

Escuela Superior Politécnica del Litoral

Facultad de Ingeniería en Mecánica y Ciencias de la Producción

Diseño e implementación de un procedimiento de optimización dinámica de robots
redundantes

INGE-2618

Proyecto Integrador

Previo la obtención del Título de:

Ingeniero Mecatrónico

Presentado por:

Andrea Katherine Espinoza Iñiguez

Fabricio Ricardo Quimis Marcillo

Guayaquil - Ecuador

Año: 2024

Dedicatoria

El presente proyecto lo dedico a mis padres, Manuel Espinoza y Lety Iñiguez, por tener fe en mí y gracias a su sacrificio y esfuerzo me han ayudado a estar donde estoy, gracias por estar siempre para mí.

A mis hermanos María Espinoza y Marco Espinoza, por ser un apoyo.

A mis abuelitos Marco Iñiguez y Martha Guzmán, por estar siempre pendientes de que no me haga falta nada y estar al tanto de mi bienestar.

Mis abuelitos Leticia Espinoza y Gonzalo Espinoza por tenerme siempre en sus oraciones.

A mis mascotas Yicsi y Lilith, por siempre ser mis compañeras fieles de estudio y ser un apoyo incondicional.

Andrea Katherine Espinoza Iñiguez

Dedicatoria

Dedico el presente proyecto a mis padres, Nereicy Marcillo y Freddy Quimis por su apoyo inagotable y por ser mi inspiración para alcanzar mis metas.

A mis hermanos, Lady Quimis, Freddy Quimis y Luis Iñiguez por ser mi compañía, mi fortaleza y mi ejemplo constante de superación. Gracias por estar siempre a mi lado y ser mi apoyo en los momentos difíciles.

A mis abuelos, Miguel Marcillo y Yolanda Merino, por estar pendientes de todo lo que me hacía falta. Dedico este trabajo especialmente a mi abuelita Yolanda, quien, aunque hace 9 años que no está conmigo, siempre fue un ejemplo de superación y sabiduría, y sé que, desde donde esté, me cuida.

Fabricio Ricardo Quimis Marcillo

Agradecimientos

Mi más sincero agradecimiento a mis padres, sin su fe en mi incluso cuando yo no.

Al Ph.D. Carlos Saldarriaga y el Ph.D Marcelo Fajardo, gracias por su guía en esta recta final.

Mi compañero de tesis Fabricio Quimis, por tu compromiso y el gran trabajo en equipo.

A las amistades que me dio la universidad, Carlos Pacheco por ser la primera amistad y una de las mejores, gracias por siempre estar.

A mis facuamigos los mejores amigos que me pudo dar la vida, que sin saberlo me dieron el valor para salir adelante.

Gracias a Zaida, Abel, y Camila quienes siempre llevare en mi corazón.

A Xavier Magallanes, quien ha sido un pilar fundamental y una de las mejores personas que he conocido, gracias por siempre confiar en mis capacidades.

A Eliana Asanza por sus consejos y palabras que me reconfortaron.

Andrea Katherine Espinoza Iñiguez

Agradecimientos

Mi más sincero agradecimiento a mi papá y a mi mamá, porque sin sus consejos y palabras de aliento, incluso en los momentos en que dudaba de mí mismo, no habría llegado hasta aquí.

A toda mi familia, especialmente a mis primas Andrea Marcillo y Adriana Marcillo, por sus consejos, ayuda y apoyo en los momentos difíciles.

Al Ph.D. Carlos Saldarriaga y al Ph.D. Marcelo Fajardo, por su guía, carisma y paciencia en esta etapa final.

A mi compañera de tesis, Andrea Espinoza, gracias por tu compromiso y por el excelente trabajo en equipo.

Al club de mecatrónica, donde conocí a los mejores amigos que la vida me pudo dar, quienes, sin saberlo, me dieron el valor para seguir adelante.

A mis más cercanos amigos, por estar a mi lado y ayudarme a seguir adelante cuando sentí que no podía con todo.

Fabricio Ricardo Quimis Marcillo

Declaración Expresa

Nosotros Andrea Katherine Espinoza Iñiguez y Fabricio Ricardo Quimis Marcillo, acordamos y reconocemos que:

La titularidad de los derechos patrimoniales de autor (derechos de autor) del proyecto de graduación corresponderá al autor o autores, sin perjuicio de lo cual la ESPOL recibe en este acto una licencia gratuita de plazo indefinido para el uso no comercial y comercial de la obra con facultad de sublicenciar, incluyendo la autorización para su divulgación, así como para la creación y uso de obras derivadas. En el caso de usos comerciales se respetará el porcentaje de participación en beneficios que corresponda a favor del autor o autores.

La titularidad total y exclusiva sobre los derechos patrimoniales de patente de invención, modelo de utilidad, diseño industrial, secreto industrial, software o información no divulgada que corresponda o pueda corresponder respecto de cualquier investigación, desarrollo tecnológico o invención realizada por mí/nosotros durante el desarrollo del proyecto de graduación, pertenecerán de forma total, exclusiva e indivisible a la ESPOL, sin perjuicio del porcentaje que me/nos corresponda de los beneficios económicos que la ESPOL reciba por la explotación de mi/nuestra innovación, de ser el caso.

En los casos donde la Oficina de Transferencia de Resultados de Investigación (OTRI) de la ESPOL comunique los autores que existe una innovación potencialmente patentable sobre los resultados del proyecto de graduación, no se realizará publicación o divulgación alguna, sin la autorización expresa y previa de la ESPOL.

Guayaquil, 23 de mayo del 2024.



Andrea Katherine
Espinoza Iñiguez



Fabricio Ricardo Quimis
Marcillo

Evaluadores

Marcelo Fajardo, Ph.D.

Profesor de Materia

Carlos Saldarriaga, Ph.D.

Tutor de proyecto

Resumen

El proyecto se centra en la optimización dinámica de robots redundantes, cruciales en la industria moderna debido a su flexibilidad y precisión. El objetivo principal fue desarrollar un algoritmo que permita mejorar las respuestas dinámicas de estos robots, minimizando vibraciones y maximizando su estabilidad robótica y eficiencia operativa. Para lograrlo, se utilizaron herramientas como Python y bibliotecas específicas (NumPy, SymPy, Matplotlib) para simular y analizar diferentes configuraciones del robot. El proyecto identificó y analizó las frecuencias naturales y los coeficientes de amortiguamiento mediante la Transformada Rápida de Fourier, permitiendo ajustar los parámetros de control de manera precisa. Los resultados mostraron una mejora significativa en la estabilidad del robot, especialmente en configuraciones con frecuencias naturales más altas, que resultaron en una mayor agilidad, aunque con un mayor riesgo de resonancia. Finalmente, se concluye que la optimización de la modulación dinámica es esencial para lograr un rendimiento óptimo en robots redundantes, y se recomienda seguir perfeccionando el algoritmo y aplicarlo en diversos entornos industriales.

Palabras Clave: Optimización dinámica, robots redundantes, Transformada Rápida de Fourier, modulación de vibraciones, estabilidad robótica.

Abstract

The project focuses on the dynamic optimization of redundant robots, which are crucial in modern industry due to their flexibility and precision. The main objective was to develop an algorithm that improves the dynamic responses of these robots, minimizing vibrations and maximizing their robotic stability and operational efficiency. To achieve this, tools such as Python and specific libraries (NumPy, SymPy, Matplotlib) were used to simulate and analyze different robot configurations. The project identified and analyzed the natural frequencies and damping coefficients using the Fast Fourier Transform (FFT), allowing for precise control parameter adjustments. The results showed a significant improvement in the robot's stability, especially in configurations with higher natural frequencies, which led to greater agility, although with an increased risk of resonance. Finally, it is concluded that dynamic modulation optimization is essential for achieving optimal performance in redundant robots, and it is recommended to continue refining the algorithm and applying it in various industrial environments.

Keywords: *Dynamic optimization, redundant robots, Fast Fourier Transform, vibration modulation, robotic stability.*

Índice general

Resumen.....	I
<i>Abstract</i>	II
Índice general	III
Abreviaturas	V
Simbología	VI
Índice de figuras.....	VII
Índice de tablas.....	IX
Capítulo 1	1
1.1 Introducción.....	2
1.2 Descripción del problema.....	4
1.3 Justificación del problema	5
1.4 Objetivos.....	6
1.4.1 Objetivo general	6
1.4.2 Objetivos específicos.....	6
1.5 Marco Teórico	6
1.5.1 Robots Redundantes	6
1.5.2 Modelado de sistemas dinámicos	7
1.5.3 Cinemática Inversa	9
1.5.4 Frecuencias Naturales.....	10
1.5.5 Transformada Rápida de Fourier (FFT)	10
1.5.6 Sistema similar a un sistema Resorte-Amortiguador	10
1.5.7 Teoría de vibraciones	11
1.5.8 Metodología analítica para el análisis de vibraciones	13
1.5.9 Estado del arte	13
Capítulo 2.....	15
2.1 Metodología.....	16

2.2	Definición óptima de problemática	16
2.3	Selección de alternativas.....	17
2.3.1	Requerimientos.....	17
2.3.2	Alternativas de diseño	18
2.3.3	Matriz de decisión	20
2.4	Proceso de diseño	20
2.4.1	Metodología del diseño	20
2.4.2	Selección de herramientas y Entorno de Desarrollo	21
2.4.3	Desarrollo del algoritmo.....	22
Capítulo 3	28
3.1	Resultados y análisis.....	29
3.1.1	Primera configuración	30
3.1.2	Segunda configuración.....	38
3.1.3	Tercera configuración.....	43
3.1.4	Cuarta Configuración	48
3.1.5	Análisis Comparativo de las Configuraciones	52
3.1.6	Análisis de control.....	53
3.1.7	Análisis de costos	55
Capítulo 4	57
4.1	Conclusiones y recomendaciones	58
4.1.1	Conclusiones	58
4.1.2	Recomendaciones.....	59
REFERENCIAS	61
APÉNDICES	64

Abreviaturas

DFT	Discrete Fourier Transform
DoF	Degrees of Freedom
FFT	Fast Fourier Transform
GDL	Grados de Libertad
QNRB	Non Rigid Body
RB	Rigid Body
NZP	Null Zero Potential
ZP	Zero Potential

Simbología

°	Grados
Hz	Hertz
kg	Kilogramos
m	Metros
rad	Radianes
s	Segundos

Índice de figuras

Figura 1.1 Robots en la industria manufacturera	2
Figura 1.2 Robots de servicio para uso profesional por aplicación.....	3
Figura 1.3 Cinemática inversa de un robot de 6 GDL.....	7
Figura 1.4 Ley de control de impedancia	9
Figura 1.5 Sistema masa-resorte-amortiguador de 3 GDL no restringido y no conservativo.....	11
Figura 1.6 Modo de vibración de un sistema de 3GDL	12
Figura 2.1 Metodología del diseño	21
Figura 2.2 Algoritmo que lleva a cabo el código	25
Figura 2.3 Respuesta de q_t en función de t	27
Figura 3.1 Gráfica de QNZP primera configuración.....	33
Figura 3.2 Gráfica de $Q(t)$ primera configuración	34
Figura 3.3 FFT del primer eslabón de la primera configuración.....	35
Figura 3.4 FFT del segundo eslabón de la primera configuración.....	36
Figura 3.5 FFT del tercer eslabón de la primera configuración	37
Figura 3.6 Gráfica de QNBZ segunda configuración	39
Figura 3.7 Gráfica de $Q(t)$ segunda configuración.....	40
Figura 3.8 FFT del primer eslabón de la segunda configuración.....	41
Figura 3.9 FFT del segundo eslabón de la segunda configuración	41
Figura 3.10 FFT del tercer eslabón de la segunda configuración	42
Figura 3.11 Gráfica de QNBZ tercera configuración.....	44
Figura 3.12 Gráfica de $Q(t)$ tercera configuración.....	45
Figura 3.13 FFT del primer eslabón de la tercera configuración	46
Figura 3.14 FFT del segundo eslabón de la tercera configuración	46
Figura 3.15 FFT del tercer eslabón de la tercera configuración.....	47
Figura 3.16 Gráfica de QNBZ cuarta configuración	49
Figura 3.17 Gráfica de $Q(t)$ cuarta configuración	50
Figura 3.18 FFT del primer eslabón de la cuarta configuración	51
Figura 3.19 FFT del segundo eslabón de la cuarta configuración.....	51
Figura 3.20 FFT del tercer eslabón de la cuarta configuración.....	52
Figura 5.1 Respuesta q_{rnb} con una fuerza impulso externa.....	68
Figura 5.2 Respuesta total con una fuerza impulso externa	68
Figura 5.3 Matriz de inercia	76

Figura 5.4 Matriz articular de rigidez.....	76
Figura 5.5 Matriz de amortiguamiento.....	76
Figura 5.6 QZP del sistema.....	77
Figura 5.7 Valores propios.....	77
Figura 5.8 Ecuaciones de QNZP+.....	77

Índice de tablas

Tabla 2.1 Requerimientos clave del sistema	17
Tabla 2.2 Alternativas de solución	18
Tabla 2.3 Criterios de diseño.....	19
Tabla 2.4 Matriz de decisión	20
Tabla 2.5 Comparación entre sistemas mecánicos discretos y robóticos.....	23
Tabla 3.1 Parámetros primera configuración	30
Tabla 3.2 Resultados primera configuración.....	32
Tabla 3.3 Parámetros segunda configuración.....	38
Tabla 3.4 Resultados segunda configuración	39
Tabla 3.5 Parámetros tercera configuración.....	43
Tabla 3.6 Resultados tercera configuración	44
Tabla 3.7 Parámetros cuarta configuración.....	48
Tabla 3.8 Resultados cuarta configuración	49
Tabla 3.9 Razón de amortiguamiento y frecuencia naturales de la primera configuración	54
Tabla 3.10 Razón de amortiguamiento y frecuencias naturales de la segunda configuración.....	54
Tabla 3.11 Razón de amortiguamiento y frecuencias naturales de la tercera configuración	54
Tabla 3.12 Razón de amortiguamiento y frecuencias naturales de la cuarta configuración	54
Tabla 3.13 Costos Asociados	55

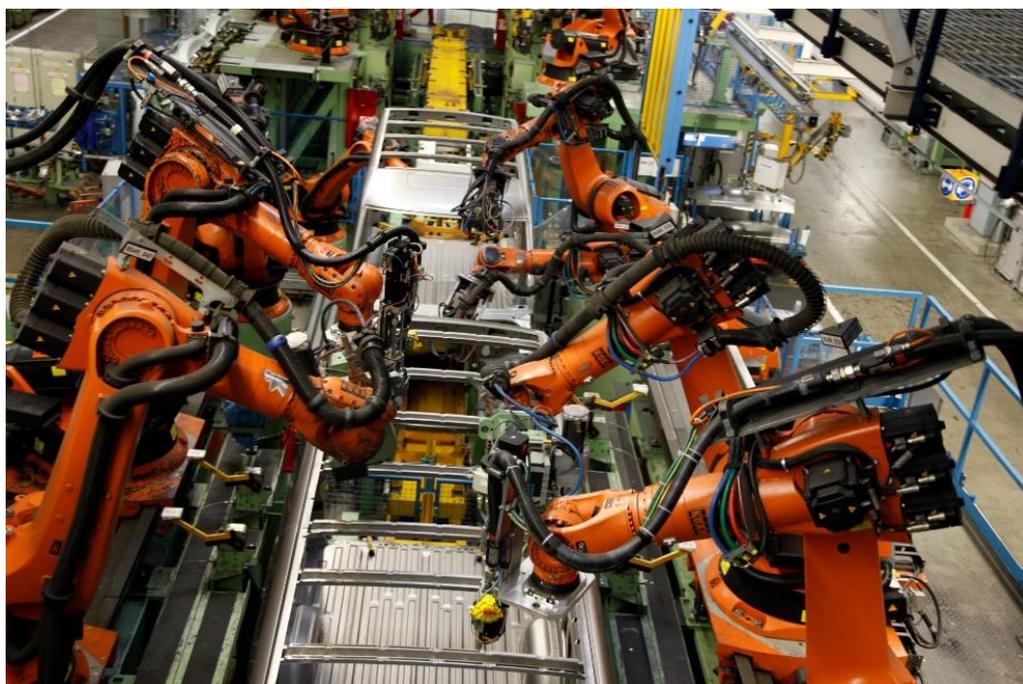
CAPÍTULO 1

1.1 Introducción

En un mundo cada vez más orientado hacia la automatización y la eficiencia industrial, la optimización de los sistemas robóticos juega un papel crucial en la mejora de la productividad y la calidad en sectores como las industrias manufactureras evidenciado en **Figura 1.1** el transporte y el almacenamiento. Los avances en robótica permiten a las empresas mejorar sus procesos, reducir costos y aumentar la precisión y velocidad de sus operaciones [1]. Sin embargo, el diseño e implementación de procedimientos que permitan una optimización dinámica de sistemas robóticos se vuelve esencial para enfrentar los desafíos inherentes a la complejidad de estos sistemas.

Figura 1.1

Robots en la industria manufacturera



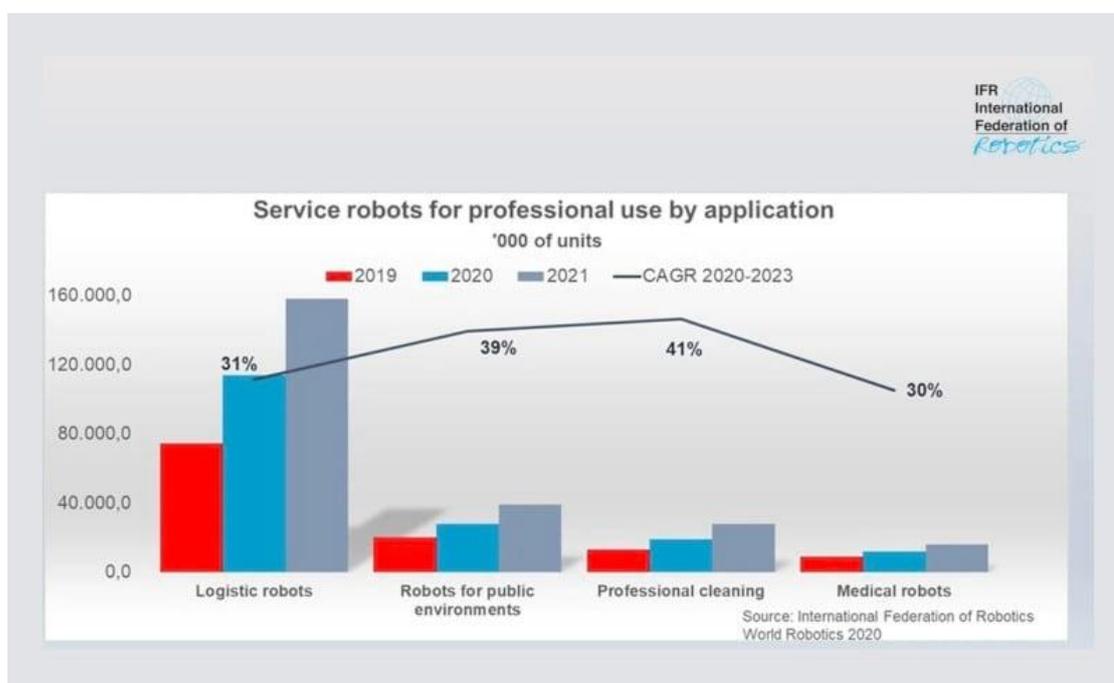
Nota: Imagen extraída de [2]

La robótica moderna no solo se centra en la ejecución de tareas repetitivas y precisas, sino que también busca adaptarse a entornos cambiantes y a la realización de actividades cada vez más complejas. Esta evolución es evidente en diversas aplicaciones profesionales de la robótica, como

se muestra en la **Figura 1.2** aquí es donde la redundancia en los robots, que se refiere a la presencia de más grados de libertad de los estrictamente necesarios para realizar una tarea, presenta tanto oportunidades como desafíos. Por un lado, permite una mayor flexibilidad y capacidad de maniobra, facilitando la ejecución de movimientos complejos y la adaptación a obstáculos imprevistos. Por otro, introduce complicaciones adicionales en el modelado y control dinámico de estos sistemas, dado que cada grado de libertad adicional incrementa la complejidad del sistema de control necesario para gestionar el robot de manera eficiente. [3]

Figura 1.2

Robots de servicio para uso profesional por aplicación



Nota: Imagen extraída de [4]

El desarrollo de metodologías avanzadas para la optimización dinámica de robots redundantes resulta fundamental para maximizar su rendimiento y utilidad. En este contexto, emplear técnicas como la Transformada Rápida de Fourier (FFT) para identificar los modos de vibración predominantes y ajustar los parámetros dinámicos puede transformar significativamente el rendimiento de los robots redundantes. La FFT permite descomponer las señales en sus

componentes frecuenciales, facilitando la identificación de frecuencias críticas que afectan el comportamiento dinámico del robot[5].

Además, es crucial considerar el impacto de la optimización dinámica no solo en términos de eficiencia y precisión operativa, sino también en la reducción del desgaste mecánico y el consumo energético. Al minimizar las vibraciones y ajustar adecuadamente los parámetros dinámicos, se puede prolongar la vida útil de los componentes del robot y mejorar su sostenibilidad. Este enfoque contribuye a la creación de sistemas robóticos más robustos y confiables, capaces de operar de manera efectiva en entornos industriales exigentes[6].

La optimización dinámica de sistemas robóticos, por lo tanto, no es solo una cuestión de mejorar el rendimiento inmediato, sino también de asegurar la longevidad y sostenibilidad del sistema en su conjunto. La integración de técnicas avanzadas de análisis y control en el diseño de robots redundantes abre nuevas posibilidades para su aplicación en una amplia variedad de sectores, desde la manufactura y el transporte hasta la medicina y la exploración espacial.

1.2 Descripción del problema

El diseño y control de robots redundantes presentan desafíos significativos debido a la complejidad inherente en la identificación de sus parámetros dinámicos. Esta complejidad se deriva de la no linealidad de las expresiones dinámicas y la falta de datos exhaustivos sobre las características físicas y geométricas de los componentes. Además, obtener respuestas dinámicas basadas en parámetros geométricos de forma analítica es una tarea ardua y a menudo imprecisa.

Para superar estos desafíos, es necesario desarrollar una metodología precisa que permita identificar y ajustar los parámetros dinámicos más influyentes en el comportamiento de los robots redundantes. Esta metodología debe optimizar la eficiencia operativa de los robots, reduciendo las vibraciones y el desgaste mecánico para prolongar su vida útil y mejorar la precisión en la ejecución de tareas.

El proyecto abordará estos problemas mediante simulaciones de respuestas dinámicas y la aplicación de la Transformada Rápida de Fourier (FFT) para identificar y ajustar los modos de movimiento predominantes. Esta aproximación no solo mejorará la eficiencia y precisión de los robots en sectores industriales clave, sino que también contribuirá a la sostenibilidad mediante la reducción del consumo de energía y el desgaste mecánico de los componentes.

1.3 Justificación del problema

Resolver esta problemática es crucial debido a su impacto directo en la eficiencia y productividad de los procesos industriales. La optimización dinámica de robots redundantes no solo mejora la calidad y precisión en la ejecución de tareas, sino que también contribuye a la reducción de costos operativos al minimizar errores y desperdicios [7].

Además, esta optimización tiene importantes implicaciones en términos de sostenibilidad y rentabilidad para las empresas. Al prolongar la vida útil de los componentes y reducir el consumo de energía, se promueve una operación más eficiente y respetuosa con el medio ambiente [8]. La mejora en la eficiencia de los motores y actuadores de los robots también resulta en una operación más suave, reduciendo el mantenimiento requerido y aumentando la confiabilidad del sistema.

En un contexto más amplio, la optimización de robots redundantes puede abrir nuevas oportunidades de negocio al permitir la automatización de procesos más complejos que anteriormente eran inviables o demasiado costosos [9]. Esto no solo mejora la competitividad de las empresas en el mercado global, sino que también facilita la adopción de tecnologías robóticas avanzadas en pequeñas y medianas empresas, que anteriormente no podían acceder a estas soluciones debido a limitaciones de costo o complejidad técnica [10].

1.4 Objetivos

1.4.1 Objetivo general

Desarrollar un código basado en una función de optimización de parámetros dinámicos de robots redundantes para mejorar las respuestas dinámicas de robots redundantes y así obtener mejor precisión y eficiencia operativa.

1.4.2 Objetivos específicos

1. Identificar y manipular las redundancias en el sistema para simplificar la modelación y análisis dinámico.
2. Generar gráficos que combinen las soluciones de movimiento rígido y no rígido para representar la respuesta global de vibración del sistema basado en la ecuación de movimiento formulada.
3. Determinar las frecuencias naturales con mayor impacto en el comportamiento dinámico del sistema utilizando técnicas basadas en la Transformada Rápida de Fourier (FFT).
4. Desarrollar un algoritmo que optimice la modulación dinámica en robots redundantes a partir de sus parámetros físicos y de control, adaptándolo a diferentes escenarios operativos, para reducir las vibraciones y el desgaste mecánico y así prolongar la vida útil de los componentes.

1.5 Marco Teórico

1.5.1 Robots Redundantes

Los robots redundantes tienen más grados de libertad de los necesarios para realizar una tarea específica [11]. Esta redundancia ofrece una mayor flexibilidad y adaptabilidad en la ejecución de tareas, aunque también añade una complejidad extra en el control y modelado del robot [12]. Se puede observar un ejemplo en la figura **Figura 1.3**.

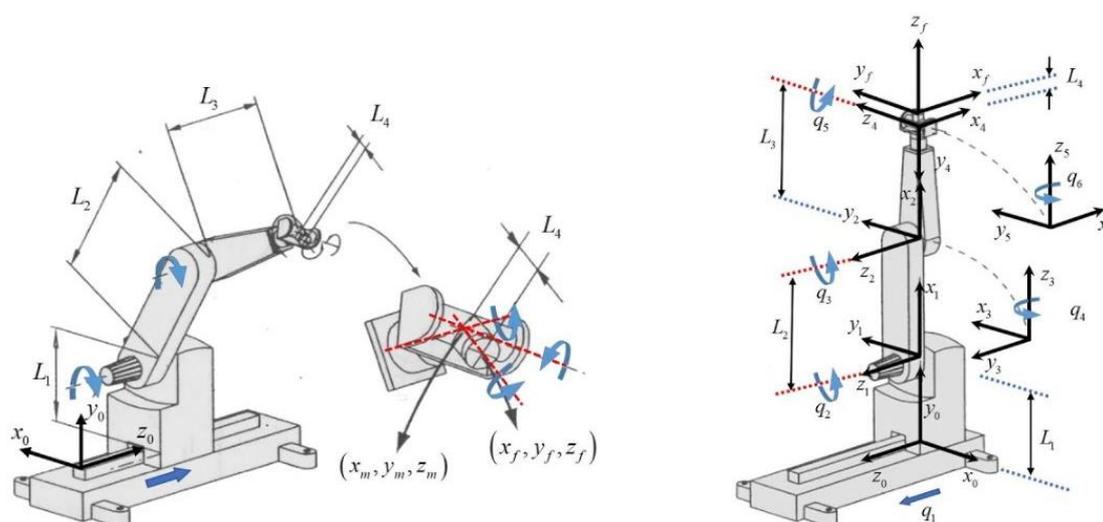
Muchas configuraciones de estos robots presentan ventajas significativas en comparación con los robots convencionales, tales como una mayor tolerancia a fallos y la capacidad de sortear obstáculos [11]. Entre estas ventajas se incluye la optimización de la fuerza de los actuadores, logrando encontrar la solución que mejor proyecte la fuerza en el efector final [12].

Principalmente, estos robots se utilizan en procesos industriales donde la eficiencia y la eficacia son primordiales [7]. La capacidad de adaptarse rápidamente a los tiempos de fabricación en diversas industrias es una de sus ventajas más destacadas, permitiendo una producción más flexible y eficiente [8]. Además, su habilidad para realizar tareas complejas con alta precisión los hace indispensables en entornos donde los errores pueden tener consecuencias significativas [13].

Un ejemplo más claro es en el área médica en donde los robots redundantes pueden ser utilizados para cirugías de alta precisión, donde la capacidad de evitar errores es crítica [13].

Figura 1.3

Cinemática inversa de un robot de 6 GDL



Nota: Imagen extraída de [14]

1.5.2 Modelado de sistemas dinámicos

En un sistema mecánico dinámico como lo es un robot, las ecuaciones de movimiento describen la relación entre las fuerzas actuantes y los desplazamientos, velocidades y aceleraciones

presentes. Para sistemas de múltiples grados de libertad, estas ecuaciones se expresan en forma matricial como:

$$M[q(t)]\ddot{q}(t) + h[q(t), \dot{q}(t)] + v[q(t)] = \tau(t) + \tau_{ext} \quad (1.1)$$

Donde $\tau(t)$ es el vector (n) de par generalizado aplicado a las ecuaciones, $q(t)$ es el vector (n) de las variables de las articulaciones del brazo, $\dot{q}(t)$ y $\ddot{q}(t)$ son los vectores (n) de la velocidad y aceleración del brazo, $M[q(t)]$ es la matriz (n x n) simétrica inercial denominada matriz de inercia, $h[q(t), \dot{q}(t)]$ es el vector de fuerzas (n) de fuerzas de Coriolis y centrífuga y por último $v[q(t)]$ es el vector (n) de las fuerzas de carga gravitatoria[15].

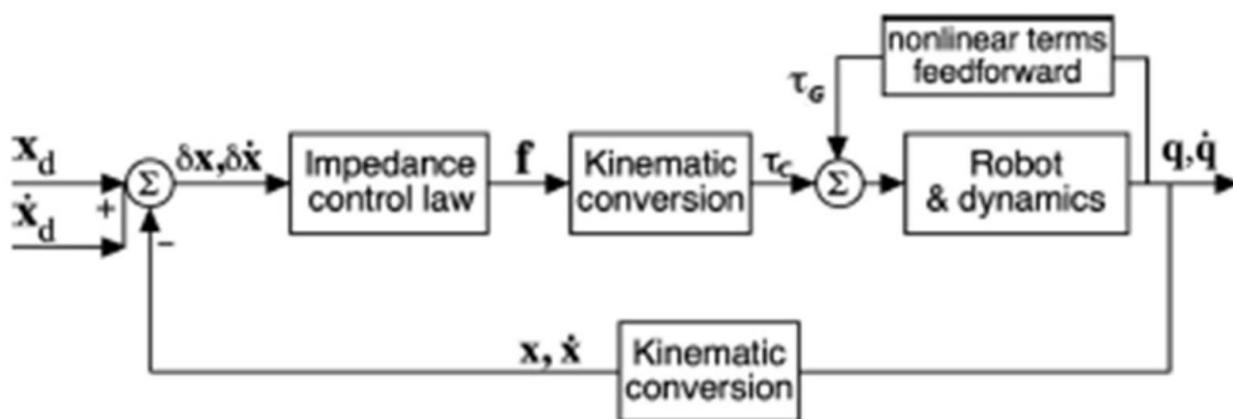
En el contexto de control de robots, estas ecuaciones de movimiento se complementan con una ley de control de impedancia como se muestra en la **Figura 1.4**. En el diagrama de bloques, el control de impedancia se utiliza para calcular el par de control τ_c necesario para ajustar la posición y velocidad del extremo del robot al objetivo deseado, tomando en cuenta la interacción con el entorno. El término τ_G se añade como un componente de alimentación adelantada para compensar los efectos no lineales, como la gravedad o fuerzas externas, resultando en el par total aplicado τ . Esta implementación permite que el robot sea capaz de adaptarse dinámicamente a su entorno mientras minimiza vibraciones y optimiza su comportamiento dinámico.

Figura 1.4

Ley de control de impedancia

$$\tau_c = \mathbf{J}^T [\mathbf{K}_C(\mathbf{x} - \mathbf{x}_d) + \mathbf{C}_C(\dot{\mathbf{x}} - \dot{\mathbf{x}}_d)]$$

$$\tau = \tau_c + \tau_G$$



1.5.3 Cinemática Inversa

La cinemática inversa es una de las principales herramientas en el control de robots, permitiendo determinar las posiciones angulares de las articulaciones para que el efector final alcance una posición y orientación deseada. En [31] se explica que la cinemática inversa involucra resolver ecuaciones no lineales que relacionan la posición del efector final con las variables articulares del robot. El libro detalla tanto métodos geométricos como numéricos para resolver este problema. Los métodos geométricos, que se basan en relaciones espaciales, son eficaces para robots con configuraciones sencillas. Por otro lado, los métodos numéricos, como el algoritmo de Newton-Raphson, se utilizan para robots más complejos, donde pueden existir múltiples soluciones para la misma posición final del efector. Estos métodos permiten calcular con precisión los ángulos articulares necesarios, asegurando que el robot realice movimientos exactos y eficientes, un aspecto crucial para aplicaciones que requieren alta precisión y repetibilidad.

1.5.4 Frecuencias Naturales

Las frecuencias naturales de un sistema son aquellas a las que el sistema tiende a oscilar en ausencia de amortiguamiento y excitación externa. Conocer estas frecuencias es crucial para predecir y controlar el comportamiento dinámico del robot. Las frecuencias naturales están determinadas por las propiedades inerciales y rígidas del robot y se calculan resolviendo el problema de valores propios asociado con el sistema de ecuaciones diferenciales:

$$(K - \omega^2 M)\phi = 0 \quad (1.2)$$

donde ω representa las frecuencias naturales y ϕ son los modos de vibración correspondientes. Minimizar las vibraciones no deseadas ajustando estas frecuencias naturales es esencial para evitar resonancias que afecten negativamente la precisión y estabilidad del robot[13].

1.5.5 Transformada Rápida de Fourier (FFT)

La Transformada Rápida de Fourier (FFT) es una técnica eficiente para computar la Transformada de Fourier Discreta (DFT) y se utiliza para analizar las frecuencias de un sistema dinámico. La FFT permite descomponer una señal en sus componentes frecuenciales, facilitando la identificación de las frecuencias naturales predominantes en el comportamiento dinámico del robot. Este análisis es crucial para ajustar los parámetros dinámicos y mejorar el rendimiento del sistema [16]. En robots redundantes la FFT ayuda identificar modos de vibración críticos permitiendo diseñar estrategias de control que mitiguen las vibraciones, mejorando la precisión y prolongando la vida útil del robot. En aplicación como la cirugía minimizar vibraciones es esencial para asegurar procedimientos precisos y seguros[13].

1.5.6 Sistema similar a un sistema Resorte-Amortiguador

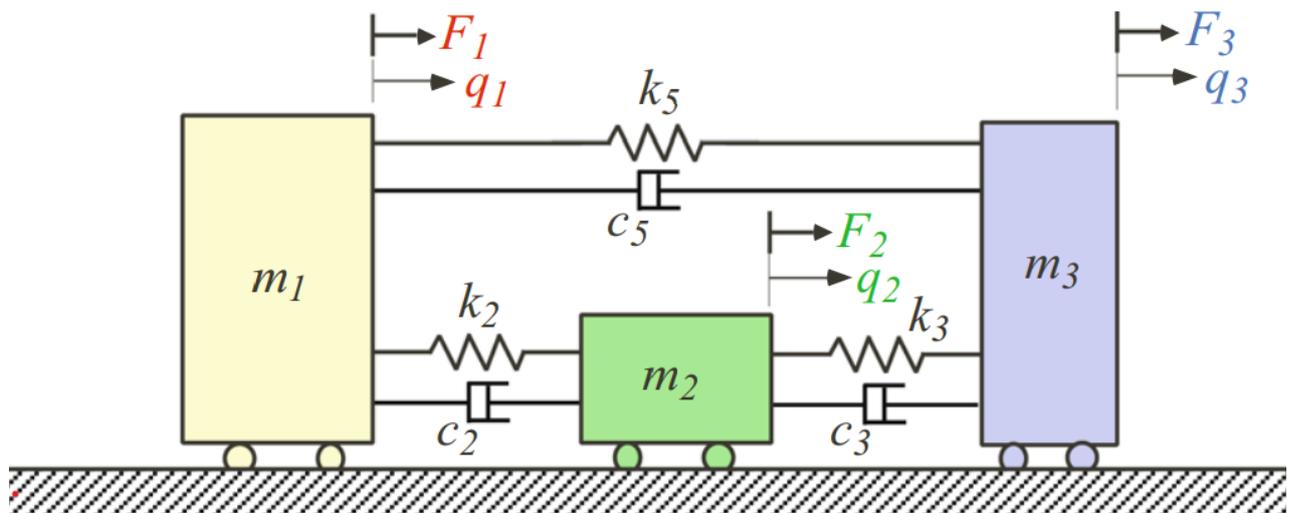
Un modelo simplificado para entender la dinámica de robots redundantes es el sistema resorte-amortiguador, donde el comportamiento de los componentes del robot se compara con el de resortes y amortiguadores **Figura 1.5**. En este modelo, la matriz de rigidez K representa las

propiedades elásticas, mientras que la matriz de amortiguamiento C modela las fuerzas disipativas en el sistema[17]. Modelar el robot como un sistema de este tipo permite predecir las respuestas dinámicas ante diferentes excitaciones y ajustar los parámetros de control en consecuencia. Esto es particularmente útil en aplicaciones donde la precisión es crítica, como en la fabricación de componentes electrónicos o en la industria aeroespacial [7]

Además, la implementación del control de impedancia, desarrollado por Neville Hogan, permite que el robot ajuste dinámicamente su rigidez y amortiguamiento para interactuar de manera segura y precisa con su entorno. Este tipo de control es esencial en aplicaciones donde la interacción con el entorno debe ser regulada con precisión, como en la cirugía robótica o en tareas de ensamblaje delicadas. Mediante el control de impedancia, se puede mejorar significativamente la estabilidad y la precisión del robot al manejar perturbaciones externas [18].

Figura 1.5

Sistema masa-resorte-amortiguador de 3 GDL no restringido y no conservativo



Nota: Imagen extraída de [19]

1.5.7 Teoría de vibraciones

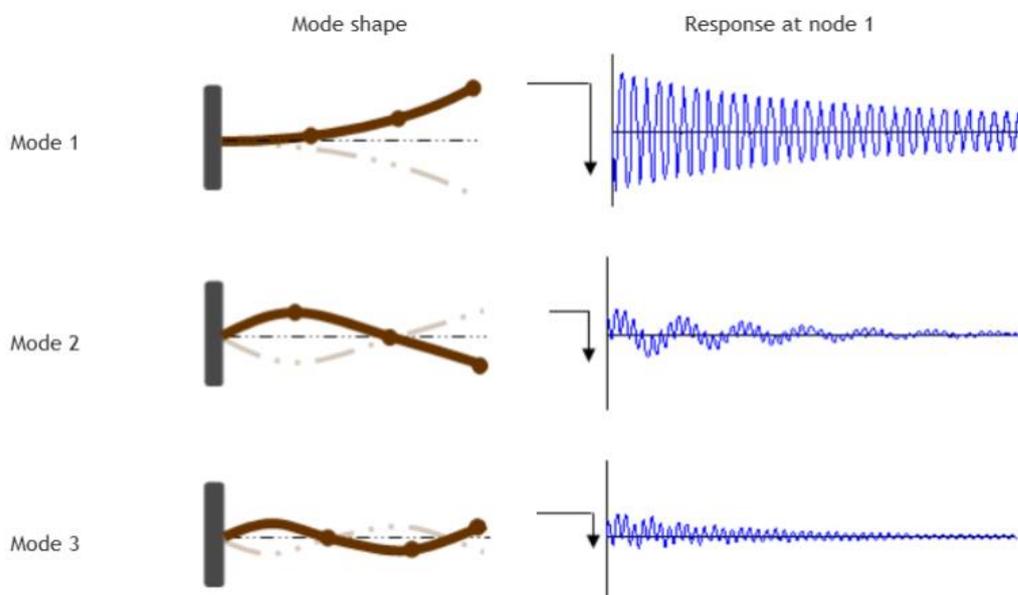
La teoría de vibraciones se ocupa del estudio del movimiento oscilatorio de los sistemas físicos y es fundamental en el análisis de la dinámica de robots. En el contexto de los robots

redundantes, la teoría de vibraciones ayuda a comprender cómo las diferentes partes del robot responden a diversas excitaciones y cómo se pueden controlar estas respuestas para mejorar el rendimiento y la precisión del robot. En la **Figura 1.6** tenemos un sistema reaccionando a vibraciones.

Un aspecto crucial de la teoría de vibraciones es el análisis de los modos propios y las frecuencias naturales de un sistema. Los modos propios son las formas específicas en que un sistema puede oscilar naturalmente, y cada modo propio está asociado con una frecuencia natural. En un robot redundante, es esencial identificar estos modos y frecuencias para diseñar estrategias de control que minimicen las vibraciones no deseadas y eviten la resonancia, que puede causar daños o fallos en el sistema[20]. El análisis de vibraciones también incluye el estudio de la amortiguación, que es el proceso por el cual la energía de las vibraciones se disipa en un sistema. La amortiguación es crucial para controlar las vibraciones en robots redundantes, ya que permite reducir la amplitud de las oscilaciones y mejorar la estabilidad del robot. La teoría de vibraciones proporciona las herramientas necesarias para modelar y analizar la amortiguación en sistemas complejos, permitiendo diseñar robots que operen de manera más eficiente y precisa[21].

Figura 1.6

Modo de vibración de un sistema de 3GDL



Nota: Imagen extraída de [22]

1.5.8 Metodología analítica para el análisis de vibraciones

El análisis de vibraciones en robots redundantes puede beneficiarse de metodologías analíticas avanzadas. En este contexto, la metodología presentada en el artículo "Analytical methodology for the analysis of vibration for unconstrained discrete systems and applications to impedance control of redundant robots" [19] es particularmente relevante. Esta metodología se enfoca en la síntesis de sistemas descritos por matrices de masa, amortiguamiento y rigidez, utilizando conceptos cinemáticos fundamentales de movimientos de cuerpo rígido y no rígido. El modo de cuerpo rígido no resulta en un cambio neto en la energía potencial de la matriz de rigidez del sistema mecánico discreto de múltiples grados de libertad (DoF).

La metodología se ilustra mediante un sistema mecánico discreto no restringido, demostrando cómo obtener respuestas de vibración libre y forzada, así como las características dinámicas del sistema en términos de frecuencia natural (ω_n) y razón de amortiguamiento (ζ). Además, se aplica al control de impedancia de robots redundantes, donde el modo de cuerpo rígido es equivalente a los movimientos de un robot redundante que no resultan en un cambio neto en la energía potencial, también conocido como el modo de potencial cero (ZP)[19]

1.5.9 Estado del arte

Actualmente, disponemos de herramientas avanzadas para el análisis dinámico de robots, que facilitan la simulación y evaluación de sus comportamientos. Sin embargo, estas herramientas aún tienen limitaciones en proporcionar información analítica detallada sobre el comportamiento dinámico de robots redundantes. Las ecuaciones que describen estos sistemas son altamente no lineales y complejas, lo que requiere un gran poder computacional y técnicas avanzadas para un análisis preciso. Por esta razón, es fundamental desarrollar nuevos métodos que aborden estas limitaciones y ofrezcan una comprensión más profunda y detallada de la dinámica de robots redundantes

1.5.9.1 Métodos Numéricos

Los métodos numéricos, como el Método de Elementos Finitos (FEM) y el Método de Diferencias Finitas (FDM), son esenciales para analizar sistemas complejos como los robots redundantes, permitiendo la solución aproximada de ecuaciones diferenciales complejas. Sin embargo, estos métodos no proporcionan información analítica detallada sobre el comportamiento dinámico debido a su naturaleza aproximada, lo cual puede afectar la precisión en optimizaciones finas y ajustes dinámicos en tiempo real [23] [24]. Además, la necesidad de discretización del dominio y el alto poder computacional requerido pueden introducir errores numéricos y limitaciones en la resolución, aunque siguen siendo herramientas valiosas para el análisis preliminar y diseño de sistemas robóticos [25][26].

1.5.9.2 ANSYS

Es un software de simulación multifísica ampliamente utilizado para modelar y simular el comportamiento dinámico de robots, incluyendo robots redundantes, bajo diferentes condiciones operativas. A pesar de sus capacidades avanzadas, ANSYS presenta limitaciones al analizar en profundidad robots redundantes debido a la complejidad en el modelado de sistemas con múltiples grados de libertad y la necesidad de alta resolución para capturar detalles dinámicos precisos [27][28]. Además, la simulación de largas duraciones o sistemas con alta frecuencia de vibración puede ser computacionalmente costosa, limitando su uso en análisis de tiempo real, aunque sigue siendo una herramienta poderosa para la validación de diseños robóticos [29], [30].

CAPÍTULO 2

2.1 Metodología

En el presente capítulo se expuso la metodología empleada para la creación de la solución, además de la capacidad para cumplir con los objetivos establecidos. Se especificaron las soluciones y se describió el proceso mediante el cual se llegó a ellas, basándose en la descripción del problema y centrándose en cumplir con las especificaciones. A continuación, se ofreció una descripción más detallada de la solución mencionada, con un enfoque en el análisis de los diferentes sistemas o robots redundantes.

2.2 Definición óptima de problemática

Principalmente, se comenzó por una revisión exhaustiva de la literatura para identificar las características clave y los desafíos asociados con el análisis de la dinámica de robots redundantes en diferentes áreas en la que se desempeñan. Esto permitió establecer una comprensión clara de la problemática y de las posibles soluciones en cuanto a análisis de vibraciones que abordaron los diferentes autores como [20] donde los sistemas son conservadores y definidos positivos. Según [18] el control de impedancia se ocupa de la interacción del robot con su entorno y se modela como un sistema multidimensional de masa y resorte-amortiguador, similar a los sistemas discretos en la teoría de la vibración mecánica. Incluso, la metodología aquí es útil para casos donde el robot es redundante, que requieren especial atención porque son sistemas mecánicos discretos no conservadores y sin restricciones. Para llevar a cabo el análisis de la dinámica y la respuesta de vibración de robots redundantes, se ha desarrollado una secuencia de pasos específica [19]. Sin embargo, aún no existe un código general que abarque completamente el análisis de la dinámica para cualquier robot redundante. Desarrollar dicho código sería muy útil para ahorrar tiempo en el análisis de la dinámica y, de este modo, mejorar de manera más eficiente las tareas de los manipuladores.

2.3 Selección de alternativas

2.3.1 Requerimientos

En el proceso de desarrollo de un sistema robótico optimizado, es fundamental establecer un conjunto de requerimientos claros y bien definidos que orienten el diseño, implementación y evaluación del sistema. Estos requerimientos aseguran que el sistema no solo cumpla con las especificaciones técnicas necesarias, sino que también sea adaptable, escalable, confiable y económicamente viable. A continuación, la **Tabla 2.1** resume los requerimientos clave que deben ser considerados en la función del diseño y optimización del robot.

Tabla 2.1

Requerimientos clave del sistema

Requerimientos	
Adaptabilidad	Número de configuraciones geométricas soportadas sin necesidad de ajustes significativos.
Escalabilidad	Capacidad del sistema para mantener el rendimiento con un aumento de la carga de trabajo.
Confiability	Tasa de errores durante las operaciones críticas.
Costo	Costo asociado con la implementación y ejecución del sistema, incluyendo recursos computacionales y tiempo de desarrollo.
Eficiencia	Reducción porcentual en el tiempo de inactividad y aumento en la productividad
Mantenimiento	Tiempo promedio necesario para aplicar actualizaciones
Sostenibilidad	Reducción porcentual en el consumo de energía en comparación con sistemas previos similares.

2.3.2 Alternativas de diseño

Tabla 2.2

Alternativas de solución

Alternativa 1	Optimizar las funciones obtenidas mediante diferentes métodos numéricos, ajustándolas a prueba y error hasta alcanzar la respuesta requerida.
Alternativa 2	Generar una función basada en datos de machine learning a través de resultados obtenidos con métodos analíticos u otros diversos
Alternativa 3	A partir de una metodología analítica, parametrizar y optimizar las respuestas dinámicas de un robot manipulador redundante.

La **Tabla 2.3** presenta una comparación de diversos criterios de evaluación, indicando, en una escala relativa, la importancia de un criterio sobre otro, con el fin de determinar la prioridad y el peso de cada uno.

Tabla 2.3

Criterios de diseño

Criterio	Adaptabilidad	Escalabilidad	Confiabilidad	Costo	Eficiencia	Mantenimiento	Sostenibilidad	$\Sigma+1$	Ponderación
Adaptabilidad	-	1	0.5	0.5	1	0.5	0.5	4	0.15
Escalabilidad	0	-	1	1	1	1	0.5	4.5	0.25
Confiabilidad	0	0	-	0.5	0.5	0.5	0.5	2	0.10
Costo	0.5	0	0.5	-	1	1	0.5	3.5	0.20
Eficiencia	0	0	0.5	0	-	0.5	0.5	1.5	0.15
Mantenimiento	0	0	0	0	0	-	0.5	0.5	0.05
Sostenibilidad	0	0	0.5	0.5	0.5	0	-	1.5	0.10
							SUMA	17.5	1

2.3.3 Matriz de decisión

Tabla 2.4

Matriz de decisión

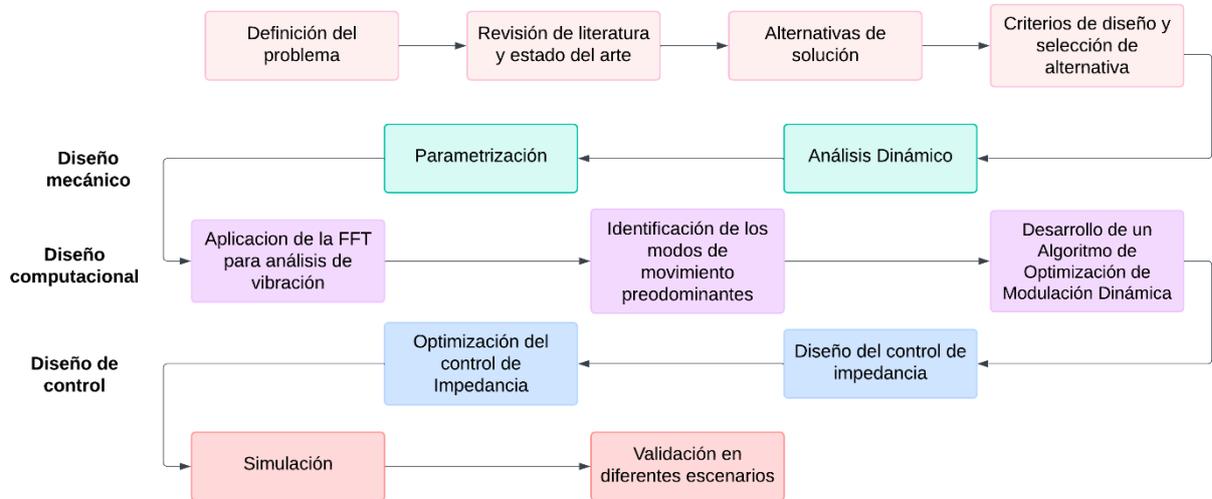
Alternativas	Criterios							Resultados	
	Adaptabilidad	Escalabilidad	Confiabilidad	Costo	Eficiencia	Mantenimiento	Sostenibilidad	Σ	Prioridad
1	0.075	0.15	0.06	0.14	0.09	0.07	0.04	0.62	3
								5	
2	0.105	0.20	0.07	0.16	0.12	0.08	0.03	0.77	2
								5	
3	0.135	0.225	0.08	0.18	0.135	0.09	0.04	0.88	1
								5	

La **Tabla 2.4** representa una matriz de decisión mediante la cual se puede concluir que la solución 3.

2.4 Proceso de diseño

2.4.1 Metodología del diseño

Una vez establecidos los requerimientos y seleccionar la alternativa más prometedora para la optimización de robots redundantes, es crucial seguir un proceso de diseño meticuloso y estructurado. Este proceso garantiza que cada decisión de diseño esté alineada con los objetivos planteados y que las soluciones desarrolladas sean efectivas y sostenibles. En esta sección, se presenta la metodología adoptada, que integra análisis dinámicos avanzados y técnicas de control de impedancia, para optimizar el rendimiento y la estabilidad de los robots. La **Figura 2.1** ilustra las diferentes etapas del proceso, que van desde la definición precisa del problema hasta la validación de las soluciones mediante simulaciones exhaustivas. Esta metodología asegura un enfoque integral, permitiendo que el diseño del sistema robótico no solo cumpla con los requerimientos técnicos, sino que también responda de manera eficiente a las demandas operativas más exigentes.

Figura 2.1*Metodología del diseño*

2.4.2 Selección de herramientas y Entorno de Desarrollo

Se evaluaron varias herramientas de programación y entornos de desarrollo. Inicialmente se utilizó código en Matlab y similares como Wólfram Mathematica, sin embargo, se optó por utilizar Python debido a varias ventajas clave. Python no solo ofrece una gran facilidad para estructurar los bloques y funciones de código, sino que también es de código abierto, lo que permite una mayor flexibilidad y personalización en el desarrollo. El acceso a una vasta comunidad de usuarios y desarrolladores proporciona abundante documentación, soporte y recursos adicionales. Además, su naturaleza de código abierto significa que no hay costos de licencia, lo que lo convierte en una opción económica y accesible para proyectos de cualquier escala. Estas características permiten a los investigadores y desarrolladores adaptar las herramientas a sus necesidades específicas, integrar nuevas tecnologías y colaborar más fácilmente con otros. Python fue utilizado para cada una de las etapas del algoritmo, que inicialmente se probó en robots simples como el 3R planar para el análisis de la dinámica a través del algoritmo mencionado. Además, se seleccionaron bibliotecas específicas como NumPy, SymPy y Matplotlib para facilitar el análisis y la visualización de los datos recolectados, incluyendo gráficos y otras visualizaciones.

2.4.3 Desarrollo del algoritmo

El análisis de las características dinámicas de ω y ζ en el espacio modal permite la síntesis para modular el amortiguamiento del sistema de manera analítica. La metodología que se expondrá proporciona una base sólida para la optimización y control de robots redundantes, mejorando su desempeño en aplicaciones prácticas. El algoritmo diseñado se basa en dicha metodología [19]. Donde se empieza con la premisa de la dinámica de un robot:

Partiendo de la ecuación (1.1), tenemos el término T_m que se refiere al torque que afecta el motor. Este término se puede reemplazar de la siguiente forma, conservando ciertas propiedades y matrices esenciales:

$$K[(q)q(t)] - v[(q)\dot{q}(t)] - h[q(t), \dot{q}(t)] - Cq(t) \quad (2.1)$$

Remplazando en la ecuación (1.1) nos quedaría de la siguiente forma:

$$M\ddot{q}(t) + C[\dot{q}(t)] + Kq(t) = \tau_{ext} \quad (2.2)$$

Esta ecuación es similar a la ecuación (1.1) describiendo la masa, amortiguación y rigidez de un sistema robótico. Pero debemos recordar que todas estas matrices están definidas en el espacio cartesiano, para los valores de la matriz K y C se pueden elegir según sea necesaria la aplicación. El control de impedancia en el espacio articular es de interés primario ya que es ahí donde se maneja el control físico de los motores y codificadores.

Así es necesario calcular las respectivas matrices en el espacio articular donde la matriz de inercia se puede calcular por diferentes métodos, y las matrices de rigidez y amortiguamiento se usa el jacobiano del robot que se está analizando de esta manera

$$C = J^T C_c J \quad (2.3)$$

Para seguir con el algoritmo, recordemos la descomposición modal del capítulo 1, donde se estudia el movimiento de un sistema según sus modos de vibración. En este contexto, $q_{RB}(t)$ y $q_{NRB}(t)$ representan las componentes de cuerpo rígido y no rígido, respectivamente. Para referirse

a sistemas robóticos la nomenclatura varia llamándolos cambio de energía potencial por lo que en la **Tabla 2.5** se hace la analogía entre un sistema mecánico discreto y dicho sistema robótico.

Tabla 2.5

Comparación entre sistemas mecánicos discretos y robóticos

Vibración mecánica	Control de impedancia robótica
Modo de cuerpo rígido	Cambio de energía potencial nula
RB	ZP
$\frac{1}{2} q_{RB}^T K q_{RB} = 0$	$\frac{1}{2} q_{ZP}^T K q_{ZP} = 0$
Modo no cuerpo rígido	Cambio de energía potencial no nula
NRB	NZP

En este caso, se cambian los términos. Sin embargo, como dice la teoría de vibraciones, se puede separar en dos cuerpos en el caso del sistema mecánico: RB (cuerpo rígido) y en el sistema robótico como Zero Potencial (ZP). Así mismo, NRB se cambia a NZP (Zero Potencial no nula)

Tabla 2.5.

Donde q_{RB} (cuerpo rígido) o Zero Potencial (ZP) representa los movimientos globales del robot, como traslaciones y rotaciones, que no implican deformaciones internas de rigidez.

Por otro lado, q_{NRB} incluye los desplazamientos relativos del robot que corresponden a sus modos de vibración internos. Este término es relevante para robots que emplean materiales flexibles o blandos, así como actuadores que requieren modelaciones no rígidas para lograr una precisión adecuada.

Este método es versátil, ya que tiene un caso para manipuladores redundantes como no redundantes, en el caso de robots no redundantes si no se restringe la respuesta dinámica solo incluiría la parte de NZP.

En tal caso, para ZP motion, está representada como U_0 que es un vector del espacio vacío de la matriz de rigidez este vector puede ser encontrado obteniendo:

$$q_{zp}(t) = u_0\beta(t) \quad (2.4)$$

Donde β es una función del tiempo. Basada en la siguiente transformación para una excitación externa Q

$$\beta(t) = \mathcal{L}^{-1}\left\{\frac{u_0^T Q}{s^2 + (u_0^T C u_0)s}\right\} \quad (2.5)$$

Como se puede observar usamos la transformada inversa de Laplace. Además, esta se sustituye en la ecuación (2.2) obteniendo en el caso de vibración libre:

$$u_0^T M u_0 \ddot{\beta}(t) + u_0^T C u_0 \dot{\beta}(t) = u_0^T \tau_{ext} \quad (2.6)$$

Dado de las matrices K y C pasan por transformaciones similares a través de la matriz jacobiana, el segundo término de la ecuación (2.6) desaparece. Además, ya que el vector u_0 está normalizado, la ecuación se puede simplificar quedando:

$$\ddot{\beta}(t) = u_0^T \tau_{ext} \quad (2.7)$$

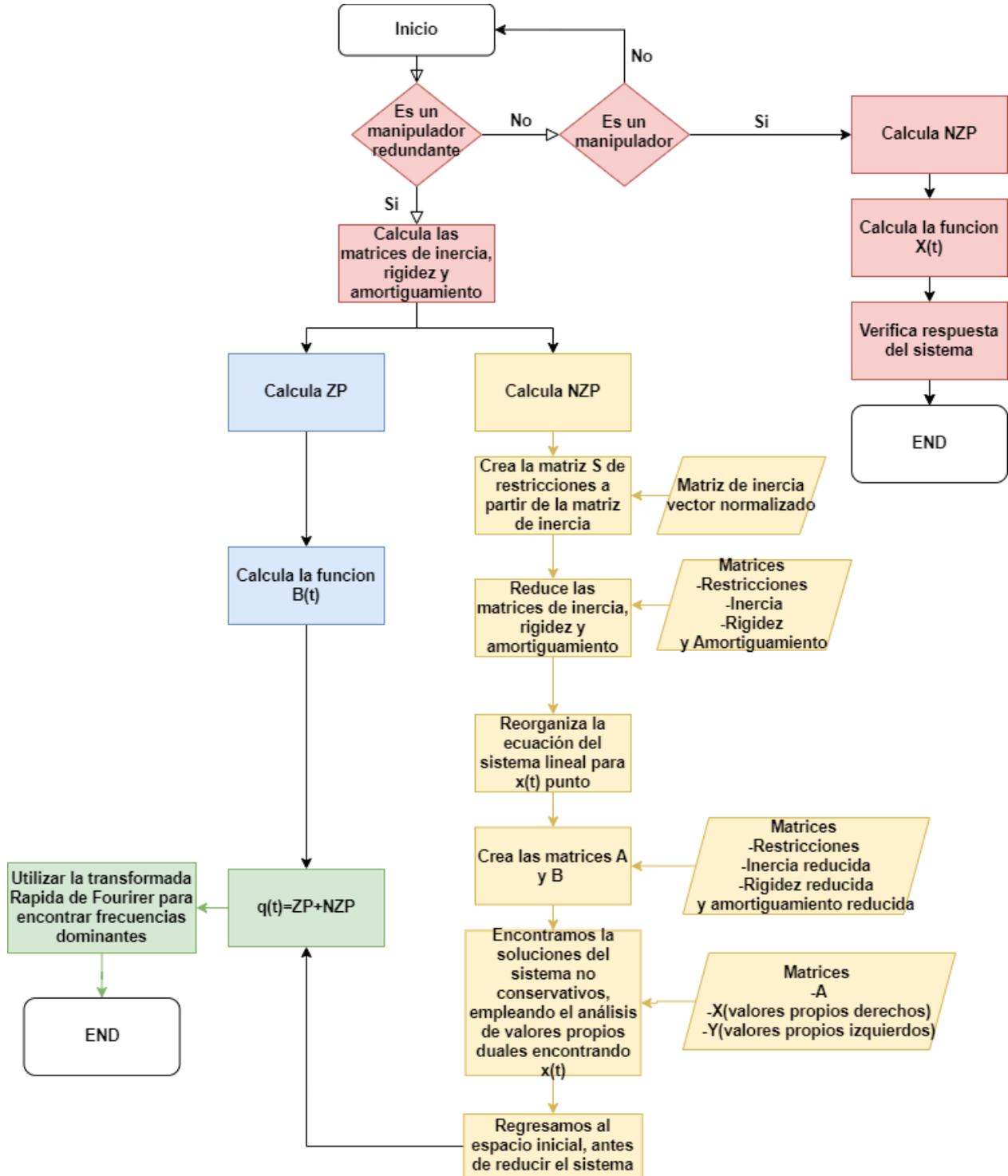
Para determinar los movimientos oscilatorios, eliminamos el modo ZP conocido, q_i , imponiendo sobre el vector de estado $q(t)$ una matriz de restricción S $n \times (n-1)$ para construir las matrices M' , C' y K' .

Una vez que el sistema NZP es positivo definido usando las matrices ya reducidas, se obtienen las respuestas NZP según se describe en la teoría para sistemas mecánicos discretos.

Finalmente, encontrando ambos términos es necesario combinarlos, ZP y NZP. En el diagrama de flujo de la **Figura 2.2** se describe la función del algoritmo final.

Figura 2.2

Algoritmo que lleva a cabo el código

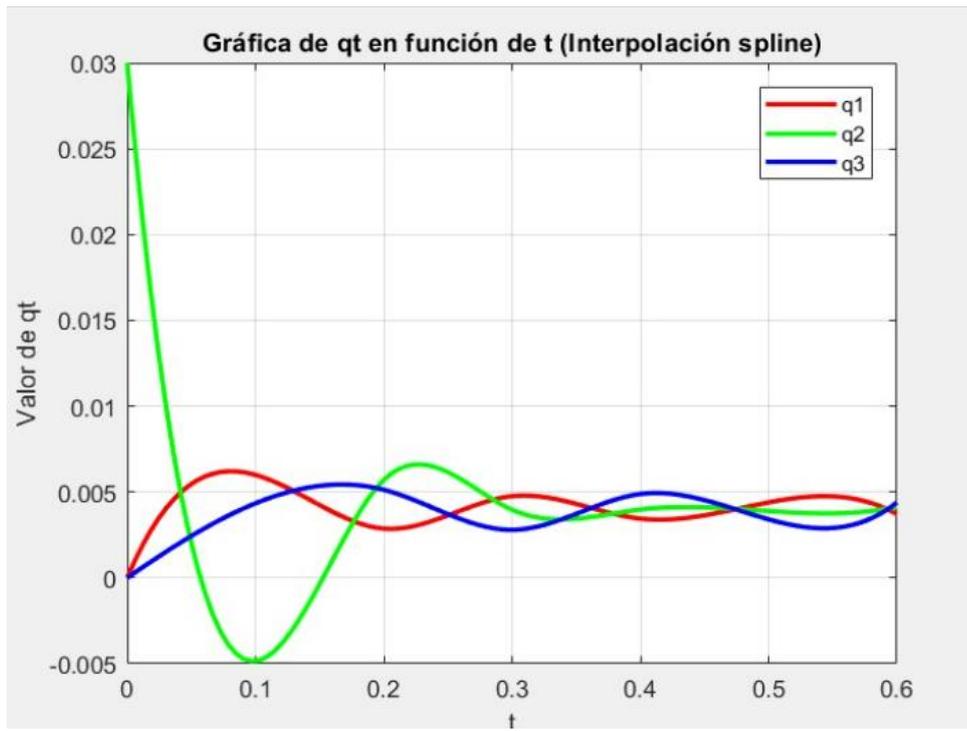


En este contexto, una vez obtenida la respuesta dinámica del modelo, a medida que aumentan los grados de libertad en el manipulador y coordenadas cartesianas por describir o controlar, también lo hacen sus frecuencias naturales. Por lo tanto, es esencial además de útil contar con un método que analice y determine cuáles de estas frecuencias afectan más al sistema o predominan. Dado que hay numerosas frecuencias, es importante evitar el análisis de aquellas que no son relevantes para el sistema, ya que esto puede consumir recursos, desgaste en los actuadores y tiempo innecesarios. Para abordar este desafío, se empleará la Transformada Rápida de Fourier (FFT), que permite identificar de manera eficiente las frecuencias más significativas en el comportamiento dinámico del sistema. La FFT es una herramienta fundamental en el análisis de señales, ya que descompone una señal en sus componentes de frecuencia, facilitando la identificación de los modos de vibración predominantes y su posterior optimización en el control del manipulador.

En la **Figura 2.3** se muestra la respuesta completa de un sistema mecánico, similar a los ejemplos mencionados en [19] donde las curvas (q_1 , q_2 , q_3) revelan modos de vibración con diferentes frecuencias y amortiguaciones, destacándose un desfase respecto al punto de inicio. Este desfase indica un desequilibrio en las condiciones iniciales y fuerzas externas. La superposición de modos rígidos (baja frecuencia) y no rígidos (alta frecuencia) genera una respuesta global compleja. La validación del algoritmo a través de esta gráfica sugiere la necesidad de optimizar parámetros dinámicos y de control para reducir vibraciones y mejorar la eficiencia y estabilidad el sistema, aplicando técnicas como el análisis FFT y simulaciones dinámicas para minimizar resonancias y prolongar la vida útil de los componentes.

Figura 2.3

Respuesta de qt en función de t



CAPÍTULO 3

3.1 Resultados y análisis

La presentación de los resultados obtenidos a partir de la simulación dinámica del robot 3R planar se realizó considerando diversas configuraciones de parámetros mecánicos y de control. Se evaluaron múltiples escenarios para analizar cómo las variaciones en estos parámetros afectan el comportamiento del robot. El análisis de los resultados se centra en tres aspectos clave: el cambio de energía potencial cero (QNZP), la respuesta del sistema en el dominio del tiempo ($Q(t)$), y un análisis de frecuencia mediante la Transformada Rápida de Fourier (FFT).

El objetivo de este capítulo es manipular las diferentes configuraciones de parámetros, tales como las masas de los eslabones, los valores de rigidez y amortiguamiento, y los ángulos articulares iniciales y como afectan el comportamiento dinámico del robot. Para ello, se han realizado simulaciones bajo cuatro composiciones diferentes de parámetros, cuyos resultados se presentan y analizan en detalle.

3.1.1 Primera configuración

Tabla 3.1

Parámetros primera configuración

Parámetro	Valor
Masas de los eslabones (kg)	Eslabón 1: 20 Eslabón 2: 10 Eslabón 3: 5
Longitudes de los eslabones (m)	Eslabón 1: 0.4 Eslabón 2: 0.25 Eslabón 3: 0.2
Posiciones del centro de masa (m)	Eslabón 1: 0.4 / 2 Eslabón 2: 0.25 / 2 Eslabón 3: 0.2 / 2
Ángulos articulares iniciales (rad)	θ_1 : 0.529 θ_2 : 0.6 θ_3 : 0.105
Parámetros de rigidez (N/m)	K_x : 400 K_y : 300
Parámetros de amortiguamiento (Ns/m)	C_x : 30 C_y : 10
Desplazamientos iniciales (m)	q_x : 0.15 q_y : 0.10
Velocidades iniciales (m/s)	V_x : 0 V_y : 0

Con los parámetros que se han establecido para la configuración en la **Tabla 3.1** se han calculado los datos obtenidos en la **Tabla 3.2**. Se ha seguido el mismo procedimiento para calcular los resultados en todas las configuraciones.

Para inducir la vibración libre en el robot 3R planar, primero se determinó una condición inicial desplazando el efector final ciertos centímetros en las direcciones x y y desde su posición

inicial. Para calcular los ángulos articulares que permitieron al efector final alcanzar esta nueva posición específica en el plano, se utilizó un proceso basado en la cinemática inversa. Donde los ángulos articulares resultantes están representados en la primera fila de la **Tabla 3.2**.

Estos ángulos articulares resultantes han sido fundamentales para dirigir el movimiento del robot, permitiendo que el efector final alcanzara con precisión la posición objetivo. Posteriormente, el robot fue liberado desde esta condición para analizar su comportamiento en vibración libre. El mismo procedimiento se aplicó para obtener los ángulos articulares en las cuatro configuraciones consideradas.

El vector QZP ha representado los desplazamientos asociados con el modo de cuerpo rígido del robot 3R planar en un momento específico de la simulación. Este modo de cuerpo rígido ha sido una descripción de los desplazamientos angulares de las articulaciones del robot cuando se ha asumido que todas sus articulaciones y eslabones se han movido en conjunto, sin que haya habido deformaciones internas o flexiones; es decir, el robot se ha comportado como un cuerpo sólido.

Para obtener este vector, se ha comenzado determinando las propiedades dinámicas fundamentales del sistema, como las matrices de inercia y rigidez. Con estas matrices establecidas, se ha procedido a calcular las velocidades angulares. El resultado ha sido un vector que indica los desplazamientos angulares en cada una de las articulaciones del robot. Estas velocidades angulares se presentan en la segunda fila de la **Tabla 3.2**.

El cálculo de los valores propios se ha realizado bajo el modo de vibración libre, lo que ha permitido analizar cómo el sistema oscila naturalmente cuando no está sujeto a fuerzas externas. Este enfoque ha sido clave para identificar las frecuencias naturales en las que el sistema puede vibrar cuando se deja libre para moverse sin restricciones.

El proceso ha comenzado con la utilización de las matrices dinámicas clave del sistema, que incluyen la matriz de inercia, la matriz de rigidez, y la matriz de amortiguamiento.

Para obtener los valores propios, se ha construido la matriz dinámica global del sistema en espacio de estados utilizando las matrices de inercia, rigidez y amortiguamiento. Luego, se ha formulado y resuelto la ecuación característica asociada a esta matriz dinámica. Este proceso ha involucrado encontrar las raíces del polinomio característico derivado de la matriz, las cuales corresponden a los valores propios del sistema.

Los valores propios obtenidos han proporcionado las frecuencias naturales del robot, reflejando las tasas a las que el sistema tendería a vibrar si se le permite oscilar libremente sin ninguna intervención externa.

Tabla 3.2

Resultados primera configuración

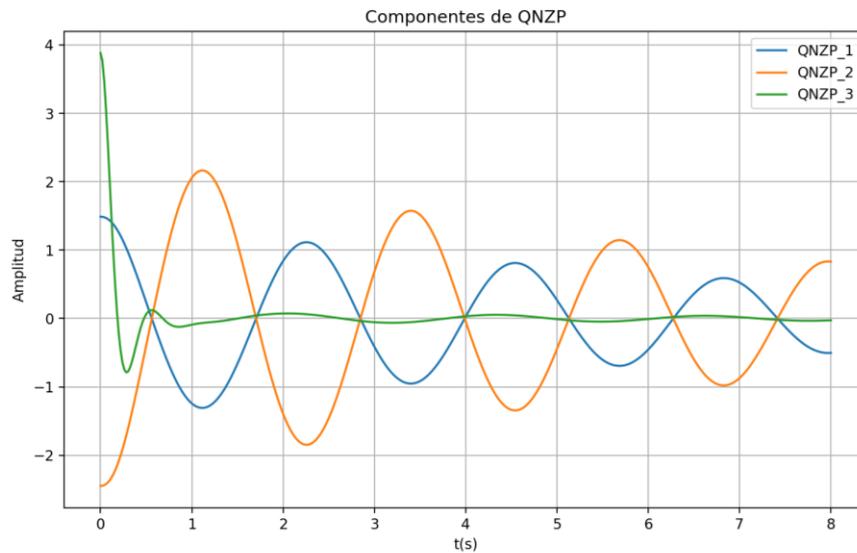
Parámetro	Valor
Ángulos articulares iniciales (rad)	$\theta_1: 1.3322$ $\theta_2: -0.7416$ $\theta_3: 0.6434$
Vector QZP (rad)	$\begin{bmatrix} -0.1567 \\ 1.706 \\ -3.238 \end{bmatrix}$
Frecuencias (rad/s)	[12.3435, 2.7540, 2.7540, 12.3435]
Frecuencias (Hz)	[1.9639, 0.4383, 0.4383, 1.9639]

En la **Figura 3.1** se han mostrado los componentes de QNZP, donde, después de haber sido reducidos por la matriz de restricciones para evitar singularidades y trabajar de forma más efectiva, se ha vuelto al espacio normal. Esta gráfica ha ilustrado la evolución temporal de las tres componentes de *QNZP*, representando cómo las articulaciones del robot 3R planar han respondido bajo las condiciones simuladas.

En el eje horizontal se ha representado el tiempo en segundos (t), mientras que en el eje vertical se ha mostrado la amplitud de las oscilaciones, expresada en radianes. Las curvas de color azul, naranja y verde han correspondido a las diferentes componentes de *QNZP*, que han estado asociadas con las articulaciones del robot.

Figura 3.1

Gráfica de QNZP primera configuración



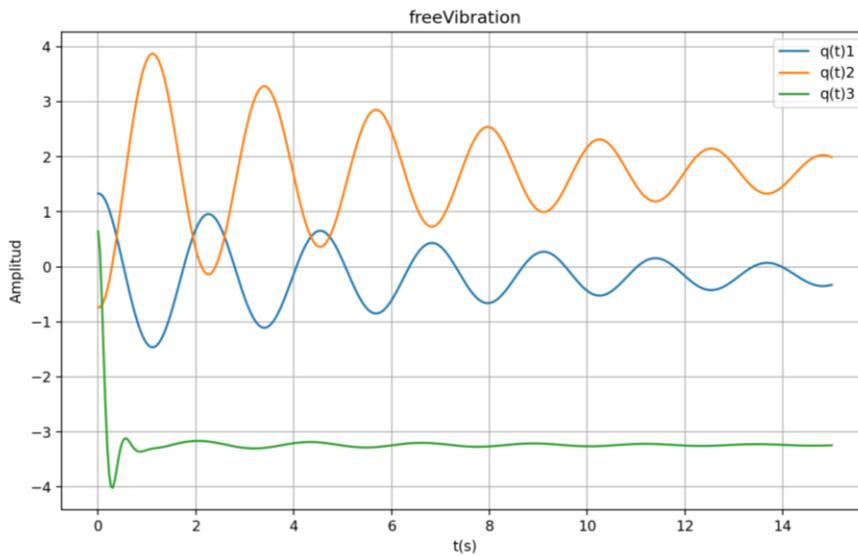
En la **Figura 3.2** se han mostrado las componentes de $Q(t)$ en el espacio normal, representando la respuesta temporal de los desplazamientos angulares de las articulaciones del robot 3R planar en un escenario de vibración libre. Esta gráfica ha ilustrado cómo los desplazamientos de las tres articulaciones han evolucionado a lo largo del tiempo bajo condiciones dinámicas sin restricciones.

El proceso de cálculo de $Q(t)$ ha comenzado con la integración de las ecuaciones de movimiento del sistema, utilizando las matrices de inercia y rigidez para determinar cómo el sistema responde a una perturbación inicial.

En el eje horizontal se ha representado el tiempo en segundos (t), mientras que el eje vertical ha mostrado los desplazamientos en radianes (rad). Las curvas azul, naranja y verde han correspondido a las diferentes componentes de $Q(t)$, asociadas con las tres articulaciones del robot.

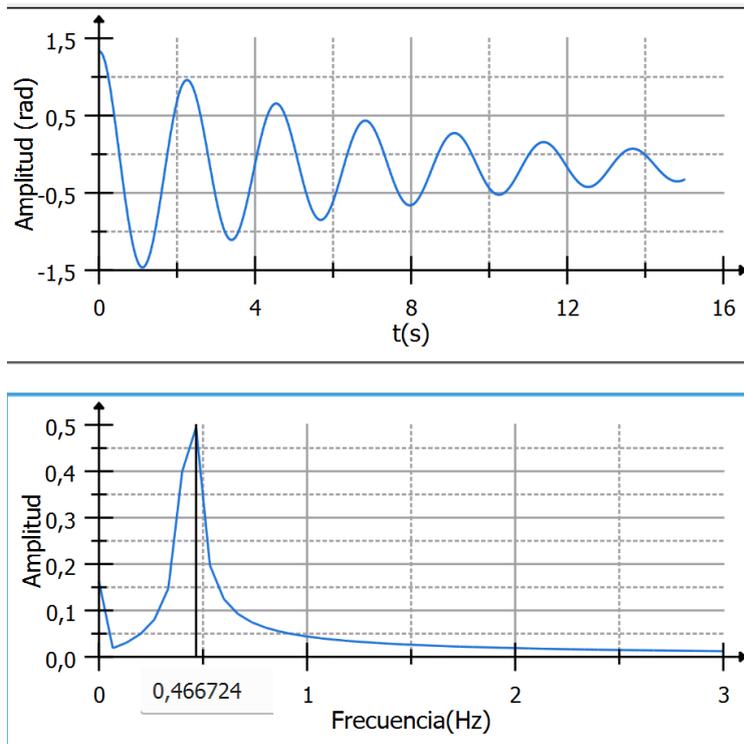
Figura 3.2

Gráfica de $Q(t)$ primera configuración

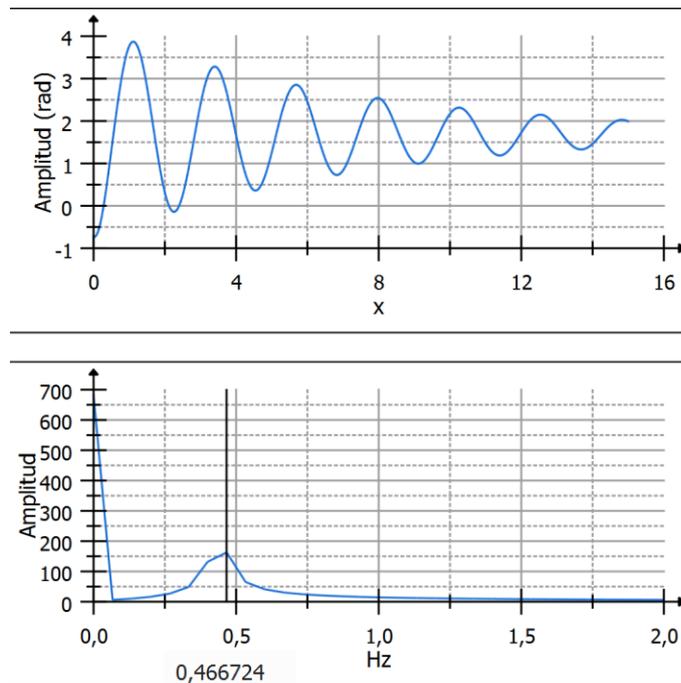


En la **Figura 3.3**, **Figura 3.4** y **Figura 3.5** se presentan las gráficas correspondientes a las FFT aplicadas a la respuesta de los eslabones $q(t)_1$, $q(t)_2$ y $q(t)_3$ respectivamente, que han sido obtenidas en la primera configuración del sistema. Estas gráficas proporcionan una descomposición de las señales de tiempo en sus componentes frecuenciales.

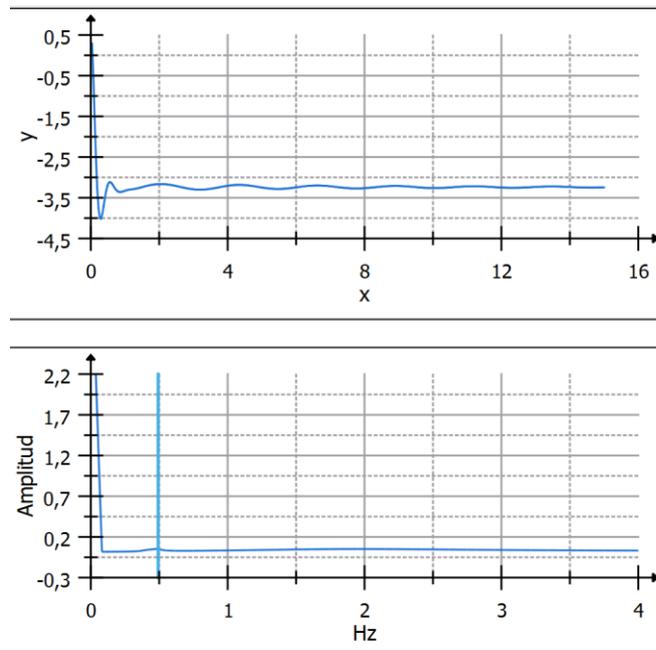
En la **Figura 3.3** se ha mostrado la FFT de la primera componente $q(t)_1$, que corresponde al desplazamiento angular de la primera articulación del robot. La gráfica superior muestra la evolución de $q(t)_1$ en el dominio del tiempo, donde se ha podido observar un comportamiento oscilatorio con una tendencia a estabilizarse alrededor de 0 rad/s. La gráfica inferior, que representa la FFT, ha revelado que la mayor amplitud está concentrada en una frecuencia muy baja, cercana a 0.4 Hz. Esta baja frecuencia ha indicado que la primera articulación tiene un modo de vibración predominante a bajas frecuencias. Esta frecuencia relativamente baja ha sugerido que la primera articulación se mueve de manera más lenta en comparación con las otras, lo que es consistente con el análisis de las propiedades dinámicas del sistema en esta configuración específica.

Figura 3.3*FFT del primer eslabón de la primera configuración*

La **Figura 3.4** representa la FFT de la segunda componente $q(t)_2$, correspondiente al desplazamiento angular de la segunda articulación del robot. Aquí, se puede observar en la gráfica del tiempo que $q(t)_2$ presenta oscilaciones más pronunciadas que $q(t)_1$, o que ha sugerido una mayor energía dinámica en esta articulación. La gráfica de la FFT ha mostrado un pico significativo alrededor de 0.466 Hz, indicando que la segunda articulación tiene un modo de vibración predominante en esa frecuencia. Este resultado ha sido clave, ya que ha indicado que la segunda articulación podría ser más susceptible a resonancias en esa frecuencia específica, lo que podría afectar la estabilidad del sistema si no se gestiona adecuadamente.

Figura 3.4*FFT del segundo eslabón de la primera configuración*

Finalmente, en **Figura 3.5** se ha presentado la FFT de la tercera componente $q(t)_3$, que se refiere al desplazamiento angular de la tercera articulación del robot. En la gráfica del tiempo, se puede observar que $q(t)_3$ muestra oscilaciones de menor amplitud, lo que indica un comportamiento más amortiguado o con menos energía. Sin embargo, la FFT ha revelado un pico en una frecuencia alrededor de 0.466 Hz, lo que ha sugerido que, aunque las oscilaciones sean menos pronunciadas en el dominio del tiempo, la tercera articulación tiene un modo de vibración significativo a esta frecuencia. Esta información ha sido crucial para el diseño del sistema de control, ya que cualquier perturbación cercana a esta frecuencia podría excitar este modo de vibración, afectando la precisión y estabilidad del movimiento del robot.

Figura 3.5*FFT del tercer eslabón de la primera configuración*

La importancia de estas FFT para la configuración específica radica en su capacidad para identificar las frecuencias a las que cada articulación es más susceptible. Las frecuencias identificadas coinciden con los valores propios obtenidos en el análisis de vibración libre, confirmando la consistencia entre los resultados temporales y frecuenciales. Esta relación directa entre las frecuencias naturales y las respuestas dinámicas del sistema en el dominio del tiempo resalta la importancia de considerar estos modos de vibración al optimizar los parámetros de control y las configuraciones del robot. En particular, esta configuración destaca la necesidad de prestar atención especial a la segunda y tercera articulaciones, que muestran una tendencia a oscilar en frecuencias que podrían estar dentro de rangos críticos de resonancia.

3.1.2 Segunda configuración

Tabla 3.3

Parámetros segunda configuración

Parámetro	Valor
Masas de los eslabones (kg)	Eslabón 1: 20 Eslabón 2: 10 Eslabón 3: 5
Longitudes de los eslabones (m)	Eslabón 1: 0.4 Eslabón 2: 0.25 Eslabón 3: 0.2
Posiciones del centro de masa (m)	Eslabón 1: 0.4 / 2 Eslabón 2: 0.25 / 2 Eslabón 3: 0.2 / 2
Ángulos articulares iniciales (rad)	$\theta_1: 0.529$ $\theta_2: 0.6$ $\theta_3: 0.105$
Parámetros de rigidez (N/m)	$K_x: 200$ $K_y: 300$
Parámetros de amortiguamiento (Ns/m)	$C_x: 40$ $C_y: 30$
Desplazamientos iniciales (m)	$q_x: 0.05$ $q_y: 0.00$
Velocidades iniciales (m/s)	$V_x: 0$ $V_y: 0$

Tabla 3.4

Resultados segunda configuración

Parámetro	Valor
Ángulos articulares (rad)	θ_1 : 1.1598
	θ_2 : -0.9126
	θ_3 : 0.9868
Vector QZP (rad/s)	$\begin{bmatrix} -0.1076 \\ 1.1718 \\ -2.2234 \end{bmatrix}$
Frecuencias (rad/s)	[9.0559, 2.6543, 2.6543, 9.0559]
Frecuencias (Hz)	[1.4403, 0.4224, 0.4224, 1.4403]

Figura 3.6

Gráfica de QNZP segunda configuración

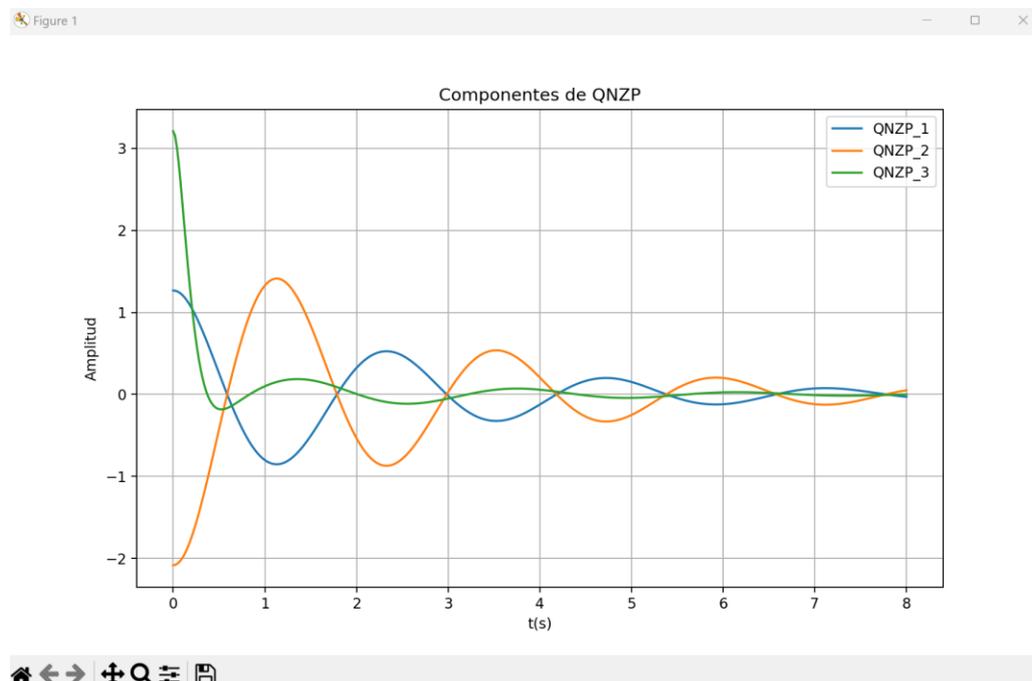
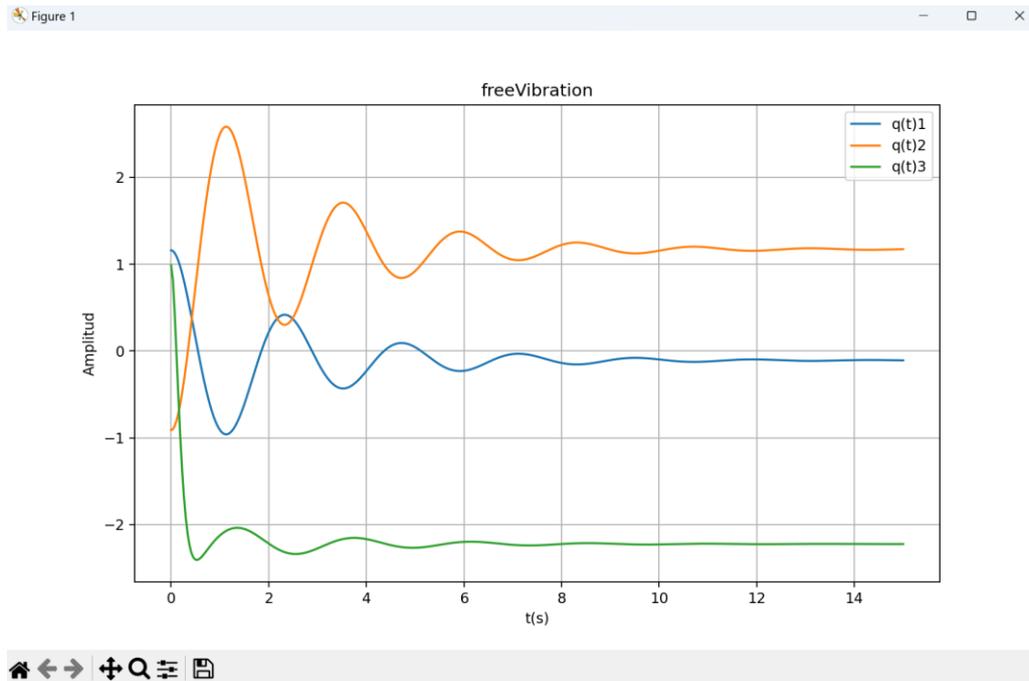


Figura 3.7Gráfica de $Q(t)$ segunda configuración

En la segunda configuración, los análisis de las FFT, representados en la **Figura 3.8**, **Figura 3.9** y **Figura 3.10**, han identificado modos de vibración predominantes en frecuencias cercanas a 0.403 Hz para $q(t)_1$, 0.406 Hz para $q(t)_2$ y 1.431 Hz para $q(t)_3$. Estos resultados muestran que las frecuencias naturales en esta configuración están ligeramente más elevadas en comparación con las configuraciones anteriores, lo cual sugiere que las articulaciones están oscilando en modos más rápidos. Este comportamiento destaca la importancia de ajustar los parámetros del sistema de control, especialmente para la tercera articulación, para mitigar el riesgo de resonancias y asegurar un funcionamiento estable del robot bajo estas condiciones operativas.

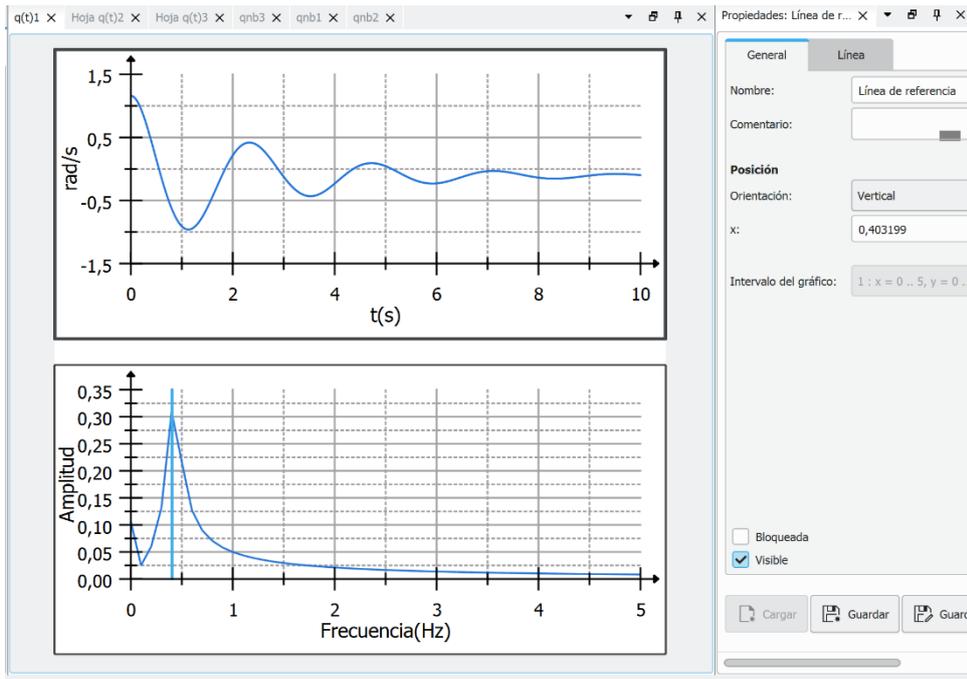
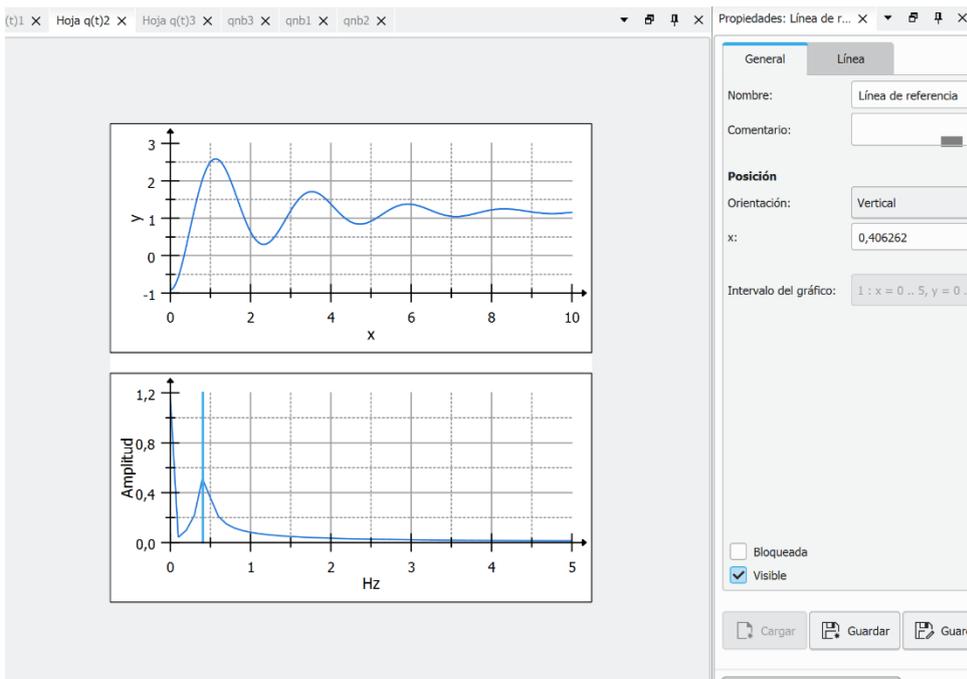
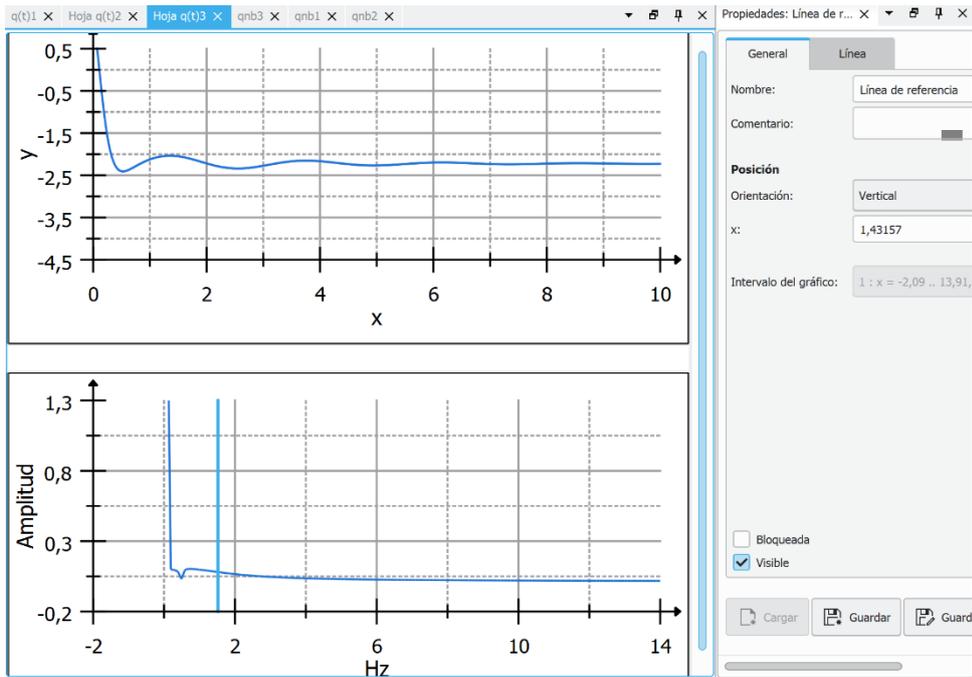
Figura 3.8*FFT del primer eslabón de la segunda configuración***Figura 3.9***FFT del segundo eslabón de la segunda configuración*

Figura 3.10*FFT del tercer eslabón de la segunda configuración*

3.1.3 Tercera configuración

Tabla 3.5

Parámetros tercera configuración

Parámetro	Valor
Masas de los eslabones (kg)	Eslabón 1: 10 Eslabón 2: 5 Eslabón 3: 2
Longitudes de los eslabones (m)	Eslabón 1: 0.4 Eslabón 2: 0.25 Eslabón 3: 0.2
Posiciones del centro de masa (m)	Eslabón 1: 0.4 / 2 Eslabón 2: 0.25 / 2 Eslabón 3: 0.2 / 2
Ángulos articulares iniciales (rad)	$\theta_1: 0.529$ $\theta_2: 0.6$ $\theta_3: 0.105$
Parámetros de rigidez (N/m)	$K_x: 200$ $K_y: 300$
Parámetros de amortiguamiento (Ns/m)	$C_x: 40$ $C_y: 30$
Desplazamientos iniciales (m)	$q_x: 0.00$ $q_y: 0.01$
Velocidades iniciales (m/s)	$V_x: 0$ $V_y: 0$

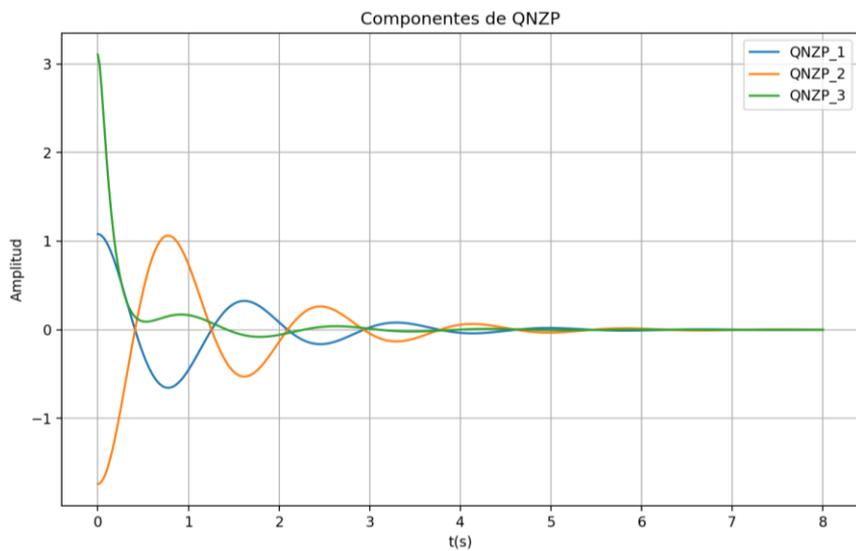
Tabla 3.6

Resultados tercera configuración

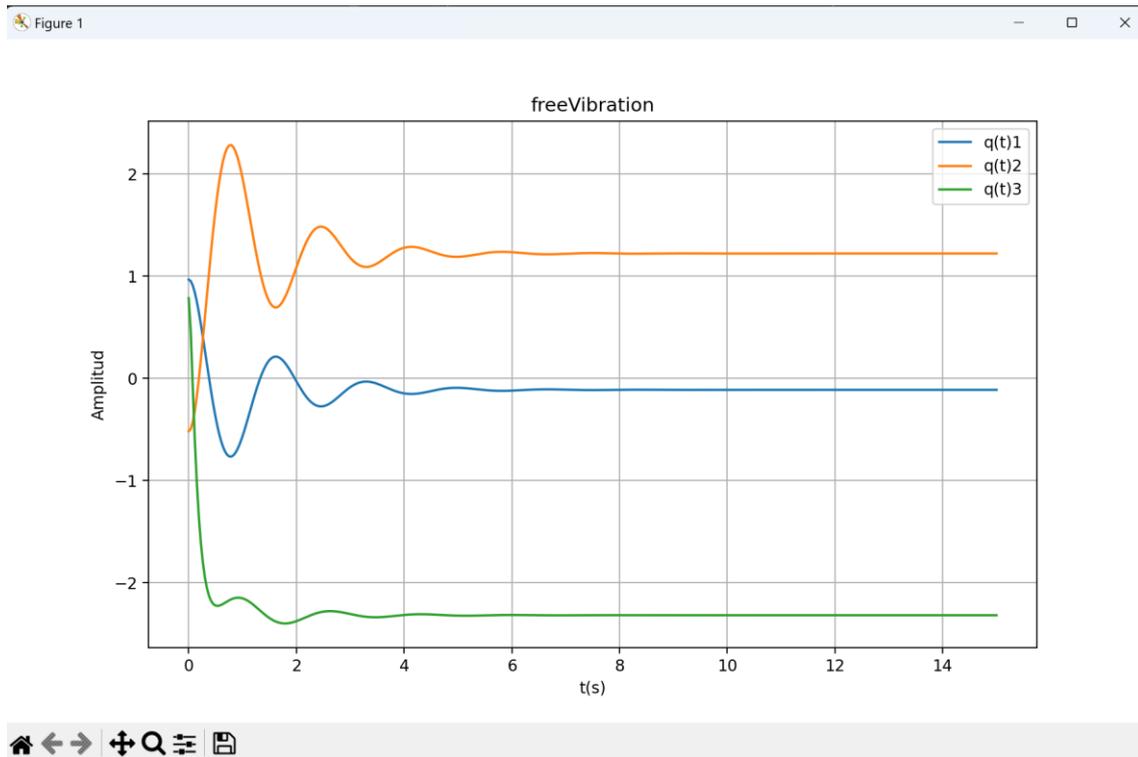
Parámetro	Valor
Ángulos articulares (rad)	$\theta_1: 0.9674$ $\theta_2: -0.5184$ $\theta_3: 0.7850$
Vector QRB (rad/s)	$\begin{bmatrix} -0.1122 \\ 1.2222 \\ -2.319 \end{bmatrix}$
Frecuencias (rad/s)	[3.8337, 3.8337, 6.4337, 31.6006]
Frecuencias (Hz)	[0.6101, 0.6101, 1.0239, 5.0294]

Figura 3.11

Gráfica de QNBZ tercera configuración



(x, y) = (3.696, -0.457)

Figura 3.12Gráfica de $Q(t)$ tercera configuración

En la tercera configuración, los análisis de las FFT representados en la **Figura 3.13**, **Figura 3.14** y **Figura 3.15** han revelado que las frecuencias predominantes para las tres articulaciones son 0.609 Hz, 0.614 Hz, y 1.043 Hz para $q(t)_1$, $q(t)_2$ y $q(t)_3$ respectivamente. Estas frecuencias reflejan un cambio en los modos de vibración naturales del sistema, lo cual podría implicar una mayor sensibilidad a las perturbaciones externas en estas frecuencias específicas. Este resultado subraya la necesidad de una optimización cuidadosa del control, enfocada en minimizar el impacto de estas frecuencias sobre la estabilidad y el rendimiento del robot en esta configuración particular.

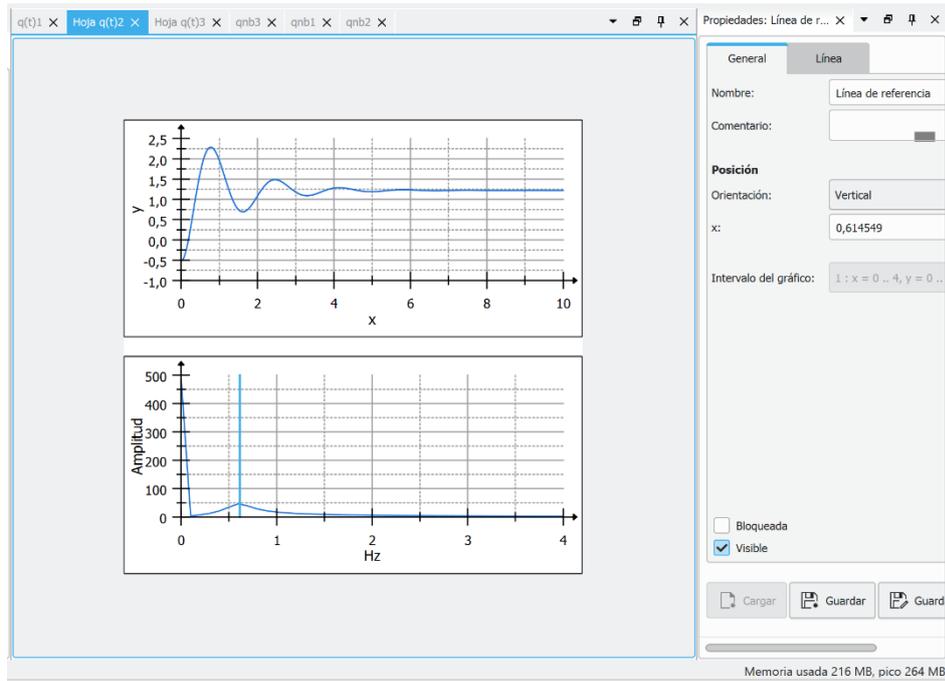
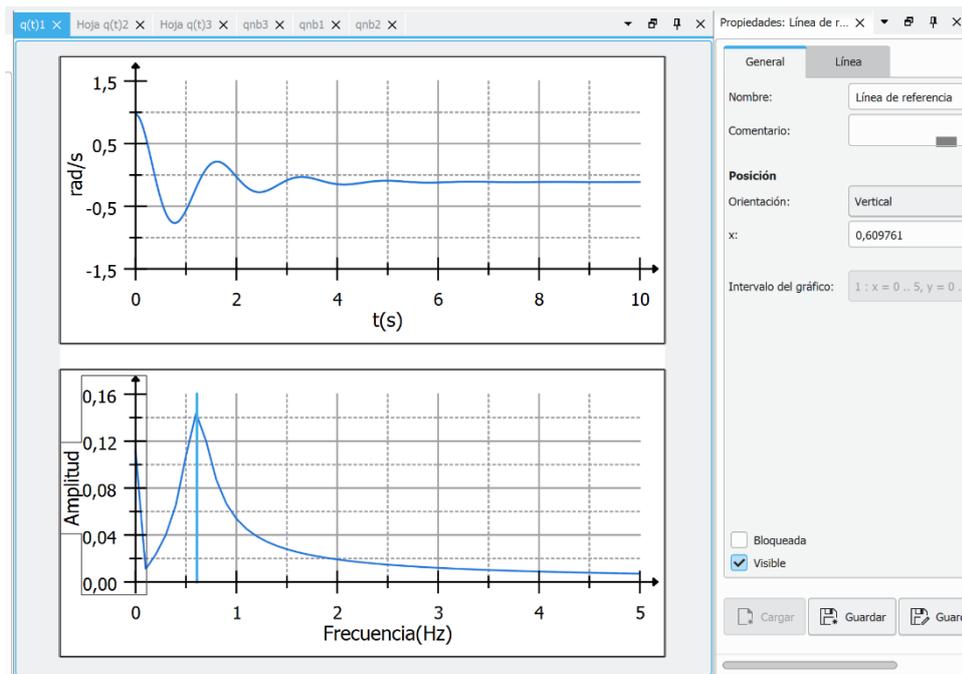
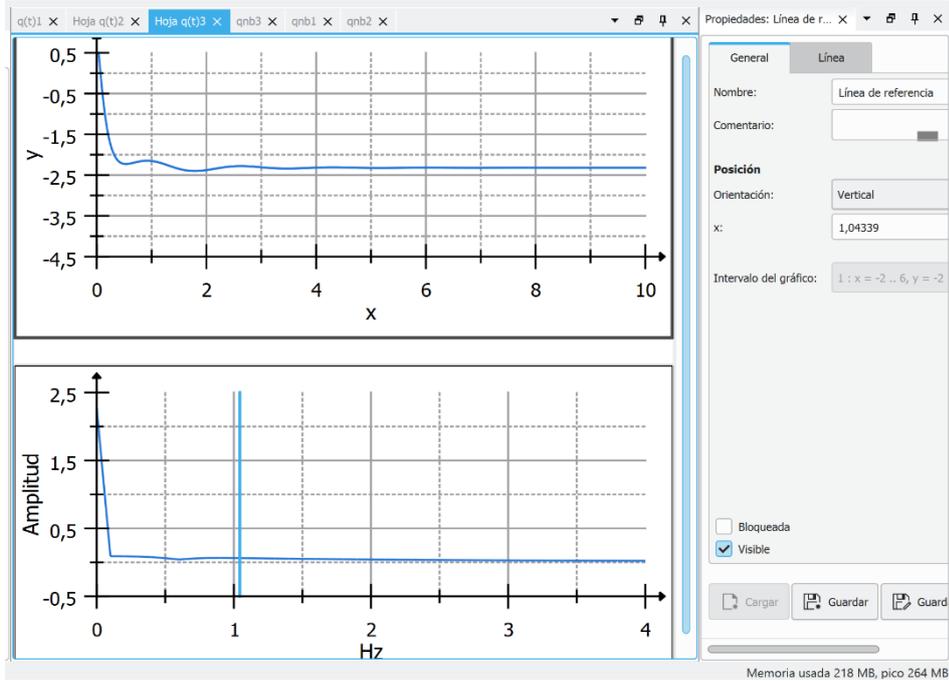
Figura 3.13*FFT del primer eslabón de la tercera configuración***Figura 3.14***FFT del segundo eslabón de la tercera configuración*

Figura 3.15*FFT del tercer eslabón de la tercera configuración*

3.1.4 Cuarta Configuración

Tabla 3.7

Parámetros cuarta configuración

Parámetro	Valor
Masas de los eslabones (kg)	Eslabón 1: 10 Eslabón 2: 5 Eslabón 3: 2
Longitudes de los eslabones (m)	Eslabón 1: 0.4 Eslabón 2: 0.25 Eslabón 3: 0.2
Posiciones del centro de masa (m)	Eslabón 1: 0.4 / 2 Eslabón 2: 0.25 / 2 Eslabón 3: 0.2 / 2
Ángulos articulares iniciales (rad)	θ_1 : 0.529 θ_2 : 0.6 θ_3 : 0.105
Parámetros de rigidez (N/m)	K_x : 300 K_y : 300
Parámetros de amortiguamiento (Ns/m)	C_x : 20 C_y : 15
Desplazamientos iniciales (m)	q_x : 0.012 q_y : 0.00
Velocidades iniciales (m/s)	V_x : 0 V_y : 0

Tabla 3.8

Resultados cuarta configuración

Parámetro	Valor
Ángulos articulares (rad)	$\theta_1: 1.0331$
	$\theta_2: -0.6891$
	$\theta_3: 0.89$
Vector QRB (rad/s)	$\begin{bmatrix} -0.1053 \\ 1.1472 \\ -2.1766 \end{bmatrix}$
Frecuencias (rad/s)	[17.1399, 3.9060, 3.9060, 17.1399]
Frecuencias (Hz)	[2.7279, 0.6216, 0.6216, 2.779]

Figura 3.16

Gráfica de QNBZ cuarta configuración

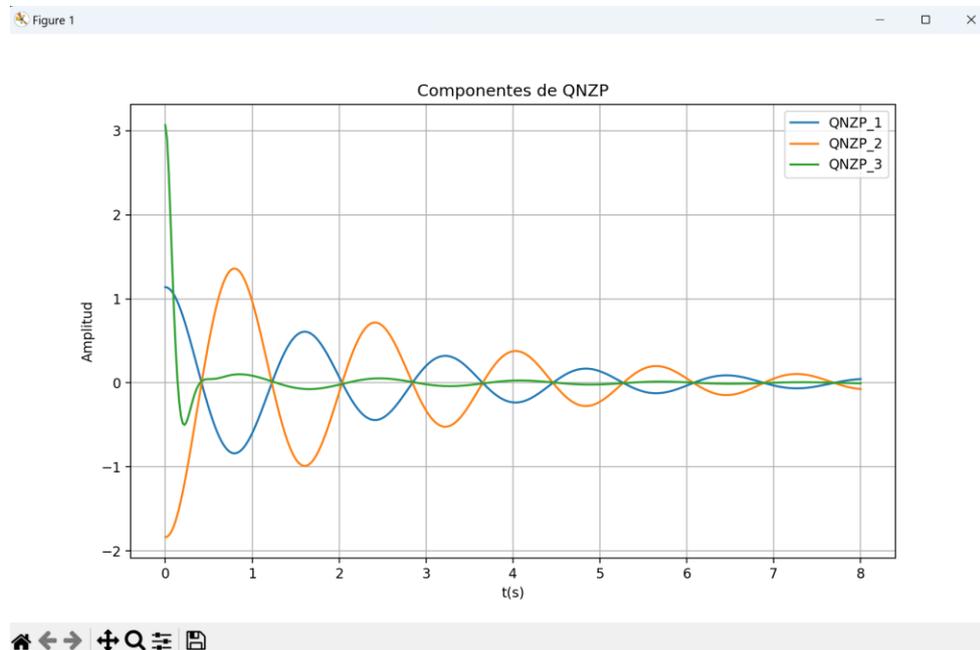
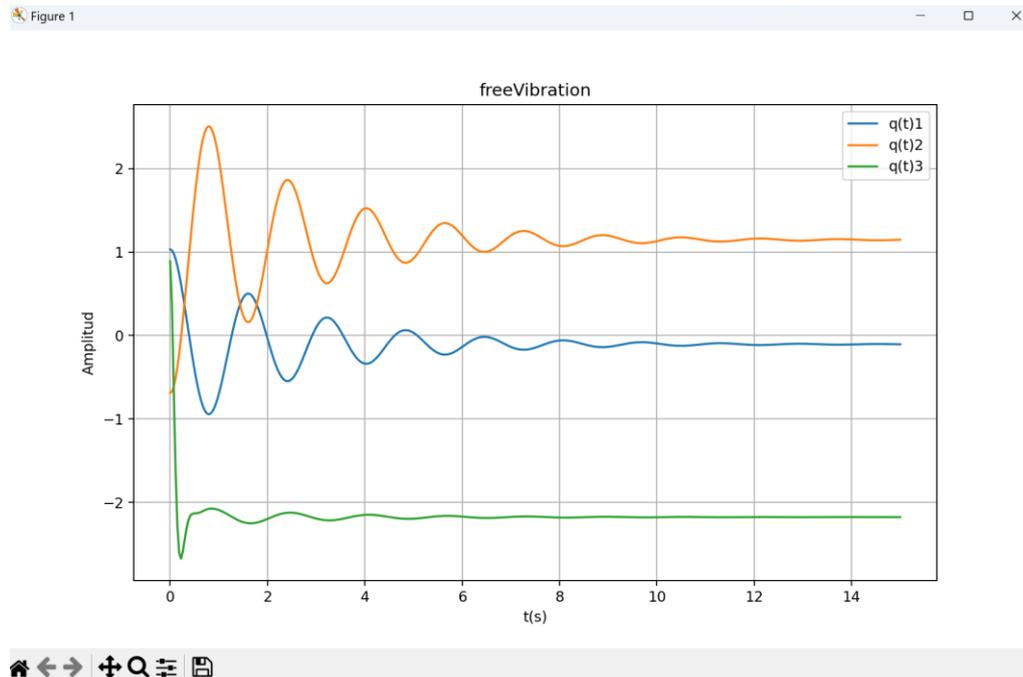


Figura 3.17

Gráfica de $Q(t)$ cuarta configuración



En la cuarta configuración, los resultados obtenidos mediante las FFT representados en la **Figura 3.18**, **Figura 3.19** y **Figura 3.20** han mostrado una distribución de frecuencias naturales en torno a 0.609 Hz $q(t)_1$, 0.614 Hz para $q(t)_2$, y 1.1 Hz para $q(t)_3$, coherentes con los valores obtenidos en configuraciones anteriores. Estas frecuencias indican que el sistema podría estar operando en un rango donde las resonancias pueden volverse más críticas, especialmente en la tercera articulación. Este análisis destaca la necesidad de un control preciso que minimice la posibilidad de que estas frecuencias afecten la estabilidad del robot, garantizando un desempeño confiable en esta configuración.

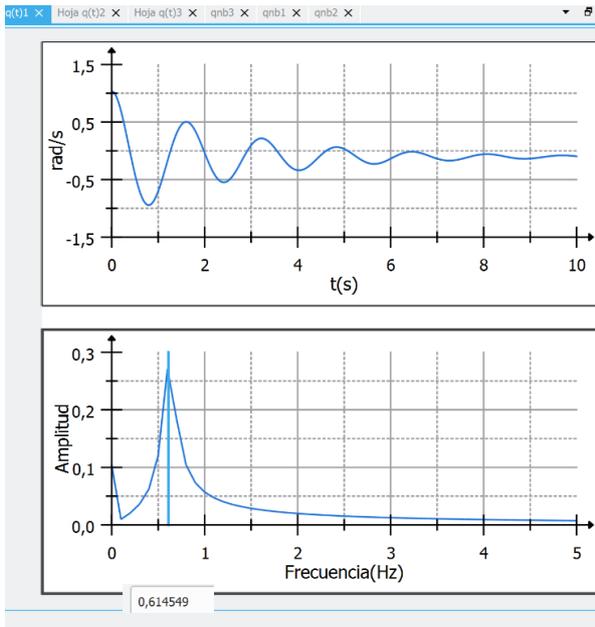
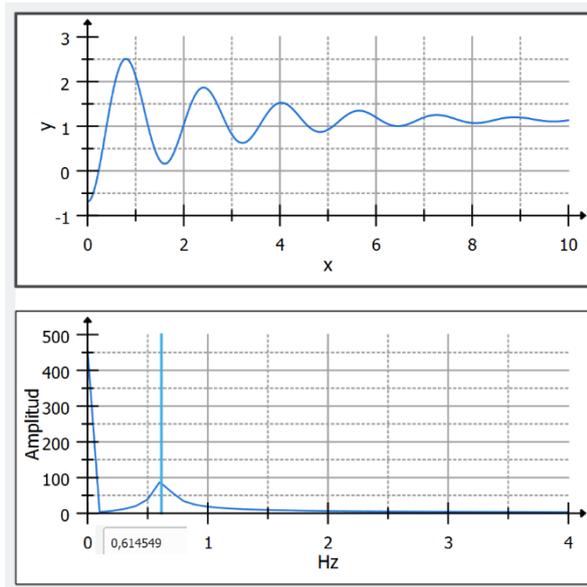
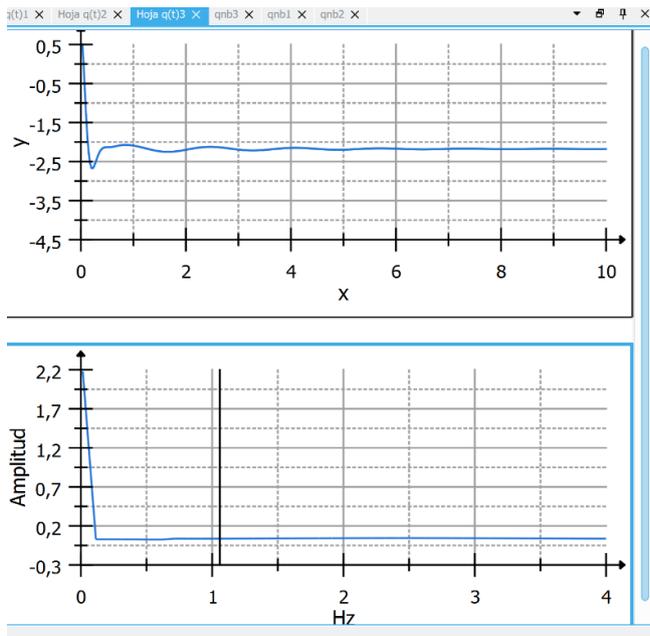
Figura 3.18*FFT del primer eslabón de la cuarta configuración***Figura 3.19***FFT del segundo eslabón de la cuarta configuración*

Figura 3.20*FFT del tercer eslabón de la cuarta configuración*

3.1.5 Análisis Comparativo de las Configuraciones

En la comparación de las cuatro configuraciones analizadas, se ha observado que las configuraciones con mayores frecuencias naturales (tercera y cuarta) han ofrecido una respuesta más ágil y rápida del sistema. Estas configuraciones han sido ideales para aplicaciones que requieren movimientos veloces y precisos.

Por otro lado, las configuraciones con frecuencias naturales más bajas han mostrado una mayor estabilidad y menor susceptibilidad a resonancias, por lo que podrían ser más adecuadas para tareas donde la precisión y la estabilidad han sido prioritarias sobre la velocidad de respuesta. Estas configuraciones deberían ser más seguras en aplicaciones que no demandan tanta rapidez, pero a su vez requerirían un control fiable y constante.

A medida que ajustábamos los parámetros de amortiguamiento y la geometría del robot en cada configuración, observamos cambios significativos en la respuesta del sistema, mejorando o empeorando su comportamiento dinámico. Estos ajustes permitieron identificar cómo diferentes combinaciones de parámetros afectan la estabilidad y eficiencia del robot durante su operación. Dependiendo de la tarea específica que deba realizar el robot, como el manejo de cargas,

movimientos precisos o rápidas aceleraciones, se puede modificar dinámicamente estos parámetros en el código para obtener una respuesta vibratoria óptima. Esto nos permite reducir las vibraciones no deseadas y maximizar el rendimiento del robot, adaptándolo a las condiciones operativas y optimizando su eficiencia según las exigencias de cada actividad.

La elección de la configuración óptima ha dependido del balance entre la necesidad de agilidad y la exigencia de estabilidad en la operación del robot. Un sistema de control bien diseñado ha considerado estos factores para garantizar que el robot pueda operar eficientemente bajo las diferentes condiciones requeridas por cada tarea.

3.1.6 Análisis de control

En base a los valores propios, se han identificado las frecuencias naturales más predominantes en el sistema, así como los coeficientes de amortiguamiento asociados a dichas frecuencias naturales los cuales se muestran en la **Tabla 3.9**, **Tabla 3.10**, **Tabla 3.11** y **Tabla 3.12**. Estos coeficientes de amortiguamiento han jugado un papel crucial en la determinación de la respuesta dinámica del sistema.

Dependiendo de las razones de amortiguamiento observadas en los resultados obtenidos, se ha podido variar la matriz de amortiguamiento del sistema para mejorar la respuesta global del robot. Esto ha implicado ajustar los valores de amortiguación de las articulaciones del robot redundante de manera que se ha optimizado la estabilidad y la precisión del sistema bajo las condiciones operativas especificadas.

La estrategia de control se ha enfocado en mitigar las resonancias y evitar la excitación de modos de vibración indeseables, especialmente en frecuencias críticas identificadas en el análisis de FFT. Al ajustar adecuadamente la matriz de amortiguamiento, se ha mejorado la estabilidad del

robot, minimizando las oscilaciones y garantizando un desempeño confiable en aplicaciones que requieren alta precisión y estabilidad.

Este enfoque ha permitido seleccionar de manera informada los valores de amortiguación para cada configuración, optimizando la respuesta del sistema según las necesidades específicas de la tarea a realizar.

Tabla 3.9

Razón de amortiguamiento y frecuencias naturales de la primera configuración

	$\lambda_{1,3}$	$\lambda_{2,4}$
ω_n	2.754 rad/s	12.3423 rad/s
ζ	0.0505	0.4422

Tabla 3.10

Razón de amortiguamiento y frecuencias naturales de la segunda configuración

	$\lambda_{1,4}$	$\lambda_{2,3}$
ω_n	9.055 rad/s	2.654 rad/s
ζ	0.8394	0.1522

Tabla 3.11

Razón de amortiguamiento y frecuencias naturales de la tercera configuración

	$\lambda_{1,2}$	λ_3	λ_4
ω_n	3.382 rad/s	6.4338 rad/s	31.6007 rad/s
ζ	0.2165	1.0000	1.0000

Tabla 3.12

Razón de amortiguamiento y frecuencias naturales de la cuarta configuración

	$\lambda_{1,4}$	$\lambda_{2,3}$
ω_n	17.133 rad/s	3.906 rad/s
ζ	0.556	0.101

3.1.7 Análisis de costos

El desarrollo de este proyecto se ha llevado a cabo utilizando herramientas de software gratuitas como Python en Visual Studio Code (VSCode) y LabPlot. Estas herramientas han permitido la implementación de algoritmos complejos y la visualización de datos sin incurrir en costos de licencias de software, lo que ha representado una ventaja económica significativa. Sin embargo, al considerar un análisis de costos integral, se ha estimado necesario tener en cuenta los recursos computacionales y el tiempo de trabajo del programador, los cuales han sido componentes esenciales en la ejecución del proyecto.

Para llevar a cabo este proyecto y obtener resultados concretos, se ha estimado la necesidad de contar con un programador durante un período de al menos 3 meses. Durante este tiempo, el programador ha sido responsable de desarrollar, probar y optimizar los algoritmos necesarios, así como de realizar análisis de datos y generar las visualizaciones correspondientes. Además, se han utilizado equipos de cómputo personales, cuyo costo también se ha considerado como parte del presupuesto del proyecto.

A continuación, la **Tabla 3.13** detalla los costos asociados a los recursos computacionales y al tiempo de trabajo del programador:

Tabla 3.13

Costos Asociados

Componente	Cantidad	Precio c/u (USD)	Total
Recursos computacionales	1	\$700	\$700
Programador (3 meses)	1	\$1000/mes	\$3000

El costo total estimado para la realización del proyecto ha sido de \$3700. Este monto ha incluido tanto la inversión en recursos computacionales como la contratación de un programador durante el tiempo necesario para el desarrollo del proyecto. Aunque los costos de software han sido nulos, el éxito del proyecto ha dependido en gran medida de la disponibilidad de recursos

computacionales adecuados y de la dedicación del programador, cuyo trabajo ha sido crucial para la implementación y optimización de los algoritmos.

Finalmente, para complementar el análisis de costos, es importante considerar no solo los recursos ya mencionados, sino también las adecuaciones o modificaciones que podrían ser necesarias en los sistemas robóticos actuales para implementar las funciones propuestas en el proyecto.

Estas adecuaciones pueden incluir ajustes en hardware, como sensores adicionales o modificaciones estructurales para mejorar la precisión y eficiencia del robot, así como actualizaciones de software necesarias para integrar los nuevos algoritmos y funciones desarrolladas. Además, el acceso a datos o mediciones específicas del robot, como su comportamiento dinámico en diferentes condiciones, puede requerir instrumentos adicionales o procesos de calibración, lo que también podría impactar en los costos totales del proyecto.

Además, puede ser necesario entrenar al personal encargado de operar los robots para que entiendan cómo funciona el nuevo sistema de control de vibraciones y cómo ajustar o interpretar los datos generados y posteriormente implementarlo en el sistema robótico.

También sería útil tener en cuenta los posibles costos asociados con la adquisición de estos datos o mediciones, así como cualquier gasto adicional relacionado con pruebas, ajustes o validaciones que deban realizarse para asegurar el correcto funcionamiento de las nuevas funciones implementadas.

CAPÍTULO 4

4.1 Conclusiones y recomendaciones

4.1.1 Conclusiones

- El objetivo de optimizar la modulación dinámica en robots redundantes se ha alcanzado de manera efectiva, logrando una mejora significativa en la respuesta del sistema y la reducción de vibraciones indeseadas. Esta optimización se ha basado en un enfoque meticuloso en el ajuste de la matriz de amortiguación, lo que ha permitido una respuesta más precisa y controlada del robot en diferentes escenarios operativos. Los resultados obtenidos confirman que una correcta parametrización dinámica es fundamental para garantizar un rendimiento óptimo en aplicaciones industriales donde la precisión, la estabilidad y la eficiencia son cruciales. Además, la reducción de vibraciones ha contribuido a prolongar la vida útil de los componentes mecánicos del robot, disminuyendo el desgaste y los costos de mantenimiento asociados.
- A lo largo del proceso, se identificaron con precisión las frecuencias naturales predominantes y los modos de vibración más críticos del sistema robótico. Esta identificación ha sido posible gracias al uso de herramientas avanzadas de análisis, como la Transformada Rápida de Fourier (FFT), que han permitido descomponer las señales en sus componentes frecuenciales y evaluar cómo estas frecuencias afectan el comportamiento dinámico del robot. La identificación de estas frecuencias ha sido clave para el desarrollo de estrategias de control más efectivas, ajustadas específicamente para mitigar las resonancias y mejorar la estabilidad del robot. Este enfoque ha permitido que el robot opere de manera fiable y segura, incluso en entornos operativos complejos y desafiantes.
- La adaptación de los coeficientes de amortiguamiento en función de las condiciones operativas ha demostrado ser un elemento crucial para alcanzar un equilibrio óptimo entre agilidad y estabilidad en los robots analizados. Al ajustar estos coeficientes, se ha logrado

minimizar las oscilaciones y vibraciones que podrían comprometer la precisión del robot durante su operación. Esta adaptación no solo ha mejorado la robustez del sistema frente a perturbaciones externas, sino que también ha permitido una mayor flexibilidad para ajustarse a diferentes tareas y requisitos operativos. Como resultado, el robot ha podido mantener un alto nivel de desempeño, prolongando al mismo tiempo la vida útil de sus componentes y reduciendo el riesgo de fallos mecánicos.

- Las gráficas obtenidas, tales como las FFT y las respuestas dinámicas en el dominio del tiempo, han sido herramientas fundamentales en el proceso de visualización y comprensión del comportamiento dinámico del sistema. Estas representaciones gráficas han proporcionado una visión clara de cómo se comporta el robot bajo diferentes configuraciones y han facilitado la identificación de patrones y tendencias que podrían no ser evidentes a través de otros métodos de análisis. La información obtenida de estas gráficas ha sido crucial para la toma de decisiones en la optimización de los parámetros de control, permitiendo realizar ajustes precisos y mejorar la respuesta global del robot. Este enfoque visual y analítico ha sido esencial para garantizar que el sistema esté adecuadamente ajustado para cumplir con las exigencias de las tareas a las que será sometido.

4.1.2 Recomendaciones

- Perfeccionar continuamente el algoritmo de modulación dinámica para incrementar la precisión en la identificación y control de los modos de vibración del robot. A medida que se avance en este perfeccionamiento, será posible mejorar aún más la eficiencia operativa del robot, especialmente en aplicaciones donde se requiere una alta precisión y un control riguroso de las vibraciones. Este perfeccionamiento también podría incluir la integración de técnicas de simulación más avanzadas, que permitan prever y ajustar la respuesta del sistema ante una gama más amplia de condiciones operativas.

- Extender la metodología desarrollada a otros sistemas robóticos que presenten diferentes grados de libertad y configuraciones mecánicas. Al aplicar esta metodología a robots con arquitecturas más complejas, se podrá evaluar su eficacia en un contexto más amplio y detectar posibles áreas de mejora. Esta extensión permitiría validar la versatilidad y aplicabilidad del enfoque propuesto, así como adaptar las soluciones a una variedad de industrias y aplicaciones, aumentando así su impacto y relevancia en el campo de la robótica.
- Investigar la inclusión de técnicas de aprendizaje automático que permitan al sistema ajustar los parámetros de control de manera autónoma y en tiempo real. Estas técnicas podrían mejorar significativamente la adaptabilidad del robot a diversas condiciones operativas, permitiéndole optimizar su desempeño sin intervención humana constante. La capacidad de aprendizaje y adaptación del robot podría ser un factor decisivo en su rendimiento a largo plazo, especialmente en entornos dinámicos o impredecibles. Además, el aprendizaje automático podría facilitar la implementación de estrategias de control más sofisticadas, que no solo respondan a las condiciones actuales, sino que también anticipen y prevengan problemas antes de que ocurran.
- Evaluar la aplicación de los resultados obtenidos en entornos industriales reales para validar su efectividad en operaciones de mayor escala y complejidad. Esta evaluación permitirá determinar si las mejoras observadas en un entorno controlado de simulación se traducen en beneficios tangibles cuando se implementan en situaciones reales de producción. Además, esta evaluación ofrecerá una retroalimentación valiosa que podría guiar futuras mejoras en la metodología y en el diseño de los sistemas de control, asegurando que los robots puedan cumplir con las expectativas de rendimiento en entornos industriales exigentes.

REFERENCIAS

- [1] C. Cheng, J. Fu, H. Su, and L. Ren, “Recent Advancements in Agriculture Robots: Benefits and Challenges,” Jan. 01, 2023, *MDPI*. doi: 10.3390/machines11010048.
- [2] Ikusi, “Aplicaciones de los robots industriales: ¿conoce todo lo necesario aquí!” Accessed: Jul. 20, 2024. [Online]. Available: <https://www.ikusi.com/mx/blog/aplicaciones-robots-industriales/>
- [3] A. Ollero, *Robótica; manipuladores y robots móviles*. Marcombo, 2001. Accessed: Jul. 20, 2024. [Online]. Available: http://www.marcombo.com/Robotica_-manipuladores-y-robots-moviles_isbn8426713130.html
- [4] García David, “Los robots móviles revolucionan la industria - infoPLC.” Accessed: Jul. 20, 2024. [Online]. Available: <https://www.infoplcn.net/plus-plus/mercado/item/110010-robots-moviles-revolucionan-industria>
- [5] Gomez Jaime, “ESTUDIO, VALIDACIÓN Y APLICACIONES DE ACELERÓMETRO TRIAXIAL MEMS PARA MEDICIÓN DE VIBRACIONES EN ESTRUCTURAS,” Universidad Austral de Chile, Valdivia, 2020.
- [6] Cervantes Héctor, “Diseño concurrente de un robot paralelo con eslabones flexibles,” CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS DEL INSTITUTO POLITÉCNICO NACIONAL, Ciudad de México, 2018.
- [7] M. Soori, B. Arezoo, and R. Dastres, “Optimization of energy consumption in industrial robots, a review,” *Cognitive Robotics*, vol. 3, pp. 142–157, Jan. 2023, doi: 10.1016/J.COGR.2023.05.003.
- [8] P. Segura Parra, “Modelo de eficiencia energética y mejora continua para la operación esbelta de robótica industrial,” May 2019, Accessed: Jul. 20, 2024. [Online]. Available: <https://repositorio.iberopuebla.mx/handle/20.500.11777/4221>
- [9] E. Cardona and J. R. Gratacós, “Automatización robótica de procesos con inteligencia artificial,” Feb. 2023, Accessed: Jul. 20, 2024. [Online]. Available: <https://upcommons.upc.edu/handle/2117/384518>
- [10] C. M. Campos, M. Ramos, M. Renato, and V. Guerra, “Adopción de tecnologías digitales 4.0 por parte de pequeñas y medianas empresas manufactureras en la Región del Biobío (Chile),” Nov. 18, 2020, *CEPAL*. Accessed: Jul. 20, 2024. [Online]. Available: <https://hdl.handle.net/11362/46387>

- [11] A. Martín, S. Terrile, A. Barrientos, and J. Del Cerro, “Hyper-redundant robots: Classification, state-of-the-art and issues,” *RIAI - Revista Iberoamericana de Automatica e Informatica Industrial*, vol. 15, no. 4, pp. 351–362, 2018, doi: 10.4995/riai.2018.9207.
- [12] B. Siciliano and L. Sciavicco, *Modelling and Control of Robot Manipulators*. Springer, 2000.
- [13] J. Rosen, B. Hannaford, and R. M. Satava, *Surgical Robotics: Systems Applications and Visions*. Springer, 2011.
- [14] Chávez César, “Cinemática inversa de un robot de 6 GDL PRR+RRR (solución paso a paso). - YouTube.” Accessed: Jul. 20, 2024. [Online]. Available: https://www.youtube.com/watch?app=desktop&v=ooMc_V83xRs
- [15] Siciliano Bruno, Sviavicco Lorenzo, Villani Luigi, and Oriolo Giuseppe, *Robotics: Modelling, Planning and Control*, 1st ed. London: Springer, 2009.
- [16] E. O. Brigham, *The fast Fourier transform and its applications*. Prentice-Hall, Inc., 1998.
- [17] D. J. Inman, *Engineering Vibration*. Pearson Education, 2013.
- [18] N. Hogan, “Impedance Control: An Approach to Manipulation: Part II—Implementation,” *J Dyn Syst Meas Control*, vol. 107, no. 1, pp. 8–16, Mar. 1985, doi: 10.1115/1.3140713.
- [19] I. Kao and C. Saldarriaga, “Analytical methodology for the analysis of vibration for unconstrained discrete systems and applications to impedance control of redundant robots,” *ROBOMECH Journal*, vol. 8, no. 1, Dec. 2021, doi: 10.1186/s40648-021-00199-0.
- [20] L. Meirovitch, *Fundamentals of Vibrations*. McGraw-Hill, 2001.
- [21] J. J. Craig, *Introduction to Robotics: Mechanics and Control*. Pearson Education, 2005.
- [22] Molero Blas, “Teoría de Vibraciones | Femap y Simcenter Nastran.” Accessed: Jul. 20, 2024. [Online]. Available: <https://iberisa.wordpress.com/tag/teoria-de-vibraciones/>
- [23] Bathe Klaus-Jürgen, *Finite Element Procedures*. Prentice Hall, 2006.
- [24] S. C. Chapra and R. P. Canale, *Numerical Methods for Engineers*, 7th ed. McGraw-Hill Education, 2015.
- [25] J. N. Reddy, *An Introduction to the Finite Element Method*, 4th ed. McGraw-Hill Education, 2019.
- [26] L. N. Trefethen and D. I. Bau, *Numerical Linear Algebra*. Society for Industrial and Applied Mathematics, 1997.
- [27] Inc. ANSYS, *ANSYS Mechanical User’s Guide*. ANSYS, Inc., 2020.
- [28] S. Moaveni, *Finite Element Analysis: Theory and Application with ANSYS*, 4th ed. Pearson, 2020.

- [29] G. R. Liu and S. S. Quek, *he Finite Element Method: A Practical Course*, 2nd ed. Butterworth-Heinemann, 2013.
- [30] R. D. Cook, D. S. Malkus, M. E. Plesha, and R. J. Witt, *Concepts and Applications of Finite Element Analysis*, 4th ed. Wiley, 2002.
- [31] K. M. Lynch and F. C. Park, *Modern Robotics: Mechanics, Planning, and Control*, 1st ed. Cambridge, U.K.: Cambridge University Press, 2017

APÉNDICES

Apéndice A

Código en Wólfram Matemática

El siguiente enlace al repositorio de GitHub corresponde al código base del método de análisis vibratorio:

Enlace de repositorio: <https://github.com/frquimis/wolfram.git>

```

Clear[m1, m2, m3, c2, c3, c5, k2, k3, k5, q1, q2, q3]

MCKvalues = {m1 -> 8, m2 -> 2, m3 -> 5, c2 -> 15, c3 -> 20, c5 -> 10, k2 -> 1200, k3 -> 1500, k5 ->
2400};
ICs = {q10 -> 0, q20 -> 0.03, q30 -> 0, q1dot0 -> 0, q2dot0 -> 0, q3dot0 -> 0};

MM = {{m1, 0, 0}, {0, m2, 0}, {0, 0, m3}};
CC = {{c2 + c5, -c2, -c5}, {-c2, c2 + c3, -c3}, {-c5, -c3, c3 + c5}};
KK = {{k2 + k5, -k2, -k5}, {-k2, k2 + k3, -k3}, {-k5, -k3, k3 + k5}};

Print["---Mass, damping, and stiffness matrices---"];
MMn = MM /. MCKvalues;
CCn = CC /. MCKvalues;
KKn = KK /. MCKvalues;

{MatrixForm[MMn], MatrixForm[CCn], MatrixForm[KKn]}

Chop[Eigensystem[N[KKn]]];

q0 = {{q10}, {q20}, {q30}} /. ICs;
q0dot = {{q1dot0}, {q2dot0}, {q3dot0}} /. ICs;

Print["---The normalized RB modal vector---"];
u = NullSpace[KKn][[1]];
u0 = u / Sqrt[u . MMn . u];
u0 = N[u0];

eq1 = u0 . MMn . {q1, q2, q3};
temp1 = Simplify[Solve[eq1 == 0, q3]];

c31 = q3 /. (temp1[[1]] /. {q2 -> 0});
c32 = q3 /. (temp1[[1]] /. {q1 -> 0});

Cstr = {{1, 0}, {0, 1}, {Chop[c31] / q1, Chop[c32] / q2}};
Print["---The constraint matrix, with q3 eliminated---"];
MatrixForm[Cstr];

MMp = Transpose[Cstr] . MMn . Cstr;
CCp = Transpose[Cstr] . CCn . Cstr;
KKp = Transpose[Cstr] . KKn . Cstr;

{MatrixForm[MMp], MatrixForm[CCp], MatrixForm[KKp]}

```

```

uQimpulse = u0 . {1, 0, 0};
damping = Chop[u0 . CCn . u0];

\[Beta]solImpulse = InverseLaplaceTransform[uQimpulse / (s^2 + damping s), s, t];
qRBimpulse = Chop[Simplify[\[Beta]solImpulse u0]];
q0NRB = q0 - (u0 . MMn . q0)[[1]] u0;
qdot0NRB = q0dot - (u0 . MMn . q0dot)[[1]] u0;

O2 = {{0, 0}, {0, 0}};
I2 = IdentityMatrix[2];

MMinv = Inverse[MMp];
mm3 = -MMinv . KKp;
mm4 = -MMinv . CCp;

A = {
  Join[O2[[1]], I2[[1]]],
  Join[O2[[2]], I2[[2]]],
  Join[mm3[[1]], mm4[[1]]],
  Join[mm3[[2]], mm4[[2]]]
};

B = {O2[[1]], O2[[2]], MMinv[[1]], MMinv[[2]]};
Print["--B--"];
MatrixForm[B];

{vals, vecs} = Chop[Simplify[Eigensystem[N[A]]]];
yi = Transpose[Inverse[vecs]];
Clear[lambda];

lambda = {vals[[3]], vals[[4]], vals[[1]], vals[[2]]};
x1 = vecs[[3]];
x2 = vecs[[4]];
x3 = vecs[[1]];
x4 = vecs[[2]];
y1 = yi[[3]];
y2 = yi[[4]];
y3 = yi[[1]];
y4 = yi[[2]];

YT = {y1, y2, y3, y4};
X = Transpose[{x1, x2, x3, x4}];
MatrixForm[{{x1, x2, x3, x4}}];
MatrixForm[{{y1, y2, y3, y4}}];
Chop[YT.X] // MatrixForm
Chop[YT.A.X] // MatrixForm
Ediag = {
  {Exp[lambda[[1]] t], 0, 0, 0},
  {0, Exp[lambda[[2]] t], 0, 0},
  {0, 0, Exp[lambda[[3]] t], 0},
  {0, 0, 0, Exp[lambda[[4]] t]}
};

x0 = {
  {q0NRB[[1]]},
  {q0NRB[[2]]},
  {qdot0NRB[[1]]},
  {qdot0NRB[[2]]}
};

xt = X . Ediag . YT . x0;
xtt = Chop[Simplify[Chop[ComplexExpand[xt]]]];
qNRB = Simplify[Cstr . {xtt[[1]], xtt[[2]]}];
(* Definición de Ediagt\[Tau] usando los valores ajustados/sincronizados con los eigenvectores *)
EdiagtTau = {
  {Exp[lambda[[1]] (t - \[Tau])], 0, 0, 0},
  {0, Exp[lambda[[2]] (t - \[Tau])], 0, 0},
  {0, 0, Exp[lambda[[3]] (t - \[Tau])], 0},
  {0, 0, 0, Exp[lambda[[4]] (t - \[Tau])]}
};

```

```

(* Integrar la integral de convolución; q3 fue eliminado en Cstr *)
\[Phi]tTau = X . EdiagtTau . YT;
StQ = Transpose[Cstr] . {1, 0, 0};
xImpulse = Chop[Simplify[ComplexExpand[X . Ediag . YT] . B . StQ]];
Print["---qNRB: Parte NRB de la respuesta de vibración de IMPULSO ---"]
qNRBimpulse = Simplify[Cstr . {xImpulse[[1]], xImpulse[[2]]}];
impPlot1 = Plot[
  {qNRBimpulse[[1]], qNRBimpulse[[2]], qNRBimpulse[[3]]},
  {t, 0, 1},
  PlotStyle -> {Red, Blue, Green},
  AxesLabel -> {t, "response"},
  PlotRange -> Automatic,
  Frame -> True,
  GridLines -> Automatic
]
StQ = Transpose[Cstr] . {1, 0, 0};
(* Cálculo de nn *)
nn = YT . B . StQ;
(* Transformaciones de InverseLaplace *)
\[Xi]1 = ComplexExpand[InverseLaplaceTransform[(nn[[1]])/(s - lambda[[1]]), s, t]];
\[Xi]2 = ComplexExpand[InverseLaplaceTransform[(nn[[2]])/(s - lambda[[2]]), s, t]];
\[Xi]3 = ComplexExpand[InverseLaplaceTransform[(nn[[3]])/(s - lambda[[3]]), s, t]];
\[Xi]4 = ComplexExpand[InverseLaplaceTransform[(nn[[4]])/(s - lambda[[4]]), s, t]];
(* Cálculo de xximp *)
xximp = Chop[ComplexExpand[X . {\[Xi]1, \[Xi]2, \[Xi]3, \[Xi]4}]];
(* Simplificación y graficación de la respuesta de vibración de impulso *)
qNRBimpulse2 = Simplify[Cstr . {xximp[[1]], xximp[[2]]}];
impPlot2 = Plot[ {qNRBimpulse2[[1]], qNRBimpulse2[[2]], qNRBimpulse2[[3]]},
  {t, 0, 1}, PlotStyle -> {Red, Blue, Green}, AxesLabel -> {t, "response"}, PlotRange -> Automatic, Frame -
  > True, GridLines -> Automatic ]

(* Impresión de mensaje de método HIIL para respuesta de impulso *)
Print["---METHOD HIIL: The overall IMPULSE response @q1, q2, q3D---"];

(* Cálculo de la respuesta de impulso total *)
qIMPULSE = qRBimpulse + qNRBimpulse2;

```

Apéndice B

Respuestas en Wólfram Matemática

Las siguientes imágenes representan las respuestas del código base en wólfram matemática basado en análisis vibratorio de un sistema mecánico:

Figura 5.1

Respuesta qrn b con una fuerza impulso externa

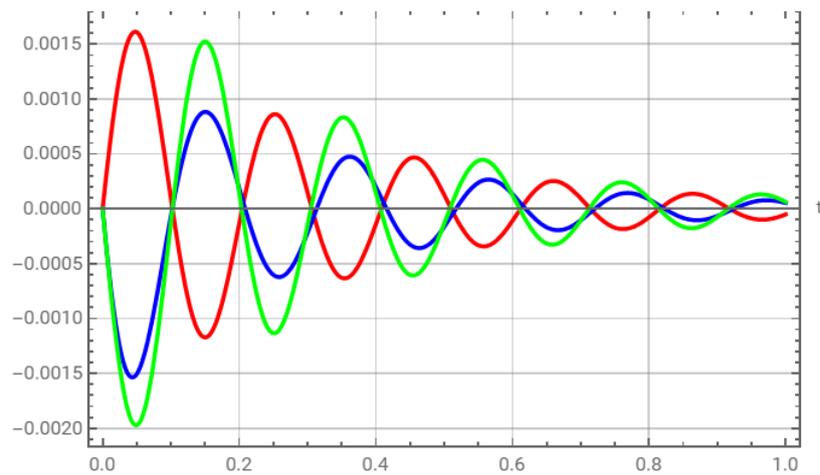
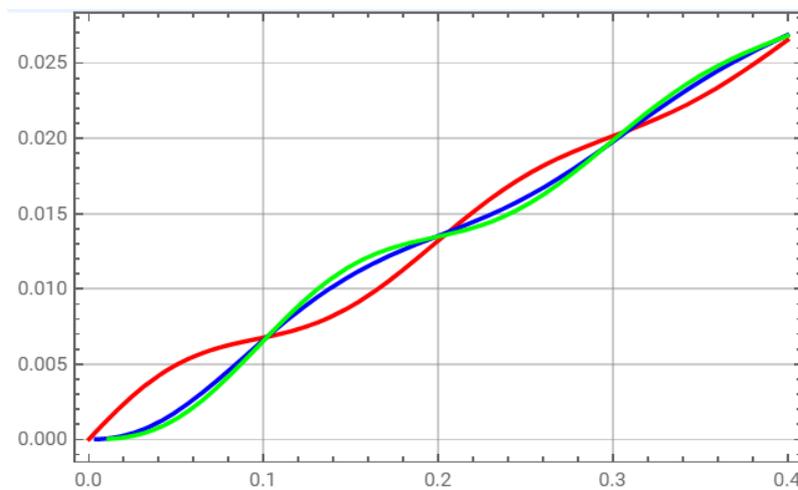


Figura 5.2

Respuesta total con una fuerza impulso externa



Apéndice C

Código de matriz de inercia en Python

El siguiente enlace al repositorio de GitHub corresponde al código generado del método de análisis vibratorio para un robot 3R planar redundante:

Enlace de repositorio: <https://github.com/frquimis/Tesis.git>

```
import sympy as sp
from sympy import nsimplify
import pickle

# Funciones importadas (deben estar definidas en rigidez.py y metodo.py)
# from Tesis.rigidez import rigidez, Amortiguamiento
# from Tesis.tensor import calcular_tensor_inercia

def denavit_hartenberg_com(a, Lcom):
    comDH = sp.Matrix([
        [0, 0, 0, 0],
        [a[0], 0, Lcom[0], 0],
        [a[1], 0, Lcom[1], 0],
        [a[2], 0, Lcom[2], 0]
    ])
    return comDH

def dh_transform(theta, d, a, alpha):
    """
    Create the Denavit-Hartenberg transformation matrix.
    """
    return sp.Matrix([
        [sp.cos(theta), -sp.sin(theta) * sp.cos(alpha), sp.sin(theta) * sp.sin(alpha), a *
        sp.cos(theta)],
        [sp.sin(theta), sp.cos(theta) * sp.cos(alpha), -sp.cos(theta) * sp.sin(alpha), a *
        sp.sin(theta)],
        [0, sp.sin(alpha), sp.cos(alpha), d],
        [0, 0, 0, 1]
    ])

def forward_kinematics_dh(DH, up_to_joint):
    """
    Calculate the forward kinematics for a chain up to a given joint.
    """
    T = sp.eye(4)
    for i in range(up_to_joint):
        theta, d, a, alpha = DH.row(i)
        T *= dh_transform(theta, d, a, alpha)
    return T
```

```

def forward_kinematics_dh_com(DH, comDH, rb):
    """
    Calculate the forward kinematics to the center of mass of a requested rigid body.
    """
    Lcom = sp.symbols(f'Lcom{rb}')

    frame = None
    for i in range(comDH.shape[0]):
        if comDH[i, 2] == Lcom:
            frame = i
            break

    if frame is None:
        raise ValueError(f"Lcom{rb} not found in comDH matrix.")

    # Forward kinematics to the previous reference frame
    H = forward_kinematics_dh(DH, frame)

    # Forward kinematics to the requested center of mass
    Rz = dh_transform(comDH[frame, 0], 0, 0, 0)
    Tz = dh_transform(0, comDH[frame, 1], 0, 0)
    Tx = dh_transform(0, 0, comDH[frame, 2], 0)

    Hcom = sp.simplify(H * Rz * Tz * Tx)
    return Hcom

def matriz_jaco(DH, comDH, a, rb):
    Lcom1 = sp.symbols(f'Lcom{rb}')
    comFrame = None
    for i in range(comDH.shape[0]):
        if comDH[i, 2] == Lcom1:
            comFrame = i
            break

    # Calcular la cinemática directa al centro de masa solicitado
    Hcom = forward_kinematics_dh_com(DH, comDH, rb)

    # Obtener el número de coordenadas generalizadas
    n = len(a)

    # Inicializar la matriz Jacobiana con ceros
    Jcom = sp.zeros(6, n)
    com1 = comDH[:, 0]

    for i in range(1, n + 1): # i toma valores 1, 2 y 3
        symbol_to_find = sp.Symbol(f'a{i}')

        jointFrame = next((j for j, val in enumerate(com1) if val == symbol_to_find), None)

        # Si la articulación actual está en un marco de referencia que no afecta el COM
        if jointFrame > comFrame:
            # Detener la construcción
            break

        H = forward_kinematics_dh(DH, jointFrame)
        Rz = dh_transform(comDH[jointFrame, 0], 0, 0, 0)

        H = H * Rz
        z = sp.Matrix(H[0:3, 2])

        r = sp.Matrix(sp.simplify(Hcom[0:3, 3] - H[0:3, 3]))

        Jcom[0:3, i - 1] = z.cross(r)
        Jcom[3:6, i - 1] = z

    return Jcom

```

```

def MatrizInercia(m, dh, comDh, a, steps):
    rb = len(m)
    n = len(a)
    D = sp.zeros(n, n)

    for j in range(1, rb + 1):
        hcom = forward_kinematics_dh_com(dh, comDh, j)
        tensor1 = calcular_tensor_inercia(m[j - 1], dh[j - 1, 2])

        I = hcom[0:3, 0:3].transpose() * tensor1 * hcom[0:3, 0:3]
        jcom = matriz_jaco(dh, comDh, a, j)
        ele1 = sp.simplify(m[j - 1] * jcom[0:3, :].transpose() * jcom[0:3, :])
        ele2 = sp.simplify(jcom[3:, :].transpose() * I * jcom[3:, :])
        D = D + sp.simplify(ele1 + ele2)
    return nsimplify(D)

# Función para calcular el tensor de inercia simbólicamente
def calcular_tensor_inercia(m, l):
    Ixx = Iyy = (1 / 12) * m * l ** 2
    Izz = (1 / 6) * m * l ** 2
    tensor = sp.Matrix([
        [Ixx, 0, 0],
        [0, Iyy, 0],
        [0, 0, Izz]
    ])
    return tensor.evalf(2)

def save_variable(variable, filename):
    with open(filename, 'wb') as file:
        pickle.dump(variable, file)

def load_variable(filename):
    with open(filename, 'rb') as file:
        return pickle.load(file)

```

Código generado para sacar la respuesta global del sistema utilizando funciones

```

import sympy as sp

from Tesis.wf import simplify_expression

def Matriz_S(Vector, Matriz):
    q1, q2, q3 = sp.symbols('q1 q2 q3')
    q = sp.Matrix([q1, q2, q3])
    u0T_M = Vector.T * Matriz
    simplificado = sp.simplify(u0T_M * q)

    equation = sp.Eq(simplificado[0], 0)

    # Resolver para q3 en términos de q1 y q2
    solution = sp.solve(equation, q3)

    Matriz_S1 = sp.zeros(3, 2)
    Matriz_S1[0, 0] = 1
    Matriz_S1[1, 1] = 1
    valorq1 = solution[0].subs({sp.symbols('q2'): 0}) / sp.symbols('q1')
    valorq2 = solution[0].subs({sp.symbols('q1'): 0}) / sp.symbols('q2')
    Matriz_S1[2, 0] = valorq1
    Matriz_S1[2, 1] = valorq2
    return Matriz_S1

def qzb(ics, vics, Uo, M):
    t = sp.symbols('t')
    beta_0 = Uo.T * M * ics
    betadot = Uo.T * M * vics

    beta_t = (betadot * t) + beta_0

    qzbr = Uo * sp.simplify(beta_t)
    qRB0 = Uo * beta_0
    qRBdot0 = Uo * betadot
    return qzbr, qRB0, qRBdot0

# NZP motions
def NZP(Mm, Cc, Rc, S1):
    Mp = (S1.T * Mm * S1)
    Cp = (S1.T * Cc * S1)
    Kp = (S1.T * Rc * S1)

    0 = sp.zeros(2, 2)
    I = sp.eye(2)
    nueva = 0.row_join(I)
    MMinv = Mp.inv()

    mm3 = (-MMinv) * Kp
    mm4 = (-MMinv) * Cp

    partInfe = mm3.row_join(mm4)
    A = nueva.col_join(partInfe)
    B = 0.col_join(MMinv)
    X1 = A.eigenvecs()
    eigen_matrix = sp.Matrix.hstack(*[vects[0] for val, mult, vects in X1])
    Y1 = (eigen_matrix.inv()).T

    return eigen_matrix, Y1, X1, B

def matriz(valores):
    valores_propios = [val for val, mult, vects in valores]
    exp_eigenvalues = [sp.exp(val * sp.symbols('t')) for val in valores_propios]
    matriz = sp.diag(*exp_eigenvalues)
    return matriz

def xt(X1, Y1, valores, B, S, Q):
    matrizval = matriz(valores)
    #x = X1 * matrizval * Y1.T * B * S.T * Q
    x = X1 * matrizval * Y1.T * Q
    return x

```

Código para las comprobaciones con diferentes escenarios en Python

```

import sympy as sp

from Tesis.Matriz_Inercia import denavit_hartenberg_com, MatrizInercia, save_variable
from Tesis.wf import matriz_jaco_planar, rigidez, Amortiguamiento, vector_N

a = [sp.symbols('a1'), sp.symbols('a2'), sp.symbols('a3')]

# centro de masa de los eslabones
Lcom = [sp.symbols('Lcom1'), sp.symbols('Lcom2'), sp.symbols('Lcom3')]

# parametros DH de centros de masa
comDH = denavit_hartenberg_com(a, Lcom)

# Example DH parameters for a 3-joint manipulator
DH = sp.Matrix([
    [0, 0, 0, 0],
    [sp.symbols('a1'), 0, sp.symbols('l1'), 0],
    [sp.symbols('a2'), 0, sp.symbols('l2'), 0],
    [sp.symbols('a3'), 0, sp.symbols('l3'), 0],
])

# condiciones iniciales de los eslabones masas etc
m = [10, 5, 2]
valores = {sp.symbols('Lcom1'): 0.4 / 2, sp.symbols('Lcom2'): 0.25 / 2, sp.symbols('Lcom3'): 0.2 / 2}
angulos = {sp.symbols('a1'): 0.529, sp.symbols('a2'): 0.6, sp.symbols('a3'): 0.105}
k = {sp.symbols('Kx'): 300, sp.symbols('Ky'): 300}
c = {sp.symbols('Cx'): 20, sp.symbols('Cy'): 15}
longitudes = {sp.symbols('l1'): 0.4, sp.symbols('l2'): 0.25, sp.symbols('l3'): 0.2}

jac = matriz_jaco_planar(DH, a)
jacEvalf = matriz_jaco_planar(DH, a).subs(angulos).subs(longitudes)

Kq = rigidez(jac[0:2, :]).subs(k)

Cq = Amortiguamiento(jac[0:2, :]).subs(c).subs(angulos).subs(longitudes)

d1 = MatrizInercia(m, DH, comDH, a, 1)
'''matrizInercia=sp.Matrix([
    [2.013, 0.4634, 0.2542],
    [0.4634, 0.6215, 0.1339],
    [0.2542, 0.1339, 0.0750]
...
])
matriz_evaluada = d1.subs(valores).subs(angulos).subs(longitudes)
sp.pprint(matriz_evaluada)

m1 = vector_N(Kq, angulos, longitudes)

save_variable(m1, 'matriz_nula.pkl')

save_variable(jacEvalf, 'Jacobiano.pkl')

Kqeva = Kq.subs(angulos).subs(longitudes)

save_variable(Cq, 'amortiguamiento.pkl')

save_variable(Kqeva, 'rigidez.pkl')
save_variable(matriz_evaluada, 'inercia.pkl')

```

```

from Tesis.Cinemática_Inversa import rrrIK, forward_kinematics_3R
from Tesis.Matriz_Inercia import load_variable, save_variable
import sympy as sp
from Tesis.metodo import Matriz_S, qzb
from Tesis.wf import espacio_N_Normali, funcion

#elementos cargados
matriz_nula_cargada = load_variable('matriz_nula.pkl')
m1 = load_variable('amortiguamiento.pkl')
jaco = load_variable('Jacobiano.pkl')
m2 = load_variable('rigidez.pkl')
m3 = load_variable('inercia.pkl')

longitudes = {sp.symbols('l1'): 0.4, sp.symbols('l2'): 0.25, sp.symbols('l3'): 0.2}
angulos = {sp.symbols('a1'): 0.529, sp.symbols('a2'): 0.6, sp.symbols('a3'): 0.105}

#creacion de distintos escenarios
qx = 0.012
qy = 0
vx = 0
vy = 0
x, y, phi = forward_kinematics_3R(0.529, 0.6, 0.105, 0.4, 0.25, 0.2)
q0 = rrrIK(0.4, 0.25, 0.2, [(x - qx), (y + qy), phi])

condicio = sp.Matrix(q0)
sp.pprint(condicio)
qdot0 = funcion(vx, vy, jaco[0:2, :])
normalizado = espacio_N_Normali(matriz_nula_cargada, m3)

qzb1, qrb, qrbdot = qzb(condicio, qdot0, normalizado, m3)
sp.pprint(qzb1)
QNRB = condicio - qrb
QNRB_dot = qdot0 - qrbdot

x0 = QNRB[0:2, :].col_join(QNRB_dot[0:2, :])

S = Matriz_S(normalizado, m3)
save_variable(qzb1, 'cuerpo_rigido.pkl')
save_variable(x0, 'xzero.pkl')
save_variable(S, 'RESTRICCIONES.pkl')

```

```

import numpy as np
import sympy as sp
from matplotlib import pyplot as plt
from sympy import Abs

from Tesis.Matriz_Inercia import load_variable, save_variable
from Tesis.metodo import NZP, xt, matrice

Mamort = load_variable('amortiguamiento.pkl')
Mrestric = load_variable('RESTRICCIONES.pkl')
Mrigi = load_variable('rigidez.pkl')
Miner = load_variable('inercia.pkl')
qo = load_variable('xcero.pkl')
derechos, izquierdos, valores, B = NZP(Miner, Mamort, Mrigi, Mrestric)
x1t = xt(derechos, izquierdos, valores, B, Mrestric, qo)
QNBZ=sp.simplify(Mrestric * x1t[0:2, :])
QNBZ= sp.simplify(sp.simplify(QNBZ))

eigenvalores = [val[0] for val in valores]
sp.pprint(eigenvalores)
modulos = [Abs(val) for val in eigenvalores]
sp.pprint(modulos)

save_variable(QNBZ, 'cuerpoNoRigido.pkl')

qNzb1_funcs = [sp.lambdify(sp.symbols('t'), comp, 'numpy') for comp in QNBZ]

# Evaluar las funciones en un rango de valores para t
t_values = np.linspace(0, 8, 400)
qzb1_values = [func(t_values) for func in qNzb1_funcs]

# Graficar las componentes de qzb1
plt.figure(figsize=(10, 6))
for i, qzb1_val in enumerate(qzb1_values):
    plt.plot(t_values, qzb1_val, label=f'QNZP_{i + 1}')
plt.xlabel('t(s)')
plt.ylabel('Amplitud')
plt.title('Componentes de QNZP')
plt.legend()
plt.grid(True)
plt.show()

import numpy as np
from matplotlib import pyplot as plt
import sympy as sp

from Tesis.Matriz_Inercia import load_variable, save_variable

qzb = load_variable('cuerpo_rigido.pkl')
qznb = load_variable('cuerpoNoRigido.pkl')

qz = qzb + qznb

save_variable(qz, 'qt.pkl')
#sp.pprint(qz)

qNzb1_funcs = [sp.lambdify(sp.symbols('t'), comp, 'numpy') for comp in qz]

# Evaluar las funciones en un rango de valores para t
t_values = np.linspace(0, 15, 400)
qzb1_values = [func(t_values) for func in qNzb1_funcs]

# Graficar las componentes de qzb1
plt.figure(figsize=(10, 6))
for i, qzb1_val in enumerate(qzb1_values):
    plt.plot(t_values, qzb1_val, label=f'q(t){i + 1}')
    np.savetxt("Qimp{}.csv".format(i + 1), np.column_stack((qzb1_val.real, t_values)), delimiter=',')
    print(qzb1_val)
plt.xlabel('t(s)')
plt.ylabel('Amplitud')
plt.title('freeVibration')
plt.legend()
plt.grid(True)
plt.show()

```

Apéndice D

Cálculos en Python más relevantes

Las siguientes imágenes son las ecuaciones y resultados más relevantes del código en Python

Figura 5.3

Matriz de inercia

```
[2.85606749130882  0.909070873791837  0.151486885611723 ]
|
|
|0.909070873791837  0.476740922941516  0.0905579614707582|
|
|0.151486885611723  0.0905579614707582  0.0408333333333333|
```

Figura 5.4

Matriz articular de rigidez

```
[194.662268261615  103.623522572035  45.1960656835169]
|
|103.623522572035  60.5847768824549  26.9173884412275|
|
|[45.1960656835169  26.9173884412275  12.0]
```

Figura 5.5

Matriz de amortiguamiento

```
[11.6342640119956  6.45993905075975  2.84178804148533 ]
|
|6.45993905075975  3.88936776395657  1.73732737880255 |
|
|[2.84178804148533  1.73732737880255  0.778158562469428]
```

Figura 5.6*QZP del sistema*

$$\begin{bmatrix} -0.105313546269596 \\ 1.14717725011584 \\ -2.17660480814521 \end{bmatrix}$$

Figura 5.7*Valores propios*

$$\begin{aligned} &[-9.52838198276524 - 14.2474318147307i, -0.395149574282438 - 3.8859841633193i \curvearrowright \\ &\curvearrowleft i, -0.395149574282438 + 3.8859841633193i, -9.52838198276524 + 14.2474318147 \\ &\curvearrowleft 307i] \end{aligned}$$

1. $-9.52838198276524 - 14.2474318147307i$
2. $-0.395149574282438 - 3.8859841633193i$
3. $-0.395149574282438 + 3.8859841633193i$
4. $-9.52838198276524 + 14.2474318147307i$

Figura 5.8*Ecuaciones de QNZP+*

$$\begin{bmatrix} (0.00221016025481463 + 0.0233425592160358i) \cdot (0.102271831822682 + 0.307329133936101i) \cdot e^{-(0.2186811313301 - 14.2474318147307i)t} + (0.00233761914332884 - 0.0242771070822589i) \cdot (0.0978498039406173 - 0.295609642466083i) \cdot e^{-(19.4519135398129 - 3.8859841633193i)t} \\ (0.0139464816113826 - 0.051459725235728i) \cdot (0.102271831822682 + 0.307329133936101i) \cdot e^{-(0.2186811313301 - 14.2474318147307i)t} + (0.014421878073762 + 0.05355111905549454i) \cdot (0.0978498039406173 - 0.295609642466083i) \cdot e^{-(19.4519135398129 - 3.8859841633193i)t} \\ (1.60253037793535 + 1.07480757758069i) \cdot e^{-(0.2186811313301 - 14.2474318147307i)t} + (0.0692108212884932 + 0.0182900184781829i) \cdot e^{-(19.4519135398129 - 3.8859841633193i)t} \\ (1.60253037793535 - 1.07480757758069i) \cdot e^{-(0.2186811313301 - 14.2474318147307i)t} + (0.133554396196224 - 0.0226723092344352i) \cdot e^{-(19.4519135398129 - 3.8859841633193i)t} - (0.216829930102131 - 0.0367962682719889i) \cdot e^{-(19.4519135398129 - 3.8859841633193i)t} \end{bmatrix}$$

