



Universidad de Valladolid



**ESCUELA DE INGENIERÍAS
INDUSTRIALES**

**UNIVERSIDAD DE VALLADOLID
ESCUELA DE INGENIERIAS INDUSTRIALES**

Grado en Ingeniería Mecánica

Estimación de propiedades dinámicas de estructuras esbeltas mediante visión artificial.

Autor:

Cabero Martín, Carlos

Tutor(es):

Lorenzana Ibán, Antolín

Magdaleno González, Álvaro

**Departamento de Construcciones
Arquitectónicas, Ingeniería del Terreno y
Mecánica de los Medios Continuos y
Teoría de Estructuras**

Valladolid, octubre 2023.



AGRADECIMIENTOS

A mis profesores por su ayuda.

A mi familia y amigos por el apoyo recibido.

RESUMEN

La visión artificial ha evolucionado enormemente en los últimos años gracias al gran avance en la informática y la generalización del uso de dispositivos de adquisición de imágenes. En este trabajo se hará uso de esta tecnología para crear un sensor óptico capaz de analizar diferentes propiedades dinámicas de ciertas maquetas de estructuras esbeltas, sistemas con vibración libre subamortiguada de uno, dos y tres grados de libertad, desarrollando así un sistema más económico y sencillo de utilizar que los acelerómetros típicamente utilizados en este tipo de aplicaciones. Se desarrollará un sistema de visión artificial, formado por un dispositivo móvil y un programa informático capaz de procesar los videos registrados, implementado en MATLAB, que estimará la frecuencia y el amortiguamiento de las estructuras bajo estudio. Los resultados obtenidos serán comparados con aquellos procedentes de los métodos tradicionales, estimados mediante el procesamiento de datos registrados mediante sensores de desplazamiento basados en tecnología láser.

PALABRAS CLAVE

Vibraciones, Análisis modal, Procesamiento de imágenes, Algoritmo, Grados de libertad.

ABSTRACT

Artificial vision has evolved drastically in the last few years thanks to the great progress in informatics and the generalisation of the use of image acquisition devices. In the following project, this technology will be used to create an optical sensor capable of analysing different dynamic properties of certain models of slender structures, systems with free underdamped vibration of one, two and three degrees of freedom, thus developing a system that is cheaper and simpler to use than those typically used in this type of application. An artificial vision system will be developed, consisting of a mobile device and a computer programme capable of processing the recorded videos, implemented in MATLAB, which will be able to analyze the frequency and damping of the studies structures. The results obtained will be compared with those from traditional methods, calculated by processing data recorded by laser-based displacement sensors.

KEY WORDS

Vibrations, Modal analysis, Image processing, Algorithm, Degrees of freedom.



ÍNDICE DE CONTENIDO

CAPÍTULO 1. INTRODUCCIÓN Y OBJETIVOS.....	1
1.1. Introducción	1
1.2. Objetivos.....	4
1.3. Estructura del Trabajo	4
CAPÍTULO 2. MARCO TEÓRICO.....	5
CAPÍTULO 3. EQUIPO NECESARIO	9
3.1. Sensores.....	9
3.2. Tarjeta de adquisición de datos.....	10
3.3. Ordenador.....	11
3.4. Estructura	12
3.5. Dispositivo móvil o cámara de video	14
3.6. Trípode.....	15
3.7. Software MATLAB.....	15
CAPÍTULO 4. DESARROLLO DEL CÓDIGO EN MATLAB	17
4.1. Primera parte: visión artificial y procesamiento de imágenes.....	17
4.2. Segunda parte: análisis de funciones	30
CAPÍTULO 5. PROCEDIMIENTO DE LABORATORIO.....	49
CAPÍTULO 6. RESULTADOS. COMPARACIÓN LASER CON VISIÓN	53
6.1. Estructura de 1 grado de libertad sin masas (1p0m).....	53
6.2. Estructura de 1 grado de libertad con masa de 25/8 kg (1p1m).....	58
6.3. Estructura de 1 grado de libertad con dos masas de 25/8 kg (1p2m).....	63
6.4. Medio continuo (plataforma de madera) sin masas (mc0)	68
6.5. Medio continuo (plataforma de madera) con masa de 90 kg (mcM)	73
6.6. Estructura de 2 grados de libertad con dos masas de 25/8 kg (2p2m).....	78



6.7. Estructura de 3 grados de libertad sin masas (3p0m)	85
6.8. Estructura de 4 grados de libertad sin masas (4p0m)	92
CAPÍTULO 7. CONCLUSIONES Y LÍNEAS FUTURAS	103
7.1. Conclusiones	103
7.2. Líneas futuras	105
7.3. Consideraciones adicionales	106
BIBLIOGRAFÍA.....	107
ANEXO I: CÓDIGO EN MATLAB	111

ÍNDICE DE FIGURAS

Figura 1: Esquema de funcionamiento de un sistema de visión artificial.....	2
Figura 2: Diagrama del flujo del proceso seguido por el algoritmo de visión artificial.....	3
Figura 3: Sistema Subamortiguado, Sobreamortiguado y Críticamente amortiguado. [5]	6
Figura 4: Sistema de 1 grado de libertad. [5]	7
Figura 5: Acelerómetro	9
Figura 6: Diagrama de funcionamiento de un sensor láser reflectivo. [6].....	10
Figura 7: Sensor láser	10
Figura 8: Tarjeta de adquisición de datos	11
Figura 9: Ordenador portátil	11
Figura 10: Software Dewesoft.....	11
Figura 11: Maqueta de estructura de 1 gdl.....	12
Figura 12: Maqueta de estructura de 2 gdl.....	12
Figura 13: Maqueta de estructura de 3 gdl.....	13
Figura 14: Maqueta de estructura de 4 gdl.....	13
Figura 15: Plataforma de madera (1 gdl)	14
Figura 16: Dispositivo móvil	14
Figura 17: Trípode	15
Figura 18: Software MATLAB R2023a. [7]	15
Figura 19: Fragmento de código donde se eliminan los datos de ejecuciones previas.....	17
Figura 20: Fragmento de código donde se importan los archivos, se obtienen las propiedades del video y se crean los vectores de coordenadas y tiempos.....	18
Figura 21: Propiedades del video.....	18
Figura 22: Fragmento de código donde se implementa el comienzo del bucle while para detectar el punto.	19
Figura 23: Resultado de la lectura del video	20
Figura 24: Resultado tras transformar cada fotograma a blanco y negro	20
Figura 25: Resultado tras segmentar la imagen extrayendo la capa de color rojo.....	21
Figura 26: Resultado de la sustracción de la capa de color rojo de la imagen en blanco y negro... ..	22
Figura 27: Histograma de la imagen.....	22
Figura 28: Imagen binarizada	23
Figura 29: Detalle del punto tras binarizar la imagen	23
Figura 30: Fragmento de código donde se implementa la segunda parte del bucle para detectar un punto y mostrar sus coordenadas.....	24
Figura 31 Elemento estructurante plano. [24].....	25
Figura 32: Imagen tras realizar la apertura morfológica, aplicar el filtro y rellenar huecos.....	26
Figura 33: Detalle del punto tras realizar la apertura morfológica, aplicar el filtro y rellenar huecos	26
Figura 34: Matriz L, donde se observa el objeto etiquetado mediante unos	27
Figura 35: Coordenadas del centroide del último punto del primer video.....	27
Figura 36: Propiedades de la región etiquetada de la matriz L, obtenidas mediante la función regionprops	28

Figura 37: Resultado final, detectando las coordenadas del punto en cada fotograma del vídeo, mostrando la imagen binarizada.....	29
Figura 38: Resultado final, detectando las coordenadas del punto en cada fotograma del vídeo, mostrando la imagen del video original.....	29
Figura 39: Matriz de coordenadas del centroide del punto.....	30
Figura 40: Fragmento de código que escala los datos de visión, los ajusta y crea un gráfico del desplazamiento en función del tiempo.....	31
Figura 41: Código para medir la longitud de la pegatina, en píxeles, de cada video.....	31
Figura 42: Medida real de la pegatina, en metros.....	32
Figura 43: Acceso a la aplicación Image Viewer	32
Figura 44: Importar la imagen.....	33
Figura 45: Importar la imagen.....	33
Figura 46: Medida de la pegatina en píxeles.....	34
Figura 47: Gráfico del desplazamiento en función del tiempo obtenido mediante visión.....	35
Figura 48: Fragmento de código que importa los datos del láser, los ajusta y crea un gráfico del desplazamiento en función del tiempo.....	36
Figura 49 Gráfico del desplazamiento en función del tiempo obtenido mediante el sensor láser .	36
Figura 50: Fragmento de código mediante el que se acondicionan los datos obtenidos por visión y se crea el correspondiente gráfico.....	37
Figura 51: Fragmento de código mediante el que se acondicionan los datos obtenidos por el láser y se crea el correspondiente gráfico.....	38
Figura 52: Gráfico acondicionado del desplazamiento en función del tiempo obtenido mediante visión.....	39
Figura 53: Gráfico acondicionado del desplazamiento en función del tiempo obtenido mediante el sensor láser.....	39
Figura 54: Fragmento de código de donde se superponen los gráficos de visión y láser.....	40
Figura 55: Superposición de las gráficas obtenidas mediante visión y láser.....	40
Figura 56: Fragmento de código que llama a las funciones de ajuste de las curvas de visión y láser.....	41
Figura 57: Acceso a la aplicación Curve Fitter.....	41
Figura 58: Selección de los datos a ajustar en la aplicación Curve Fitter.....	42
Figura 59: Desplazamiento en función del tiempo de un sistema libre amortiguado de 1 gdl, frecuencia en Hz.....	42
Figura 60: Desplazamiento en función del tiempo de un sistema libre amortiguado de 2 gdl, frecuencia en Hz.....	43
Figura 61: Desplazamiento en función del tiempo de un sistema libre amortiguado de 3 gdl, frecuencia en Hz.....	43
Figura 62: Resultados obtenidos tras el ajuste a la función deseada.....	44
Figura 63: Restricciones a los parámetros de la función para su ajuste.....	44
Figura 64: Función que ajusta los datos del sistema de 1 gdl obtenidos mediante visión.....	45
Figura 65: Gráfico del ajuste del sistema de 1 gdl obtenido mediante visión.....	45
Figura 66: Resultados del ajuste del sistema de 1 gdl obtenidos mediante visión.....	46
Figura 67: Gráfico del ajuste del sistema de 1 gdl obtenido mediante el sensor láser.....	47
Figura 68: Resultados del ajuste del sistema de 1 gdl obtenidos mediante el sensor láser.....	47



Figura 69: Interfaz del programa DewesoftX 49

Figura 70: Colocación del sensor láser y el acelerómetro para medir desplazamientos horizontales 49

Figura 71: Gráficos de desplazamiento y aceleración obtenidos por el láser y el acelerómetro 51

Figura 72: Estructura de 1 gdl sin masas 53

Figura 73: Gráfico del desplazamiento en función del tiempo 1p0m. Visión 54

Figura 74: Gráfico del desplazamiento en función del tiempo 1p0m. Láser 54

Figura 75: Gráfico acondicionado del desplazamiento en función del tiempo 1p0m. Visión..... 55

Figura 76: Gráfico acondicionado del desplazamiento en función del tiempo 1p0m. Láser 55

Figura 77: Superposición de gráficos de desplazamiento láser y visión 1p0m 56

Figura 78: Ajuste 1p0m visión 56

Figura 79: Ajuste 1p0m láser 57

Figura 80: Estructura de 1 gdl con masa de 25/8 kg 58

Figura 81: Gráfico del desplazamiento en función del tiempo 1p1m. Visión 58

Figura 82: Gráfico del desplazamiento en función del tiempo 1p1m. Láser 59

Figura 83: Gráfico acondicionado del desplazamiento en función del tiempo 1p1m. Visión..... 59

Figura 84: Gráfico acondicionado del desplazamiento en función del tiempo 1p1m. Láser 60

Figura 85: Superposición de gráficos de desplazamiento láser y visión 1p1m 60

Figura 86: Ajuste 1p1m. Visión..... 61

Figura 87: Ajuste 1p1m. Láser 62

Figura 88: Estructura de 1 gdl con dos masas de 25/8 kg 63

Figura 89: Gráfico del desplazamiento en función del tiempo 1p2m. Visión 63

Figura 90: Gráfico del desplazamiento en función del tiempo 1p2m. Láser 64

Figura 91: Gráfico acondicionado del desplazamiento en función del tiempo 1p2m. Visión..... 64

Figura 92: Gráfico acondicionado del desplazamiento en función del tiempo 1p2m. Láser 65

Figura 93: Superposición de gráficos de desplazamiento láser y visión 1p2m 65

Figura 94: Ajuste de 1p2m visión 66

Figura 95: Ajuste 1p2m láser 67

Figura 96: Plataforma de madera, medio continuo sin masas..... 68

Figura 97: Gráfico del desplazamiento en función del tiempo mc0. Visión..... 68

Figura 98: Gráfico del desplazamiento en función del tiempo mc0. Láser 69

Figura 99: Gráfico acondicionado del desplazamiento en función del tiempo mc0. Visión 69

Figura 100: Gráfico acondicionado del desplazamiento en función del tiempo mc0. Láser 70

Figura 101: Superposición de gráficos de desplazamiento láser y visión mc0 70

Figura 102: Ajuste mc0 visión 71

Figura 103: Ajuste mc0 láser 72

Figura 104: Plataforma de madera, medio continuo con masa de 90 kg 73

Figura 105: Gráfico del desplazamiento en función del tiempo mcM. Visión 73

Figura 106: Gráfico del desplazamiento en función del tiempo mcM. Láser 74

Figura 107: Gráfico acondicionado del desplazamiento en función del tiempo mcM. Visión..... 74

Figura 108: Gráfico acondicionado del desplazamiento en función del tiempo mcM. Láser 75

Figura 109: Superposición de gráficos de desplazamiento láser y visión mcM 75

Figura 110: Ajuste mcM visión 76

Figura 111: Ajuste mcM láser 77



Figura 112: Estructura de 2 gdl con dos masas de 25/8 kg.....	78
Figura 113: Gráfico del desplazamiento en función del tiempo 2p2m. Visión	78
Figura 114: Gráfico del desplazamiento en función del tiempo 2p2m. Láser	79
Figura 115: Gráfico acondicionado del desplazamiento en función del tiempo 2p2m. Visión.....	79
Figura 116: Gráfico acondicionado del desplazamiento en función del tiempo 2p2m. Láser	80
Figura 117: Superposición de gráficos de desplazamiento láser y visión 2p2m	80
Figura 118: Ajuste 2p2m a 1 gdl. Visión	81
Figura 119: Ajuste 2p2m a 1 gdl. Láser	82
Figura 120: Ajuste 2p2m a 2 gdl. Visión	83
Figura 121: Ajuste 2p2m a 2 gdl. Láser	84
Figura 122: Estructura de 3 gdl sin masas.....	85
Figura 123: Gráfico del desplazamiento en función del tiempo 3p0m. Visión	85
Figura 124: Gráfico del desplazamiento en función del tiempo 3p0m. Láser	86
Figura 125: Gráfico acondicionado del desplazamiento en función del tiempo 3p0m. Visión.....	86
Figura 126: Gráfico acondicionado del desplazamiento en función del tiempo 3p0m. Láser	87
Figura 127: Superposición de gráficos de desplazamiento láser y visión 3p0m	87
Figura 128: Ajuste 3p0m a 1 gdl. Visión	88
Figura 129: Ajuste 3p0m a 1 gdl. Láser	89
Figura 130: Ajuste 3p0m a 2 gdl. Visión	90
Figura 131: Ajuste 3p0m a 2 gdl. Láser	91
Figura 132: Estructura de 4 gdl sin masas.....	92
Figura 133: Gráfico del desplazamiento en función del tiempo 4p0m. Visión	92
Figura 134: Gráfico del desplazamiento en función del tiempo 4p0m. Láser	93
Figura 135: Gráfico acondicionado del desplazamiento en función del tiempo 4p0m. Visión.....	93
Figura 136: Gráfico acondicionado del desplazamiento en función del tiempo 4p0m. Láser	94
Figura 137: Superposición de gráficos de desplazamiento láser y visión 4p0m	94
Figura 138: Ajuste 4p0m a 1 gdl. Visión	95
Figura 139: Ajuste 4p0m a 1 gdl. Láser	96
Figura 140: Ajuste 4p0m a 2 gdl. Visión	97
Figura 141: Ajuste 4p0m a 2 gdl. Láser	98
Figura 142: Ajuste 4p0m a 3 gdl. Visión	99
Figura 143: Ajuste 4p0m a 3 gdl. Láser	100
Figura 144: Aplicación a seguimiento de dos puntos simultáneamente	105



ÍNDICE DE TABLAS

Tabla 1: Resultados 1p0m visión.....	57
Tabla 2: Resultados 1p0m láser	57
Tabla 3: Resultados 1p1m visión.....	61
Tabla 4: Resultados 1p1m láser	62
Tabla 5: Resultados 1p2m visión.....	66
Tabla 6: Resultados 1p2m láser	67
Tabla 7: Resultados mc0 visión	71
Tabla 8: Resultados mc0 láser.....	72
Tabla 9: Resultados mcM visión.....	76
Tabla 10: Resultados mcM láser	77
Tabla 11: Resultados 2p2m visión 1gdl.....	81
Tabla 12: Resultados 2p2m láser 1gdl.....	82
Tabla 13: Resultados 2p2m visión 2gdl.....	83
Tabla 14: Resultados 2p2m láser 2gdl.....	84
Tabla 15: Resultados 3p0m visión 1gdl.....	88
Tabla 16: Resultados 3p0m láser 1gdl.....	89
Tabla 17: Resultados 3p0m visión 2gdl.....	90
Tabla 18: Resultados 3p0m láser 2gdl.....	91
Tabla 19: Resultados 4p0m visión 1gdl.....	95
Tabla 20: Resultados 4p0m láser 1gdl.....	96
Tabla 21: Resultados 4p0m visión 2gdl.....	97
Tabla 22: Resultados 4p0m láser 2gdl.....	98
Tabla 23: Resultados 4p0m visión 3gdl.....	99
Tabla 24: Resultados 4p0m láser 3gdl.....	100
Tabla 25: Comparación de las frecuencias y coeficientes de amortiguamiento estimados por visión para cada ajuste	101
Tabla 26: Comparación de las frecuencias y coeficientes de amortiguamiento estimados del láser para cada ajuste	101
Tabla 27: Comparación frecuencias estimadas con laser y visión 4p0m	101
Tabla 28: Comparación coeficientes de amortiguamiento estimadas con laser y visión 4p0m	101
Tabla 29: Comparación de error relativo entre las frecuencias obtenidas mediante visión y láser	103
Tabla 30: Comparación de error relativo entre los coeficientes de amortiguamiento obtenidas mediante visión y láser.....	104

CAPÍTULO 1. INTRODUCCIÓN Y OBJETIVOS

1.1. Introducción

El estudio de las vibraciones estructurales es un campo fundamental en la ingeniería mecánica, ya que comprende una parte importante en la evaluación y diseño de estructuras. El estudio de los modos de vibración de las estructuras es necesario para garantizar su seguridad en diversas aplicaciones como edificios, puentes o maquinaria industrial. En este contexto, este trabajo busca aportar una nueva metodología para la caracterización de las vibraciones estructurales, mediante la creación de un sensor óptico a partir de un teléfono móvil que permita grabar videos y un código en MATLAB con el que analizarlos para obtener diferentes propiedades dinámicas como la frecuencia o el coeficiente de amortiguamiento.

Las técnicas tradicionales de medición de vibraciones estructurales implican el uso de sensores costosos, como sensores láser o acelerómetros. Este trabajo propone una aproximación innovadora que combina visión artificial y procesamiento de imágenes con su posterior análisis para obtener información sobre el comportamiento de las estructuras, consiguiendo realizar este análisis mediante un método más económico.

Realizando una búsqueda previa sobre este tema se han encontrado diversos trabajos que tratan sobre visión artificial y reconocimiento de imágenes, algunos de ellos desarrollados en MATLAB, como el trabajo [1], que implementa un sistema de identificación y seguimiento de objetos RGB y otros que profundizan en la teoría y funcionamiento de la visión artificial, como el [2] y el [3], pero ninguno de ellos se enfoca en el análisis de propiedades dinámicas de estructuras.

Además, existen softwares de *tracking* de objetos, como puede ser **Tracking Video Analysis and Modeling Tool** [4], una aplicación que se puede utilizar en línea o descargarse en Windows, Mac y Linux, y permite analizar diferentes propiedades como posiciones, velocidades o aceleraciones de objetos en movimiento del video importado.

“La visión artificial es una disciplina de la inteligencia artificial que aplica el procesamiento de imágenes para el reconocimiento de patrones, con el uso de algoritmos en ambientes controlados con una cantidad de iteraciones en el procesamiento de imágenes.” [2]

Hoy en día, gracias al avance en los dispositivos móviles, la visión artificial se está implementando cada vez más en todos los ámbitos, ya que es un sistema que facilita y agiliza el reconocimiento de imágenes de cualquier tipo, reduciendo los tiempos de espera y el coste de los equipos, robusteciendo el proceso aportando fiabilidad. Es por esto que tiene numerosas aplicaciones en el ámbito industrial y civil, así como en medicina o robótica. Existen dispositivos capaces de detectar formas, tamaños y colores de objetos, identificar piezas, leer códigos de barras, reconocer caracteres,

identificación facial, realizar mediciones e inspeccionar detalles que el ojo humano no es capaz de percibir.

Por eso este tipo de sistemas se utilizan en la industria en control de calidad, siendo capaces de detectar defectos en piezas, originando un proceso mucho más rápido y robusto que una inspección visual; sistemas embarcados en robots de manera que pueden detectar dónde deben realizar las diferentes operaciones; en electrónica y semiconductores, donde su reducido tamaño y formas geométricas dificulta la inspección visual por parte de una persona.

La parte de visión artificial de este trabajo es sencilla dentro de la inmensa capacidad que pueden llegar a tener estos sistemas. Se basa en la creación de un código para el reconocimiento de un punto de color rojo a lo largo de cada uno de los fotogramas de un vídeo previamente registrado de una estructura en movimiento, grabado con un dispositivo móvil, de manera que detecte las coordenadas del centroide de dicho punto para así poder obtener los desplazamientos de la estructura y analizar sus parámetros modales.

El sistema de visión artificial se diseña de manera que la cámara actúa como el ojo humano y el sistema informático como el cerebro, de tal forma que debe ser capaz de adquirir las imágenes, procesarlas y obtener la información necesaria, en este caso reconocer la presencia del objeto de estudio, y medir su posición dentro de la imagen. Se muestra un esquema del funcionamiento de este tipo de sistemas en la Figura 1.

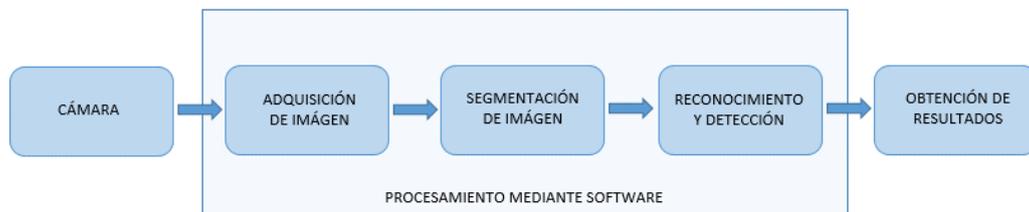


Figura 1: Esquema de funcionamiento de un sistema de visión artificial

El algoritmo para llevar a cabo la detección y procesamiento de las imágenes y la caracterización de las mismas, seguirá el proceso mostrado en la Figura 2.

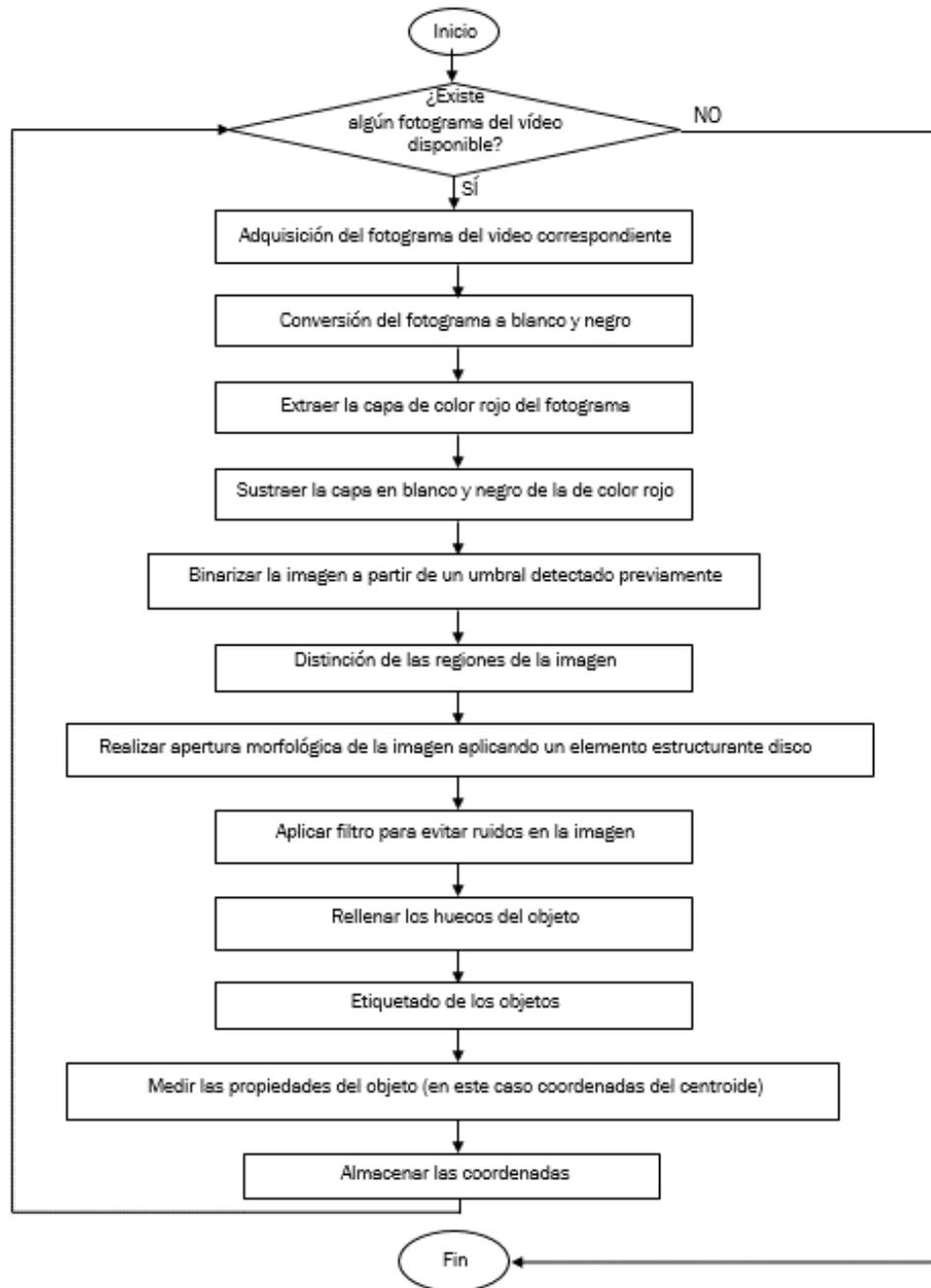


Figura 2: Diagrama del flujo del proceso seguido por el algoritmo de visión artificial

Este algoritmo se desarrollará empleando el software MATLAB, resultando simple de programar con este lenguaje ya que “el toolbox de procesamiento de imágenes contiene un conjunto de funciones de los algoritmos más conocidos para trabajar con imágenes binarias, transformaciones geométricas, morfología y manipulación de color que junto con las funciones ya integradas en MATLAB permite realizar análisis y transformaciones de imágenes a través de simples instrucciones y comandos.” [1]



1.2. Objetivos

El principal objetivo de este trabajo es desarrollar un sistema de adquisición de datos económico para analizar los parámetros modales de estructuras de uno, dos y tres grados de libertad. Para conseguirlo, se llevará a cabo en las siguientes etapas:

- Grabación de vídeo de una estructura en movimiento, con un punto rojo marcado en ella para después poder analizarlo.
- Desarrollo de un *script* en MATLAB que emplea técnicas de visión artificial y procesamiento de imágenes para identificar y seguir el punto de interés en el vídeo, que representa un punto en la estructura.
- Extracción de las posiciones de dicho punto en función del tiempo, lo que permitirá analizar la respuesta dinámica de la estructura, enfocándose en frecuencias y coeficientes de amortiguamiento.
- Ajuste de los datos obtenidos a las funciones apropiadas, lo que posibilitará la identificación de los parámetros modales de la estructura.

1.3. Estructura del Trabajo

Este trabajo se organizará en diversas secciones que abordarán cada una de las etapas mencionadas anteriormente. Se introducirá el marco teórico de dinámica de estructuras sobre el que se va a trabajar, se incluirán detalles sobre el equipo utilizado, los algoritmos implementados, los procedimientos seguidos y los resultados obtenidos, además de un análisis, comparación de los resultados finales y posibles aplicaciones futuras.

CAPÍTULO 2. MARCO TEÓRICO

Las vibraciones estructurales son oscilaciones repetitivas que ocurren en una estructura cuando esta es sometida a fuerzas dinámicas. Estas oscilaciones pueden ser causadas por diversas fuentes como cargas externas, excitaciones sísmicas, viento, maquinaria o perturbaciones iniciales. El estudio de las vibraciones estructurales es fundamental en ingeniería civil y mecánica, ya que permite evaluar la seguridad y el comportamiento de las estructuras en condiciones de carga dinámica.

Para definir estos sistemas, se van a explicar previamente una serie de conceptos:

- Frecuencia natural de vibración: $\omega_n = \sqrt{\frac{k}{m}}$ [rad/s] o $f_n = \frac{\omega_n}{2\pi}$ [Hz]
- Amortiguamiento crítico: $c_{cr} = 2m\omega_n = 2\sqrt{km} = \frac{2k}{\omega_n}$ [kg/s]
- Razón de amortiguamiento crítico: $\zeta = \frac{c}{c_{cr}} = \frac{c}{2m\omega_n}$ (Parámetro adimensional)

Las vibraciones estructurales se pueden clasificar atendiendo a distintos criterios:

Existencia de fuerzas externas:

- **Vibraciones Libres:** Ocurren cuando una estructura oscila sin la influencia de fuerzas externas después de una perturbación inicial. En este caso, la estructura vibra naturalmente debido a su rigidez y masa.
- **Vibraciones Forzadas:** Se producen cuando una estructura es sometida a una fuerza externa periódica o no periódica. Las vibraciones forzadas pueden ser de amplitud constante o variar en el tiempo.

Existencia de amortiguamiento:

- **Sistemas No Amortiguados:** En estos sistemas, no se considera la disipación de energía debido a la fricción interna o la resistencia del medio. Las vibraciones continuarán de forma indefinida una vez iniciadas, de modo que la amplitud de la vibración no disminuirá con el tiempo. Es un caso ideal.
- **Sistemas Amortiguados:** En estos sistemas, se introduce una fuerza de amortiguamiento que disipa la energía con el tiempo, lo que lleva a que las vibraciones disminuyan su amplitud hasta detenerse en un tiempo limitado.

“El proceso mediante el cual la amplitud de la vibración disminuye de manera constante se denomina amortiguamiento. La energía cinética y la energía de deformación del sistema vibratorio se disipan mediante diversos mecanismos de amortiguamiento” [5]

Este tipo de sistemas se pueden clasificar dependiendo de su coeficiente de amortiguamiento, se muestran en la Figura 3.

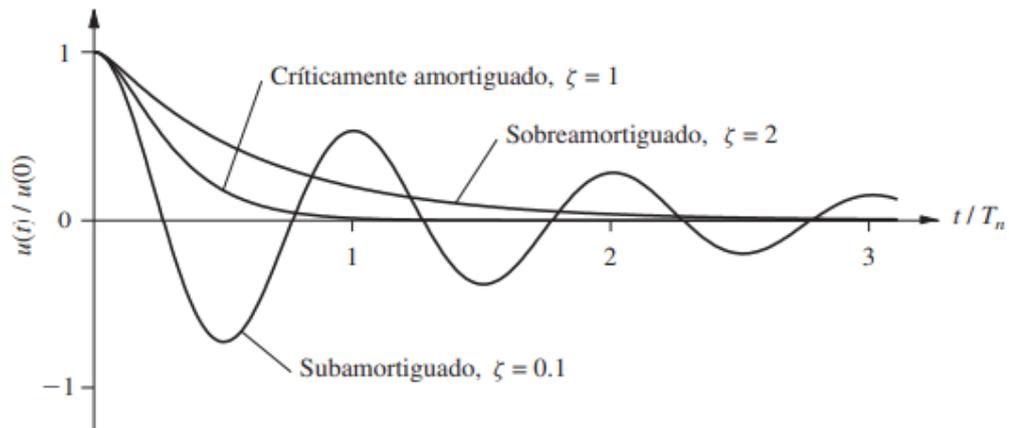


Figura 3: Sistema Subamortiguado, Sobreamortiguado y Críticamente amortiguado. [5]

Subamortiguados:

Los sistemas subamortiguados son aquellos en los que el coeficiente de amortiguamiento, ζ , es menor que 1 ($0 < \zeta < 1$). Estos sistemas presentan oscilaciones amortiguadas en respuesta a una perturbación inicial y tardan un tiempo finito en volver a su posición de equilibrio después de ser perturbados. La amplitud de las oscilaciones disminuye gradualmente con el tiempo.

Sobreamortiguados:

Los sistemas sobreamortiguados tienen un coeficiente de amortiguamiento mayor que 1 ($\zeta > 1$). En estos sistemas, las oscilaciones son prácticamente imperceptibles, ya que el amortiguamiento es tan fuerte que el sistema regresa a su posición de equilibrio sin oscilaciones significativas. El tiempo de respuesta es más rápido en comparación con los sistemas subamortiguados.

Críticamente Amortiguados:

Los sistemas críticamente amortiguados tienen un coeficiente de amortiguamiento igual a 1 ($\zeta = 1$). Estos sistemas alcanzan rápidamente su posición de equilibrio sin oscilaciones y sin demora. Son el equilibrio perfecto entre amortiguamiento y capacidad de respuesta rápida.

Grados de Libertad de un Sistema

El grado de libertad de un sistema se refiere al número mínimo de coordenadas independientes necesarias para describir completamente su estado. En el contexto de las vibraciones estructurales, los sistemas se dividen en dos categorías principales según su número de grados de libertad:

- **Sistemas de Un Grado de Libertad (1 gdl):** En estos sistemas, una sola coordenada es suficiente para describir completamente el estado del sistema. Por ejemplo, un péndulo simple que oscila en una dimensión.

- **Sistemas de n Grados de Libertad (n gdl):** Requieren dos o más coordenadas independientes para describir completamente su comportamiento vibratorio.

Sistema Vibratorio Libre Subamortiguado de 1 Grado de Libertad

En este trabajo se busca analizar los modos de vibración de maquetas de edificios, estructuras de marcos de uno dos y tres grados de libertad, al excitarla lateralmente mediante una impulso unitario y después dejarla vibrar libremente.

“Se dice que una estructura experimenta vibración libre cuando es perturbada de su posición de equilibrio estático y después se deja vibrar sin ninguna excitación dinámica externa.” [5], por lo tanto los casos de este estudio presentan vibración libre.

Además es un sistema amortiguado, ya que un sistema no amortiguado solo existiría en una situación ideal en la que no exista rozamiento. *“Aunque el amortiguamiento en las estructuras reales se debe a varios mecanismos de disipación de la energía que actúan de manera simultánea, un enfoque matemáticamente práctico consiste en idealizarlos mediante el amortiguamiento viscoso equivalente”* [5]

Y por último, es subamortiguado, ya que si *“ $\zeta < 1$, el sistema oscila alrededor de su posición de equilibrio con una amplitud que disminuye progresivamente”* [5]

Se comienza analizando el sistema de 1 grado de libertad, sistema compuesto por una masa m concentrada en un punto del nivel superior, una rigidez lateral k , y un amortiguamiento, con coeficiente de amortiguamiento c , actuando una fuerza $p(t)$ lateralmente sobre él. Los sistemas de 1 gdl se pueden aproximar mediante un marco de un nivel, como el que se muestra en la Figura 4 buena aproximación a nuestra estructura de estudio.

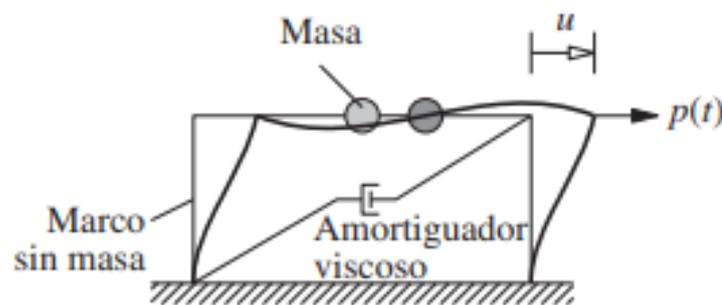


Figura 4: Sistema de 1 grado de libertad. [5]

Partiendo de la ecuación general de movimiento para sistemas de 1 gdl:

$$m\ddot{u} + c\dot{u} + ku = p(t) \quad (1)$$

Donde:

- m : masa del sistema
- c : coeficiente de amortiguamiento
- k : rigidez del muelle
- \ddot{u} : aceleración de la masa
- \dot{u} : velocidad de la masa
- u : desplazamiento de la masa
- $p(t)$: Fuerza externa aplicada

Teniendo en cuenta que el sistema es libre, no existe excitación externa a lo largo del tiempo, La ecuación de movimiento general que rige la vibración libre de un sistema de 1 gdl con subamortiguamiento, se puede expresar como:

$$m\ddot{u} + c\dot{u} + ku = 0 \quad (2)$$

Esta ecuación se puede resolver imponiendo las condiciones iniciales, de manera que se obtiene la ecuación general del desplazamiento lateral en función del tiempo de un sistema libre subamortiguado de 1 gdl, Donde ϕ es el desfase en radianes:

$$u(t) = A \sin(\omega_n t + \phi) e^{(-\zeta \omega_n t)} \quad (3)$$

Teniendo en cuenta que $f_n = \frac{\omega_n}{2\pi}$, se puede reescribir la ecuación de la siguiente manera, expresando la frecuencia en hercios:

$$u(t) = A \sin(2\pi f_n t + \phi) e^{(-\zeta 2\pi f_n t)} \quad (4)$$

Para sistemas con más de un grado de libertad, se utiliza comúnmente el método de superposición, basado en sumar la respuesta de cada uno de estos grados de libertad para obtener los desplazamientos en función del tiempo.

De esta manera, para los sistemas con más de un grado de libertad, se obtendrán las siguientes ecuaciones de desplazamiento en función del tiempo de un sistema subamortiguado, expresando la frecuencia en hercios:

Sistema de 2 grados de libertad

$$x(t) = A_1 \sin(2\pi f_1 t + \phi_1) e^{(-\zeta_1 2\pi f_1 t)} + A_2 \sin(2\pi f_2 t + \phi_2) e^{(-\zeta_2 2\pi f_2 t)} \quad (5)$$

Sistema de 3 grados de libertad

$$x(t) = A_1 \sin(2\pi f_1 t + \phi_1) e^{(-\zeta_1 2\pi f_1 t)} + A_2 \sin(2\pi f_2 t + \phi_2) e^{(-\zeta_2 2\pi f_2 t)} + A_3 \sin(2\pi f_3 t + \phi_3) e^{(-\zeta_3 2\pi f_3 t)} \quad (6)$$

CAPÍTULO 3. EQUIPO NECESARIO

Para comenzar se explicará el equipo requerido para llevar a cabo este trabajo, basado en adquirir datos de los desplazamientos y aceleraciones de diferentes estructuras en movimiento mediante dos métodos diferentes, para después analizarlos y obtener los parámetros modales de la estructura y comparar los resultados obtenidos por ambos métodos.

La adquisición de datos, conocido por sus siglas en inglés DAQ (Data Acquisition), que se basa en medir mediante un ordenador un proceso físico o eléctrico y es conformado por tres elementos principales: un sensor (en este caso un acelerómetro y un láser), una tarjeta de adquisición de datos (hardware) y un ordenador (software programable).

A continuación se detallará el equipo utilizado para llevar a cabo este proyecto:

3.1. Sensores

El sensor es el primer elemento en todo proceso de adquisición de datos, es un dispositivo capaz de captar magnitudes físicas como corrientes, voltajes, intensidad de sonido o de luz, presión, temperatura, etc. Existen diferentes tipos de sensores en función de la magnitud que se desea medir. En este caso se medirán aceleraciones (m/s^2), mediante un acelerómetro y desplazamientos (m), mediante un láser.

Acelerómetro

Los acelerómetros son dispositivos que miden la aceleración del movimiento de una estructura, formados por un material piezoeléctrico que al ser comprimido genera una corriente eléctrica proporcional a esta fuerza, que será traducida en aceleraciones. Figura 5.

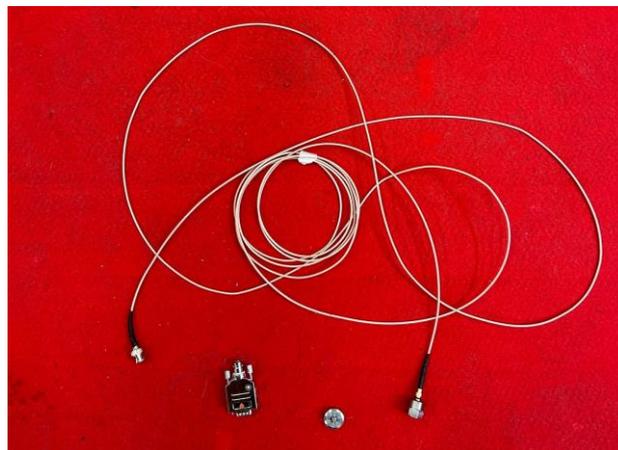


Figura 5: Acelerómetro

Láser (sensor de desplazamiento)

Este tipo de sensores utiliza un láser para emitir un haz en línea recta, que al ser visible facilita su posicionamiento. Su funcionamiento se basa en la emisión de un haz de luz por parte del emisor y la recepción por el receptor. Existen diferentes tipos, en este caso se empleará uno de modelo reflectivo, Figura 6, de forma que el emisor y el receptor se encuentran en el mismo dispositivo, reflejando el haz de luz láser en el objeto y captándola de vuelta en el receptor. En la Figura 7 se muestra el sensor láser utilizado en este trabajo.

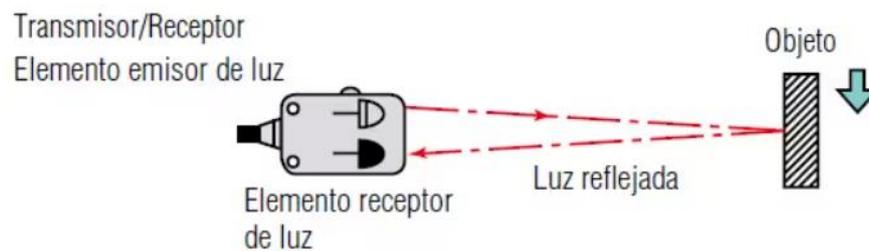


Figura 6: Diagrama de funcionamiento de un sensor láser reflectivo. [6]

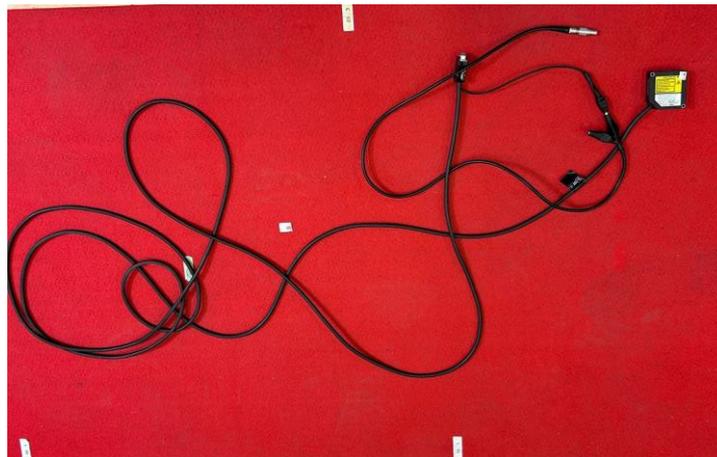


Figura 7: Sensor láser

3.2. Tarjeta de adquisición de datos

Las tarjetas de adquisición de datos sirven para transformar la señal de un sensor, analógico, en datos que puedan ser reconocidos por un sistema digital, para que la pueda leer un ordenador y realizar una determinada tarea mediante un software, en este caso DewesoftX. La tarjeta de adquisición utilizada para la toma de datos se muestra en la Figura 8.



Figura 8: Tarjeta de adquisición de datos

3.3. Ordenador

Será necesario un ordenador en el que se pueda desarrollar el software, en este caso un portátil, Figura 9. Debe tener instalado el programa DewesoftX, Figura 10, software de adquisición de datos, que permite la grabación de datos, procesamiento de señales y la visualización de datos. Este programa será necesario para visualizar las señales obtenidas por el acelerómetro y el sensor laser.



Figura 9: Ordenador portátil



Figura 10: Software Dewesoft

3.4. Estructura

Para el desarrollo de este trabajo se utilizarán las maquetas de estructuras esbeltas mostradas en las Figuras 11, 12, 13, 14 y 15, sobre las que se realizarán las mediciones. Se encuentran en el laboratorio de estructuras de la EII Paseo del Cauce, Valladolid:

Estructura de 1 grado de libertad.



Figura 11: Maqueta de estructura de 1 gdl

Estructura de 2 grados de libertad

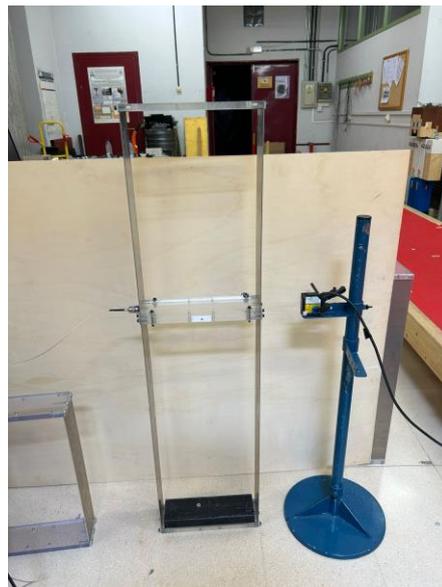


Figura 12: Maqueta de estructura de 2 gdl

Estructura de 3 grados de libertad.



Figura 13: Maqueta de estructura de 3 gdl

Estructura de 4 grados de libertad.



Figura 14: Maqueta de estructura de 4 gdl

Medio continuo (plataforma de madera)



Figura 15: Plataforma de madera (1 gdl)

3.5. Dispositivo móvil o cámara de video

Es necesario un dispositivo móvil que permita grabar videos, en este caso un iPhone 14 como el de la Figura 16. La cámara de este móvil permite grabar a 30 fps (*frames por segundo*), aunque serviría cualquier Smartphone o cámara de vídeo, siempre que su *frame rate* sea suficientemente alto para poder capturar correctamente las frecuencias que se desean medir.



Figura 16: Dispositivo móvil

3.6. Trípode

Se utiliza un trípode como el de la Figura 17 para colocar el teléfono móvil a la distancia y altura necesarias para poder captar el punto correctamente en los vídeos. En caso de no disponer de un trípode se podría sujetar el móvil con la mano o apoyado sobre otro objeto.



Figura 17: Trípode

3.7. Software MATLAB

“MATLAB es una plataforma de programación y cálculo numérico utilizada por millones de ingenieros y científicos para analizar datos, desarrollar algoritmos y crear modelos.” [7] En este caso se utilizará la versión MATLAB R2023a para estudiantes, Figura 18.

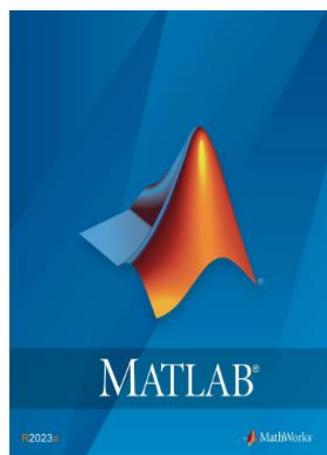


Figura 18: Software MATLAB R2023a. [7]

CAPÍTULO 4. DESARROLLO DEL CÓDIGO EN MATLAB

Para poder analizar los videos registrados se desarrollará un código en MATLAB que se puede dividir en dos partes, la primera enfocada en visión artificial y procesamiento de imágenes, con el principal objetivo de detectar las coordenadas de un punto rojo en cada fotograma de un video, situado en una estructura que vibra, y la segunda orientada a acondicionar los datos obtenidos y ajustarlos a una función, para analizar sus parámetros modales.

Para ejecutar este código se deben instalar en MATLAB los paquetes *Image Processing Toolbox* [8] y *Computer vision toolbox* [9], que permitirán ejecutar algunas funciones como importar un archivo de video o leer sus fotogramas. También el paquete *Curve Fitting Toolbox* [10] que permite ajustar curvas y graficarlas, para obtener las frecuencias y amortiguamientos de las funciones.

A continuación se explicará el código desarrollado por fragmentos. El código completo se puede encontrar en el **ANEXO I**.

4.1. Primera parte: visión artificial y procesamiento de imágenes

En primer lugar se va a describir el procedimiento seguido para crear un código en MATLAB que consiga detectar un punto rojo en cada fotograma de un video, mediante técnicas de procesamiento de imágenes y visión artificial. Este punto se encuentra ubicado en una estructura que vibra, de manera que obteniendo las coordenadas del centroide de este punto, se pueden analizar los desplazamientos de la estructura.

SEGUIMIENTO DE UN PUNTO MEDIANTE VISIÓN ARTIFICIAL

```
clc
clear
close all
cd 'C:\Users\Carlos\Desktop\TFG\ARCHIVOS'
```

Figura 19: Fragmento de código donde se eliminan los datos de ejecuciones previas

En la Figura 19 se muestra el primer fragmento del código, en el que se utiliza el comando **clear** [11] que elimina todos los elementos del *Workspace*, liberando memoria del sistema, **clc** [12] que elimina los datos del *Command Window*, para borrar todos los datos que pudieran permanecer de la ejecución de códigos previos y **close all** [13] que cierra todas las figuras que permanezcan abiertas. Así el programa está listo para ejecutarse de nuevo sin interferencia de ningún elemento de la ejecución de códigos anteriores. Se utiliza también la función **cd** [14], que cambia la carpeta actual, seleccionando aquella donde se encuentran los archivos de video y Excel a analizar.

IMPORTAR ARCHIVOS Y CREAR VECTOR DE TIEMPO Y COORDENADAS

```
% Importar archivo de video
file1 = 'carlos1.mp4';

%Importar archivo excel con los datos de los sensores recogidos en el lab
file2 = 'carlos1.xlsx';

% Crear archivo videoreader y obtener sus características.
vid = VideoReader(file1);
fps = vid.FrameRate;
num_frames = vid.NumFrames;
duracion = vid.Duration;

% Crear una matriz para almacenar las coordenadas x e y del punto
coords = zeros(num_frames, 2);

% Crear un vector de tiempo en frames (tf) desde 1 hasta el numero de
% frames del video y un vector de tiempos (t) en segundos, dividiendolo
% entre el número de frames por segundo.
tf = 0: num_frames-1;
t = (tf/30)';
```

Figura 20: Fragmento de código donde se importan los archivos, se obtienen las propiedades del video y se crean los vectores de coordenadas y tiempos

En el fragmento mostrado en la Figura 20 se comienza importando los archivos de video y Excel. El archivo **file1** corresponde a los vídeos registrados de la estructura cuyos desplazamientos se quieren analizar mediante procesamiento de imágenes y el archivo **file2** contiene los datos en Excel de los desplazamientos de la estructura adquiridos por el láser en el laboratorio.

Se importa el video usando la función **VideoReader** [15], la cual crea un objeto que permite leer archivos de video, en este caso **file1**, y se almacena en una objeto llamado **vid**, que contiene diferentes datos sobre el archivo de video (Figura 21). Además, se obtienen sus propiedades: número de fotogramas por segundo (fps) mediante la función **.FrameRate**, el número de fotogramas (frames) que contiene el video mediante **.NumFrames**, y su duración, información que se utilizará más adelante.

Property	Value
Duration	10.1350
Name	'carlos1.mp4'
Path	'C:\Users\Carlos\Desktop\TFG\VIDEOS TFG'
BitsPerPixel	24
FrameRate	29.9951
Height	1080
NumFrames	304
VideoFormat	'RGB24'
Width	1920
Tag	''
UserData	[]
CurrentTime	10.1350

Figura 21: Propiedades del video

A continuación se crea la matriz **coords**, matriz de ceros creada mediante la función **zeros** [16] que irá almacenando las coordenadas del centroide del punto rojo en cada iteración del bucle **while**. Se crean dos vectores, el primero **tf**, vector de tiempo en **frames**, que va desde 0 hasta el número de fotogramas del vídeo y **t**, vector de tiempo en segundos, dividiendo entre el **frame rate**, 30 fps.

$$tf(frames) = t(s) * 30 \left(\frac{frames}{s} \right) \quad (7)$$

En el siguiente fragmento de código, Figura 22, se comienza implementando el bucle **while** que permitirá la detección del punto utilizando técnicas de visión artificial y procesamiento de imágenes. Mediante la función **hasFrame** [17], seguirá realizando nuevas iteraciones mientras exista un nuevo frame del video (vid) disponible para leer, es decir, hasta que llegue al final del video.

DETECCIÓN DE PUNTO ROJO Y OBTENCIÓN DE COORDENADAS

```
i=1;
while hasFrame(vid)

% Leer el frame correspondiente del video
frame = readFrame(vid);

% Convertir cada imagen correspondiente a cada fotograma del video a escala
% de grises
grayframe = rgb2gray(frame);

% Extraer la capa de color rojo de cada fotograma del video
redframe = frame(:,:,1);

% Restar la capa en blanco y negro de la de color rojo
img = imsubtract(redframe, grayframe);

% En el command window: imhist(img) para ver el umbral donde se encuentra
% nuestro objeto para separarlo del resto de la imagen. En este caso 0.225

% Otra forma de binarizar la imagen (mejor)
bw = imbinarize(img, 0.225);
```

Figura 22: Fragmento de código donde se implementa el comienzo del bucle **while** para detectar el punto.

Dentro del bucle, primero se lee el fragmento del video correspondiente, mediante la función **readFrame** [18], función que lee el siguiente frame disponible de un video, y se almacena en la variable **frame**, que contendrá cada imagen del video en cada iteración del bucle. Añadiendo la función **imshow(frame)** (que se encuentra al final del código, para mostrar la imagen y será explicada posteriormente) se irá mostrando cada frame del video, Figura 23, en cada iteración del bucle **while** hasta llegar al final, de manera que se verá la consecución de imágenes como si se tratara del propio video. La función **imshow** [19] muestra por pantalla la imagen que se le especifique, en este caso, **frame**.

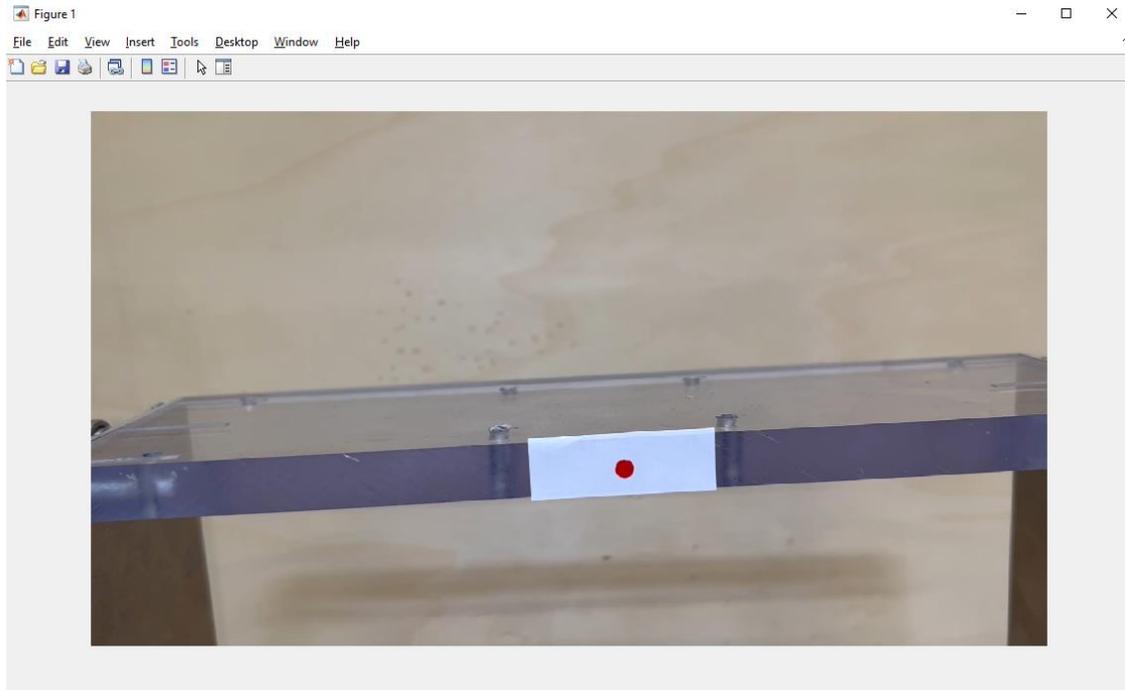


Figura 23: Resultado de la lectura del video

A continuación se convierte cada fotograma del video (frame) a blanco y negro mediante la función ***rgb2gray*** [20] y se almacena en la nueva variable llamada ***grayframe***. Esta transformación se realiza porque la imagen en blanco y negro tiene una menor cantidad de datos, por lo que su procesamiento será más rápido. Ejecutando el comando ***imshow(grayframe)***, se verá el video con cada uno de sus fotogramas en blanco y negro, como se puede observar en la Figura 24.

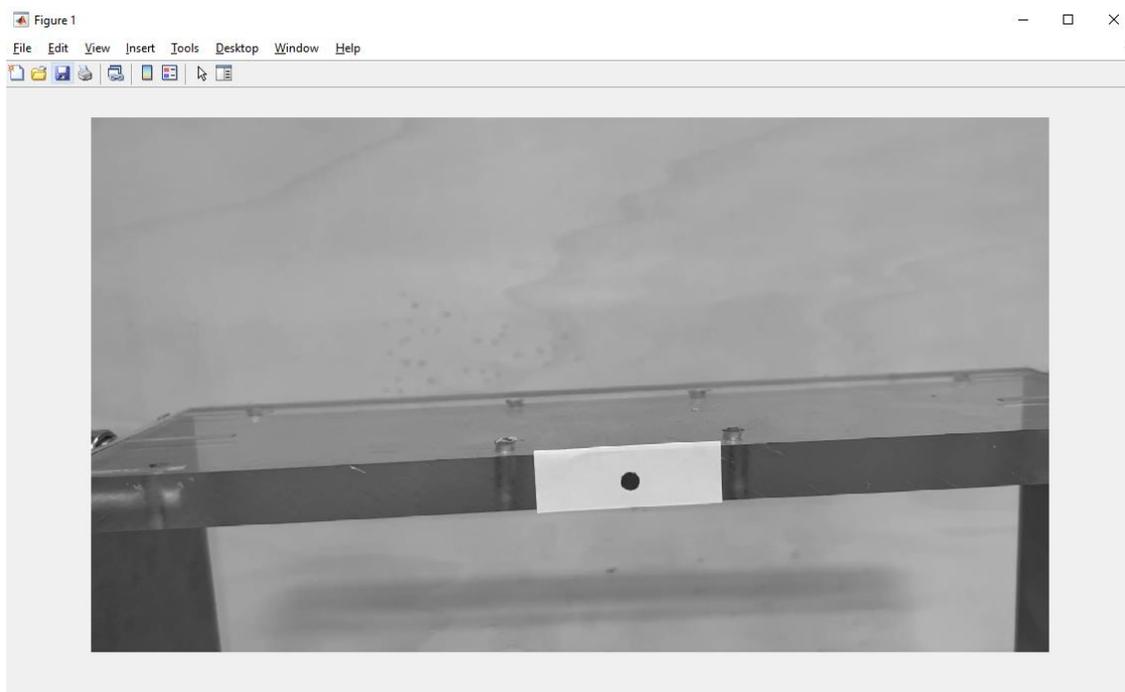


Figura 24: Resultado tras transformar cada fotograma a blanco y negro

A continuación se procede a la segmentación por color de la imagen, extrayendo la capa roja de esta, **redframe=frame(:, :, 1)**. El dos primeros valores del vector son los píxeles de las coordenadas x e y, que se mantienen todos mediante el comando “:”, de forma que permanecen las mismas dimensiones de la imagen. El tercer valor del vector es la capa de color, en este caso se extrae la roja que corresponde al 1. El 2 corresponde al verde y el 3 al azul. Se puede observar el resultado de la extracción de la capa roja en la Figura 25.



Figura 25: Resultado tras segmentar la imagen extrayendo la capa de color rojo

Una vez separadas las capas en blanco y negro y la de color rojo, se usa la función **imsubtract** [21] que permite sustraer una imagen de otra. Mediante el comando: **img = imsubtract(redframe, grayframe)**; se sustrae la imagen en blanco y negro (grayframe) de la de color rojo (redframe) y se almacena en una nueva variable llamada **img**. Ejecutando **imshow(img)**, se observa el resultado de restar la capa en blanco y negro de la capa roja, donde se aprecian los píxeles de color rojo en un tono más claro y el resto de píxeles en un tono más oscuro, pudiendo así diferenciarlo y separarlo correctamente del resto de la imagen (Figura 26).

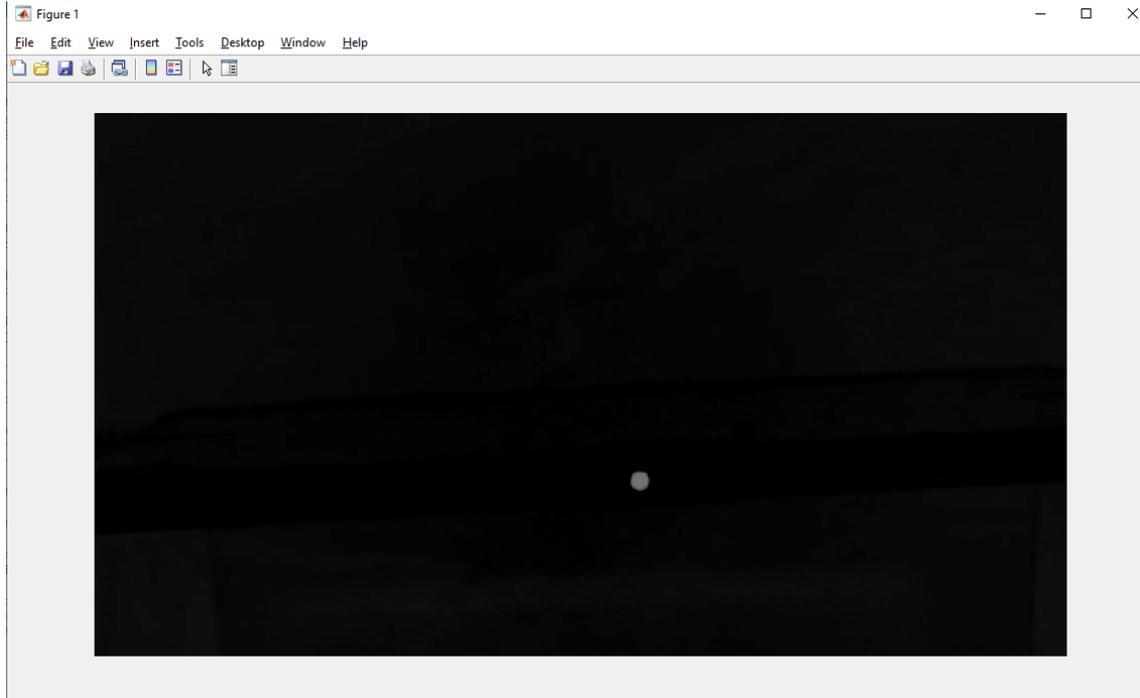


Figura 26: Resultado de la sustracción de la capa de color rojo de la imagen en blanco y negro

A continuación se procede a la binarización de la imagen, para ello se ejecuta en el *Command Window* el comando `imhist(img)` [22] que muestra una gráfica del histograma de datos de una imagen, en este caso de `img`, previamente obtenida. En cada punto se pueden obtener los valores de cada color primario, rojo, verde y azul (RBG).

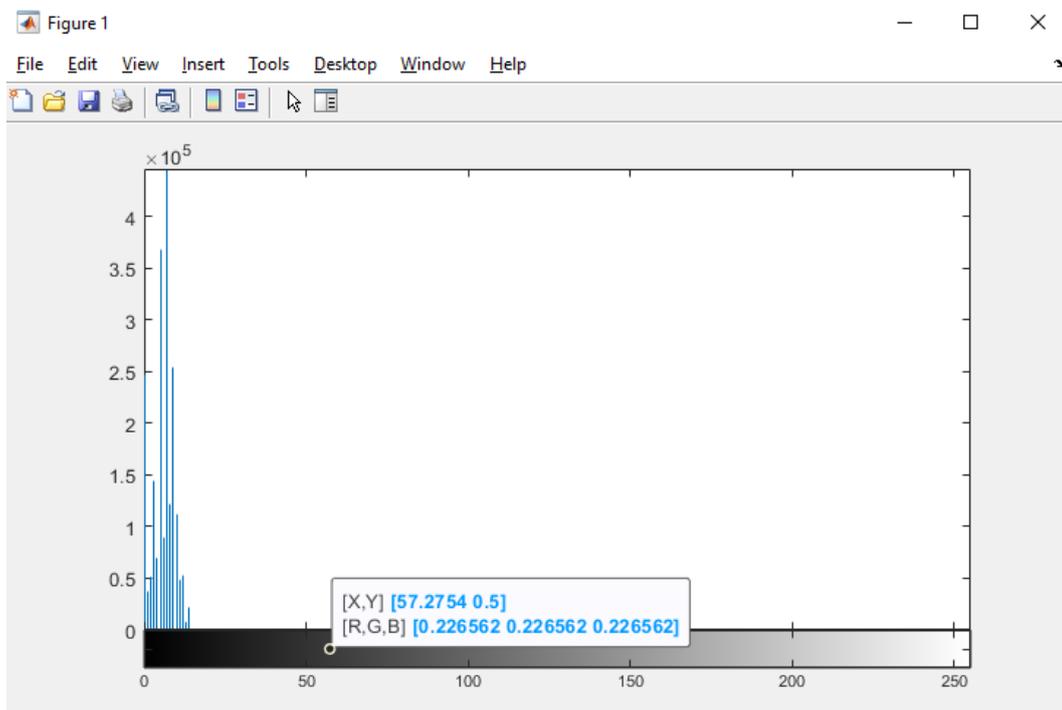


Figura 27: Histograma de la imagen

En la Figura 27 se muestra el histograma de la imagen, al ser una imagen en escala de grises contendrá 256 bins. Observando este histograma se puede obtener un umbral que permita separar el objeto que se quiere detectar del resto de la imagen. En este caso, se toma un valor de luminancia de 0,225 en el color rojo, para evitar posibles interferencias con objetos que puedan encontrarse cerca de ese umbral.

Una vez obtenido este valor para el umbral de la capa de color rojo se procederá a binarizar la imagen mediante el comando: **bw = imbinarize(img, 0.225)**. La función **imbinarize** [23] binariza una imagen 2D en escala de grises mediante el método de valor umbral. En este caso se binariza la imagen **img** a partir del umbral 0.225 y se almacena la imagen binarizada en la variable **bw**, imagen binaria de salida que corresponde a una matriz lógica del mismo tamaño que la imagen de entrada **img**, en la que se obtendrá un valor de 1 (blanco) para los píxeles que se correspondan con el color que se quiere detectar, en este caso rojo, y un valor de 0 para el resto de colores.

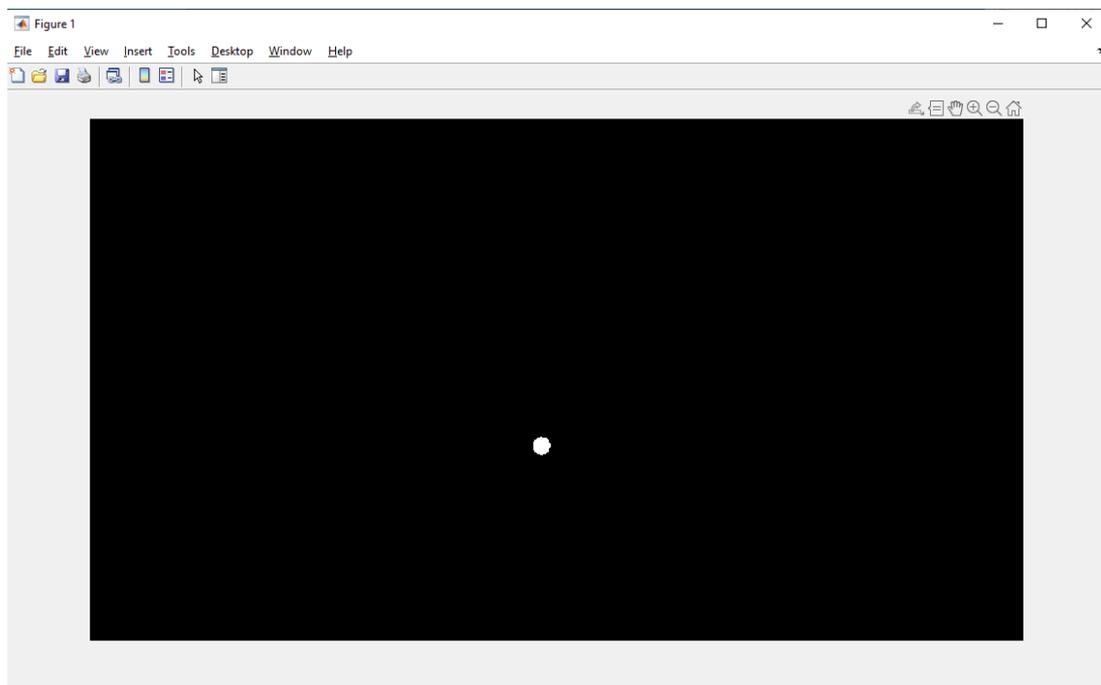


Figura 28: Imagen binarizada

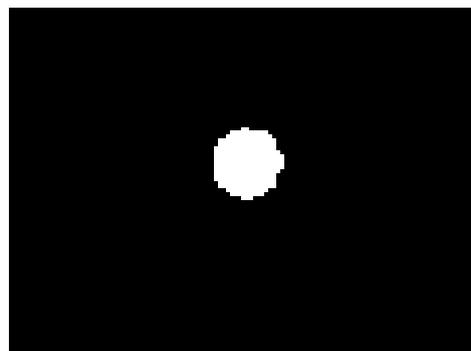


Figura 29: Detalle del punto tras binarizar la imagen

En la Figura 28 se observa la imagen completamente binarizada, de manera que se detecta el punto rojo en color blanco, almacenado como 1, completamente separado del resto de la imagen, que se almacena como 0 o negro. En la Figura 29, detalle ampliado del punto de la Figura 28 se puede observar que se detectan muchas irregularidades en los bordes del punto, pudiendo crearse también huecos dentro del mismo.

En el fragmento de código de la Figura 30 se solucionan estos problemas y se homogeneiza la imagen. Además se realiza la apertura morfológica de la imagen, se definen sus propiedades para hallar el centroide, se muestra el video imagen a imagen y se encuadra el punto con un cuadro verde mostrando sus coordenadas en cada imagen.

```
% Creamos un elemento estructurante en forma de disco de radio 1
SE = strel('disk',1);

% Realizamos una apertura morfológica de la imagen aplicandole el elemento morfológico
bw = imopen(bw,SE);

% Aplicar filtro para evitar ruidos en la imagen
bw = medfilt2(bw);

% Rellenar los huecos que contenga la imagen
bw = imfill(bw, "holes");

% Etiquetado de los objetos
[L,N]= bwlabel(bw);

% Función regionprops: medir propiedades de las regiones de imágenes (área,centroide...)
prop = regionprops(L);

% Mostrar la imagen
imshow(frame);

% Obtener el centroide del punto y almacenarlo en la matriz coords en cada iteración del bucle
%c=prop.Centroid;
%coords(i,:) = c;

for n=1:N
    c=prop(n).Centroid; % Obtener el |centroide
    rectangle('Position',prop(n).BoundingBox,'EdgeColor','g','LineWidth',2); % Encuadrar
    text(c(1),c(2),strcat('X:',num2str(c(1)),'\newline', ...
        'Y:',num2str(c(2))),'Color','green'); % Mostrar coordenadas
    % Rellenar cada columna i de la matriz coords con cada vector de posición del centroide
    coords(i,:) = c;
end
pause(0.0001);
i=i+1;
end
```

Figura 30: Fragmento de código donde se implementa la segunda parte del bucle para detectar un punto y mostrar sus coordenadas.

Una vez obtenida la imagen binarizada, **bw**, se crea un elemento estructurante morfológico: “**SE = strel('disk',1);** crea un elemento estructurante en forma de disco, donde 1 especifica el radio” [24] “Un objeto *strel* representa un elemento estructurante morfológicamente plano que es parte esencial de las operaciones morfológicas de dilatación y erosión” [24]

“Un elemento estructurante plano es un entorno que asume valores binarios, que puede ser 2D o multidimensional y en el que los píxeles con el valor true se incluyen en el cómputo morfológico, en tanto que los píxeles denominados con el valor false, no. El píxel central del elemento estructurante, llamado origen, identifica el píxel de la imagen que se está procesando” [24] La siguiente figura ilustra un elemento estructurante plano:

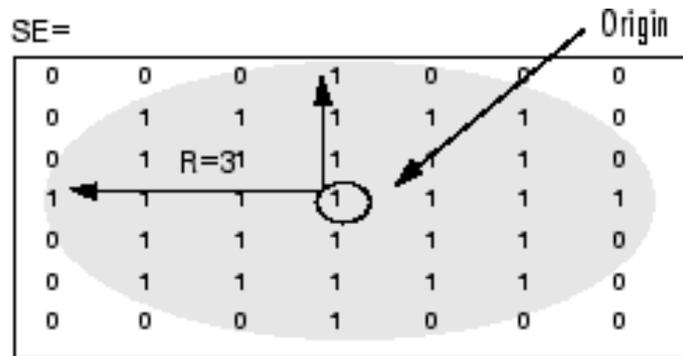


Figura 31 Elemento estructurante plano. [24]

A continuación se realiza una apertura morfológica en la imagen binaria **bw**, utilizando el elemento estructurante **SE** creado anteriormente y se almacena de nuevo en la misma variable **bw**, de forma que no sea necesario crear una nueva para cada operación y así ahorrar memoria.

“**bw = imopen(bw,SE)**: realiza una apertura morfológica en la imagen en escala de grises o binaria **bw** utilizando el elemento estructurante **SE**. La operación de apertura morfológica consiste en una erosión seguida de una dilatación utilizando el mismo elemento estructurante para ambas operaciones.” [25]

Seguidamente se añade un filtro de mediana para eliminar ruidos: “**bw = medfilt2(bw)**: realiza un filtrado de mediana de la imagen **bw** en dos dimensiones. Cada píxel de salida contiene el valor de la mediana de un entorno de 3 por 3 alrededor del píxel correspondiente de la imagen de entrada.” [26]

Una vez filtrada la imagen, utilizando la función **imfill**, se rellenan los huecos y regiones de la imagen y se vuelve a almacenar en la misma variable **bw**. “**bw = imfill(bw, “holes”)**; rellena huecos de la imagen binaria de entrada **BW**. En esta sintaxis, un hueco es un conjunto de píxeles de fondo que no se puede alcanzar rellenando el fondo desde el borde de la imagen.” [27]

En la Figura 32 se muestra el resultado final tras aplicar la apertura morfológica, el filtro y el rellenado de huecos, que se aprecia mejor en el detalle de la Figura 33, donde se puede ver el punto mejor definido, obviando las propias irregularidades del punto, ya que está dibujado a mano. Se pueden observar los bordes mejor definidos y en caso de que en algún fotograma se produjera algún hueco dentro del objeto, este habrá sido rellenado.

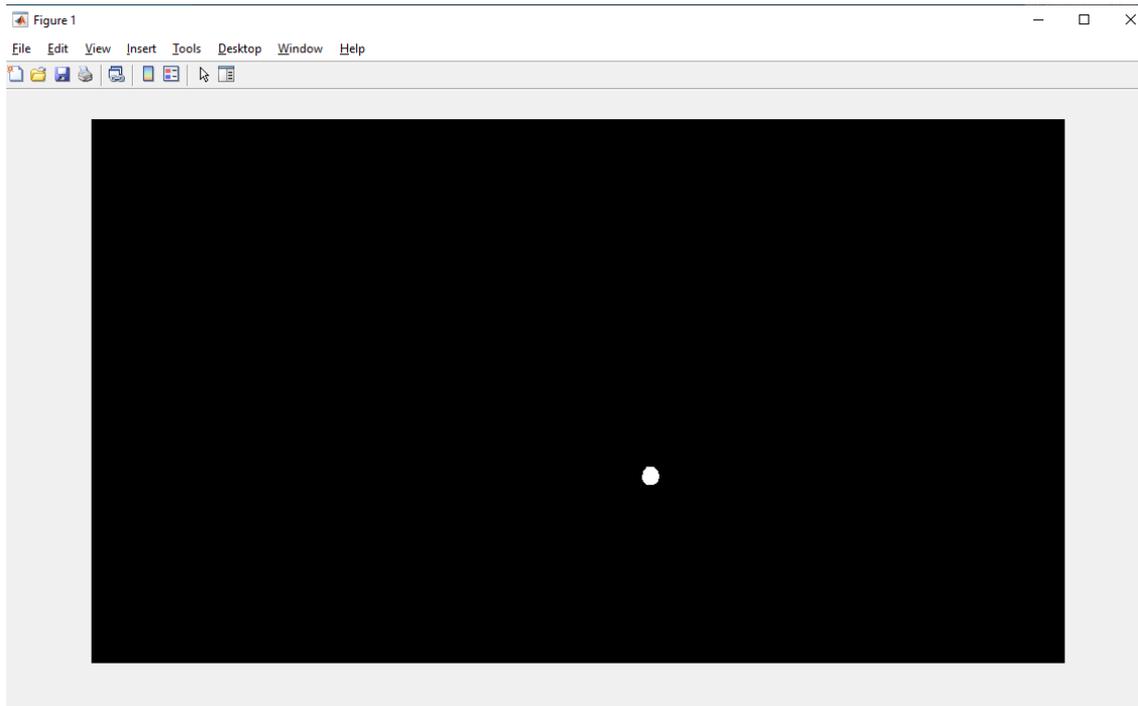


Figura 32: Imagen tras realizar la apertura morfológica, aplicar el filtro y rellenar huecos

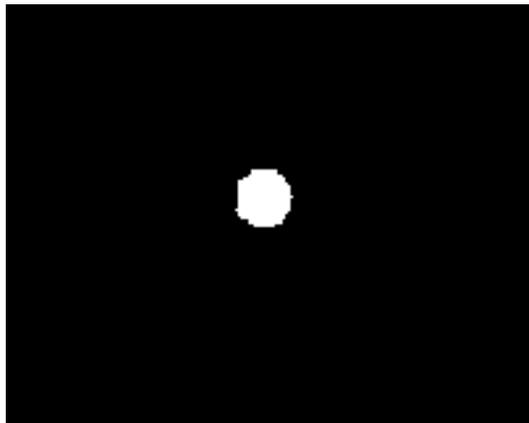


Figura 33: Detalle del punto tras realizar la apertura morfológica, aplicar el filtro y rellenar huecos

A continuación se procede al etiquetado de la imagen mediante la función **bwlabel**, función encargada de “etiquetar los componentes conectados de una imagen binaria 2D.” [26] “[L N]= **bwlabel**(bw); devuelve la matriz de etiquetas L, que contiene las etiquetas de los objetos conectados que se encuentran en BW (imagen binarizada 2D) y también devuelve N, el número de objetos conectados que se encuentran en BW.” [26]

Mediante este proceso se obtiene la matriz **L**, Figura 34, matriz con las mismas dimensiones que la imagen binarizada, **bw**, en la que se encuentra etiquetado cada pixel de la imagen, mostrando un 1 en los pixeles donde se encuentra algún elemento y 0 donde no. En este ejemplo la matriz L corresponde al último frame del primer vídeo registrado, estructura de un grado de libertad sin masas. Comparando este resultado

con las coordenadas del último frame obtenidas en la matriz **coords**, Figura 35, se observa que corresponden con el centroide del punto marcado con unos en la matriz **L**.

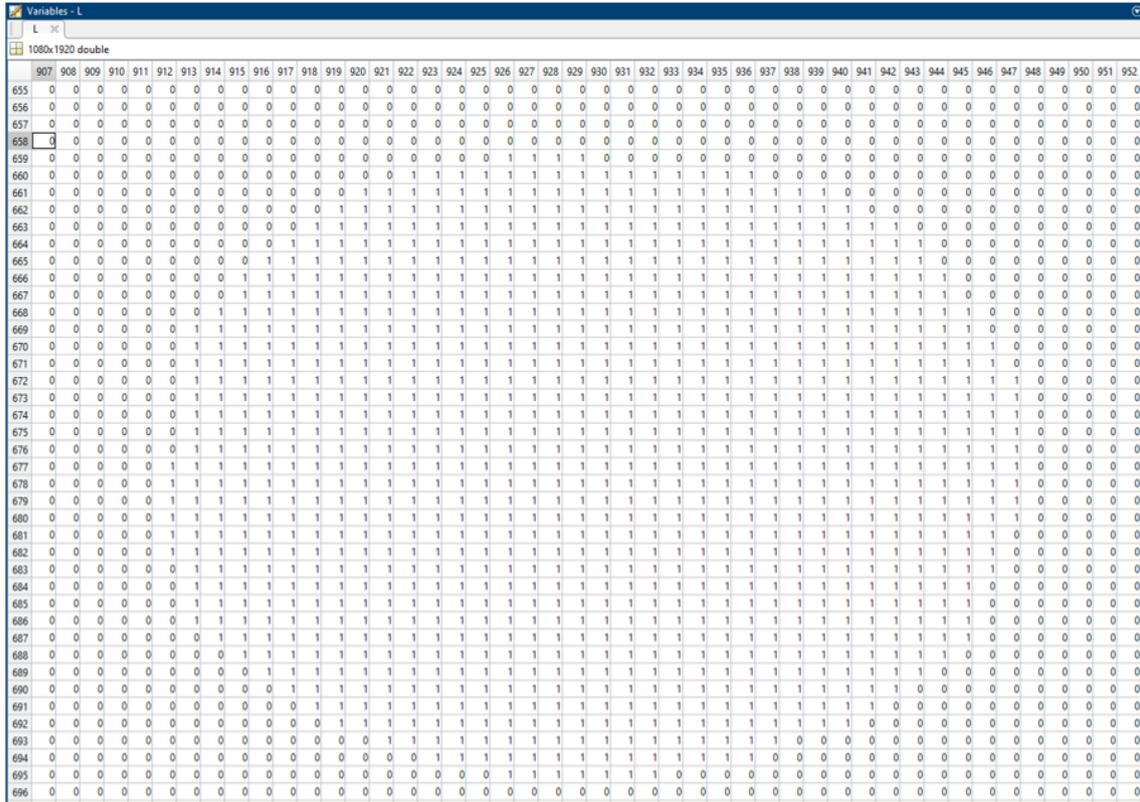


Figura 34: Matriz L, donde se observa el objeto etiquetado mediante unos

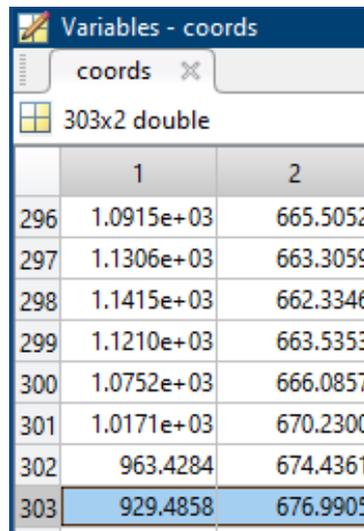


Figura 35: Coordenadas del centroide del último punto del primer video

En este punto, se implementa la función **regionprops**, que “mide propiedades como el área, los centroides y el cuadro delimitador de cada uno de los objetos (componentes conectados) de una imagen. **regionprops** es compatible con las regiones contiguas y no contiguas.” [29] En este caso: “**prop = regionprops(L)**; mide las propiedades para

cada región etiquetada en la imagen de etiqueta L.” [29] “la función devolverá las mediciones de “Area”, “Centroid” y “BoundingBox” [29]

Se muestra un ejemplo de las propiedades obtenidas mediante la variable **prop** del último frame del vídeo en la Figura 36. Se pueden obtener más propiedades de la imagen añadiendo el término **properties**: **prop = regionprops(L, properties)**, aunque en este caso no será necesario ya que solo se necesitan las propiedades del centroide y el **BoundingBox**.

Field	Value
Area	1056
Centroid	[929.4858,676.9905]
BoundingBox	[911.5000,658.5000,36,37]

Figura 36: Propiedades de la región etiquetada de la matriz L, obtenidas mediante la función *regionprops*

A continuación se implementa un bucle for, el cual, encuadra cada elemento detectado en cada imagen en un rectángulo verde y agrega a su lado las coordenadas del centroide, de manera que al transcurrir cada iteración del bucle while se observa la sucesión de cada uno de los fotogramas del video en tiempo real, encuadrando el punto en un rectángulo cuadro y mostrando sus coordenadas a la izquierda de este.

El fragmento de código marcado como comentario: **%c=prop.Centroid;** y **% coords(i,:)=c;** se implementa para abreviar el procedimiento y que no sea necesario ver transcurso de cada fotograma cada vez que se quiera analizar un video. Se almacena los valores del centroide (x,y) en la variable c y se va rellenando la matriz **coords** con cada vector c para cada iteración del bucle. Este fragmento sustituirá al bucle for previamente mencionado en caso de no querer ver el video completo.

Mediante el comando **imshow** [19] se mostrará cada fotograma por pantalla para cada iteración del bucle. Si como entrada se introduce la imagen binarizada **bw**: **imshow(bw)**, se mostrará el video con cada fotograma binarizado, Figura 37, o se puede ver el video original, en RGB, introduciendo como imagen de entrada la variable **frame**: **imshow(frame)**, de manera que reproducirá el video a color, observando el movimiento de la estructura, con el punto encuadrado y mostrando las coordenadas del centroide en cada frame, como se puede ver en la Figura 38.

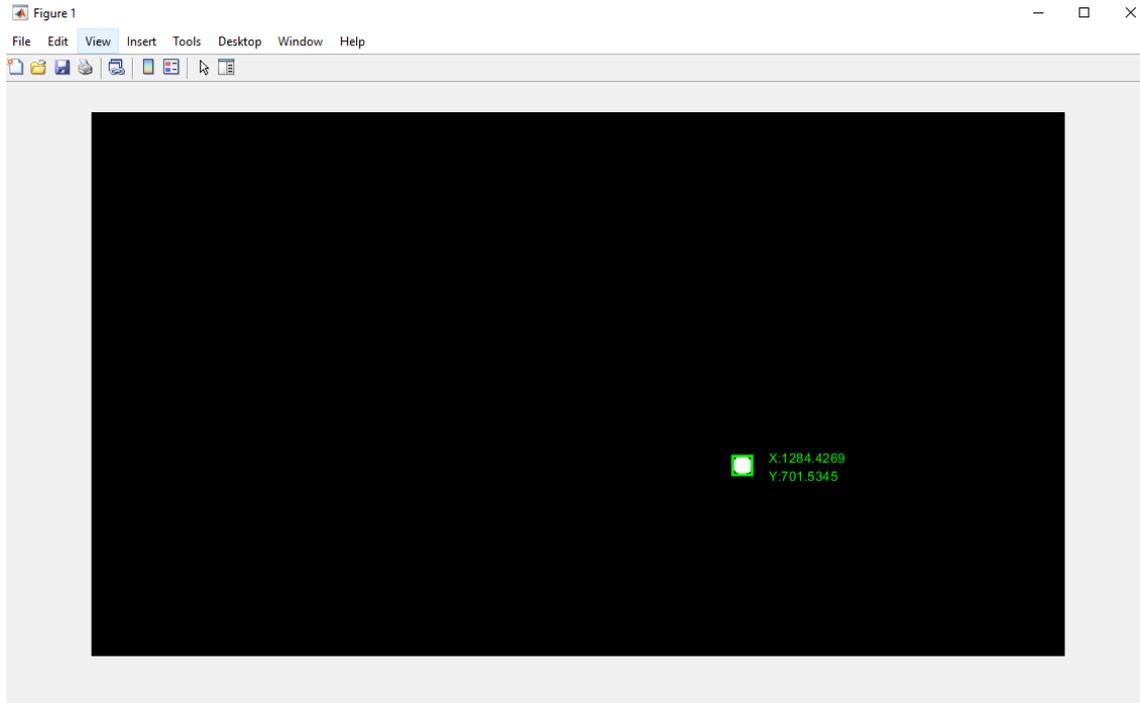


Figura 37: Resultado final, detectando las coordenadas del punto en cada fotograma del video, mostrando la imagen binarizada

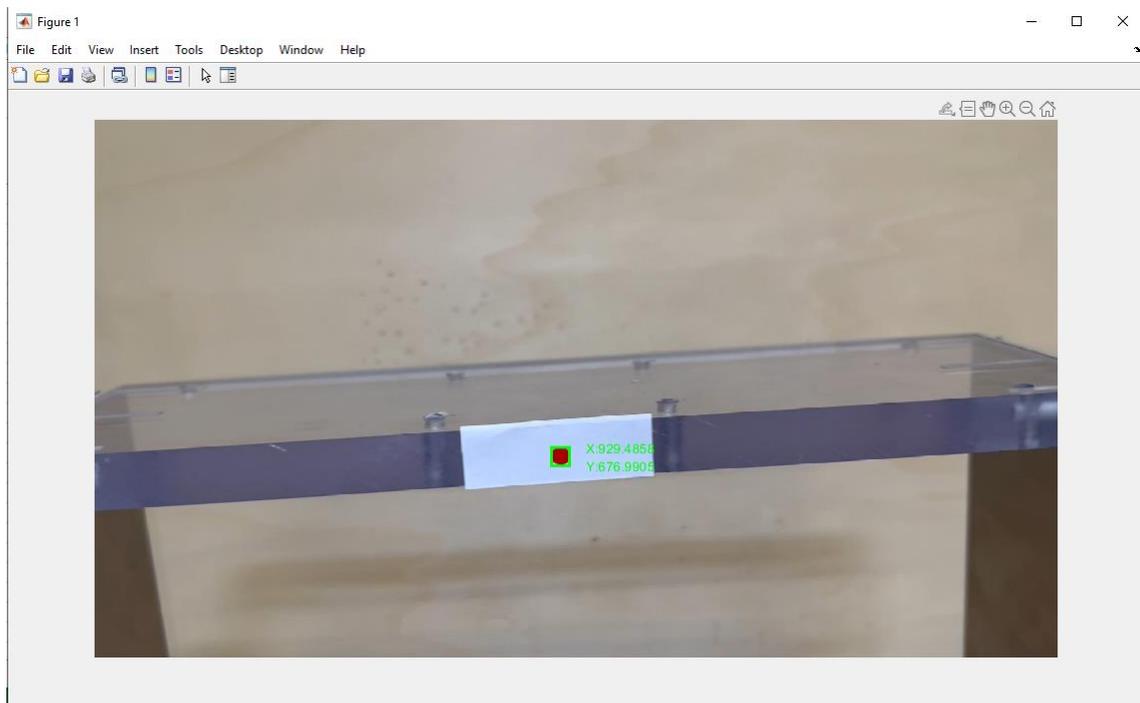


Figura 38: Resultado final, detectando las coordenadas del punto en cada fotograma del video, mostrando la imagen del video original

Por último se rellena cada columna i de la matriz **coords** con cada vector de posición del centroide (x e y): $\text{coords}(i, :) = \mathbf{c}$: la matriz **coords** previamente creada como una matriz de ceros, que se va relleno en cada iteración con el valor del centroide \mathbf{c}

(x,y). Se puede observar el comienzo de la matriz obtenida para el video del primer experimento en la Figura 39.

	1	2
1	1.0821e+03	739.2442
2	1.0819e+03	738.7044
3	1.0818e+03	738.1179
4	1.0815e+03	737.4071
5	1.0811e+03	736.7326
6	1.0805e+03	735.9510
7	1.0795e+03	735.4400
8	1.0789e+03	734.5453
9	1.0781e+03	733.8859
10	1.0778e+03	733.1150
11	1.0775e+03	732.3477
12	1.0774e+03	731.5527
13	1.0775e+03	730.7706
14	1.0772e+03	730.4845
15	1.0768e+03	729.9907
16	1.0762e+03	729.2808
17	1.0755e+03	728.3695
18	1.0747e+03	727.3485
19	1.0736e+03	726.5387
20	1.0729e+03	725.6229

Figura 39: Matriz de coordenadas del centroide del punto

En este punto, ya se ha obtenido la matriz de coordenadas del centroide del punto en cada frame del video, imprescindible para analizar el desplazamiento horizontal del punto en función del tiempo, así como de la estructura y poder compararlo con los valores obtenidos en el laboratorio mediante el sensor de desplazamiento láser.

4.2. Segunda parte: análisis de funciones

En esta segunda parte del código se escalan los datos de desplazamiento, ya que se encuentran en píxeles, se importan los datos de los desplazamientos obtenidos mediante el láser, se acondicionan ambos y finalmente se ajustan a una función para poder obtener sus parámetros modales. El primer fragmento de esta parte del código se muestra en la Figura 40.

ACONDICIONAMIENTO DE DATOS Y ANÁLISIS DE FUNCIONES

ESCALAR Y AJUSTAR LOS DATOS DE VISIÓN Y CREAR GRAFICO COMPLETO

```
% Escalar los datos obtenidos por visión de píxeles a metros mediante la relación m/pix

% Importar medida de la pegatina en píxeles del archivo excel
pegatinapix = readmatrix(file2,'sheet','Data1','range','E2');
pegatinapix = pegatinapix(1,1);

% Medida real de la pegatina en metros
pegatinam = 0.0485;

% Relación m/pix
relacion= pegatinam/pegatinapix;

% Separar las coordenadas x e y del centroide, escalarlas y eliminar la tendencia lineal
x = coords(:,1);
x = detrend(x,1);
x = x*relacion;
x = -x;

y = coords(:,2);
y = detrend(y,1);
y = y*relacion;

% Crear gráfico del desplazamiento horizontal en función del tiempo
figure(2)
plot(t,x)
%plot(t,y) % Para registros verticales
xlabel('t(s)');
ylabel('x(m)');
%ylabel('y(m)'); % Para registros verticales
title('x(t)');
%title('y(t)'); % Para registros verticales
grid on;
saveas(figure(2),'carlos1vision.jpg'); %Guardar figura como imagen
```

Figura 40: Fragmento de código que escala los datos de visión, los ajusta y crea un gráfico del desplazamiento en función del tiempo

Una vez obtenida la matriz de coordenadas se procederá a escalarla, conociendo las medidas en píxeles y en metros de un mismo elemento medible, en este caso la pegatina utilizada para marcar el punto rojo en la estructura.

Para poder medir la medida de la pegatina en píxeles en el video, se creará un nuevo script para leer el primer frame del video y obtener la medida de la pegatina para escalar los datos. Se crea un nuevo script para realizar la medida de cada caso más cómodamente sin que sea necesario esperar a que ejecute todo el código analizando todo el vídeo. Este código se puede ver en la Figura 41.

```
1
2 clear, clc
3
4 vid = VideoReader("carlos1.mp4");
5
6 frame1=readFrame(vid);
7
```

Figura 41: Código para medir la longitud de la pegatina, en píxeles, de cada video

Primero se importa el archivo de video mediante la función **videoReader** [15]. Se crea un objeto llamado **frame1** mediante la función **readFrame** [18] la cual lee el siguiente frame disponible de un video, que en este caso será el primer frame del video que se ha importado anteriormente (**vid**). Se utilizará esta imagen para poder medir la longitud de la pegatina.

Para poder escalar los datos obtenidos será necesario medir el objeto en la realidad y en cada vídeo para poder obtener la relación entre ellos. Primero se realiza la medición de la pegatina en tamaño real, en este caso tiene un largo de 0,0485 m, como se puede ver en la Figura 42.

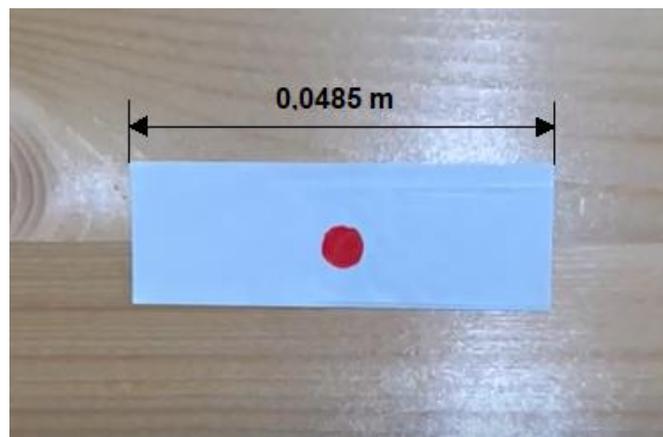


Figura 42: Medida real de la pegatina, en metros

A continuación, se debe medir el largo de la pegatina en píxeles en cada uno de los vídeos, debido a que la cámara ha podido ser posicionada a diferentes distancias para la realización de cada video. Para realizar esta medición es necesario el fragmento de código explicado previamente, **frame1 = readFrame(vid)**; el cual lee un fotograma del video (**vid**) y lo almacena en la variable **frame1**. Una vez almacenada esta imagen, se abre la aplicación de MATLAB **Image Viewer** [28], dentro del paquete **Image Processing Toolbox** [6], aplicación que sirve para visualizar explorar y medir imágenes, Figura 43.

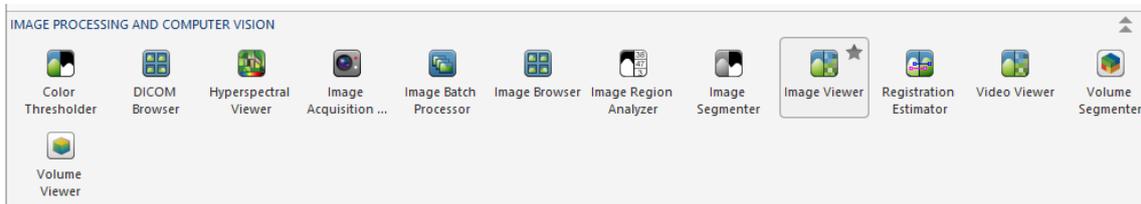


Figura 43: Acceso a la aplicación **Image Viewer**

Una vez abierta la aplicación se selecciona **Import from Workspace**, para importar la imagen del vídeo deseado, **frame1**, previamente guardada. Se puede ver el proceso en la Figura 44 y Figura 45

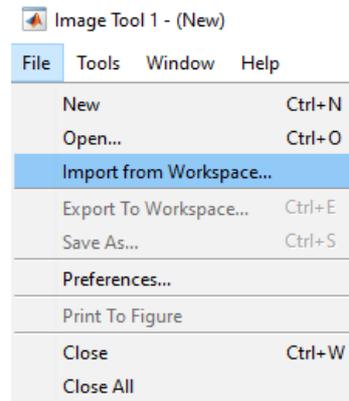


Figura 44: Importar la imagen

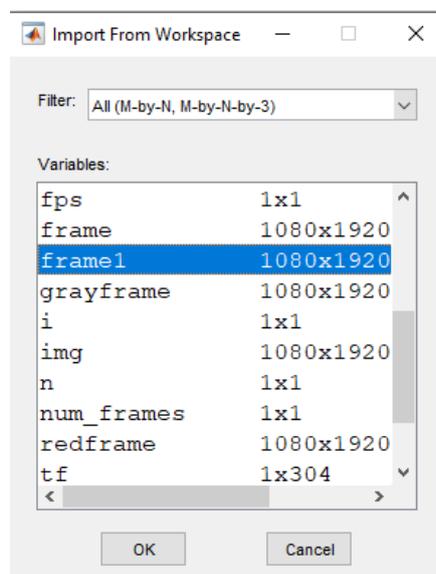


Figura 45: Importar la imagen

Una vez importada la imagen, se cargará en una ventana como la de la Figura 46 en la que se pueden realizar diversas operaciones como inspeccionar el valor RGB de los píxeles, obtener información de la imagen, ajustar el contraste y la luminosidad, recortarla o medir distancias, que en este caso es la herramienta que se utilizará.

Mediante la herramienta de medición de distancias, marcada en rojo en la Figura 46 se mide el largo de la pegatina, y en este caso, para el primer vídeo (1p0m), se obtiene una medida de 377,83 píxeles.

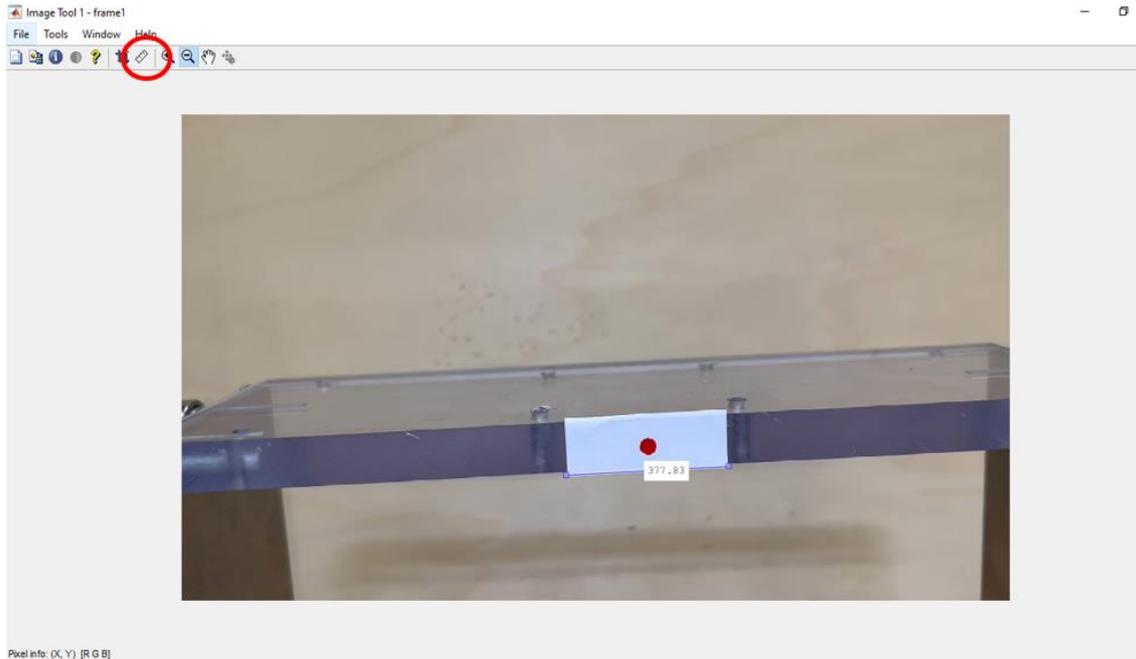


Figura 46: Medida de la pegatina en píxeles

Aplicando una simple proporción, se puede averiguar la relación entre las medidas en píxeles y las medidas en metros, donde D corresponde al desplazamiento obtenido mediante el análisis por visión y LP corresponde a la longitud de la pegatina:

$$\frac{D (pixel)}{LP(pixel)} = \frac{D (m)}{LP (m)} \quad (8)$$

Por lo tanto, la relación de conversión entre los datos del desplazamiento en metros y los datos en píxeles es la siguiente:

$$D(m) = \frac{LP(m)}{LP(pixel)} * D(pixel) \quad (9)$$

Así, una vez obtenidos los datos de los desplazamientos en píxeles mediante visión, se deben multiplicar por $\frac{LP(m)}{LP(pixel)}$ para escalarlos y transformarlos a metros, unidad en la que se encuentran los datos de desplazamiento obtenidos por el láser.

Continuando con el fragmento de la Figura 40, se comienza importando los datos de la medida de la pegatina en píxeles, almacenados previamente en el archivo de Excel correspondiente a cada video, mediante la función **readmatrix** [31] y se selecciona únicamente la primera fila y columna de la matriz, donde se encuentra la medida. Después se describe la medida real de la pegatina en metros, en este caso 0.0485 m.

Se obtienen los vectores de coordenadas x e y , separando la primera y la segunda columna de la matriz de coordenadas, se elimina la tendencia lineal mediante la función **detrend** [32] y se multiplica por la relación obtenida anteriormente, que podrá variar en cada caso, para escalar los datos a metros.

Por último se crea un gráfico de dispersión del desplazamiento horizontal, o vertical para los registros verticales, en función del tiempo, vector obtenido al inicio, y se guarda en formato de imagen con su nombre correspondiente en cada caso, mediante la función **saveas** [33]. En la Figura 47 se observa el gráfico desplazamiento horizontal en metros frente al tiempo en segundos, pudiéndose observar el impulso generado y después la función amortiguada de la estructura.

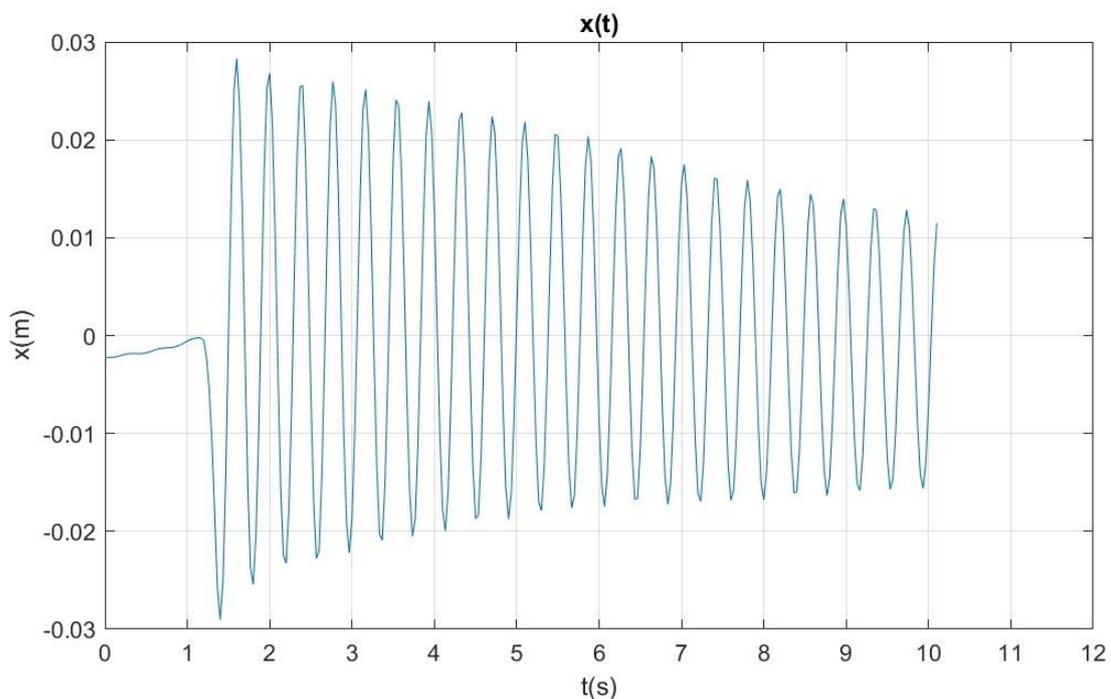


Figura 47: Gráfico del desplazamiento en función del tiempo obtenido mediante visión

A continuación se realiza el mismo procedimiento de importación y ajuste para los datos de desplazamientos obtenidos mediante el láser. Se muestra en el fragmento de código de la Figura 48.

IMPORTAR Y AJUSTAR DATOS LASER Y CREAR GRÁFICO COMPLETO

```
% Importar la matriz de datos obtenidos por el laser, almacenados en excel
laser = readmatrix(file2,'Sheet','Data1','ExpectedNumVariables',2);

% Vector de tiempos del laser
t1 = laser(:,1);

% Vector de desplazamientos del laser
desplaz = laser(:,2);

% Eliminar tendencia lineal de la función
desplaz = detrend(desplaz,1);

% Crear gráfico del desplazamiento en función del tiempo de los datos obtenidos con el laser
figure(3)
plot(t1,desplaz)
xlabel('t(s)');
ylabel('x(m)');
% Para registros verticales
% ylabel('y(m)');
title('x(t)');
% title('y(t)'); % Para registros verticales
grid on;
saveas(figure(3),'carlos1laser.jpg');
```

Figura 48: Fragmento de código que importa los datos del láser, los ajusta y crea un gráfico del desplazamiento en función del tiempo

Primero se importa el archivo que contiene los datos, se separan el vector de tiempos, $t1$ (láser) y el de desplazamientos, **desplaz**, y se le realiza a este una eliminación de la tendencia lineal.

Para finalizar, se crea el gráfico de desplazamiento en función del tiempo y se guarda como imagen con el nombre correspondiente, Figura 49. Se observa que la función de estudio comienza aproximadamente a los 5 segundos, por lo tanto se deberá acotar, eliminando la primera parte del registro, para obtener sus parámetros modales.

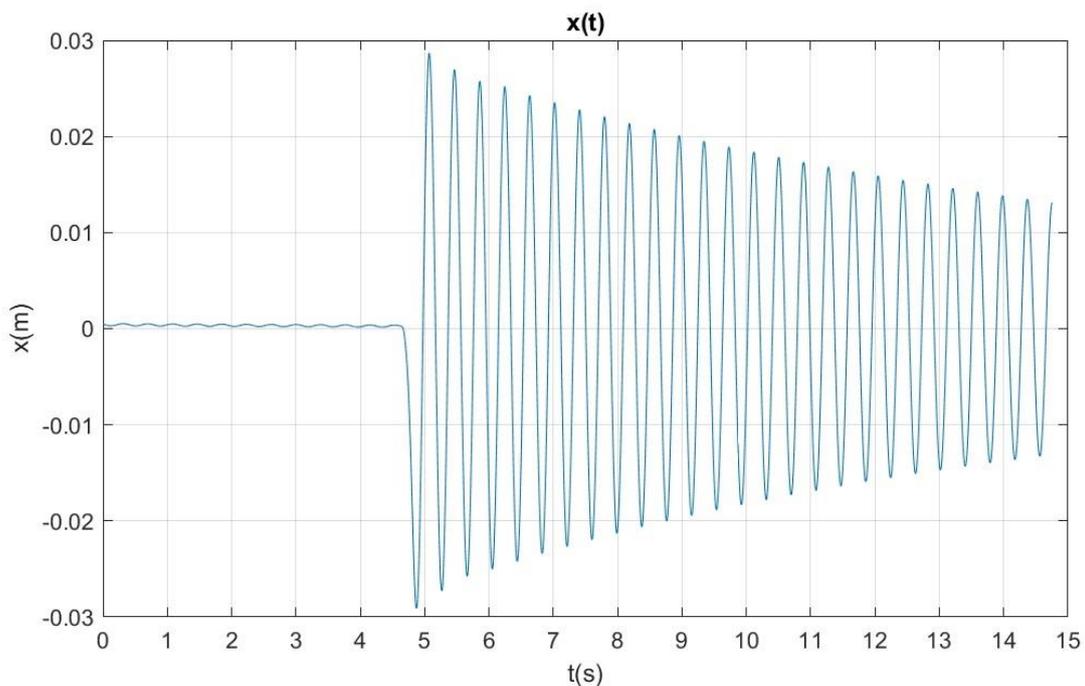


Figura 49 Gráfico del desplazamiento en función del tiempo obtenido mediante el sensor láser

En los fragmentos de la Figura 50 y Figura 51 se procede al acondicionamiento de los datos obtenidos mediante visión y por el láser respectivamente, lo que implica eliminar la parte inicial de la curva hasta que se produce el impulso, que no corresponde con la función del movimiento vibratorio de la estructura, obteniendo así una curva que nos permita ajustarla a una función de movimiento de la misma.

ACONDICIONAMIENTO DE LOS DATOS VISIÓN Y GRÁFICO DELIMITADO

```
% Se selecciona la parte del registro que se quiere ajustar, y se definen
% los límites temporales
tini = 2; % s
tfin = 10; % s

% Se define la frecuencia de muestreo
Dt = t(2)-t(1);

% Se acotan los vectores de tiempo y amplitud
tmax = tfin-tini;
tie = [0:Dt:tmax]';

ampl = x(t >= tini & t <= tfin);
%ampl = y(t >= tini & t <= tfin); %Para registros verticales

% Eliminar tendencia lineal de nuevo
ampl = detrend(ampl,1);

% Se crea el gráfico del registro seleccionado
figure(4)
plot(tie,ampl)
xlabel('t(s)');
ylabel('x(m)');
%ylabel('x(m)'); % Para registros verticales
title('x(t)');
%title('y(t)'); % Para registros verticales
grid on;
saveas(figure(4),'carlosivisionAcondicionado.jpg');
```

Figura 50: Fragmento de código mediante el que se acondicionan los datos obtenidos por visión y se crea el correspondiente gráfico

ACONDICIONAMIENTO DE LOS DATOS LÁSER Y GRÁFICO DELIMITADO

```
% (Del mismo modo que para visión, diferenciando las variables con una L)

% Se selecciona la parte del registro que se quiere ajustar, y se definen
% los límites temporales
tinil = 5; % s
tfinl = 13; % s

% Se define la frecuencia de muestreo
Dt1 = t1(2)-t1(1);

% Se acotan los vectores de tiempo y amplitud
tmax1 = tfinl-tinil;
tiel = [0:Dt1:tmax1]';

amp11 = desplaz(t1 >= tinil & t1 <= tfinl);

% Eliminar la tendencia lineal de nuevo
amp11 = detrend(amp11,1);

% Se crea el gráfico del registro seleccionado
figure(5)
plot(tiel,amp11)
xlabel('t(s)');
ylabel('x(m)');
%ylabel('x(m)'); % Para registros verticales
title('x(t)');
%title('y(t)'); % Para registros verticales
grid on;
saveas(figure(5), 'carlos1laserAcondicionado.jpg');
```

Figura 51: Fragmento de código mediante el que se acondicionan los datos obtenidos por el láser y se crea el correspondiente gráfico

Para acondicionar los datos, el procedimiento seguido para ambos métodos consiste primero en definir un tiempo inicial y un tiempo final, de manera que se acota la curva que se debe analizar. Para determinar cuál son estos instantes de tiempo, se observan los gráficos obtenidos previamente por ambos métodos, Figura 47 y Figura 49, y se decide en que instante comenzar. Estos límites variarán para cada video registrado dependiendo del tiempo transcurrido desde que se comienza a grabar hasta que se genera el impulso en la estructura y aparece la función amortiguada. Se puede observar el resultado del acondicionamiento de las funciones de visión y del láser en la Figura 52 y Figura 53 respectivamente.

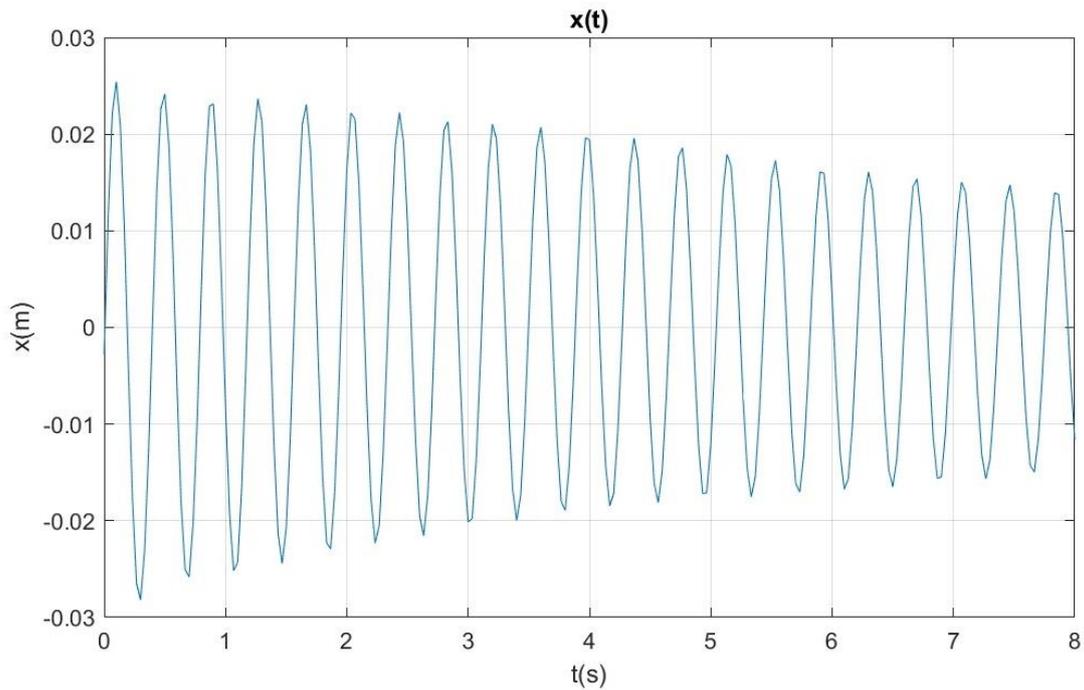


Figura 52: Gráfico acondicionado del desplazamiento en función del tiempo obtenido mediante visión

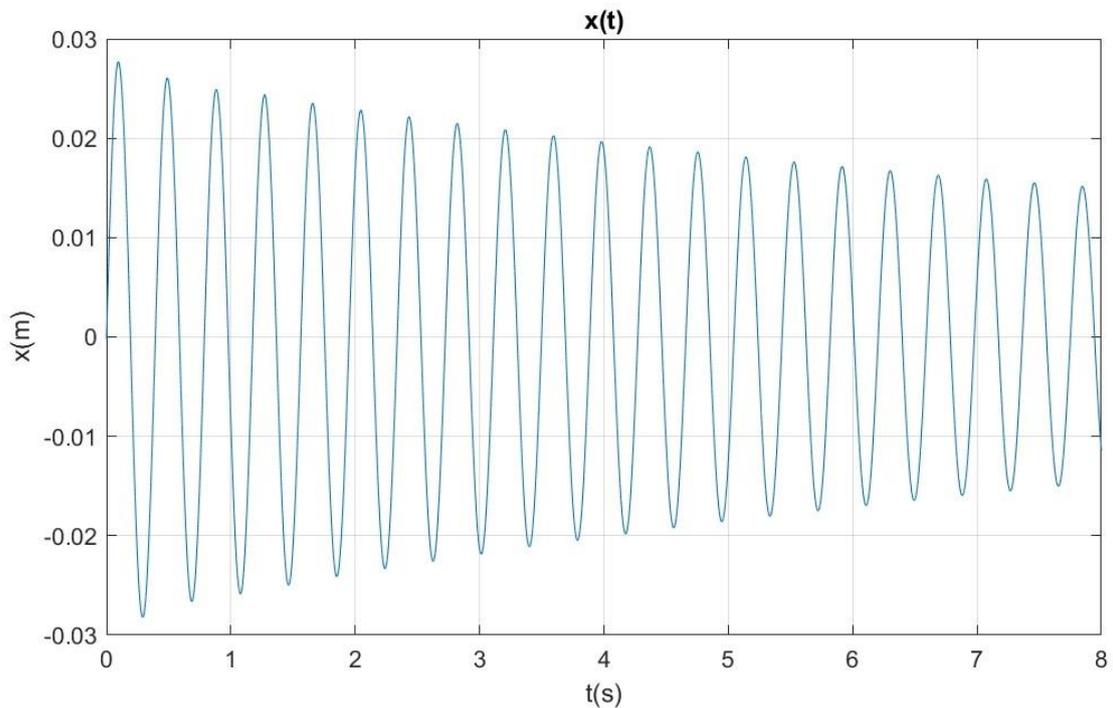


Figura 53: Gráfico acondicionado del desplazamiento en función del tiempo obtenido mediante el sensor láser

Finalmente, se superponen las dos funciones en un solo gráfico. Se puede observar el código en la Figura 54 y el gráfico en la Figura 55. La función de trazos azules corresponde al desplazamiento obtenido mediante el análisis por visión, mientras que la roja corresponde al obtenido por el láser. Se observa que las funciones en general coinciden, exceptuando el posible error humano a la hora de registrar los vídeos, en la

medición al escalar la función de visión, o las imprecisiones en la detección del punto debido a la diferencia de las frecuencias de muestreo entre ambos, siendo $1/30$ la obtenida del vídeo y $1/200$ la del láser.

SUPERPONER GRÁFICOS DE VISIÓN Y LÁSER

```
figure(6)
p = plot(tie,ampl,tiel,ampl1);
p(1).Linewidth=2;
p(2).Linewidth=1;
p(1).LineStyle='--';
xlabel('t(s)');
ylabel('x(m)');
%ylabel('y(m)'); % Para registros verticales
title('x(t) vision vs laser');
%title('y(t)'); % Para registros verticales
grid on;
saveas(figure(6),'carlosivisionvslaser.jpg');
```

Figura 54: Fragmento de código de donde se superponen los gráficos de visión y láser

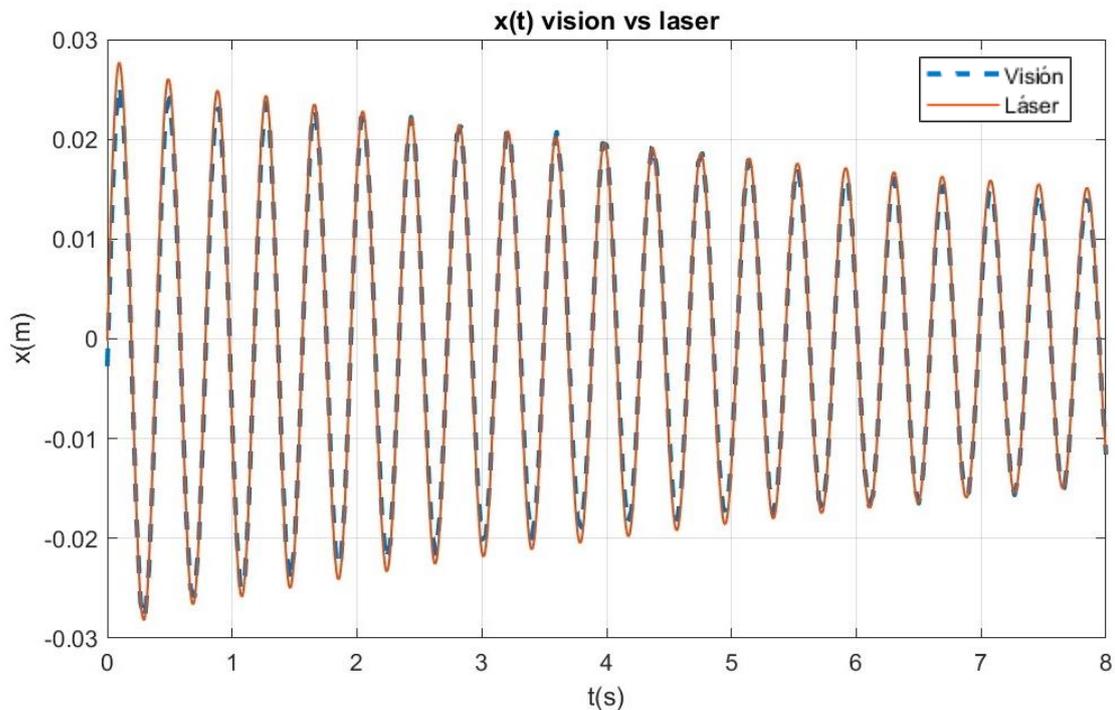


Figura 55: Superposición de las gráficas obtenidas mediante visión y láser

Por último, solo queda llamar a las funciones que ajustan las curvas a una función, Figura 56, previamente creadas mediante la aplicación **Curve Fitter** [34], aplicación de MATLAB para ajustar curvas y superficies a los datos obtenidos. A continuación, se explica el procedimiento seguido para ajustar estas curvas.

LLAMAR FUNCIÓN DE AJUSTE VISIÓN

```
[fitresultvision, gofvision] = carlosivision(tie , ampl)  
saveas(figure(7), 'carlosivisionAjuste.jpg');
```

LLAMAR FUNCIÓN DE AJUSTE LÁSER

```
[fitresultlaser, goflaser] = carlosilaser(tiel , ampl1)  
saveas(figure(8), 'carlosilaserAjuste.jpg');
```

Figura 56: Fragmento de código que llama a las funciones de ajuste de las curvas de visión y láser

Una vez acondicionadas ambas curvas se buscará una función que se ajuste a ellas mediante la aplicación **Curve Fitter** [34], para poder obtener sus parámetros modales. Se comienza iniciando la aplicación, Figura 57. También se puede acceder a ella escribiendo el comando **cftool** en el Command Window.

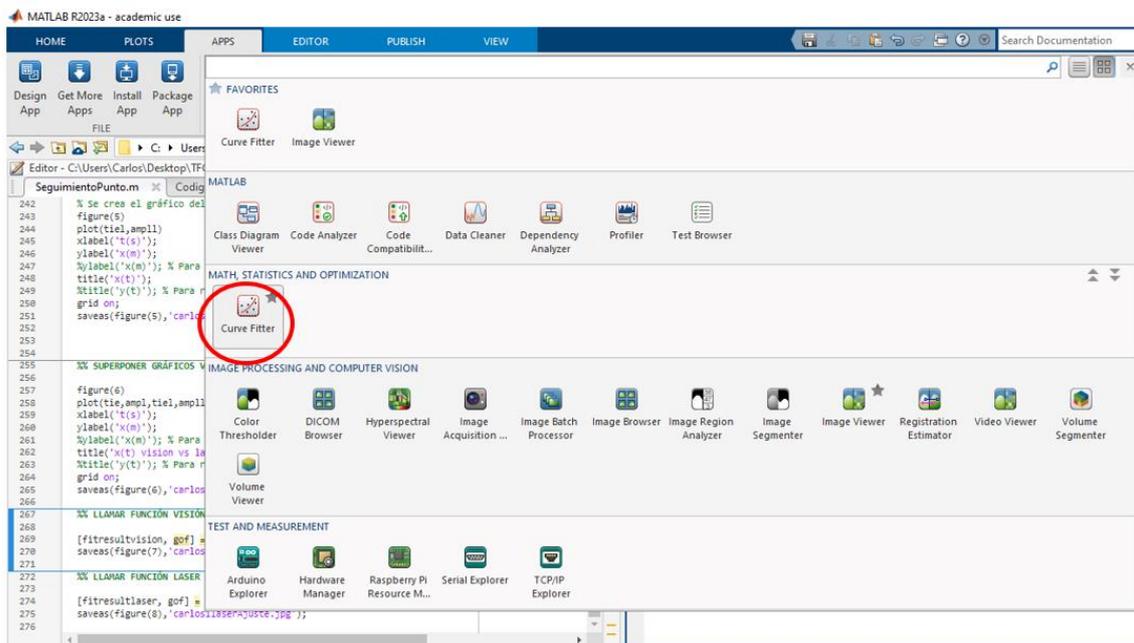


Figura 57: Acceso a la aplicación Curve Fitter

Una vez abierta la aplicación, se seleccionan los datos que se quieren ajustar en la pestaña **Select Data**, en este caso el eje x se corresponde con el tiempo **tie** y el eje y con la amplitud **ampl** (desplazamiento), y se nombra el ajuste según corresponda en cada caso.

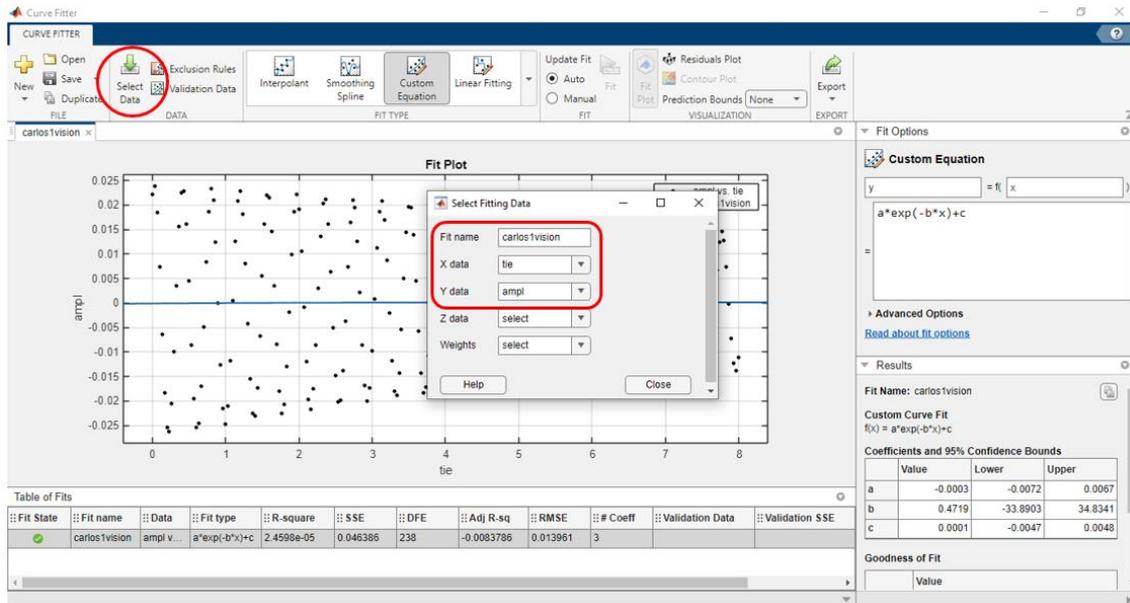


Figura 58: Selección de los datos a ajustar en la aplicación Curve Fitter

A continuación se selecciona el tipo de ajuste a realizar (Figura 58), en este caso **Custom equation**, ya que se debe ajustar a la ecuación de un sistema libre subamortiguado de un grado de libertad, para este ejemplo, dos o tres grados de libertad según corresponda en otros casos. Para estructuras de un grado de libertad, se puede reescribe la ecuación (4) de modo que resulte sencillo introducirla en el programa, se escribirá de la forma mostrada en la Figura 59.

$$x = f(t) = A \cdot \exp(-Z \cdot 2 \cdot \pi \cdot w \cdot t) \cdot \sin(2 \cdot \pi \cdot w \cdot t + f_i)$$

Figura 59: Desplazamiento en función del tiempo de un sistema libre amortiguado de 1 gdl, frecuencia en Hz

Donde los términos se corresponden con:

- x: desplazamiento horizontal del punto (m)
- A: amplitud (m)
- Z: coeficiente de amortiguamiento
- w: frecuencia (Hz)
- fi: desfase en radianes
- t: tiempo (s)
- 2*pi: término de conversión de la frecuencia de rad/s a Hz

Para los casos con más de un grado de libertad, la ecuación a ajustar se obtendrá mediante el método de superposición, obteniendo las siguientes ecuaciones a partir de las ecuaciones (5) y (6) respectivamente:

Para 2 grados de libertad, Figura 60:

$$x = f(t) = A1 \cdot \exp(-Z1 \cdot 2 \cdot \pi \cdot w1 \cdot t) \cdot \sin(2 \cdot \pi \cdot w1 \cdot t + fi1) + A2 \cdot \exp(-Z2 \cdot 2 \cdot \pi \cdot w2 \cdot t) \cdot \sin(2 \cdot \pi \cdot w2 \cdot t + fi2)$$

Figura 60: Desplazamiento en función del tiempo de un sistema libre amortiguado de 2 gdl, frecuencia en Hz

Para 3 grados de libertad, Figura 61:

$$x = f(t) = A1 \cdot \exp(-Z1 \cdot 2 \cdot \pi \cdot w1 \cdot t) \cdot \sin(2 \cdot \pi \cdot w1 \cdot t + fi1) + A2 \cdot \exp(-Z2 \cdot 2 \cdot \pi \cdot w2 \cdot t) \cdot \sin(2 \cdot \pi \cdot w2 \cdot t + fi2) + A3 \cdot \exp(-Z3 \cdot 2 \cdot \pi \cdot w3 \cdot t) \cdot \sin(2 \cdot \pi \cdot w3 \cdot t + fi3)$$

Figura 61: Desplazamiento en función del tiempo de un sistema libre amortiguado de 3 gdl, frecuencia en Hz

Una vez introducida la ecuación a la que se debe ajustar cada curva, en algunos casos la aplicación ajusta automáticamente la curva a la función deseada, mostrando los resultados de todas las constantes implicadas en la ecuación: A, Z, fi y w, como se puede ver en el gráfico y en la tabla inferior derecha, señalados en verde en la Figura 62. En otras ocasiones, especialmente en funciones con más de un grado de libertad, no se ajusta de manera automática y es necesario introducir restricciones a los valores iniciales y los límites inferiores y superiores de estos parámetros para conseguir que la función ajuste correctamente a la curva deseada, en la tabla señalada en rojo en la Figura 63.



Figura 62: Resultados obtenidos tras el ajuste a la función deseada

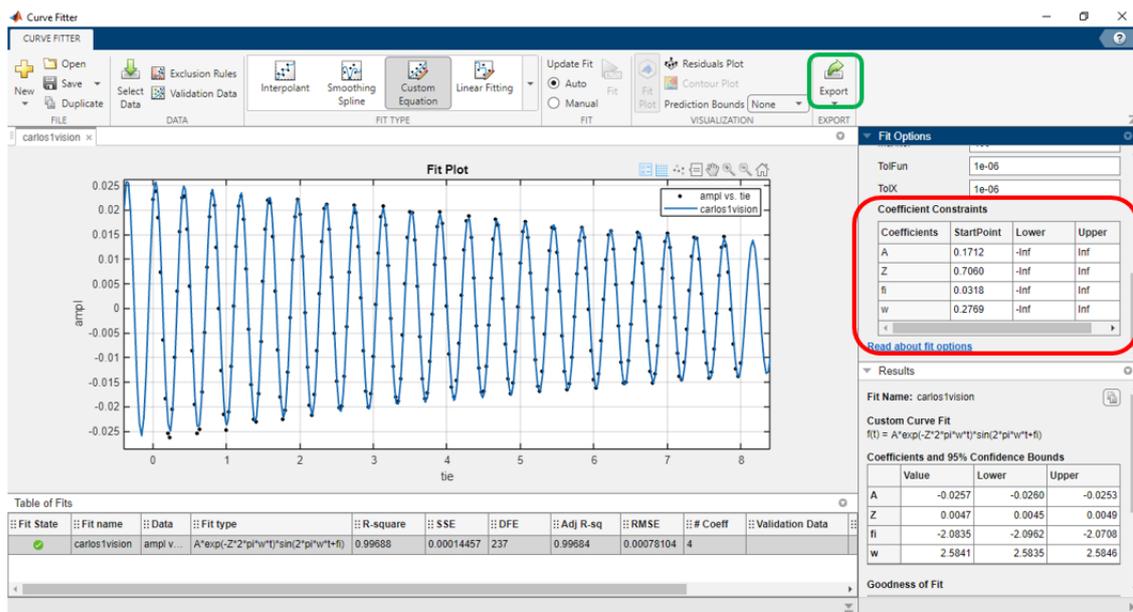


Figura 63: Restricciones a los parámetros de la función para su ajuste

Una vez obtenido el ajuste de la función, se exporta como código en la pestaña **Export**, señalada en verde en la Figura 63 y se guarda con el nombre correspondiente, Figura 64, de esta manera, se obtiene la función que se ejecutará al ser llamada por el fragmento de código mencionado anteriormente en la Figura 56.

```
1 function [fitresultvision, gof] = carlos1vision(tie, ampl) %tini=2 tfin=10
2 %CREATEFIT(TIE,AMPL)
3 % Create a fit.
4 %
5 % Data for 'carlos1' fit:
6 %   X Input: tie
7 %   Y Output: ampl
8 % Output:
9 %   fitresult : a fit object representing the fit.
10 %   gof : structure with goodness-of fit info.
11 %
12 % See also FIT, CFIT, SFIT.
13
14 % Auto-generated by MATLAB on 30-Aug-2023 18:16:27
15
16 %% Fit: 'carlos1'.
17 [xData, yData] = prepareCurveData( tie, ampl );
18
19 % Set up fittype and options.
20 ft = fittype( 'A*exp(-Z*2*pi*w*t)*sin(2*pi*w*t+fi)', 'independent', 't', 'dependent', 'x' );
21 opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
22 opts.Display = 'Off';
23 opts.StartPoint = [0.678735154857773 0.757740130578333 0.743132468124916 0.392227019534168];
24
25 % Fit model to data.
26 [fitresultvision, gof] = fit( xData, yData, ft, opts );
27
28 % Plot fit with data.
29 figure(7);
30 h = plot( fitresultvision, xData, yData );
31 legend( h, 'ampl vs. tie', 'carlos1', 'Location', 'NorthEast', 'Interpreter', 'none' );
32 % Label axes
33 xlabel( 'tie', 'Interpreter', 'none' );
34 ylabel( 'ampl', 'Interpreter', 'none' );
35 grid on
```

Figura 64: Función que ajusta los datos del sistema de 1 gdl obtenidos mediante visión

Una vez ejecutado el fragmento de código correspondiente al ajuste de la función de visión, se obtendrán los resultados mostrados en la Figura 65 y Figura 66. En el gráfico se pueden apreciar en azul los puntos obtenidos mediante la detección por visión, tras procesar la posición del punto en cada fotograma, mientras que en rojo se muestra la función que se ha conseguido para ajustar la curva en azul.

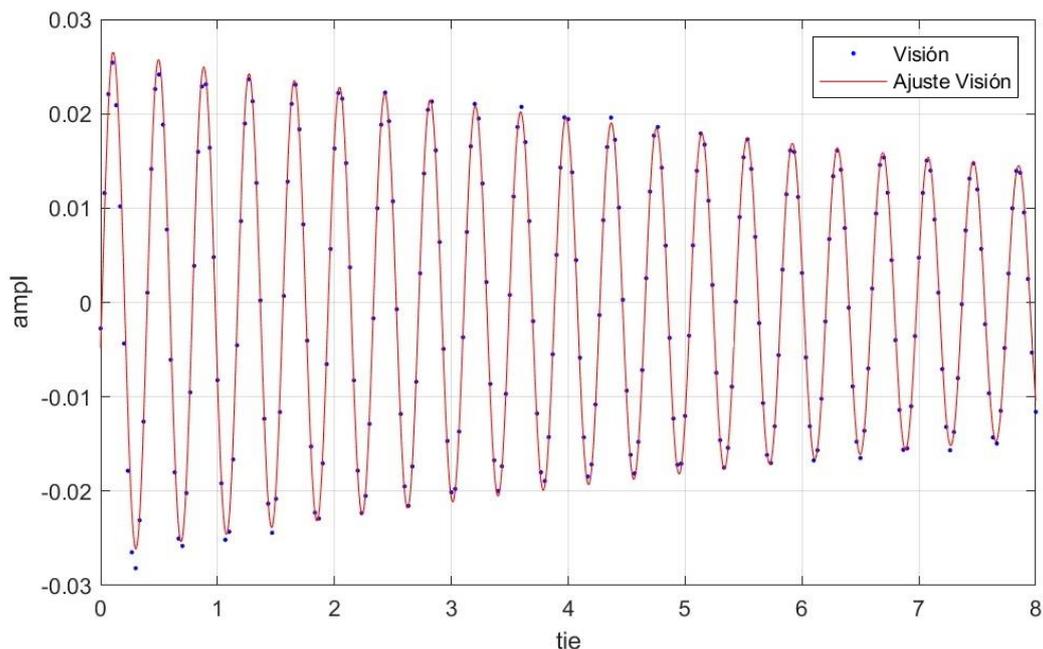


Figura 65: Gráfico del ajuste del sistema de 1 gdl obtenido mediante visión

En la Figura 66 se observan los resultados de la frecuencia, coeficiente de amortiguamiento, amplitud y desfase obtenidos a partir de esta función, además de diferentes parámetros que evalúan la bondad del ajuste, como por ejemplo el R^2 , que adquiere un valor de 0,9968, valor bastante bueno, lo que significa que la función obtenida se ajusta correctamente a la curva.

```
fitresultvision =  
  
General model:  
fitresultvision(t) = A*exp(-Z*2*pi*w*t)*sin(2*pi*w*t+fi)  
Coefficients (with 95% confidence bounds):  
A = 0.02677 (0.02631, 0.02724)  
Z = 0.004802 (0.004526, 0.005078)  
fi = -0.1794 (-0.1965, -0.1624)  
w = 2.582 (2.581, 2.583)  
  
gofvision =  
  
struct with fields:  
  
sse: 2.9520e-04  
rsquare: 0.9940  
dfe: 237  
adjrsquare: 0.9939  
rmse: 0.0011
```

Figura 66: Resultados del ajuste del sistema de 1 gdl obtenidos mediante visión

Si se realiza el mismo procedimiento con los datos obtenidos por el láser para el primer experimento, se obtendrá un ajuste similar el anterior. Como se puede ver en la Figura 67, el ajuste es más preciso, ya que la frecuencia de adquisición de datos del láser es mucho mayor que la de visión, siendo 200 datos/segundo la del este mientras que la de visión 30 datos/segundo, ya que el video está grabado a 30 fps. Aunque este ajuste sea más preciso visualmente, se puede comprobar que los resultados obtenidos por el láser, mostrados en la Figura 68, se asemejan bastante a los obtenidos anteriormente mediante el análisis por visión.

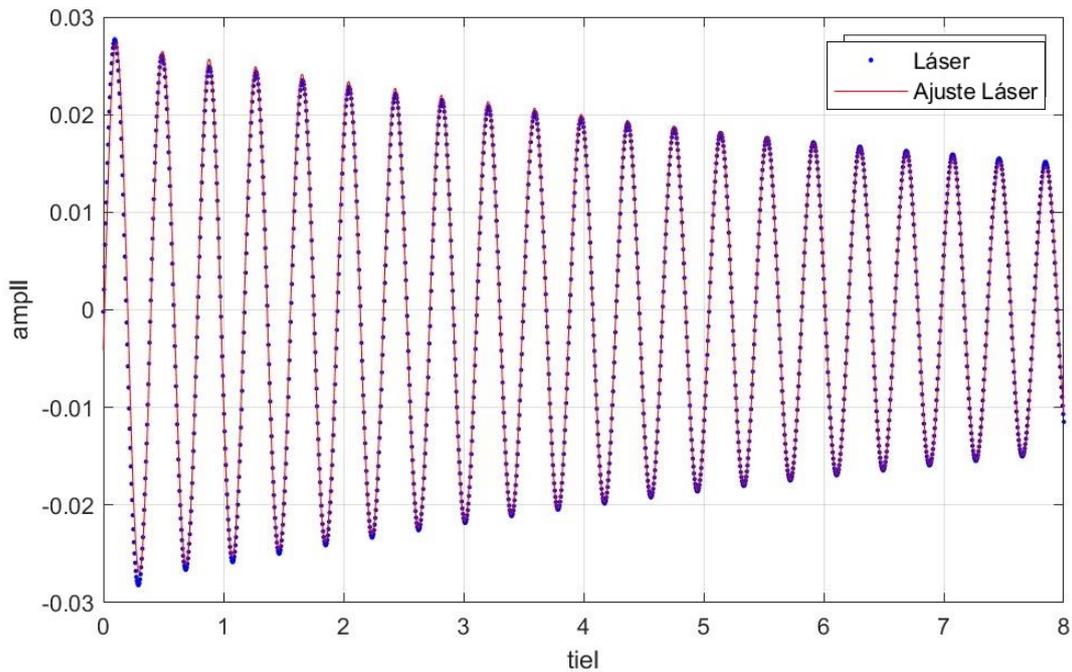


Figura 67: Gráfico del ajuste del sistema de 1 gdl obtenido mediante el sensor láser

Los parámetros modales que se utilizarán para comprobar que los resultados obtenidos son buenos serán la frecuencia y el coeficiente de amortiguamiento, ya que son los más representativos para la caracterización modal. Se puede comprobar que los resultados obtenidos por ambos métodos se asemejan bastante para este primer experimento, mostrado como ejemplo. La amplitud no será un parámetro a comparar ya que puede diferir en función del ajuste.

```

fitresultlaser =

General model:
fitresultlaser(t) = A*exp(-Z*2*pi*w*t)*sin(2*pi*w*t+fi)
Coefficients (with 95% confidence bounds):
A = 0.02758 (0.02743, 0.02774)
Z = 0.004938 (0.004849, 0.005027)
fi = -0.1497 (-0.1552, -0.1442)
w = 2.582 (2.582, 2.582)

goflaser =

struct with fields:

    sse: 0.0014
    rsquare: 0.9958
    dfe: 1597
    adjrsquare: 0.9958
    rmse: 9.5283e-04
    
```

Figura 68: Resultados del ajuste del sistema de 1 gdl obtenidos mediante el sensor láser

CAPÍTULO 5. PROCEDIMIENTO DE LABORATORIO.

Teniendo todo el material necesario descrito en el primer apartado, se procede a la colocación del equipo y los instrumentos. A continuación se describe el procedimiento seguido para obtener los diferentes experimentos en el laboratorio.

Primero se enciende el ordenador y se inicia el software *DewesoftX*. Se conecta la tarjeta de adquisición de datos al ordenador y los sensores de aceleración y láser en los puertos 1 y 4 de la tarjeta, respectivamente. En la Figura 69 se muestra la pantalla de inicio del programa, donde se puede ver que están siendo utilizados los puertos 1 y 4 de la tarjeta de adquisición, el primero midiendo aceleraciones en m/s^2 y el cuarto mide la intensidad de corriente recibida del láser.

ID	Used	C	Name	Ampl. name	Range	Measurement	Min	Values	Max	Physical quantity	Units	Zero	Setup
1	Used	Green	AI 1	DSI-ACC-0.16Hz	10000 mV	IEPE	-10000...	-821,3	10000,00	Acceleration	m/s ²	Zero	Setup
2	Unused	Cyan	AI 2	SIRIUS-HD-STGS	10 V	Voltage	-10,00	-0,1503	10,00		V	Zero	Setup
3	Unused	Red	AI 3	SIRIUS-HD-STGS	10 V	Voltage	-10,00	-0,0350	10,00		V	Zero	Setup
4	Used	Magenta	AI 4	SIRIUS-HD-STGS	200 mA	Current	-200,00	58,947	200,00		mA	Zero	Setup
5	Unused	Blue	AI 5	SIRIUS-HD-STGS	10 V	Voltage	-10,00	-0,1533	10,00		V	Zero	Setup

Figura 69: Interfaz del programa DewesoftX

Se colocan el acelerómetro y el láser en su posición adecuada, como se observa en la Figura 70. Los dos sensores se colocarán en la dirección horizontal del movimiento, lo que permitirá medir los desplazamientos y aceleraciones horizontales de la estructura.



Figura 70: Colocación del sensor láser y el acelerómetro para medir desplazamientos horizontales

El acelerómetro se coloca en la superficie lateral de la estructura, de manera que medirá las vibraciones en dirección horizontal (vertical en la plataforma) de esta y el láser se coloca en el otro lateral de la estructura de manera que permita medir los desplazamientos horizontales (o verticales) de la misma.

Delante de la estructura desde donde se está realizando la foto se encuentra el trípode con el teléfono móvil, que funcionará como un sensor de desplazamiento, de manera que permita grabar el punto rojo para analizarlo después y medir sus desplazamientos, en este caso horizontales (verticales en la plataforma), mediante el código en MATLAB previamente detallado.

Una vez montado todo el equipo se procede a realizar las mediciones necesarias, las cuales se analizarán y compararán posteriormente, para cada una de las diferentes estructuras. El procedimiento es el siguiente:

Primero se coloca la estructura que se quiere analizar, con sus masas correspondientes en cada caso, se acopla el acelerómetro, se orienta el láser para medir el desplazamiento horizontal (vertical en la plataforma) y se coloca el teléfono móvil de manera que grabe con claridad el punto rojo.

Una vez colocados todos los sensores, se comienza la grabación con el móvil y la adquisición de datos de los sensores, e instantáneamente se excita la estructura, en este caso mediante la aplicación de una fuerza horizontal (un golpe seco con la mano) a la estructura en la zona de estudio. En el caso de la plataforma, la fuerza que se aplica es vertical, en el primer caso un golpe seco y en el segundo caso una persona de 90kg subida encima de la misma.

En la Figura 71 se puede ver la pantalla del programa donde se reflejan los datos de las mediciones de aceleración y desplazamiento recogidas por el acelerómetro y el láser. Los valores de la aceleración se obtienen en m/s^2 y los de desplazamiento en m, ambos en función del tiempo en s.

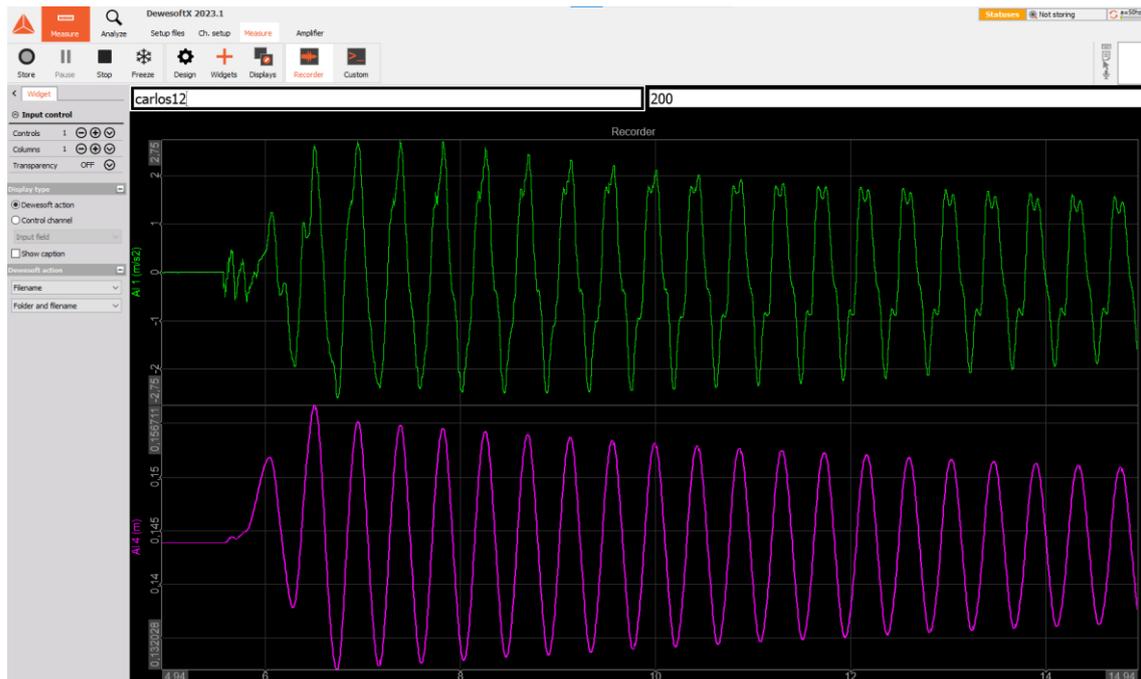


Figura 71: Gráficos de desplazamiento y aceleración obtenidos por el láser y el acelerómetro

Se deja vibrar un tiempo suficiente para que se pueda apreciar la amortiguación del movimiento, y se finaliza el video y la adquisición de datos de los sensores. Después de realizar todas las mediciones se exportan los datos adquiridos por los sensores en Excel y se guardan los vídeos registrados para su posterior análisis

Los diferentes experimentos realizados se codificarán con un número que indica el número de pisos de la estructura, seguida de una p (piso) y las masas que se colocan en ella, seguido de una m (masas). La plataforma de madera se codificará como mc (medio continuo) seguido de un 0 si no se coloca ninguna masa o una M si se coloca una masa sobre ella. Las masas colocadas sobre las maquetas son de 25/8 kg, mientras que la de la estructura de madera es de 90 kg. Los experimentos llevados a cabo fueron los siguientes:

- 1) 1p0m: Estructura vertical de 1 grado de libertad sin masas.
- 2) 1p1m: Estructura vertical de 1 grado de libertad con una masa de 25/8 kg.
- 3) 1p2m: Estructura vertical de 1 grado de libertad con dos masas de 25/8 kg
- 4) mc0: Medio continuo (plataforma de madera) sin masas.
- 5) mcM: Medio continuo (plataforma de madera) con masa de 90 kg.
- 6) 2p2m: Estructura vertical de 2 grados de libertad con dos masas de 25/8 kg.
- 7) 3p0m: Estructura vertical de 3 grados de libertad sin masas.
- 8) 4p0m: Estructura vertical de 4 grados de libertad sin masas.

CAPÍTULO 6. RESULTADOS. COMPARACIÓN LASER CON VISIÓN

Para poder comparar los resultados del láser con los obtenidos mediante el análisis por visión artificial de debe analizar la frecuencia de vibración y el coeficiente de amortiguamiento para cada caso. Se estudiarán, analizarán y compararán cada uno de los casos que se van a mostrar, los más representativos para diferentes sistemas de 1, 2 y 3 grados de libertad, utilizando el código creado en MATLAB explicado anteriormente.

Los resultados muestran los gráficos obtenidos por cada método, sus correspondientes gráficos acondicionados para poder ajustar la función de desplazamiento, la superposición de las curvas obtenidas por ambos métodos, para corroborar que coinciden, los gráficos de las funciones de ajuste obtenidas para cada uno de ellos y finalmente la tabla de resultados que muestra la función de ajuste en cada caso y los parámetros modales w , Z , A y f_i , además del parámetro de bondad del ajuste R^2 .

Los sistemas de más de un grado de libertad se analizan desde un grado de libertad hasta el número de grados de la estructura, de manera que se obtienen los parámetros para cada grado de libertad de la estructura, comprobando que los de cada ajuste coinciden con los resultados finales. A continuación se muestran los resultados obtenidos para cada uno de los casos estudiados:

6.1. Estructura de 1 grado de libertad sin masas (1p0m)



Figura 72: Estructura de 1 gdl sin masas

Para el experimento 1p0m, obtenido tras excitar la estructura de la Figura 72 y grabar simultáneamente con el láser la videocámara, se obtiene la gráfica de la Figura 73 tras procesar los fotogramas. El registro del láser se muestra en la Figura 74.

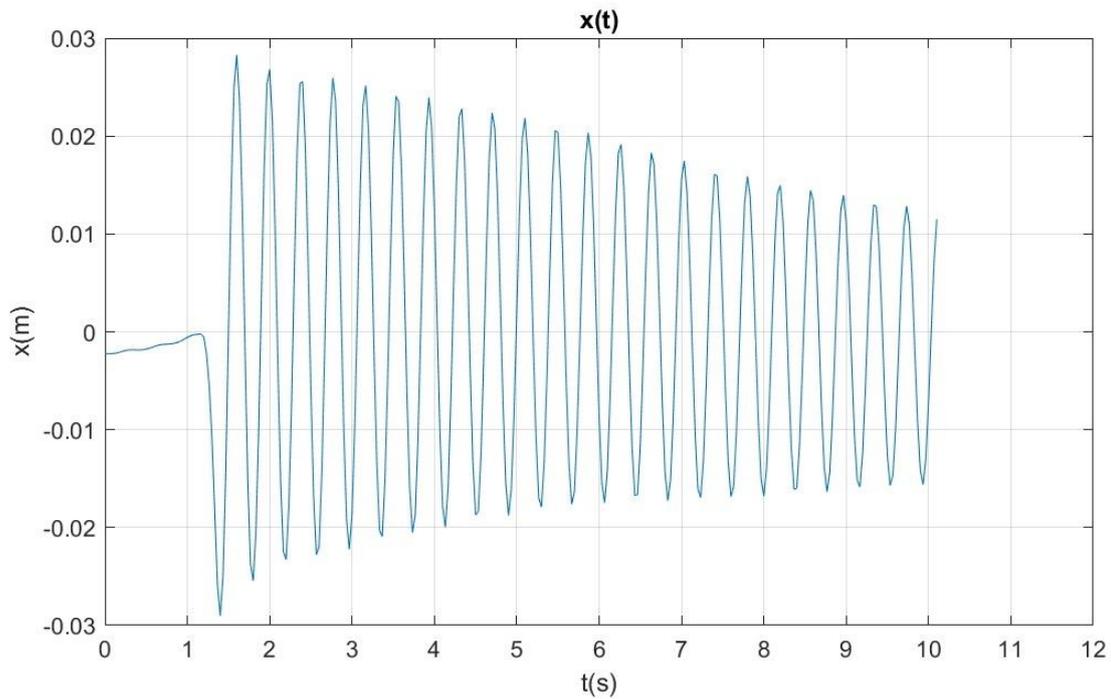


Figura 73: Gráfico del desplazamiento en función del tiempo 1p0m. Visión

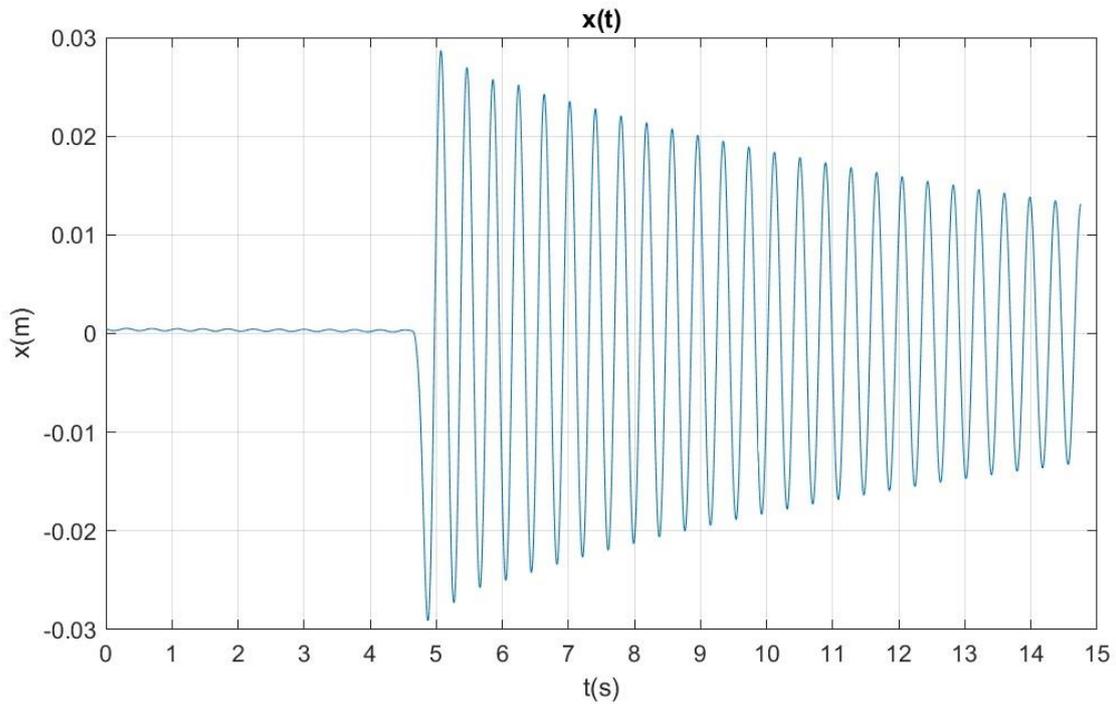


Figura 74: Gráfico del desplazamiento en función del tiempo 1p0m. Láser

Se procede al acondicionamiento de los datos, es decir, eliminar el registro previo a la excitación acotando la función a un intervalo de 8 segundos, resultando las gráficas de la Figura 75 y Figura 76. Superponiendo ambas curvas, se observa (Figura 77) que son similares pero no iguales, por lo que tras su ajuste se obtendrán parámetros ligeramente distintos.

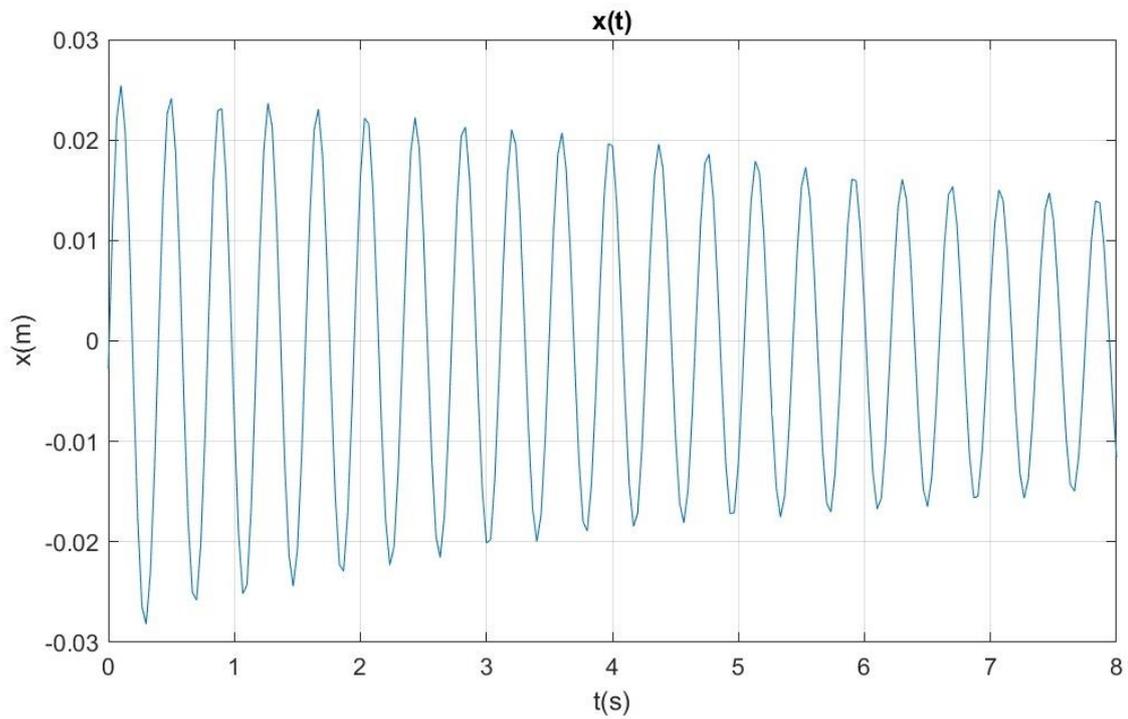


Figura 75: Gráfico acondicionado del desplazamiento en función del tiempo 1p0m. Visión

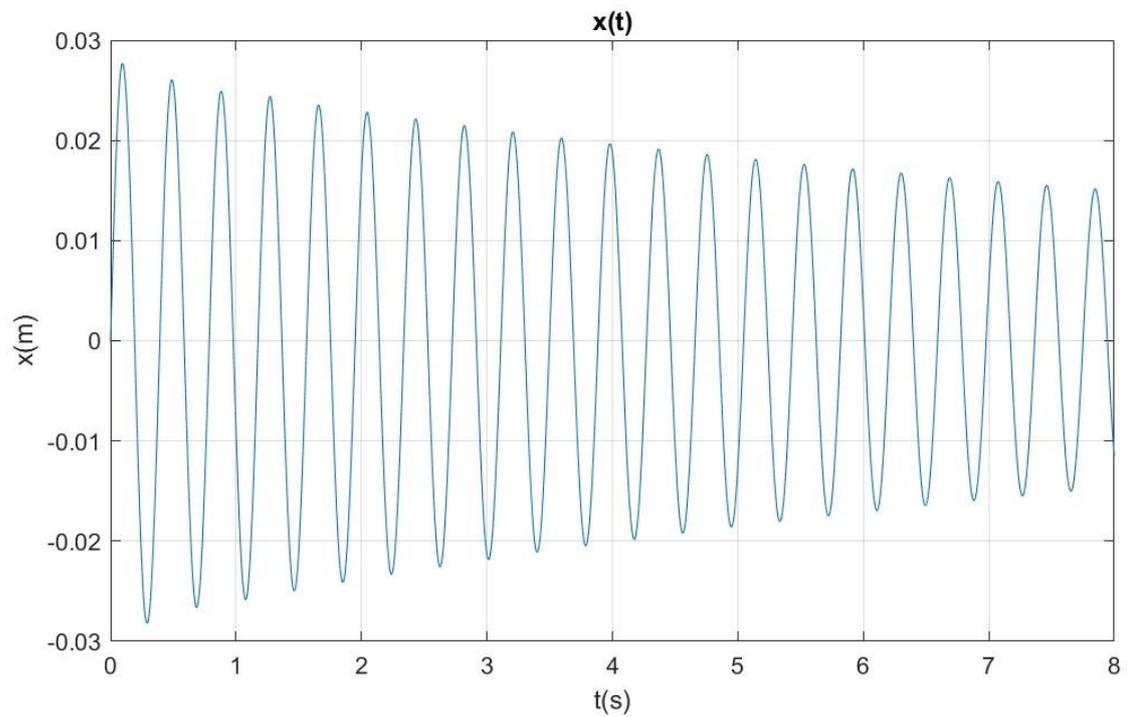


Figura 76: Gráfico acondicionado del desplazamiento en función del tiempo 1p0m. Láser

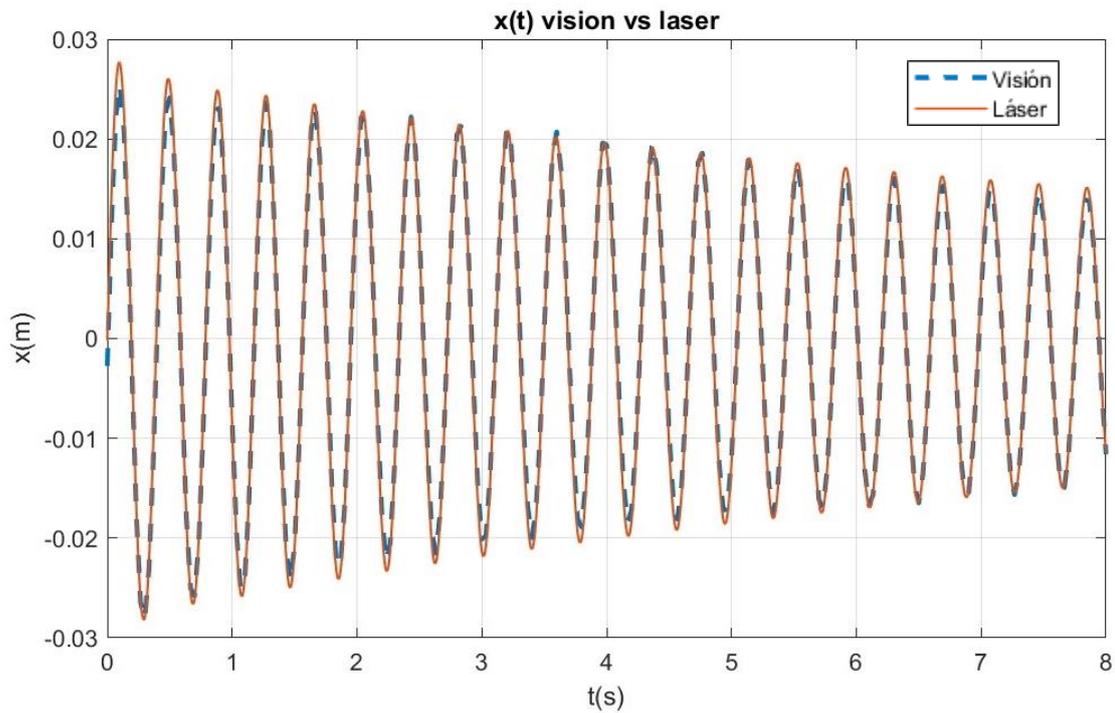


Figura 77: Superposición de gráficos de desplazamiento láser y visión 1p0m

La Figura 78 muestra el ajuste obtenido para este experimento, en azul el registro original y en rojo la función ajustada, cuyos parámetros se listan en la Tabla 1. Análogamente la Figura 79 y la Tabla 2 corresponden al ajuste de la curva registrada por láser.

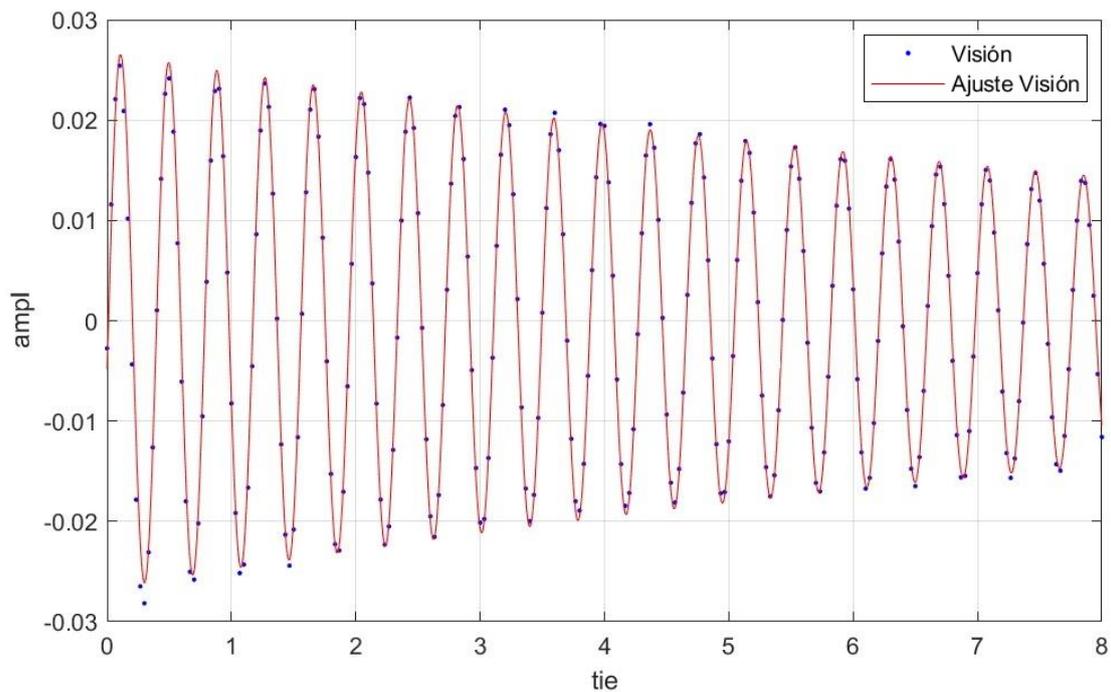


Figura 78: Ajuste 1p0m visión

$\text{fitresultvision}(t) = A \cdot \exp(-Z \cdot 2 \cdot \pi \cdot w \cdot t) \cdot \sin(2 \cdot \pi \cdot w \cdot t + f_i)$
Coefficients (with 95% confidence bounds):
A = 0,02677 (0,02631, 0,02724)
Z = 0,004802 (0,004526, 0,005078)
$f_i = -0,1794 (-0,1965, -0,1624)$
w = 2,582 (2,581, 2,583)
rsquare: 0,9940

Tabla 1: Resultados 1p0m visión

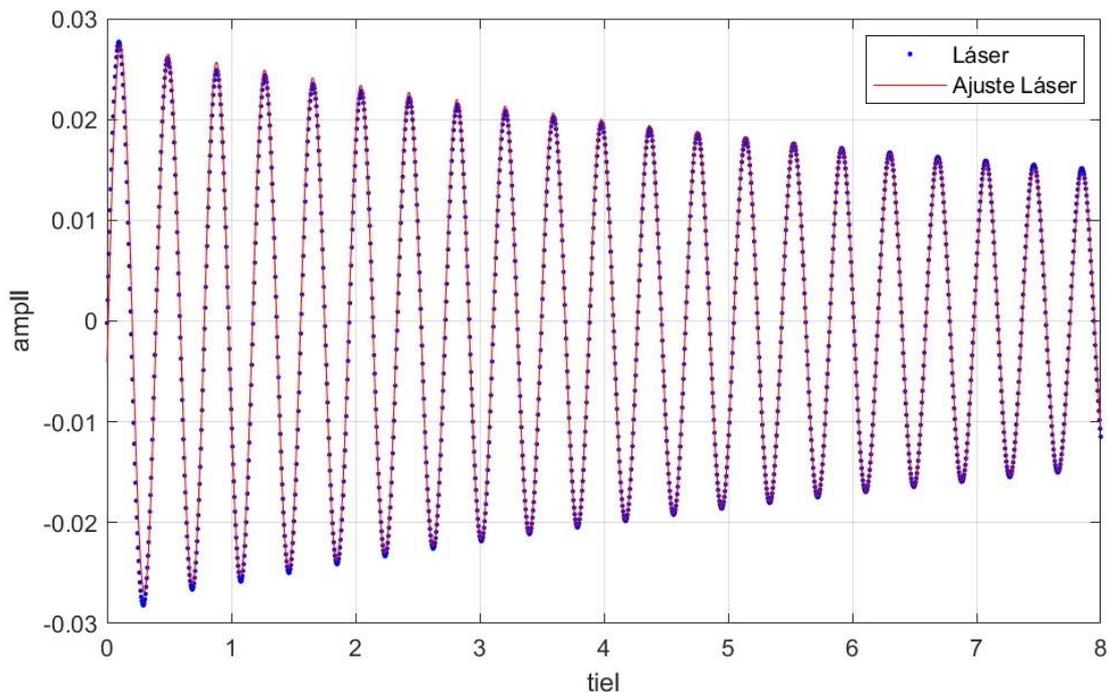


Figura 79: Ajuste 1p0m láser

$\text{fitresultvision}(t) = A \cdot \exp(-Z \cdot 2 \cdot \pi \cdot w \cdot t) \cdot \sin(2 \cdot \pi \cdot w \cdot t + f_i)$
Coefficients (with 95% confidence bounds):
A = 0,02758 (0,02743, 0,02774)
Z = 0,004938 (0,004849, 0,005027)
$f_i = -0,1497 (-0,1552, -0,1442)$
w = 2,582 (2,582, 2,582)
rsquare: 0,9958

Tabla 2: Resultados 1p0m láser

Como se ha comentado, de todos los parámetros ajustados los más representativos para la caracterización modal son la frecuencia y el coeficiente de amortiguamiento. Para las 4 cifras significativas tenidas en cuenta, el error relativo en la estimación de la frecuencia es nulo y para el amortiguamiento es de tan solo 2.75%.

6.2. Estructura de 1 grado de libertad con masa de 25/8 kg (1p1m)



Figura 80: Estructura de 1 gdl con masa de 25/8 kg

Para el experimento 1p1m se excita la estructura de la Figura 80 y se graba simultáneamente con la cámara y el láser. Tras procesar el video y los datos obtenidos por el láser, se registran las gráficas de la Figura 81 y Figura 82 respectivamente.

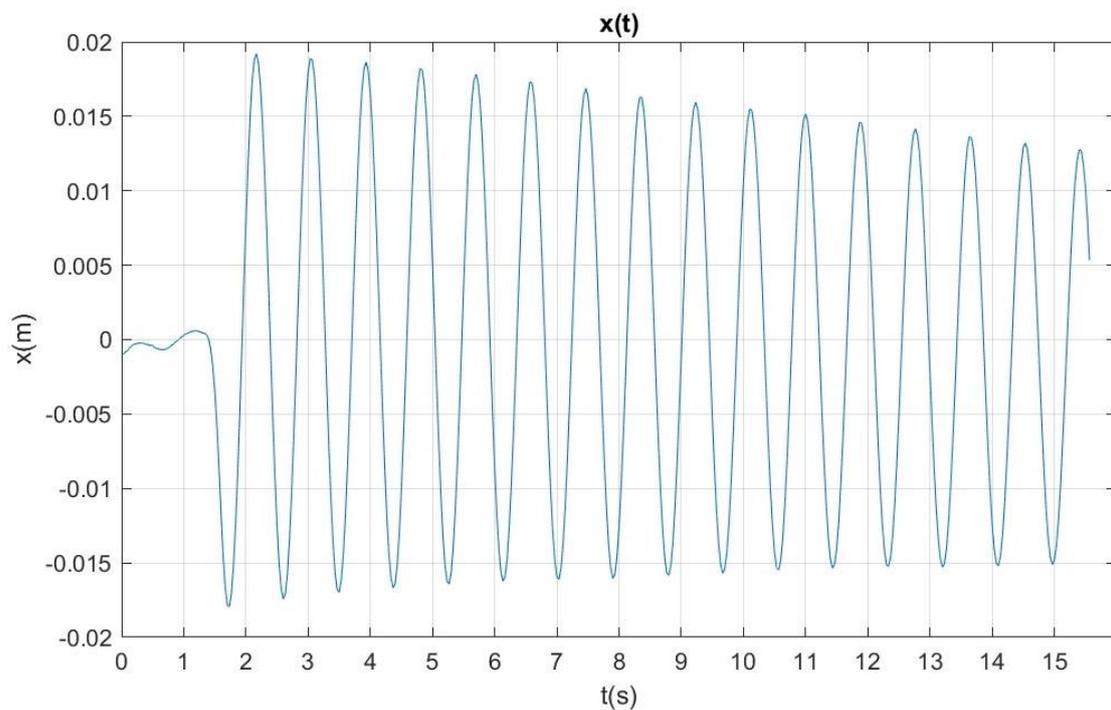


Figura 81: Gráfico del desplazamiento en función del tiempo 1p1m. Visión

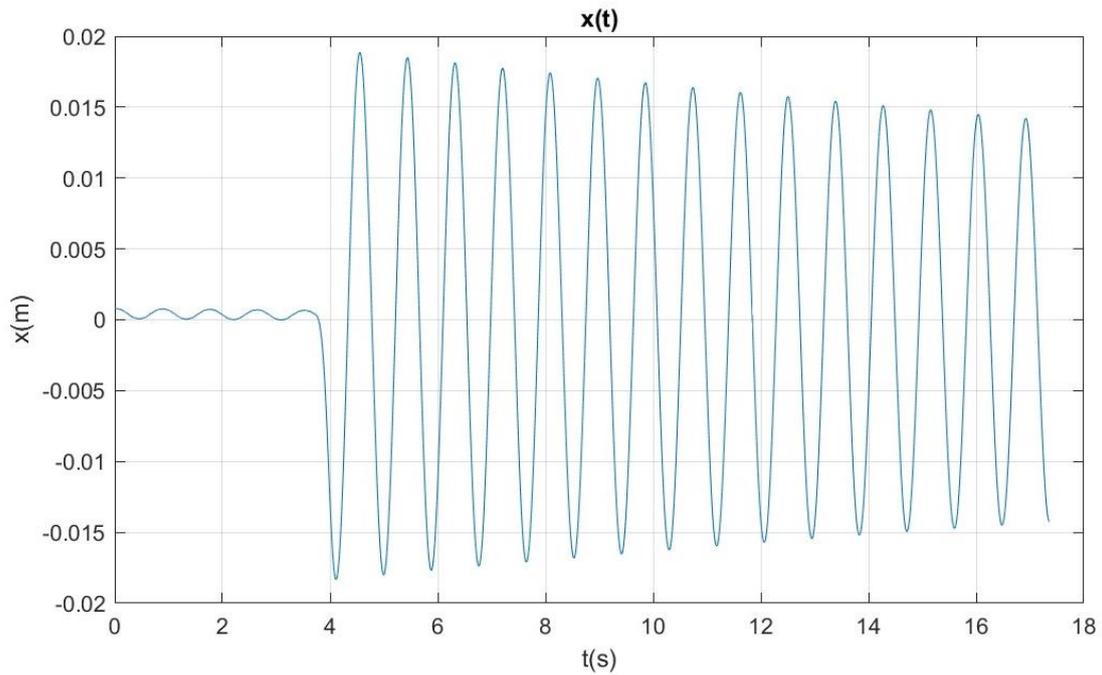


Figura 82: Gráfico del desplazamiento en función del tiempo 1p1m. Láser

Se elimina el registro previo a la excitación, acotando la función a un intervalo de 10 segundos en este caso, resultando las gráficas acondicionadas de la Figura 83 y Figura 84. Como se puede observar en la superposición de las dos curvas (Figura 85), son muy similares pero no iguales, por lo que tras realizar el ajuste, los parámetros obtenidos por ambos métodos no serán exactamente los mismos.

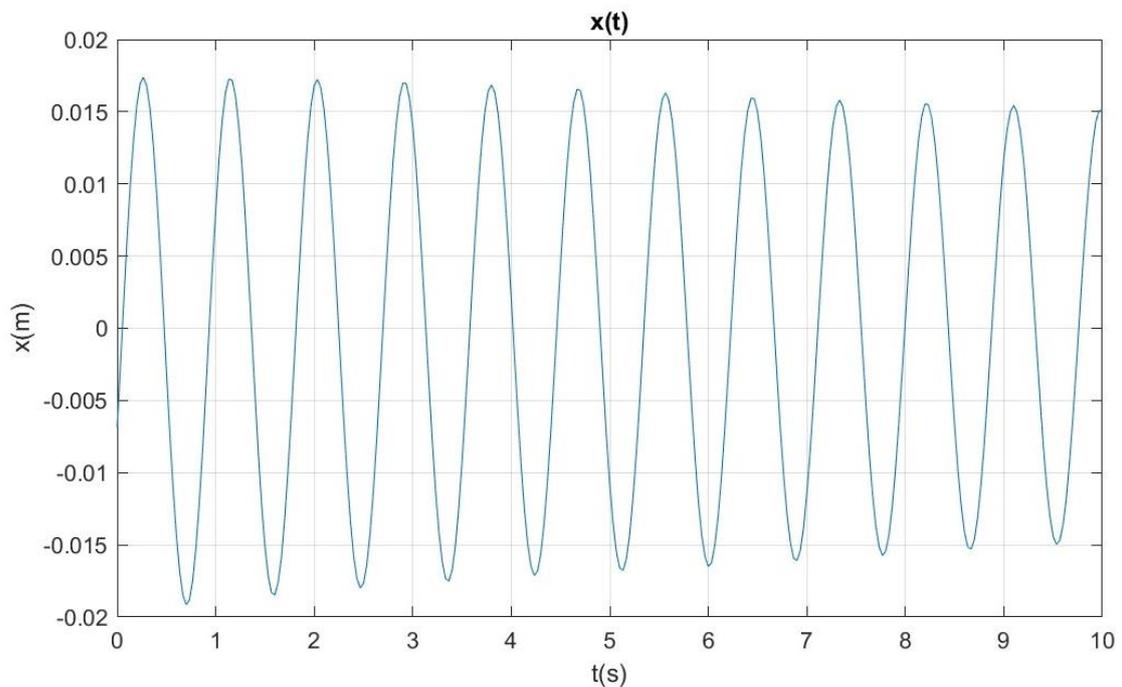


Figura 83: Gráfico acondicionado del desplazamiento en función del tiempo 1p1m. Visión

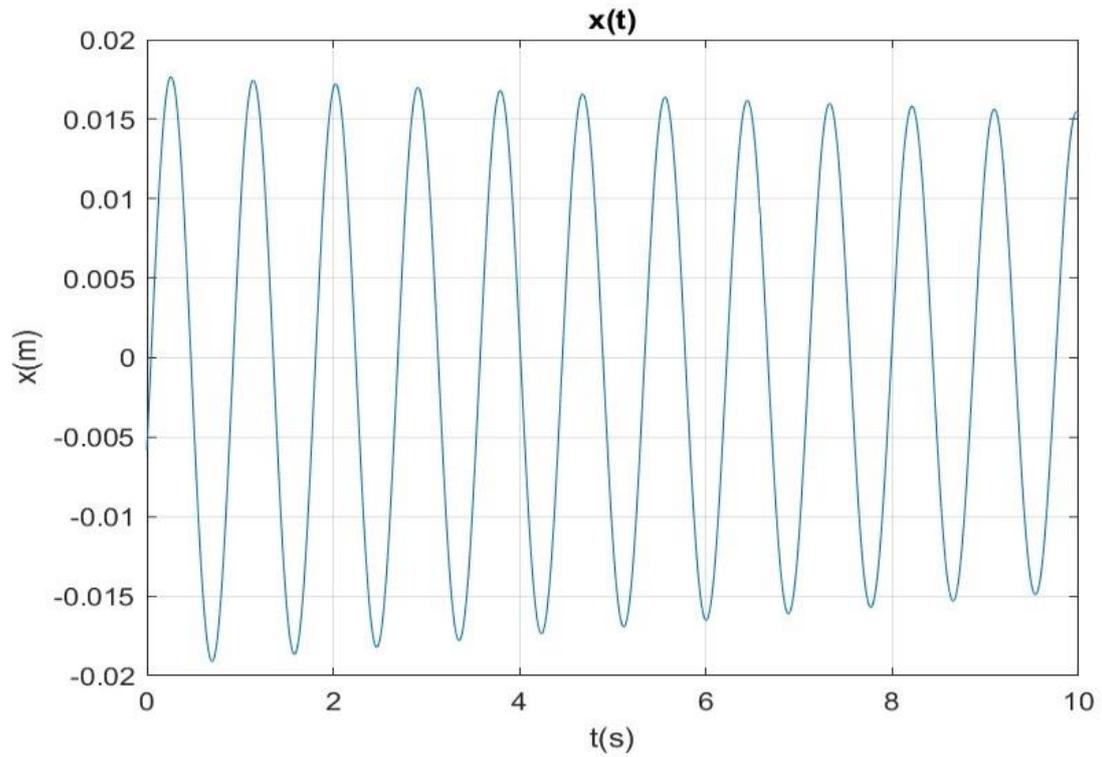


Figura 84: Gráfico acondicionado del desplazamiento en función del tiempo 1p1m. Láser

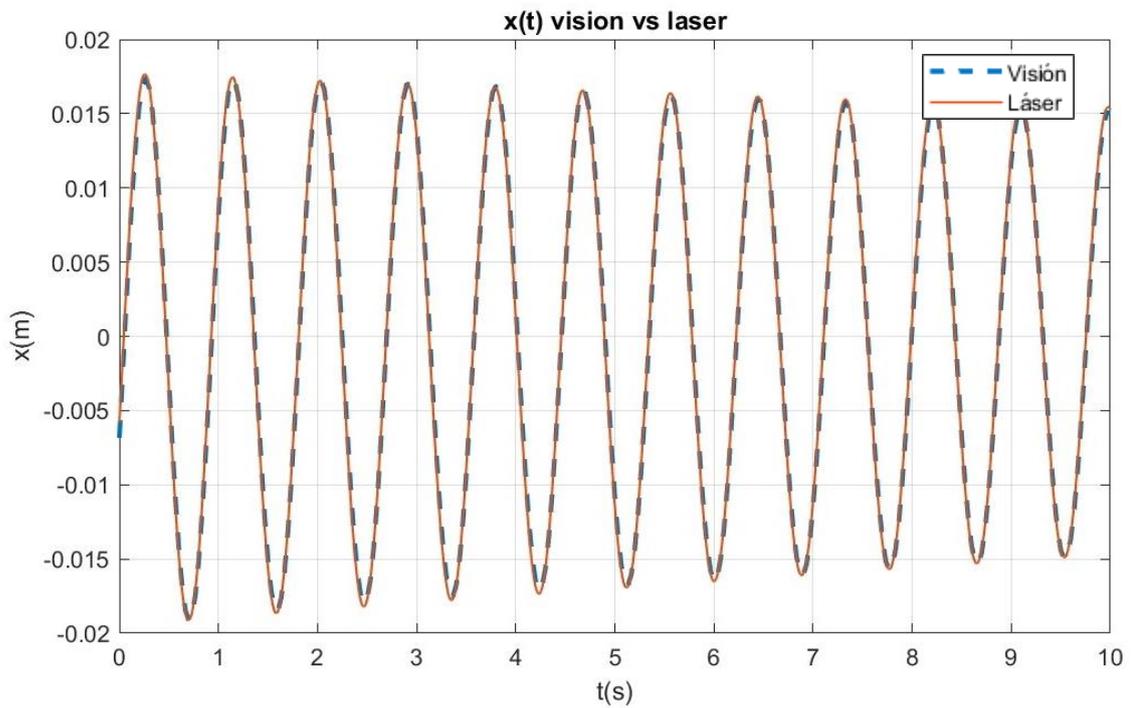


Figura 85: Superposición de gráficos de desplazamiento láser y visión 1p1m

El ajuste obtenido en este caso se muestra en la Figura 86, en azul los datos adquiridos y en rojo la función que se ajusta a estos. En la Tabla 3 se muestran los parámetros de este ajuste. De la misma manera se pueden observar los datos correspondientes al registro del láser en la Figura 87 y la Tabla 4.

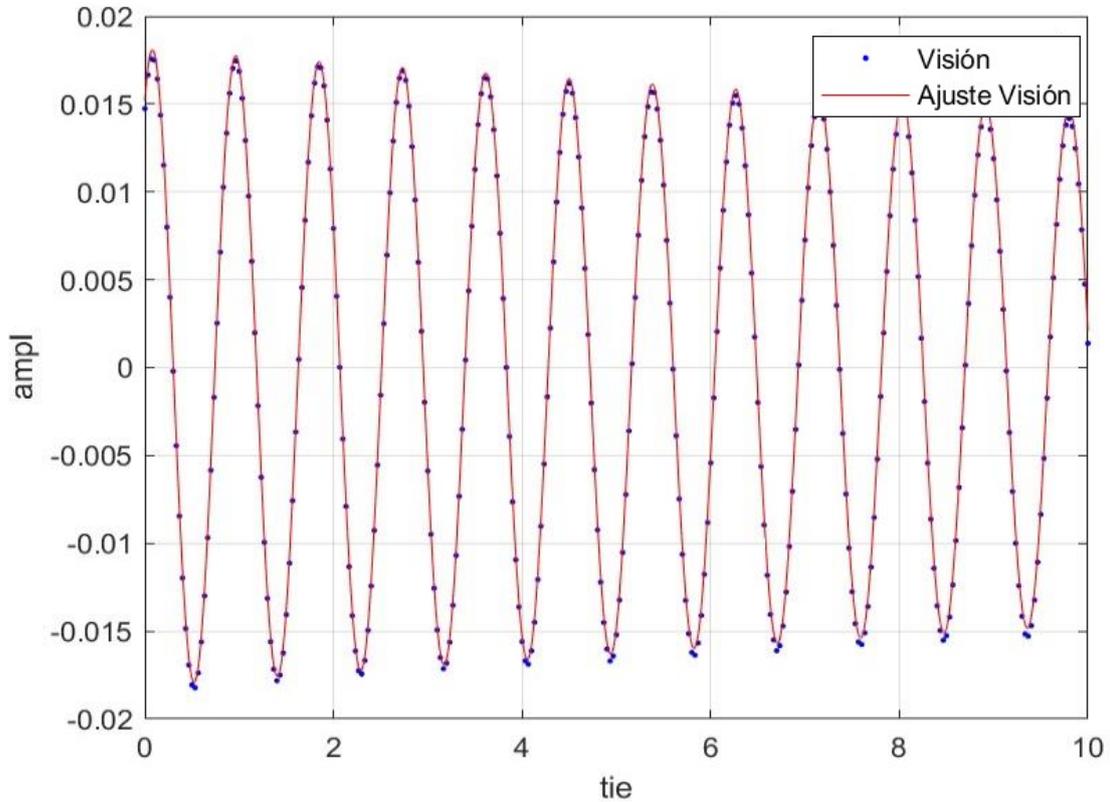


Figura 86: Ajuste 1p1m. Visión

$\text{fitresultvision}(t) = A \cdot \exp(-Z \cdot 2 \cdot \pi \cdot w \cdot t) \cdot \sin(2 \cdot \pi \cdot w \cdot t + f_i)$
Coefficients (with 95% confidence bounds):
A = -0,01841 (-0,01856, -0,01826)
Z = 0,002818 (0,002613, 0,003023)
f _i = -3,454 (-3,462, -3,446)
w = 1,132 (1,132, 1,132)
rsquare: 0,9987

Tabla 3: Resultados 1p1m visión

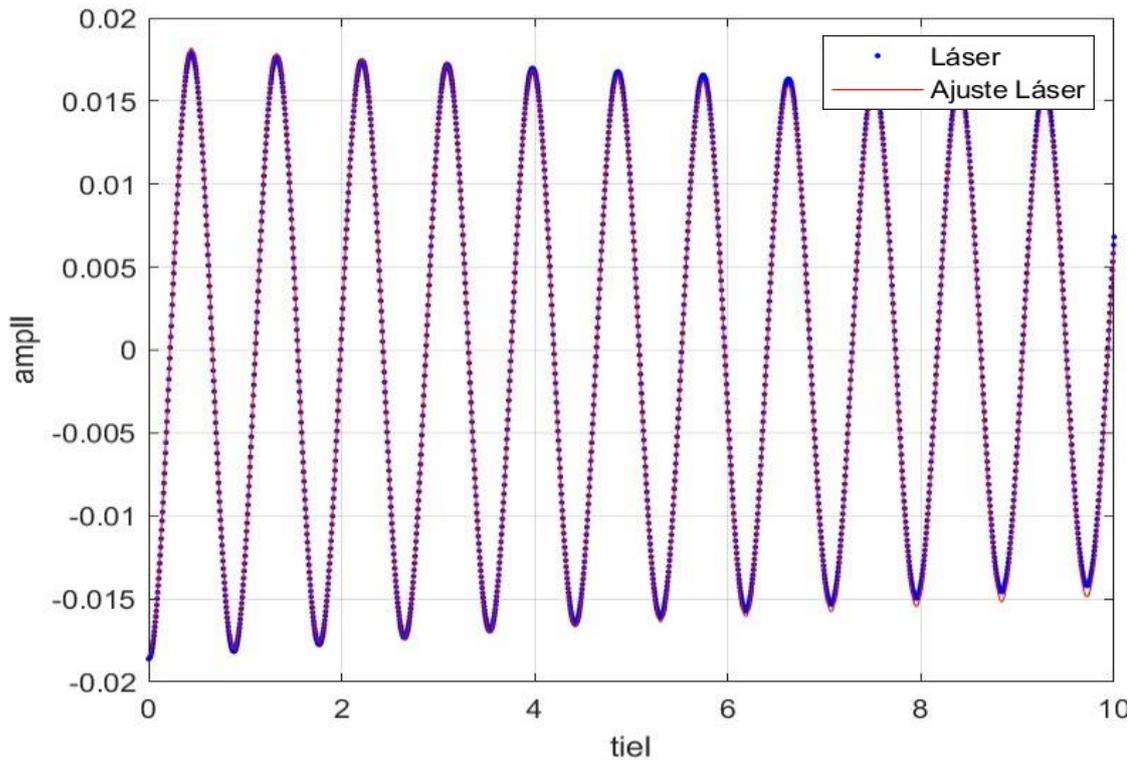


Figura 87: Ajuste 1p1m. Láser

$\text{fitresultláser}(t) = A \cdot \exp(-Z \cdot 2 \cdot \pi \cdot w \cdot t) \cdot \sin(2 \cdot \pi \cdot w \cdot t + f_i)$
Coefficients (with 95% confidence bounds):
A = -0,01851 (-0,01857, -0,01845)
Z = 0,002829 (0,002745, 0,002912)
f _i = -3,411 (-3,415, -3,408)
w = 1,132 (1,132, 1,132)
rsquare: 0,9985

Tabla 4: Resultados 1p1m láser

Para la caracterización modal se analizará la frecuencia y el coeficiente de amortiguamiento, parámetros más representativos. El error relativo en la estimación de la frecuencia es nulo al igual que en el caso anterior, mientras que el amortiguamiento es menor, presentando un error del 0,39%.

Se puede observar también que al añadir una masa a la misma estructura del experimento anterior (1p0m), la frecuencia y el coeficiente de amortiguamiento disminuyen.

6.3. Estructura de 1 grado de libertad con dos masas de 25/8 kg (1p2m)



Figura 88: Estructura de 1 gdl con dos masas de 25/8 kg

Para el experimento 1p2m, obtenido al excitar la estructura de la Figura 88 y grabar simultáneamente con el láser y con el dispositivo móvil, se obtiene la gráfica de la Figura 89 tras realizar el procesamiento del video, y el registro del láser mostrado en la Figura 90.

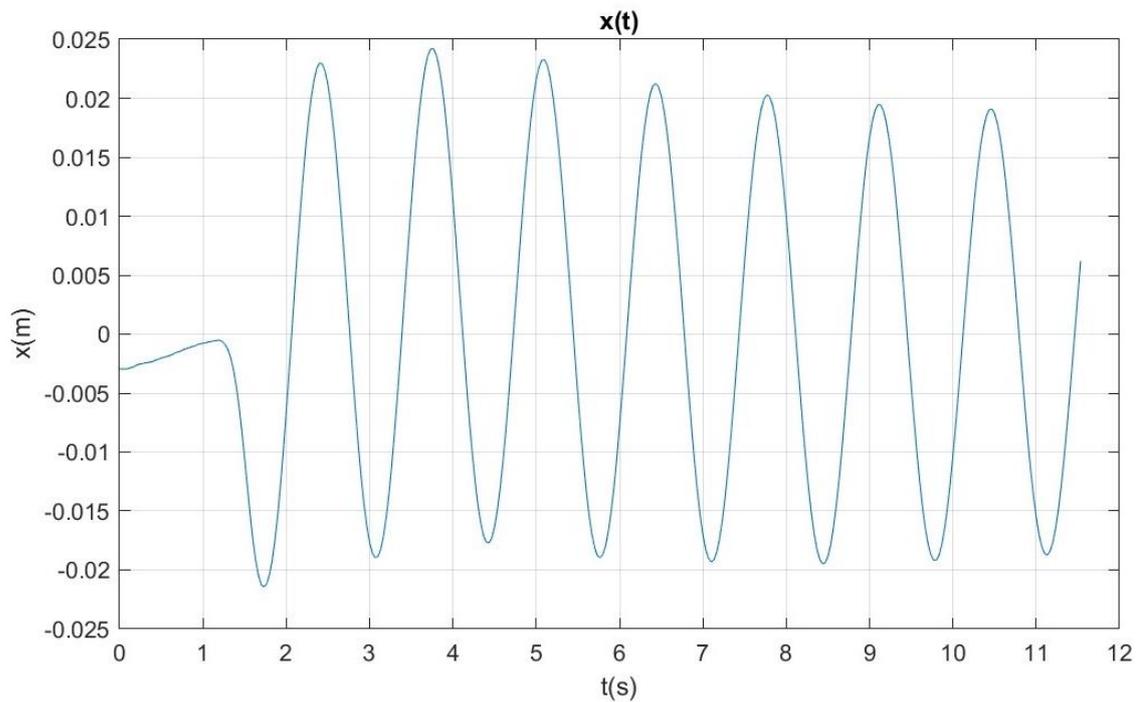


Figura 89: Gráfico del desplazamiento en función del tiempo 1p2m. Visión

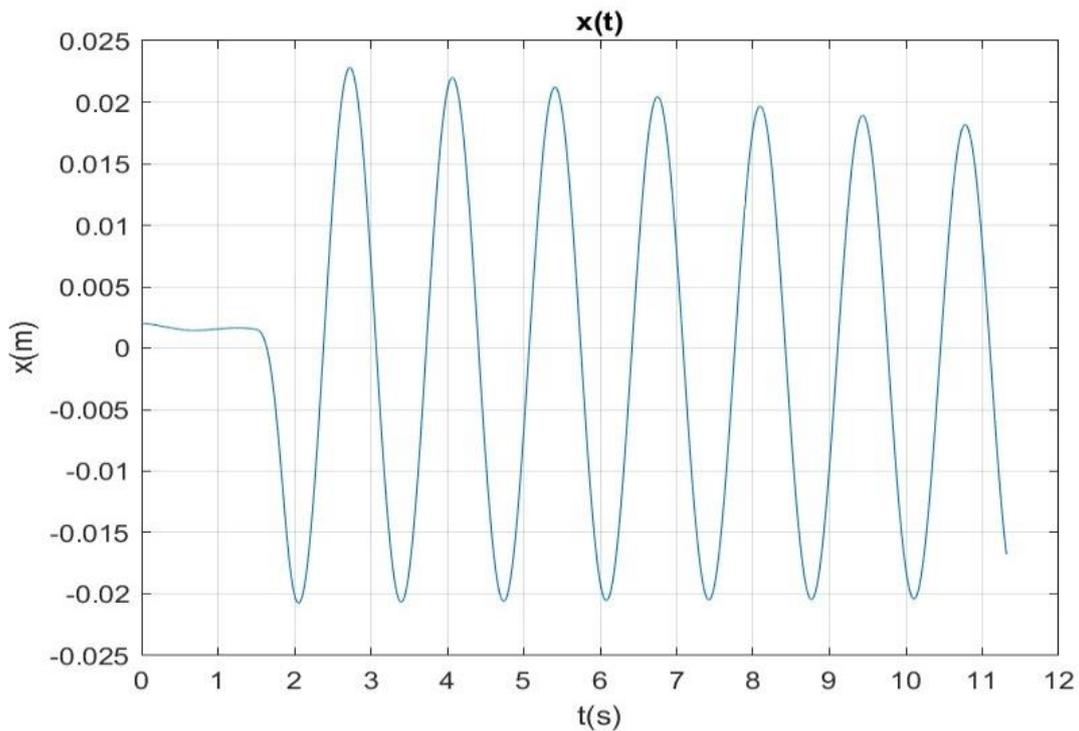


Figura 90: Gráfico del desplazamiento en función del tiempo 1p2m. Láser

Tras acondicionar los datos eliminando el registro previo a la excitación y acotando la función a un intervalo de 8 segundos, se obtienen las gráficas de la Figura 91 y Figura 92. En la Figura 93 se observa que son similares pero existen pequeñas diferencias, por lo que tras el ajuste de ambas curvas los parámetros obtenidos variarán ligeramente.

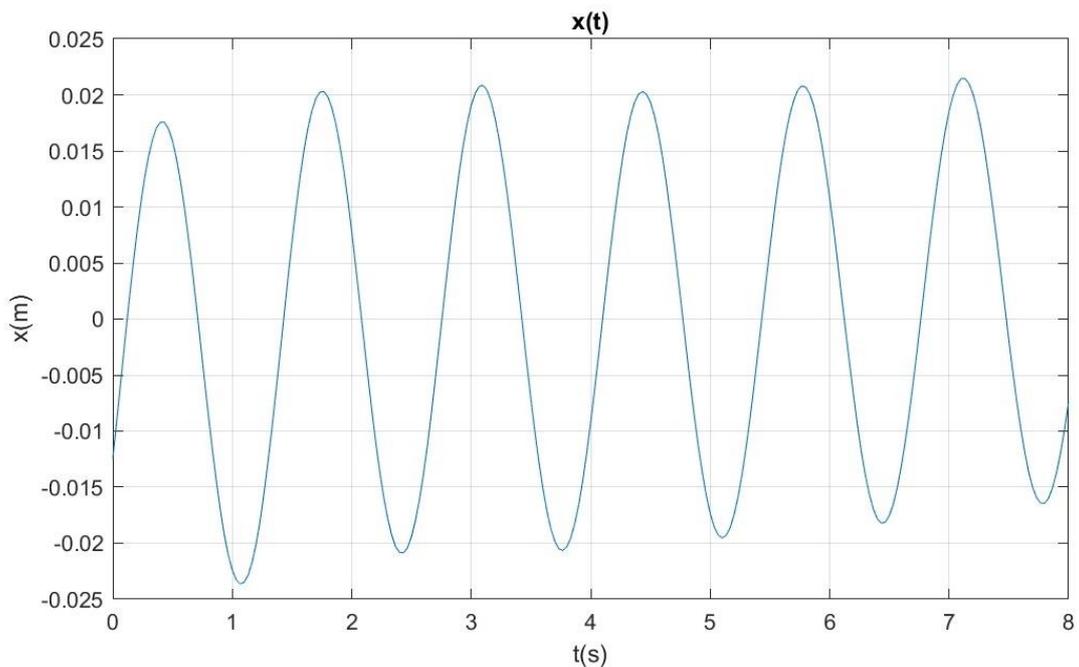


Figura 91: Gráfico acondicionado del desplazamiento en función del tiempo 1p2m. Visión

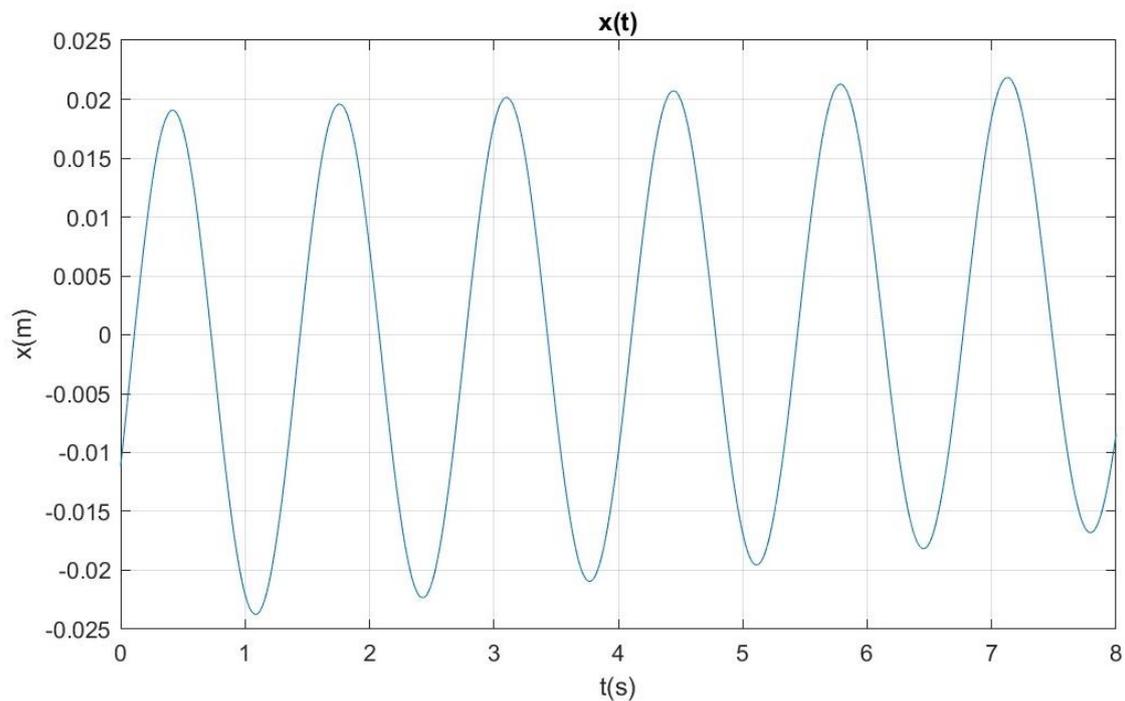


Figura 92: Gráfico acondicionado del desplazamiento en función del tiempo 1p2m. Láser

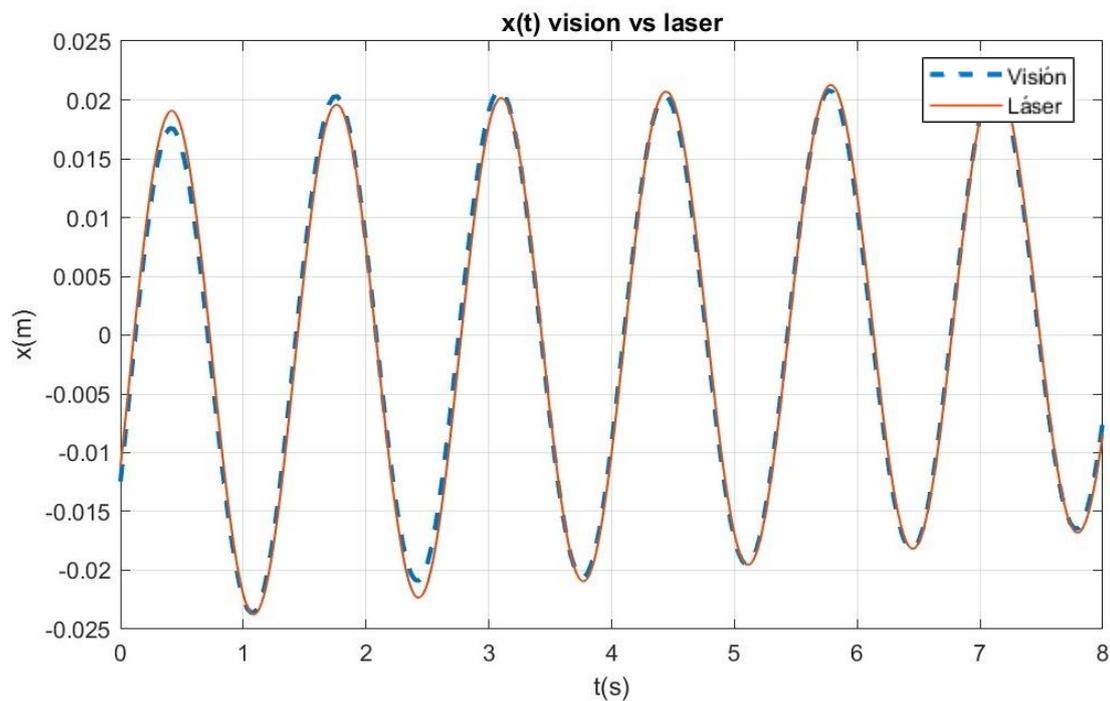


Figura 93: Superposición de gráficos de desplazamiento láser y visión 1p2m

La Figura 94 muestra el ajuste obtenido para este experimento, en azul el registro original y en rojo la función ajustada, cuyos parámetros se presentan en la Tabla 5. Análogamente la Figura 95 y la Tabla 6 corresponden al ajuste del registro del láser.

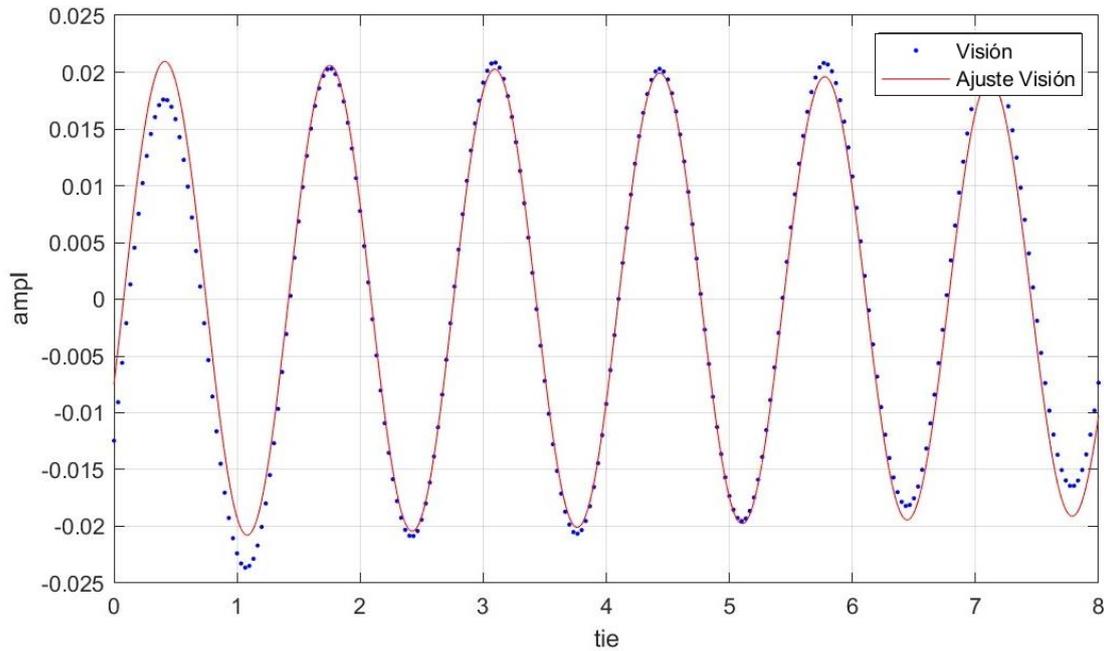


Figura 94: Ajuste de 1p2m visión

$\text{fitresultvision}(t) = A \cdot \exp(-Z \cdot 2 \cdot \pi \cdot w \cdot t) \cdot \sin(2 \cdot \pi \cdot w \cdot t + f_i)$
Coefficients (with 95% confidence bounds):
A = 0,02107 (0,02039, 0,02174)
Z = 0,002671 (0,001175, 0,004166)
f _i = -0,365 (-0,396, -0,334)
w = 0,7459 (0,7448, 0,747)
rsquare: 0,9842

Tabla 5: Resultados 1p2m visión

En este caso, estimando la frecuencia mediante ambos métodos se obtiene un error relativo muy pequeño, de 0,8%, en cambio, para el coeficiente de amortiguamiento es mucho mayor, ascendiendo a 13,62%.

De la misma manera que al añadir una masa en el experimento anterior la frecuencia se vio disminuida, al añadir otra masa, la frecuencia disminuye también en este caso. El coeficiente de amortiguamiento también debería disminuir como se puede ver en el análisis por visión, pero no disminuye el resultado obtenido por el láser, debido al error comentado anteriormente entre ambos métodos.

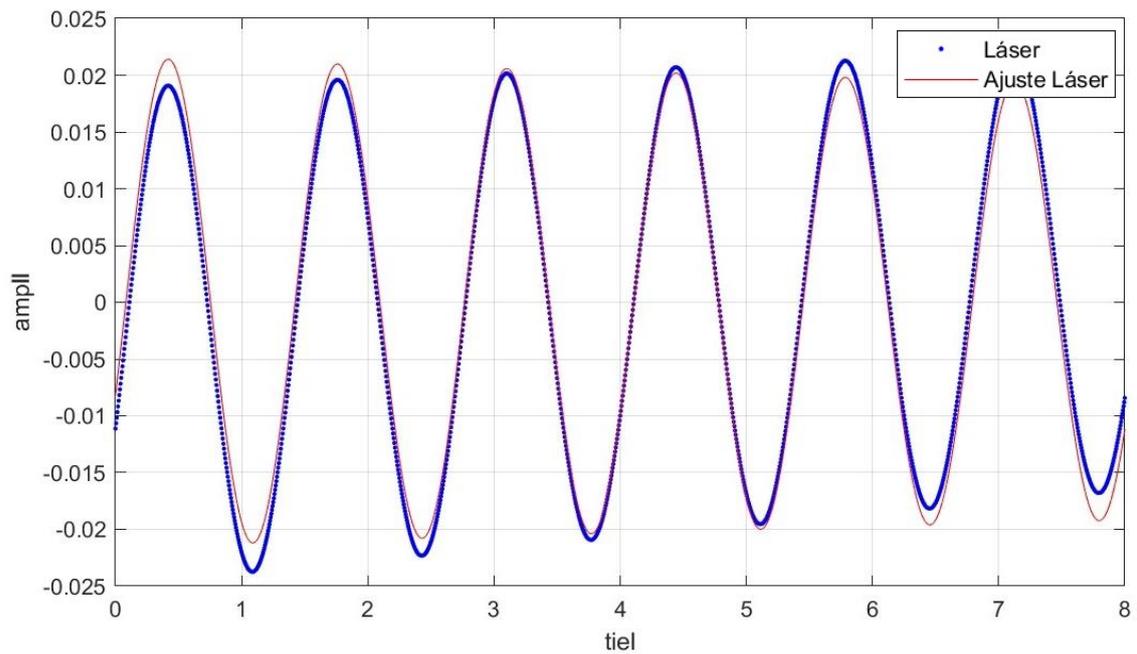


Figura 95: Ajuste 1p2m láser

$\text{fitresultláser}(t) = A \cdot \exp(-Z \cdot 2 \cdot \pi \cdot w \cdot t) \cdot \sin(2 \cdot \pi \cdot w \cdot t + f_i)$
Coefficients (with 95% confidence bounds):
A = 0,02153 (0,02129, 0,02177)
Z = 0,003092 (0,002567, 0,003617)
f _i = -0,3873 (-0,3982, -0,3764)
w = 0,7453 (0,7449, 0,7457)
rsquare: 0,9868

Tabla 6: Resultados 1p2m láser

6.4. Medio continuo (plataforma de madera) sin masas (mc0)



Figura 96: Plataforma de madera, medio continuo sin masas

Para el experimento mc0, obtenido tras excitar la estructura de la Figura 96 y grabar simultáneamente con el láser y con la cámara, se obtiene la gráfica de la Figura 97 tras procesar el video registrado, y la de la Figura 98 a partir de los datos adquiridos por el láser. Al ser un medio continuo tiene infinitos grados de libertad pero en este caso solo se manifiesta el primero, grado de libertad que se analizará.

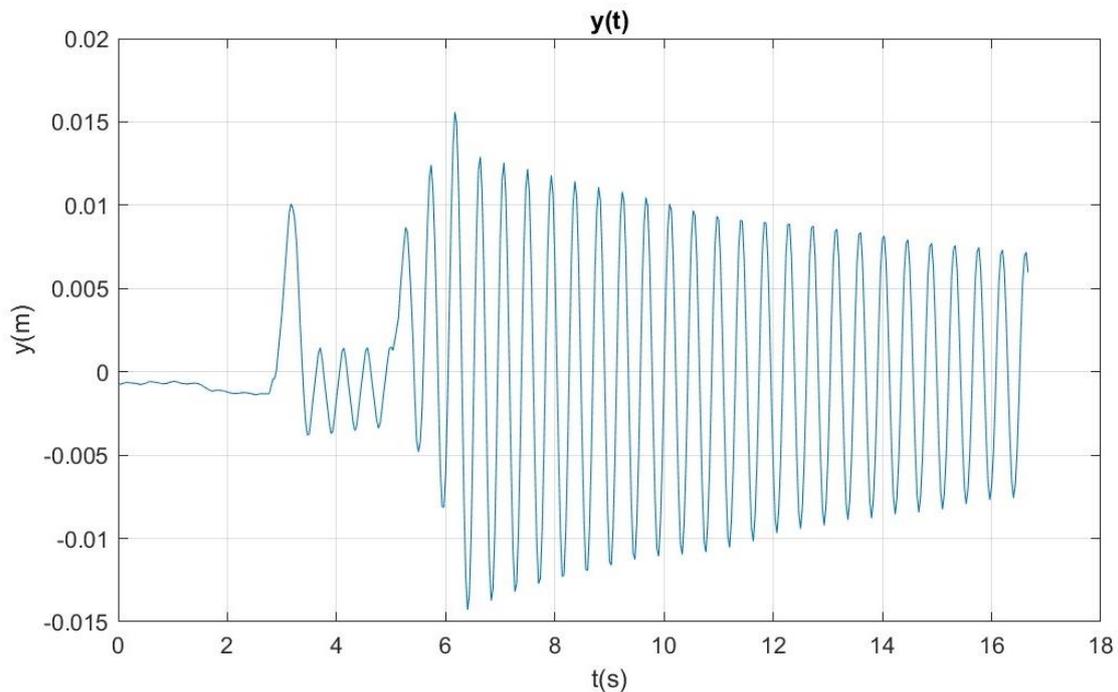


Figura 97: Gráfico del desplazamiento en función del tiempo mc0. Visión

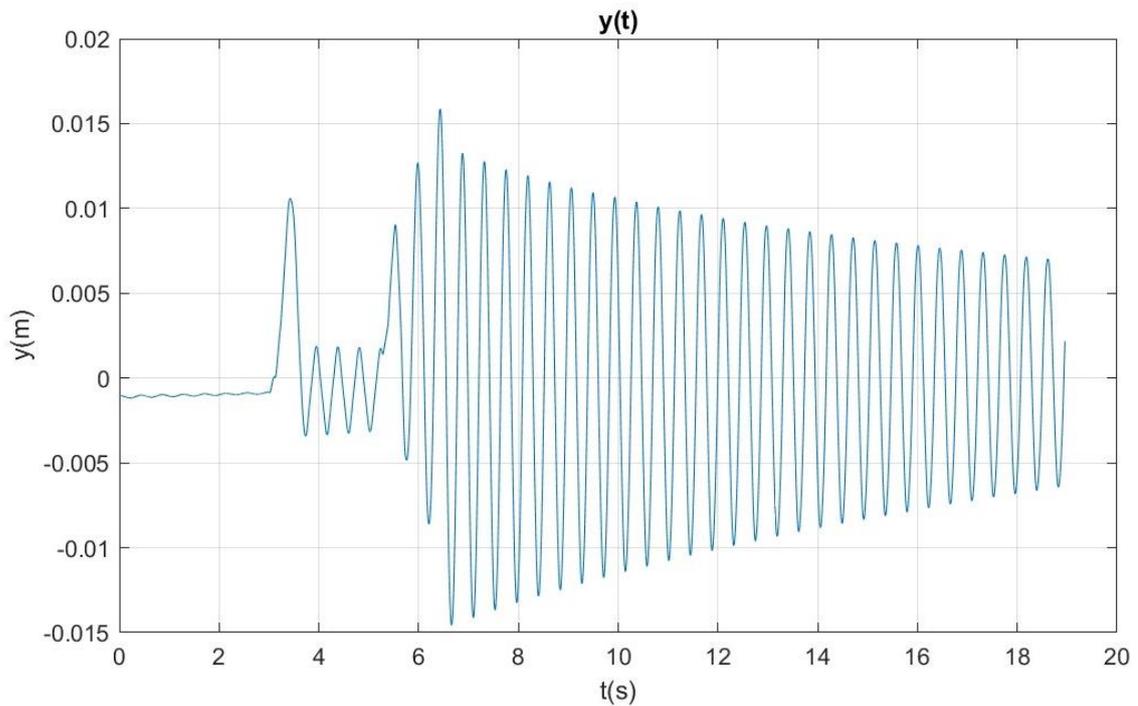


Figura 98: Gráfico del desplazamiento en función del tiempo mc0. Láser

Acondicionando estos datos, se elimina el registro previo a la excitación y se acota la función a un intervalo de 10 segundos, obteniendo las gráficas de la Figura 99 y Figura 100. Superponiendo ambas, se observa en la Figura 101 que casi idénticas pero se pueden ver pequeñas diferencias.

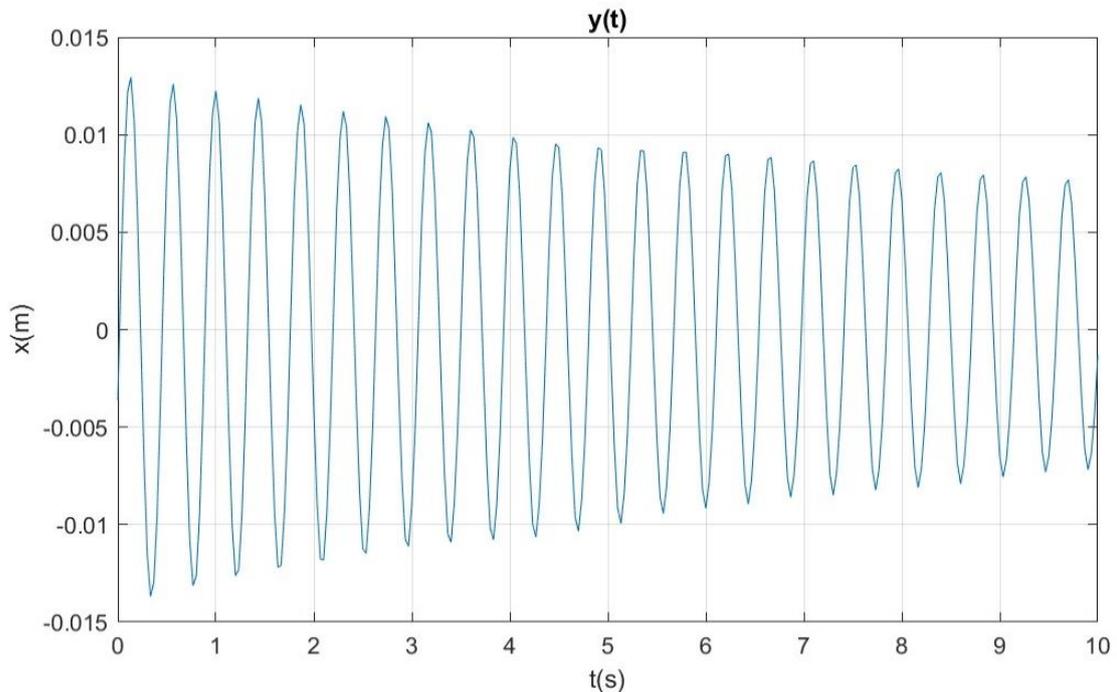


Figura 99: Gráfico acondicionado del desplazamiento en función del tiempo mc0. Visión

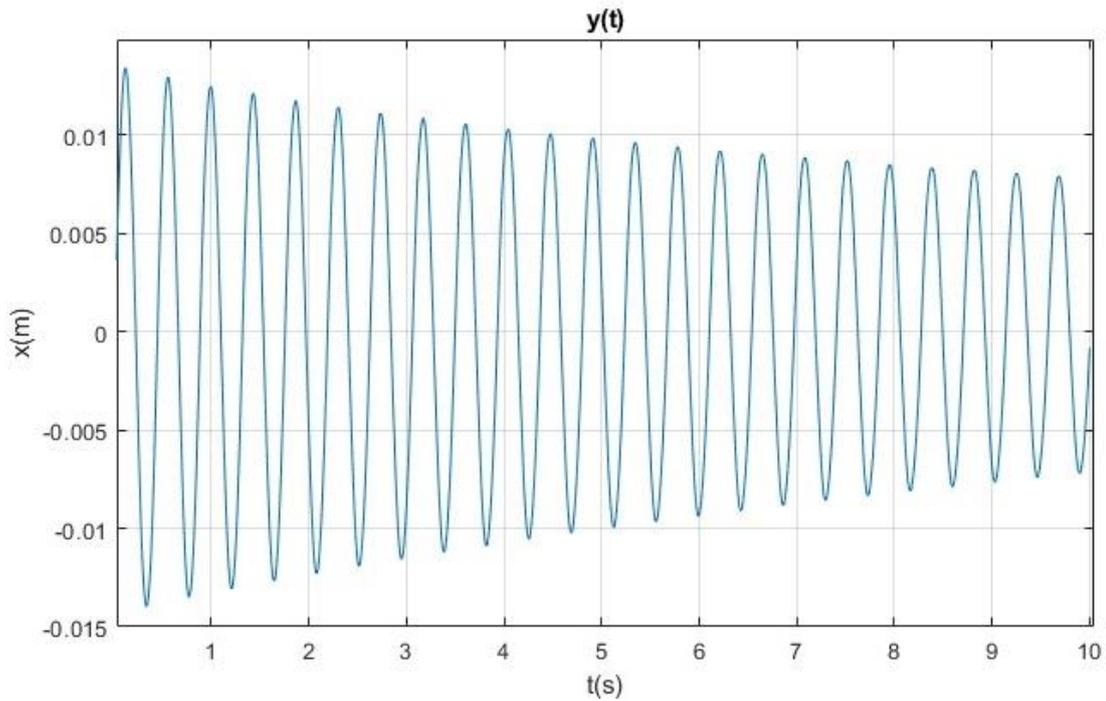


Figura 100: Gráfico acondicionado del desplazamiento en función del tiempo mc0. Láser

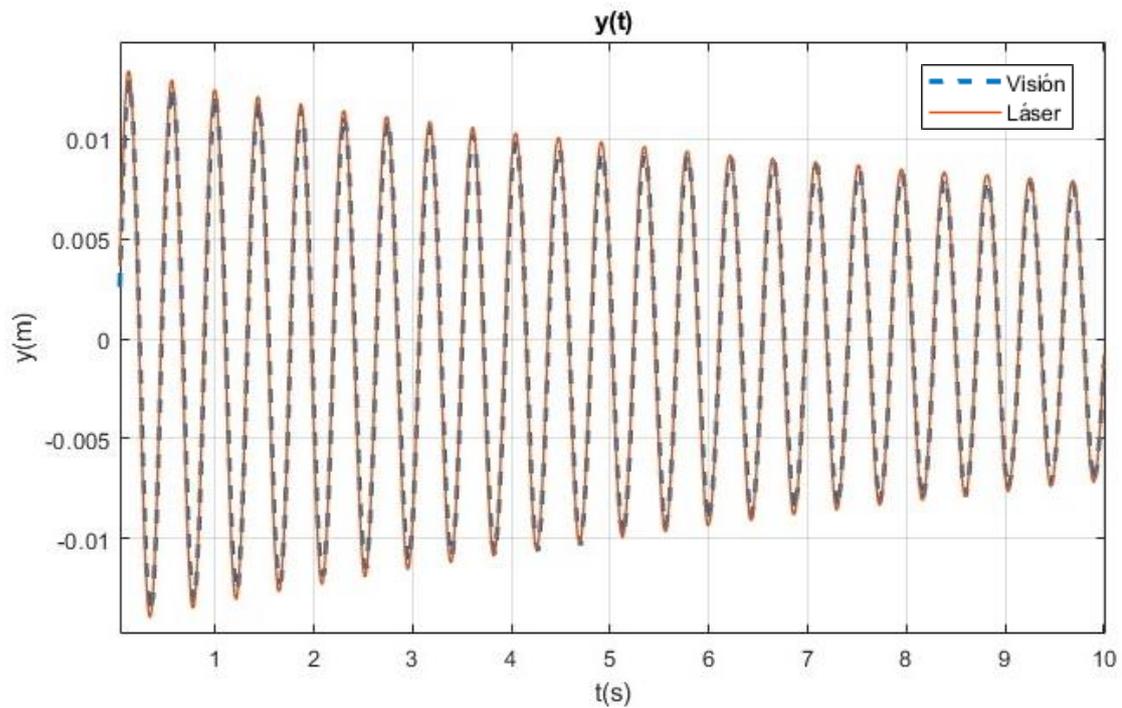


Figura 101: Superposición de gráficos de desplazamiento láser y visión mc0

En la Figura 102 se muestra el ajuste obtenido en este caso, en azul el registro original y en rojo la función ajustada, cuyos parámetros se representan en la Tabla 7. De forma semejante, la Figura 103 y la Tabla 8 corresponden al ajuste obtenido del láser.

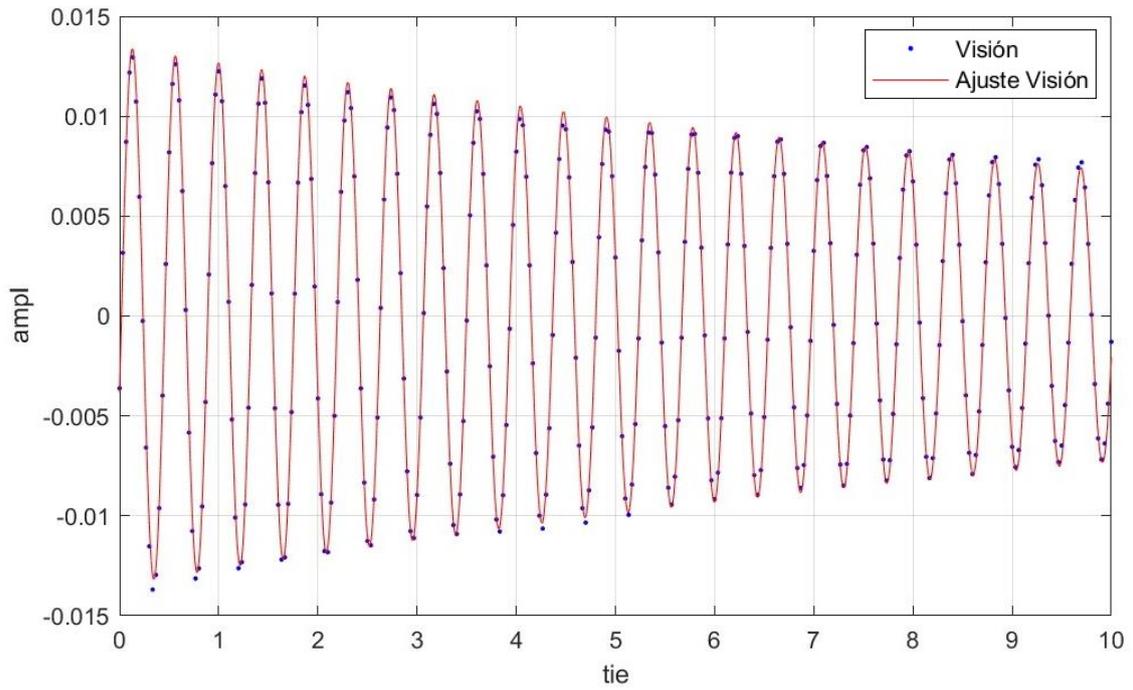


Figura 102: Ajuste mc0 visión

$\text{fitresultvision}(t) = A \cdot \exp(-Z \cdot 2 \cdot \pi \cdot w \cdot t) \cdot \sin(2 \cdot \pi \cdot w \cdot t + f_i)$
Coefficients (with 95% confidence bounds):
A = -0,01348 (-0,01362, -0,01335)
Z = 0,00425 (0,004109, 0,004392)
f _i = -3,428 (-3,438, -3,418)
w = 2,300 (2,300, 2,300)
rsquare: 0,9975

Tabla 7: Resultados mc0 visión

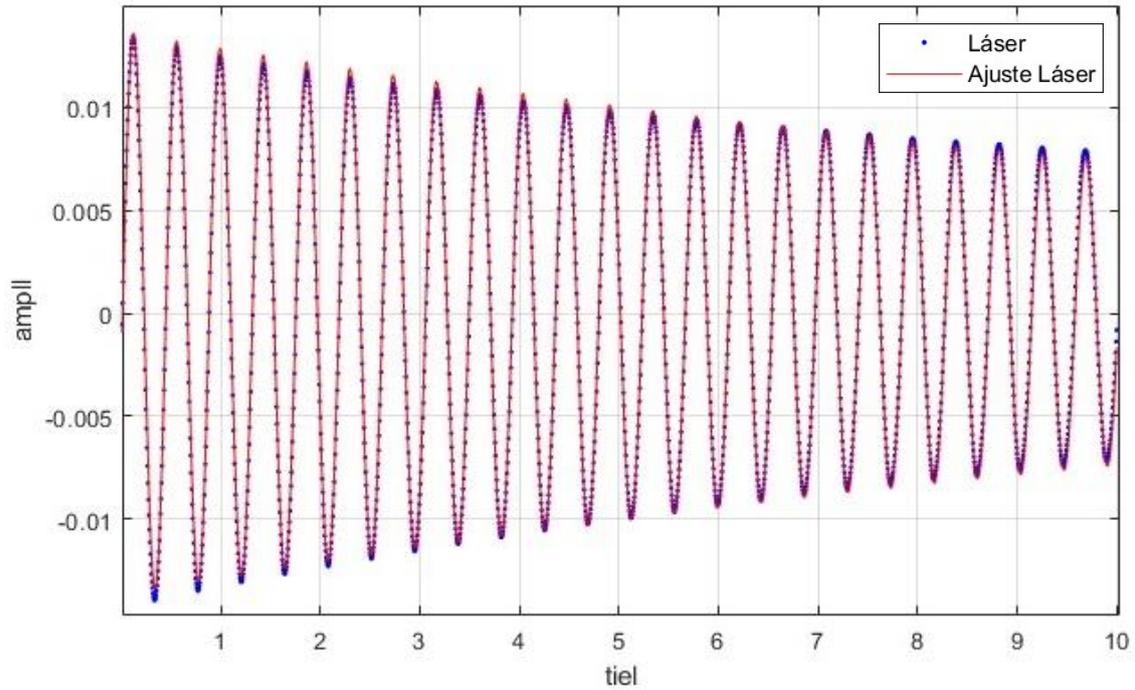


Figura 103: Ajuste mc0 láser

$\text{fitresultláser}(t) = A \cdot \exp(-Z \cdot 2 \cdot \pi \cdot w \cdot t) \cdot \sin(2 \cdot \pi \cdot w \cdot t + f_i)$
Coefficients (with 95% confidence bounds):
A = 0,01371 (0,01366, 0,01376)
Z = 0,004274 (0,004221, 0,004328)
$f_i = -0,2084 (-0,2121, -0,2046)$
w = 2,300 (2,300, 2,300)
rsquare: 0,9976

Tabla 8: Resultados mc0 láser

Atendiendo a los parámetros resultantes, se observa que el error entre las frecuencias obtenidas por ambos métodos es nulo y el del coeficiente de amortiguamiento es muy pequeño, 0,56%, de tal forma que los resultados conseguidos para este experimento son óptimos.

6.5. Medio continuo (plataforma de madera) con masa de 90 kg (mcM)

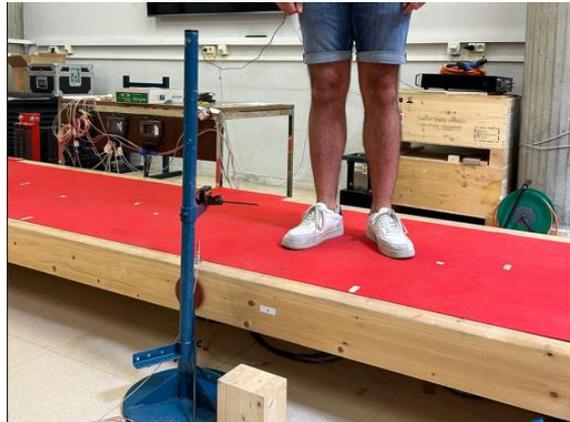


Figura 104: Plataforma de madera, medio continuo con masa de 90 kg

Excitando la estructura de la Figura 104, misma plataforma que la del experimento anterior con una persona encima, se registran los datos, grabando simultáneamente con el láser y la videocámara. Procesando el video registrado se obtiene la gráfica de la Figura 105, y graficando los datos adquiridos por el láser, la Figura 106. Se la misma manera que el experimento anterior, el medio continuo tiene infinitos grados de libertad, pero solo se manifiesta el primero, por lo tanto será ajustado a este grado de libertad.

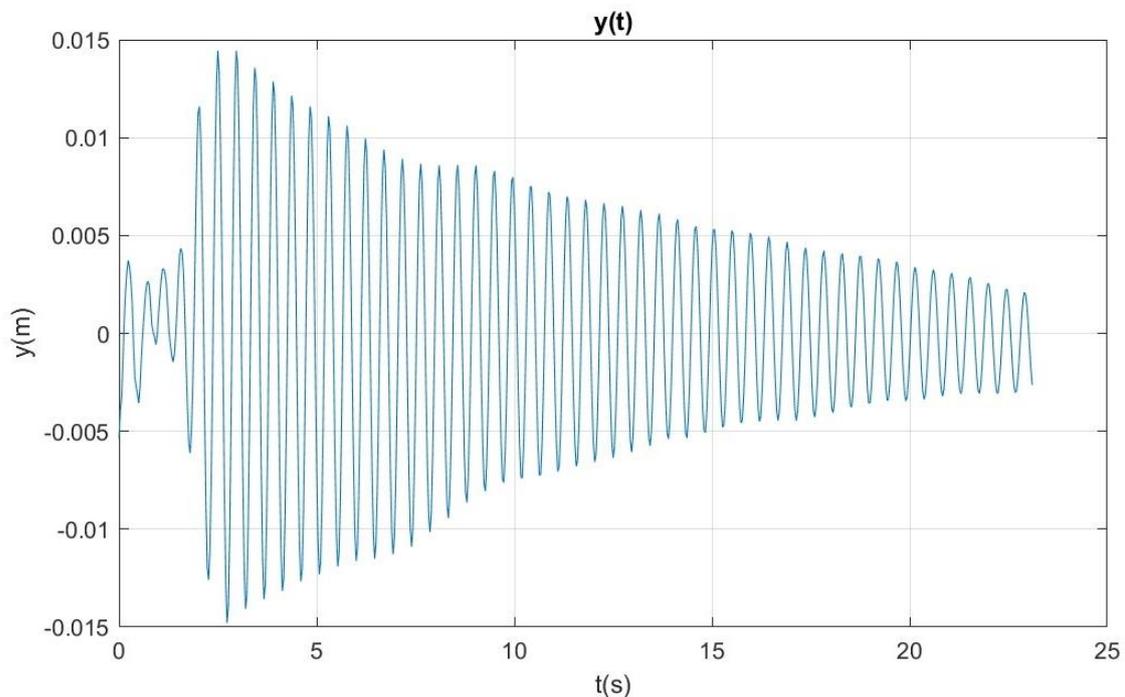


Figura 105: Gráfico del desplazamiento en función del tiempo mcM. Visión

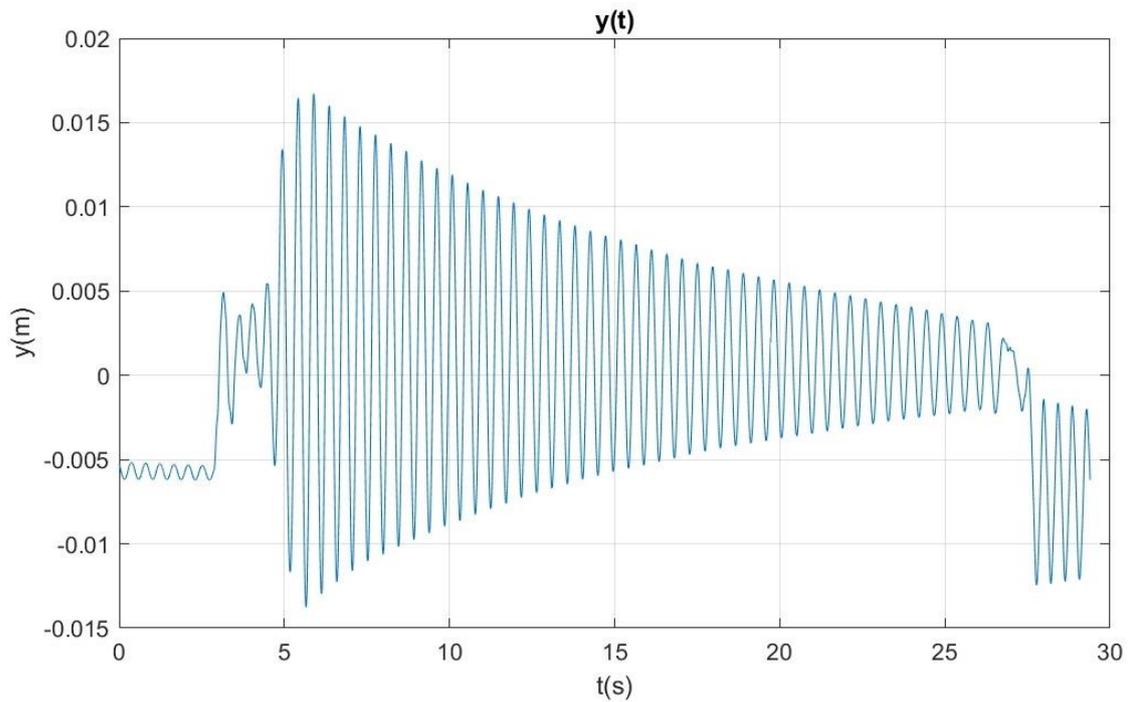


Figura 106: Gráfico del desplazamiento en función del tiempo mcM. Láser

Una vez obtenidos los gráficos del desplazamiento se procede a eliminar el registro previo a la excitación, acotando la función a un intervalo de 15 segundos, resultando las gráficas de la Figura 107 y Figura 108. Superponiendo ambas curvas, se observa (Figura 109) que son semejantes, con alguna pequeña diferencia entre ellas.

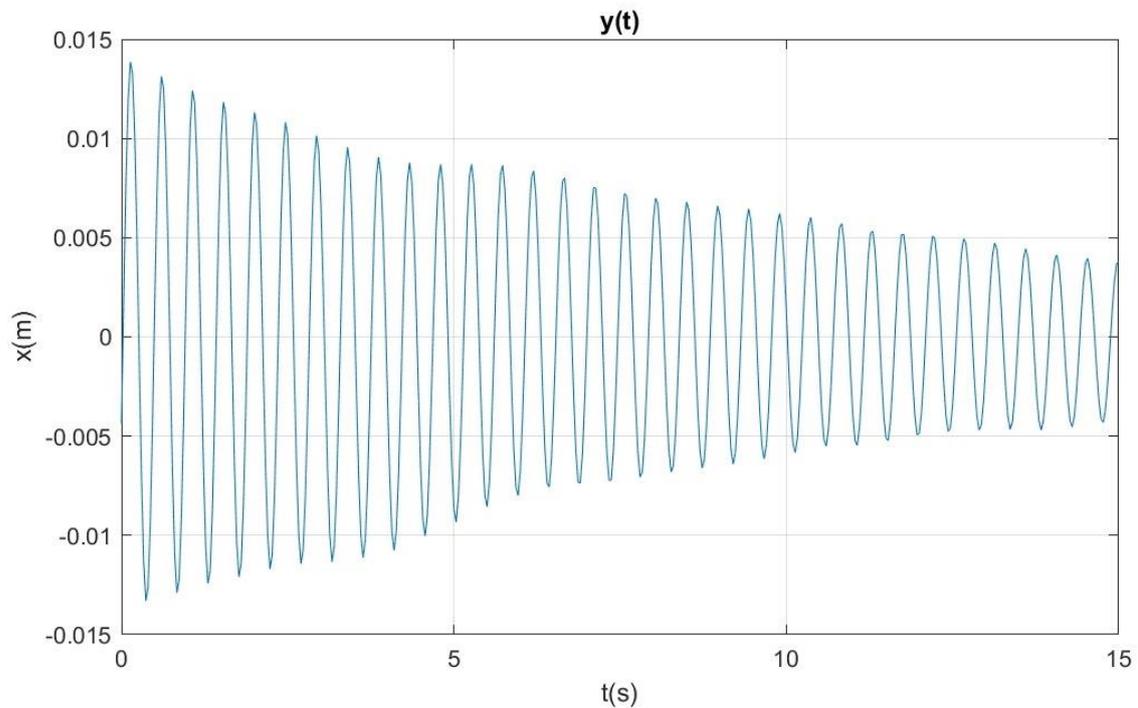


Figura 107: Gráfico acondicionado del desplazamiento en función del tiempo mcM. Visión

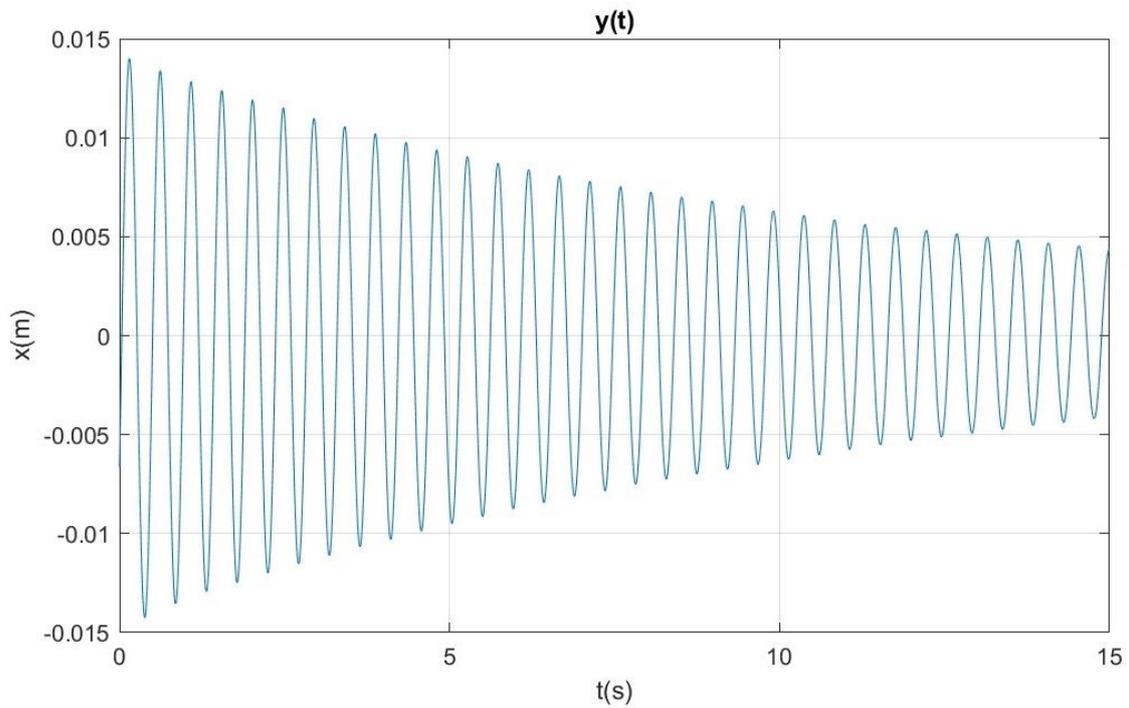


Figura 108: Gráfico acondicionado del desplazamiento en función del tiempo mcM. Láser

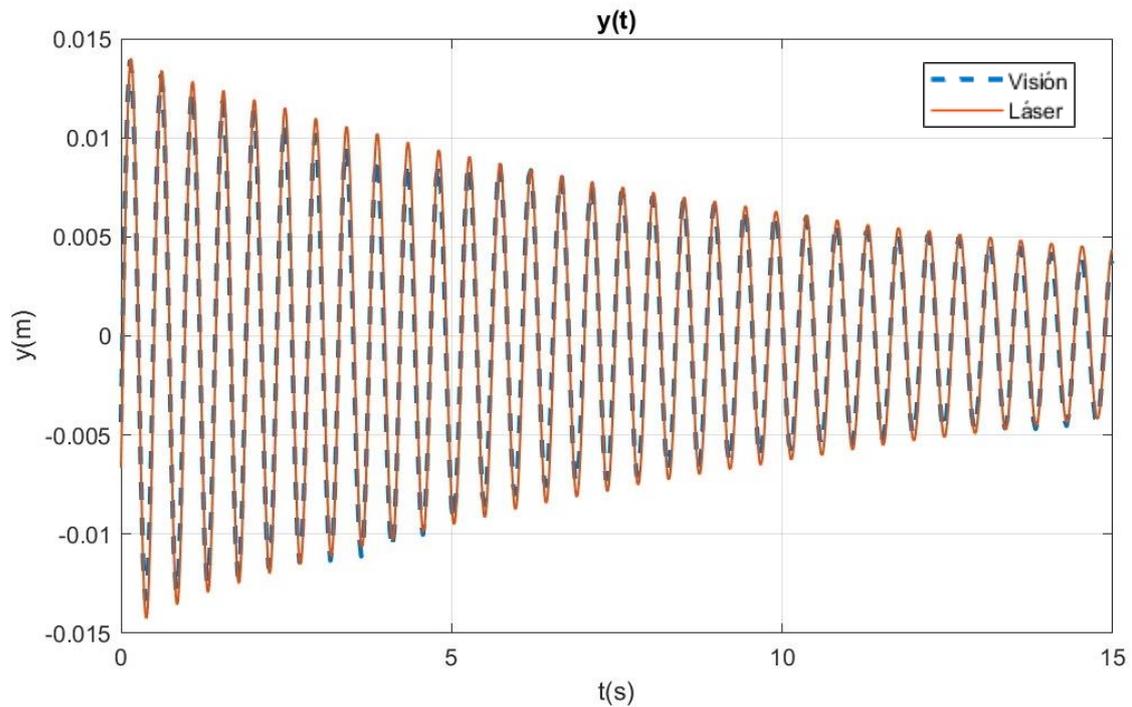


Figura 109: Superposición de gráficos de desplazamiento láser y visión mcM

La Figura 110 muestra el ajuste obtenido, en azul los datos del registro original y en rojo la función ajustada, cuyos parámetros se muestran en la Tabla 9. Análogamente, los resultados del ajuste del láser se exponen en la Figura 111 y la Tabla 10.

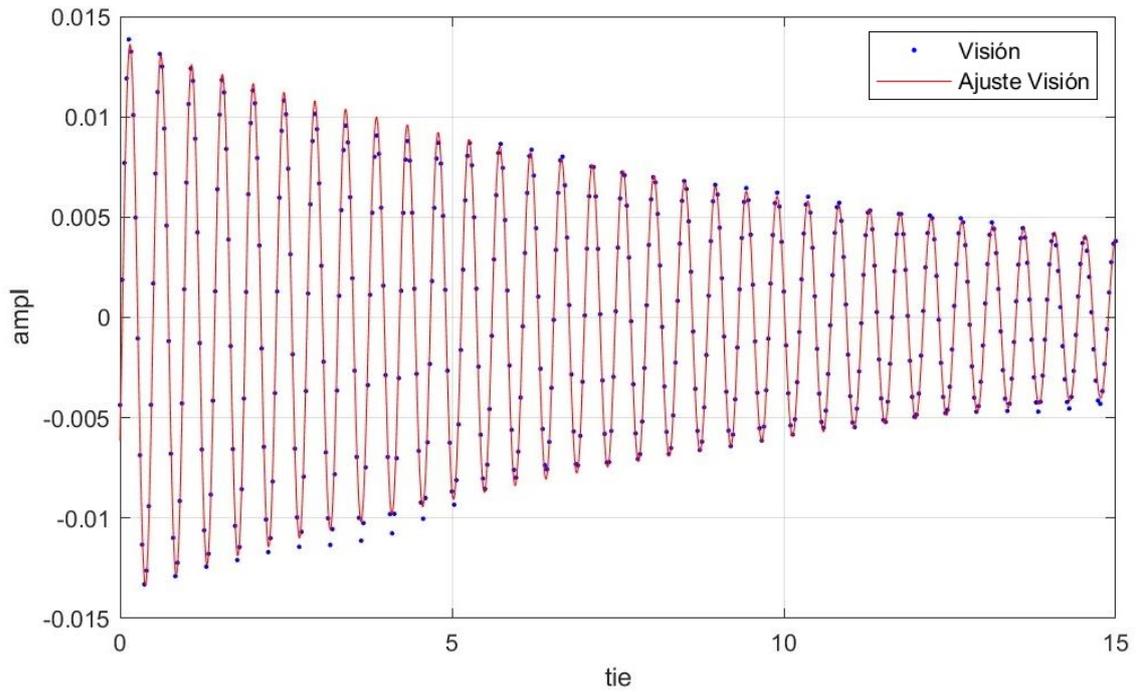


Figura 110: Ajuste mcM visión

$\text{fitresultvision}(t) = A \cdot \exp(-Z \cdot 2 \cdot \pi \cdot w \cdot t) \cdot \sin(2 \cdot \pi \cdot w \cdot t + f_i)$
Coefficients (with 95% confidence bounds):
A = 0,0138 (0,01362, 0,01398)
Z = 0,006155 (0,005994, 0,006315)
$f_i = -0,4625 (-0,4753, -0,4497)$
w = 2,154 (2,154, 2,155)
rsquare: 0,9924

Tabla 9: Resultados mcM visión

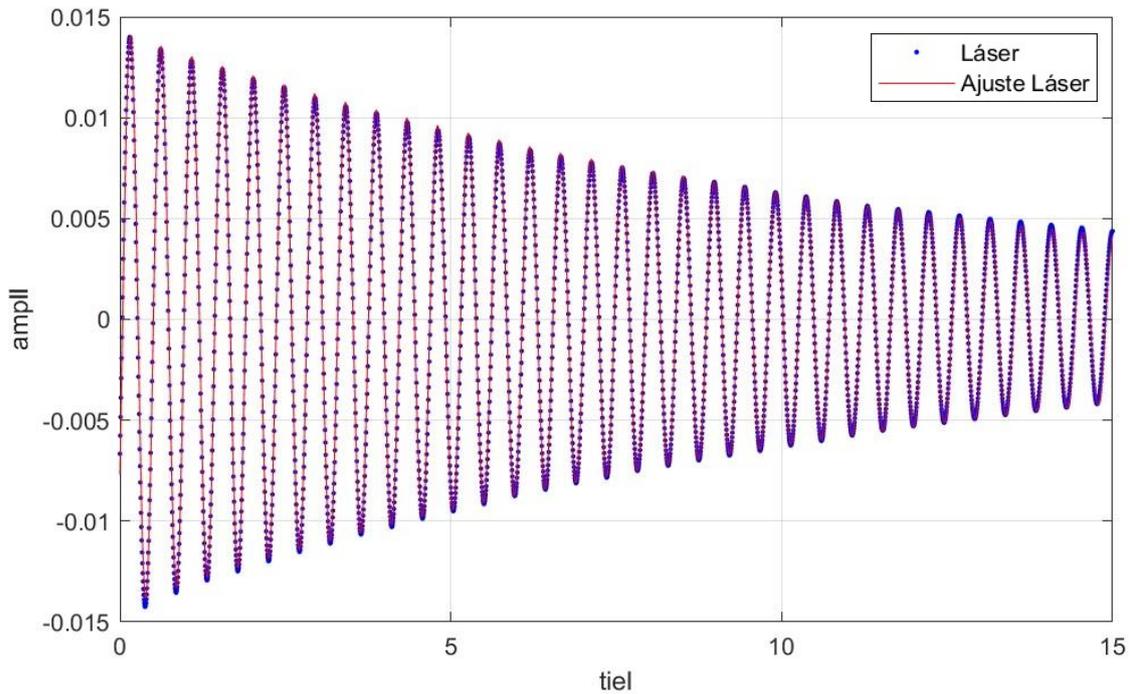


Figura 111: Ajuste mcM láser

$\text{fitresultláser}(t) = A \cdot \exp(-Z \cdot 2 \cdot \pi \cdot w \cdot t) \cdot \sin(2 \cdot \pi \cdot w \cdot t + f_i)$	
Coefficients (with 95% confidence bounds):	
A =	-0,01426 (-0,01432, -0,0142)
Z =	0,00607 (0,00602, 0,006119)
f _i =	-3,711 (-3,715, -3,707)
w =	2,154 (2,154, 2,154)
rsquare: 0,9951	

Tabla 10: Resultados mcM láser

Calculando los errores relativos de los parámetros más representativos para la caracterización modal entre ambos métodos, se observa que el error en la estimación de las frecuencias es nulo mientras que el del coeficiente de amortiguamiento es pequeño, en este caso 1,4%, por lo tanto se puede decir que los datos obtenidos son buenos.

Es notable que el amortiguamiento crece al colocar una masa (en este caso una persona) sobre la plataforma. Se puede ver que en el experimento mc0, plataforma sin masas, era alrededor de 0,0043% mientras que en este caso, añadiendo una masa, crece hasta 0,006% aproximadamente. La frecuencia, por el contrario, disminuye de 2,3 a 2,15 Hz.

6.6. Estructura de 2 grados de libertad con dos masas de 25/8 kg (2p2m)



Figura 112: Estructura de 2 gdl con dos masas de 25/8 kg

Para este experimento, se excita la estructura de dos grados de libertad de la Figura 112 y se graba simultáneamente con el láser y la videocámara, obteniendo las gráficas de la Figura 113 y Figura 114 para visión y láser respectivamente. Se realizará primero el ajuste a un grado de libertad, obteniendo ciertos parámetros modales, que se compararán más tarde con los obtenidos en el ajuste a los dos grados de libertad.

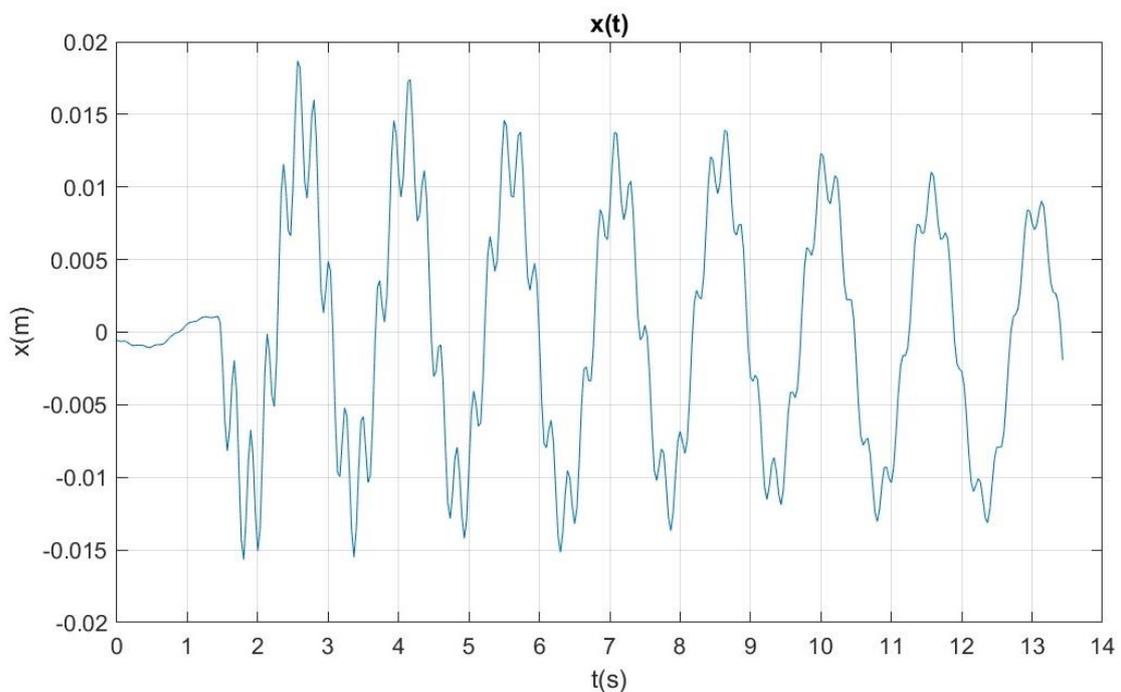


Figura 113: Gráfico del desplazamiento en función del tiempo 2p2m. Visión

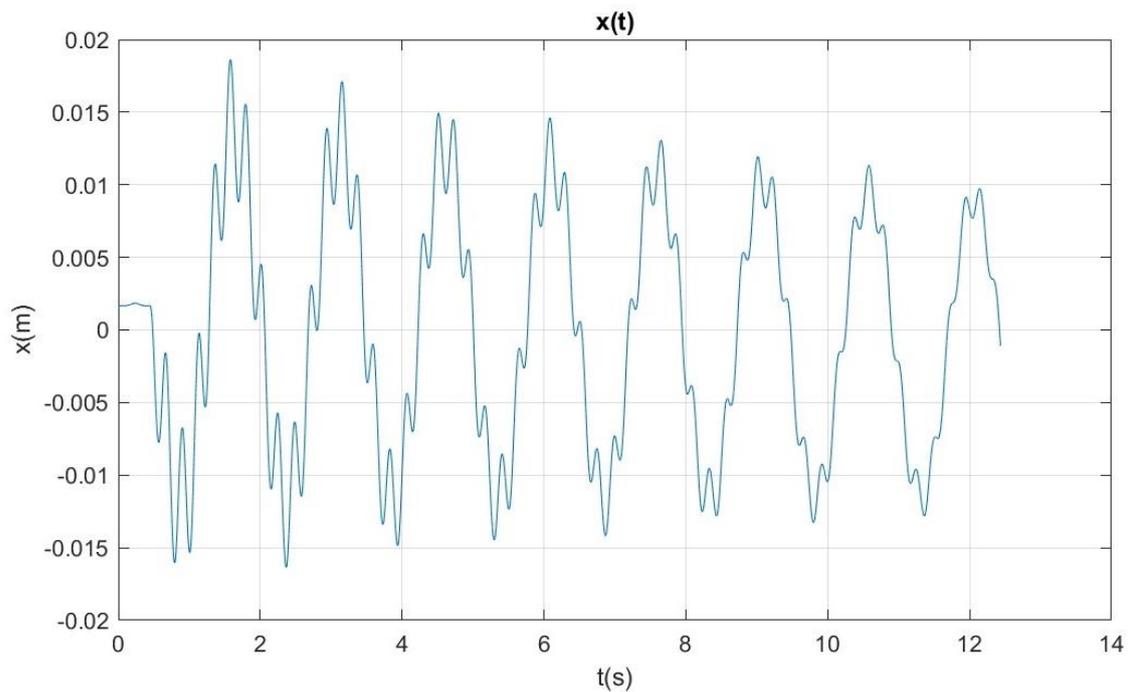


Figura 114: Gráfico del desplazamiento en función del tiempo 2p2m. Láser

De la misma forma que para los experimentos anteriores, se acondicionan los registros obtenidos, acotando la función a un intervalo de 10 segundos, resultando las gráficas de la Figura 115 y Figura 116. Superponiendo ambas, Figura 117, se observa que los registros son semejantes, excepto por pequeñas diferencias.

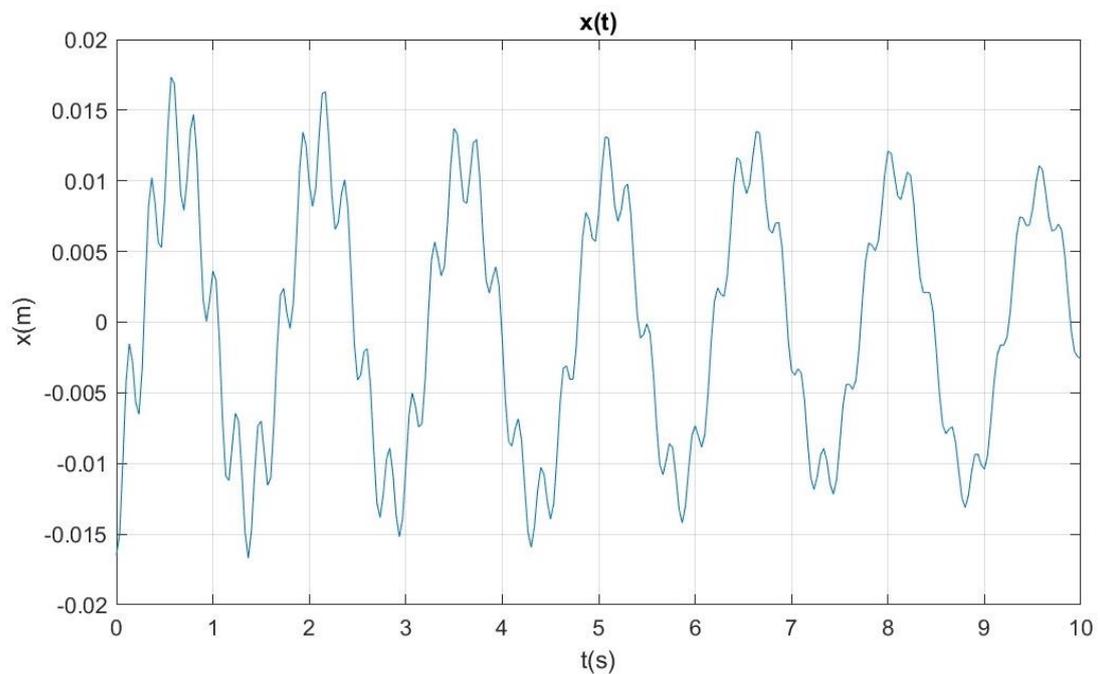


Figura 115: Gráfico acondicionado del desplazamiento en función del tiempo 2p2m. Visión

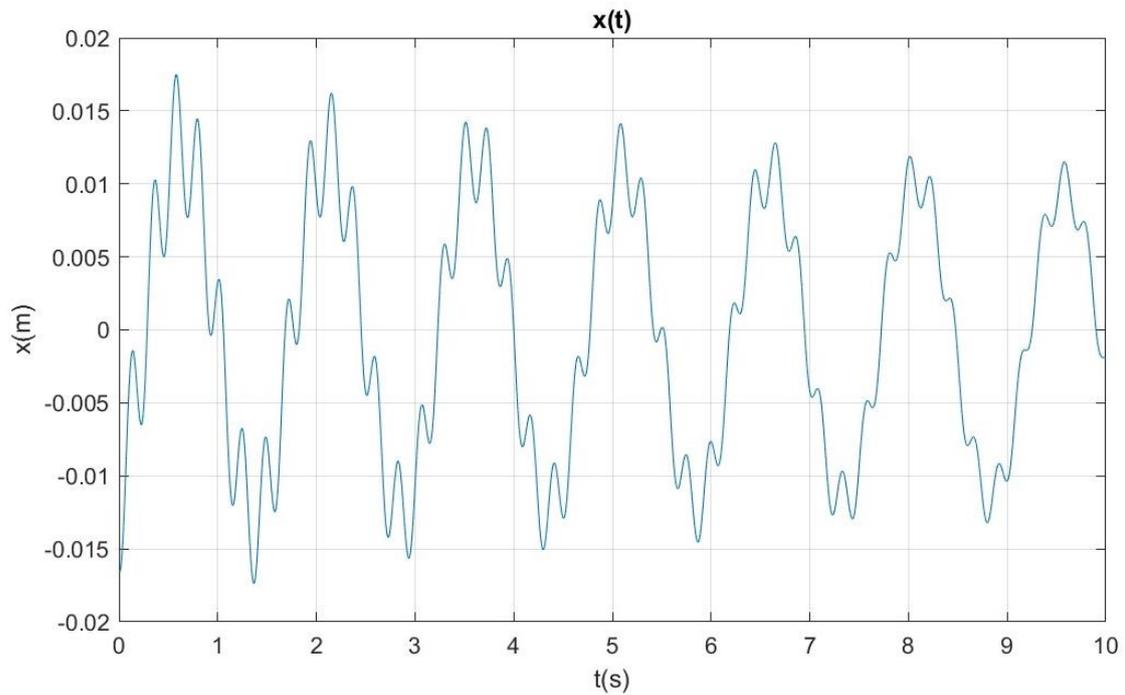


Figura 116: Gráfico acondicionado del desplazamiento en función del tiempo 2p2m. Láser

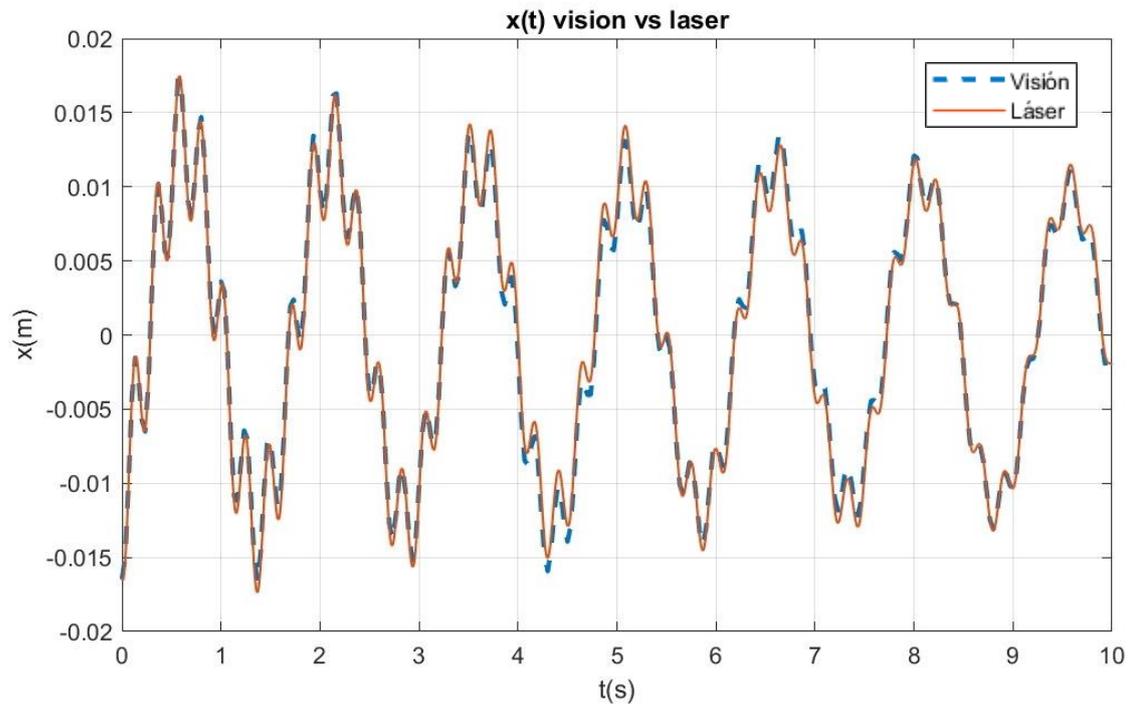


Figura 117: Superposición de gráficos de desplazamiento láser y visión 2p2m

Los gráficos y parámetros obtenidos para el ajuste a un grado de libertad para el análisis por visión se muestran en la Figura 118 y Tabla 11, mientras que los obtenidos para los datos registrados por el láser se muestran en la Figura 119 y Tabla 12.

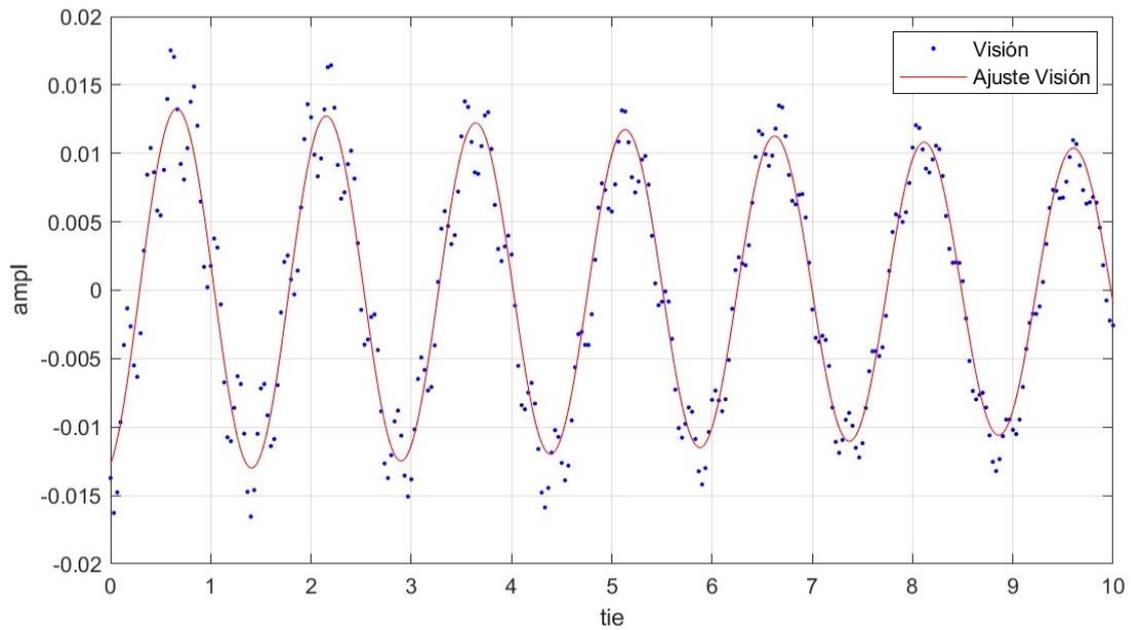


Figura 118: Ajuste 2p2m a 1 gdl. Visión

$\text{fitresultvision}(t) = A \cdot \exp(-Z \cdot 2 \cdot \pi \cdot w \cdot t) \cdot \sin(2 \cdot \pi \cdot w \cdot t + f_i)$
Coefficients (with 95% confidence bounds):
A = -0,0135 (-0,01432, -0,01268)
Z = 0,006479 (0,003802, 0,009157)
f _i = 1,916 (1,857, 1,975)
w = 0,671 (0,6692, 0,6727)
rsquare: 0,9270

Tabla 11: Resultados 2p2m visión 1gdl

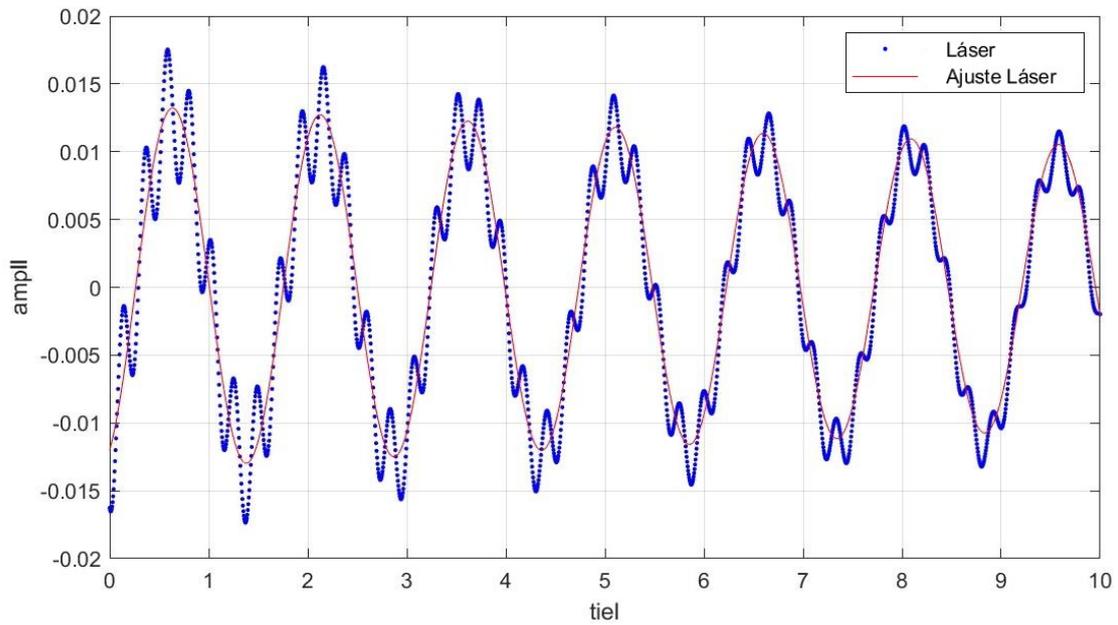


Figura 119: Ajuste 2p2m a 1 gdl. Láser

$\text{fitresultláser}(t) = A1 \cdot \exp(-Z1 \cdot 2 \cdot \pi \cdot w1 \cdot t) \cdot \sin(2 \cdot \pi \cdot w1 \cdot t + fi1)$
Coefficients (with 95% confidence bounds):
A1 = 0,01343 (0,01312, 0,01374)
Z1 = 0,006001 (0,004993, 0,007009)
fi1 = -1,106 (-1,128, -1,084)
w1 = 0,6708 (0,6701, 0,6714)
rsquare: 0,9305

Tabla 12: Resultados 2p2m láser 1gdl

Como se ha mencionado, tras realizar el ajuste a un grado de libertad se obtienen frecuencias casi idénticas por los dos métodos, con un error tan solo de 0,03%, mientras que el error en el coeficiente de amortiguamiento es peor, en este caso de 7,96%.

A continuación se realiza el ajuste a dos grados de libertad, del que obtenemos las gráficas mostradas en la Figura 120 y Figura 121 y los parámetros modales mostrados en la Tabla 13 y Tabla 14, respectivamente para los registros de visión y del láser.

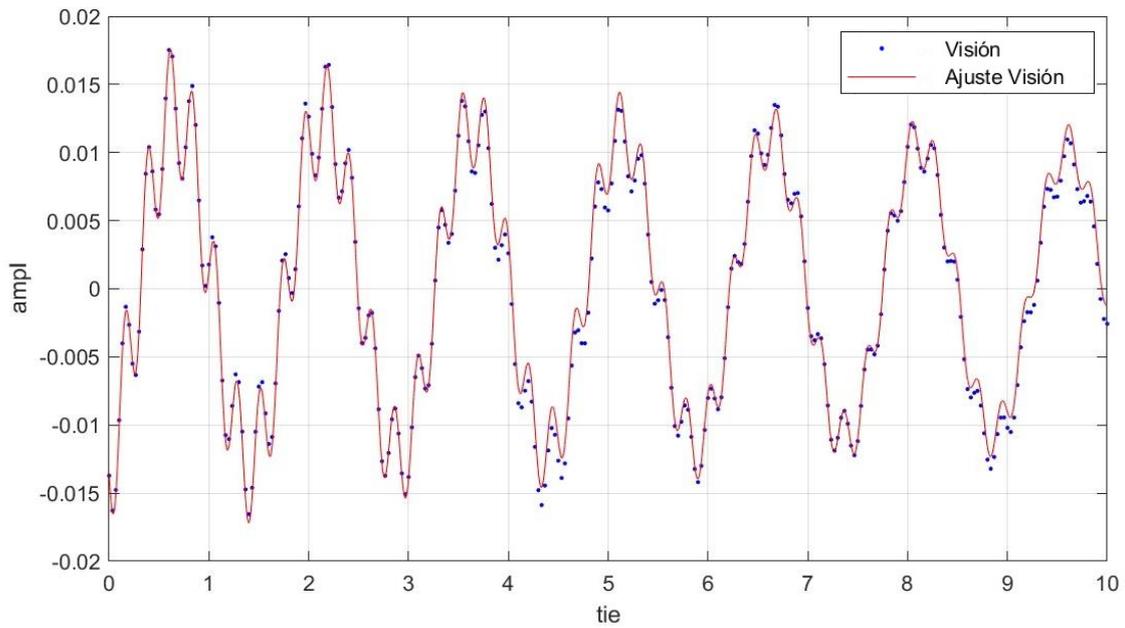


Figura 120: Ajuste 2p2m a 2 gdl. Visión

$\text{fitresultvision}(t) = A1 \cdot \exp(-Z1 \cdot 2 \cdot \pi \cdot w1 \cdot t) \cdot \sin(2 \cdot \pi \cdot w1 \cdot t + fi1) + A2 \cdot \exp(-Z2 \cdot 2 \cdot \pi \cdot w2 \cdot t) \cdot \sin(2 \cdot \pi \cdot w2 \cdot t + fi2)$
Coefficients (with 95% confidence bounds):
A1 = 0,01334 (0,01309, 0,01359)
A2 = -0,00512 (-0,005422, -0,004818)
Z1 = 0,006052 (0,005228, 0,006876)
Z2 = 0,00429 (0,00377, 0,004811)
fi1 = -1,221 (-1,239, -1,202)
fi2 = 0,2253 (0,1668, 0,2838)
w1 = 0,6708 (0,6703, 0,6713)
w2 = 4,443 (4,44, 4,445)
rsquare: 0,9932

Tabla 13: Resultados 2p2m visión 2gdl

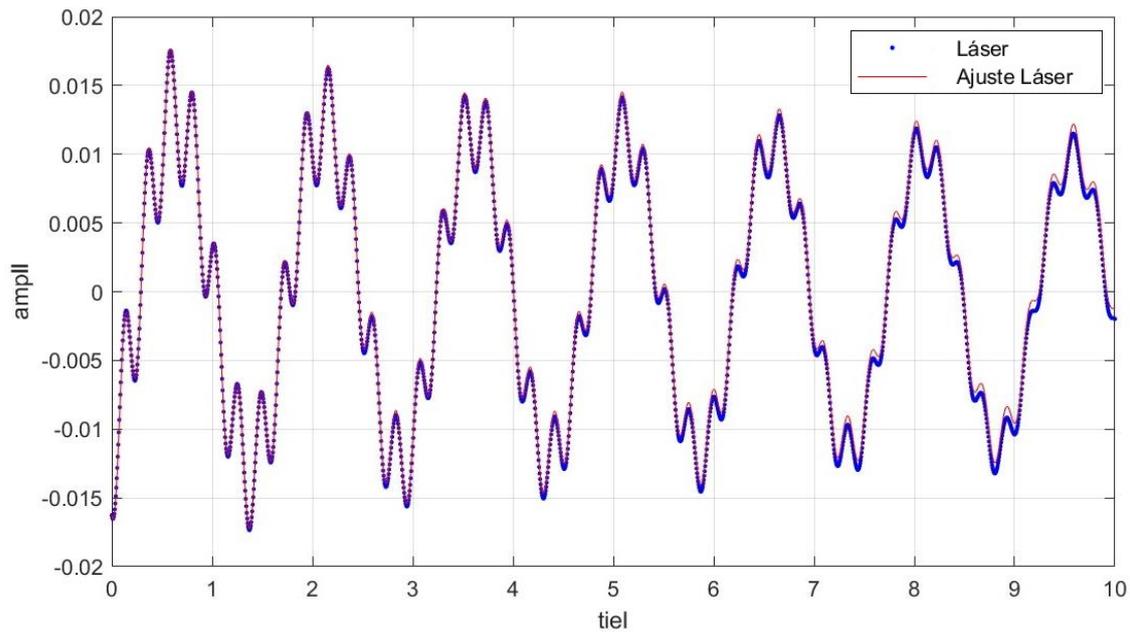


Figura 121: Ajuste 2p2m a 2 gdl. Láser

$\text{fitresultláser}(t) = A1 \cdot \exp(-Z1 \cdot 2 \cdot \pi \cdot w1 \cdot t) \cdot \sin(2 \cdot \pi \cdot w1 \cdot t + fi1) + A2 \cdot \exp(-Z2 \cdot 2 \cdot \pi \cdot w2 \cdot t) \cdot \sin(2 \cdot \pi \cdot w2 \cdot t + fi2)$
Coefficients (with 95% confidence bounds):
A1 = 0,01333 (0,01327, 0,01339)
A2 = 0,005147 (0,005077, 0,005217)
Z1 = 0,005727 (0,005535, 0,005918)
Z2 = 0,004323 (0,004203, 0,004444)
fi1 = -1,103 (-1,107, -1,099)
fi2 = -2,145 (-2,159, -2,132)
w1 = 0,6706 (0,6705, 0,6708)
w2 = 4,442 (4,442, 4,443)
rsquare: 0,9975

Tabla 14: Resultados 2p2m láser 2gdl

Se observa que la frecuencia para el primer grado de libertad analizado es bastante similar en los dos ajustes siendo 0,671 y 0,6708 Hz para visión y 0,6708 y 0,6706 para el láser. Los coeficientes de amortiguamiento son 0,6479% y 0,6052% para visión y 0,6001% y 0,5727% en el caso del láser, siendo ambos resultados razonablemente parecidos.

Comparando los resultados obtenidos por los dos métodos para el ajuste a dos grados de libertad, se comprueba que los errores relativos entre las frecuencias son muy pequeños: 0,03% y 0,02%, y para el coeficiente de amortiguamiento el error es un poco más grande, pero aceptable: 5,67% y 0,76%.

6.7. Estructura de 3 grados de libertad sin masas (3p0m)



Figura 122: Estructura de 3 gdl sin masas

En el experimento 3p0m se excita la estructura de la Figura 122, de tres grados de libertad, aunque como se verá más tarde, solo se manifestarán dos de ellos por la forma de dar el impulso. De la misma manera que en los casos anteriores se graba simultáneamente con el láser y la videocámara, y tras procesar el vídeo se obtienen las gráficas de la Figura 123 para visión y Figura 124 para el láser.

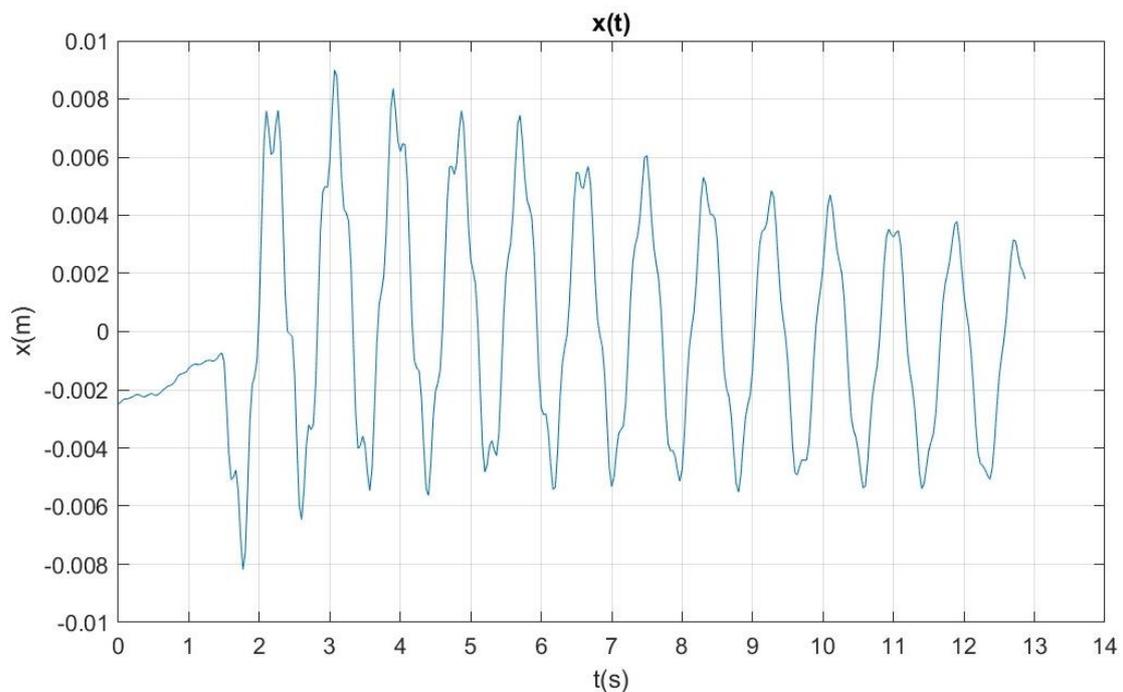


Figura 123: Gráfico del desplazamiento en función del tiempo 3p0m. Visión

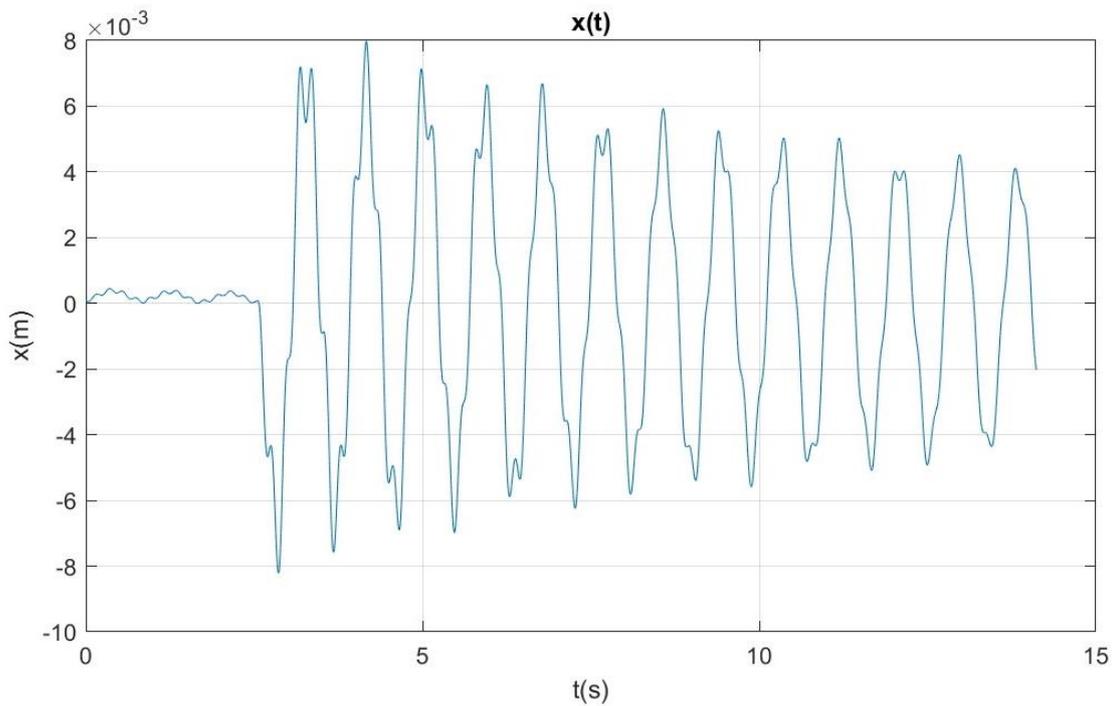


Figura 124: Gráfico del desplazamiento en función del tiempo 3p0m. Láser

Se procede al acondicionamiento de los datos, se elimina el registro previo a la excitación y se acota la función a un intervalo de 10 segundos, obteniendo las gráficas de la Figura 125 y Figura 126. En la Figura 127 se observan curvas similares con pequeñas diferencias, por lo que al realizar el ajuste se obtendrán resultados que podrán diferir.

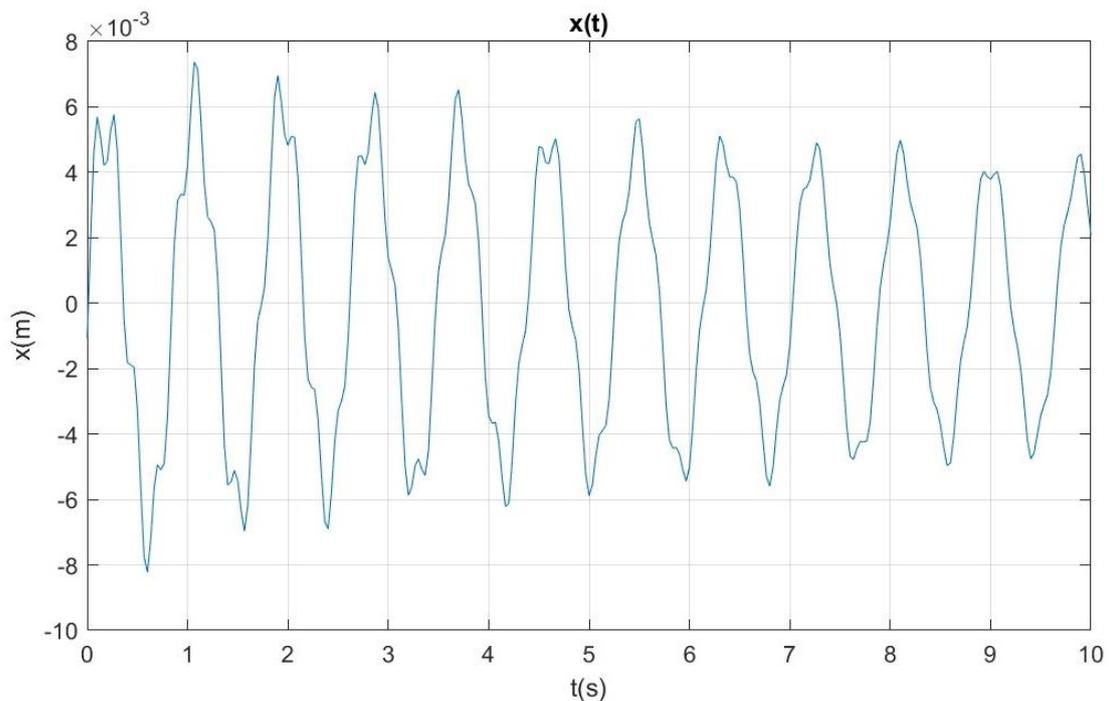


Figura 125: Gráfico acondicionado del desplazamiento en función del tiempo 3p0m. Visión

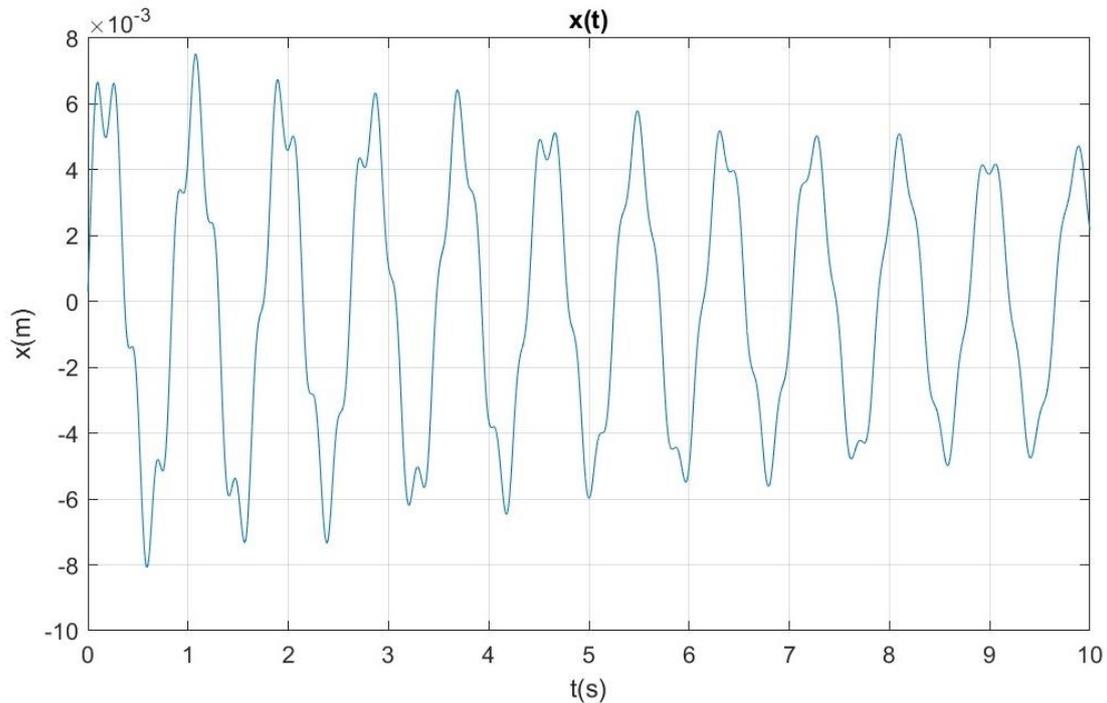


Figura 126: Gráfico acondicionado del desplazamiento en función del tiempo 3p0m. Láser

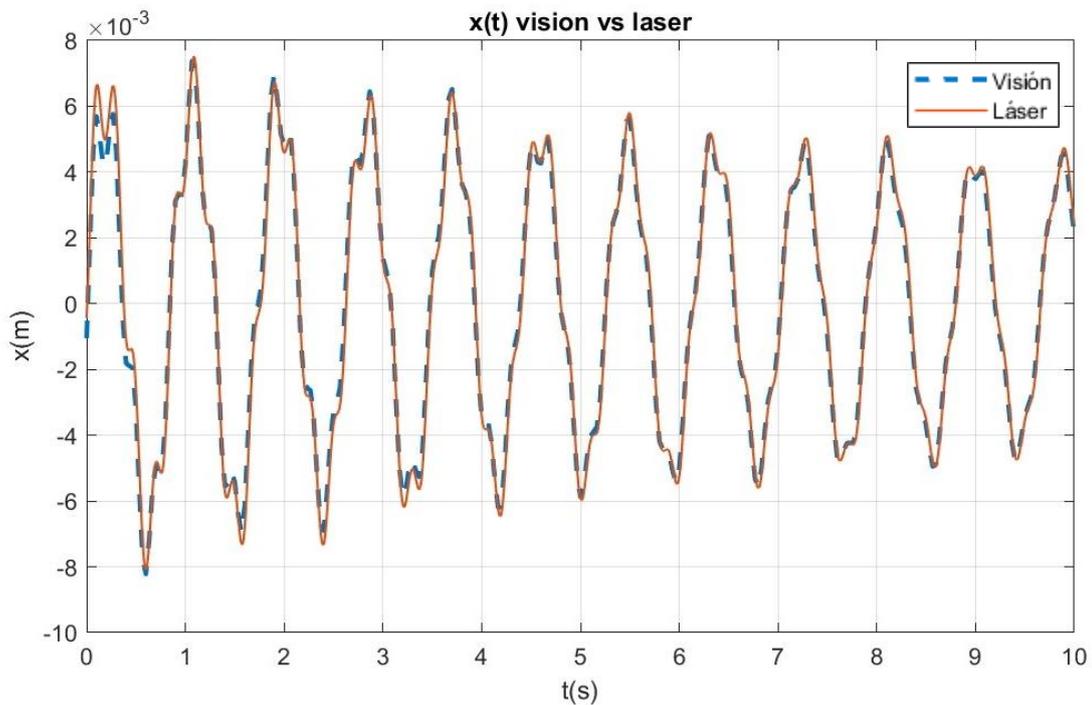


Figura 127: Superposición de gráficos de desplazamiento láser y visión 3p0m

Como en el experimento anterior, primero se ajusta la curva a un grado de libertad para comparar los resultados, obteniendo la gráfica de la Figura 128 y Tabla 15 para visión y la gráfica de la Figura 129 y Tabla 16 para el láser. Se aprecia un error relativo nulo entre las frecuencias y un error de 7,38% para los coeficientes de amortiguamiento.

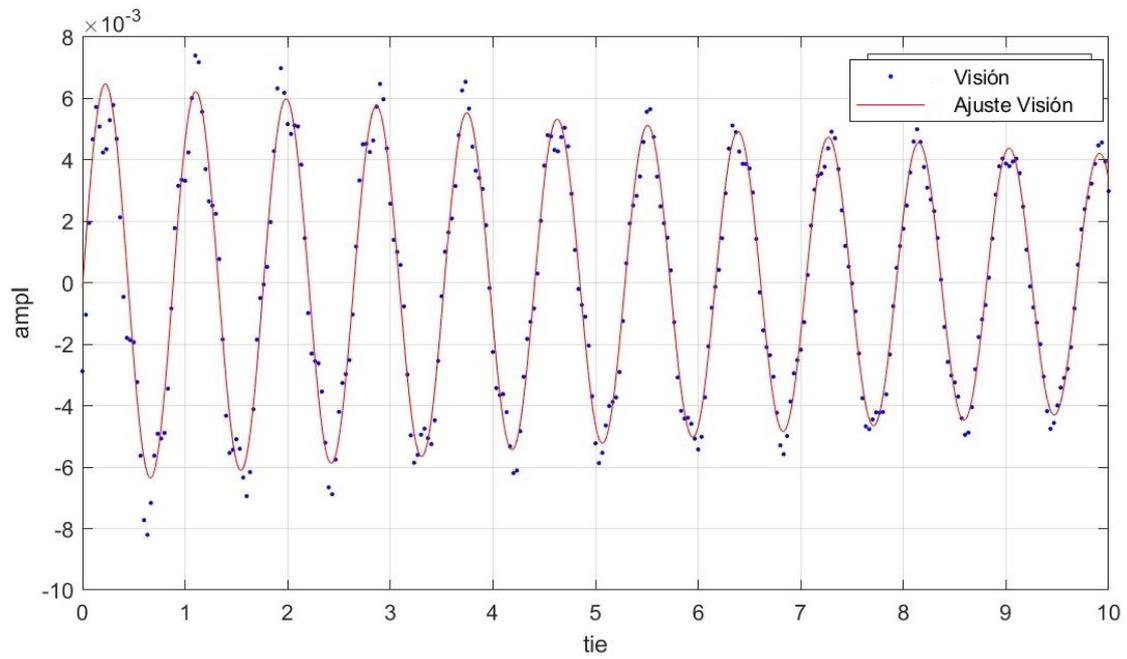


Figura 128: Ajuste 3p0m a 1 gdl. Visión

$\text{fitresultvision}(t) = A1 \cdot \exp(-Z1 \cdot 2 \cdot \pi \cdot w1 \cdot t) \cdot \sin(2 \cdot \pi \cdot w1 \cdot t + fi1)$
Coefficients (with 95% confidence bounds):
A1 = -0,006526 (-0,006778, -0,006275)
Z1 = 0,006216 (0,005167, 0,007264)
fi1 = -2,917 (-2,956, -2,878)
w1 = 1,135 (1,134, 1,136)
rsquare: 0,9661

Tabla 15: Resultados 3p0m visión 1gdl

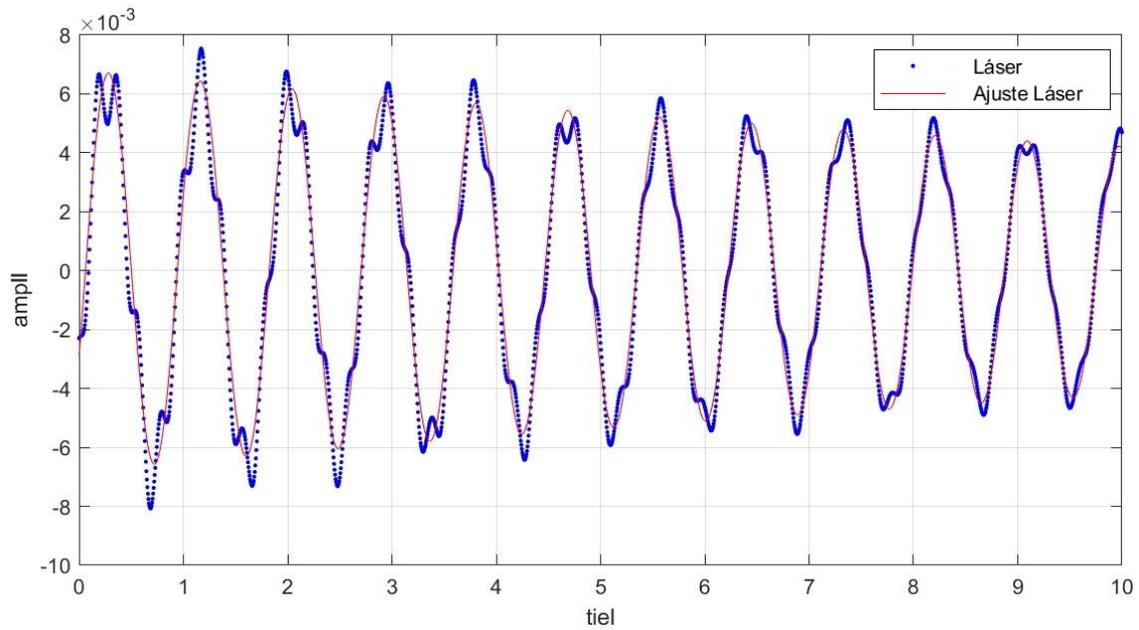


Figura 129: Ajuste 3p0m a 1 gdl. Láser

$\text{fitresultláser}(t) = A1 \cdot \exp(-Z1 \cdot 2 \cdot \pi \cdot w1 \cdot t) \cdot \sin(2 \cdot \pi \cdot w1 \cdot t + fi1)$
Coefficients (with 95% confidence bounds):
A1 = 0,006774 (0,006681, 0,006868)
Z1 = 0,006711 (0,00633, 0,007092)
fi1 = 0,1776 (0,1635, 0,1917)
w1 = 1,135 (1,134, 1,135)
rsquare: 0,9699

Tabla 16: Resultados 3p0m láser 1gdl

A continuación se realiza el ajuste a dos grados de libertad, obteniendo las gráficas representadas en la Figura 130 y Figura 131, y los parámetros mostrados en la Tabla 17 y Tabla 18, por visión y láser respectivamente.

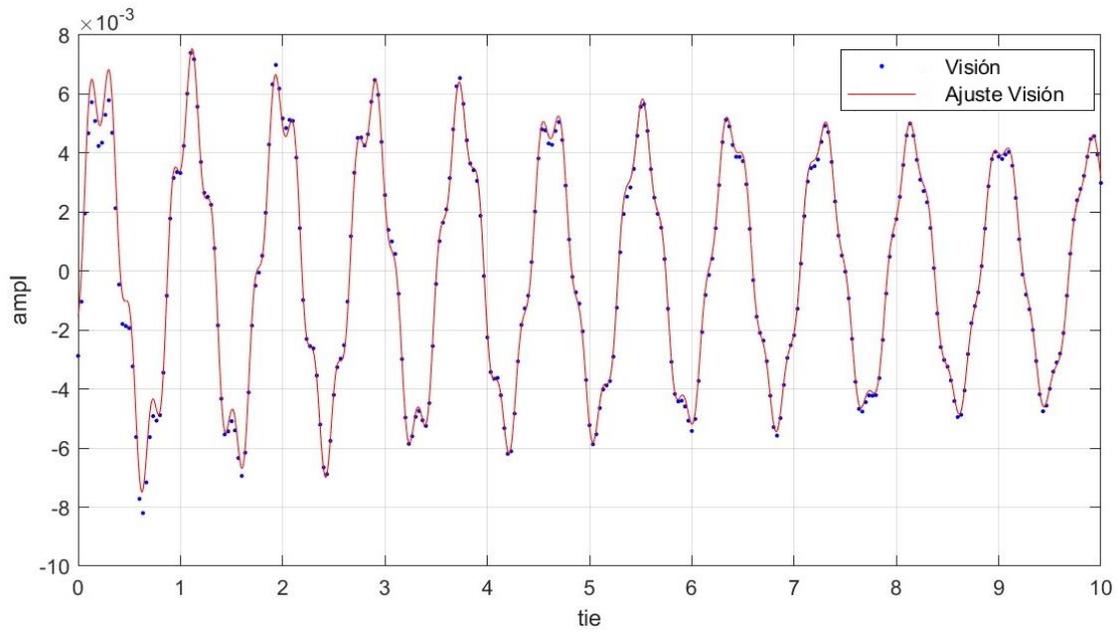


Figura 130: Ajuste 3p0m a 2 gdl. Visión

$\text{fitresultvision}(t) = A1 \cdot \exp(-Z1 \cdot 2 \cdot \pi \cdot w1 \cdot t) \cdot \sin(2 \cdot \pi \cdot w1 \cdot t + fi1) + A2 \cdot \exp(-Z2 \cdot 2 \cdot \pi \cdot w2 \cdot t) \cdot \sin(2 \cdot \pi \cdot w2 \cdot t + fi2)$	
Coefficients (with 95% confidence bounds):	
A1 =	-0,006514 (-0,006619, -0,00641)
A2 =	0,001509 (0,001384, 0,001634)
Z1 =	0,00617 (0,005734, 0,006606)
Z2 =	0,004052 (0,003387, 0,004717)
fi1 =	3,367 (3,351, 3,383)
fi2 =	-1,002 (-1,084, -0,9191)
w1 =	1,135 (1,135, 1,136)
w2 =	4,993 (4,99, 4,997)

Tabla 17: Resultados 3p0m visión 2gdl

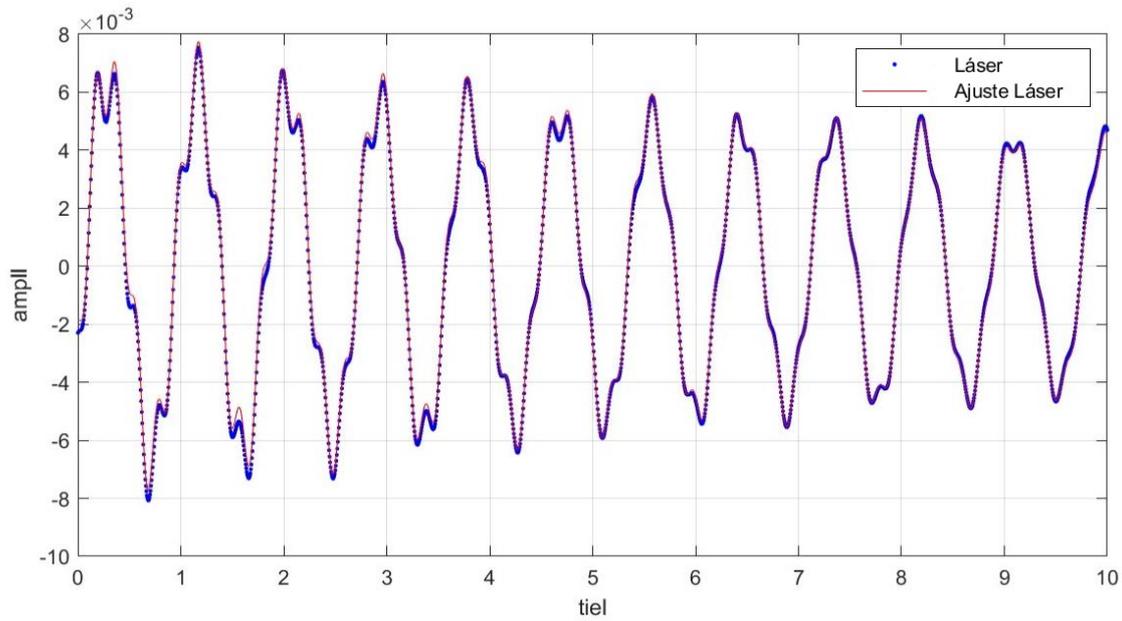


Figura 131: Ajuste 3p0m a 2 gdl. Láser

$\text{fitresultláser}(t) = A1 \cdot \exp(-Z1 \cdot 2 \cdot \pi \cdot w1 \cdot t) \cdot \sin(2 \cdot \pi \cdot w1 \cdot t + fi1) + A2 \cdot \exp(-Z2 \cdot 2 \cdot \pi \cdot w2 \cdot t) \cdot \sin(2 \cdot \pi \cdot w2 \cdot t + fi2)$
Coefficients (with 95% confidence bounds):
A1 = -0,006762 (-0,00679, -0,006735)
A2 = 0,001535 (0,001503, 0,001568)
Z1 = 0,006674 (0,006564, 0,006785)
Z2 = 0,004083 (0,003912, 0,004253)
fi1 = 3,317 (3,313, 3,321)
fi2 = -1,175 (-1,196, -1,154)
w1 = 1,135 (1,135, 1,135)
w2 = 4,992 (4,991, 4,993)
rsquare: 0,9975

Tabla 18: Resultados 3p0m láser 2gdl

Atendiendo a los parámetros más representativos, se obtiene un error nulo para primera frecuencia estimada y un error de 0,02% para la segunda. Analizando el amortiguamiento, como en el caso anterior, se obtiene un error mayor para el primer modo, de 7,55% y uno menor para el segundo, de 0,76%.

6.8. Estructura de 4 grados de libertad sin masas (4p0m)



Figura 132: Estructura de 4 gdl sin masas

Para el experimento 4p0m, obtenido tras excitar la estructura de la Figura 132 y grabar simultáneamente con el láser y el dispositivo móvil, se obtiene la gráfica de la Figura 133 tras procesar los fotogramas. El registro obtenido por el láser se muestra en la Figura 134. Esta estructura posee cuatro grados de libertad, aunque solo se manifestarán tres de ellos por la forma de aplicar el impulso a la misma.

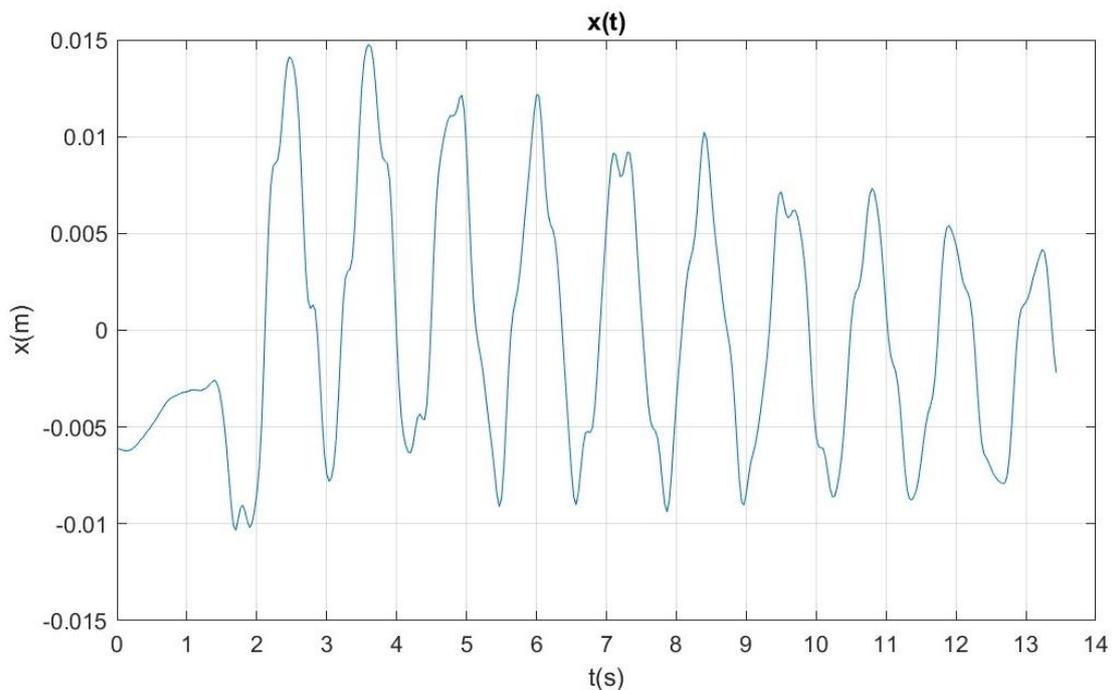


Figura 133: Gráfico del desplazamiento en función del tiempo 4p0m. Visión

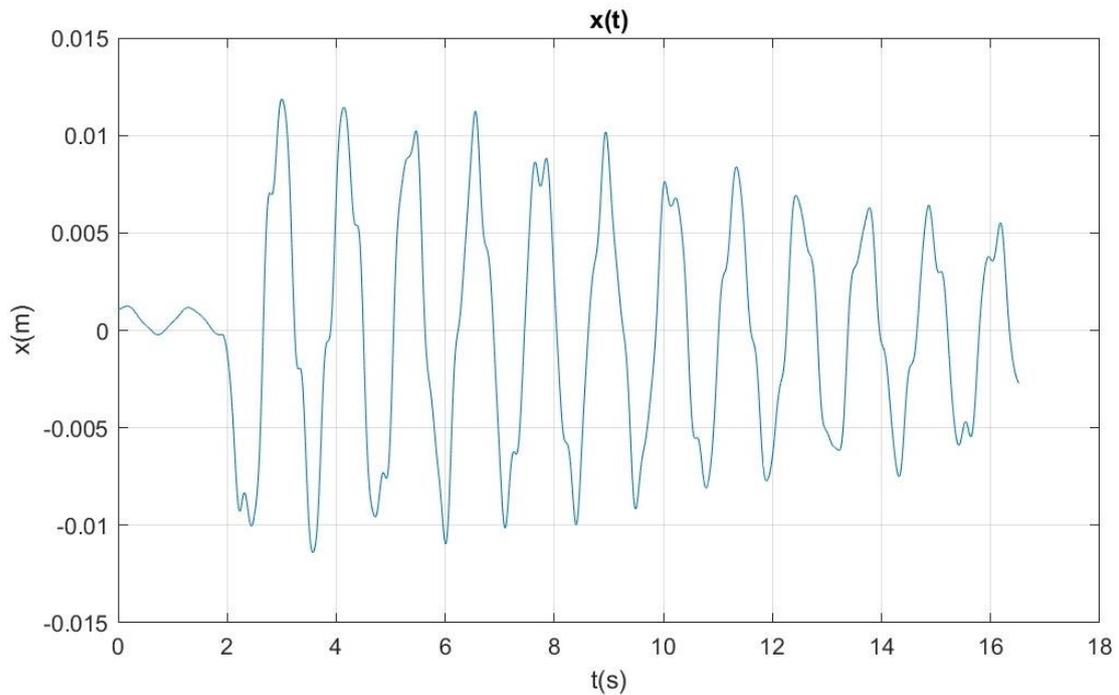


Figura 134: Gráfico del desplazamiento en función del tiempo 4p0m. Láser

Tras acondicionar los datos registrados por ambos métodos, eliminando el registro previo a la excitación y acotando la función a un intervalo de 10 segundos, se obtienen las gráficas de la Figura 135 y Figura 136. Superponiendo ambas curvas, Figura 137, se ven pequeñas diferencias entre ellas, lo que puede provocar que los resultados varíen entre los dos métodos.

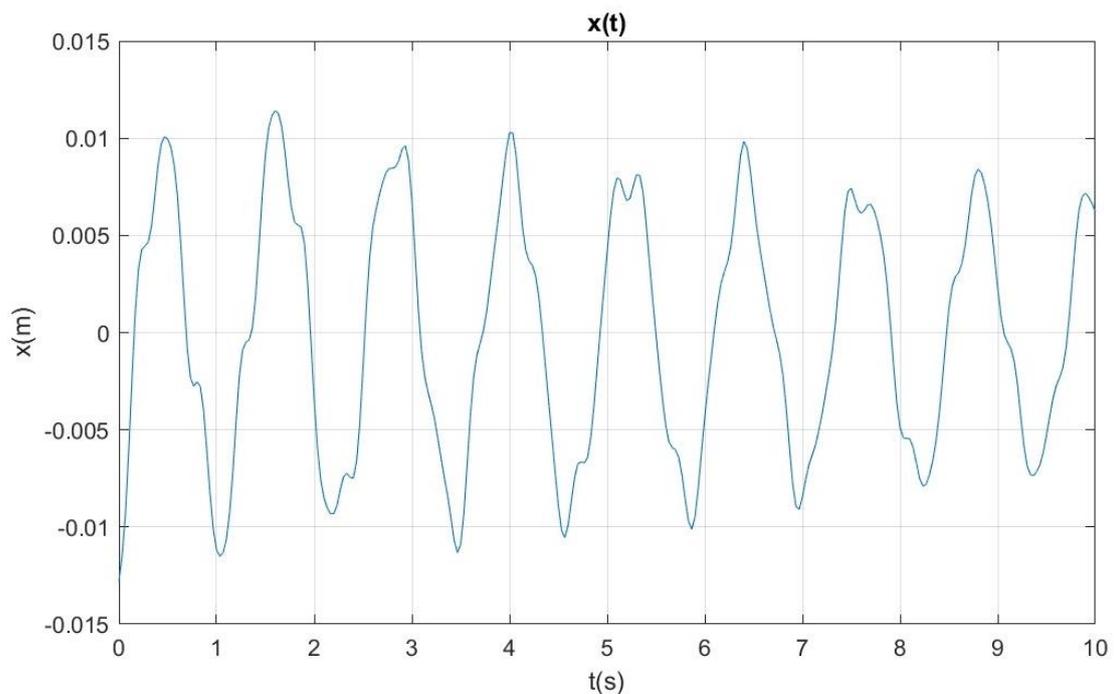


Figura 135: Gráfico acondicionado del desplazamiento en función del tiempo 4p0m. Visión

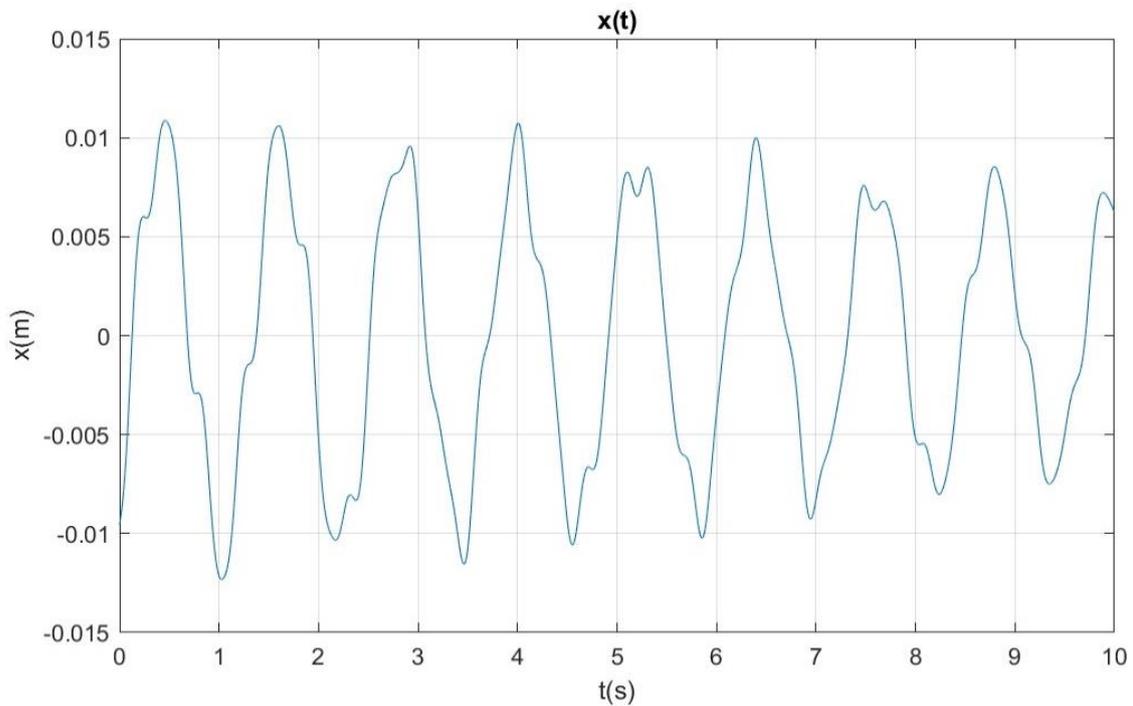


Figura 136: Gráfico acondicionado del desplazamiento en función del tiempo 4p0m. Láser

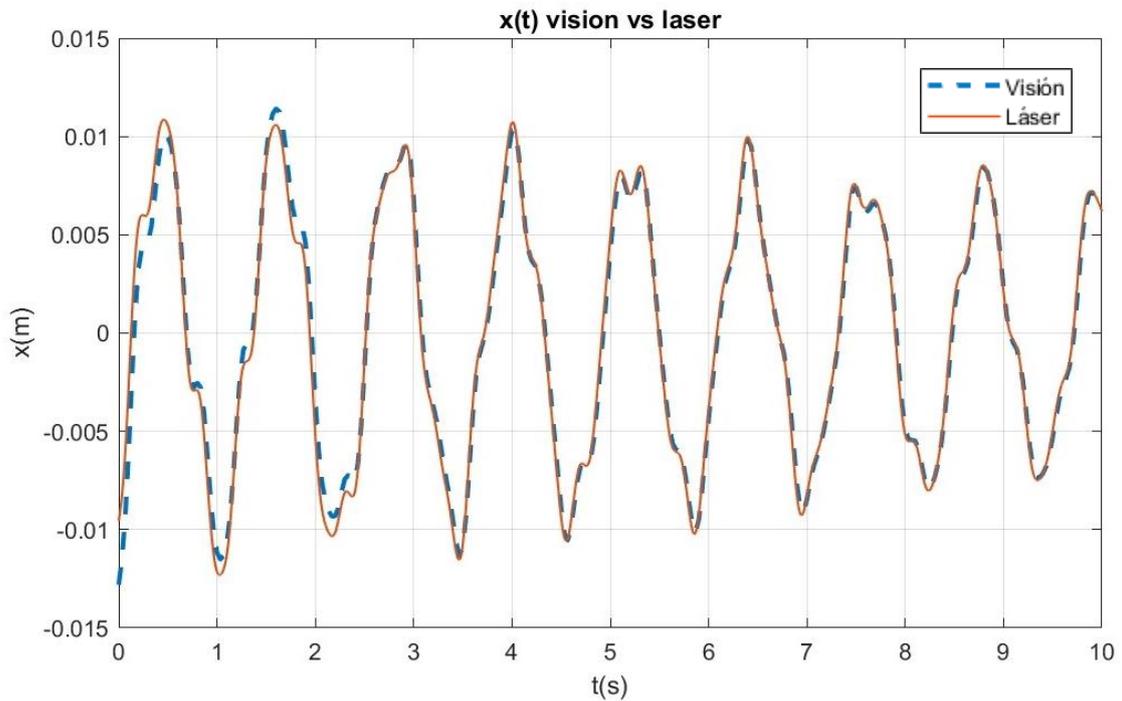


Figura 137: Superposición de gráficos de desplazamiento láser y visión 4p0m

Como en los experimentos de más de un grado de libertad anteriores, se ajustara primero a uno, luego a dos y por último a tres para comparar los resultados de los parámetros obtenidos. Se comienza con el ajuste a un grado de libertad.

En la Figura 138 y Figura 139 se muestran las gráficas del ajuste a un grado de libertad para visión y laser respectivamente, los parámetros modales se presentan en la Tabla 19 y Tabla 20.

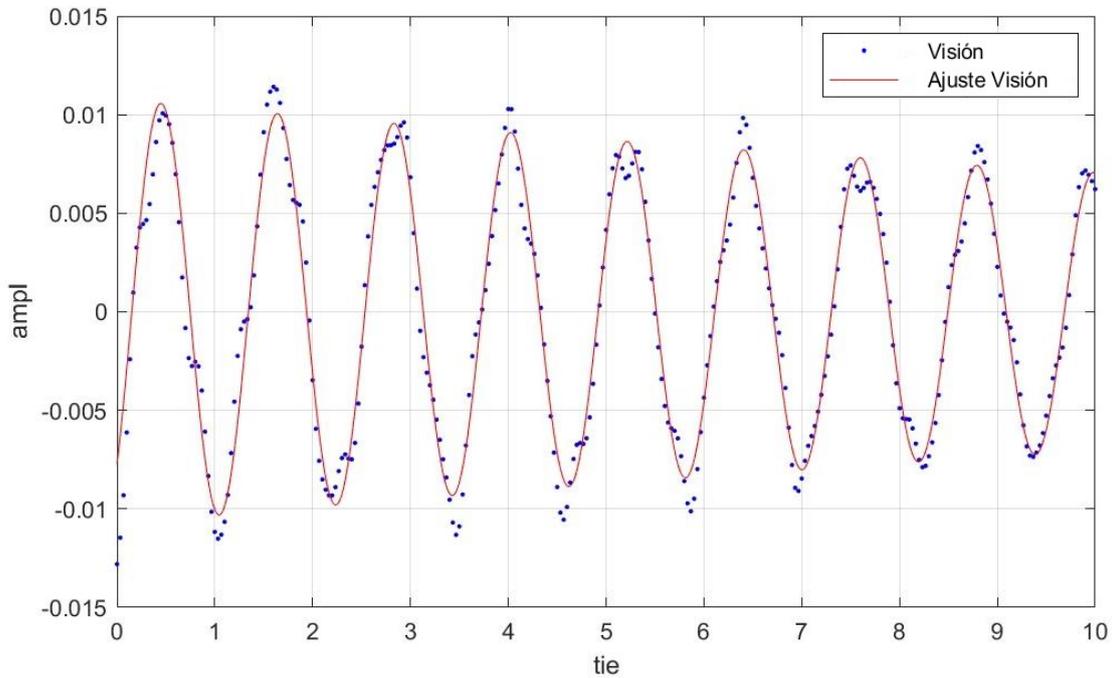


Figura 138: Ajuste 4p0m a 1 gdl. Visión

$\text{fitresultvision}(t) = A1 \cdot \exp(-Z1 \cdot 2 \cdot \pi \cdot w1 \cdot t) \cdot \sin(2 \cdot \pi \cdot w1 \cdot t + fi1)$
Coefficients (with 95% confidence bounds):
A = 0,01078 (0,01031, 0,01124)
Z = 0,008016 (0,006432, 0,009599)
fi = -0,7996 (-0,8412, -0,7579)
w = 0,8391 (0,8378, 0,8404)
rsquare: 0,9590

Tabla 19: Resultados 4p0m visión 1gdl

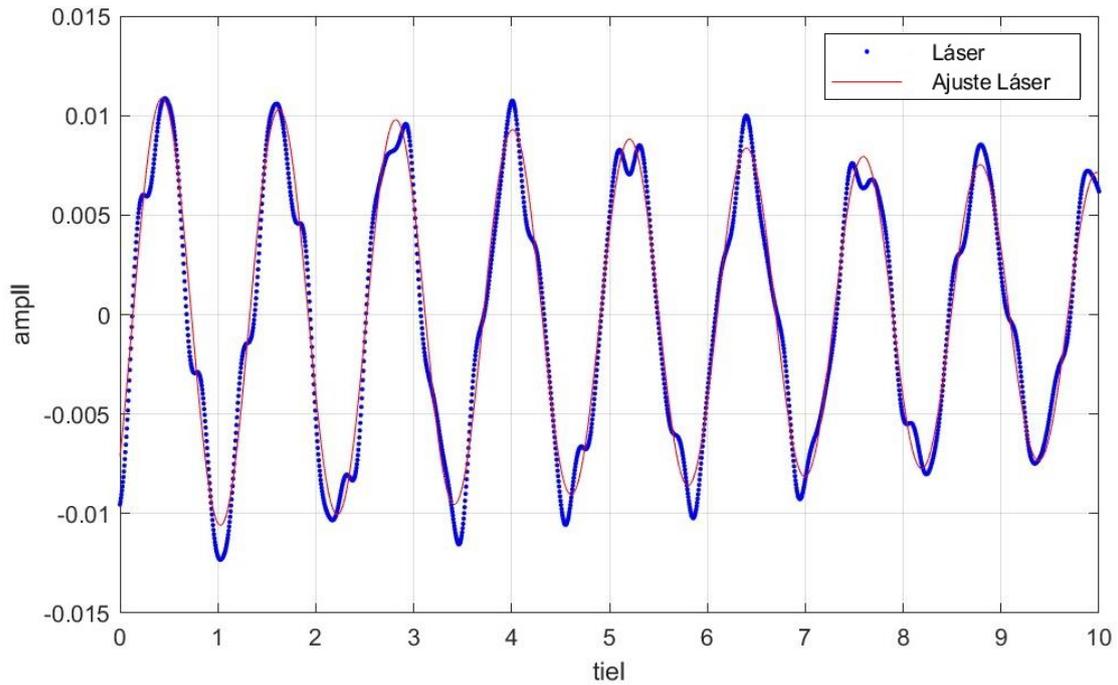


Figura 139: Ajuste 4p0m a 1 gdl. Láser

$\text{fitresultláser}(t) = A1 \cdot \exp(-Z1 \cdot 2 \cdot \pi \cdot w1 \cdot t) \cdot \sin(2 \cdot \pi \cdot w1 \cdot t + fi1)$
Coefficients (with 95% confidence bounds):
A = 0,01108 (0,01092, 0,01124)
Z = 0,008342 (0,0078, 0,008884)
fi = -0,6977 (-0,7118, -0,6836)
w = 0,8377 (0,8372, 0,8381)
rsquare: 0,9677

Tabla 20: Resultados 4p0m láser 1gdl

De la misma manera, se realiza el ajuste a dos grados de libertad, del que se obtendrán las gráficas mostradas en la Figura 140 y Figura 141 y los parámetros de la Tabla 21 y Tabla 22.

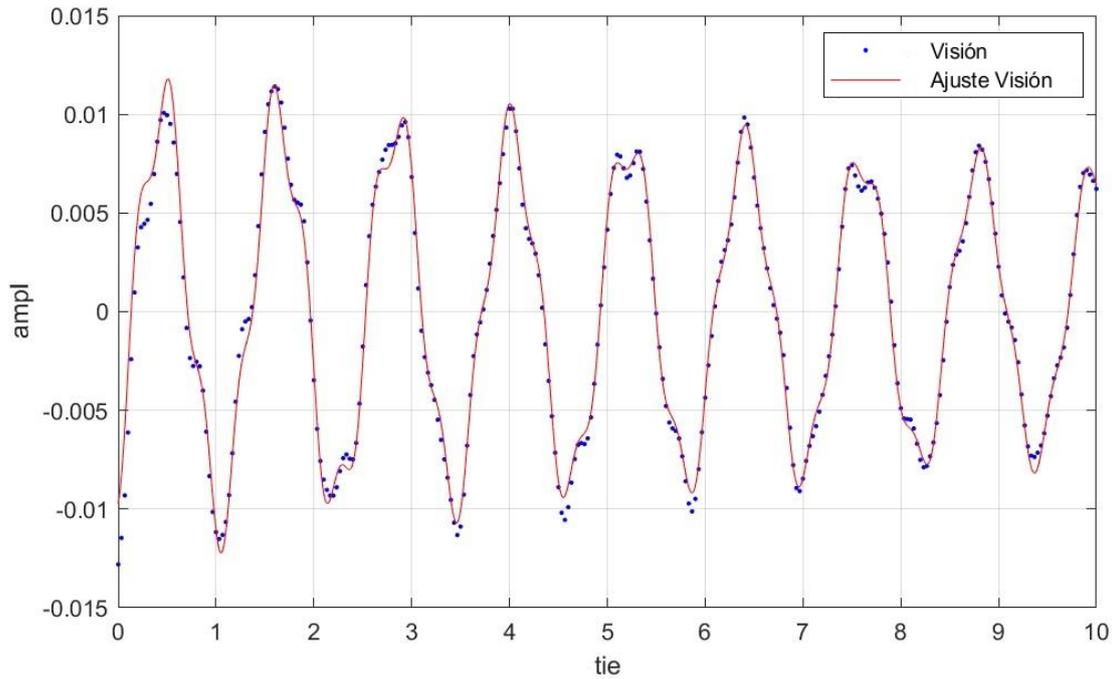


Figura 140: Ajuste 4p0m a 2 gdl. Visión

$\text{fitresultvision}(t) = A1 \cdot \exp(-Z1 \cdot 2 \cdot \pi \cdot w1 \cdot t) \cdot \sin(2 \cdot \pi \cdot w1 \cdot t + fi1) + A2 \cdot \exp(-Z2 \cdot 2 \cdot \pi \cdot w2 \cdot t) \cdot \sin(2 \cdot \pi \cdot w2 \cdot t + fi2)$
Coefficients (with 95% confidence bounds):
A1 = -0,01071 (-0,01097, -0,01044)
A2 = -0,002165 (-0,002447, -0,001884)
Z1 = 0,007857 (0,006961, 0,008754)
Z2 = 0,004648 (0,003065, 0,006232)
fi1 = 2,345 (2,321, 2,368)
fi2 = 1,255 (1,123, 1,387)
w1 = 0,8391 (0,8383, 0,8398)
w2 = 2,892 (2,887, 2,896)
rsquare: 0,9871

Tabla 21: Resultados 4p0m visión 2gdl

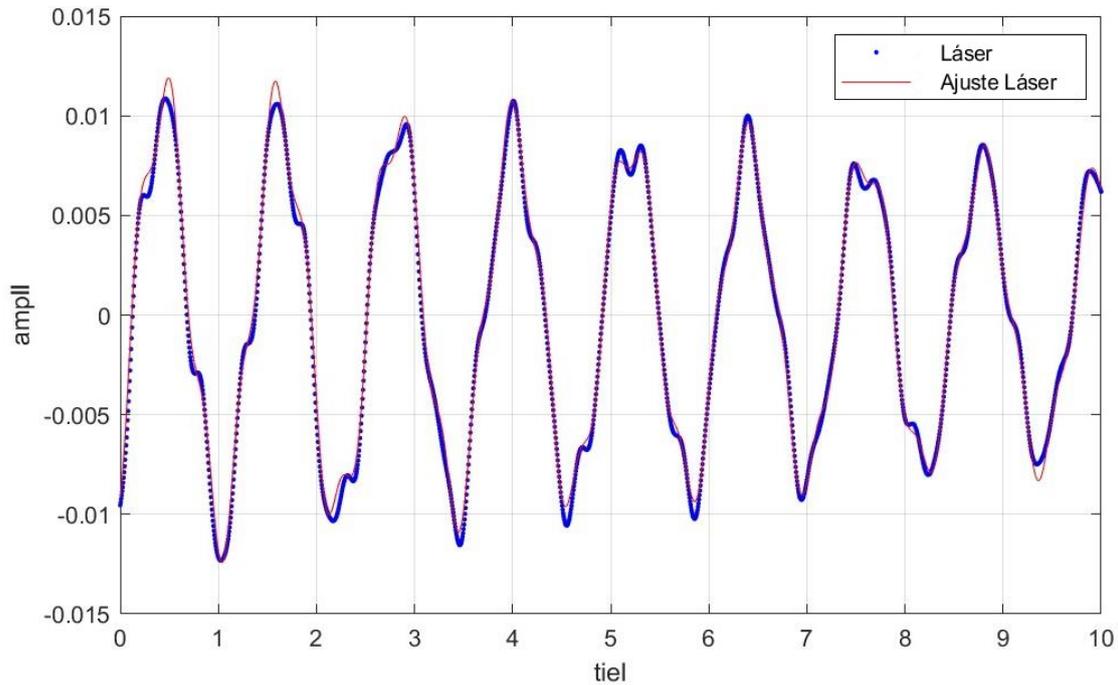


Figura 141: Ajuste 4p0m a 2 gdl. Láser

$\text{fitresultláser}(t) = A1 \cdot \exp(-Z1 \cdot 2 \cdot \pi \cdot w1 \cdot t) \cdot \sin(2 \cdot \pi \cdot w1 \cdot t + fi1) + A2 \cdot \exp(-Z2 \cdot 2 \cdot \pi \cdot w2 \cdot t) \cdot \sin(2 \cdot \pi \cdot w2 \cdot t + fi2)$
Coefficients (with 95% confidence bounds):
A1 = 0,01106 (0,01099, 0,01113)
A2 = -0,002018 (-0,002089, -0,001946)
Z1 = 0,008327 (0,008097, 0,008557)
Z2 = 0,003786 (0,003375, 0,004196)
fi1 = -0,6982 (-0,7042, -0,6922)
fi2 = 1,49 (1,454, 1,525)
w1 = 0,8377 (0,8375, 0,8379)
w2 = 2,890 (2,889, 2,891)
rsquare: 0,9942

Tabla 22: Resultados 4p0m láser 2gdl

Por último, se realiza el ajuste a tres grados de libertad, de forma que se podrán estimar las tres frecuencias y coeficientes de amortiguamiento que tiene la estructura. Los gráficos de este ajuste se representan en la Figura 142 y Figura 143 y los parámetros modales se muestran en la Tabla 23 y Tabla 24, de visión y del láser respectivamente, viendo que la función obtenida se ajusta perfectamente a la curva.

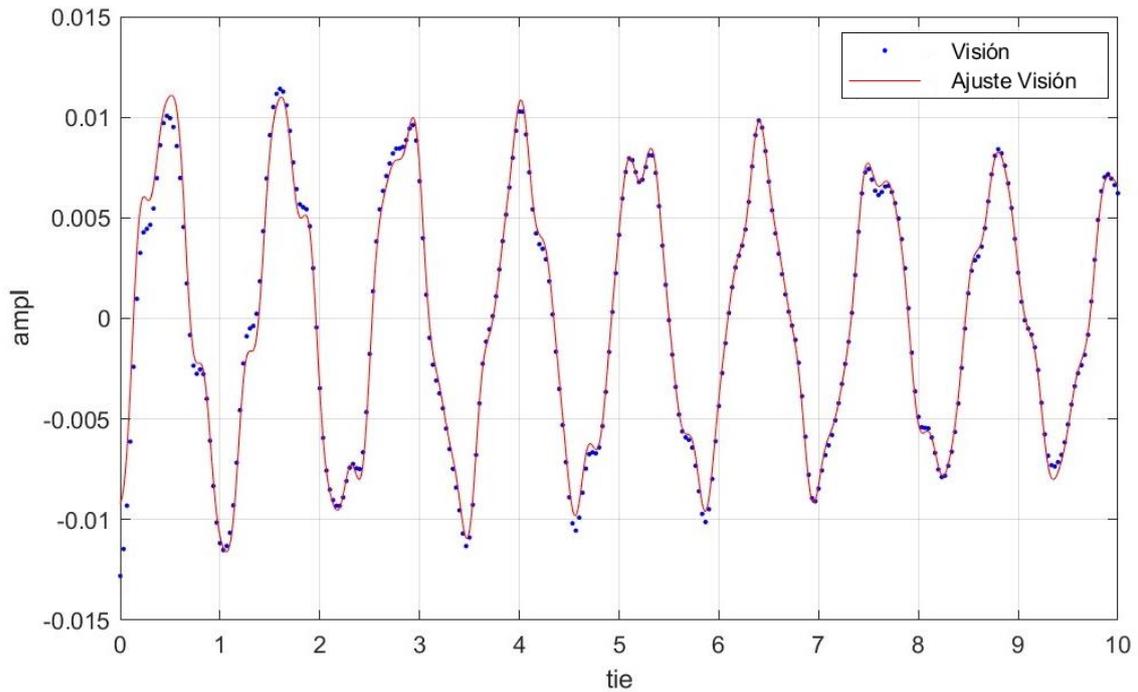


Figura 142: Ajuste 4p0m a 3 gdl. Visión

$\text{fitresultvision}(t) = A1 \cdot \exp(-Z1 \cdot 2 \cdot \pi \cdot w1 \cdot t) \cdot \sin(2 \cdot \pi \cdot w1 \cdot t + fi1) +$ $A2 \cdot \exp(-Z2 \cdot 2 \cdot \pi \cdot w2 \cdot t) \cdot \sin(2 \cdot \pi \cdot w2 \cdot t + fi2) +$ $A3 \cdot \exp(-Z3 \cdot 2 \cdot \pi \cdot w3 \cdot t) \cdot \sin(2 \cdot \pi \cdot w3 \cdot t + fi3)$
Coefficients (with 95% confidence bounds):
A1 = -0,0107 (-0,01094, -0,01047)
A2 = 0,002138 (0,00189, 0,002387)
A3 = -0,0007238 (-0,0009818, -0,0004659)
Z1 = 0,007841 (0,007047, 0,008636)
Z2 = 0,004529 (0,003124, 0,005935)
Z3 = 0,003256 (0,0004871, 0,006026)
fi1 = 2,345 (2,324, 2,366)
fi2 = -1,879 (-1,997, -1,761)
fi3 = 5,558 (5,205, 5,912)
w1 = 0,839 (0,8384, 0,8397)
w2 = 2,891 (2,887, 2,896)
w3 = 4,671 (4,658, 4,684)
rsquare: 0,9901

Tabla 23: Resultados 4p0m visión 3gdl

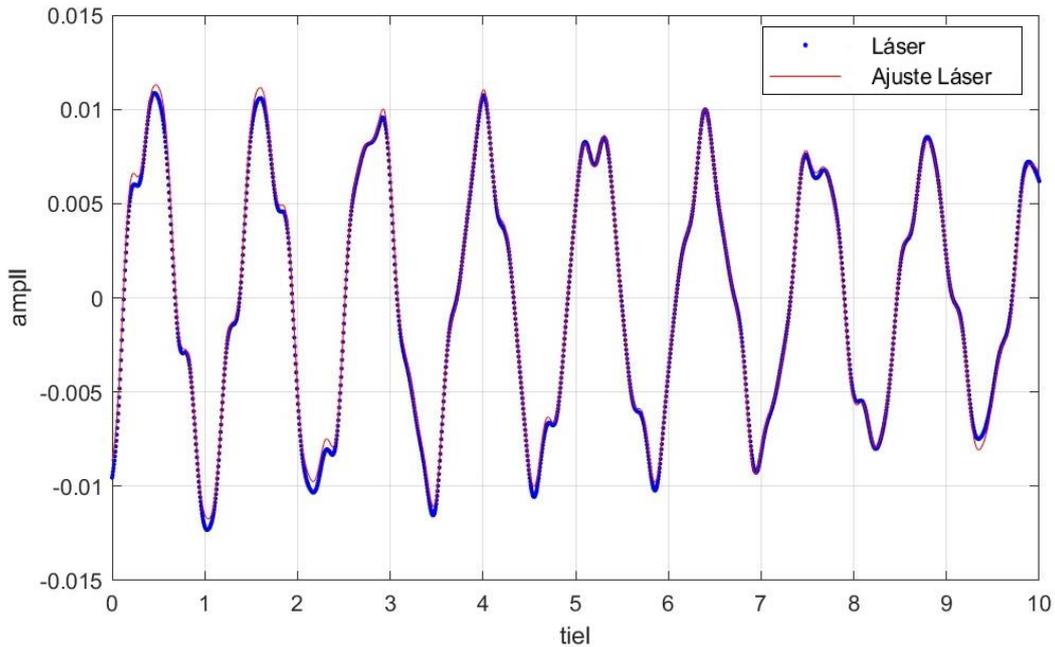


Figura 143: Ajuste 4p0m a 3 gdl. Láser

$\text{fitresultláser}(t) = A1 \cdot \exp(-Z1 \cdot 2 \cdot \pi \cdot w1 \cdot t) \cdot \sin(2 \cdot \pi \cdot w1 \cdot t + fi1) +$ $A2 \cdot \exp(-Z2 \cdot 2 \cdot \pi \cdot w2 \cdot t) \cdot \sin(2 \cdot \pi \cdot w2 \cdot t + fi2) +$ $A3 \cdot \exp(-Z3 \cdot 2 \cdot \pi \cdot w3 \cdot t) \cdot \sin(2 \cdot \pi \cdot w3 \cdot t + fi3)$
Coefficients (with 95% confidence bounds):
A1 = 0,01105 (0,011, 0,0111)
A2 = -0,001985 (-0,002035, -0,001935)
A3 = 0,0007093 (0,0006573, 0,0007613)
Z1 = 0,008292 (0,008131, 0,008454)
Z2 = 0,003635 (0,003346, 0,003924)
Z3 = 0,002818 (0,002272, 0,003364)
fi1 = 5,586 (5,582, 5,591)
fi2 = 7,781 (7,756, 7,806)
fi3 = 2,496 (2,424, 2,569)
w1 = 0,8376 (0,8375, 0,8378)
w2 = 2,89 (2,889, 2,89)
w3 = 4,676 (4,674, 4,679)
rsquare: 0,9972

Tabla 24: Resultados 4p0m láser 3gdl

Tras analizar los parámetros más representativos para la caracterización modal, en la Tabla 25 se observa que la primera y la segunda frecuencia estimadas por visión en cada ajuste son casi idénticas, mientras que los coeficientes de amortiguamiento difieren un poco más pero son bastante similares.

Visión			
Frecuencia (Hz)		Coeficiente de amortiguamiento (%)	
w1	0,8391	Z1	0,8016
	0,8391		0,7857
	0,8390		0,7841
w2	2,8920	Z2	0,4648
	2,8910		0,4529

Tabla 25: Comparación de las frecuencias y coeficientes de amortiguamiento estimados por visión para cada ajuste

Análogamente, en la Tabla 26 se muestran los parámetros estimados para el láser en cada ajuste, observando que las frecuencias son casi idénticas y los coeficientes de amortiguamiento difieren más, igual que en el análisis por visión.

Láser			
Frecuencia (Hz)		Coeficiente de amortiguamiento (%)	
w1	0,8377	Z1	0,8342
	0,8377		0,8327
	0,8376		0,8292
w2	2,8900	Z2	0,3786
	2,8900		0,3635

Tabla 26: Comparación de las frecuencias y coeficientes de amortiguamiento estimados del láser para cada ajuste

Comparando los parámetros estimados por visión con los del láser, se observa en la Tabla 27 que las frecuencias estimadas mediante ambos métodos con casi idénticas para este experimento, con un error muy pequeño, por el contrario en la Tabla 28 se observa que los coeficientes de amortiguamiento obtenidos difieren más, presentando un error más grande entre sus resultados.

4p0m			
Frecuencias (Hz)	Visión	Láser	Error relativo (%)
w1	0,8390	0,8376	0,1671
w2	2,8910	2,8900	0,0346
w3	4,6710	4,6760	0,1069

Tabla 27: Comparación frecuencias estimadas con láser y visión 4p0m

4p0m			
Coeficiente de amortiguamiento (%)	Visión	Láser	Error relativo (%)
Z1	0,7841	0,8292	5,4390
Z2	0,4529	0,3635	24,5942
Z3	0,3256	0,2818	15,5429

Tabla 28: Comparación coeficientes de amortiguamiento estimadas con láser y visión 4p0m

CAPÍTULO 7. CONCLUSIONES Y LÍNEAS FUTURAS

7.1. Conclusiones

Observando los resultados mostrados en el apartado anterior, se puede concluir que el sensor de desplazamiento óptico creado a partir de un dispositivo móvil y un código basado en visión artificial y procesamiento de imágenes, combinado con su posterior análisis, funciona correctamente y es una buena opción para estimar los parámetros modales, en este caso frecuencias y coeficientes de amortiguamiento, de diferentes estructuras esbeltas de uno dos y tres grados de libertad. A continuación se mostrará una comparativa de ambos parámetros obtenidos por los dos métodos.

En la Tabla 29 se puede comprobar que el error relativo en la frecuencia obtenida por ambos métodos es muy pequeño, no alcanzando el 1% en ninguno de los casos, por consiguiente, los resultados obtenidos para las frecuencias de vibración son óptimos.

Experimento	Frecuencia (Hz)	Visión	Láser	Error relativo (%)
1p0m	w1	2,5820	2,5820	0,0000
1p1m	w1	1,1320	1,1320	0,0000
1p2m	w1	0,7459	0,7453	0,0805
mc0	w1	2,3000	2,3000	0,0000
mcM	w1	2,1540	2,1540	0,0000
2p2m	w1	0,6708	0,6706	0,0298
	w2	4,4430	4,4420	0,0225
3p0m	w1	1,1350	1,1350	0,0000
	w2	4,9930	4,9920	0,0200
4p0m	w1	0,8390	0,8376	0,1671
	w2	2,8910	2,8900	0,0346
	w3	4,6710	4,6760	0,1069

Tabla 29: Comparación de error relativo entre las frecuencias obtenidas mediante visión y láser

En la Tabla 30 se muestra el error relativo de los coeficientes de amortiguamiento obtenidos por los dos métodos en los diferentes casos estudiados. En este caso, los resultados son menos precisos, encontrando errores de hasta 24%, aunque por lo general no superan un 10%, ya que el amortiguamiento es más susceptible a variaciones dependiendo del intervalo de la función analizada o de la duración de la misma.

Estos errores, más notables entre los coeficientes de amortiguamiento pueden ser debidos a la imprecisión en el ajuste, al posible error humano a la hora de registrar los vídeos, el error en las mediciones para escalar la función de visión, o las imprecisiones en la detección del punto debido a la diferencia de las frecuencias de muestreo entre los dos métodos, 1/30 la del vídeo (30 fps) y 1/200 la del láser (200 datos/segundo)

Experimento	Coefficiente de amortiguamiento (%)	Visión	Láser	Error relativo (%)
1p0m	Z1	0,4802	0,4938	2,7542
1p1m	Z1	0,2818	0,2829	0,3888
1p2m	Z1	0,2671	0,3092	13,6158
mc0	Z1	0,425	0,4274	0,5615
mcM	Z1	0,6155	0,607	1,4003
2p2m	Z1	0,6052	0,5727	5,6749
	Z2	0,429	0,4323	0,7634
3p0m	Z1	0,617	0,6674	7,5517
	Z2	0,4052	0,4083	0,7592
4p0m	Z1	0,7841	0,8292	5,4390
	Z2	0,4529	0,3635	24,5942
	Z3	0,3256	0,2818	15,5429

Tabla 30: Comparación de error relativo entre los coeficientes de amortiguamiento obtenidas mediante visión y láser

Mediante este experimento se puede confirmar que el sensor de desplazamiento óptico creado es una buena opción a la hora de analizar los parámetros modales de una estructura esbelta, ya que es un método barato, sencillo de implementar, siendo necesario únicamente un teléfono móvil que permita grabar vídeos y un ordenador con un el software MATLAB y el código desarrollado. Poseyendo estas tres cosas, bastante comunes en este ámbito, el coste para implementarlo sería cero, comparado con el sensor láser, que puede alcanzar los 2000 € o el acelerómetro que se encuentra alrededor de los 500 €.

El código implementado es práctico y versátil, ya que se puede analizar cualquier video de estas características únicamente sustituyendo el nombre del archivo en el programa.

Además ofrece la ventaja de posibilitar el análisis de desplazamientos en las dos direcciones del plano, x e y, simultáneamente, al contrario que el sensor láser o el

acelerómetro que solo son capaces de medir en una dirección, por lo que en este caso el precio ascendería al doble, 4000 € para el láser y 1000 € para el acelerómetro.

Como posible mejora, cabe mencionar que cuanto mayor sea el *frame rate* (fotogramas por segundo), más precisos serán los resultados obtenidos. En este caso los videos han sido grabados a 30 fps, si fueran grabados a 60 fps (muchos móviles hoy en día son capaces de grabar a esta velocidad, algunos hasta 120), se obtendrían el doble de datos en cada segundo del vídeo, y por lo tanto un mejor resultado. Aun así, no alcanzarían los 200 datos/segundo del láser.

7.2. Líneas futuras

Otra ventaja de este sensor óptico es su capacidad para detectar dos o más puntos simultáneamente, como en el ejemplo mostrado en la Figura 144, en el que se desea analizar al mismo tiempo el desplazamiento de la estructura y el movimiento del péndulo. Esta es una línea futura que puede tener una aplicación interesante a la hora de estudiar este tipo de movimientos.

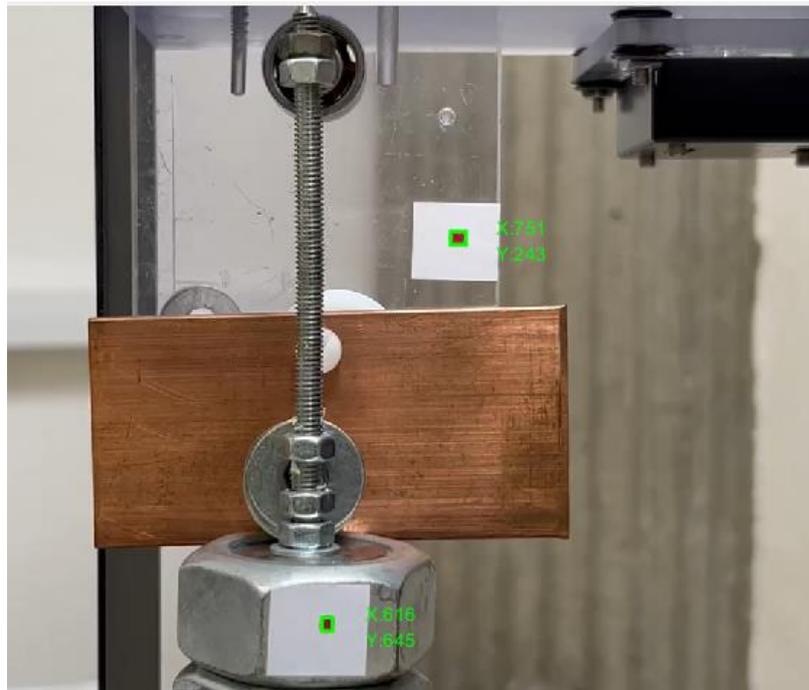


Figura 144: Aplicación a seguimiento de dos puntos simultáneamente

Se puede considerar como aplicación futura de este trabajo cualquier experimento en el que se quiera estudiar propiedades dinámicas de estructuras esbeltas que implique la medición de desplazamientos de un punto de la misma. Se puede sustituir el sistema clásico de adquisición por este sistema al alcance de cualquier estudiante o profesor de ingeniería.

7.3. Consideraciones adicionales

En este apartado se detallará como ha sido la planificación a la hora de realizar este trabajo, dividiéndolo en las diferentes etapas, una estimación de las horas invertidas en cada una de estas, las competencias transversales adquiridas y aspectos económicos.

A continuación, se expondrán las diferentes etapas del proceso, con una duración desde julio de 2023 hasta mediados de octubre de 2023, detallando las horas de trabajo implicadas en cada una de las etapas:

- Propuesta del tema e investigación sobre el mismo: **20 horas**
- Desarrollo de una primera versión del código (a partir de un video de prueba): **60 horas**
- Aplicación de esta versión a un ejemplo real para comprobar su funcionamiento: **10 horas**
- Optimización del código para la consecución de los objetivos deseados: **40 horas**
- Realización de los experimentos en el laboratorio (obtención de datos): **5 horas**
- Procesamiento de los videos de cada experimento: **30 horas**
- Obtención de funciones de ajuste y análisis de resultados: **30 horas**
- Redacción del trabajo y presentación: **110 horas**

La realización del presente trabajo ha supuesto un total de 305 horas, distribuidas en las etapas mencionadas, en las que se han desarrollado conocimientos sobre dinámica de estructuras y vibraciones estructurales, ampliando los conocimientos adquiridos durante el grado. Además se ha aprendido a programar en MATLAB, enfocado en visión artificial y análisis de funciones, pero cuyo aprendizaje puede resultar útil para cualquier aplicación futura. Y principalmente se ha conseguido el objetivo del trabajo, crear un sistema de visión para la estimación de propiedades dinámicas alternativo a los sistemas tradicionales basados en sensores, que puede tener diferentes aplicaciones en un futuro y suponer una disminución de costes.

Por último, como ya se ha comentado, el coste de la realización de este proyecto es nulo, ya que las maquetas de estudio se encuentran en el laboratorio de estructuras y el software se puede obtener de manera gratuita en versión de estudiante por pertenecer a la universidad, además, no es necesaria la compra de un ordenador para implementar el software ni de un dispositivo móvil para adquirir los videos ya que ya se cuenta con ellos. Por lo tanto, es un sistema que puede ser utilizado por cualquier persona en posesión de un dispositivo de adquisición de imágenes y un ordenador con el software MATLAB.

BIBLIOGRAFÍA

- [1] Acosta Jiménez R.J., R. M. (2017). *Sistema De Identificación y seguimiento de objetos de colores RGB para una plataforma rotatoria mediante visión artificial empleando el software de Matlab*. Universidad de Magdalena, Ingeniería Electrónica
- [2] Santa María Pinedo, J. C. (2021). *Reconocimiento de patrones de imágenes a través de un sistema de visión artificial en MATLAB*. Recuperado el Septiembre de 2023, de Revista Científica De Sistemas E Informática: <https://doi.org/10.51252/rcsi.v1i2.131>
- [3] Borrella Petisco, B (2022). *Introducción a la visión artificial. Procesos y aplicaciones*. Universidad Complutense de Madrid, Ingeniería Matemática.
- [4] *Tracker video analysis and modeling tool for physics education*. <https://physlets.org/tracker/>
- [5] Chopra, A. K. (2014). *Dinámica de estructuras* (4 ed.). Pearson.
- [6] Keyence. (s.f.). *Qué es un sensor láser de tipo de reconocimiento de "luz recibida"*. Recuperado el 10 Septiembre 2023, de Fundamentos del sensor: https://www.keyence.com.mx/ss/products/sensor/sensorbasics/láser_light/info/
- [7] MathWorks. (s.f.). Matlab. Recuperado el 17 de Julio de 2023, de Matemáticas. Gráficas. Programación.: <https://es.mathworks.com/products/matlab.html>
- [8] Mathworks. (s.f.). *Image Processing Toolbox*. Recuperado el 17 de Julio de 2023, de Centro de ayuda Matlab: https://es.mathworks.com/help/images/index.html?searchHighlight=image%20processing&s_tid=srchtitle_support_results_1_image%20processing
- [9] MathWorks. (s.f.). *Computer Vision Toolbox*. Recuperado el 17 de Julio de 2023, de Centro de ayuda Matlab: https://es.mathworks.com/help/vision/index.html?s_tid=srchtitle_site_search_1_computer%20vision
- [10] MathWorks. (s.f.). *Curve Fitting Toolbox*. Recuperado el 17 de Julio de 2023, de Centro de ayuda Matlab: https://es.mathworks.com/help/curvefit/index.html?searchHighlight=curve%20fitting%20&s_tid=srchtitle_support_results_1_curve%20fitting%20
- [11] MathWorks. (s.f.). *clear*. Recuperado el 17 de Julio de 2023, de Centro de ayuda MatLab: https://es.mathworks.com/help/matlab/ref/clc.html?searchHighlight=clc&s_tid=srchtitle_support_results_1_clc



- [12] MathWorks. (s.f.). *clc*. Recuperado el 17 de Julio de 2023, de Centro de ayuda MatLab: https://es.mathworks.com/help/matlab/ref/clc.html?searchHighlight=clc&tid=srchtitle_support_results_1_clc
- [13] MathWorks. (s.f.). *close*. Recuperado el 17 de Julio de 2023, de Centro de ayuda MatLab: https://es.mathworks.com/help/matlab/ref/close.html?s_tid=doc_ta#d126e199811
- [14] MathWorks. (s.f.). *cd*. Recuperado el 17 de Julio de 2023, de Centro de ayuda MaLab: https://es.mathworks.com/help/matlab/ref/cd.html?searchHighlight=cd&tid=srchtitle_support_results_1_cd
- [15] MathWorks. (s.f.). *VideoReader*. Recuperado el 18 de Julio de 2023, de Centro de ayuda MatLab: https://es.mathworks.com/help/matlab/ref/videoreader.html?searchHighlight=videoreader&tid=srchtitle_support_results_1_videoreader
- [16] MathWorks. (s.f.). *zeros*. Recuperado el 18 de Julio de 2023, de Centro de ayuda Matlab: https://es.mathworks.com/help/matlab/ref/zeros.html?searchHighlight=zeros&tid=srchtitle_support_results_1_zeros
- [17] MathWorks. (s.f.). *hasFrame*. Recuperado el 18 de Julio de 2023, de Centro de ayuda Matlab: https://es.mathworks.com/help/matlab/ref/videoreader.hasframe.html?searchHighlight=hasframe&tid=srchtitle_support_results_1_hasframe
- [18] MathWorks. (s.f.). *readFrame*. Recuperado el 18 de Julio de 2023, de Centro de ayuda Matlab: https://es.mathworks.com/help/matlab/ref/videoreader.readframe.html?searchHighlight=readframe&tid=srchtitle_support_results_1_readframe
- [19] MathWorks. (s.f.). *imshow*. Recuperado el Agosto de 2023, de Centro de ayuda Matlab: https://es.mathworks.com/help/matlab/ref/imshow.html?s_tid=doc_ta
- [20] MathWorks. (s.f.). *rgb2gray*. Recuperado el 18 de Julio de 2023, de Centro de ayuda Matlab: https://es.mathworks.com/help/matlab/ref/rgb2gray.html?s_tid=doc_ta
- [21] MathWorks. (s.f.). *imsubtract*. Recuperado el 18 de Julio de 2023, de Centro de ayuda Matlab: https://es.mathworks.com/help/images/ref/imsubtract.html?s_tid=doc_ta

- [22] Mathworks. (s.f.). *imhist*. Recuperado el 18 de Julio de 2023, de Centro de ayuda Matlab: <https://es.mathworks.com/help/images/ref/imhist.html?lang=en>
- [23] MathWorks. (s.f.). *imbinarize*. Recuperado el 18 de Julio de 2023, de Centro de ayuda Matlab: https://es.mathworks.com/help/images/ref/imbinarize.html?searchHighlight=i%20binarize&s_tid=srchtitle_support_results_1_imbinarize
- [24] Mathworks. (s.f.). *strel*. Recuperado el 18 de Julio de 2023, de Centro de ayuda Matlab: https://es.mathworks.com/help/images/ref/strel.html?searchHighlight=strel&s_tid=srchtitle_support_results_1_strel
- [25] Mathworks. (s.f.). *imopen*. Recuperado el 18 de Julio de 2023, de Centro de ayuda Matlab: https://es.mathworks.com/help/images/ref/imopen.html?searchHighlight=imopen&s_tid=srchtitle_support_results_1_imopen
- [26] Mathworks. (s.f.). *medfilt2*. Recuperado el 18 de Julio de 2023, de Centro de ayuda Matlab: https://es.mathworks.com/help/images/ref/medfilt2.html?searchHighlight=medfilt2&s_tid=srchtitle_support_results_1_medfilt2
- [27] MathWorks. (s.f.). *imfill*. Recuperado el 18 de Julio de 2023, de Centro de ayuda Matlab: https://es.mathworks.com/help/images/ref/imfill.html?searchHighlight=imfill&s_tid=srchtitle_support_results_1_imfill
- [28] MathWorks. (s.f.). *bwlabel*. Recuperado el 18 de Julio de 2023, de Centro de ayuda Matlab: https://es.mathworks.com/help/images/ref/bwlabel.html?searchHighlight=bwlabel&s_tid=srchtitle_support_results_1_bwlabel
- [29] Mathworks. (s.f.). *regionprops*. Recuperado el 18 de Julio de 2023, de Centro de ayuda Matlab: https://es.mathworks.com/help/images/ref/regionprops.html?s_tid=doc_ta
- [30] Mathworks. (s.f.). *Image Viwer*. Recuperado el 3 de Agosto de 2023, de Centro de ayuda Matlab: https://es.mathworks.com/help/images/ref/imageviewer-app.html?searchHighlight=image%20viewer&s_tid=srchtitle_support_results_1_image%20viewer
- [31] Mathworks. (s.f.). *readmatrix*. Recuperado el 3 de Agosto de 2023, de Centro de ayuda Matlab:



https://es.mathworks.com/help/matlab/ref/readmatrix.html?searchHighlight=readmatrix&s_tid=srchtitle_support_results_1_readmatrix

- [32] MathWorks. (s.f.). *detrend*. Recuperado el 25 de Agosto de 2023, de Centro de ayuda Matlab: [https://es.mathworks.com/help/matlab/ref/detrend.html?s_tid=doc ta](https://es.mathworks.com/help/matlab/ref/detrend.html?s_tid=doc_ta)
- [33] MathWorks. (s.f.). *saveas*. Recuperado el 29 de Agosto de 2023, de Centro de ayuda Matlab: [https://es.mathworks.com/help/matlab/ref/saveas.html?searchHighlight=sav eas&s_tid=srchtitle_support_results_1_saveas](https://es.mathworks.com/help/matlab/ref/saveas.html?searchHighlight=saveas&s_tid=srchtitle_support_results_1_saveas)
- [34] Mathworks. (s.f.). *Curve Fitter*. Recuperado el 2 de Septiembre de 2023, de Centro de ayuda Matlab: [https://es.mathworks.com/help/curvefit/curvefitter-app.html?searchHighlight=curve%20fitter&s_tid=srchtitle_support_results_1_c urve%20fitter](https://es.mathworks.com/help/curvefit/curvefitter-app.html?searchHighlight=curve%20fitter&s_tid=srchtitle_support_results_1_curve%20fitter)

ANEXO I: CÓDIGO EN MATLAB

SEGUIMIENTO DE UN PUNTO MEDIANTE VISIÓN ARTIFICIAL

```
clc
clear
close all
cd 'C:\Users\Carlos\Desktop\TFG\TODO'
```

IMPORTAR ARCHIVOS Y CREAR VECTOR DE TIEMPO Y COORDENADAS

```
% Importar archivos de video registrados
file1 = '1p0m.mp4';

% Importar archivo excel con los datos de los sensores recogidos en el
laboratorio
file2 = '1p0m.xlsx';

% Crear archivo videoreader y obtener sus características.
vid = VideoReader(file1);
fps = vid.FrameRate;
num_frames = vid.NumFrames;
duracion = vid.Duration;

% Crear una matriz para almacenar las coordenadas x e y del punto
coords = zeros(num_frames, 2);

% Crear un vector de tiempo en frames (tf) desde 1 hasta el numero de
frames del video y un vector de tiempos (t) en segundos, dividiendolo entre
el número de frames por segundo.
tf = 0: num_frames-1;
t = (tf/30)';
```

DETECCIÓN DE PUNTO ROJO Y OBTENCIÓN DE COORDENADAS

```
i=1;
while hasFrame(vid)

% Leer el frame correspondiente del video
frame = readFrame(vid);

% Convertir cada imagen correspondiente a cada fotograma del video a
escala de grises
grayframe = rgb2gray(frame);

% Extraer la capa de color rojo de cada fotograma del video
redframe = frame(:,:,1);

% Restar la capa en blanco y negro de la de color rojo
```

```
img = imsubtract(redframe, grayframe);

% En el command window: imhist(img) para ver el umbral donde se encuentra
nuestro objeto para separarlo del resto de la imagen. En este caso 0.225

% Otra forma de binarizar la imagen (mejor)
bw = imbinarize(img, 0.225);

% Creamos un elemento estructurante en forma de disco de radio 1
SE = strel('disk',1);

% Realizamos una apertura morfológica de la imagen aplicándole el elemento
morfológico
bw = imopen(bw,SE);

% Aplicar filtro para evitar ruidos en la imagen
bw = medfilt2(bw);

% Rellenar los huecos que contenga la imagen
bw = imfill(bw, "holes");

% Etiquetado de los objetos
[L,N]= bwlabel(bw);

% Función regionprops: medir propiedades de las regiones de imágenes
(área,centroide...)
prop = regionprops(L);

% Mostrar la imagen
%imshow(frame);

% Obtener el centroide del punto y almacenarlo en la matriz coords en cada
iteración del bucle
%c=prop.Centroid;
%coords(i,:) = c;

for n=1:N
    c=prop(n).Centroid; % Obtener el centroide

rectangle('Position',prop(n).BoundingBox,'EdgeColor','g','LineWidth',2); %
Encuadrar
    text(c(1),c(2),strcat('X:',num2str(c(1)),'\newline', ...
        'Y:',num2str(c(2))), 'Color','green'); % Mostrar coordenadas
    % Rellenar cada columna i de la matriz coords con cada vector de
posición del centroide
    coords(i,:) = c;
end
pause(0.0001);
```

```
i=i+1;  
end
```

ACONDICIONAMIENTO DE DATOS Y ANÁLISIS DE FUNCIONES

ESCALAR Y AJUSTAR LOS DATOS DE VISIÓN Y CREAR GRAFICO COMPLETO

```
% Escalar los datos obtenidos por visión de pixeles a metros mediante la  
relación m/pix  
  
% Importar medida de la pegatina en pixeles del archivo excel  
pegatinapix = readmatrix(file2, 'sheet', 'Data1', 'range', 'E2');  
pegatinapix = pegatinapix(1,1);  
  
% Medida real de la pegatina en metros  
pegatinam = 0.0485;  
  
% Relación m/pix  
relacion= pegatinam/pegatinapix;  
  
% Separar las coordenadas x e y del centroide, escalarlas y eliminar la  
tendencia lineal  
x = coords(:,1);  
x = detrend(x,1);  
x = x*relacion;  
x = -x;  
  
y = coords(:,2);  
y = detrend(y,1);  
y = y*relacion;  
  
% Crear gráfico del desplazamiento horizontal en funcion del tiempo  
figure(2)  
plot(t,x)  
%plot(t,y) % Para registros verticales  
xlabel('t(s)');  
ylabel('x(m)');  
%ylabel('y(m)'); % Para registros verticales  
title('x(t)');  
%title('y(t)'); % Para registros verticales  
grid on;  
saveas(figure(2), '1p0mvision.jpg'); %Guardar figura como imagen
```

IMPORTAR Y AJUSTAR DATOS LASER Y CREAR GRÁFICO COMPLETO

```
% Importar la matriz de datos obtenidos por el laser, almacenados en excel  
laser = readmatrix(file2, 'Sheet', 'Data1', 'ExpectedNumVariables', 2);
```

```
% Vector de tiempos del laser
tl = laser(:,1);

% Vector de desplazamientos del laser
desplaz = laser(:,2);

% Eliminar tendencia lineal de la función
desplaz = detrend(desplaz,1);

% Crear gráfico del desplazamiento en función del tiempo de los datos
obtenidos con el laser
figure(3)
plot(tl,desplaz)
xlabel('t(s)');
ylabel('x(m)');
%ylabel('y(m)'); % Para registros verticales
title('x(t)');
%title('y(t)'); % Para registros verticales
grid on;
saveas(figure(3), '1p0mlaser.jpg');
```

ACONDICIONAMIENTO DE LOS DATOS VISIÓN Y GRÁFICO DELIMITADO

```
% Se selecciona la parte del registro que se quiere ajustar, y se definen
los límites temporales
tini = 2; % s
tfin = 10; % s

% Se define la frecuencia de muestreo
Dt = t(2)-t(1);

% Se acotan los vectores de tiempo y amplitud
tmax = tfin-tini;
tie = [0:Dt:tmax]';

ampl = x(t >= tini & t <= tfin);
%ampl = y(t >= tini & t <= tfin); %Para registros verticales

% Eliminar tendencia lineal de nuevo
ampl = detrend(ampl,1);

% Se crea el gráfico del registro seleccionado
figure(4)
plot(tie,ampl)
xlabel('t(s)');
ylabel('x(m)');
%ylabel('x(m)'); % Para registros verticales
title('x(t)');
%title('y(t)'); % Para registros verticales
```

```
grid on;  
saveas(figure(4), '1p0mvisionAcondicionado.jpg');
```

ACONDICIONAMIENTO DE LOS DATOS LÁSER Y GRÁFICO DELIMITADO

```
%(Del mismo modo que para visión, diferenciando las variables con una L)  
  
% Se selecciona la parte del registro que se quiere ajustar, y se definen  
% los límites temporales  
tinil = 5; % s  
tfinl = 13; % s  
  
% Se define la frecuencia de muestreo  
Dtl = t1(2)-t1(1);  
  
% Se acotan los vectores de tiempo y amplitud  
tmaxl = tfinl-tinil;  
tiel = [0:Dtl:tmaxl]';  
  
ampll = desplaz(tl >= tinil & tl <= tfinl);  
  
% Eliminar la tendencia lineal de nuevo  
ampll = detrend(ampll,1);  
  
% Se crea el gráfico del registro seleccionado  
figure(5)  
plot(tiel,ampll)  
xlabel('t(s)');  
ylabel('x(m)');  
%ylabel('x(m)'); % Para registros verticales  
title('x(t)');  
%title('y(t)'); % Para registros verticales  
grid on;  
saveas(figure(5), 'carlos1laserAcondicionado.jpg');
```

SUPERPONER GRÁFICOS DE VISIÓN Y LÁSER

```
figure(6)  
p = plot(tie,ampl,tiel,ampll);  
p(1).LineWidth=2;  
p(2).LineWidth=1;  
p(1).LineStyle='--';  
xlabel('t(s)');  
ylabel('x(m)');  
%ylabel('y(m)'); % Para registros verticales  
title('x(t) vision vs laser');  
%title('y(t)'); % Para registros verticales  
grid on;
```

```
saveas(figure(6), '1p0mvisionVSlaser.jpg');
```

LLAMAR FUNCIÓN DE AJUSTE VISIÓN

```
[fitresultvision, gofvision] = 1p0mvision(tie , ampl)  
saveas(figure(7), '1p0mvisionAjuste.jpg');
```

LLAMAR FUNCIÓN DE AJUSTE LÁSER

```
[fitresultlaser, goflaser] = 1p0mlaser(tiel , ampl1)  
saveas(figure(8), '1p0mlaserAjuste.jpg');
```