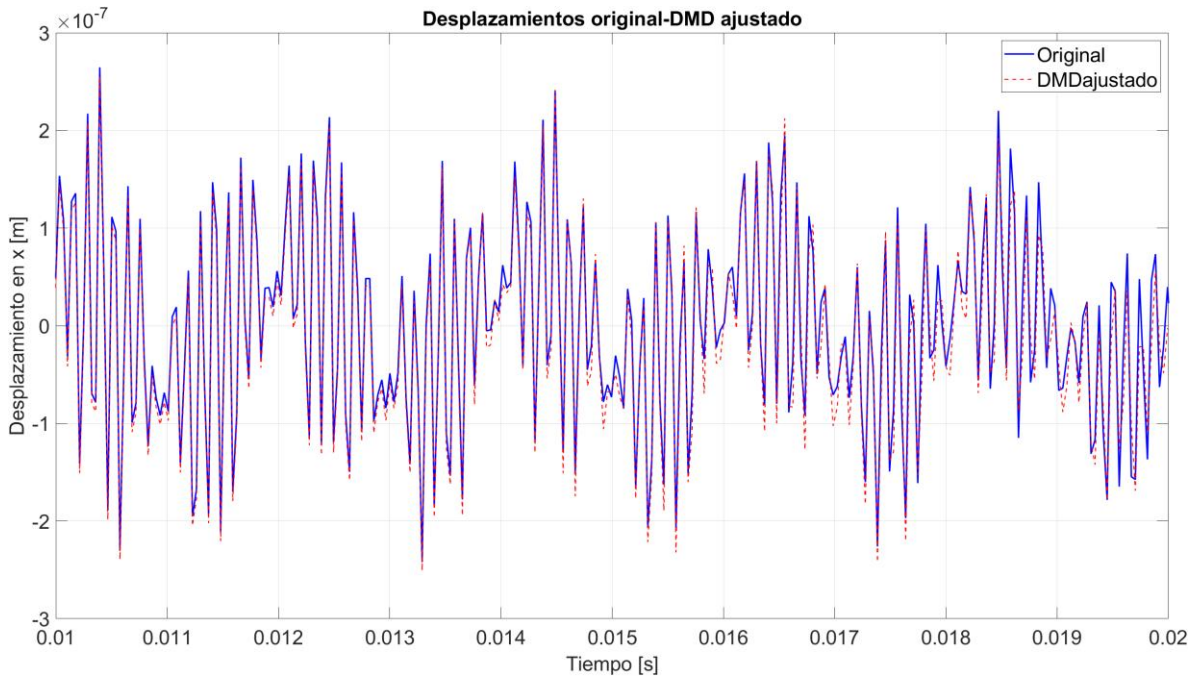


Figura 40. Acercamiento de la figura 38 para un intervalo de tiempo de 0.055 a 0.065 segundos.

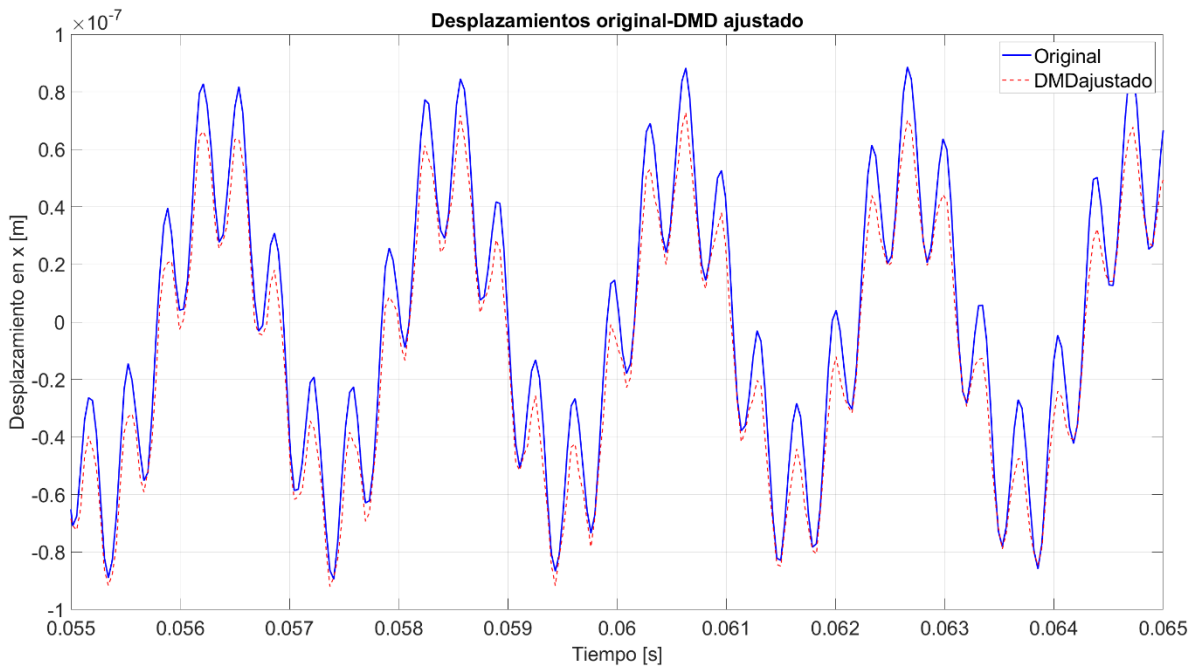


Figura 41. Desplazamientos datos originales y datos DMD nodo 158.

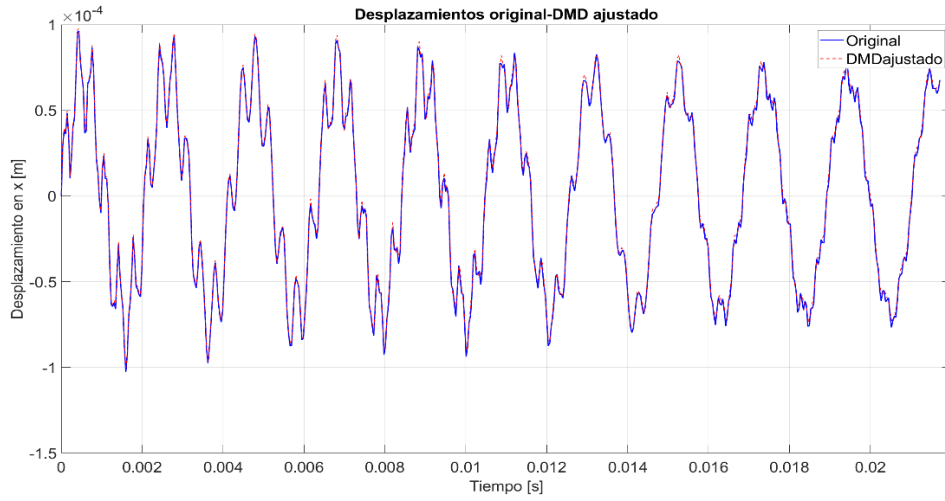


Figura 42. Desplazamientos datos originales y datos DMD nodo 234.

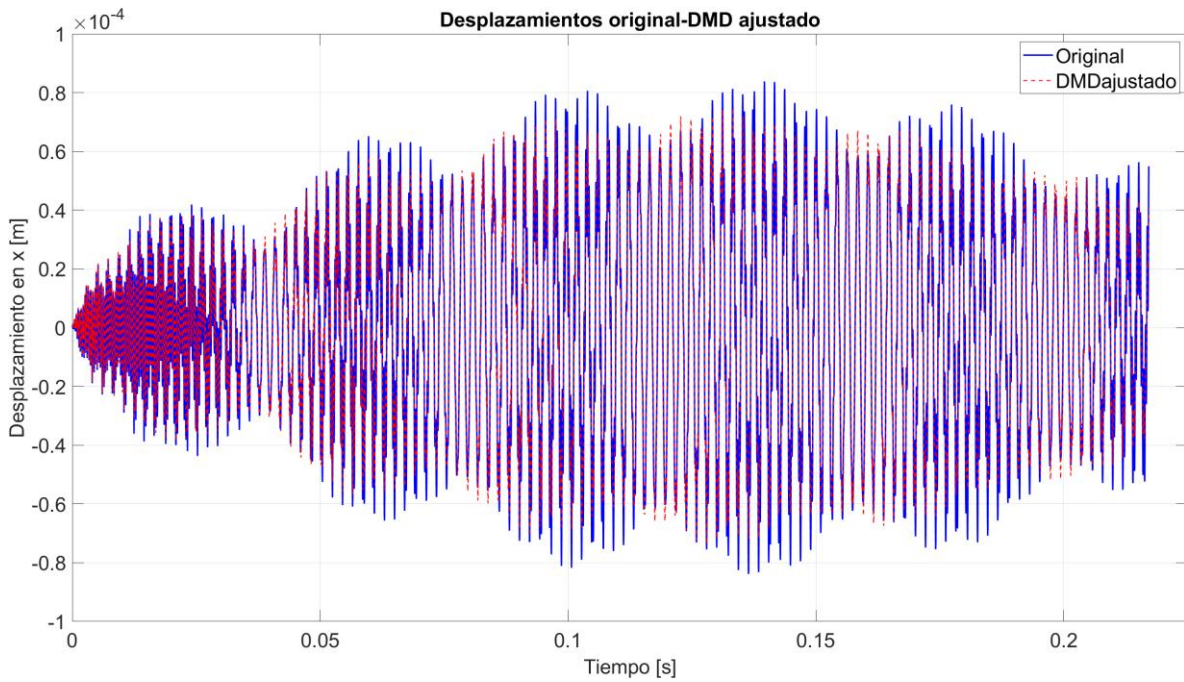
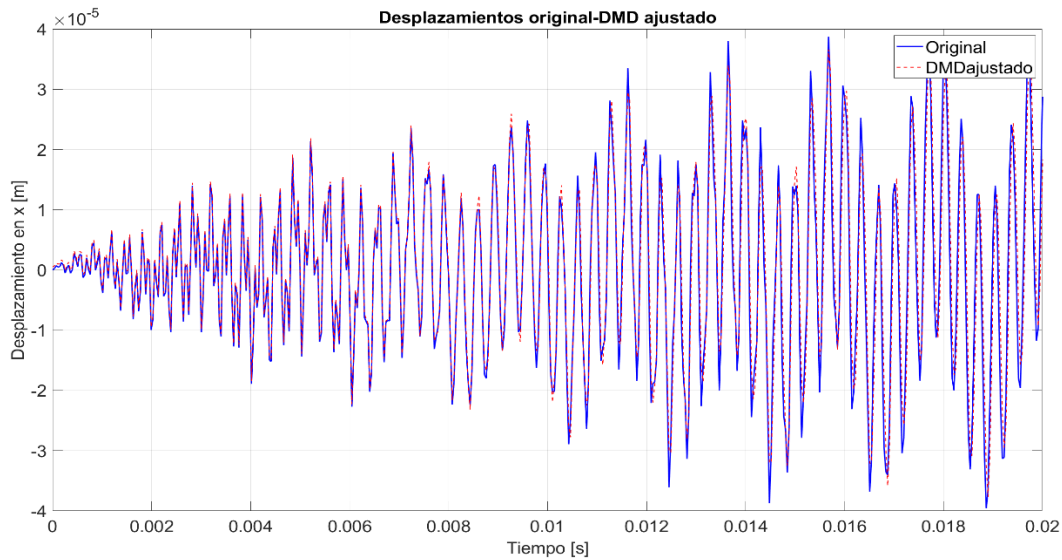


Figura 43. Acercamiento de la figura 42.



Los nodos 11, 158 y 234 donde su dinámica es reconstruida en las gráficas anteriores (gráfica azul información original, gráfica roja información DMD) muestran claramente un comportamiento diferente. El desplazamiento del nodo 11 es en x ubicado en el mango del diapasón, el cual presenta un comportamiento no lineal (sección 4.4) con una totalidad de 1,800 instantáneas analizadas, presenta un comportamiento con algunos desfases, pero no mayores al 5 % siendo aceptable para la construcción del modelo matemático. Los nodos 158 y 234 de desplazamientos en el eje y , se encuentran en la base de las puntas del diapasón con 600 y 6,000 instantáneas usadas para su análisis teniendo el nodo 158 un ajuste totalmente perfecto, mientras que el nodo 234 con algunos desfases en su reconstrucción, los cuales pueden ser causados por ruido en la información.

6.3.4.2 Validación. En las siguientes gráficas se muestra gráficamente la FFT de la información original (gráfica azul) y la calculada por el algoritmo DMD (gráfica roja).

Figura 44. FFT datos originales y datos DMD nodo 11.

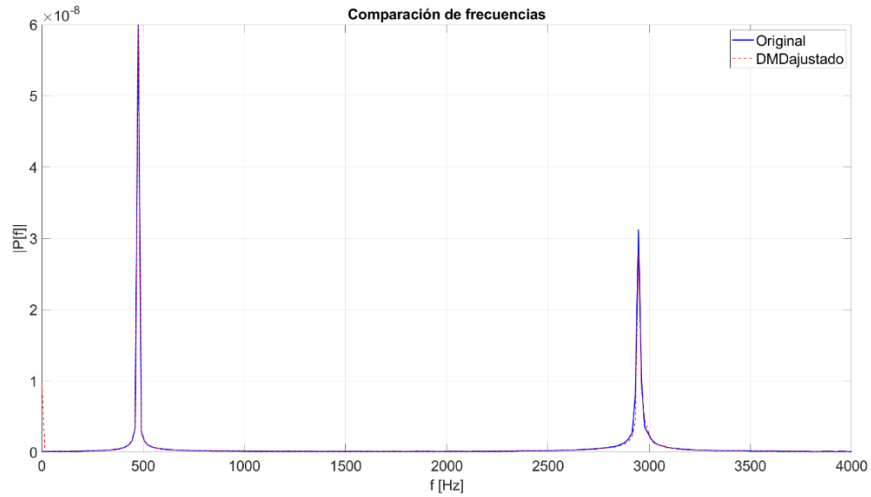
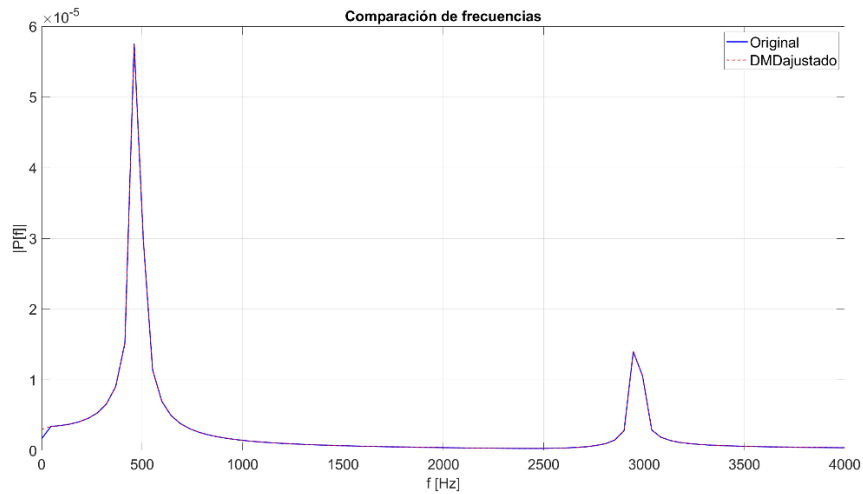
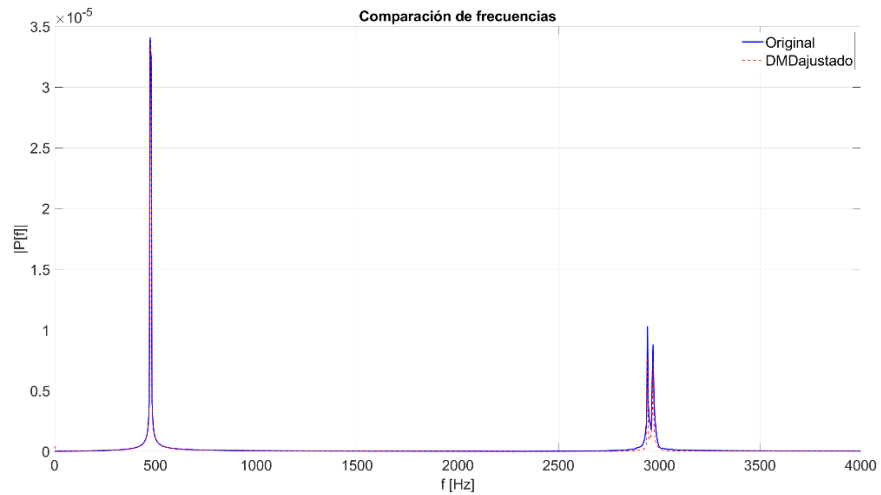


Figura 45. FFT datos originales y datos DMD nodo 158.



De la misma manera, que en la sección 6.2.3.2 se presentan dos graficas con dos picos correspondientes a la frecuencia de vibración del diapason, la cual está muy cerca de la frecuencia fundamental, sin embargo, aparece una pequeña desviación de la frecuencia por el hecho del cálculo a partir de un análisis modal sin restricciones, mientras que en el análisis transitorio se tienen restricciones.

Figura 46. FFT datos originales y datos DMD nodo 234.



Cabe resaltar que como fue mostrado en el cuadro 2, se realizaron estudios para varios nodos, con el fin de observar el comportamiento del algoritmo DMD para la reconstrucción de la dinámica del diapasón acústico, según el número de ciclos, número de instantáneas, Δt y su ubicación geométrica, con el propósito de posteriormente reconstruir los modos para cada instantánea del diapasón completo.

6.4 DMD diapasón completo

Obtenida la matriz de desplazamientos (y) para todos los nodos, se observa que su dimensión es de $2,077 \times 600$, es decir, sin las 48 filas de vectores nulos presente en la data original. Antes de proceder a realizar la reconstrucción 3D del comportamiento del diapasón, se hace necesario añadir estas 48 filas a la matriz solución, para que el número correspondiente a cada nodo sea coincidente con el de la data original.

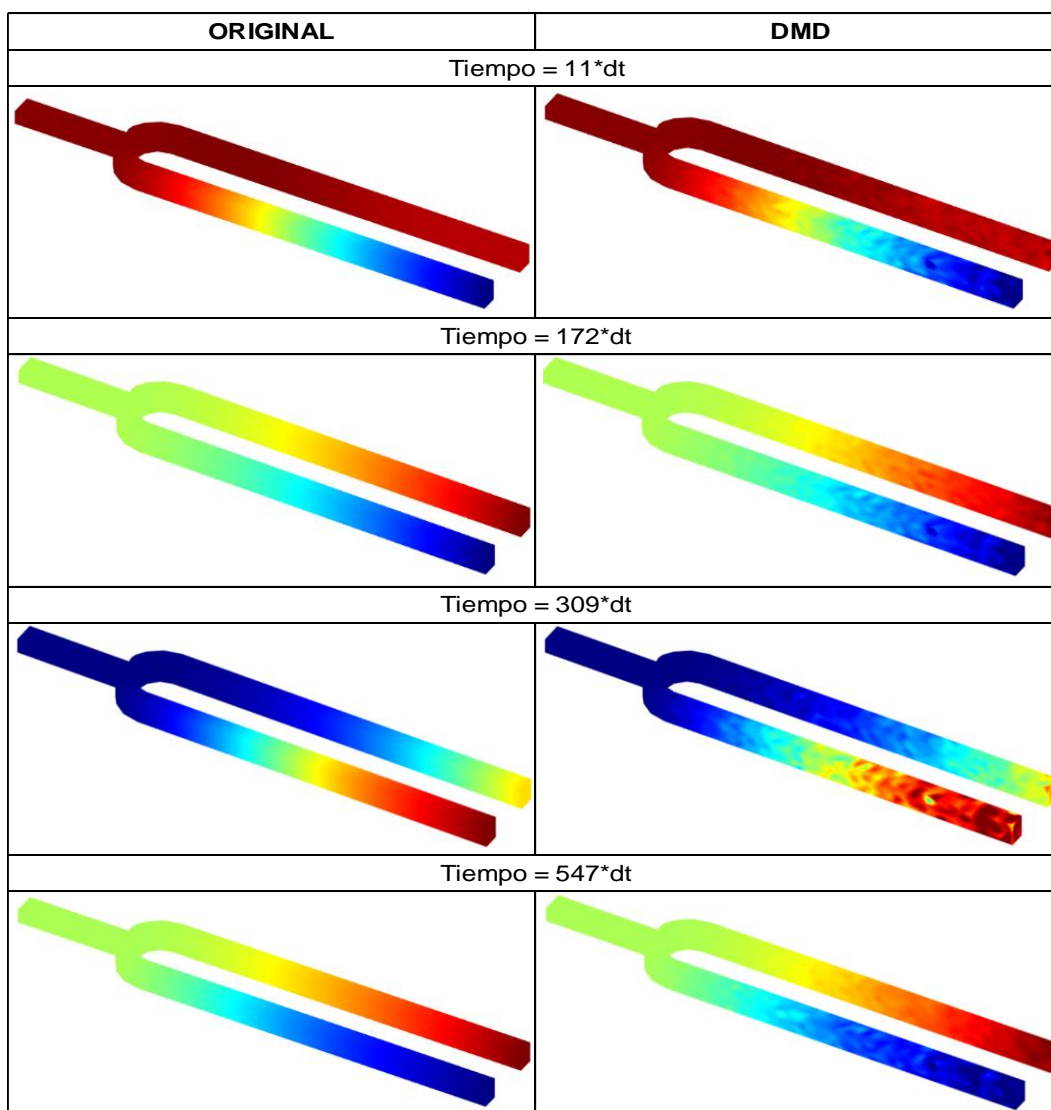
Estos desplazamientos hallados, junto con los vectores de desplazamiento 0, corresponden a los datos de entrada para el algoritmo *Instantaneas_diapason_completo* que permite generar la figura 3D del diapasón, cuyos resultados se validan en primer lugar con la figura construida tomando datos originales de la simulación.

Se deben cargar los datos de desplazamiento y para 600 instantáneas, obtenidos en el análisis transitorio y los datos del modelo PDE.

El algoritmo crea una carpeta, en la cual se almacenan como imágenes de formato *.png*, la representación gráfica del diapasón con datos originales y con datos DMD, para cada instantánea.

Parte de los resultados obtenidos, se ilustran a continuación:

Figura 47. Representación 3D del diapasón construido con datos originales y con datos DMD.



6.4.1 Frecuencia de vibración del diapasón. No basta sólo con validar la implementación del algoritmo por medio de la representación 3D, sino que se hace necesario verificar que este tiene la capacidad de determinar la frecuencia fundamental en cada nodo, cuyo valor será cercano, dado que el algoritmo es una aproximación.

La solución a las ecuaciones de movimiento para un sistema que describe un comportamiento vibratorio está dada en función de los valores propios λ y vectores propios W_r del sistema.

Para que el algoritmo pueda reconstruir la dinámica del sistema, es fundamental encontrar los valores propios adecuados, pues para sistemas dinámicos que vibran, estos datos no solo representan sus frecuencias y modos de vibrar, sino también la estabilidad del sistema.

Estos valores se muestran de manera matricial y donde cada uno corresponde a la solución de la ecuación exponencial del valor propio.

$$\lambda = e^{Omega*t} \quad (Ec\ 26)$$

Los valores propios obtenidos se encuentran en formato de números complejos, por lo cual, estos valores estarán dados en parejas conjugadas, es decir, cada valor propio de la forma $\lambda = \alpha + \beta i$, tendrá su pareja dada por $\lambda = \alpha - \beta i$.

La cantidad de datos presentes en la matriz *Omega* varían para cada nodo según el cálculo del mínimo error en la DMD.

De la Ec 31 se obtiene la ecuación usada en el algoritmo DMD y que permite determinar las frecuencias circulares *Omega* para cada pequeño incremento *dt* en el tiempo (Ec 24), las cuales al igual que λ están dadas por números complejos.

Los valores de frecuencia circular para cada uno de los 2,077 nodos se obtienen haciendo uso de la matriz diagonal *Omega* que está formada por el resultado de la

Ec 19, la cual se aplica a cada valor de la diagonal presente en la matriz de valores propios.

Teniendo en cuenta que:

$$\ln(\lambda) = \ln|\lambda| + \arg(\lambda) \mathbf{i}$$

$$\ln(\lambda) = \ln\left(\sqrt{\alpha^2 + \beta^2}\right) + \arctg\left(\frac{\beta}{\alpha}\right) \mathbf{i} \quad (\text{Ec 27})$$

Para $|\lambda| > 0$ y \arctg en radianes, en cada valor Ω expresado también en formato complejo, su parte real depende de la magnitud del valor λ y su componente imaginaria del ángulo formado en el plano complejo.

6.4.1.1 Estabilidad del sistema. Para seleccionar entre un conjunto de datos ubicados en la matriz Ω el valor correcto de la frecuencia fundamental, es necesario conocer para qué valor λ el sistema cuenta con una solución dinámicamente estable.

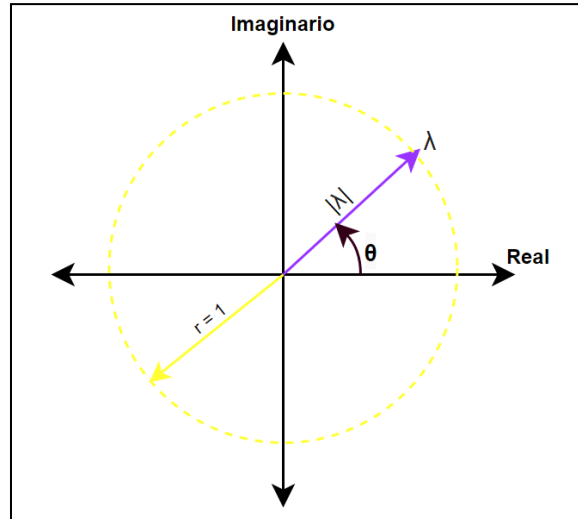
La forma de determinar la estabilidad de un sistema dinámico en tiempo discreto difiere de la usada en sistemas de tiempo continuo, puesto que el criterio de parte real negativa o positiva de un valor propio, no es válido.

Para sistemas en tiempo discreto, es necesario ubicar los valores propios en el plano complejo, y cuya magnitud se encuentra limitada por el círculo unitario, debido a que para $|\lambda| > 1$, el sistema matemáticamente se va hacia el infinito, generando una inestabilidad al considerar el sistema para n tiempos futuros.

$$\lambda^n = |\lambda|^n * e^{\mathbf{i}n\theta} \quad (\text{Ec 28})$$

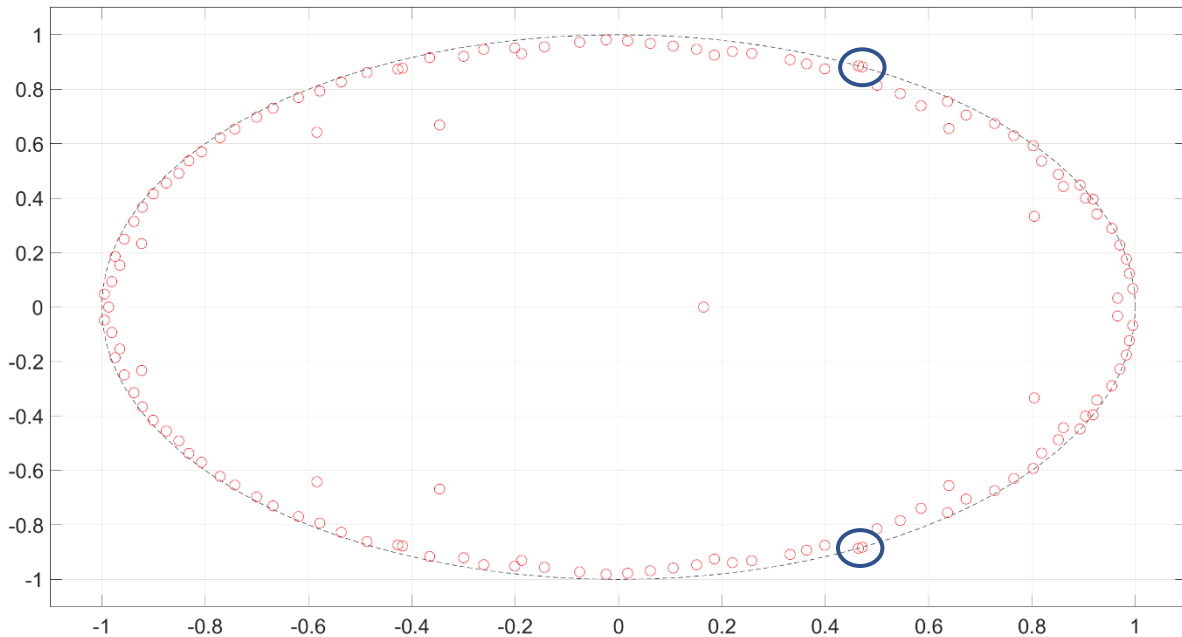
Por tal razón, 1 es la máxima magnitud que puede tomar un valor propio para que el sistema no sea inestable, por lo tanto, para valores propios dentro de un círculo unitario, la Ec 33 decaerá hasta 0, característica de un sistema estable.

Figura 48. Valor propio en el plano complejo con magnitud limitada por una circunferencia unitaria.



Para el caso del algoritmo DMD, la solución dinámicamente estable del sistema se produce para aquellos valores propios cuya magnitud les permite situarse exactamente sobre el perímetro de la circunferencia o que su valor sea prácticamente 1. Los demás valores propios ubicados dentro la circunferencia unitaria se consideran ruido, es decir, no son datos relevantes. Esto es fácil de observar si se crea una circunferencia unitaria en la que se grafiquen todos los valores propios encontrados para un nodo, en donde se podrá observar que los datos del semicírculo superior son los conjugados de los del semicírculo inferior, así:

Figura 49. Valores propios para el nodo 1,450 ubicados sobre una circunferencia unitaria.

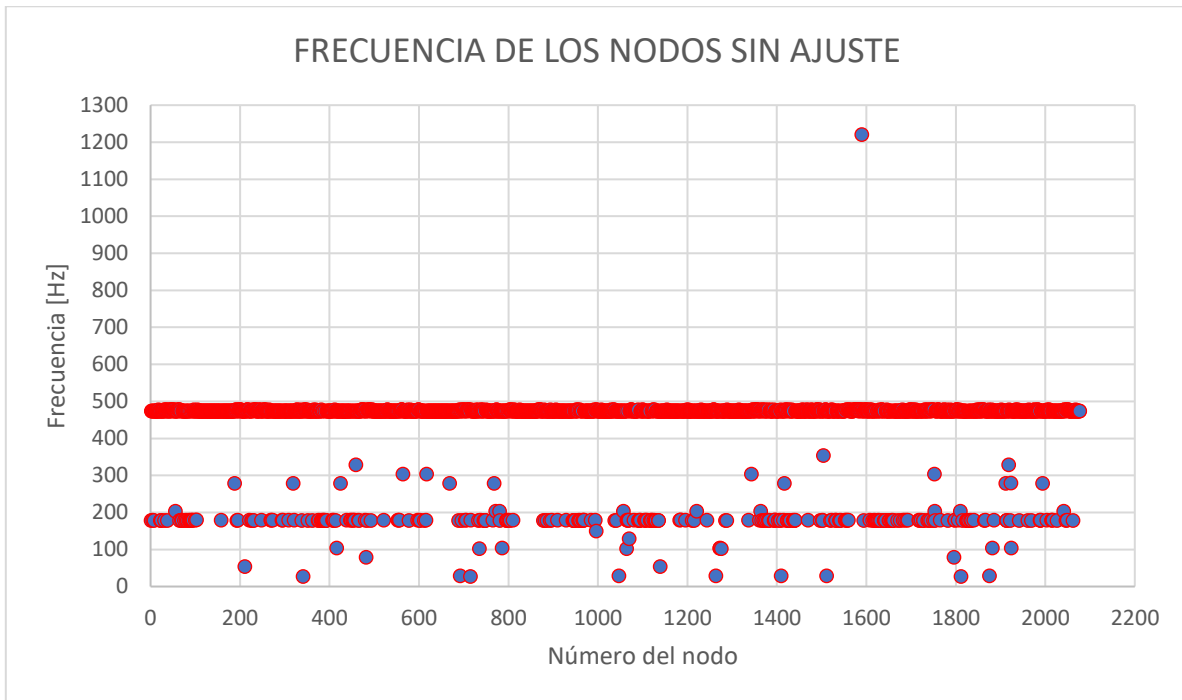


El diapasón será dinámicamente estable cuando la frecuencia fundamental corresponda al valor propio de magnitud 1.

Con base en lo anterior y teniendo en cuenta que $\ln(1) = 0$ (para un valor propio ubicado exactamente en el perímetro), la correcta elección en la matriz *Omega* del valor que contiene a la frecuencia fundamental, será aquel que cuente con su parte real igual o aproximadamente 0, es decir, el valor de parte real más pequeña en toda la matriz.

Con el criterio anterior, el valor elegido contiene en su parte imaginaria a la frecuencia fundamental en *rad/s*, para luego aplicar la función *calculo_frecuencias* para cada nodo, la cual convierte estos valores en frecuencias dadas en *Hz* y los organiza en un vector.

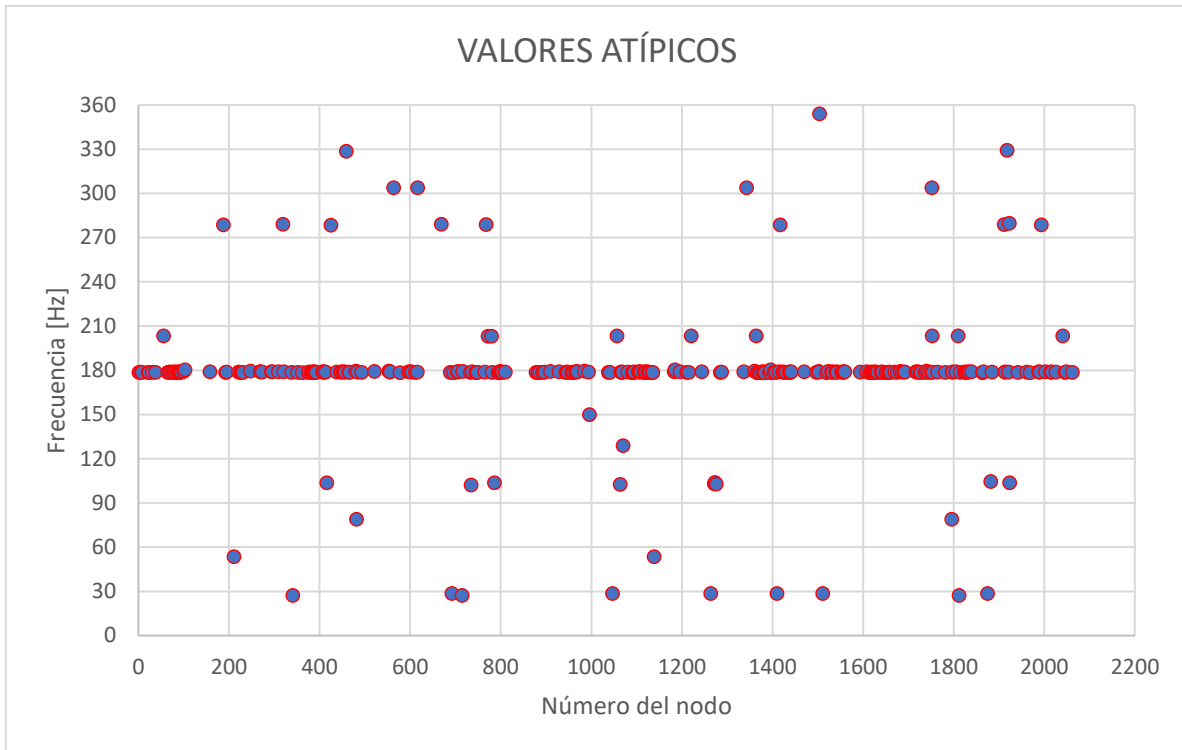
Figura 50. Dispersión de la frecuencia para cada nodo elegida con el criterio de círculo unitario con mínimo y máximo de Hankel de 10 a 200 respectivamente.



Se observa que existe una gran cantidad de valores dispersos de la frecuencia fundamental aproximada.

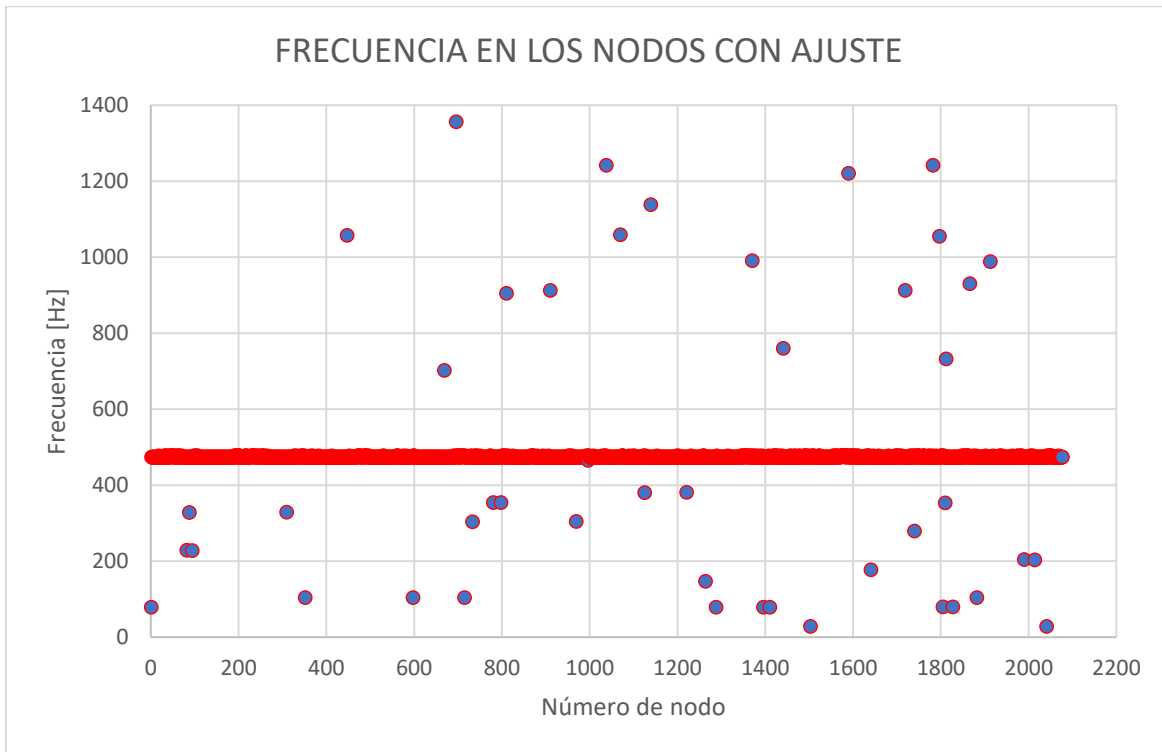
Por tal razón y para mejorar estos resultados, se usa la función *Valor_atipico* que detecta estos outliers filtrando por medio de un rango de percentiles.

Figura 51. Valores atípicos



Debido a que el ajuste depende del mínimo y máximo fijado para la matriz de Hankel, se intenta corregir este desajuste aumentando el rango de operación de la matriz, asignando un mínimo de 200 y un máximo de 400 sólo para estos nodos atípicos, pues para los demás nodos el DMD logró reconstruir la dinámica.

Figura 52. Nueva dispersión de la frecuencia para cada nodo elegida con el criterio de círculo unitario con mínimo y máximo de Hankel de 200 a 400 respectivamente.



Como se observa en la figura anterior, se logra una reducción significativa de la cantidad de valores dispersos, obteniendo que sólo un poco más del 2% de la totalidad de los nodos se mantienen en esta condición.

Para aquellos nodos que aún persisten como atípicos, es necesario aumentar el rango mínimo y máximo de Hankel, sin embargo, debido al poco porcentaje de nodos atípicos que resulta del ajuste, se descarta esta operación, puesto que el DMD ya ha logrado reconstruir la dinámica de cerca del 98% de los nodos del diapasón.

7. CONCLUSIONES

Del planteamiento, desarrollo y validación de este documento, se enlistan a continuación conclusiones, las cuales sintetizan los aspectos alcanzados:

- Se generó un modelo estructural del diapasón, por medio del módulo *Partial Differential Equation (PDE)* incluido en el *Toolbox* de Matlab v.R2019b, como acción inicial para los análisis modal y transitorio, a través del Método de Elementos Finitos (MEF).
- Se obtuvo que para un rango de frecuencias de -1 Hz a $4,000 \text{ Hz}$, el diapasón cuenta con doce modos de vibración, encontrando que el séptimo con valor aproximado de 460.42 [Hz] corresponde al primer modo flexible y por consiguiente a la frecuencia fundamental del diapasón para las características mecánicas y dimensionales dadas.
- Se obtuvieron grandes conjuntos de datos organizados matricialmente referentes a los desplazamientos en tres componentes, los cuales se tomaron en 6,000 instantes de tiempo y para 2,125 nodos, una vez excitado el diapasón en su modo fundamental.
- Se validó que el algoritmo DMD ajustado con error cuadrático medio, reconstruye la dinámica para nodos individuales en diferentes puntos de la geometría del diapasón. Esto, a través de la transformada rápida de Fourier (FFT), en la cual se observa que el valor de las dos frecuencias determinadas corresponde aproximadamente al valor del modo fundamental y modo clang, respectivamente.

- Se observó que para un valor mínimo de la matriz de Hankel de 10 y un máximo de 200, el algoritmo ajustado con error cuadrático medio es capaz de reconstruir el 85 % de la dinámica del sistema tridimensional, y que aumentando estos valores de 200 a 400 para los datos atípicos resultantes, el porcentaje de reconstrucción aumenta hasta un valor aproximado del 98 %.
- Se corroboró que el sistema reconstruido es dinámicamente estable para cada nodo, al evidenciar que ubicando sobre el plano complejo los valores propios obtenidos, existirán siempre mínimo dos datos que se sitúan de manera muy aproximada sobre el perímetro de la circunferencia unitaria y los demás datos justo al interior de ella.

BIBLIOGRAFÍA

AGAPITO, Rubén. Cálculo exacto de la matriz exponencial. Pro Mathematica. 2014 p 59-60.

BARATA, João, et al. The Moore–Penrose Pseudoinverse: A Tutorial Review of the Theory. Brazilian journal of physics. 2012, vol. 42, nro. 1. pp. 146-165

BRUNTON, Steven L. y KUTZ, J. Nathan. Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control. Cambridge: Cambridge University Press, 2019. ISBN 9781108422093.

COMSOL. Finding Answers to the Tuning Fork Mystery with Simulation. [Sitio web]. Fecha de publicación: 13 de abril del 2018. [Consulta: 18 de febrero del 2021]. Disponible en: <https://n9.cl/0rgu3>

DAS, Suddhasattwa y GIANNAKIS, Dimitrios. Delay-Coordinate Maps and the Spectra of Koopman Operators. Journal of statistical physics. 2019, vol. 175, nro. 6. p. 1111-1114.

DUARTE V., Oscar G. Análisis de sistemas dinámicos lineales. [En línea]. Bogotá: 2006. Disponible en: <https://n9.cl/ltu4p>

KIM, Young, et al. Acoustic damping optimization of the injector based on dynamic mode decomposition in a subscale combustor. Journal of Mechanical Science and Technology. 2017, vol. 31, nro. 9. p. 4511-4520

KOLMAN, Bernard y HILL, David R.. Algebra lineal. 8 ed. México, D.F.: Pearson Educación, 2006. ISBN 970-26-0696-9.

KUTZ, J. N., et al. Dynamic Mode Decomposition. Society for Industrial and Applied Mathematics, 2016. 241 p.

LEHMANN. Erich y CASELLA. George. Theory of point estimation. 2nd ed. New York: Springer Science & Business Media, 1998. p 48-56. ISBN 0-387-98502-6

LOAIZA RAMIREZ Manuel. Diseño y simulación de un criptosistema caótico para comunicaciones seguras. [En línea]. México: Departamento de Computación, Electrónica, Física e Innovación, Escuela de Ingeniería y Ciencias, 2006. Disponible en: <https://n9.cl/0u492>

MICHEL, Anthony N.; HOU, Ling y LIU, Derong. Stability of Dynamical Systems: On the Role of Monotonic and Non-Monotonic Lyapunov Functions. 2nd ed. Birkhäuser, Cham, 2015. 653 p.

MORENO GARCÍA, Francisco Ernesto y BECERRA VARGAS, José Armando. Estudio y Análisis de sistemas dinámicos. 1st ed. Cúcuta (Colombia): Universidad Francisco de Paula Santander, 2012. 195 p.

OPPENHEIM. Alan; WILLSKY. Alan y NAWAB. Hamid. Señales y sistemas. 2nd ed. Naucalpan de Juárez: Pearson Educación, 1998. p 358-380. ISBN 917-17-0116-X

SCHMID, PETER J. Dynamic mode decomposition of numerical and experimental data. Journal of fluid mechanics. 2010, vol. 656, nro. August. p. 5-28

STANLEY, Grossman S y FLORES, José. Algebra lineal. 7 ed. México, D.F.: McGraw-Hill, 2012. ISBN 978-607-15-0760-0.

STRANG, Gilbert. Linear algebra and its applications. 3 ed. United States of America: Thomson Learning, 1988. ISBN 0-15-551005-3.

ZDYBAL, kamila. POD and DMD Decomposition of numerical and experiemntal data. [En línea]. Von Karman Institute for Fluid Dynamics in Belgium, 2016. Disponible en: <https://github.com/kamilazdybal/POD-DMD-decompositions.git>

WATANABE, Junzo. Hankel matrices and Hankel ideals. ResearchGate. 1995, vol. 32, p. 11-21

ANEXOS

Anexo A. Funciones complementarias para los algoritmos DMD.

- Función DMD estándar con un valor de entrada n_{ajuste} para nodos individuales.

```
function [X_DMD] = FDMD_individual(n_ajuste,n,nodo,dt,tiempo)
%FDMD_INDIVIDUAL corresponde al algoritmo DMD estándar creado para ejecutarse en
UN SOLO NODO,
%el cual es alimentado por datos organizados a partir de la matriz de
%Hankel, tomando un valor n_ajuste preestablecido.
%
%X_DMD] = FDMD_individual(n_ajuste,n,nodo,dt,tiempo)
%
%Detailed explanation goes here

X_DMD = zeros(n_ajuste,n);
X = zeros(n_ajuste,(n-n_ajuste));
X_desplazada = zeros(n_ajuste,(n-n_ajuste));

for ii = 1:n_ajuste
    for jj = 1:(n-n_ajuste)
        X(ii,jj) = nodo(1,jj+ii-1);
        X_desplazada(ii,jj) = nodo(1,jj+ii);
    end
end
[U,S,V]= svd(X, 'econ');
r = rank(S);
S_r = S(1:r,1:r); % Valores singulares truncados en el rank de A
U_r = U(:,1:r); % Vectores a la izquierda truncados
V_r = V(:,1:r); % Vectores a la derecha truncados
A_reducida = U_r'*X_desplazada*V_r*pinv(S_r);
[W_r,Lambda]= eig(A_reducida);
Omega = diag(log(diag(Lambda)))/dt;
Phi = X_desplazada*V_r*pinv(S_r)*W_r;
alpha = S_r*V_r(1,:)' ;
b = (W_r*Lambda)\alpha;

for tt=1:n
    X_DMD(:,tt)= Phi*exp(Omega*tiempo(tt))*b;
end
end
```

- Función DMD para calcular una nueva matriz de Hankel, partiendo de la ubicación del mínimo error encontrado para nodos individuales.

```
Function [X_DMDajustada,Omega]=FDMDajustada_individual(n_ajuste,n,nodo,dt,tiempo)
%FDMDajustada corresponde al algoritmo DMD estándar ejecutado para UN SOLO
%NODO, con la variante de que ahora toma como valor de entrada un n_ajuste
%nuevo, calculado de acuerdo al error encontrado anteriormente.
%
```

```

[X_DMDajustada, Omega] = FDMDajustada_individual(n_ajuste, n, nodo, dt, tiempo)
%
%Las salidas de esta función corresponden a la matriz X_DMDajustada
%(reconstrucción del DMD con error) y a la matriz diagonal Omega, a
%partir de la cual se calcularán las frecuencias fundamentales
%implicitas en la reconstrucción

X = zeros(n_ajuste, (n-n_ajuste));
X_DMDajustada = zeros(n_ajuste, n);
X_desplazada = zeros(n_ajuste, (n-n_ajuste));

for iii=1: n_ajuste
    for jjj=1: n-n_ajuste
        X(iii, jjj)=nodo(1, jjj+iii-1);
        X_desplazada(iii, jjj)=nodo(1, jjj+iii);
    end
end

[U, S, V]= svd(X, 'econ');
r = rank(S);
S_r = S(1:r, 1:r); % Valores singulares truncados en el rank de A
U_r = U(:, 1:r); % Vectores a la izquierda truncados
V_r = V(:, 1:r); % Vectores a la derecha truncados
A_reducida = U_r'*X_desplazada*V_r*pinv(S_r);
[W_r, Lambda]= eig(A_reducida);
Omega = diag(log(diag(Lambda)))/dt;
Phi = X_desplazada*V_r*pinv(S_r)*W_r;
alpha = S_r*V_r(1, :)';
b = (W_r*Lambda)\alpha;

for ttt=1:n
    X_DMDajustada(:, ttt) = Phi*exp(Omega*tiempo(ttt))*b;
end
end

```

- Función DMD estándar con un valor de entrada n_{ajuste} dependiente del valor mínimo y máximo de la matriz Hankel para todo el diapasón.

```

function [X_DMD] = FDMD(n_ajuste, n, nodo, uy_intervalos, dt, tiempo)
%FDMD corresponde al algoritmo DMD estándar, el cual es alimentado por
%datos organizados a partir de la matriz de Hankel, tomando como rango de
%organización un mínimo y máximo pre-establecido
%
[X_DMD] = FDMD(n_ajuste, n, nodo, uy_intervalos, dt, tiempo)
%
%Detailed explanation goes here

X_DMD = zeros(n_ajuste, n);
X = zeros(n_ajuste, (n-n_ajuste));
X_desplazada = zeros(n_ajuste, (n-n_ajuste));

for ii = 1:n_ajuste
    for jj = 1:(n-n_ajuste)
        X(ii, jj) = uy_intervalos(nodo, jj+ii-1);
        X_desplazada(ii, jj) = uy_intervalos(nodo, jj+ii);
    end
end

```

```

end
[U,S,V]= svd(X,'econ');
r = rank(S);
S_r = S(1:r,1:r); % Valores singulares truncados en el rank de A
U_r = U(:,1:r); % Vectores a la izquierda truncados
V_r = V(:,1:r); % Vectores a la derecha truncados
A_reducida = U_r'*X_desplazada*V_r*pinv(S_r);
[W_r,Lambda] = eig(A_reducida);
Omega = diag(log(diag(Lambda)))/dt;
Phi = X_desplazada*V_r*pinv(S_r)*W_r;
alpha = S_r*V_r(1,:)' ;
b = (W_r*Lambda)\alpha;

for tt=1:n
    X_DMD(:,tt)= Phi*exp(Omega*tiempo(tt))*b;
end
end

```

- Función DMD para calcular una nueva matriz de Hankel, partiendo de la ubicación del mínimo error encontrado para el diapasón completo.

```

function
[X_DMDajustada,Omega]=FDMDajustada(n_ajuste,n,nodo,uy_intervalos,dt,tiempo)
%FDMDajustada corresponde al algoritmo DMD estándar, con la variante de que
%ahora toma como valor de entrada un n_ajuste nuevo, calculado de acuerdo
%al error encontrado anteriormente.
%
%[X_DMDajustada,Omega] = FDMDajustada(n_ajuste,n,nodo,uy_intervalos,dt,tiempo)
%
%Las salidas de esta función corresponden a la matriz X_DMDajustada
%(reconstrucción del DMD con error) y a la matriz diagonal Omega, a
%partir de la cual se calcularán las frecuencias fundamentales
%implícitas en la reconstrucción

X = zeros(n_ajuste, (n-n_ajuste));
X_DMDajustada = zeros(n_ajuste,n);
X_desplazada = zeros(n_ajuste, (n-n_ajuste));

for iii=1: n_ajuste
    for jjj=1: n-n_ajuste
        X(iii,jjj)=uy_intervalos(nodo,jjj+iii-1);
        X_desplazada(iii,jjj)=uy_intervalos(nodo,jjj+iii);
    end
end

[U,S,V]= svd(X,'econ');
r = rank(S);
S_r = S(1:r,1:r); % Valores singulares truncados en el rank de A
U_r = U(:,1:r); % Vectores a la izquierda truncados
V_r = V(:,1:r); % Vectores a la derecha truncados
A_reducida = U_r'*X_desplazada*V_r*pinv(S_r);
[W_r,Lambda]= eig(A_reducida);
Omega = diag(log(diag(Lambda)))/dt;
Phi = X_desplazada*V_r*pinv(S_r)*W_r;
alpha = S_r*V_r(1,:)' ;
b = (W_r*Lambda)\alpha;

```

```

for ttt=1:n
    X_DMDajustada(:,ttt)= Phi*exp(Omega*tiempo(ttt))*b;
end
end

```

- Función para calcular la transformada rápida de Fourier (FFT).

```

function [f,P_1] = Diapason_FFT(Desplazamientos,muestreo)
L = numel(Desplazamientos);
L = L + mod(L,2);
Y = fft(Desplazamientos);
P_2 = abs(Y/L);
P_1 = P_2(1:L/2+1);
P_1(2:end-1) = 2*P_1(2:end-1);
f = muestreo*(0:(L/2))/L;
end

```

- Función para eliminar las filas de ceros de la matriz con la totalidad los nodos. Esta función también crea una nueva matriz con el número de instantáneas que desee el usuario.

```

function [Data_original , uy_intervalos] =
eliminar_ceros(uy_nodos_totales,instantaneas)
%ELIMINAR_CEROS es una función que permite eliminar las filas dadas por
%vectores nulos producto de las condiciones de frontera del análisis
%transitorio que fija dos superficies del diapasón, por ende, los nodos
%presentes en estas carecen de movimiento alguno
%
%[Data_original , uy_intervalos] = eliminar_ceros(uy_nodos_totales , instantaneas)
%
%Los ceros se eliminan para evitar inconsistencias en el cálculo del error.

%% La cantidad máxima de datos disponibles es para 6000 instantes
instantaneas_max = 6000;

%% Organización de datos previa a la eliminación de ceros, con el propósito de
obtener la matriz final de trabajo
m = size(uy_nodos_totales,1);

for cont = 1:instantaneas
    uy_intervalos = zeros(m,cont);
end

for cont = 1:instantaneas
    columna = cont*(instantaneas_max/instantaneas);
    uy_intervalos(:,cont) = uy_nodos_totales(:,columna);
end

%% Eliminación de filas nulas
Data_original = uy_intervalos;

```

```

eliminar_ceros = sum(abs(uy_intervalos),1)==0;
uy_intervalos(sum(abs(uy_intervalos),2)==0,:)=[];
uy_intervalos(:,eliminar_ceros) = [];

```

```
end
```

- Función para calcular la frecuencia fundamental del diapasón.

```

function [Frec_abs,Frec_neg,Frec_1er_fila] = calculo_frecuencias(Omega)
%CALCULO_FRECUENCIAS Determina la frecuencia fundamental en Hz presente en
%el vector Omega obtenido por el DMD ajustado
%
%      [Frecuencias_abs,      Frecuencias_neg,      Frecuencias_1er_fila]      =
calculo_frecuencias(Omega, nodo)
%
%La función toma el valor imaginario de un dato del vector Omega,
%condicionado a tres condiciones de análisis. Se toma el valor
%imaginario del dato, como una frecuencia circular [rad/s]

%% Diagonal de la matriz diagonal Omega, obtenida por el DMD ajustado
Omega_vector = diag(Omega); %Vector con elementos complejos

%% Cálculo de la frecuencia tomando el valor más cercano a cero, incluyendo
negativos y positivos
Omega_minimo_abs = min(abs(real(Omega_vector))); %Toma el valor real más
pequeño del vector Omega
Omega_fila_abs = (abs(real(Omega_vector))==Omega_minimo_abs); %Vector lógico.
Muestra datos lógicos (1 y 0), mostrando el valor 1 para los datos que cumplen la
igualdad
Omega_valor_abs = Omega_vector(Omega_fila_abs); %Muestra los valores completos
(real + complejo) del vector Omega que cumplen la condición (1 lógico)
%Crea un vector con todas las frecuencias determinadas para cada nodo
Frec_abs = imag(Omega_valor_abs/(2*pi))'; %De los datos que cumplen la
condición, calcula la frecuencia en Hz únicamente con su parte imaginaria

%% Cálculo de la frecuencia tomando sólo el valor negativo más cercano a cero
Omega_neg = (real(Omega_vector)<0); %Vector lógico, que muestre que datos reales
son menores que 0
Omega_neg_valor = real(Omega_vector(Omega_neg)); %Muestra el valor referente
al 1 lógico, es decir, datos menores que 0
Omega_minimo_neg = min(abs(real(Omega_neg_valor))); %Se halla el valor más
cercano a cero haciendo uso del valor absoluto
Omega_fila = (abs(real(Omega_vector))==Omega_minimo_neg); %Vector lógico que
devuelve 1 en la fila del vector principal que cumple la igualdad
Omega_valor_neg = Omega_vector(Omega_fila);
Frec_neg = imag(Omega_valor_neg/(2*pi))';

%% Cálculo de la frecuencia sólo para la primer fila de datos
Frec_1er_fila = imag(Omega_vector(1)/(2*pi));

end

```

- Función para encontrar los valores atípicos de las frecuencias calculadas.

```

function [nodo_atipico_abs,nodo_atipico_neg,nodo_atipico_1er_fila] =
Valor_atipico(All_frec_abs,All_frec_neg,All_frec_1er_fila)

```

```

%VALOR_ATIPICO debido a que con el valor minimo o maximo establecido para
%la matriz de Hankel no siempre el algoritmo encuentra la frecuencia
%fundamental ubicada en alguna de las tres condiciones (especialmente para
%las frecuencias obtenidas para el minimo valor de datos reales positivos y
%negativos), se generan valores atipicos o alejados del patron de la
%frecuencia fundamental.
%
%[nodo_atipico_abs,nodo_atipico_neg,nodo_atipico_1er_fila]           =
Valor_atipico(All_frec_abs,All_frec_neg,All_frec_1er_fila)
%
%El algoritmo encuentra el nodo que para tales condiciones posee valores
%atipicos, con base en lo alejado que este se encuentre del patrón de
%datos, haciendo comparación por desviación estándar de la mediana o por un
%rango de valores percentiles.

v_logico_abs = isoutlier(All_frec_abs(:,1),'percentiles',[15, 100]);
nodo_atipico_abs = find(v_logico_abs);

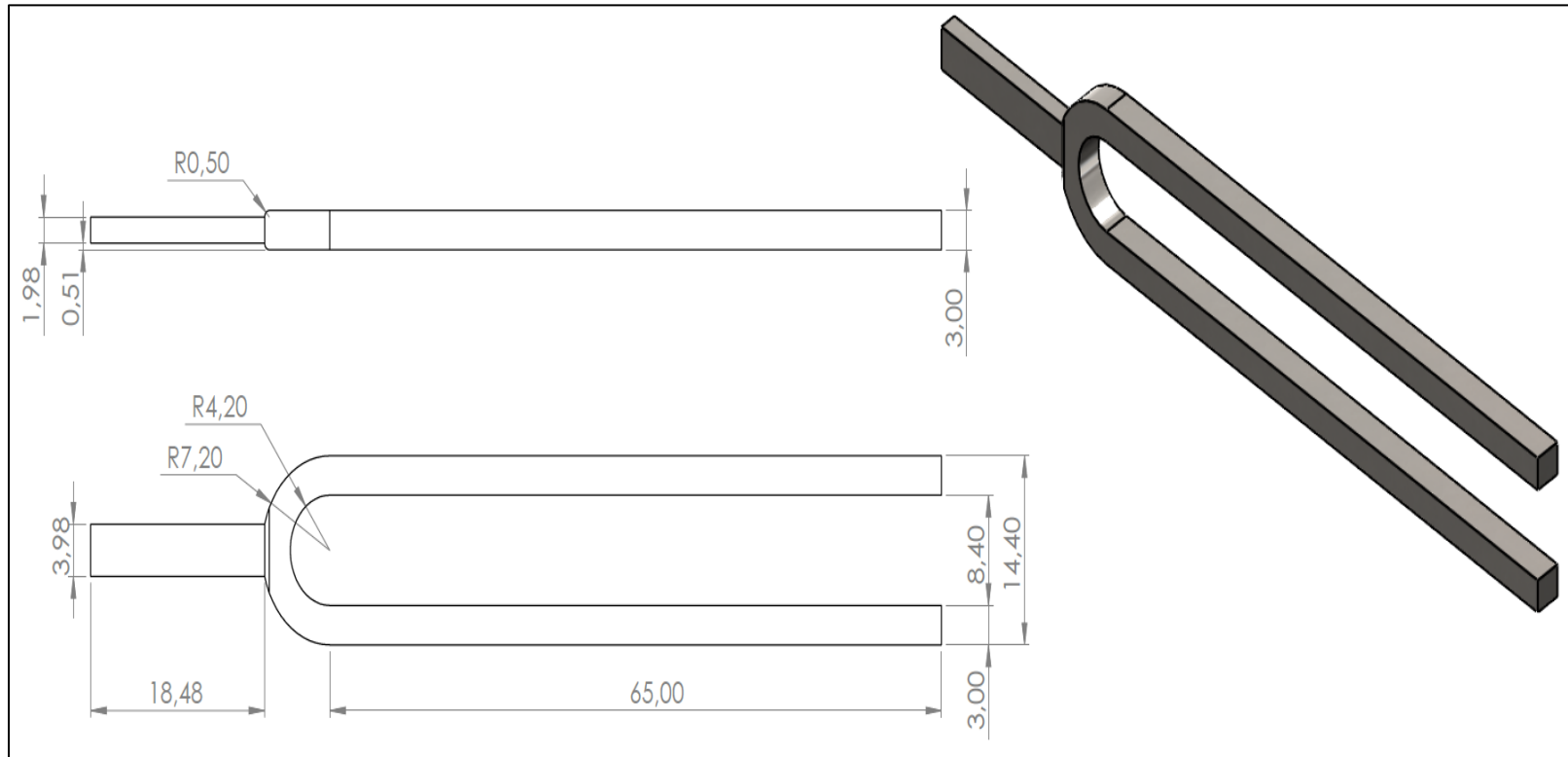
v_logico_neg = isoutlier(All_frec_neg(:,1));
nodo_atipico_neg = find(v_logico_neg);

v_logico_1era_fila = isoutlier(All_frec_1er_fila);
nodo_atipico_1er_fila = find(v_logico_1era_fila);

end

```

Anexo B. Detalle dimensional del diapasón.



Anexo C. Algoritmo para el análisis modal del diapasón.

```
clear all; close all; clc;

% ANÁLISIS MODAL

modelo_modal = createpde('structural','modal-solid');
importGeometry(modelo_modal,'Diapason_acustico.stl');

figure
pdegplot(modelo_modal)

% Propiedades mecánicas de un diapasón construido en acero.
% Datos en sistema internacional
E = 210E9; % Modulo de elasticidad
nu = 0.3; % Coeficiente de Poisson
rho = 8000; % Densidad

structuralProperties(modelo_modal,'YoungsModulus',E,'PoissonsRatio',nu,'MassDensity',rho);

%Generación del mallado para el análisis modal

figure
mallado_modal = generateMesh(modelo_modal,'Hmax',0.001);
pdemesh(mallado_modal, 'NodeLabels', "off")

solucion_modal = solve(modelo_modal,'FrequencyRange',[-1,4000]*2*pi);
num_modos_frecuencia = 1:numel(solucion_modal.NaturalFrequencies);
tabla_modos = table(num_modos_frecuencia.',solucion_modal.NaturalFrequencies/2/pi);
%frecuencias en Hz
tabla_modos.Properties.VariableNames = {'No. modos','Frecuencia'};
disp(tabla_modos);

modos = animarmodosdiapason(solucion_modal);
% movie(figure('units','normalized','position',[0 0 1 1]),modos,5,30)
```

Anexo D. Algoritmo para el análisis transitorio del diapasón.

```
clear all; close all; clc;

% ANÁLISIS TRANSITORIO

modelo_transitorio = createpde('structural','transient-solid');
importGeometry(modelo_transitorio,'Diapason_acustico.stl');

% Propiedades mecánicas
% Datos en sistema internacional
E = 210E9; % Modulo de elasticidad
nu = 0.3; % Coeficiente de Poisson
rho = 8000; % Densidad
structuralProperties(modelo_transitorio,'YoungsModulus',E,'PoissonsRatio',nu,'MassDensity',rho);

% Generación del mallado para el análisis transitorio
mallado_transitorio = generateMesh(modelo_transitorio,'Hmax',0.003);
pdemesh(mallado_transitorio,'NodeLabels','on','FaceAlpha',0.4);

figure('units','normalized','Position',[0 0 1 1])
pdeplot(modelo_transitorio,'FaceLabels','on','FaceAlpha',0.5)
view(-50,15)
title('Modelado con etiquetas en sus caras')

% Condiciones de frontera que asemeja el agarre del diapasón con una mano o
% su ubicación vertical sobre una superficie
structuralBC(modelo_transitorio,'Face',[21,22],'Constraint','fixed');

Frecuencia_fundamental = 2.892909520864917e+03;
Periodo = 2*pi/Frecuencia_fundamental; %Primer modo (frecuencia fundamental)

% Golpe en instante corto de tiempo para excitar el modo fundamental.
structuralBoundaryLoad(modelo_transitorio,'Face',11,'Pressure',5E6,'EndTime',Periodo/300);
%Condiciones iniciales restringiendo el movimiento en el tiempo = 0
structuralIC(modelo_transitorio,'Displacement',[0;0;0],'Velocity',[0;0;0]);

% PARA SOLUCION GENERADA POR 10 CICLOS
ciclos_10 = 10; %Número de veces que se repite el periodo
muestreo_datos = 60/Periodo; % Cantidad de datos que se mostrarán en un tiempo = 1s
tiempo = linspace(0,ciclos_10*Periodo,ciclos_10*Periodo*muestreo_datos);
sol_transitorio_10ciclos = solve(modelo_transitorio,tiempo)

%Para la cara 12 tomando una diagonal de nodos (3)

%Vector de los nodos ubicados en la cara 12
Nodos_Cara12 = findNodes(mallado_transitorio,'region','Face',12);

Matriz_cara12 = sol_transitorio_10ciclos.Displacement.uy(Nodos_Cara12(:,:));
uy_nodo24 = sol_transitorio_10ciclos.Displacement.uy(Nodos_Cara12(:,2),:);
uy_nodo25 = sol_transitorio_10ciclos.Displacement.uy(Nodos_Cara12(:,3),:);
```

```

uy_nodo317 = sol_transitorio_10ciclos.Displacement.uy(Nodos_Cara12(:,5),:);

%Para el centro de la punta 1 (donde se golpea el diapason)
uy_nodo144 = sol_transitorio_10ciclos.Displacement.uy(144,:);
uy_nodo166 = sol_transitorio_10ciclos.Displacement.uy(166,:);

%Para nodos ubicados cerca al inicio de la punta 1 desde su base
uy_nodo136 = sol_transitorio_10ciclos.Displacement.uy(136,:);
uy_nodo158 = sol_transitorio_10ciclos.Displacement.uy(158,:);

% PARA SOLUCION GENERADA POR 100 CICLOS
ciclos_100 = 100;
tiempo = linspace(0,ciclos_100*Periodo,ciclos_100*Periodo*muestreo_datos);
sol_transitorio_100ciclos = solve(modelo_transitorio,tiempo)

%Para todos los nodos del diapason
uy_nodos_totales = sol_transitorio_100ciclos.Displacement.uy;

%Para la cara 16 tomando una diagonal de nodos (3)

%Vector de los nodos ubicados en la cara 16
Nodos_Cara16 = findNodes(mallado_transitorio,'region','Face',16);

Matriz_car16 = sol_transitorio_100ciclos.Displacement.uy(Nodos_Cara16(:,,:));
uy_nodo32 = sol_transitorio_100ciclos.Displacement.uy(Nodos_Cara16(:,4),:);
uy_nodo369 = sol_transitorio_100ciclos.Displacement.uy(Nodos_Cara16(:,5),:);
uy_nodo2 = sol_transitorio_100ciclos.Displacement.uy(Nodos_Cara16(:,1),:);

%Para el centro de la punta 2
uy_nodo242 = sol_transitorio_100ciclos.Displacement.uy(242,:);
uy_nodo43 = sol_transitorio_100ciclos.Displacement.uy(43,:);

%Para nodos ubicados cerca al inicio de la punta 1 desde su base
uy_nodo234 = sol_transitorio_100ciclos.Displacement.uy(234,:);
uy_nodo36 = sol_transitorio_100ciclos.Displacement.uy(36,:);

% PARA SOLUCION GENERADA POR 30 CICLOS
ciclos_30 = 30;
tiempo = linspace(0,ciclos_30*Periodo,ciclos_30*Periodo*muestreo_datos);
sol_transitorio_30ciclos = solve(modelo_transitorio,tiempo)

%Para nodos ubicados en el mango del diapason y DESPLAZAMIENTO AXIAL
ux_nodo11 = sol_transitorio_30ciclos.Displacement.ux(11,:);
ux_nodo13 = sol_transitorio_30ciclos.Displacement.ux(13,:);

```

Anexo E. Algoritmo DMD con entrada de un valor n_{ajuste} .

```
clear all; close all; clc;

load('DATA_caral2.mat')

%Ajuste nodo 317
ajuste = 31;

%% Pre-asignaciones
nodo = uy_nodo317;

instantaneas_maximas = 6000;
instantaneas = 600;
tiempo = linspace(0,ciclos_10*Periodo,instantaneas);
dt = tiempo(1,2);
n = length(tiempo);

%%
n_ajuste = ajuste;
[X_DMD] = FDMD_individual(n_ajuste,n,nodo,dt,tiempo);
Desplazamiento_uy = real(X_DMD(1,:));

%%
figure
plot(tiempo,nodo,'b-','linewidth',1);
hold on;
plot(tiempo,Desplazamiento_uy,'g--','LineWidth',3);
grid on
title({'Desplazamientos original-DMD'})
legend('Original','DMD');
xlabel('Tiempo (s)'); ylabel('Desplazamiento en y (m)')

[f_original,P_original] = Diapason_FFT(nodo,muestreo_datos);
[f_DMD,P_DMD] = Diapason_FFT(Desplazamiento_uy,muestreo_datos);

figure
plot(f_original,P_original,'LineWidth',1.5);
hold on;
plot(f_DMD,P_DMD,'g--','LineWidth',3);
grid on
title({'Comparación de frecuencias'})
xlabel('f (Hz)')
ylabel('|P(f)|')
xlim([0,4000])
legend('Original','DMD');
```

Anexo F. Algoritmo DMD con error cuadrático medio.

```
clear all; close all; clc;

load('DATA_centropuntas.mat')

%% Ajuste nodo 242, 43
minimo = 10;
maximo = 100;

%% Pre-asignaciones
nodo = uy_nodo43;

instantaneas_maximas = 6000;
instantaneas = 6000;
tiempo = linspace(0,ciclos_100*Periodo,instantaneas);
dt = tiempo(1,2);
n = length(tiempo);

error = zeros(1,maximo-minimo+1);
cont = 0;

%%
for i = minimo:maximo

    cont = cont+1;

    n_ajuste = i;
    [X_DMD] = FDMD_individual(n_ajuste,n,nodo,dt,tiempo);
    Desplazamiento_uy = real(X_DMD(1,:));

    error(cont) = immse(Desplazamiento_uy,nodo);
    Encontrar_error = find(error==(min(error)));
    K = error(Encontrar_error);

end

%% Mejor ajuste

n_ajuste = (Encontrar_error+minimo-1);
[X_DMDajustada,Omega] = FDMDajustada_individual(n_ajuste,n,nodo,dt,tiempo);
Desplazamiento_uy_ajustado = real(X_DMDajustada(1,:));

%%
figure
bar(minimo:maximo,error)
title({'Error medio cuadrático'})
xlabel('Rango analizado'); ylabel('Error (m)')

figure
plot(tiempo,nodo,'b-','linewidth',1.5);hold on;
plot(tiempo,Desplazamiento_uy_ajustado,'r--','LineWidth',1);grid on
title({'Desplazamientos original-DMD ajustado'})
legend('Original','DMDajustado');
```

```

xlabel('Tiempo (s)'); ylabel('Desplazamiento en x (m)')

[foriginal,Poriginal] = Diapason_FFT(nodo,muestreo_datos);
[fajustada,Pajustada] = Diapason_FFT(Desplazamiento_uy_ajustado,muestreo_datos);

figure
plot(foriginal,Poriginal,'b-','LineWidth',1.5);hold on;
plot(fajustada,Pajustada,'r--','LineWidth',1);grid on
title({'Comparación de frecuencias'})
xlabel('f (Hz)')
ylabel('|P(f)|')
xlim([0,4000])
legend('Original','DMDejustado');

```

Anexo G. Algoritmo DMD diapasón completo.

```
clear all; close all; clc;

%% Se cargan los datos de esplazamiento y periodo para la simulación con 100
ciclos
load('DATA_ciclos100.mat','uy_nodos_totales','Periodo')

format long

%% Instantáneas para el cálculo que no excedan las máximas posibles (6000) y
organización de la data
instantaneas = 600;
[Data_original , uy_intervalos] = eliminar_ceros(uy_nodos_totales ,
instantaneas);

%% Mínimo y máximo establecido para la organización de datos usando matriz de
Hankel
minimo=10;
maximo=200;

%%
tiempo = linspace(0,100*Periodo,instantaneas);
dt = tiempo(1,2);
n = length(tiempo);

%%
mm = size(uy_intervalos,1);
error = zeros(1,maximo-minimo+1);
Desplazamientos_totales = zeros(mm,n);
All_frec_abs = zeros(mm,2);
All_frec_neg = zeros(mm,2);
All_frec_1er_fila = zeros(mm,1);

%%
for nodo = 1:mm

    cont = 0;

    for i = minimo:maximo
        cont = cont + 1;

        n_ajuste = i;
        [X_DMD] = FDMD(n_ajuste,n,nodo,uy_intervalos,dt,tiempo);
        Desplazamiento_uy = real(X_DMD(1,:));

    %%
        error(cont) = immse(Desplazamiento_uy,uy_intervalos(nodo,:));
        Encontrar_error = find(error==(min(error)));
    end

    %% Best Fitting

    n_ajuste = (Encontrar_error+minimo-1);
```

```

[X_DMDajustada, Omega, W_r, Lambda] =
FDMDajustada(n_ajuste, n, nodo, uy_intervalos, dt, tiempo);

%Primera fila de la reconstrucción DMD una vez aplicado el error
Desplazamiento_uy_ajustado = real(X_DMDajustada(1, :));

% Matriz con todos los nodos del diapason (sin las filas de ceros)...
% ...obtenida con la primer fila de cada DMD ajustado con el error
Desplazamientos_totales(nodo, :) = Desplazamiento_uy_ajustado;

[Frec_abs, Frec_neg, Frec_1er_fila] = calculo_frecuencias(Omega);

%% Vectores que acumulan las frecuencias calculadas para cada uno de los
nodos en los tres casos
All_frec_abs(nodo, :) = Frec_abs; %Estas frecuencias se obtienen para cuando
el sistema es dinamicamente estable, es decir, magnitud_lambda aprox = 1.
All_frec_neg(nodo, :) = Frec_neg;
All_frec_1er_fila(nodo, :) = Frec_1er_fila;

end

%% Análisis de datos para las frecuencias calculadas en los tres casos (outliers)
[nodo_atipico_abs, nodo_atipico_neg, nodo_atipico_1er_fila] =
Valor_atipico(All_frec_abs, All_frec_neg, All_frec_1er_fila);

%% Validación de la estabilidad del sistema por medio de los valores propios
ubicados en el circulo unitario.
%Sólo determina la circunferencia unitaria cuando este código se compila de
%individual por nodo. Si se compila en ciclo "for", el circulo unitario
%obtenido corresponderá al último nodo calculado (2.077).

magnitudes_lambda = abs(diag(Lambda));

%% Circunferencia unitaria para validar la estabilidad dinámica del sistema.
figure
theta = (0:1:100)*2*pi/100;
plot(cos(theta), sin(theta), 'k--') % plot unit circle
hold on, grid on
scatter(real(diag(Lambda)), imag(diag(Lambda)), 'ok')
axis([-1.1 1.1 -1.1 1.1]);

```

Anexo H. Algoritmo para la reconstrucción 3D del diapasón con datos originales y con datos DMD.

```
clear all; close; clc;

load("DATA_modelado.mat")
load('DATA_uy_600instantes.mat')
load('DMD_600_INSTANTANEAS.mat')

%Se toman sólo 200 instantes de tiempo, de los 6000 que compomen...
%... la data para 100 ciclos, es decir, se muestran los datos para cada...
%... 30ava columna de tiempo, alternando el diferencial de tiempo a: ...
%... dt = Periodo/2. Se muestran 2 instantes por cada periodo.

if exist ('instantaneas-3D','dir')
    rmdir('instantaneas-3D\','s');
end

if ~ exist ('instantaneas-3D', 'dir')
    mkdir ('instantaneas-3D\');
end

cd .\instantaneas-3D\

for columna = 1:600
    nombre_original = sprintf('%d_Original.png',columna);
    figure("Units","normalized","Position",[0 0.091 0.5 0.75],"Visible","off");
    pdeplot3D(modelo_transitorio,"ColorMapData", Data_original(:,columna))
    colorbar('off')
    limites = caxis;
    c = colorbar('south');
    c.Label.String = 'Desplazamiento - y [m]';
    title(sprintf(['ORIGINAL \n Tiempo = %d dt'], columna))
    print(nombre_original, '-dpng')

    nombre_DMD = sprintf('%d_DMD.png',columna);
    figure("Units","normalized","Position",[0.5 0.091 0.5 0.75],"Visible","off");
    pdeplot3D(modelo_transitorio, "ColorMapData",
Desplz_600inst_conceros(:,columna))
    colorbar('off')
    c = colorbar('south');
    c.Label.String = 'Desplazamiento - y [m]';
    caxis([limites])
    title(sprintf(['DMD \n Tiempo = %d dt'], columna))
    print(nombre_DMD, '-dpng')
end

%Las ilustraciones 3D para el diapasón original y el reconstruido se
%alamecenan en una carpeta como imágenes de extensión .png
oldFolder = cd("../");
```