

Universidad de Ingeniería y Tecnología

TESIS DE PREGRADO

Diseño y simulación de un controlador de impedancia en robots manipuladores para interacción con superficies irregulares

Fabian, Joao

Award date:
2022

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

CARRERA DE INGENIERÍA ELECTRÓNICA



**DISEÑO Y SIMULACIÓN DE UN CONTROLADOR
DE IMPEDANCIA EN ROBOTS MANIPULADORES
PARA INTERACCIÓN CON SUPERFICIES
IRREGULARES**

TESIS

Para optar el título profesional de Ingeniero Electrónico

AUTOR:

Joao Ulises Fabián Lizárraga 

ASESOR:

Oscar Efraín Ramos Ponce 

Lima - Perú

2022

Dedicatoria:

Al maestro Splinter.

Agradecimientos:

A mi familia, Ruth, Oscar y Romina, en el orden que prefieran leerlo.

A PROCENCIA por el apoyo financiero en el desarrollo de esta tesis mediante el proyecto titulado “Control Háptico Remoto de un Sistema Robótico para Aplicación en Cirugía”, con CONTRATO 144-2018-FONDECYT/BM.

Índice general

	Pág.
RESUMEN	1
ABSTRACT	2
CAPÍTULO 1 INTRODUCCIÓN	3
CAPÍTULO 2 ANTECEDENTES	5
2.1 Control de interacción para robots manipuladores	5
2.1.1 Sistemas con sensado externo	5
2.1.2 Sistemas sin sensado externo	6
2.2 Librerías para estimación de la dinámica	7
2.3 Simuladores físicos	8
2.4 Simuladores robóticos	10
CAPÍTULO 3 MARCO TEÓRICO	11
3.1 Representaciones en el espacio cartesiano	11
3.2 Modelamiento cinemático	12
3.2.1 Cinemática directa	13
3.2.2 Cinemática inversa	14
3.2.3 Jacobiano analítico	14
3.3 Modelamiento dinámico	15
3.3.1 Modelo general de robots manipuladores	15
3.4 Sistema de control	16

3.5	<i>Robot Operating System</i>	17
3.5.1	Elementos principales de ROS	17
3.5.2	Gazebo	19
3.5.3	<i>Rigid-Body Dynamics Library</i>	20
3.6	Robot UR5	20
CAPÍTULO 4 MÉTODOLÓGÍA		22
4.1	Configuración en el entorno ROS	22
4.1.1	Señales de control	22
4.1.2	Flujo de datos	23
4.2	Acondicionamiento del simulador Gazebo	24
4.2.1	Modelo URDF del robot UR5	24
4.2.2	Creación del ambiente de prueba	25
4.2.3	Habilitación de física de colisiones	25
4.3	Implementación del sistema de control	25
4.3.1	Análisis cinemático	26
4.3.2	Análisis dinámico	29
4.3.3	Control de posición por dinámica inversa	32
4.3.4	Control de impedancia	34
CAPÍTULO 5 RESULTADOS Y DISCUSIONES		37
5.1	Control en un entorno sin obstáculos	40
5.1.1	Respuesta sin interacción e impedancia de rigidez baja	40
5.1.2	Respuesta sin interacción e impedancia de rigidez media	44
5.1.3	Respuesta sin interacción e impedancia de rigidez alta	48
5.2	Control en un entorno con un obstáculo	52
5.2.1	Respuesta con interacción e impedancia de rigidez baja	54
5.2.2	Respuesta con interacción e impedancia de rigidez media	58
5.2.3	Respuesta con interacción e impedancia de rigidez alta	62

5.3 Análisis de errores	66
CONCLUSIONES	69
TRABAJO A FUTURO	70
REFERENCIAS BIBLIOGRÁFICAS	71
APÉNDICE	77

Índice de tablas

3.1	Características del robot UR5	21
4.1	Parámetros de Denavit Hartenberg para el robot UR5.	28
5.1	Distancia euclidiana entre la posición del efector final \vec{p} en estado estacionario y la posición deseada \vec{p}_d	67
5.2	Resumen cualitativo del comportamiento obtenido en cada caso estudiado.	68

Índice de figuras

3.1	Cadena cinemática de n juntas. La posición cartesiana de los sistemas de referencia de la base y el efector final están descritas por (x_b, y_b, z_b) y (x, y, z) respectivamente.	13
3.2	Diagrama de elementos principales de ROS interconectados. El flujo de datos va del Nodo 1 hacia el Nodo 2, y del Nodo 2 hacia el Nodo 3. El Nodo 4 es definido, pero no intercambia información con alguno de los otros nodos.	18
3.3	Ambiente de trabajo de Gazebo a) vacío, b) con un robot UR5, c) con objetos, y d) con un robot UR5 y objetos.	19
3.4	Robot UR5 de seis grados de libertad.	21
4.1	Esquema de transformación para la señal de control. La señal \vec{u} , definida por el controlador, ingresa a un modelo estimado del robot. El comportamiento dinámico del sistema controlado es descrito por la aceleración $\ddot{\vec{q}}_r$, que ese integra para hallar la velocidad $\dot{\vec{q}}_r$ y la posición \vec{q}_r	23
4.2	Flujo general de transmisión de datos entre el simulador Gazebo y el nodo de control definido en ROS. \vec{q} y $\dot{\vec{q}}$ representan las posiciones articulares actuales del robot simulado, mientras que \vec{q}_r representa la posición de referencia enviada desde el nodo de control.	24
4.3	Sistemas de referencia definidos para el robot UR5. Los vectores rojo, verde y azul, representan los ejes x , y y z , respectivamente.	27
4.4	Sistema dinámico en una dimensión. La masa m se mueve en dirección del eje x , y es afectada por el estímulo u y la fuerza F_{ext}	34

4.5	Sistema dinámico masa-resorte-amortiguador acoplado a una posición deseada x_d . La masa m se mueve en dirección del eje x , y es afectada por un estímulo F_{ext} y la impedancia generada por un resorte de constante K_d y un amortiguador de constante D_d	35
5.1	Diagrama de bloques de la implementación de los controladores de posición e impedancia para la simulación del robot UR5 en Gazebo. El esquema permite intercambiar entre los controladores sin pérdida de generalidad.	38
5.2	Entorno de simulación sin obstáculo.	40
5.3	Comportamiento controlado del efector final en el espacio cartesiano para los parámetros K_P , K_D , D_{d1} y K_{d1} . El robot parte de la posición inicial (magenta), hacia la posición deseada (verde).	41
5.4	Comportamiento temporal del efector final en los ejes Cartesianos para el robot controlado para los parámetros K_P , K_D , D_{d1} y K_{d1} . Los paneles, de arriba hacia abajo, representan a x , y y z respectivamente.	42
5.5	Comportamiento temporal la orientación del robot controlado para los parámetros K_P , K_D , D_{d1} y K_{d1} . Los paneles, de arriba hacia abajo, representan la orientación parametrizada por los ángulos de Euler ζ , ϕ y ψ respectivamente.	43
5.6	Comportamiento temporal del esfuerzo en cada junta del robot controlado para los parámetros K_P , K_D , D_{d1} y K_{d1} . La variable u_i representa el esfuerzo aplicado por la junta i , donde este subíndice toma valores de 1 a 6 para juntas consecutivas de la base hacia el efector final.	44
5.7	Comportamiento controlado del efector final en el espacio cartesiano para los parámetros K_P , K_D , D_{d2} y K_{d2} . El robot parte de la posición inicial (magenta), hacia la posición deseada (verde).	45
5.8	Comportamiento temporal del efector final en los ejes Cartesianos para el robot controlado para los parámetros K_P , K_D , D_{d2} y K_{d2} . Los paneles, de arriba hacia abajo, representan a x , y y z respectivamente.	46

5.9	Comportamiento temporal la orientación del robot controlado para los parámetros K_P , K_D , D_{d2} y K_{d2} . Los paneles, de arriba hacia abajo, representan la orientación parametrizada por los ángulos de Euler ζ , ϕ y ψ respectivamente.	47
5.10	Comportamiento temporal del esfuerzo en cada junta del robot controlado para los parámetros K_P , K_D , D_{d2} y K_{d2} . La variable u_i representa el esfuerzo aplicado por la junta i , donde este subíndice toma valores de 1 a 6 para juntas consecutivas de la base hacia el efector final.	48
5.11	Comportamiento controlado del efector final en el espacio cartesiano para los parámetros K_P , K_D , D_{d3} y K_{d3} . El robot parte de la posición inicial (magenta), hacia la posición deseada (verde).	49
5.12	Comportamiento temporal del efector final en los ejes Cartesianos para el robot controlado para los parámetros K_P , K_D , D_{d3} y K_{d3} . Los paneles, de arriba hacia abajo, representan a x , y y z respectivamente.	50
5.13	Comportamiento temporal la orientación del robot controlado para los parámetros K_P , K_D , D_{d3} y K_{d3} . Los paneles, de arriba hacia abajo, representan la orientación parametrizada por los ángulos de Euler ζ , ϕ y ψ respectivamente.	51
5.14	Comportamiento temporal del esfuerzo en cada junta del robot controlado para los parámetros K_P , K_D , D_{d3} y K_{d3} . La variable u_i representa el esfuerzo aplicado por la junta i , donde este subíndice toma valores de 1 a 6 para juntas consecutivas de la base hacia el efector final.	52
5.15	Entorno de simulación con obstáculo.	53
5.16	De izquierda hacia derecha, capturas de la interacción del robot con el objeto de superficie irregular a través del tiempo. En (a) el comportamiento bajo el controlador de posición y orientación y en (b) el comportamiento bajo el controlador por impedancia. En ambos casos, las posiciones y orientaciones iniciales (en púrpura) y deseadas (en verde) son las mismas.	54

5.17	Comportamiento controlado del efector final en el espacio cartesiano para los parámetros K_P , K_D , D_{d1} y K_{d1} . El robot parte de la posición inicial (magenta), hacia la posición deseada (verde). Los planos (esmeralda) representan la posición de un objeto en el espacio.	55
5.18	Comportamiento temporal del efector final en los ejes Cartesianos para el robot controlado para los parámetros K_P , K_D , D_{d1} y K_{d1} , cuando este interactúa con un objeto en su trayectoria. Los paneles, de arriba hacia abajo, representan a x , y y z respectivamente.	56
5.19	Comportamiento temporal de la orientación del robot controlado para los parámetros K_P , K_D , D_{d1} y K_{d1} , cuando este interactúa con un objeto en su trayectoria. Los paneles, de arriba hacia abajo, representan la orientación parametrizada por los ángulos de Euler ζ , ϕ y ψ respectivamente. . .	57
5.20	Comportamiento temporal del esfuerzo en cada junta del robot controlado para los parámetros K_P , K_D , D_{d1} y K_{d1} , cuando este interactúa con un objeto en su trayectoria. La variable u_i representa el esfuerzo aplicado por la junta i , donde este subíndice toma valores de 1 a 6 para juntas consecutivas de la base hacia el efector final.	58
5.21	Comportamiento controlado del efector final en el espacio cartesiano para los parámetros K_P , K_D , D_{d2} y K_{d2} . El robot parte de la posición inicial (magenta), hacia la posición deseada (verde). Los planos (esmeralda) representan la posición de un objeto en el espacio.	59
5.22	Comportamiento temporal del efector final en los ejes Cartesianos para el robot controlado para los parámetros K_P , K_D , D_{d2} y K_{d2} , cuando este interactúa con un objeto en su trayectoria. Los paneles, de arriba hacia abajo, representan a x , y y z respectivamente.	60

5.23	Comportamiento temporal de la orientación del robot controlado para los parámetros K_P , K_D , D_{d2} y K_{d2} , cuando este interactúa con un objeto en su trayectoria. Los paneles, de arriba hacia abajo, representan la orientación parametrizada por los ángulos de Euler ζ , ϕ y ψ respectivamente.	61
5.24	Comportamiento temporal del esfuerzo en cada junta del robot controlado para los parámetros K_P , K_D , D_{d2} y K_{d2} , cuando este interactúa con un objeto en su trayectoria. La variable u_i representa el esfuerzo aplicado por la junta i , donde este subíndice toma valores de 1 a 6 para juntas consecutivas de la base hacia el efector final.	62
5.25	Comportamiento controlado del efector final en el espacio cartesiano para los parámetros K_P , K_D , D_{d3} y K_{d3} . El robot parte de la posición inicial (magenta), hacia la posición deseada (verde). Los planos (esmeralda) representan la posición de un objeto en el espacio.	63
5.26	Comportamiento temporal del efector final en los ejes Cartesianos para el robot controlado para los parámetros K_P , K_D , D_{d3} y K_{d3} . cuando este interactúa con un objeto en su trayectoria. Los paneles, de arriba hacia abajo, representan a x , y y z respectivamente.	64
5.27	Comportamiento temporal de la orientación del robot controlado para los parámetros K_P , K_D , D_{d3} y K_{d3} , cuando este interactúa con un objeto en su trayectoria. Los paneles, de arriba hacia abajo, representan la orientación parametrizada por los ángulos de Euler ζ , ϕ y ψ respectivamente.	65
5.28	Comportamiento temporal del esfuerzo en cada junta del robot controlado para los parámetros K_P , K_D , D_{d3} y K_{d3} , cuando este interactúa con un objeto en su trayectoria. La variable u_i representa el esfuerzo aplicado por la junta i , donde este subíndice toma valores de 1 a 6 para juntas consecutivas de la base hacia el efector final.	66

RESUMEN

Los manipuladores robóticos son incluidos en actividades que implican su interacción con objetos y seres vivos. Para asegurar el correcto desarrollo de tales actividades es necesario que el robot sea controlado tomando en cuenta restricciones cinemáticas y dinámicas específicas. Dependiendo del sistema de sensado, estas restricciones pueden ser definidas antes de iniciar alguna actividad o estimadas durante el trayecto del robot.

En este trabajo de tesis, se presenta el diseño de un sistema de control de impedancia para el robot UR5 sin el uso de sensores externos. Este sistema permite que el robot se desplace de posiciones y orientaciones iniciales, a deseadas específicas, admitiendo interacciones entre este y objetos de superficie irregular durante su trayecto. Para conseguir una interacción delicada, sin afectar la convergencia del robot hacia el estado deseado, se define un comportamiento dinámico en la descripción del controlador. Este comportamiento es definido como una impedancia mecánica compuesta por elementos lineales de rigidez, amortiguamiento e inercia.

El desempeño del controlador propuesto es evaluado a través de simulaciones dinámicas del robot UR5 desarrolladas en dos ambientes. Un ambiente permite el movimiento del robot sin restricciones, mientras que el otro presenta un objeto que interfiere en su trayectoria. Para ambos ambientes, se compara el desempeño del controlador por impedancia y un controlador de posición clásico. Los resultados muestran que, en un ambiente sin restricciones, ambas formulaciones convergen al estado deseado. Sin embargo, en casos donde el robot interactúa durante su trayectoria, el controlador por impedancia es el único que converge a la posición y orientación deseadas.

ABSTRACT

DESIGN AND SIMULATION OF AN IMPEDANCE CONTROLLER FOR ROBOTIC MANIPULATORS FOR INTERACTION WITH UNEVEN SURFACES

Robotic manipulators are included in tasks that imply interaction with objects and living beings. It is necessary to control the robot taking into account specific kinematic and dynamic constraints to ensure excellent performance in these activities. Depending on the sensing systems, these constraints could be set before the robot starts to perform any task, or estimated during its operation.

In this work, we present a control system design for the UR5 robot considering no exteroceptive force/torque sensing. This controller allows the robot to move from an initial pose to a desired one handling possible interactions of the robot's end-effector with uneven surface objects during its trajectory. A dynamic desired behavior is defined in the controller's synthesis to achieve a gentle interaction without affecting the convergence of the robot to the desired state. This desired behavior is chosen as a mechanical impedance conformed by linear elements of inertia, damping and stiffness.

The performance of the proposed impedance controller is tested by simulating dynamically the controlled UR5 robot in two environments. One of these allows the robot to move in space without external constraints, while the other has an object positioned in order to interfere with the robot's trajectory. For both environments, the impedance controller is evaluated by comparing it to the robot's behavior when it is controlled in position. The results show that, in an unconstrained environment, both formulations converge to the desired pose. However, in cases where the robot interacts during its trajectory, the behavior of the impedance controlled system clearly outperforms the position control.

Capítulo 1

INTRODUCCIÓN

Los robots manipuladores son fabricados para interactuar con su entorno. Desde el primer robot industrial [1], hasta manipuladores modernos desarrollados para operar en ámbitos industriales y de investigación [2, 3], la actividad de estos ha implicado desempeñar tareas tales como transporte de objetos, soldadura y ensamblado [4, 5]. Por un periodo prolongado de tiempo, el uso de controladores de posición clásicos fue suficiente para que robots manipuladores pudieran cumplir con sus tareas. Sin embargo, un creciente número de aplicaciones donde se incluyen manipuladores, tales como cirugía, pulido y pintado, precisan tomar en cuenta distintos requisitos adicionales que garanticen interacciones seguras y delicadas entre estos y las superficies que tocan [6, 7].

Una aplicación notable, implica el movimiento del robot manipulador en ambientes con objetos de naturaleza rígida o blanda. Donde, la superficie por la que el robot se desplaza puede ser considerada irregular, debido a desigualdades que se podrían encontrar en su forma. Tomando en cuenta estas restricciones, se han desarrollado distintas técnicas con la finalidad de que el robot llegue a una posición deseada. Técnicas enfocadas en eludir objetos intermedios, proponen la definición de trayectorias a seguir. Sin embargo, estas implican un conocimiento previo del posicionamiento y características geométricas de los objetos en el entorno [8]. Otros métodos se basan en la recuperación del robot después de colisionar con algún objeto, de modo que este pueda continuar hacia una posición deseada a pesar de perturbaciones de este tipo [9, 10]. También se hacen uso de sensores y cámaras para mapear el entorno del robot durante su traslación, de modo que se pueda recalcular la trayectoria de este, en caso existiese un objeto en su recorrido [11, 12].

En este trabajo de tesis, se presenta el desarrollo y simulación de un sistema de control, que permite abordar restricciones impuestas al robot manipulador en su interacción con una superficie irregular. Este se basa en la imposición de un comportamiento dinámico para el acople entre el efector final del robot manipulador, y la superficie en contacto. Debido a que las restricciones impuestas por la superficie son en posición, se define el comportamiento dinámico como una impedancia mecánica lineal. Así, en lugar de definir una trayectoria para que el manipulador llegue hasta una posición deseada, este interactuará con la superficie y la bordeará, regulando la fuerza que aplica, de acuerdo a la impedancia establecida.

Este documento está organizado en seis capítulos. En el capítulo dos, se describen herramientas computacionales y algoritmos usados en la estimación y control de robots manipuladores. En el capítulo tres, se describen brevemente consideraciones teóricas necesarias para el desarrollo de un controlador por impedancia, además, se describen herramientas computacionales usadas para estimaciones de comportamientos dinámicos del robot, y el entorno de simulación. En el capítulo cuatro se describen las consideraciones tomadas para la implementación del sistema de control en el robot UR5, así como la creación de distintos ambientes de prueba. En el capítulo cinco se presentan los resultados obtenidos a partir de simulaciones del robot manipulador controlado por impedancia. En el capítulo seis se discuten los resultados obtenidos y se presentan trabajos a futuro.

Capítulo 2

ANTECEDENTES

En esta sección se presentan algoritmos, librerías computacionales y *software* relacionados con el objetivo de este trabajo de tesis. De este modo, se describen brevemente algoritmos usados para abordar el problema de interacción de robots con superficies irregulares, herramientas comúnmente usadas para realizar estimaciones físicas y herramientas relacionadas con la simulación dinámica de robots.

2.1 Control de interacción para robots manipuladores

Los robots manipuladores son controlados tomando como referencia actividades que desarrollarán. Así, para el diseño de tales sistemas de control es necesario tomar en cuenta restricciones del robot tales como velocidades máximas y mínimas, rangos de alcance, saturación eléctrica de sus motores, entre otros. En actividades que implican la traslación del robot en entornos desconocidos, o en su interacción con objetos externos, es necesario además tomar en cuenta restricciones de este entorno, con la finalidad de asegurar una interacción segura. A continuación, se presentan algunos sistemas de control diseñados para robots manipuladores que interactúan directamente con su entorno, considerando situaciones donde el robot posee una realimentación externa, y situaciones donde no existe tal realimentación.

2.1.1 Sistemas con sensado externo

El principal objetivo de usar dispositivos externos tales como cámaras, LIDARs y sensores ultrasónicos, es generar trayectorias deseadas para el robot, a fin de evitar que

este colisione violentamente con su entorno [13]. La generación de tales trayectorias deseadas puede realizarse de forma *offline*, realizando mapeos previos del entorno en el que el manipulador desarrollará sus actividades, o de forma *online*, recalculando su trayectoria a través de realimentación recibida mientras este se mueve [14].

Para generación *offline* de trayectorias, diversos autores proponen el uso de dispositivos tales como cámaras y sensores ultrasónicos [15], de modo que el sistema detecte objetos y posibles colisiones previamente. En estos casos, debido a que existe un tiempo entre el cálculo de la trayectoria y el movimiento del robot, la necesidad principal es que este esté controlado para seguir la trayectoria de manera precisa [13]. En la generación *online* de trayectorias, los trabajos se centran en el uso de dispositivos tales como cámaras de video 3D [16] o sistemas de captura 3D [17], que le permiten al robot reconocer objetos en su trayecto, sin tener que recurrir a procesamientos costosos computacionalmente durante la trayectoria. Para trayectorias generadas *online*, el control del robot debe asegurar el seguimiento preciso del robot a tales trayectorias, además de permitir cambios en esta debido a posibles obstáculos [13].

2.1.2 Sistemas sin sensado externo

Los sistemas que carecen de realimentación por dispositivos externos, no pueden conocer su entorno *a priori*, por lo que no pueden evadir objetos existentes. Ante interacciones inevitables, diversos trabajos proponen asegurar que estas, entre el efector final del robot y tales objetos, sean controladas o razonablemente delicadas. Para conseguir este comportamiento, se propone el uso de sensores de fuerza y torque, que permiten al robot conocer cuándo interactúa con un objeto y cuál es la magnitud de tal interacción [14, 18].

Algunos trabajos se centran en controlar la posición del robot considerando un lazo externo de control que permite regular la fuerza que estos aplican [19, 20]. Otros se enfocan en la interacción delicada del robot con su entorno, sin la necesidad de definir

una magnitud de fuerza deseada específica. Este comportamiento se consigue definiendo restricciones dinámicas deseadas en sus controladores, las cuales permiten que el efector final se comporte como si existieran elementos externos entre este y el estado deseado. Los comportamientos dinámicos impuestos en estos casos, son con mayor frecuencia expresiones lineales y son representadas mecánicamente como impedancias [18, 21], complianzas [22, 23] y admitancias [24, 25]. De manera similar, algunos trabajos proponen el control del manipulador garantizando interacciones delicadas restringidas por comportamientos dinámicos, sin hacer uso de sensores fuerza o torque [26, 27].

2.2 Librerías para estimación de la dinámica

La dinámica de robots manipuladores puede ser calculada a partir de modelos analíticos. Sin embargo, para robots con un alto número de grados de libertad, el desarrollo de soluciones analíticas a sus modelos dinámicos representa una mayor complejidad. Para estos, es conveniente utilizar librerías, cuyos algoritmos recursivos permiten resolver numéricamente la dinámica en cada instante discreto de tiempo. A continuación, se enlistan y describen algunas de las librerías más usadas para estimaciones en modelos dinámicos de sistemas robóticos.

- La librería Metapod permite realizar computaciones eficientes de la dinámica de robots, bajo un entorno libre de modelos. Debido a esta característica, Metapod permite realizar estimaciones para una gran cantidad de robots [28]. Su funcionamiento se basa en tres algoritmos principales: *Composite Rigid Body Algorithm*, *Recursive Newton Euler Algorithm* y *Composite Hybrid Dynamics Algorithm* [28].
- El *Dynamic Animation Robotics Toolkit*, conocido por sus siglas DART, es una librería de código abierto para sistemas robóticos. La computación de DART permite hacer estimaciones dinámicas y cinemáticas para modelos de estos sistemas [29]. Esta librería permite redefinir los sistemas de referencia de las juntas de los robots,

dando una mayor flexibilidad en los cálculos [29]. DART fue creado en el 2011, por lo que es considerado nuevo con respecto a otras librerías que presentan características similares .

- La librería Pinocchio permite estimar numéricamente el comportamiento dinámico de sistemas multicuerpo. Entre sus herramientas se encuentran algoritmos que permiten hallar numéricamente la dinámica inversa, y las matrices de inercia y Coriolis que describen la dinámica de sistemas robóticos. Por otro lado, permite realizar estimaciones cinemáticas relacionadas con jacobianos, posiciones, velocidades y aceleraciones [30].
- El *Rigid-Body Dynamics Library*, de siglas RBDL, es una librería altamente eficiente que permite calcular numéricamente las dinámicas directa e inversa de sistemas robóticos. Sus algoritmos recursivos, además, permiten hallar cada término que compone la dinámica del robot. Complementariamente, permite realizar cálculos cinemáticos, tomando como referencia descripciones computacionales del robot [31].

2.3 Simuladores físicos

Los simuladores físicos son librerías computacionales o *software*, que permiten observar efectos físicos en objetos virtuales. Estos efectos se resumen en consecuencias de la gravedad, colisiones y dinámica de cuerpos. Debido a su funcionalidad, los simuladores físicos son usados en una gran variedad de aplicaciones, que incluyen videojuegos, simulaciones dinámicas y películas. A continuación, se presentan algunos de los simuladores físicos más utilizados en distintas áreas, con sus respectivas características más relevantes.

- Havok Physics es un simulador físico de código cerrado desarrollado por la empresa Havok, propiedad de Microsoft. Este se encuentra altamente optimizado y permite

estimar comportamientos dinámicos para cuerpos rígidos. Su enfoque principal son acciones que implican colisiones e interacción entre cuerpos. Además, es un simulador que recibe soporte para distintas plataformas. Debido a estas características, Havok es ampliamente usado en la industria de videojuegos, siendo un simulador usado por más de la mitad de los videojuegos más vendidos [32].

- *Bullet Physics*, conocido como *Bullet*, es un simulador físico de código abierto descrito como rápido y estable [33]. Este permite la detección de colisiones, así como la estimación del comportamiento dinámico de objetos rígidos y blandos. Permite la aplicación de restricciones a objetos definidos, lo que permite mostrar un mayor realismo en objetos como motores. Se encuentra optimizado para distintas plataformas y para su uso en plataformas de computación en paralelo. *Bullet* es ampliamente usado en simulaciones para videojuegos, efectos visuales en películas, robótica y aprendizaje automático [33].
- El *Open Dynamics Engine*, conocido por sus siglas ODE, es un simulador físico de código abierto que permite detectar colisiones y estimar comportamientos dinámicos de objetos rígidos. ODE permite crear objetos con distintas geometrías en distintos ambientes [34]. Por sus características, es usado en aplicaciones tales como videojuegos y robótica. ODE es el simulador físico configurado por defecto en distintos simuladores robóticos tales como Coppeliasim, Gazebo y Webots.
- El simulador *Multi-Joint dynamics with Contact*, conocido como MuJoCo, puede ser usado bajo una licencia paga y es desarrollado por Roboti LLC. Este permite realizar estimaciones físicas eficientes en ambientes virtuales donde pueden coexistir distintos elementos. Debido a sus características, MuJoCo es usado en áreas tales como biomecánica, robótica, animación 3D, entre otros [35].

2.4 Simuladores robóticos

Los simuladores robóticos permiten observar el comportamiento de robots reales en entornos virtuales. De esta forma, la ejecución de simulaciones facilita realizar pruebas previas a una implementación real. Los simuladores robóticos usan simuladores físicos con la finalidad de que los objetos en el ambiente de simulación se comporten bajo efectos dinámicos realistas. A continuación, se presentan descripciones de simuladores robóticos ampliamente usados.

- El *software* Klampt permite simular sistemas robóticos formados por cuerpos rígidos. Este se centra en el modelado, simulación y planeamiento de sistemas cuya actividad principal es la locomoción. Además, debido al uso de simuladores físicos, permite estimar la mecánica de contacto entre objetos. Las simulaciones de colisión se basan en enmallados triangulares, cuya construcción puede ser hecha por *software* de diseño asistido por computadora [36]. Klampt se limita a usar el simulador físico ODE para sus cálculos dinámicos.
- CoppeliaSim es un *software* multiplataforma que permite simular sistemas robóticos multicuerpo, así como objetos rígidos ubicados en un entorno [37]. Una característica principal de CoppeliaSim es su versatilidad, los nodos de control pueden ser escritos en C, C++, Python, Matlab y Lua. Además, soporta cuatro simuladores físicos distintos: Bullet, ODE, Vortex y Newton [37].
- Gazebo es un *software* que permite el diseño y simulación de sistemas robóticos. Además, permite crear entornos donde pueden coexistir distintos robots y objetos. Este simulador es ampliamente utilizado para aplicaciones industriales y de investigación. Gazebo tiene soporte para los simuladores físicos Bullet y ODE [38].

Capítulo 3

MARCO TEÓRICO

En esta sección se describen brevemente las herramientas teóricas y computacionales que se usarán para desarrollar el trabajo de tesis. En principio, se definen técnicas que permiten describir el posicionamiento de un robot en el espacio, así como relaciones existentes en sus velocidades. Se describe el modelo dinámico general de los robots manipuladores y se presenta una herramienta computacional que permitirá realizar estimaciones a partir de este. De igual forma se definen las leyes de control a implementar, y se presentan el entorno robótico y el entorno de simulación que serán usados.

3.1 Representaciones en el espacio cartesiano

Un punto en el espacio cartesiano tridimensional está descrito por su posición y orientación. La posición puede ser descrita con coordenadas cartesianas (x, y, z) o con coordenadas esféricas (r, θ, ρ) . La orientación puede describirse usando ángulos de Euler (ζ, ϕ, ψ) (considerando el orden de rotación ZYX), cuaterniones unitarios (ω, a_i, b_i, c_i) o una matriz de rotación R de dimensión 3×3 , con respecto a un sistema de referencia.

Considerando posiciones cartesianas y una matriz de rotación con componentes R_{ij} , estas se pueden representar con respecto a un sistema de referencia como

$$T = \begin{bmatrix} R_{11} & R_{12} & R_{13} & x \\ R_{21} & R_{22} & R_{23} & y \\ R_{31} & R_{32} & R_{33} & z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (3.1)$$

donde T se denomina matriz de transformación homogénea. De manera similar, la posición y la orientación se pueden representar en forma vectorial como

$$\vec{p} = (x, y, z, \zeta, \phi, \psi). \quad (3.2)$$

En robótica de manipuladores, el vector constituido por posiciones y orientaciones también es denominado pose[14].

Las representaciones espaciales (3.1) y (3.2) son equivalentes, dado que la matriz R puede parametrizarse en ángulos de Euler (ζ, ϕ, ψ) haciendo uso de transformaciones conocidas [13, 14].

3.2 Modelamiento cinemático

La representación cinemática de un robot serial de n grados de libertad es mostrada en la Figura 3.1. La posición angular de cada una de sus juntas se define en un vector $\vec{q} = (q_1, q_2, q_3, \dots, q_n)$, mientras que la posición y la orientación del efector final, con respecto a la base del robot, se definen como $\vec{p} = (x, y, z, \zeta, \phi, \psi)$. Los modelos cinemáticos permiten hallar relaciones geométricas entre \vec{p} y \vec{q} sin considerar efectos generados por alguna fuerza o torque [39].

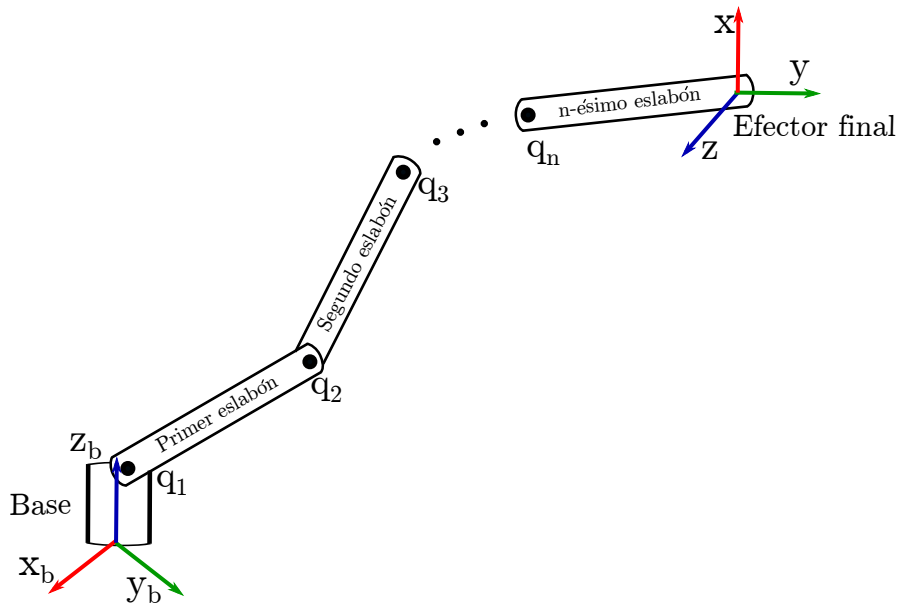


FIGURA 3.1: Cadena cinemática de n juntas. La posición cartesiana de los sistemas de referencia de la base y el efector final están descritas por (x_b, y_b, z_b) y (x, y, z) respectivamente.

3.2.1 Cinemática directa

La cinemática directa permite definir posiciones y orientaciones \vec{p} del efector final, en función de las posiciones articulares \vec{q} del robot [39]. Para la cadena cinemática mostrada en la Figura 3.1, esta relación se define como

$$\vec{p} = f(\vec{q}), \quad (3.3)$$

donde f es una función no lineal [39]. La cinemática directa en robots manipuladores puede desarrollarse a partir de cálculos geométricos o usando la convención de Denavit Hartenberg [39].

3.2.2 Cinemática inversa

La cinemática inversa es recíproca a la cinemática directa. Esta permite conocer la configuración articular requerida \vec{q} , para que el efector final del robot alcance posiciones y orientaciones \vec{p} específicas [40]. Así, en relación con (3.3), la cinemática inversa se define como

$$\vec{q} = f^{-1}(\vec{p}). \quad (3.4)$$

Debido a la naturaleza de f , su inversión puede producir que \vec{q} tenga de cero a infinitas soluciones [40]. La cinemática inversa de robots manipuladores puede desarrollarse usando métodos geométricos. Sin embargo, cuando el robot tiene un alto número de grados de libertad, usualmente mayor a tres, resulta conveniente la solución a partir de métodos numéricos [40].

3.2.3 Jacobiano analítico

La relación entre velocidades articulares $\dot{\vec{q}}$ y velocidades en el espacio cartesiano $\dot{\vec{p}}$, para un robot de n grados de libertad, puede definirse como

$$\dot{\vec{p}} = J_A \dot{\vec{q}}, \quad (3.5)$$

donde J_A es una matriz $6 \times n$ denominada Jacobiano analítico [14]. Considerando las velocidades en (3.5) como variaciones instantáneas individuales de \vec{p} con respecto a \vec{q} , el Jacobiano analítico puede ser descrito de la forma

$$J_A = \begin{bmatrix} \frac{\partial x}{\partial q_1} & \frac{\partial x}{\partial q_2} & \dots & \frac{\partial x}{\partial q_n} \\ \frac{\partial y}{\partial q_1} & \frac{\partial y}{\partial q_2} & \dots & \frac{\partial y}{\partial q_n} \\ \frac{\partial z}{\partial q_1} & \frac{\partial z}{\partial q_2} & \dots & \frac{\partial z}{\partial q_n} \\ \frac{\partial \zeta}{\partial q_1} & \frac{\partial \zeta}{\partial q_2} & \dots & \frac{\partial \zeta}{\partial q_n} \\ \frac{\partial \phi}{\partial q_1} & \frac{\partial \phi}{\partial q_2} & \dots & \frac{\partial \phi}{\partial q_n} \\ \frac{\partial \psi}{\partial q_1} & \frac{\partial \psi}{\partial q_2} & \dots & \frac{\partial \psi}{\partial q_n} \end{bmatrix}.$$

3.3 Modelamiento dinámico

A diferencia de modelos puramente cinemáticos, los modelos dinámicos sí incluyen efectos generados por inercias, así como fuerzas y torques externos. De este modo, para un sistema modelado dinámicamente, es posible estimar su comportamiento tomando en cuenta parámetros tales como la inercia de cada elemento, gravedad, fuerzas de fricción, colisiones, perturbaciones externas, entre otros [14].

3.3.1 Modelo general de robots manipuladores

En el espacio articular, el modelo dinámico para robots manipuladores de n grados de libertad se puede definir como

$$H(\vec{q})\ddot{\vec{q}} + C(\vec{q}, \dot{\vec{q}})\dot{\vec{q}} + \vec{g}(\vec{q}) = \vec{u}, \quad (3.6)$$

donde \vec{q} es el vector $n \times 1$ de desplazamientos articulares, \vec{u} es el vector $n \times 1$ de torques aplicados a cada junta, $H(\vec{q})$ es una matriz simétrica positiva $n \times n$ que representa la inercia, $C(\vec{q}, \dot{\vec{q}})\dot{\vec{q}}$ es un vector $n \times 1$ de torques generados por fuerzas centripeta y de Coriolis, y $\vec{g}(\vec{q})$ es el vector $n \times 1$ de fuerzas gravitacionales [39, 41].

El sistema definido en (3.6) puede ser reescrito como

$$H(\vec{q})\ddot{\vec{q}} + \vec{n}(\vec{q}, \dot{\vec{q}}) = \vec{u}, \quad (3.7)$$

donde $\vec{n}(\vec{q}, \dot{\vec{q}}) = C(\vec{q}, \dot{\vec{q}})\dot{\vec{q}} + \vec{g}(\vec{q})$ es conocido como el vector de efectos no lineales [14, 31].

Para tareas donde el objetivo es analizar y controlar el robot manipulador en referencia al comportamiento dinámico de su efector final, se usa la formulación en el espacio operacional [42]. A partir de la relación definida en (3.5), se puede redefinir (3.6) como

$$H_x(\vec{q})\ddot{\vec{p}} + C_x(\vec{q}, \dot{\vec{q}})\dot{\vec{p}} + \vec{g}_x(\vec{q}) = J_A^{-T}\vec{u}, \quad (3.8)$$

donde los parámetros de inercia, coriolis y de fuerzas gravitacionales, son parametrizados respectivamente en el espacio cartesiano como

$$H_x(\vec{q}) = J_A(\vec{q})^{-T} H(\vec{q}) J_A(\vec{q})^{-1},$$

$$C_x(\vec{q}, \dot{\vec{q}}) = J_A(\vec{q})^{-T} C(\vec{q}, \dot{\vec{q}}) J_A(\vec{q})^{-1} - H_x(\vec{q}) \dot{J}_A(\vec{q}) J_A(\vec{q})^{-1}$$

y

$$\vec{g}_x(\vec{q}) = J_A(\vec{q})^{-T} g(\vec{q}).$$

3.4 Sistema de control

Los sistemas de control son diseñados con la finalidad de llevar a sistemas dinámicos de un estado inicial a uno deseado a partir de la aplicación de una señal de control \vec{u} . El diseño del sistema de control para un robot puede depender de características tales como el grado de incerteza en su modelamiento, restricciones impuestas por el entorno, o comportamientos específicos deseados [41].

Los controladores para robots manipuladores son diseñados de forma que se pueda lidiar con no linealidades propias del sistema [41]. En su forma más ideal, para los espacios articular y operacional, se puede conseguir el control del sistema asegurando con exactitud la convergencia de este al estado deseado, partiendo de la cancelación de los términos no lineales en los sistemas definidos por (3.7) y (3.8), respectivamente. Esta metodología es denominada control por dinámica inversa, que sin embargo requiere de la estimación exacta del modelo del robot [14].

Aplicaciones tales como el pulido de vidrios, o cortes en cirugía, requieren de una interacción delicada del robot con su entorno. Para tener éxito en este tipo de tareas, es necesario garantizar estabilidad en el acople entre el robot y su entorno [18]. El control por impedancia permite abordar este requerimiento a través de la imposición de un comportamiento dinámico específico. Esta imposición tiene como objetivo generar una reacción dinámica por fuerza, ante fuerzas de movimiento percibidas por el robot [18, 43].

3.5 Robot Operating System

El *Robot Operating System*, conocido por sus siglas ROS, es un entorno que permite trabajar de forma indiferente con *software* y *hardware* de sistemas robóticos [44]. Su funcionamiento se basa en la reducción de complejidad de estos sistemas a través de su fragmentación en módulos. Estos módulos pueden ser reutilizados para tareas con un fin diferente, haciendo variaciones en la estructura de la transmisión y recepción de datos [44].

3.5.1 Elementos principales de ROS

Un nodo de ROS es un archivo independiente que ejecuta un proceso definido. Un nodo puede ser programado de forma que este pueda enviar y recibir datos. Diversos nodos pueden ser conectados entre sí haciendo uso de esquemas convenientes de transmisión de

datos. Los nodos se pueden interpretar como subpartes de aplicaciones robóticas. Así, por ejemplo, un nodo puede representar el control de la rueda de un robot móvil, mientras que otro, el procesamiento de imágenes recibidas a través de una cámara. Los nodos pueden ser reutilizados para aplicaciones distintas.

Los tópicos de ROS son el medio por el cuál los nodos envían y reciben datos. Un tópico es unidireccional y define el tipo de dato que se transmitirá. Para una aplicación robótica definida, los nodos son configurados para enviar o recibir cierto tipo de datos, a través de tópicos específicos.

En ROS, los publicadores y suscriptores son las unidades principales de transmisión de datos entre nodos. La definición de un publicador en un nodo implica el envío de datos por parte de este nodo. De modo complementario, la definición de un suscriptor implica la recepción de datos por este nodo. Para ambos casos, los tipos de datos deben ser congruentes con los tipos de datos definidos en los tópicos mediante los cuales se transmitirá el mensaje.

En la Figura 3.2, se muestra la representación gráfica de un flujo de trabajo entre los elementos principales de ROS.

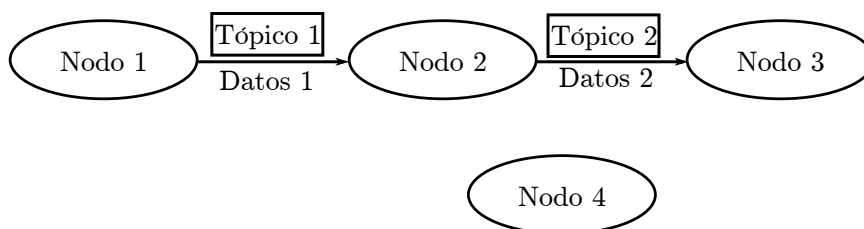


FIGURA 3.2: Diagrama de elementos principales de ROS interconectados. El flujo de datos va del Nodo 1 hacia el Nodo 2, y del Nodo 2 hacia el Nodo 3. El Nodo 4 es definido, pero no intercambia información con alguno de los otros nodos.

3.5.2 Gazebo

Gazebo es un simulador de código abierto orientado al estudio de sistemas robóticos comprendidos por mecanismos actuados y sensores [45]. Este permite estimar el comportamiento de sistemas ubicados en entornos tridimensionales. Los entornos y sistemas robóticos pueden ser construidos desde cero, o importados desde librerías compatibles con el *software* [38]. En la Figura 3.3 se muestran distintas escenas ilustrativas de un ambiente en Gazebo.

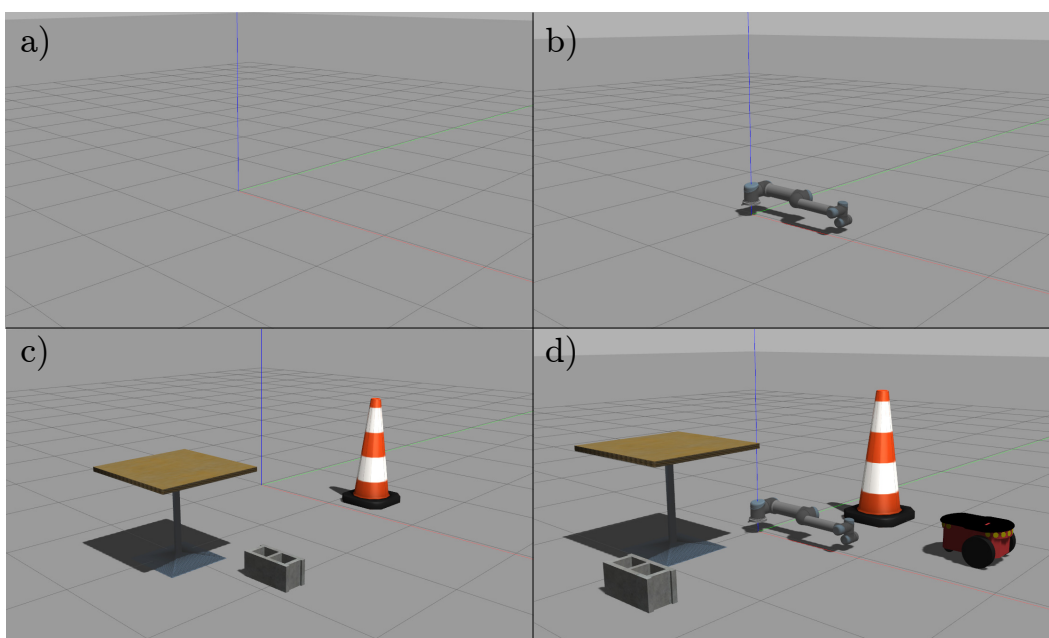


FIGURA 3.3: Ambiente de trabajo de Gazebo a) vacío, b) con un robot UR5, c) con objetos, y d) con un robot UR5 y objetos.

Para estimar el comportamiento dinámico de sistemas y objetos en un ambiente, Gazebo permite el uso de distintos simuladores físicos. Los comportamientos dinámicos simulados incluyen colisiones y estímulos externos. Debido a estas características, Gazebo permite probar algoritmos en sistemas robóticos, de una forma rápida y eficiente.

3.5.3 *Rigid-Body Dynamics Library*

El *Rigid-Body Dynamics Library* (RBDL) es un paquete computacional que permite calcular numéricamente soluciones a las ecuaciones dinámicas de manipuladores robóticos. La computación de RBDL se basa en una implementación eficiente del algoritmo recursivo de Newton-Euler [31]. La formulación del algoritmo de Newton-Euler describe el balance de fuerzas y torques que afectan al sistema. Así, este algoritmo permite hallar los torques requeridos en cada articulación del robot, para que estas actúen a aceleraciones específicas.

El paquete RBDL puede ser integrado a ROS. Así, resultados numéricos obtenidos a partir de RBDL pueden ser enviados y recibidos desde nodos en esquemas definidos para aplicaciones en sistemas robóticos.

3.6 Robot UR5

El robot UR5 tiene seis grados de libertad y es construido y distribuido por la marca danesa Universal Robots. Este robot es usado principalmente en tareas colaborativas que implican la carga de artículos livianos [3]. Una figura del robot UR5 es mostrada en la Figura 3.4, y sus respectivas características se muestran en la Tabla 3.1.

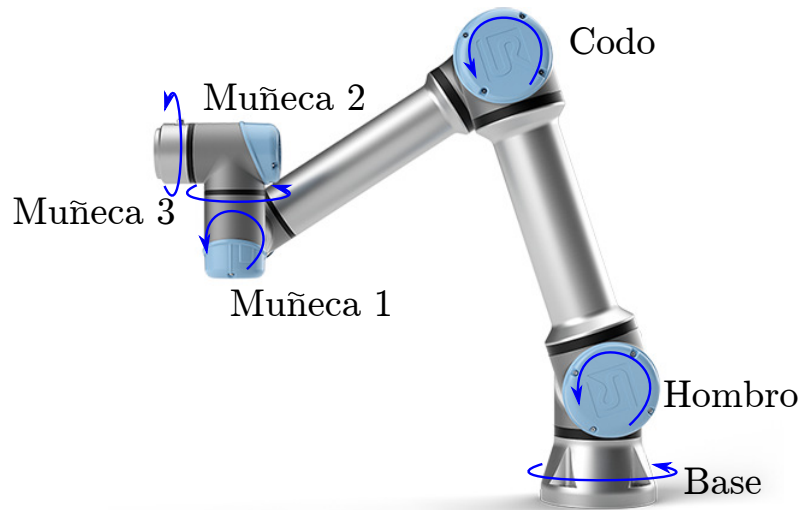


FIGURA 3.4: Robot UR5 de seis grados de libertad. Imagen modificada a partir de [46].

Especificaciones Generales		
Carga útil (Kg)	5	
Alcance (mm)	850	
Grados de libertad	6 articulaciones giratorias	
Movimiento		
	Radio de Acción (rad)	Velocidad Máxima (rad/s)
Base	$\pm 2\pi$	π
Hombro	$\pm 2\pi$	π
Codo	$\pm 2\pi$	π
Muñeca 1	$\pm 2\pi$	π
Muñeca 2	$\pm 2\pi$	π
Muñeca 3	$\pm 2\pi$	π
Dinámica		
	Masa (Kg)	Centro de masa
Eslabón 1	3.7	(0, -0.02561, 0.00193)
Eslabón 2	8.393	(0.2125, 0, 0.11336)
Eslabón 3	2.33	(0.15, 0.0, 0.0265)
Eslabón 4	1.219	(0, -0.0018, 0.01634)
Eslabón 5	1.219	(0, 0.0018, 0.01634)
Eslabón 6	0.1879	(0, 0, -0.001159)

TABLA 3.1: Características del robot UR5 [3].

Capítulo 4

MÉTODOLOGÍA

En esta sección se detalla la metodología seguida para controlar el robot UR5 de seis grados de libertad en un ambiente de Gazebo. La simulación es estructurada para ejecutarse haciendo uso de ROS. El sistema de control es diseñado de modo que el robot pueda movilizarse de una posición inicial a una posición final, interactuando con una o más superficies irregulares. Para este fin, se presenta la implementación de dos sistemas de control distintos: un controlador de posición por dinámica inversa y un controlador de impedancia.

4.1 Configuración en el entorno ROS

El simulador Gazebo se encuentra vinculado a ROS, de modo que entre estos existe un intercambio de datos. Así, estados físicos del robot simulado pueden transmitirse hacia ROS, permitiendo ejecutar procesamientos determinados. De forma similar, ROS puede enviar datos relevantes hacia Gazebo, de modo que el robot sea comandado hacia estados deseados. Todos los nodos son escritos en Python, bajo el formato definido para nodos de ROS.

4.1.1 Señales de control

El robot UR5 solo puede ser comandado por señales de velocidad o posición articulares, debido a configuraciones internas establecidas por el fabricante. Esta limitación se extiende al robot simulado, por lo que el envío directo de torques desde el sistema de control hacia el robot simulado no está permitido. Debido a esto, se propone hacer una

transformación conveniente entre señales de control por torque, a señales de control por posición. El esquema de transformación se ilustra en el diagrama mostrado en la Figura 4.1. En este, el sistema de control genera torques de actuación \vec{u} que actúan en un modelo estimado del robot, tras esto, se estima numéricamente la respuesta en posiciones articulares ante una actuación por torque. Estas posiciones estimadas servirán como señales de control para el robot simulado.

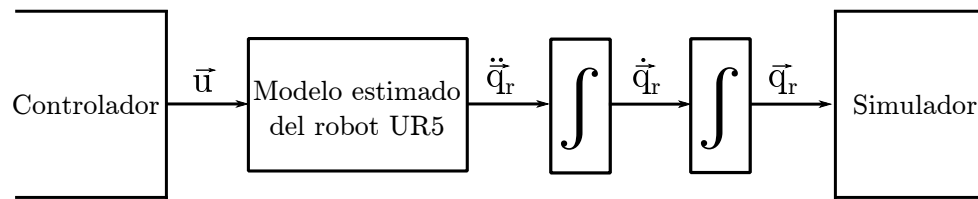


FIGURA 4.1: Esquema de transformación para la señal de control. La señal \vec{u} , definida por el controlador, ingresa a un modelo estimado del robot. El comportamiento dinámico del sistema controlado es descrito por la aceleración $\ddot{\vec{q}}_r$, que se integra para hallar la velocidad $\dot{\vec{q}}_r$ y la posición \vec{q}_r .

4.1.2 Flujo de datos

El robot UR5 simulado en Gazebo es controlado a través del entorno ROS. Con este fin, es necesario definir nodos y tópicos que permitan un flujo de datos adecuado. La información relevante que debe enviar Gazebo son las posiciones y velocidades articulares del robot. Así, se pueden realizar estimaciones dinámicas y cinemáticas que permitirán la ejecución de los controladores. Por otro lado, el nodo del controlador debe hacer el envío de posiciones articulares de referencia, cuyo fin es actuar como señales de control en el robot simulado. El flujo general de datos se muestra en la Figura 4.2

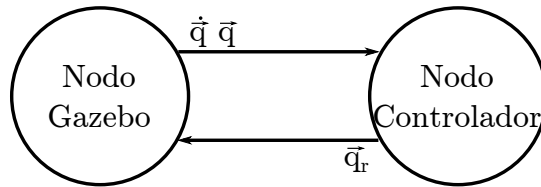


FIGURA 4.2: Flujo general de transmisión de datos entre el simulador Gazebo y el nodo de control definido en ROS. $\dot{\vec{q}}$ y \vec{q} representan las posiciones articulares actuales del robot simulado, mientras que \vec{q}_r representa la posición de referencia enviada desde el nodo de control.

4.2 Acondicionamiento del simulador Gazebo

El uso de Gazebo como simulador permite conocer el comportamiento dinámico del robot. Los ambientes en Gazebo son construidos a partir de un espacio vacío. Así, es necesario definir el modelo mecánico del robot UR5 y los objetos con los que este interactuará. Para que la interacción entre robot y los objetos sea realista, es necesario habilitar la física de colisiones en la configuración del modelo computacional del robot.

4.2.1 Modelo URDF del robot UR5

Gazebo escribe y lee modelos mecánicos y ambientes, en archivos de formato SDF. Sin embargo, la interfaz entre ROS y Gazebo, permite que este último pueda leer modelos mecánicos a partir de archivos en formato URDF. Los archivos URDF son escritos en lenguaje XML y contienen información mecánica específica del robot.

El modelo URDF del robot UR5 es desarrollado y proporcionado por ROS Industrial. Este contiene características específicas de inercia, longitud, centros de gravedad y geometrías genéricas del robot. Además, el archivo URDF contiene el enlace a la descripción mecánica del robot UR5 en un enmallado generado por diseño asistido por computadora. El grado de detalle del enmallado determina que la apariencia y la mecánica de colisiones del robot simulado sean similares a las del robot real.

4.2.2 Creación del ambiente de prueba

Para poder ejecutar una simulación en gazebo, es necesario construir un ambiente que contenga todos los elementos a simular. Dependiendo de las necesidades en la simulación, el ambiente puede tener únicamente al robot, más de un robot, o puede tener además objetos rígidos.

Los ambientes para realizar simulaciones con el robot UR5, deben contener al robot y objetos que emulen superficies irregulares. Así, el robot y los objetos se encontrarán localizados en posiciones fijas, y se comandará el robot de modo que durante su trayectoria interactúe con estos objetos.

4.2.3 Habilitación de física de colisiones

El modelo del robot UR5 en formato URDF, presenta una configuración interna de control en posición para simulaciones en Gazebo. Debido a esto, las simulaciones del robot no muestran un comportamiento apegado a la realidad. Cuando el robot realiza actividades ilícitas, tales como colisiones, este se inestabiliza, presenta fricción nula y muestra un comportamiento incongruente. La solución propuesta ante este problema se basa en reescribir el modelo URDF del robot y controladores internos asociados. Con la modificación se busca que el control interno para el robot sea de esfuerzo en lugar que de posición. Este cambio requiere establecer controladores individuales internos para cada junta, de modo que cada junta tenga un comportamiento dinámico realista.

4.3 Implementación del sistema de control

Para la implementación del sistema de control y la simulación del robot controlado, es necesario definir módulos que intercambien datos con el simulador. El desarrollo

de estos parte de la descripción de relaciones cinemáticas que permiten conocer el posicionamiento del robot, así como herramientas computacionales que permiten la estimación de su comportamiento dinámico. Considerando estos elementos, será posible definir el sistema de control, así como esquemas que permitan el envío de las señales de control adecuadas al robot simulado. Por otro lado, para la implementación de un sistema sin interrupciones generadas por divergencias en su formulación matemática, se definen técnicas que permiten abordar fallas específicas en cálculos cinemáticos y dinámicos.

4.3.1 Análisis cinemático

La cinemática directa del robot UR5 es hallada usando la convención de Denavit Hartenberg. Esta se basa en el elección de sistemas de referencia para cada junta del robot, como se muestra en la Figura 4.3.

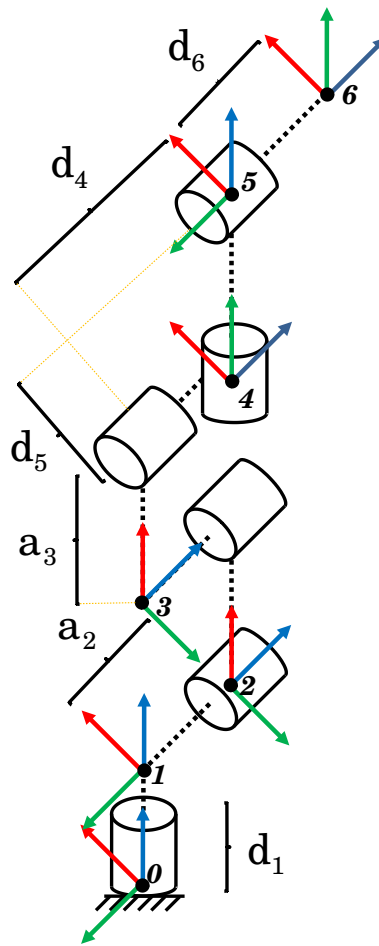


FIGURA 4.3: Sistemas de referencia definidos para el robot UR5. Los vectores rojo, verde y azul, representan los ejes x , y y z , respectivamente.

A partir de los sistemas de referencia definidos, se asignan cuatro parámetros a cada eslabón. Para el i -ésimo eslabón, los parámetros de Denavit Hartenberg son la longitud del eslabón r_i , el ángulo de rotación del eslabón α_i , el desplazamiento articular de la articulación q_i , y el desplazamiento de la articulación d_i . Los parámetros de Denavit Hartenberg para el robot UR5 se resumen en la Tabla 4.1.

Eslabón	q_i	d_i	r_i	α_i
1	q_1	0.0892	0	$\pi/2$
2	q_2	0	0.425	0
3	q_3	0	0.392	0
4	q_4	0.1093	0	$\pi/2$
5	q_5	0.09475	0	$\pi/2$
6	q_6	0.0825	0	0

TABLA 4.1: Parámetros de Denavit Hartenberg para el robot UR5.

Usando los parámetros mostrados en la Tabla 4.1, se definen matrices de transformación que describen la relación entre los sistemas de referencia $\{i-1\}$ e $\{i\}$ como

$${}^{i-1}T_i = \begin{bmatrix} cq_i & -c\alpha_i sq_i & s\alpha_i sq_i & r_i cq_i \\ sq_i & c\alpha_i cq_i & -s\alpha_i cq_i & r_i sq_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (4.1)$$

donde c y s son las funciones coseno y seno respectivamente. La matriz definida en (4.1) presenta la estructura de (3.1), compuesta por una matriz de rotación R de dimensión 3×3 , y un vector de posiciones cartesianas (x, y, z) .

Usando matrices definidas por (4.1), se obtiene la relación efector final con respecto a la base del robot, resolviendo

$${}^0T_{ef} = \prod_{i=0}^6 {}^{i-1}T_i. \quad (4.2)$$

La orientación del efector final se redefine con ángulos de Euler, transformando la matriz rotación R para (4.1) a un vector (ζ, ϕ, ψ) . Finalmente, el vector de posiciones y orientaciones se define como $\vec{p} = (x, y, z, \zeta, \phi, \psi)$.

Debido a la dimensión del robot UR5, la cinemática inversa se calcula haciendo uso de métodos numéricos. Considerando pequeños espacios de tiempo, la variación de las posiciones articulares \vec{q} se define como

$$\vec{q}' := \vec{q} + \Delta\vec{q}. \quad (4.3)$$

La variación $\Delta\vec{q}$ mostrada en (4.3), se puede hallar a partir de la definición del jacobiano analítico (3.5). Para pequeños espacios de tiempo, el jacobiano puede definirse como

$$\Delta\vec{p} = J_A \Delta\vec{q}. \quad (4.4)$$

Por lo tanto, la variación en \vec{q} con respecto a la diferencia entre la posición cartesiana actual y una posición deseada, se define como

$$\Delta\vec{q} = J_A^{-1} \Delta\vec{p},$$

que, para evitar singularidades puede redefinirse como

$$\Delta\vec{q} = J^T \Delta\vec{p}. \quad (4.5)$$

El uso de la transpuesta en lugar de la inversa garantiza resultados similares en el cálculo de la cinemática inversa [40].

4.3.2 Análisis dinámico

El comportamiento dinámico del robot UR5 es estimado haciendo uso del paquete computacional RBDL. Este permite estimar los términos de inercia, y de efectos no lineales, mostrados en el modelo general indicado en (3.7), para un torque específico y viceversa.

Como ejemplo, se utiliza el paquete RBDL para estimar parámetros de la dinámica del robot UR5 no forzado, en un instante de tiempo donde sus posiciones y velocidades articulares son $\vec{q} = (0.05, 0.2, 0.3, 0.08, 0.5, 0.6)$ y $\dot{\vec{q}} = (0.01, 0.2, 0.03, 0.1, 0.3, 0.02)$, respectivamente.

Teniendo posiciones y velocidades conocidas, se resuelve para obtener los términos de inercia y de efectos no lineales para el modelo (3.7) reescrito como

$$H(\vec{q})\ddot{\vec{q}} + \vec{n}(\vec{q}, \dot{\vec{q}}) = \vec{0}.$$

Los resultados de la computación usando RBDL son la matriz de inercia estimada

$$\hat{H}(\vec{q}) = \begin{bmatrix} 3.89 & 0.12 & 0.04 & 0 & -0.2 & 0 \\ 0.12 & 3.9 & 1.48 & 0.24 & 0 & 0.02 \\ 0.04 & 1.48 & 0.84 & 0.24 & 0 & 0.02 \\ 0 & 0.24 & 0.24 & 0.24 & 0 & 0.01 \\ -0.2 & 0 & 0 & 0 & 0.25 & 0 \\ 0 & 0.02 & 0.02 & 0.01 & 0 & 0.04 \end{bmatrix}$$

y el vector estimado de efectos no lineales

$$\hat{\vec{n}}(\vec{q}, \dot{\vec{q}}) = \begin{bmatrix} 0.03 \\ -56.29 \\ -13.66 \\ 0.1 \\ 0 \\ 0.01 \end{bmatrix}.$$

Nótese que, debido a que en este caso el vector de velocidades $\dot{\vec{q}}$ no es nulo, el vector de efectos no lineales se compone del término de Coriolis $\hat{C}(\vec{q}, \dot{\vec{q}})\dot{\vec{q}}$ y el vector de

efectos gravitacionales $\hat{g}(\vec{q})$. Estos términos podrían ser estimados individualmente, sin embargo, el uso del término compuesto $\hat{n}(\vec{q}, \dot{\vec{q}})$ no afectará en la definición del controlador.

Para la computación con RBDL es necesario el uso de posiciones articulares \vec{q} . Así, debido a que en el sistema de control a definir, se hace uso de posiciones cartesianas para tareas relacionadas con el espacio operacional, es necesario hacer la transformación necesaria a través de la cinemática inversa, mostrada en (4.5).

De manera similar, los parámetros de inercia $\hat{H}(\vec{q})$, Coriolis $\hat{C}(\vec{q}, \dot{\vec{q}})\dot{\vec{q}}$ y de efectos gravitacionales $\hat{g}(\vec{q})$ del robot UR5, estimados haciendo uso de RBDL, se encuentran definidos en el espacio articular. Para definir estos parámetros en el espacio operacional, siguiendo la formulación de (3.8), es necesario calcular el jacobiano analítico. Este, definido en (3.5), puede ser estimado numéricamente usando la relación

$$\frac{\partial \vec{p}}{\partial q_\gamma} \approx \frac{\vec{p}(\vec{q} + \delta q_\gamma) - \vec{p}(\vec{q})}{\delta q_\gamma},$$

para $\gamma \in [1, 6]$, donde $\vec{p}(\vec{q})$ es el vector de posiciones y orientaciones en el espacio operacional, hallado por la cinemática directa, y δq_γ representa una variación pequeña en los valores articulares.

Para configuraciones articulares específicas, es posible obtener una matriz singular como jacobiano, implicando la inexistencia de su inversa. Para abordar este problema, se define la inversa por mínimos cuadrados amortiguados

$$J_A^\ominus = J_A^T (J_A J_A^T + \lambda^2 I)^{-1}, \quad (4.6)$$

donde λ es un número real pequeño, e I es una matriz identidad 6×6 . Debido a los errores adicionales que puede generar el cálculo de (4.6), este se computa únicamente cuando el rango del jacobiano es menor a seis.

4.3.3 Control de posición por dinámica inversa

En esta formulación, el objetivo es controlar el manipulador de modo que el efector final pueda trasladarse de una posición articular inicial hacia una posición articular deseada. En ese sentido, se define

$$\vec{u} = \hat{H}(\vec{q})\vec{y} + \hat{n}(\vec{q}, \dot{\vec{q}}), \quad (4.7)$$

donde $\hat{H}(\vec{q})$ y $\hat{n}(\vec{q}, \dot{\vec{q}})$ representan la matriz de inercia y el vector de efectos no lineales, respectivamente, estimados con exactitud. Si se reemplaza (4.7) en (3.7), se obtiene

$$H(\vec{q})(\ddot{\vec{q}} - \vec{y}) = 0,$$

que se puede resolver como

$$\ddot{\vec{q}} = \vec{y}. \quad (4.8)$$

La expresión mostrada en (4.8) puede interpretarse como un sistema desacoplado, controlado en aceleración por la entrada \vec{y} . Si para una trayectoria deseada \vec{q}_d , se define el vector \vec{y} como

$$\vec{y} = \ddot{\vec{q}}_d + K_P\vec{e}_q + K_D\dot{\vec{e}}_q, \quad (4.9)$$

con $\vec{e} = \vec{q}_d - \vec{q}$, y matrices diagonales K_P y K_D . Para valores adecuados de K_P y K_D se puede garantizar seguimiento de los desplazamientos \vec{q} a la trayectoria deseada \vec{q}_d . Reemplazando (4.9) en (4.8), se define la dinámica del error del seguimiento de trayectoria como

$$\ddot{\vec{e}}_q + K_D\dot{\vec{e}}_q + K_P\vec{e}_q = 0. \quad (4.10)$$

En el espacio operacional, el objetivo es conseguir que la posición y orientación \vec{p} del efector final, sigan a una trayectoria deseada \vec{p}_d [42]. Para el sistema mostrado en

(3.8), si se cancelan los términos no lineales, se obtiene

$$J_A(\vec{q})\ddot{\vec{q}} + \dot{J}_A(\vec{q}, \dot{\vec{q}})\dot{\vec{q}} = \vec{y}, \quad (4.11)$$

cuya definición es análoga a (4.8).

De manera similar a (4.9), se define

$$\vec{y} = J_A^{-1}(\vec{q})(\ddot{\vec{p}}_d + K_P\vec{e}_p + K_D\dot{\vec{e}}_p - \dot{J}_A(\vec{q}, \dot{\vec{q}})\dot{\vec{q}}), \quad (4.12)$$

con $\vec{e}_p = \vec{p}_d - \vec{p}$ y matrices diagonales K_P y K_D . Reemplazando (4.12) en (4.11) se define la dinámica del error en espacio operacional \vec{e}_p como

$$\ddot{\vec{e}}_p + K_D\dot{\vec{e}}_p + K_P\vec{e}_p = 0. \quad (4.13)$$

La elección de $K_D = 2\sqrt{K_P}$ para los sistemas de segundo orden mostrados en (4.10) y (4.13), garantiza que el comportamiento de estos será críticamente amortiguado. Por lo tanto, se garantiza la convergencia exponencial de \vec{q} a \vec{q}_d y de \vec{p} a \vec{p}_d [14].

Nótese que el sistema de control descrito en esta subsección, adquiere la denominación de controlador de posición debido a que se parte de la definición en el espacio articular, donde se busca controlar las posiciones articulares. Sin embargo, cuando se realiza la derivación del controlador en el espacio operacional, este permite controlar el sistema hacia posiciones y orientaciones deseadas de su efector final. Esta observación es importante, debido a que la implementación será realizada tomando posiciones y orientaciones en el espacio operacional, siguiendo la formulación de (4.12).

4.3.4 Control de impedancia

Si se considera una masa que se mueve a través del eje x , afectada por una fuerza externa F_{ext} y actuada por un estímulo controlado u , como se muestra en la Figura 4.4. Sin considerar fricción con el suelo, este sistema puede ser modelado como

$$m\ddot{x} = u - F_{ext}, \quad (4.14)$$

donde \ddot{x} es la aceleración del sistema a través del eje x .

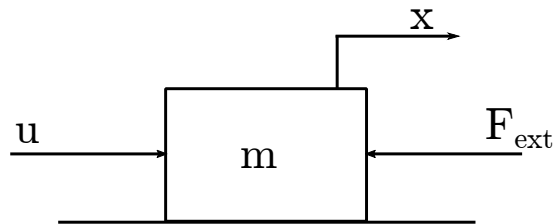


FIGURA 4.4: Sistema dinámico en una dimensión. La masa m se mueve en dirección del eje x , y es afectada por el estímulo u y la fuerza F_{ext} .

Además, se define el sistema mostrado en la Figura 4.5, cuyo comportamiento representa el acople entre una masa m y una posición variable x_d . El sistema de acople consta de un amortiguador y un resorte lineales, cuyas constantes son D_d y K_d respectivamente. Este sistema puede ser descrito como

$$m\ddot{x} = D_d(\dot{x}_d - \dot{x}) + K_d(x_d - x) - F_{ext}. \quad (4.15)$$

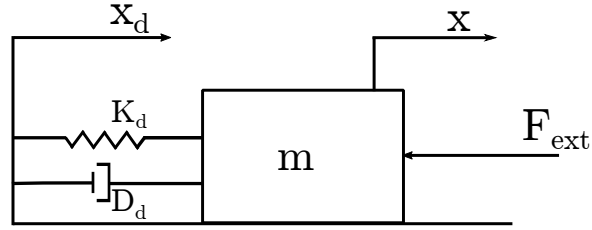


FIGURA 4.5: Sistema dinámico masa-resorte-amortiguador acoplado a una posición deseada x_d . La masa m se mueve en dirección del eje x , y es afectada por un estímulo F_{ext} y la impedancia generada por un resorte de constante K_d y un amortiguador de constante D_d .

Despejando u a partir de la definición de fuerzas externas F_{ext} en (4.14) y (4.15), se obtiene

$$u = D_d(\dot{x}_d - \dot{x}) + K_d(x_d - x). \quad (4.16)$$

Así, la aplicación de (4.16) en el sistema definido en (4.14), permitirá que este se comporte bajo el efecto dinámico de la impedancia definida en (4.15). Si además se agrega una inercia deseada a (4.16), la entrada de control se definirá como

$$u = M_d(\ddot{x}_d - \ddot{x}) + D_d(\dot{x}_d - \dot{x}) + K_d(x_d - x). \quad (4.17)$$

En robots manipuladores, el esquema de control de impedancia se deriva de forma similar que para sistemas unidimensionales definidos por (4.14). Agregando un vector de fuerzas externas \vec{F}_{ext} a (3.7), el sistema se define como

$$H(\vec{q})\ddot{\vec{q}} + \vec{n}(\vec{q}, \dot{\vec{q}}) + \vec{F}_{ext} = \vec{u}.$$

Considerando este nuevo modelo para un robot manipulador, de forma análoga al sistema definido en (4.15) se define el comportamiento dinámico deseado

$$\hat{H}(\vec{q})\ddot{\vec{q}} + \hat{\vec{n}}(\vec{q}) = M_d(\ddot{\vec{q}}_d - \ddot{\vec{q}}) + D_d(\dot{\vec{q}}_d - \dot{\vec{q}}) + K_d(\vec{q}_d - \vec{q}) - \vec{F}_{ext}. \quad (4.18)$$

Para conseguir que el robot manipulador siga el comportamiento deseado (4.18), en el espacio articular, se define la entrada de control

$$\vec{u} = M_d^{-1}(D_d(\dot{\vec{q}}_d - \dot{\vec{q}}) + K_d(\vec{q}_d - \vec{q})) + \hat{H}(\vec{q})\ddot{\vec{q}}_d + \hat{n}(\vec{q}, \dot{\vec{q}}). \quad (4.19)$$

En el espacio operacional, se puede redefinir el sistema (3.8) bajo efectos de una fuerza externa \vec{F}_{ext} como

$$H_x(\vec{q})\ddot{\vec{p}} + C_x(\vec{q}, \dot{\vec{q}})\dot{\vec{p}} + \vec{g}_x(\vec{q}) + \vec{F}_{ext} = J_A^{-T}\vec{u}. \quad (4.20)$$

Con la finalidad de que \vec{p} converja a \vec{p}_d bajo un comportamiento similar a (4.15) y (4.18), se define la señal de control a implementar, como

$$\vec{u} = J_A^T(\hat{H}_x(\vec{q})\vec{y} + \hat{C}_x(\vec{q}, \dot{\vec{q}})\dot{\vec{p}} + \hat{g}_x(\vec{q})), \quad (4.21)$$

donde

$$\vec{y} = \ddot{\vec{p}}_d + M_d^{-1}(D_d(\dot{\vec{p}}_d - \dot{\vec{p}}) + K_d(\vec{p}_d - \vec{p})).$$

Capítulo 5

RESULTADOS Y DISCUSIONES

En este capítulo se presentan y discuten los resultados obtenidos tras simular dinámicamente un robot UR5 controlado en el espacio operacional. Las pruebas son realizadas de modo que el efector final pueda ser trasladado de una pose inicial

$$\vec{p}_0 = (0.474, 0.454, 0.288, 0.114, -0.176, -2.561),$$

hacia una pose final

$$\vec{p}_d = (-0.307, 0.377, 0.109, 2.372, -0.052, 3.078).$$

Los resultados son mostrados como comparaciones entre la respuesta obtenida cuando el robot es controlado por dinámica inversa, y cuando este es controlado usando la formulación propuesta para el control por impedancia. De forma análoga a la Figura 4.1, en la Figura 5.1 se presenta un esquema de la implementación de los controladores de posición e impedancia, de forma intercambiable e independiente. En el esquema, el bloque “Controlador” corresponde al controlador de posición por dinámica inversa, descrito en (4.7), o al controlador por impedancia, descrito en (4.21). Ambos controladores son intercambiados dependiendo de los resultados a obtener. Debido a las restricciones en los comandos de control del robot UR5, la señal de control es procesada con la finalidad de enviar posiciones articulares al robot simulado en Gazebo. Como se muestra en el esquema, la señal de control \vec{u} es convertida en posiciones articulares \vec{q}_r , y la realimentación del sistema es transformada a posiciones y orientaciones en el espacio cartesiano \vec{p} con la finalidad de computar los controladores definidos en el espacio cartesiano.

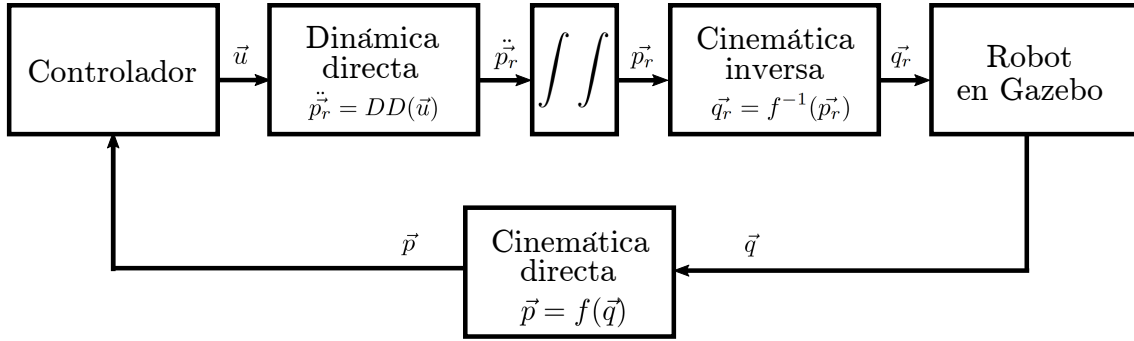


FIGURA 5.1: Diagrama de bloques de la implementación de los controladores de posición e impedancia para la simulación del robot UR5 en Gazebo. El esquema permite intercambiar entre los controladores sin pérdida de generalidad.

El controlador de posición por dinámica inversa es definido en (4.7), y los parámetros elegidos para su descripción son:

$$K_P = \text{diag}(800, 800, 600, 600, 600, 600)$$

y

$$K_D = \text{diag}(57, 57, 49, 49, 49, 49),$$

donde $\text{diag}(\cdot)$ es un operador que define los elementos diagonales de una matriz diagonal. Estos valores son elegidos de modo que se cumpla la condición $K_D = 2\sqrt{K_P}$, que permite garantizar que el comportamiento del sistema controlado sea críticamente amortiguado. Una condición adicional para la elección de estos valores es la compensación que debe existir con los parámetros del modelo dinámico del robot y los pasos de tiempo propios de la computación numérica del controlador. Además, a partir de (4.7), se puede inferir que la contribución de los parámetros K_P y K_D pueden determinar la inestabilidad del sistema o la saturación de los actuadores del robot, en caso los valores elegidos fueran muy grandes.

Para el controlador por impedancia, se usa la ley de control definida en (4.21). Considerando esta, se realizan las simulaciones para tres casos con parámetros distintos

que dependen del grado de rigidez de la impedancia impuesta. Para el caso con impedancia de baja rigidez, se usan los parámetros

$$D_{d1} = \text{diag}(600, 267, 530, 100, 167, 167)$$

y

$$K_{d1} = \text{diag}(60, 150, 60, 10, 50, 15).$$

Par el caso con rigidez media, los parámetros usados son

$$D_{d2} = \text{diag}(900, 400, 800, 150, 250, 250)$$

y

$$K_{d2} = \text{diag}(80, 200, 80, 10, 60, 20).$$

Por último, para el caso con rigidez alta, se definen los parámetros

$$D_{d3} = \text{diag}(1300, 800, 1200, 550, 400, 400)$$

y

$$K_{d3} = \text{diag}(90, 210, 90, 30, 70, 30).$$

Las simulaciones para cada uno de los casos descritos previamente son ejecutadas haciendo uso de los nodos `ID_OS_controller` y `Imp_OS_controller` en el entorno ROS (ver Apéndice A). En estos nodos, los valores correspondientes a la rigidez son variados internamente dependiendo del caso a evaluar. Además, ambos nodos permiten exportar valores articulares y cartesianos del robot durante la simulación, lo que permite graficar la trayectoria del efector final y realizar un análisis cuantitativo de este comportamiento.

5.1 Control en un entorno sin obstáculos

Las simulaciones mostradas en esta sección son ejecutadas en un entorno dinámico desconocido para el robot. Este, mostrado en la Figura 5.2, permite que el robot pueda trasladarse libremente en el espacio Cartesiano desde la pose inicial \vec{p}_0 , hacia la pose deseada \vec{p}_d . Los resultados mostrados a lo largo de esta sección describen el comportamiento espacial y temporal de la pose del robot, así como del esfuerzo aplicado por cada actuador durante su movimiento.

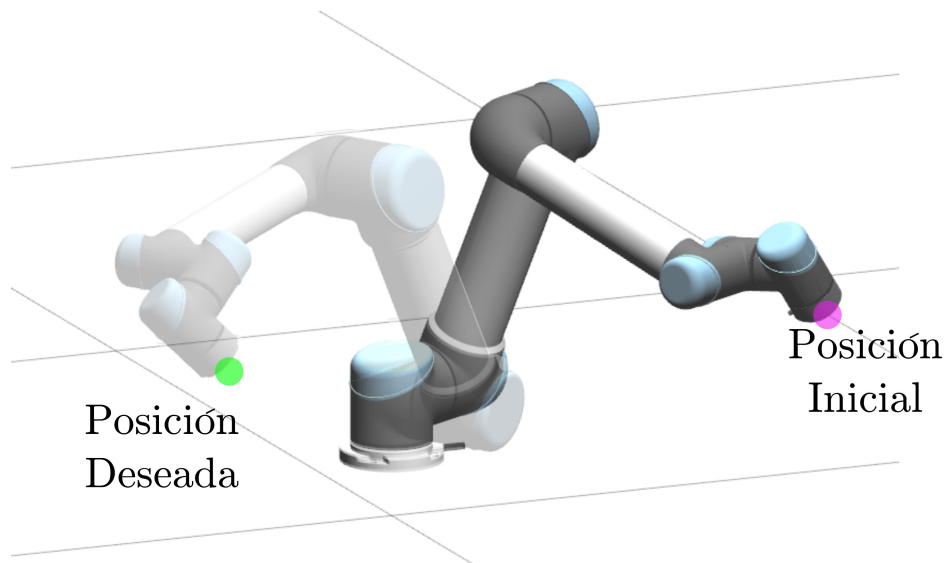


FIGURA 5.2: Entorno de simulación sin obstáculo.

5.1.1 Respuesta sin interacción e impedancia de rigidez baja

Las simulaciones son realizadas para el robot controlado en posición por dinámica inversa, con parámetros K_P y K_D , y para el robot controlado por impedancia, con parámetros D_{d1} y K_{d1} .

En la Figura 5.3 se presenta el comportamiento del efector final en el espacio Cartesiano para el sistema controlado por ambas formulaciones. En esta se observa que,

cuando el robot tiene la capacidad de moverse libremente en el espacio, ambos controladores garantizan su convergencia a la posición final deseada. Debido a que el estado final es definido únicamente por una posición deseada, la trayectoria seguida por el robot difiere en ambos casos. Sin embargo, el comportamiento en estado estacionario no es afectado, alcanzando estados finales con distancias euclidianas hacia la posición deseada de 3.79×10^{-5} m y 6.01×10^{-4} m para el sistema controlado por dinámica inversa e impedancia, respectivamente.

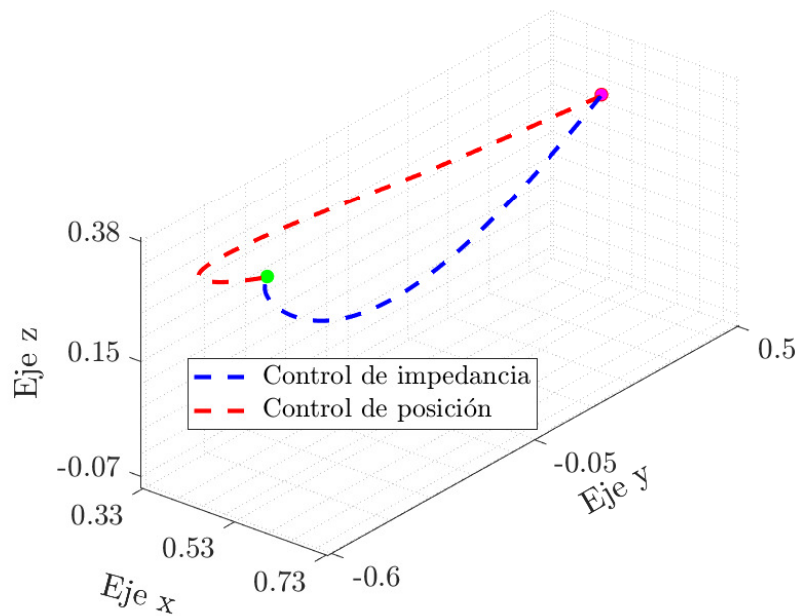


FIGURA 5.3: Comportamiento controlado del efector final en el espacio cartesiano para los parámetros K_P , K_D , D_{d1} y K_{d1} . El robot parte de la posición inicial (magenta), hacia la posición deseada (verde).

En la Figura 5.4, el comportamiento espacial correspondiente a la Figura 5.3 es mostrado desde una perspectiva temporal para el robot en cada eje Cartesiano. En esta se puede observar que el robot converge a las posiciones deseadas de una forma amortiguada en los ejes x , y y z , con errores respectivos de 2.11×10^{-5} m, 1.62×10^{-5} m y 2.69×10^{-5} m para el sistema controlado por dinámica inversa, y 2.45×10^{-6} m, 1.84×10^{-4} m y 5.72×10^{-4} m, para el sistema controlado por impedancia. Las variaciones existentes en

sus estados transitorios no presentan desviaciones que afecten sus estados finales, por lo que la convergencia hacia los estados deseados es asegurada con errores pequeños.

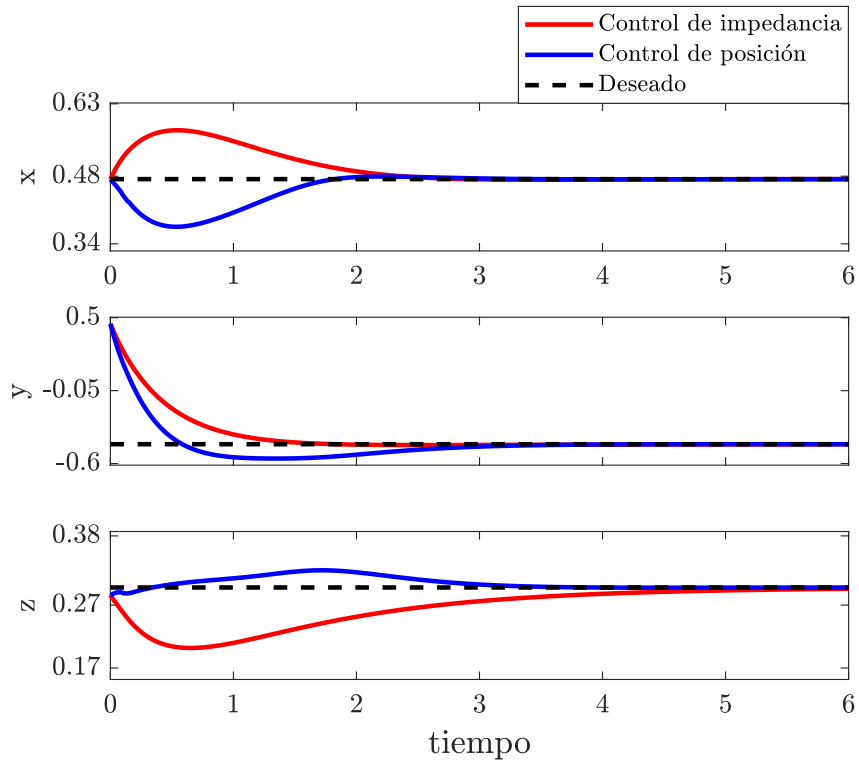


FIGURA 5.4: Comportamiento temporal del efector final en los ejes Cartesianos para el robot controlado para los parámetros K_P , K_D , D_{d1} y K_{d1} . Los paneles, de arriba hacia abajo, representan a x , y y z respectivamente.

Los comportamientos mostrados en las Figuras 5.3 y 5.4 permiten identificar el desarrollo de la traslación del robot. Sin embargo, para conocer el comportamiento completo de la pose del robot, es necesario conocer también la evolución de las orientaciones. En la Figura 5.5 se muestra el comportamiento temporal de los ángulos de orientación ζ , ϕ y ψ . En esta se puede observar que, al igual que en la posición, la tendencia en cada ángulo es su convergencia a los estados deseados. En este caso, para ζ , ϕ y ψ , los errores respectivos en estado estacionario son de 3.25×10^{-5} rad, 4.17×10^{-6} rad y 42×10^{-3} rad, cuando el sistema es controlado por dinámica inversa, y 3.25×10^{-5} rad, 4.30×10^{-4} rad y 7.61×10^{-7} rad, cuando este es controlado por impedancia.

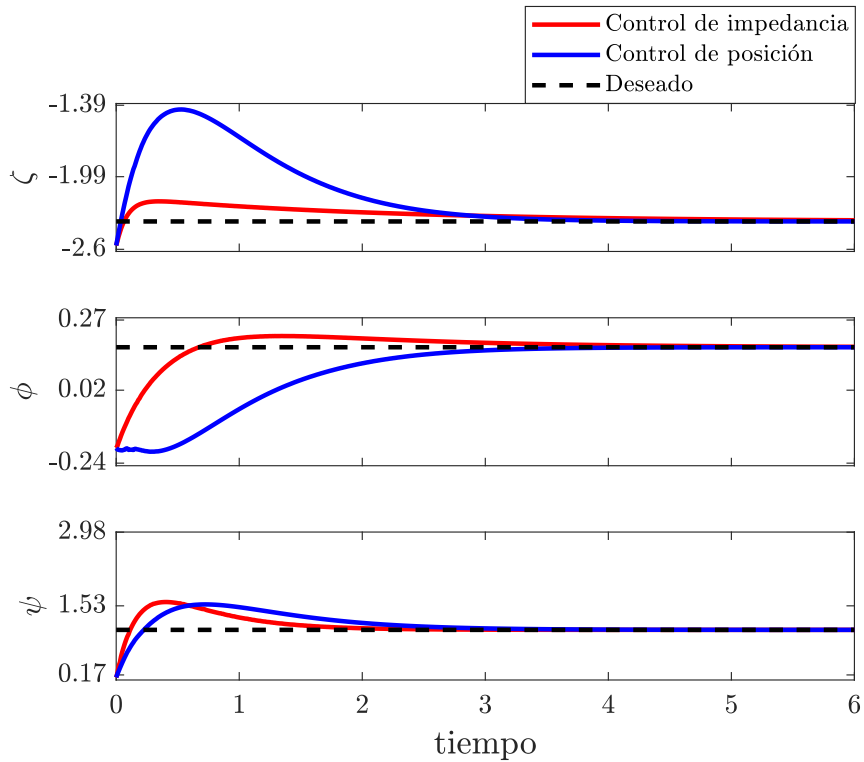


FIGURA 5.5: Comportamiento temporal la orientación del robot controlado para los parámetros K_P , K_D , D_{d1} y K_{d1} . Los paneles, de arriba hacia abajo, representan la orientación parametrizada por los ángulos de Euler ζ , ϕ y ψ respectivamente.

En la Figura 5.6 se presenta el esfuerzo producido por cada junta con respecto al tiempo, cuando el robot es controlado por ambas formulaciones. Como se observa para ambos controladores, los esfuerzos en cada junta tienen amplitudes que se encuentra dentro de los rangos de operación del robot. Las variaciones en cada junta son pequeñas, y todas se mantienen estables, permitiendo que el robot converja a la pose deseada.

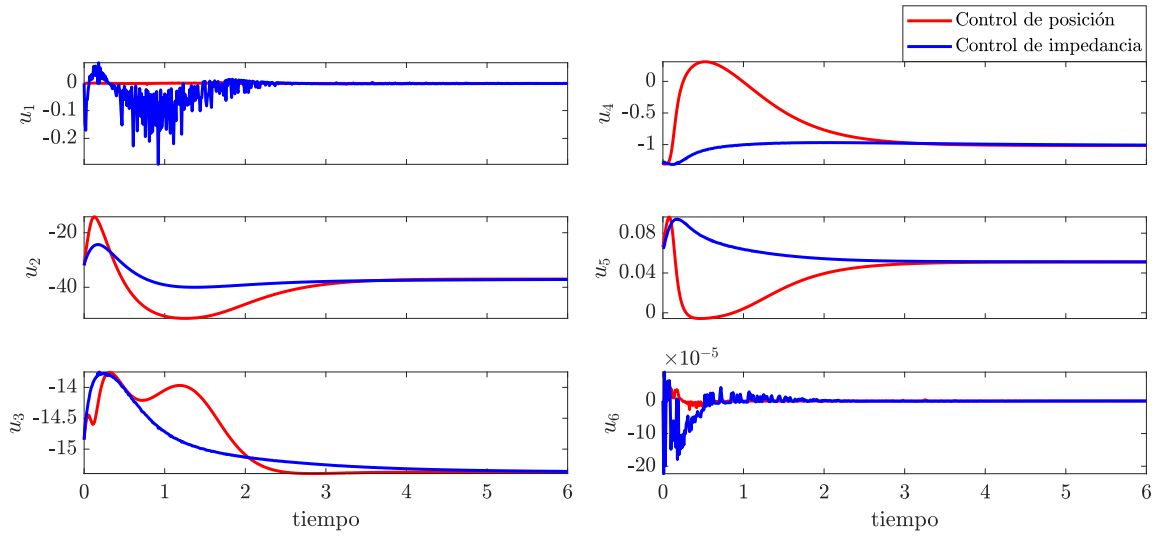


FIGURA 5.6: Comportamiento temporal del esfuerzo en cada junta del robot controlado para los parámetros K_P , K_D , D_{d1} y K_{d1} . La variable u_i representa el esfuerzo aplicado por la junta i , donde este subíndice toma valores de 1 a 6 para juntas consecutivas de la base hacia el efector final.

5.1.2 Respuesta sin interacción e impedancia de rigidez media

Las simulaciones son realizadas para el robot controlado en posición por dinámica inversa, con parámetros K_P y K_D , y para el robot controlado por impedancia, con parámetros D_{d2} y K_{d2} .

En la Figura 5.21 se muestra el comportamiento del efector final en el espacio Cartesiano, cuando el sistema es controlado por dinámica inversa y por impedancia. Como se observa, cuando el movimiento del robot es libre, la convergencia a la posición deseada puede asegurarse para ambas formulaciones. Además, las trayectorias seguidas por el robot para alcanzar su estado final es similar para ambos controladores, a pesar de no ser un requisito específico en el desarrollo de estos. Así, el desempeño de ambos controladores en estado estacionario es respaldado por las distancias euclidianas hacia el estado deseado de 3.78×10^{-5} m y 1.81×10^{-4} m, para los controladores por dinámica inversa e impedancia, respectivamente.

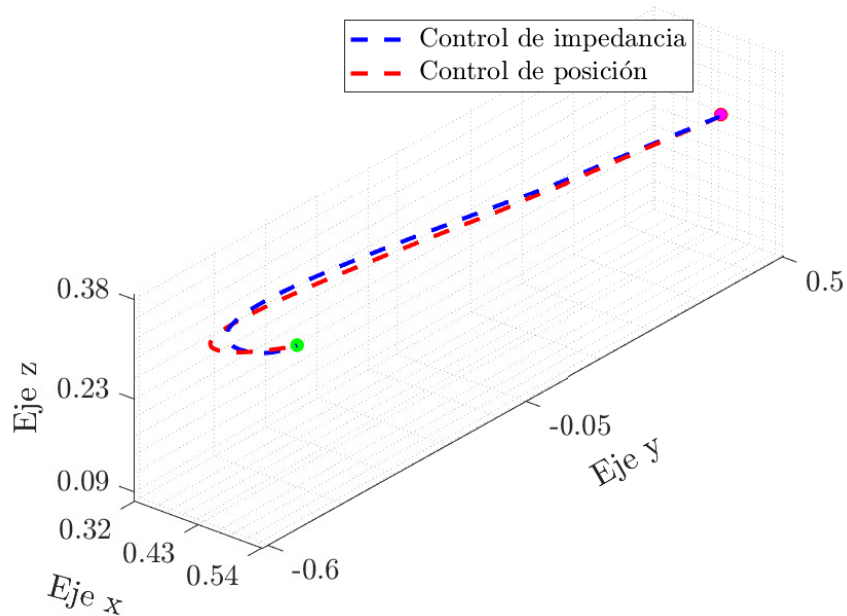


FIGURA 5.7: Comportamiento controlado del efector final en el espacio cartesiano para los parámetros K_P , K_D , D_{d2} y K_{d2} . El robot parte de la posición inicial (magenta), hacia la posición deseada (verde).

En la Figura 5.8 se muestra el comportamiento temporal del efector final en cada eje Cartesiano. En esta imagen se muestra que, después de un transitorio, en ambas formulaciones el robot consigue convergir a la posición deseada para cada eje. Además, al igual que en el caso tridimensional, las trayectorias temporales seguidas para ambas formulaciones son similares. Los errores obtenidos en ejes x , y y z , son respectivamente 7.44×10^{-6} m, 1.78×10^{-4} m y 3.59×10^{-5} m para el sistema controlado por dinámica inversa, y 2.11×10^{-5} m, 1.62×10^{-5} m y 2.69×10^{-5} m, para el sistema controlado por impedancia.

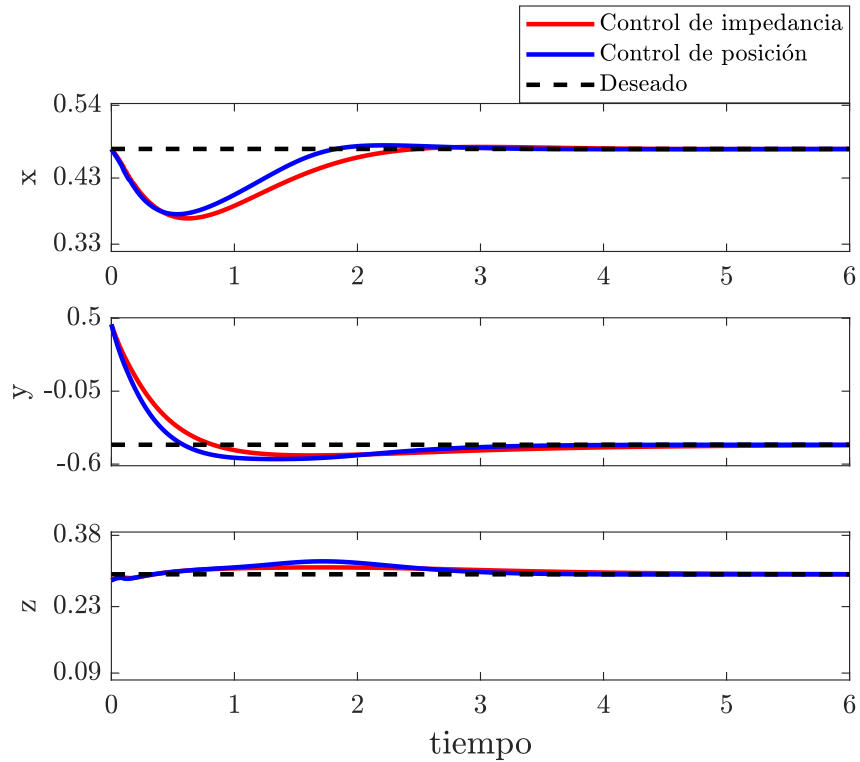


FIGURA 5.8: Comportamiento temporal del efector final en los ejes Cartesianos para el robot controlado para los parámetros K_P , K_D , D_{d2} y K_{d2} . Los paneles, de arriba hacia abajo, representan a x , y y z respectivamente.

Para complementar la descripción del comportamiento de la pose del efector final, en la Figura 5.9 se muestra el comportamiento temporal de su orientación descrita por los ángulos de Euler ζ , ϕ y ψ . Se observa que, la convergencia en estado estacionario al estado deseado es garantizada para ambos controladores. Sin embargo, a diferencia del comportamiento en posición, en la etapa transitoria la orientación en ζ tiende a mostrar una variación notable entre controladores. A pesar de las variaciones en transición, en estado estable los errores para ζ , ϕ y ψ son respectivamente 4×10^{-3} rad, 7.61×10^{-4} rad y 5.81×10^{-7} rad, cuando el sistema es controlado por dinámica inversa, y 3.95×10^{-5} rad, 4.17×10^{-6} rad y 4.08×10^{-5} rad, cuando este es controlado por impedancia.

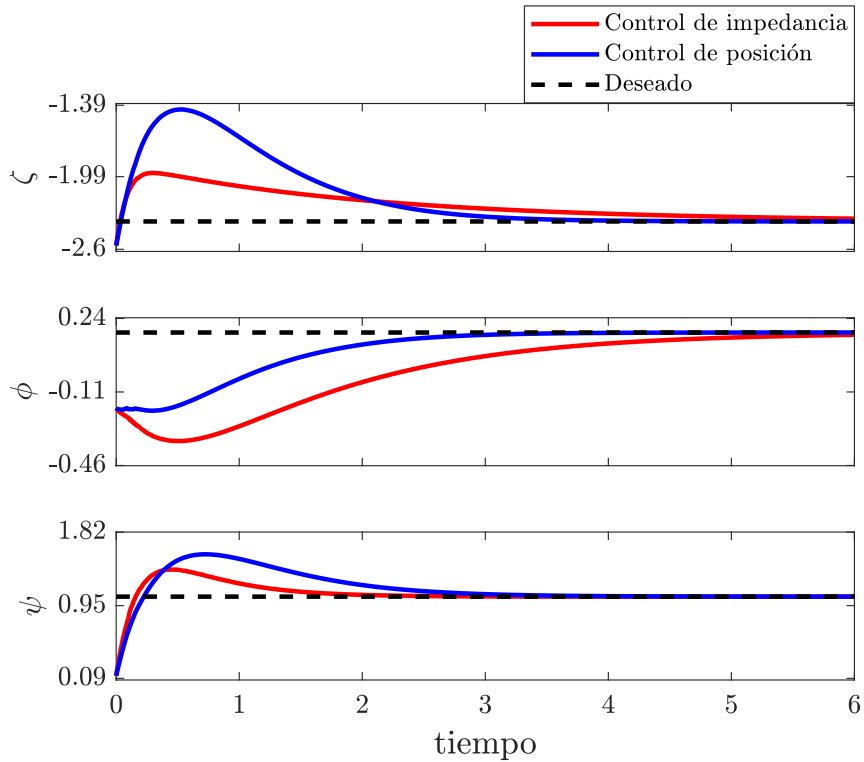


FIGURA 5.9: Comportamiento temporal la orientación del robot controlado para los parámetros K_P , K_D , D_{d2} y K_{d2} . Los paneles, de arriba hacia abajo, representan la orientación parametrizada por los ángulos de Euler ζ , ϕ y ψ respectivamente.

El esfuerzo interno generado por cada junta del robot en su traslación hacia la pose deseada es mostrado en la Figura 5.10. Como se observa, existen variaciones pequeñas en el esfuerzo aplicado por cada junta. Sin embargo, estas variaciones no afectan a la convergencia del robot a la pose deseada, como se mostró en figuras previas correspondientes a la posición y orientación.

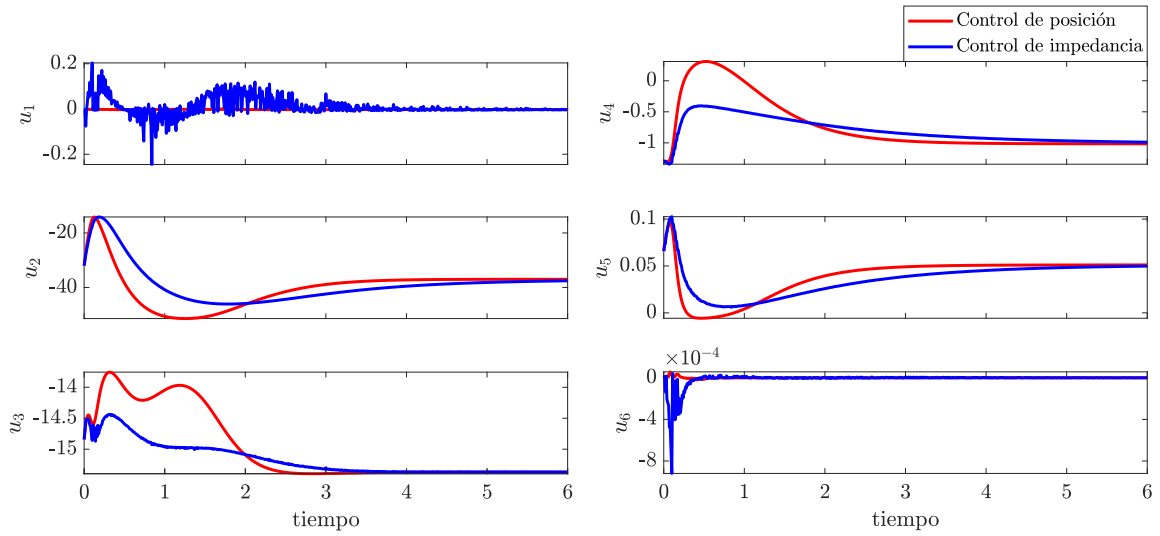


FIGURA 5.10: Comportamiento temporal del esfuerzo en cada junta del robot controlado para los parámetros K_P , K_D , D_{d2} y K_{d2} . La variable u_i representa el esfuerzo aplicado por la junta i , donde este subíndice toma valores de 1 a 6 para juntas consecutivas de la base hacia el efector final.

5.1.3 Respuesta sin interacción e impedancia de rigidez alta

Las simulaciones son realizadas para el robot controlado en posición por dinámica inversa, con parámetros K_P y K_D , y para el robot controlado por impedancia, con parámetros D_{d3} y K_{d3} .

En la Figura 5.11 se muestra el comportamiento espacial del efector final del robot. Como se observa, cuando el sistema es controlado por ambas formulaciones, sin interacciones en su trayectoria, el efector final converge a la posición deseada con exactitud. Las distancias euclidianas de la posición final con respecto a la posición deseada es de 3.78×10^{-5} m para el sistema controlador por dinámica inversa, y 2.63×10^{-6} m para el sistema controlado por impedancia.

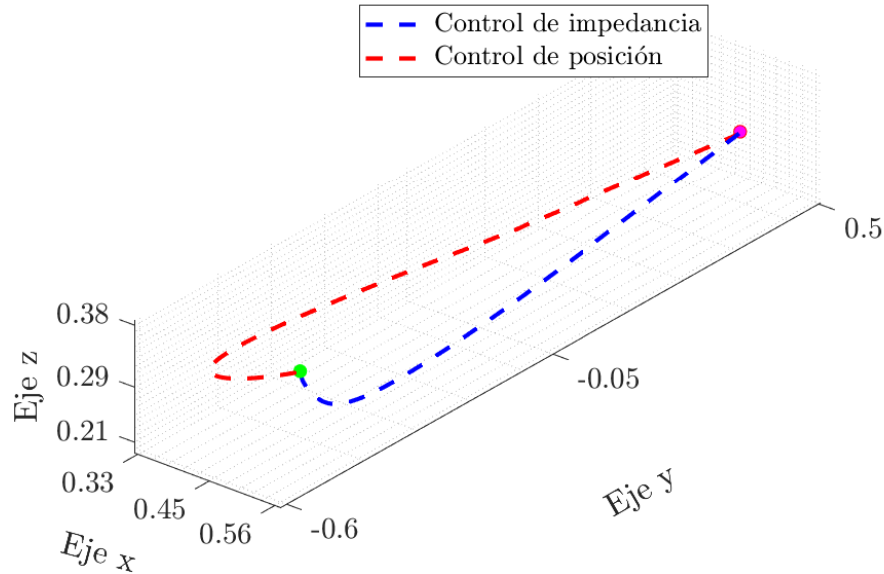


FIGURA 5.11: Comportamiento controlado del efector final en el espacio cartesiano para los parámetros K_P , K_D , D_{d3} y K_{d3} . El robot parte de la posición inicial (magenta), hacia la posición deseada (verde).

En la Figura 5.12 se muestra el comportamiento temporal del movimiento del efector final en los ejes Cartesianos x , y y z . En esta se puede observar que el comportamiento temporal en cada eje, es consistente con el comportamiento mostrado en la Figura 5.11. Para ambas formulaciones se garantiza la convergencia del sistema al estado deseado, con variaciones mínimas en su trayectoria. Las variaciones en el trayecto, se dan debido a que el objetivo del controlador es únicamente alcanzar una posición deseada. Los errores de posición en estado estacionario, para cada eje, son respectivamente 2.12×10^{-5} m, 1.62×10^{-5} m y 2.69×10^{-5} m para el sistema controlado por dinámica inversa, y 4.85×10^{-9} m, 4×10^{-7} m y 2.6×10^{-6} m, para el sistema controlado por impedancia.

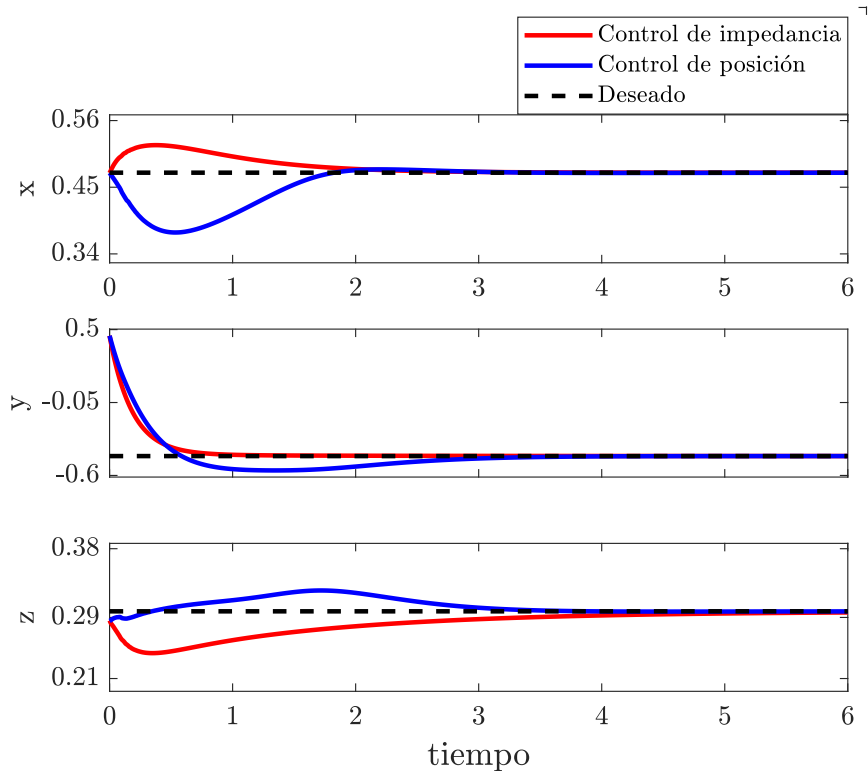


FIGURA 5.12: Comportamiento temporal del efector final en los ejes Cartesianos para el robot controlado para los parámetros K_P , K_D , D_{d3} y K_{d3} . Los paneles, de arriba hacia abajo, representan a x , y y z respectivamente.

El comportamiento temporal de la orientación es mostrado en la Figura 5.13. Se puede observar que, la convergencia al estado deseado es garantizada por ambos controladores con exactitud. Sin embargo, existe una variación notable en el estado transitorio, principalmente en el sistema controlado por dinámica inversa. Este comportamiento, al igual que en el comportamiento en posición, se debe a la definición del objetivo a controlar, que no abarca restricciones en el estado transitorio. Los errores en estado estacionario para cada ángulo de orientación es respectivamente 3.96×10^{-3} rad, 4.17×10^{-4} rad y 4.08×10^{-7} rad, cuando el sistema es controlado por dinámica inversa, y 3.95×10^{-5} rad, 1.64×10^{-6} rad y 8.23×10^{-10} rad, cuando este es controlado por impedancia.

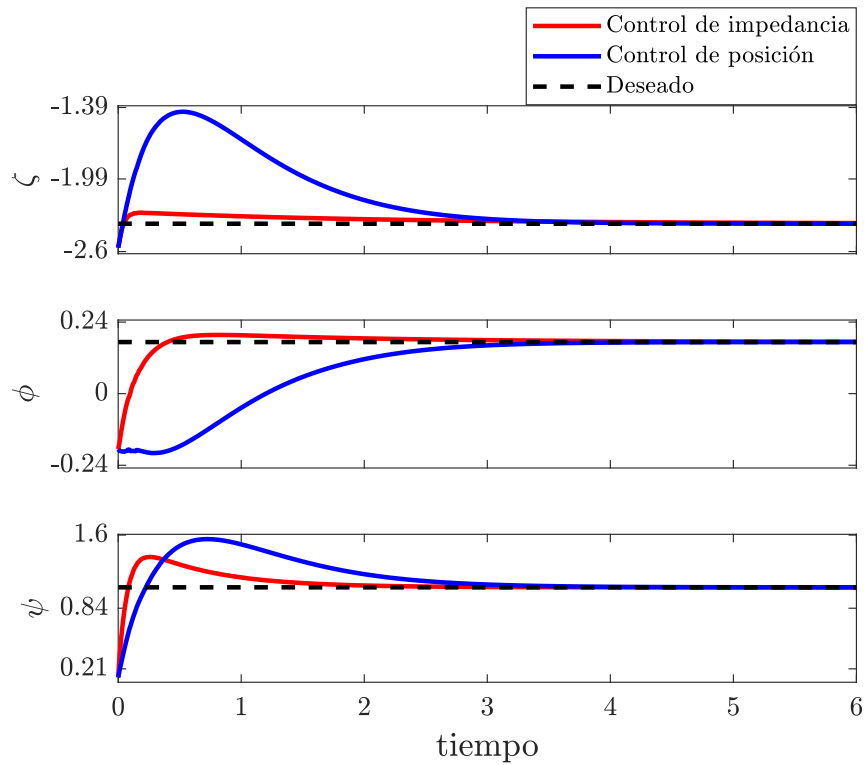


FIGURA 5.13: Comportamiento temporal la orientación del robot controlado para los parámetros K_P , K_D , D_{d3} y K_{d3} . Los paneles, de arriba hacia abajo, representan la orientación parametrizada por los ángulos de Euler ζ , ϕ y ψ respectivamente.

En la Figura 5.14 se muestra que, a pesar de que la variación entre ambas formulaciones no es notable, las amplitudes son mayores que en los casos donde se eligen parámetros para impedancias de rigidez baja y media. A pesar de esto, la convergencia en el estado estable es garantizada para ambos controladores. De esta forma los comportamientos de posición y orientación convergen a la pose deseada.

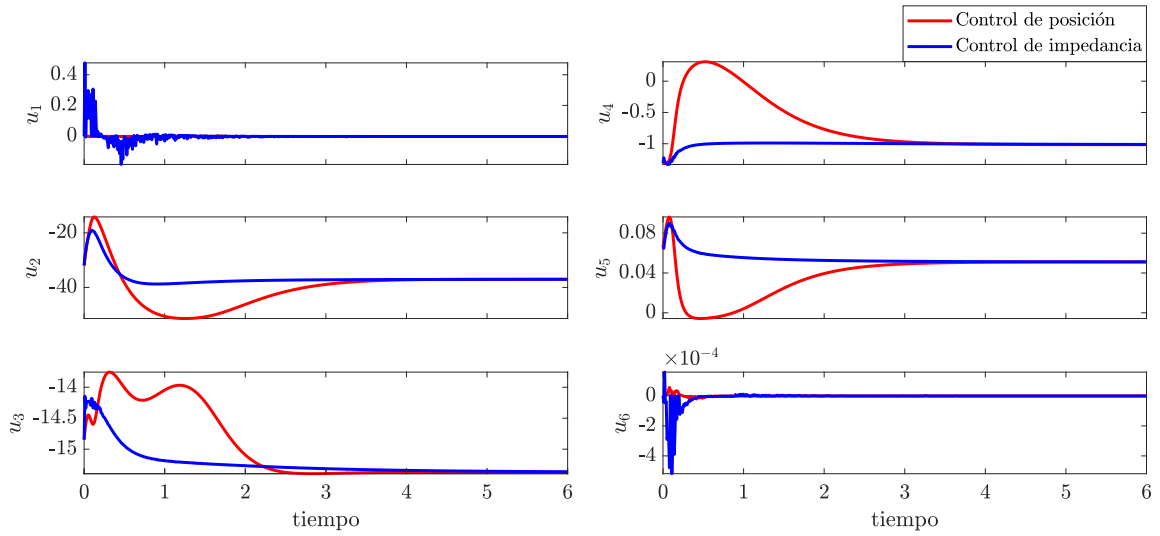


FIGURA 5.14: Comportamiento temporal del esfuerzo en cada junta del robot controlado para los parámetros K_P , K_D , D_{d3} y K_{d3} . La variable u_i representa el esfuerzo aplicado por la junta i , donde este subíndice toma valores de 1 a 6 para juntas consecutivas de la base hacia el efector final.

5.2 Control en un entorno con un obstáculo

A diferencia de la sección previa, en esta se presentan simulaciones para un entorno donde el robot interactúa con una superficie irregular durante su trayectoria. El entorno global, donde se muestra el posicionamiento del robot y del objeto de interacción, así como las posiciones inicial y deseada, se muestra en la Figura 5.15. En este, la traslación del robot en el espacio Cartesiano desde la pose inicial \vec{p}_0 , hacia la pose deseada \vec{p}_d , es interrumpida por un objeto.

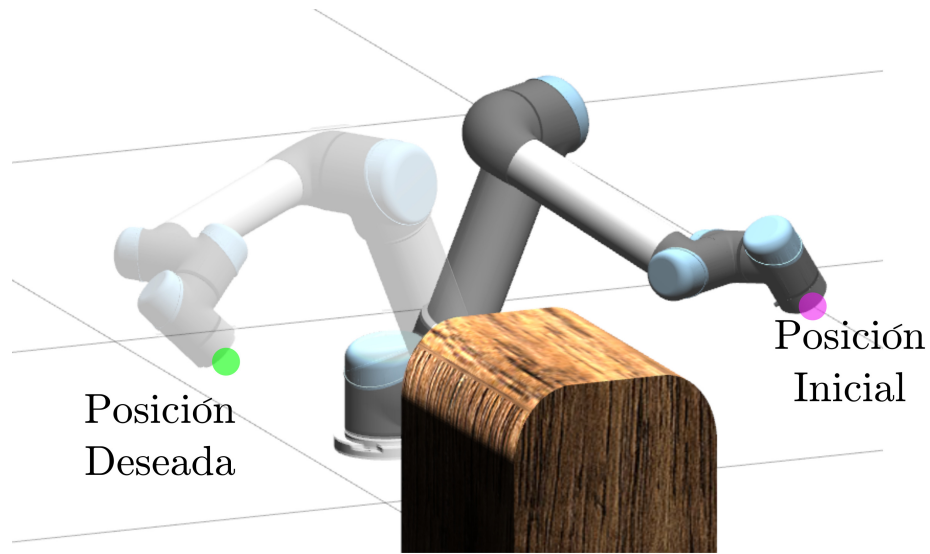


FIGURA 5.15: Entorno de simulación con obstáculo.

En la Figura 5.16 se muestran dos secuencias de imágenes, tomadas en distintos instantes de tiempo. Estas secuencias ilustran el comportamiento del robot al trasladarse de un estado inicial a un estado deseado, bajo la acción del controlador de posición y el controlador por impedancia. Los resultados mostrados a lo largo de esta sección describen el comportamiento espacial y temporal de la pose del robot, así como del esfuerzo aplicado por cada actuador durante su movimiento y los efectos generados por la interacción con el objeto.

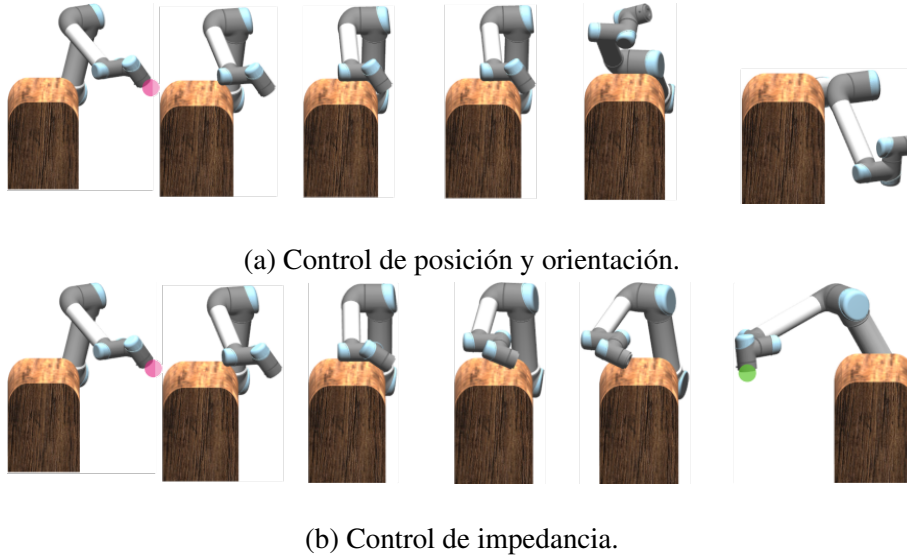


FIGURA 5.16: De izquierda hacia derecha, capturas de la interacción del robot con el objeto de superficie irregular a través del tiempo. En (a) el comportamiento bajo el controlador de posición y orientación y en (b) el comportamiento bajo el controlador por impedancia. En ambos casos, las posiciones y orientaciones iniciales (en púrpura) y deseadas (en verde) son las mismas.

5.2.1 Respuesta con interacción e impedancia de rigidez baja

Las simulaciones son realizadas para el robot controlado en posición por dinámica inversa, con parámetros K_P y K_D , y para el robot controlado por impedancia, con parámetros D_{d1} y K_{d1} .

El comportamiento espacial del efector final se muestra en la Figura 5.17, donde se observan claras variaciones con respecto al comportamiento sin interacciones, mostrado en la Figura 5.3 para los mismos parámetros de controlador. Se puede observar que, cuando existe un objeto que interfiere en la trayectoria del robot, el comportamiento del robot controlado por dinámica inversa presenta un cambio brusco cuando el robot interactúa con el objeto, esto como producto de inestabilidades generadas en el simulador. Por otro lado, el controlador por impedancia permite que el robot sobrepase los efectos generados por la interacción, y después de un estado transitorio poco amortiguado, converja a la posición

deseada con un error significativo con respecto al comportamiento sin interacción. La distancia euclidiana obtenida para el controlador por impedancia en estado estacionario, con respecto a la posición deseada es de 0.02 m.

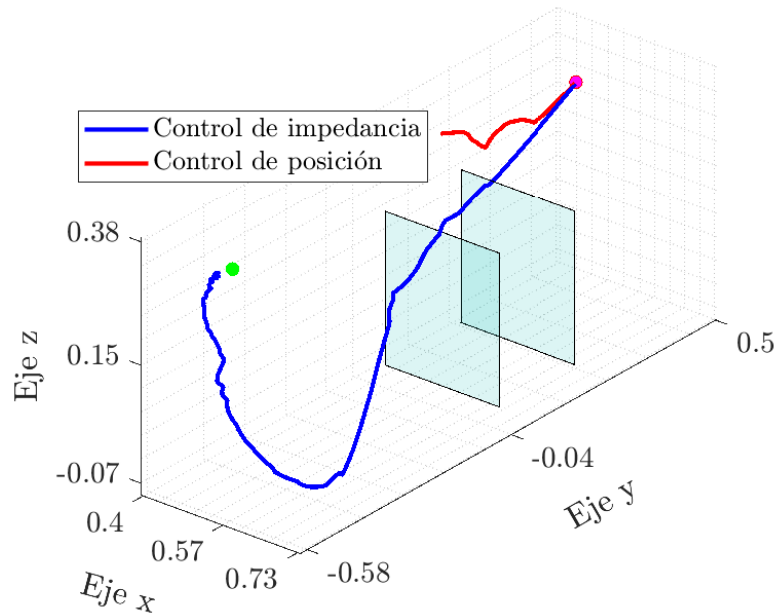


FIGURA 5.17: Comportamiento controlado del efector final en el espacio cartesiano para los parámetros K_P , K_D , D_{d1} y K_{d1} . El robot parte de la posición inicial (magenta), hacia la posición deseada (verde). Los planos (esmeralda) representan la posición de un objeto en el espacio.

En la Figura 5.18 se muestra el comportamiento del efector final controlado, para cada eje Cartesiano. Se observa que, para ambos controladores, cuando el robot interactúa en su trayectoria, este experimenta efectos notorios a lo largo de su trayectoria. En el caso del control por dinámica inversa, la interacción genera inestabilidad en el simulador, mientras que el control por impedancia permite su recuperación, sin mucha exactitud en la posición a la que converge en el estado final. Los errores en x , y y z para el sistema controlado por impedancia son 0.02 m, 0.01 m y 0.02 m respectivamente.

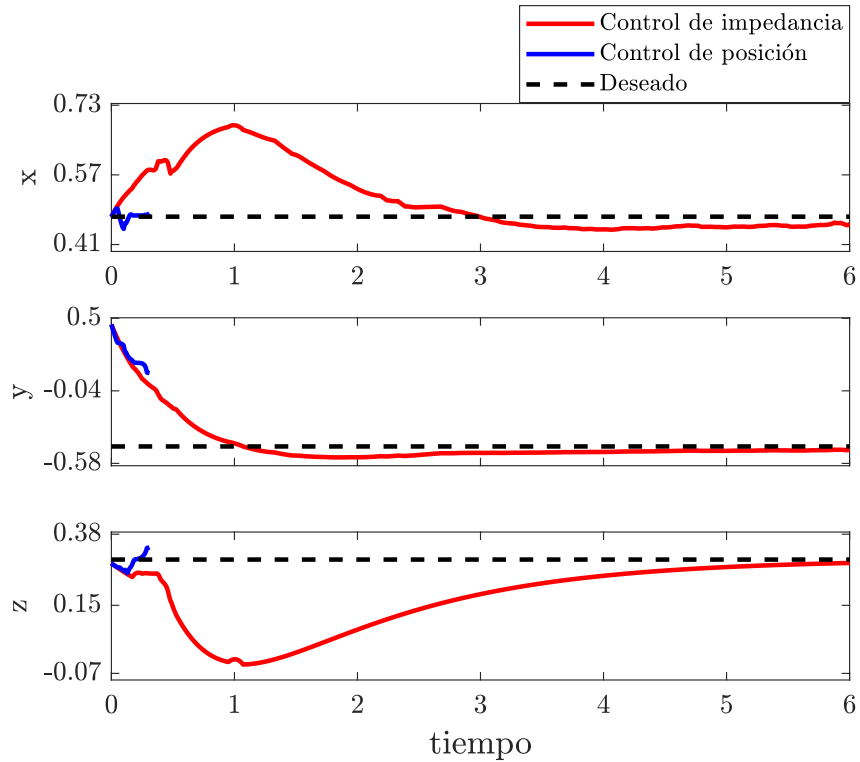


FIGURA 5.18: Comportamiento temporal del efector final en los ejes Cartesianos para el robot controlado para los parámetros K_P , K_D , D_{d1} y K_{d1} , cuando este interactúa con un objeto en su trayectoria. Los paneles, de arriba hacia abajo, representan a x , y y z respectivamente.

De forma similar al comportamiento temporal de la posición, mostrado en la Figura 5.18, la Figura 5.19 muestra el comportamiento de la orientación del robot en ángulos de Euler. Como se observa en este caso, a diferencia de los comportamientos en posición mencionados previamente, cuando el robot es controlado por impedancia, el ángulo ψ tiende a oscilar sin un patrón periódico. Esto a pesar de que los otros dos ángulos de orientación sí consiguen convergir a los estados deseados. Los errores en estado estacionario de los ángulos ζ , ϕ y ψ son 0.004 rad, 0.006 rad y 0.6 rad respectivamente.

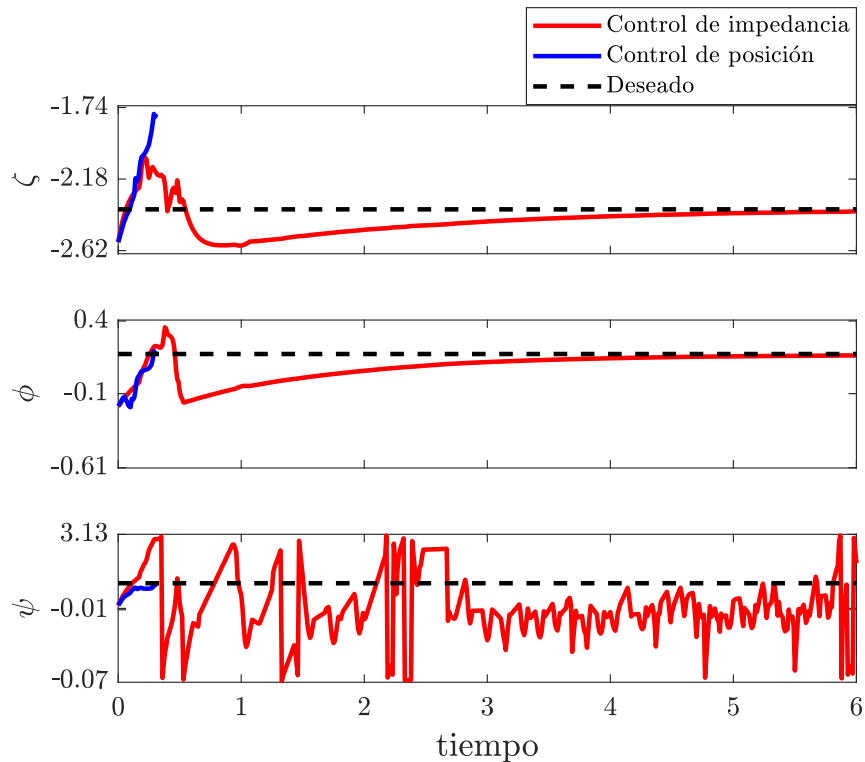


FIGURA 5.19: Comportamiento temporal de la orientación del robot controlado para los parámetros K_P , K_D , D_{d1} y K_{d1} , cuando este interactúa con un objeto en su trayectoria. Los paneles, de arriba hacia abajo, representan la orientación parametrizada por los ángulos de Euler ζ , ϕ y ψ respectivamente.

El comportamiento de la pose del efector final, mostrado previamente, es extendido en la Figura 5.20, donde se muestra el esfuerzo interno realizado por cada junta, para ambos controladores. Como puede observarse, para el control por dinámica inversa, cuando el robot interactúa en su trayectoria, las juntas son llevadas a sus límites, generando saturación en estos. Este efecto se repite discontinuamente desde su límite positivo a su límite negativo consecutivamente. Por otro lado, cuando el robot es controlado por impedancia, se muestra que cuando este interactúa, algunas juntas llegan a su límite. Sin embargo, a pesar de esto, el controlador permite que estas salgan del estado de saturación y se recuperen volviendo a comportarse dentro de un rango permitido por sus restricciones físicas.

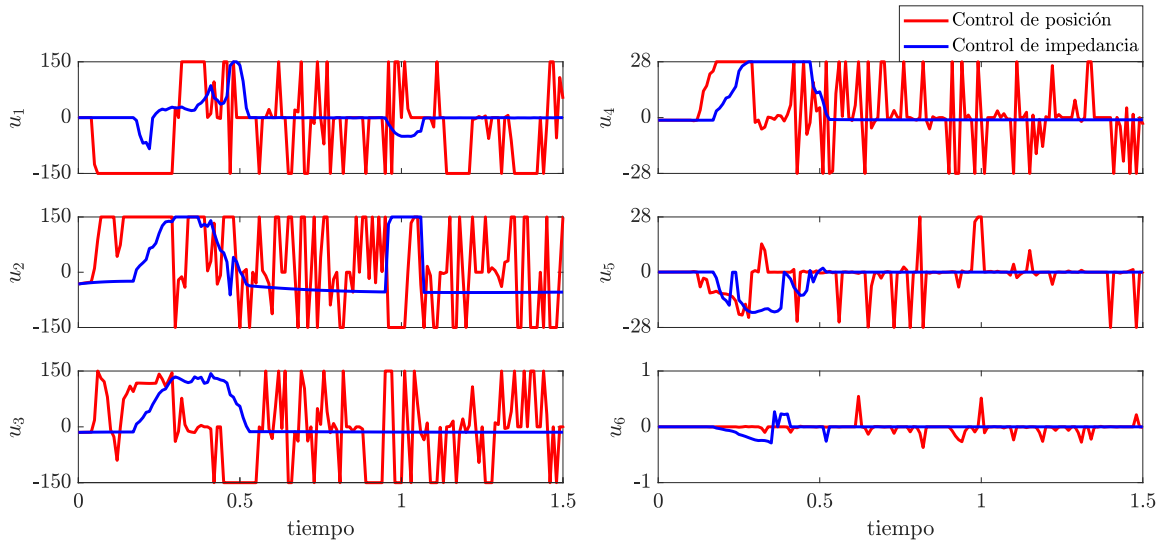


FIGURA 5.20: Comportamiento temporal del esfuerzo en cada junta del robot controlado para los parámetros K_P , K_D , D_{d1} y K_{d1} , cuando este interactúa con un objeto en su trayectoria. La variable u_i representa el esfuerzo aplicado por la junta i , donde este subíndice toma valores de 1 a 6 para juntas consecutivas de la base hacia el efector final.

5.2.2 Respuesta con interacción e impedancia de rigidez media

Las simulaciones son realizadas para el robot controlado en posición por dinámica inversa, con parámetros K_P y K_D , y para el robot controlado por impedancia, con parámetros D_{d2} y K_{d2} .

El comportamiento en el espacio del efector final, controlado por dinámica inversa y por impedancia, es mostrado en la Figura 5.21. Como se observa, la interacción con un objeto en su trayectoria genera efectos distintos para ambos controladores. En el caso del sistema controlado en posición, el simulador se inestabiliza, generando un corte brusco en la trayectoria. En el caso del sistema controlado por impedancia, el efector final sigue una trayectoria similar a la forma del objeto con el que interactúa, y posteriormente consigue llegar a la posición final deseada con un error muy pequeño en estado estable. La distancia euclidiana entre la posición final alcanzada por el robot controlado en impedancia, y la posición de referencia, es de 4.47×10^{-4} m.

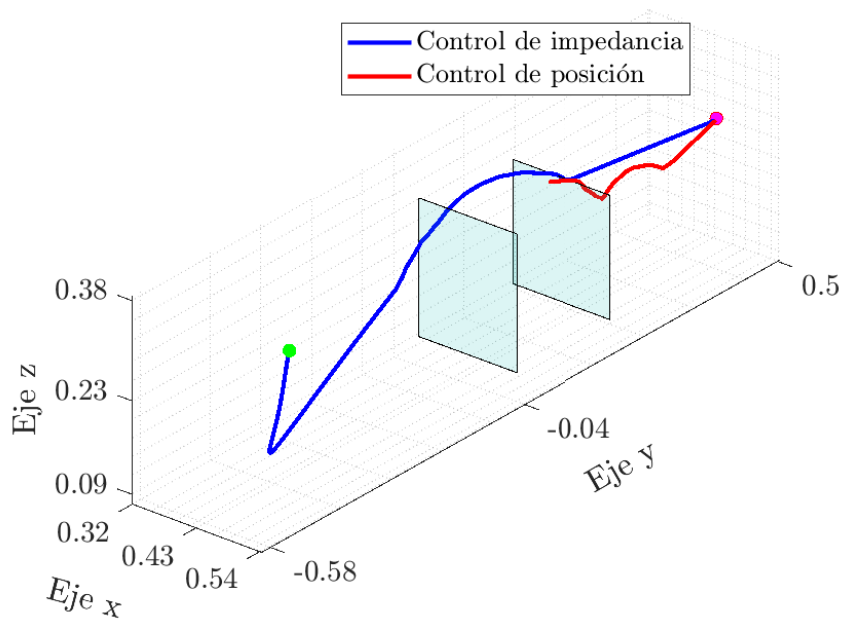


FIGURA 5.21: Comportamiento controlado del efector final en el espacio cartesiano para los parámetros K_P , K_D , D_{d2} y K_{d2} . El robot parte de la posición inicial (magenta), hacia la posición deseada (verde). Los planos (esmeralda) representan la posición de un objeto en el espacio.

El comportamiento temporal de cada eje, para el robot controlado por ambas formulaciones, es mostrado en la Figura 5.22. De forma similar a la Figura 5.21, cuando el sistema se controla por dinámica inversa, existe un corte brusco en la trayectoria, debido a la inestabilidad del simulador generada por la interacción con el objeto. Por otro lado, cuando el sistema es controlado por impedancia, las variaciones principales se observan en la trayectoria de los ejes x y y . Sin embargo, después de estas variaciones, el robot tiende a convergir a los estados deseados para cada uno con exactitud. Para el sistema controlado por impedancia, los errores en estado estable para x , y y z son respectivamente 9.51×10^{-7} m, 1.53×10^{-4} m y 4.2×10^{-4} m

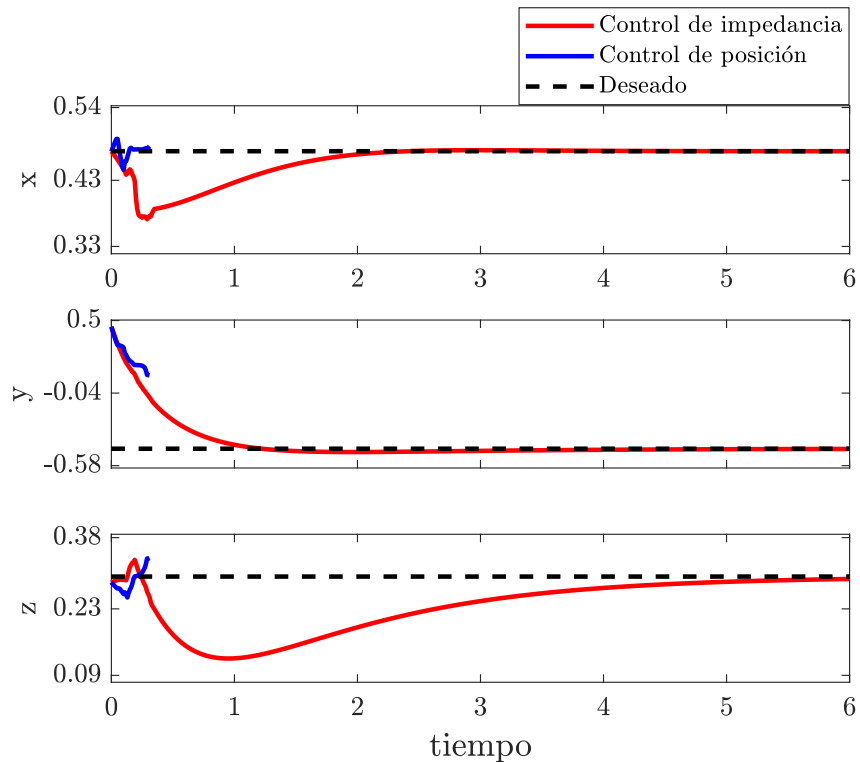


FIGURA 5.22: Comportamiento temporal del efector final en los ejes Cartesianos para el robot controlado para los parámetros K_P , K_D , D_{d2} y K_{d2} , cuando este interactúa con un objeto en su trayectoria. Los paneles, de arriba hacia abajo, representan a x , y y z respectivamente.

Para complementar la información temporal del efector final, se muestra el comportamiento de las orientaciones en la Figura 5.23. Como se puede observar, de forma análoga a la posición, después de variaciones generadas por la interacción con el objeto, los ángulos de las orientaciones tienden a convergir a los estados deseados de forma amortiguada. El efecto generado por la inestabilidad del sistema en posición, también es mostrado en la orientación, donde la evolución se corta de manera súbita. Los errores en estado estable, para el sistema controlado en impedancia, son 5.22×10^{-4} m, 1.96×10^{-4} m y 5.69×10^{-4} m para ζ , ϕ y ϕ respectivamente.

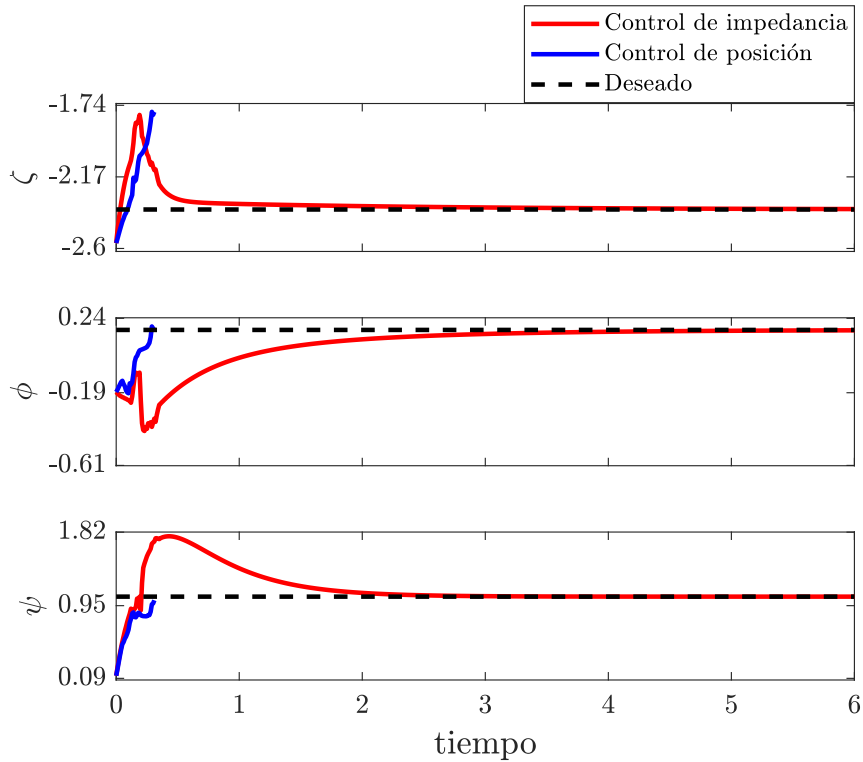


FIGURA 5.23: Comportamiento temporal de la orientación del robot controlado para los parámetros K_P , K_D , D_{d2} y K_{d2} , cuando este interactúa con un objeto en su trayectoria. Los paneles, de arriba hacia abajo, representan la orientación parametrizada por los ángulos de Euler ζ , ϕ y ψ respectivamente.

El esfuerzo generado por cada junta, para el sistema controlado por ambas formulaciones, se muestra en la Figura 5.24. Como se observa, para el sistema controlado por dinámica inversa, se muestra un comportamiento de saturación que cambia de límites superior a inferior súbitamente y de manera continua durante la interacción del con el objeto. Por otro lado, se observa que cuando el sistema es controlado en impedancia, a pesar de experimentar saturación en sus actuadores, estos tienden a recuperarse y volver a un comportamiento normal, dentro de los rangos físicos del sistema.

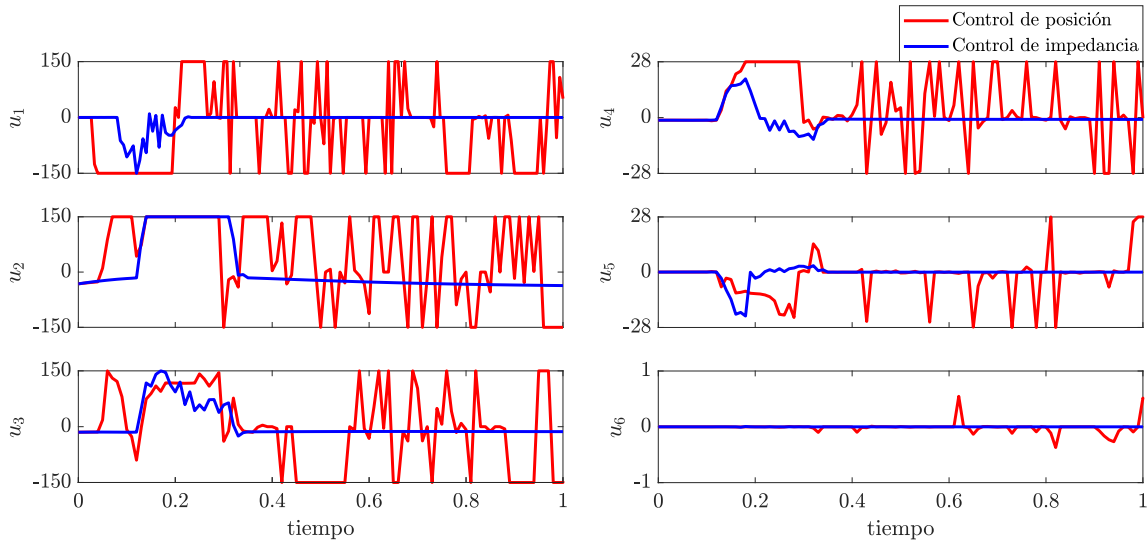


FIGURA 5.24: Comportamiento temporal del esfuerzo en cada junta del robot controlado para los parámetros K_P , K_D , D_{d2} y K_{d2} , cuando este interactúa con un objeto en su trayectoria. La variable u_i representa el esfuerzo aplicado por la junta i , donde este subíndice toma valores de 1 a 6 para juntas consecutivas de la base hacia el efector final.

5.2.3 Respuesta con interacción e impedancia de rigidez alta

Las simulaciones son realizadas para el robot controlado en posición por dinámica inversa, con parámetros K_P y K_D , y para el robot controlado por impedancia, con parámetros D_{d3} y K_{d3} .

En la Figura 5.25 se muestra el comportamiento del efector final, en el espacio Cartesiano. Como se observa, bajo estas condiciones, cuando el robot interactúa con un objeto en su trayectoria, sin importar el tipo de controlador, este se inestabiliza, por lo que el robot no consigue llegar a la pose deseada.

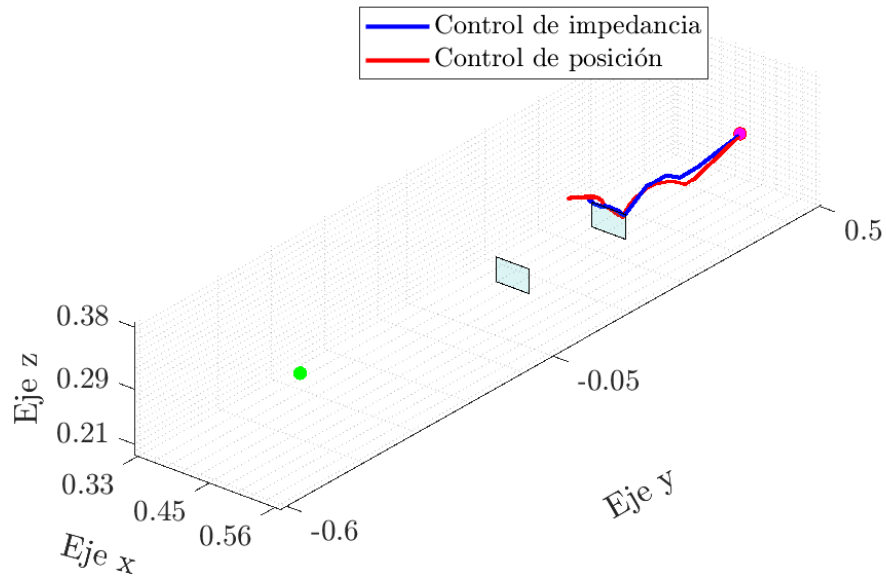


FIGURA 5.25: Comportamiento controlado del efector final en el espacio cartesiano para los parámetros K_P , K_D , D_{d3} y K_{d3} . El robot parte de la posición inicial (magenta), hacia la posición deseada (verde). Los planos (esmeralda) representan la posición de un objeto en el espacio.

El comportamiento temporal del efector final en cada eje Cartesiano, mostrado en la Figura 5.27, permite observar una variación súbita en el comportamiento del robot. Independientemente del controlador, el comportamiento en cada eje es consistente con el comportamiento espacial, mostrado en la Figura 5.25. Cuando el robot interactúa durante su trayectoria, el simulador se inestabiliza, iniciando una variación brusca.

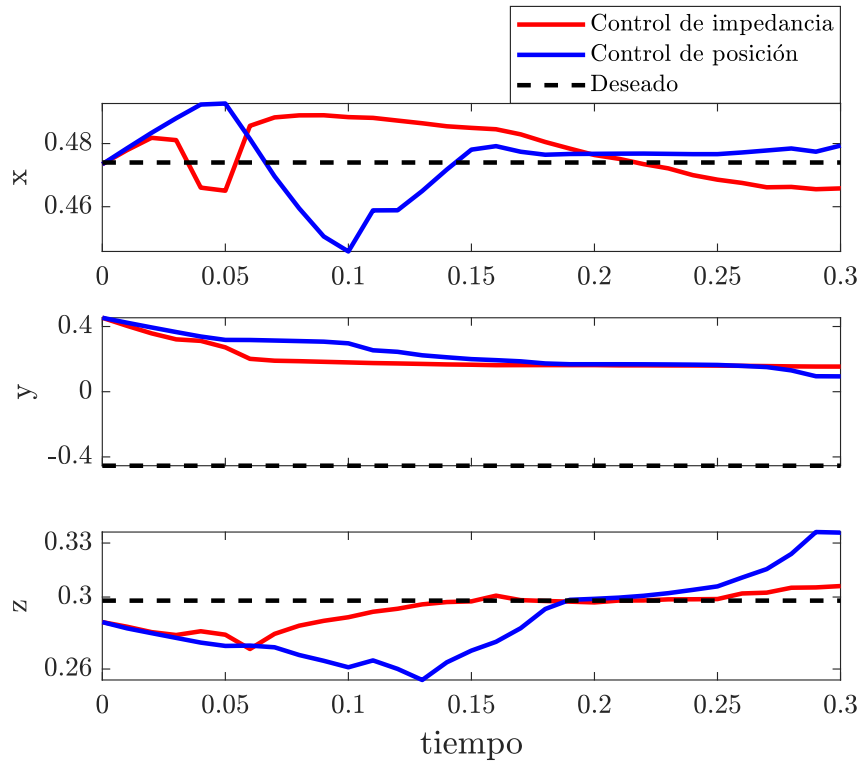


FIGURA 5.26: Comportamiento temporal del efector final en los ejes Cartesianos para el robot controlado para los parámetros K_P , K_D , D_{d3} y K_{d3} . cuando este interactúa con un objeto en su trayectoria. Los paneles, de arriba hacia abajo, representan a x , y y z respectivamente.

De manera similar a la posición, el comportamiento en orientación responde a la inestabilidad generada. Este comportamiento se muestra en la Figura 5.27. Como se puede observar, a partir de la interacción, se genera un cambio brusco en los ángulo de orientación, de forma consistente con el comportamiento del robot en posición.

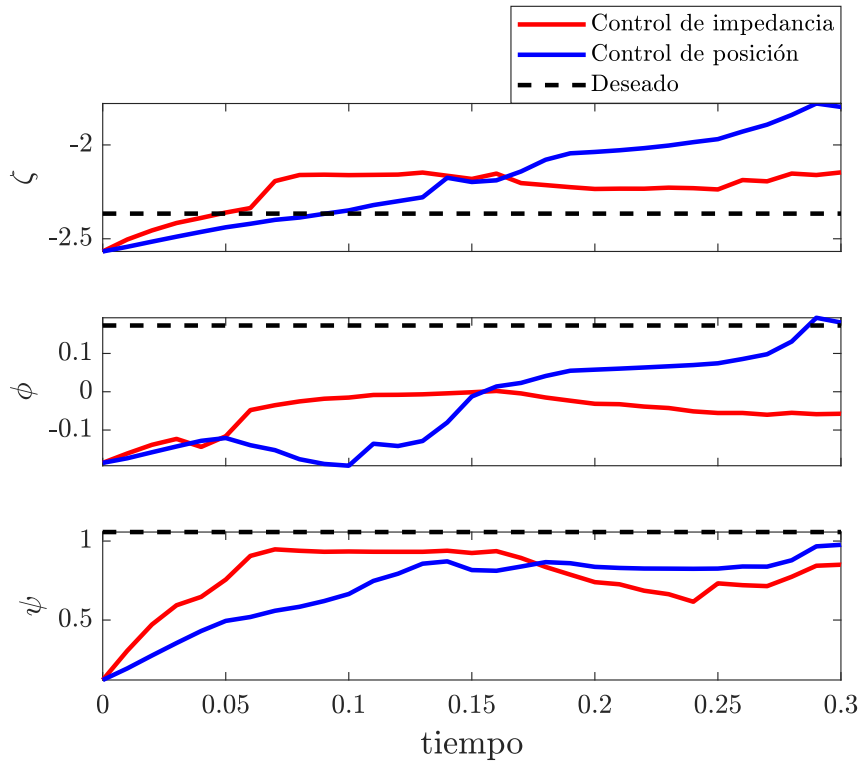


FIGURA 5.27: Comportamiento temporal de la orientación del robot controlado para los parámetros K_P , K_D , D_{d3} y K_{d3} , cuando este interactúa con un objeto en su trayectoria. Los paneles, de arriba hacia abajo, representan la orientación parametrizada por los ángulos de Euler ζ , ϕ y ψ respectivamente.

El esfuerzo aplicado por cada junta es mostrado en la Figura 5.28. En esta se observa que, en los casos de control por dinámica inversa e impedancia, cuando el robot interactúa con el objeto, las juntas son llevadas a sus límites físicos, y no se recuperan por el resto de la simulación.

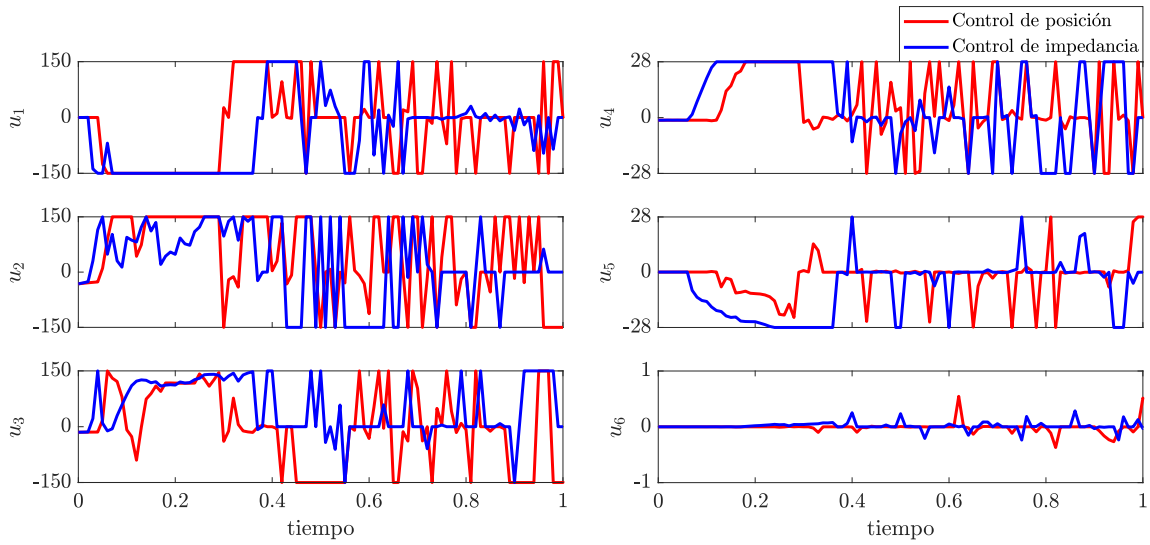


FIGURA 5.28: Comportamiento temporal del esfuerzo en cada junta del robot controlado para los parámetros K_P , K_D , D_{d3} y K_{d3} , cuando este interactúa con un objeto en su trayectoria. La variable u_i representa el esfuerzo aplicado por la junta i , donde este subíndice toma valores de 1 a 6 para juntas consecutivas de la base hacia el efector final.

5.3 Análisis de errores

En la Tabla 5.1 se muestra un resumen de los errores en el espacio Cartesiano para los distintos casos mostrados en la sección previa. El error hallado es definido como la distancia euclidiana desde la posición en estado estacionario del efector final hasta la posición deseada. Los valores mostrados corresponden únicamente al error hallado para las posiciones debido a que las orientaciones presentan un comportamiento acorde a estos.

	Control de impedancia			Control por
	Rigidez baja	Rigidez media	Rigidez alta	dinámica inversa
$\ \vec{p} - \vec{p}_d\ $ sin obstáculo	6.01×10^{-4} m	1.81×10^{-4} m	2.63×10^{-6} m	3.78×10^{-5} m
$\ \vec{p} - \vec{p}_d\ $ con obstáculo	0.02 m	4.47×10^{-4} m	Inestable	Inestable

TABLA 5.1: Distancia euclidiana entre la posición del efector final \vec{p} en estado estacionario y la posición deseada \vec{p}_d .

Los errores obtenidos muestran que para un ambiente sin obstáculos, la respuesta del robot controlado por dinámica inversa o por impedancia permite la convergencia de su efector final a la posición deseada. Sin embargo, en un ambiente donde el robot interactúa durante su trayectoria, el sistema controlado por dinámica inversa se inestabiliza, por lo que no llega a la posición deseada. En el mismo escenario, cuando el sistema es controlado por impedancia, se desarrollan tres comportamientos dependiendo de la definición de su impedancia. Cuando la rigidez de la impedancia deseada es baja, se da la convergencia al estado deseado con un error significativo. De forma contraria, cuando la impedancia se define con una rigidez alta, la interacción genera la aplicación de una fuerza muy grande que genera la inestabilidad del sistema. Finalmente, cuando la impedancia deseada es definida por una rigidez adecuada, la convergencia es asegurada con un error semejante a cuando el sistema es controlado sin obstáculos. Las descripciones cualitativas de los resultados obtenidos son resumidas en la Tabla 5.2, donde se sigue el formato mostrado en la Tabla 5.1.

	Control de impedancia			Control por dinámica inversa
	Rigidez baja	Rigidez media	Rigidez alta	
Sin obstáculo	El efector final converge al estado deseado con un error despreciable, manteniendo los esfuerzos de control dentro de los límites de los actuadores.	El efector final converge al estado deseado con un error despreciable, manteniendo los esfuerzos de control dentro de los límites de los actuadores.	El efector final converge al estado deseado con un error despreciable, manteniendo los esfuerzos de control dentro de los límites de los actuadores.	El efector final converge al estado deseado con un error despreciable, manteniendo los esfuerzos de control dentro de los límites de los actuadores.
Con obstáculo	El efector final converge al estado deseado con un error considerable, manteniendo los esfuerzos de control dentro de los límites de los actuadores.	El efector final converge al estado deseado con un error despreciable, manteniendo los esfuerzos de control dentro de los límites de los actuadores.	El efector final no converge al estado deseado y el esfuerzo de control en los actuadores se satura debido a los límites en estos. El simulador se inestabiliza durante la colisión entre robot y el obstáculo.	El efector final no converge al estado deseado y el esfuerzo de control en los actuadores se satura debido a los límites en estos. El simulador se inestabiliza durante la colisión entre robot y el obstáculo.

TABLA 5.2: Resumen cualitativo del comportamiento obtenido en cada caso estudiado.

CONCLUSIONES

El manipulador robótico UR5 es controlado para alcanzar establemente una pose deseada. Con este fin, se diseña un sistema de control que permite su convergencia a un estado final específico con errores pequeños en términos de la distancia euclidiana. La precisión en su comportamiento es asegurada para escenarios donde el robot se mueve sin restricciones externas, así como cuando este interactúa con objetos de superficie irregular. El controlador se define de modo que la interacción intermedia en su trayectoria emule el comportamiento de una impedancia mecánica lineal.

Para escenarios en los que el robot puede moverse libremente, el controlador por impedancia presenta un desempeño similar a un control por posición clásico, permitiendo su convergencia a la pose deseada con errores en el orden de los micrómetros. En escenarios donde el robot interactúa con su entorno durante su trayecto, el control de posición tiende a inestabilizarse, mientras que el comportamiento del controlador de impedancia varía dependiendo de los parámetros que lo describen. Para una impedancia deseada muy rígida, el robot se comporta de manera similar a cuando es controlado en posición, por lo que no converge a la pose final deseada. Una reducción cuantiosa de la rigidez deseada, permite que el robot interactúe de forma delicada sin convergir a la posición deseada. La elección de ganancias adecuadas para la impedancia deseada permitieron que el robot interactúe de forma delicada durante su trayectoria y además converja al estado deseado con una distancia euclidiana en el orden de los milímetros, desempeño apropiado para los objetivos de este trabajo de tesis.

TRABAJO A FUTURO

Con la finalidad de mejorar el desempeño general del controlador, se propone que a futuro se realice la medición *in situ* de parámetros implicados en las descripciones cinemática y dinámica del robot. Esto permitirá que el robot sea descrito matemáticamente de una forma más fiel que usando los parámetros provistos por el fabricante en los archivos URDF. Por otro lado, el desarrollo del controlador podría extenderse con una etapa que permita calcular el factor de rigidez de una forma adaptativa. Esto permitirá regular la impedancia del controlador como función de variaciones en la superficie de contacto. De igual forma, se propone el cambio del esquema de acondicionamiento de la señal de control a algún esquema moderno con menos fuentes de error, esto puede conseguirse usando esquemas basados en aprendizaje automático. Para el desarrollo de un controlador que lidie con fenómenos apegados a la realidad, se propone incluir tiempos muertos, pérdida de información y modelos de impedancia no lineales. Con el fin de obtener resultados más óptimos con respecto al control de posición, se propone la inclusión del esquema de control por modos deslizantes, que de acuerdo a la literatura, tiene errores despreciables en estado estable. Por último, queda pendiente la implementación y prueba del controlador en un escenario real donde el robot interactúe con superficies irregulares.

REFERENCIAS BIBLIOGRÁFICAS

- [1] J. G. C. Devol, “Programmed article transfer,” Jun. 13 1961, uS Patent 2,988,237.
- [2] G. Hirzinger, N. Sporer, A. Albu-Schaffer, M. Hahnle, R. Krenn, A. Pascucci, and M. Schedl, “Dlr’s torque-controlled light weight robot iii-are we reaching the technological limits now?” in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, vol. 2. IEEE, 2002, pp. 1710–1716.
- [3] U. Robots, “Ur5 technical specifications,” 2016.
- [4] S. Erkaya, “Investigation of joint clearance effects on welding robot manipulators,” *Robotics and Computer-Integrated Manufacturing*, vol. 28, no. 4, pp. 449–457, 2012.
- [5] K. Harada, T. Foissotte, T. Tsuji, K. Nagata, N. Yamanobe, A. Nakamura, and Y. Kawai, “Pick and place planning for dual-arm manipulators,” in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 2281–2286.
- [6] F. Chen, H. Zhao, D. Li, L. Chen, C. Tan, and H. Ding, “Contact force control and vibration suppression in robotic polishing with a smart end effector,” *Robotics and Computer-Integrated Manufacturing*, vol. 57, pp. 391–403, 2019.
- [7] H. Su, J. Sandoval, P. Vieyres, G. Poisson, G. Ferrigno, and E. De Momi, “Safety-enhanced collaborative framework for tele-operated minimally invasive surgery using a 7-dof torque-controlled robot,” *International Journal of Control, Automation and Systems*, vol. 16, no. 6, pp. 2915–2923, 2018.

- [8] M. S. Menon, V. Ravi, and A. Ghosal, “Trajectory planning and obstacle avoidance for hyper-redundant serial robots,” *Journal of Mechanisms and Robotics*, vol. 9, no. 4, 2017.
- [9] H. Wu, S. Luo, H. Lin, S. Duan, Y. Guan, and J. Rojas, “Recovering from external disturbances in online manipulation through state-dependent revertive recovery policies,” in *2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 2018, pp. 166–173.
- [10] C. Schindlbeck and S. Haddadin, “Unified passivity-based cartesian force/impedance control for rigid and flexible joint robots via task-energy tanks,” in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 440–447.
- [11] K.-S. Hwang, Y.-Y. Cho, and W.-C. Jiang, “A visual servoing with collision avoidance mechanism for redundant manipulators,” in *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2018, pp. 1821–1826.
- [12] J. Pomares, F. Chaumette, and F. Torres, “Adaptive visual servoing by simultaneous camera calibration,” in *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE, 2007, pp. 2811–2816.
- [13] P. Corke, *Robotics, vision and control: fundamental algorithms in MATLAB® second, completely revised*. Springer, 2017, vol. 118.
- [14] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010.
- [15] C. Ganesh, G. Dietz, and J. Jambor, “Ultrasonic sensor-based motion control for robotic manipulators,” in *Conference Proceedings., IEEE International Conference on Systems, Man and Cybernetics*. IEEE, 1989, pp. 796–797.

- [16] K. Ohno, T. Nomura, and S. Tadokoro, “Real-time robot trajectory estimation and 3d map construction using 3d camera,” in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2006, pp. 5279–5285.
- [17] R. Rossi, A. Santamaria-Navarro, J. Andrade-Cetto, and P. Rocco, “Trajectory generation for unmanned aerial manipulators through quadratic programming,” *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 389–396, 2016.
- [18] A. Albu-Schaffer, C. Ott, U. Frese, and G. Hirzinger, “Cartesian impedance control of redundant robots: Recent results with the dlr-light-weight-arms,” in *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, vol. 3. IEEE, 2003, pp. 3704–3709.
- [19] H. Chaudhary, V. Panwar, R. Prasad, and N. Sukavanam, “Adaptive neuro fuzzy based hybrid force/position control for an industrial robot manipulator,” *Journal of Intelligent Manufacturing*, vol. 27, no. 6, pp. 1299–1308, 2016.
- [20] A. K. Ravandi, E. Khanmirza, and K. Daneshjou, “Hybrid force/position control of robotic arms manipulating in uncertain environments based on adaptive fuzzy sliding mode control,” *Applied Soft Computing*, vol. 70, pp. 864–874, 2018.
- [21] M. Keppler, D. Lakatos, C. Ott, and A. Albu-Schaffer, “Elastic structure preserving impedance ($es\pi$) control for compliantly actuated robots,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 5861–5868.
- [22] L. Han, L. Yan, and W. Xu, “A lightweight redundant manipulator with high stable wireless communication and compliance control,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 6622–6627.
- [23] L. Yan, Z. Mu, W. Xu, and B. Yang, “Coordinated compliance control of dual-arm robot for payload manipulation: Master-slave and shared force control,” in

2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).
IEEE, 2016, pp. 2697–2702.

- [24] E. Mariotti, E. Magrini, and A. De Luca, “Admittance control for human-robot interaction using an industrial robot equipped with a f/t sensor,” in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 6130–6136.
- [25] A. Q. Keemink, H. van der Kooij, and A. H. Stienen, “Admittance control for physical human–robot interaction,” *The International Journal of Robotics Research*, vol. 37, no. 11, pp. 1421–1444, 2018.
- [26] A. De Luca and R. Mattone, “Sensorless robot collision detection and hybrid force/motion control,” in *Proceedings of the 2005 IEEE international conference on robotics and automation*. IEEE, 2005, pp. 999–1004.
- [27] B. Yao, Z. Zhou, L. Wang, W. Xu, Q. Liu, and A. Liu, “Sensorless and adaptive admittance control of industrial robot in physical human- robot interaction,” *Robotics and Computer-Integrated Manufacturing*, vol. 51, pp. 158–168, 2018.
- [28] M. Naveau, J. Carpentier, S. Barthelemy, O. Stasse, and P. Soueres, “Metapod—template meta-programming applied to dynamics: Cop-com trajectories filtering,” in *2014 IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2014, pp. 401–406.
- [29] J. Lee, M. Grey, S. Ha, T. Kunz, S. Jain, Y. Ye, S. Srinivasa, M. Stilman, and C. Liu, “Dart: Dynamic animation and robotics toolkit,” *Journal of Open Source Software*, vol. 3, no. 22, p. 500, 2018.
- [30] J. Carpentier, F. Valenza, N. Mansard *et al.*, “Pinocchio: fast forward and inverse dynamics for poly-articulated systems,” <https://stack-of-tasks.github.io/pinocchio>, 2015–2019.

- [31] M. L. Felis, “Rbd: an efficient rigid-body dynamics library using recursive algorithms,” *Autonomous Robots*, vol. 41, no. 2, pp. 495–511, 2017.
- [32] “Havok physicstm,” Último Acceso el 07-05-2020. [Online]. Available: <https://www.havok.com/products/havok-physics/>
- [33] E. Coumans *et al.*, “Bullet physics library,” *Open source: bulletphysics.org*, vol. 15, no. 49, p. 5, 2013.
- [34] R. Smith *et al.*, “Open dynamics engine,” 2005.
- [35] E. Todorov, T. Erez, and Y. Tassa, “Mujoco: A physics engine for model-based control,” in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 5026–5033.
- [36] K. Hauser, “Intelligent motion laboratory at uiuc,” Último Acceso el 07-05-2020. [Online]. Available: <http://www.klampt.org/>
- [37] E. Rohmer, S. P. Singh, and M. Freese, “Coppeliassim (formerly v-rep): A versatile and scalable robot simulation framework,” in *Proc. Int. Conf. Intell. Robots Syst.*, 2013, pp. 1321–1326.
- [38] N. Koenig and A. Howard, “Design and use paradigms for gazebo, an open-source multi-robot simulator,” in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 3. IEEE, 2004, pp. 2149–2154.
- [39] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot modeling and control*. John Wiley & Sons, 2020.
- [40] S. R. Buss, “Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods,” *IEEE Journal of Robotics and Automation*, vol. 17, no. 1-19, p. 16, 2004.

- [41] J.-J. E. Slotine and W. Li, "On the adaptive control of robot manipulators," *The international journal of robotics research*, vol. 6, no. 3, pp. 49–59, 1987.
- [42] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE Journal on Robotics and Automation*, vol. 3, no. 1, pp. 43–53, 1987.
- [43] N. Hogan, "Impedance control: An approach to manipulation," in *1984 American control conference*. IEEE, 1984, pp. 304–313.
- [44] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.
- [45] C. E. Agüero, N. Koenig, I. Chen, H. Boyer, S. Peters, J. Hsu, B. Gerkey, S. Paepcke, J. L. Rivero, J. Manzo *et al.*, "Inside the virtual robotics challenge: Simulating real-time robotic disaster response," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 2, pp. 494–506, 2015.
- [46] "Ur5 collaborative robot arm: Flexible and lightweight robot arm," Último Acceso el 07-05-2020. [Online]. Available: <https://www.universal-robots.com/products/ur5-robot/>

APÉNDICE

A. Archivos para la ejecución de las simulaciones

Los archivos usados en el desarrollo de este trabajo de tesis se encuentran en el repositorio [utec_tesis](https://github.com/terdegrede/utec_tesis). La distribución de archivos en este repositorio sigue el orden de un workspace de ROS, teniendo como carpetas principales: `build`, `devel`, `install` y `src`. Para clonar el repositorio haciendo uso de Git, se debe seguir la sintaxis habitual en el *Command Line* de Linux:

```
git clone https://github.com/terdegrede/utec_tesis.git
```

Para la reproducción de este trabajo, se debe tomar en cuenta que los nodos y las simulaciones fueron desarrollados en la distribución Ubuntu 16.04, por lo que se debe contar con ROS Kinetic Kame.

Los archivos relevantes para las simulaciones realizadas en este trabajo de tesis son descritas a continuación:

- `/test_imp/src/scripts_tesis/init_pos`: Este nodo permite que el robot se traslade, partiendo de su estado inicial predeterminado, de modo que el efector final se encuentre en posiciones y orientaciones específicas. La posición y orientación deseada se define en un vector de seis elementos, bajo el formato de un `numpy.array`.

- `/test_imp/src/scripts_tesis/ID_OS_controller`: Este nodo permite que el robot se traslade de modo que el efector final vaya de posiciones y orientaciones iniciales, a orientaciones y posiciones deseadas. Al usar este nodo, la transición de estados estará comandada bajo el controlador de posición por dinámica inversa descrito en el trabajo de tesis. Para evitar efectos generados por velocidades o aceleraciones no deseadas, se recomienda que el robot sea trasladado a una posición inicial usando el nodo `init_pos` y que el control sea realizado desde ese punto. Dado que este nodo es editable, los valores de rigidez pueden ser modificados internamente cambiando los parámetros K_d y K_p . Este nodo permite exportar valores articulares y cartesianos en archivos `.dat`.

- `/test_imp/src/scripts_tesis/Imp_OS_controller`: Este nodo permite que el robot se traslade de modo que el efector final vaya de posiciones y orientaciones iniciales, a orientaciones y posiciones deseadas. Al usar este nodo, la transición de estados estará comandada bajo el controlador de impedancia descrito en el trabajo de tesis. Al igual que el nodo de control de posición, para evitar efectos generados por velocidades o aceleraciones no deseadas, se recomienda que el robot sea trasladado a una posición inicial usando el nodo `init_pos` y que el control sea realizado desde ese punto. Dado que este nodo es editable, los valores de rigidez pueden ser modificados internamente cambiando los parámetros M_d y K_d . Este nodo permite exportar valores articulares y cartesianos en archivos `.dat`.

- `/test_imp/src/scripts_tesis/funcs2`: Este archivo contiene funciones usadas por los nodos `Imp_OS_controller` y `ID_OS_controller`. Ambos nodos tienen declarada la función `import` internamente.

- `/universal_robot/ur_gazebo/launch/workspace_tesis.world`: Este archivo contiene la descripción espacial del espacio de trabajo. Así, en este archivo se encuentra definida la posición fija del robot y la posición del objeto de superficie irregular con el que interactúa.

- `/universal_robot/ur_gazebo/launch/workspace_tesis.launch`: Este archivo xml enlaza el archivo `.world` para que pueda ser ejecutado en Gazebo haciendo uso del comando `roslaunch` de ROS.
- `/controller/arm_controller_ur5_effort.yaml`: Este es un archivo que contiene controladores de esfuerzo internos para cada junta del robot. Cada junta es controlada bajo un esquema PID, por lo que las ganancias pueden ser modificadas internamente. Este es un archivo externo creado exclusivamente durante el desarrollo del controlador por impedancia de este trabajo de tesis.
- `/universal_robot/ur_description/urdf/*`: Estos archivos describen la mecánica del robot, que implica efectos inerciales, colisiones, enmallados y geometrías. El archivo URDF es necesario para hacer uso de la librería RBDL, incluida en los nodos `Imp_OS_controller` y `ID_OS_controller`.