



UNIVERSIDAD
DE MÁLAGA



ESCUELA DE INGENIERÍAS INDUSTRIALES

Departamento

Ingeniería de Sistemas y Automática

Área de Conocimiento

Ingeniería de sistemas y automática

Lenguajes y sistemas informáticos

TRABAJO FIN DE GRADO

CONTROL DE MOVIMIENTO DE UN INSTRUMENTO QUIRÚRGICO ENDOWRIST MEDIANTE UN DISPOSITIVO HÁPTICO

Grado en Ingeniería Electrónica, Robótica y Mecatrónica

Autor: Pilar Valencia Castro

Tutor: Irene Rivas Blanco

Cotutor: Álvaro Galán Cuenca

MÁLAGA, Septiembre de 2023

AGRADECIMIENTOS

A mi familia por tener fe ciega en mi.

RESUMEN

En los últimos años, la cirugía asistida por plataformas robóticas ha generado un creciente interés, gracias a los numerosos beneficios que presenta tanto para los pacientes como para el personal sanitario. Esto ha llevado a que diversos grupos de investigación estén centrando sus esfuerzos en avanzar en este campo.

El presente trabajo de fin de grado se ha realizado dentro del grupo de investigación de robótica médica de la Universidad de Málaga, que trabaja en el desarrollo de plataformas robóticas para aplicaciones de robótica quirúrgica. En este sentido, el objetivo general de este trabajo es el diseño e implementación del control de movimiento de una herramienta quirúrgica robótica acoplada al efector final de un brazo robótico. Esta herramienta, llamada *EndoWrist*, es un tipo de herramienta especialmente diseñada para la plataforma robótica comercial *Da Vinci*, y presenta un gran interés gracias a su punta articulada, la cual aumenta el número de grados de libertad de movimiento respecto a las herramientas convencionales.

En este trabajo se ha utilizado un brazo robótico UR3, de la empresa *Universal Robots*, como asistente quirúrgico, así como el dispositivo háptico *Phantom Omni* para comandar el movimiento de la herramienta. El control de los grados de libertad adicionales de la herramienta se ha realizado mediante un sistema de motores controlado con una placa Arduino. Finalmente, todo el software desarrollado se ha integrado en el entorno ROS.

Para el control de la herramienta a través de este dispositivo háptico se trabajará sobre el control de movimiento actual realizado en el entorno de ROS, y se programará para trasladar los movimientos realizados con el dispositivo háptico en el desplazamiento que se espera que realice la herramienta. Para ello, se cuenta con plataformas tales como MATLAB, Arduino IDE, y ROS como entorno de comunicación entre todos los elementos.

El objetivo final será conseguir mover la herramienta según el movimiento del dispositivo háptico respetando el punto de fulcro y, además, conseguir controlar los cuatro grados de libertad adicionales que ofrece la herramienta *EndoWrist*.

ABSTRACT

In recent years, robot-assisted surgery has generated increasing interest, thanks to the numerous benefits it offers both patients and healthcare personnel. This has led various research groups to focus their efforts on advancing in this field.

This final degree project was conducted within the medical robotics research group at the University of Malaga, which works on the development of robotic platforms for surgical robotics applications. In this context, the overall objective of this project is the design and implementation of motion control for a robotic surgical tool attached to the end effector of a robotic arm. This tool, called *EndoWrist*, is a type of specially designed tool for the commercial robotic platform *Da Vinci* and is of great interest due to its articulated tip, which increases the number of degrees of freedom of movement compared to conventional tools.

In this project, a UR3 robotic arm from *Universal Robots* has been used as a surgical assistant, along with the haptic device *Phantom Omni* to command the movement of the tool. The control of the additional degrees of freedom of the tool has been carried out through a motor system controlled by an Arduino board. Finally, all the developed software has been integrated into the ROS environment.

For the control of the tool through this haptic device, we will work on the current motion control implemented in the ROS environment and we will program it to translate the movements made with the haptic device into the expected displacement of the tool. To do this, we have platforms such as MATLAB, Arduino IDE, and ROS as the communication environment between all elements.

The ultimate goal is to move the tool according to the movement of the haptic device while respecting the fulcrum point and, furthermore, to control the four additional degrees of freedom that the *EndoWrist* tool offers.

PALABRAS CLAVE

Robótica quirúrgica, *EndoWrist*, dispositivo háptico, grado de libertad, *roll*, *pitch*, *yaw*, *jaw*, PWM, punto de fulcro, ROS, algoritmo

ÍNDICE GENERAL

1. Introducción	1
1.1. Contexto de la robótica quirúrgica	2
1.2. Antecedentes de este proyecto	3
1.3. Objetivo del proyecto	4
1.4. Estructura de la memoria	5
2. Descripción del Sistema	7
2.1. Brazo robótico	8
2.2. Instrumento EndoWrist	8
2.3. Acople entre el robot y la herramienta	10
2.4. Dispositivo háptico	14
2.4.1. Twee	15
2.5. Conclusiones	16
3. Control de Movimiento de la Herramienta	17
3.1. Estudio previo de la herramienta	18
3.2. Primitivas de movimiento de cada grado de libertad	19
3.2.1. Roll	20
3.2.2. Pitch	22
3.2.3. Yaw	24
3.2.4. Jaw	26

3.3. Pruebas y cálculo de errores	28
3.3.1. Pruebas	28
3.3.2. Cálculo de errores	33
3.4. Conclusiones	36
4. Esquema de Control de Movimiento	37
4.1. Modelo de la tarea	38
4.2. Esquema de control	42
4.2.1. Transformación Inicial	43
4.2.2. Control del Punto de Fulcro	44
4.2.3. Cinemática Inversa	46
4.2.4. Control de los Motores	47
4.3. Conclusiones	48
5. Implementación del Algoritmo de Control	49
5.1. Introducción a ROS	50
5.2. Desarrollo por bloques	52
5.2.1. Dispositivo háptico	53
5.2.2. Brazo robótico	54
5.2.3. Algoritmo de control	56
5.3. Conclusiones	57
6. Puesta en Marcha y Resultados	59
6.1. Movimiento del instrumento	60
6.2. Movimiento del robot	64
6.2.1. Prueba en simulación	64

- 6.2.2. Prueba en el robot real 66
- 6.3. Movimiento del conjunto del robot con la herramienta 67
- 6.4. Conclusiones 70
- 7. Conclusiones Finales y Futuras Líneas de Trabajo 71**
 - 7.1. Conclusiones finales 72
 - 7.2. Futuras líneas de trabajo 72
- Referencias bibliográficas 74**

ÍNDICE DE FIGURAS

1.1. Plataforma robótica <i>Da Vinci</i>	2
2.1. Brazo robótico UR3	8
2.2. Instrumento <i>EndoWrist</i>	9
2.3. Discos de la herramienta	9
2.4. Motor DC Micro Metal	10
2.5. Arduino UNO	10
2.6. Mecanismo contenedor de los motores	11
2.7. Mecanismo de acople entre la caja de motores y el robot	11
2.8. Acople final entre la herramienta y el brazo robótico	12
2.9. Nuevas piezas de acople para la herramienta	12
2.10. Ensamblado de las piezas para el montaje final	13
2.11. Nuevo mecanismo de acople entre la caja de motores y el robot	13
2.12. Acople final del sistema	14
2.13. Phantom Omni	15
2.14. Twee	15
3.1. Movimiento de la herramienta	18
3.2. Movimiento del grado de libertad <i>roll</i>	20
3.3. Gráfica de los experimentos para el <i>roll</i>	20
3.4. Interpolación polinomial para el <i>roll</i>	21

3.5. Movimiento del grado de libertad <i>pitch</i>	22
3.6. Gráfica de los experimentos para el <i>pitch</i>	22
3.7. Interpolación polinomial para el <i>pitch</i>	23
3.8. Movimiento del grado de libertad <i>yaw</i>	24
3.9. Gráfica para los experimentos del <i>yaw</i>	24
3.10. Interpolación polinomial para el <i>yaw</i>	25
3.11. Movimiento del grado de libertad <i>jaw</i>	26
3.12. Gráfica de los experimentos del <i>jaw</i>	26
3.13. Interpolación polinomial para el <i>jaw</i>	27
3.14. Interfaz para las pruebas de la herramienta	28
3.15. Movimiento de 90° del <i>roll</i>	29
3.16. Movimiento de 40° del <i>pitch</i>	30
3.17. Movimiento de 45° del <i>yaw</i>	31
3.18. Movimiento de 30° del <i>jaw</i>	32
3.19. Error para el <i>roll</i>	33
3.20. Error para el <i>pitch</i>	34
3.21. Error para el <i>yaw</i>	35
3.22. Error para el <i>jaw</i>	35
4.1. Dibujo del sistema simplificado	38
4.2. Conjunto de sistemas de referencia en los dispositivos del laboratorio . .	39
4.3. Relaciones de transformación del sistema	40
4.4. Medidas de la herramienta con respecto al efector final del robot	41
4.5. Esquema de control	42
4.6. Relación entre la base del dispositivo háptico y la muñeca de la herramienta	43

4.7. Grados de libertad relacionados con el Phantom	47
5.1. Nodos y topics del sistema	52
5.2. Comando <i>rostopic echo /phantom/pose</i> en la terminal del ordenador	53
5.3. Comando <i>rostopic echo /phantom/joint_states</i> en la terminal del ordenador	53
5.4. Comando <i>rostopic echo /phantom/twee</i> en la terminal del ordenador	54
5.5. Comando <i>rostopic echo /tf</i> en la terminal del ordenador	54
5.6. Comando <i>rostopic echo /joint_states</i> en la terminal del ordenador	55
5.7. Comando <i>rostopic echo /ur_hardware_interface/script_command</i> en la terminal	55
5.8. Código correspondiente al <i>roll</i> en Arduino IDE	57
6.1. Eje correspondiente al <i>roll</i> en el dispositivo háptico	60
6.2. Movimiento del <i>roll</i> con el lápiz	61
6.3. Movimiento del <i>pitch</i> en el dispositivo háptico	61
6.4. Movimiento del <i>pitch</i> con el lápiz	62
6.5. Movimiento del <i>yaw</i> en el dispositivo háptico	62
6.6. Movimiento del <i>yaw</i> con el lápiz	63
6.7. Movimiento del <i>jaw</i> en el Twee	63
6.8. Movimiento del <i>jaw</i> con el lápiz	64
6.9. Posición inicial del robot en simulación	65
6.10. Simulación del movimiento del robot	66
6.11. Movimiento real del robot	67
6.12. Falso abdomen del paciente	68
6.13. Movimiento del robot en el abdomen	69
6.14. Movimiento de la pinza en el abdomen	69

ÍNDICE DE CUADROS

3.1. Relación entre los discos y los grados de libertad de la herramienta . . .	19
4.1. Sistemas de referencia del conjunto	38
4.2. Recopilación de ecuaciones de movimiento para el <i>roll</i> , <i>pitch</i> , <i>yaw</i> y <i>jaw</i> .	48

INTRODUCCIÓN

En este primer capítulo de la memoria se presenta la robótica quirúrgica como objeto de estudio, se recopilan los antecedentes de este proyecto y se plantean los objetivos del proyecto final con el fin de acatarlos de uno en uno a lo largo de toda la memoria.

Finalmente, se presenta un resumen de los capítulos del presente proyecto y una breve descripción de cada uno de ellos.

1.1 CONTEXTO DE LA ROBÓTICA QUIRÚRGICA

La robótica quirúrgica es una rama innovadora de la medicina que combina la tecnología robótica avanzada con la experiencia y habilidades de los cirujanos. Esta disciplina ha revolucionado la forma en que se realizan ciertos procedimientos quirúrgicos, ofreciendo beneficios significativos tanto para los médicos como para los pacientes.

El contexto de la robótica quirúrgica se basa en el desarrollo de sistemas robóticos altamente sofisticados que permiten a los cirujanos realizar intervenciones de manera más precisa, menos invasiva y con mayor control. Estos sistemas están compuestos por una consola de control, brazos robóticos y una cámara de alta definición, todo ello conectado a través de una plataforma informática avanzada. Un ejemplo de sistema robótico es la plataforma *Da Vinci* [1] y sus herramientas *EndoWrist* [2] (figura 1.1).



Figura 1.1: Plataforma robótica *Da Vinci*

Una de las principales ventajas de la robótica quirúrgica es su capacidad para realizar cirugías mínimamente invasivas. A través de pequeñas incisiones, los brazos robóticos equipados con instrumentos quirúrgicos de precisión pueden acceder a áreas del cuerpo con mayor facilidad y menor traumatismo para los tejidos circundantes. Esto se traduce en una recuperación más rápida, menos dolor postoperatorio y cicatrices más pequeñas para los pacientes.

Además, la robótica quirúrgica permite una mayor precisión y destreza en los movimientos del cirujano. Los brazos robóticos eliminan los temblores naturales de la mano humana y ofrecen una escala de movimiento más amplia, lo que permite una mejor manipulación de los instrumentos quirúrgicos. Los sistemas también proporcionan una visión en 3D de alta definición, lo que mejora la percepción espacial del

cirujano y le permite visualizar los tejidos con mayor detalle [3].

Otra ventaja importante es la capacidad para superar las limitaciones físicas de los cirujanos. Algunos procedimientos pueden requerir movimientos complejos y precisos en áreas difíciles de alcanzar. Los brazos robóticos junto con los instrumentos *EndoWrist* pueden realizar estas tareas de manera más efectiva, lo que amplía el alcance de los procedimientos quirúrgicos y permite realizar intervenciones más complejas.

Sin embargo, es importante tener en cuenta que la robótica quirúrgica no reemplaza completamente a los cirujanos. Los sistemas robóticos son herramientas sofisticadas que requieren un entrenamiento y experiencia adecuados para su uso efectivo. Los cirujanos siguen siendo los responsables de tomar decisiones clínicas, planificar y supervisar los procedimientos quirúrgicos.

En resumen, la robótica quirúrgica ha transformado el campo de la cirugía al proporcionar a los cirujanos herramientas avanzadas para realizar procedimientos más precisos, menos invasivos y con una recuperación más rápida. A medida que la tecnología continúa avanzando, es probable que veamos un mayor desarrollo y aplicación de la robótica quirúrgica en diferentes áreas de la medicina, brindando así beneficios aún mayores para los pacientes.

1.2 ANTECEDENTES DE ESTE PROYECTO

Este proyecto se ha desarrollado dentro del grupo de robótica médica de la Universidad de Málaga, que dispone de un asistente robótico formado por un brazo robótico UR3 junto con una herramienta *EndoWrist*, que permite aumentar la capacidad de movimiento del sistema. Dado que las herramientas *EndoWrist* están especialmente diseñadas para utilizarse con el sistema *Da Vinci*, surge la necesidad, por un lado, de acoplar esta herramienta con el robot UR3, y por otro, de desarrollar un sistema de control capaz de controlar los grados de libertad de la herramienta.

El problema del acople de la herramienta con el brazo robótico se resolvió en el TFG titulado “*Diseño e implementación de un mecanismo de control de un instrumento quirúrgico EndoWrist*” [4], en el que se diseña un mecanismo de acople entre la herramienta *EndoWrist* y el brazo robótico UR3, que contiene cuatro motores dispuestos para controlar los cuatro grados de libertad adicionales que presenta dicha herramienta.

Por otro lado, queda sin resolver el problema del control de estos los grados de

libertad adicionales que presenta la herramienta. En este sentido, el objetivo de este proyecto es implementar dicho control de la herramienta junto con el control del brazo robótico a través de un dispositivo háptico.

1.3 OBJETIVO DEL PROYECTO

El objetivo principal del presente proyecto es implementar un esquema de control de movimiento del instrumento quirúrgico *EndoWrist* acoplado al brazo robótico UR3 mencionado anteriormente, mediante un dispositivo háptico. Para ello, pasaremos por distintas fases de trabajo:

1. **Modificaciones en el diseño del acople de la herramienta.** Para poder hacer un control completo de los grados de libertad de la herramienta, esta debe tener un acople robusto y una buena conexión con los motores. Por ello, se propone una mejora en el diseño de este acople para obtener mejores resultados.
2. **Estudio de movimiento del instrumento *EndoWrist*.** Para hacer un control de un instrumento complejo como el que tenemos, se debe hacer un estudio sobre sus grados de libertad, cómo se mueven a través de nuestros motores y qué combinaciones de motores mueven cada grado de libertad, ya que, como veremos, los grados de libertad no están desacoplados sino que un mismo motor puede mover distintos grados de libertad según se estimulen.
3. **Diseño del esquema de control de movimiento del sistema.** El esquema de control de movimiento se divide en dos problemas diferentes: el control del efector final del robot teniendo en cuenta el movimiento alrededor del punto de fulcro, y el control de movimiento de los grados de libertad de la herramienta a través de los motores del mecanismo de acople entre el brazo robótico y el instrumento.
4. **Implementación del algoritmo de control.** Una vez realizado el esquema de control de movimiento, se procederá a su implementación en software. Para ello, utilizaremos ROS como sistema global, el cual comunicará el dispositivo háptico, el robot, y los motores de la herramienta.
5. **Experimentación y resultados.** Para verificar el correcto funcionamiento del esquema de control implementado en la fase anterior, se procederá a realizar una serie de experimentos con el sistema completo.

1.4 ESTRUCTURA DE LA MEMORIA

A lo largo del presente documento se tratan los objetivos del proyecto planteados anteriormente, estando cada uno de los capítulos dedicados a cada uno de ellos. De esta forma, la memoria se divide en un total de siete capítulos:

- **Capítulo 1: Introducción.** Capítulo actual.
- **Capítulo 2: Descripción del Sistema.** Se detallan los distintos elementos del sistema y las modificaciones de algunos de ellos.
- **Capítulo 3: Control de Movimiento de la Herramienta.** Se realizan los experimentos pertinentes para conocer a fondo el comportamiento de la herramienta.
- **Capítulo 4: Esquema de Control de Movimiento.** Se plantea el esquema de control a seguir y las ecuaciones para controlar el sistema.
- **Capítulo 5: Implementación del Algoritmo de Control.** Se implementa el esquema de control en software a través de algoritmos.
- **Capítulo 6: Puesta en Marcha y Resultados.** Se realizan experimentos que prueben en buen funcionamiento del sistema.
- **Capítulo 7: Conclusiones Finales y Futuras Líneas de Trabajo.** Se concluye el proyecto y se proponen ideas para ampliar y mejorar el proyecto.

DESCRIPCIÓN DEL SISTEMA

En este capítulo se van a presentar en detalle los diferentes elementos que componen nuestro sistema. Este consta de cuatro elementos principales: el brazo robótico, la herramienta *EndoWrist*, el acople entre ambos, y el dispositivo háptico. Será este último el que controle el conjunto, y se colocará según la comodidad del cirujano.

Los cuatro elementos se encuentran disponibles en el laboratorio de robótica médica de la Universidad de Málaga.

2.1 BRAZO ROBÓTICO

El brazo robótico que se va a emplear es el UR3, de *Universal Robots* (figura 2.1). Es un robot colaborativo diseñado para trabajar junto con los humanos en entornos de producción. Es compacto y ligero, y ofrece una gran versatilidad en sus aplicaciones.

El UR3 tiene un alcance de 50 centímetros y una carga útil de 3 kilogramos, lo que lo hace ideal para tareas que requieren movimientos precisos y delicados, como es nuestro caso. Presenta seis grados de libertad, todos con una rotación 360° excepto en el extremo, que presenta una rotación infinita, lo que le dota de gran destreza y variedad de movimientos.



Figura 2.1: Brazo robótico UR3

2.2 INSTRUMENTO ENDOWRIST

La herramienta que se empleará será un instrumento quirúrgico *EndoWrist* (figura 2.2).

EndoWrist es una innovadora herramienta utilizada en cirugía mínimamente invasiva. Comprende una serie de instrumentos quirúrgicos miniaturizados y articulados que se utilizan a través de pequeñas incisiones. Están diseñados para proporcionar a

los cirujanos una destreza y precisión excepcionales durante la cirugía, superando las limitaciones de las herramientas tradicionales utilizadas en la cirugía abierta.

La característica clave de *EndoWrist* es su capacidad de replicar los movimientos de la mano humana con una mayor amplitud de movimiento y una mayor precisión. Los instrumentos están equipados con articulaciones flexibles que permiten una rotación de 360° y una inclinación de hasta 90°, lo que proporciona una amplia gama de movimientos y una mayor facilidad de manipulación dentro del cuerpo del paciente.



Figura 2.2: Instrumento *EndoWrist*

En concreto, la herramienta ofrece cuatro grados de libertad, controlados por cuatro discos incluidos en la herramienta (figura 2.3), aunque la herramienta posee un total de cinco discos, quedando uno de ellos inutilizado.



Figura 2.3: Discos de la herramienta

2.3 ACOPLE ENTRE EL ROBOT Y LA HERRAMIENTA

Para poder utilizar la herramienta *EndoWrist* en el sistema, esta debe ir unida al brazo robótico para formar un único conjunto. Para ello, se parte del acople diseñado en el proyecto presentado en el apartado 1.3, “*Diseño e implementación de un mecanismo de control de un instrumento quirúrgico EndoWrist*”.

Este acople permite unir la herramienta al robot y, además, con los motores que veremos a continuación encajados en cada disco, permite controlar cada grado de libertad gracias al microcontrolador Arduino UNO.

- **Motor DC Micro Metal.** Para poder girar los discos de la herramienta, se han utilizado cuatro motores DC Micro Metal con reductora 50:1 serie LP (figura 2.4), uno para cada disco.



Figura 2.4: Motor DC Micro Metal

- **Arduino UNO.** Para el control de los cuatro motores se ha utilizado un microcontrolador Arduino UNO (figura 2.5). Ha sido elegido por su facilidad de programación ya que su tarea es sencilla.

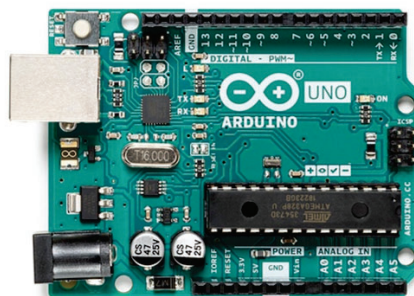


Figura 2.5: Arduino UNO

- **Piezas de unión.** Teniendo en cuenta las características del instrumento *EndoWrist*, se ha diseñado en 3D el mecanismo de acople mostrado en las figuras 2.6 y 2.7.

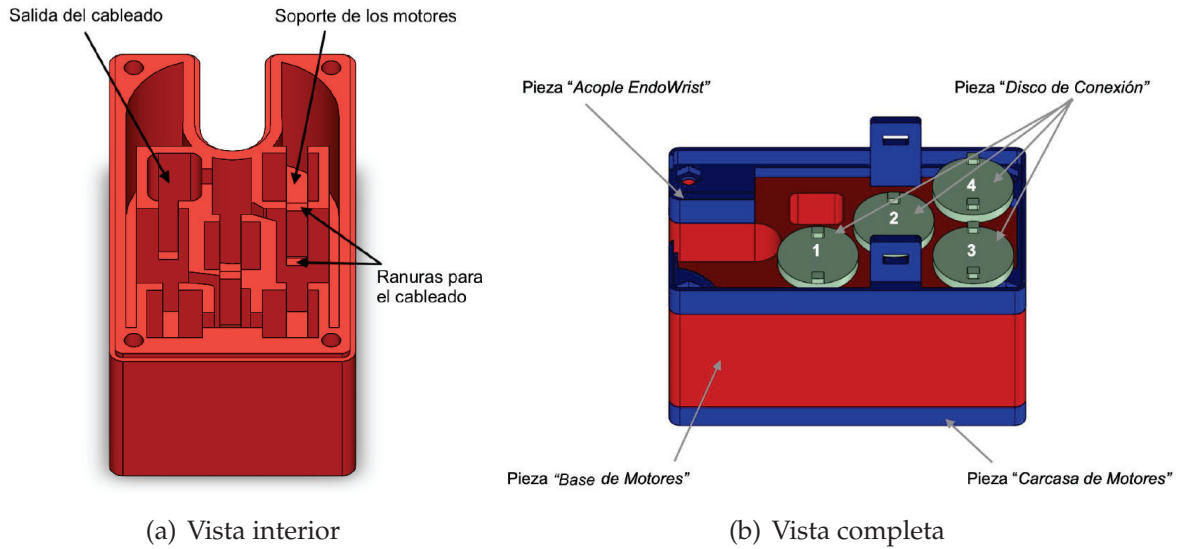


Figura 2.6: Mecanismo contenedor de los motores

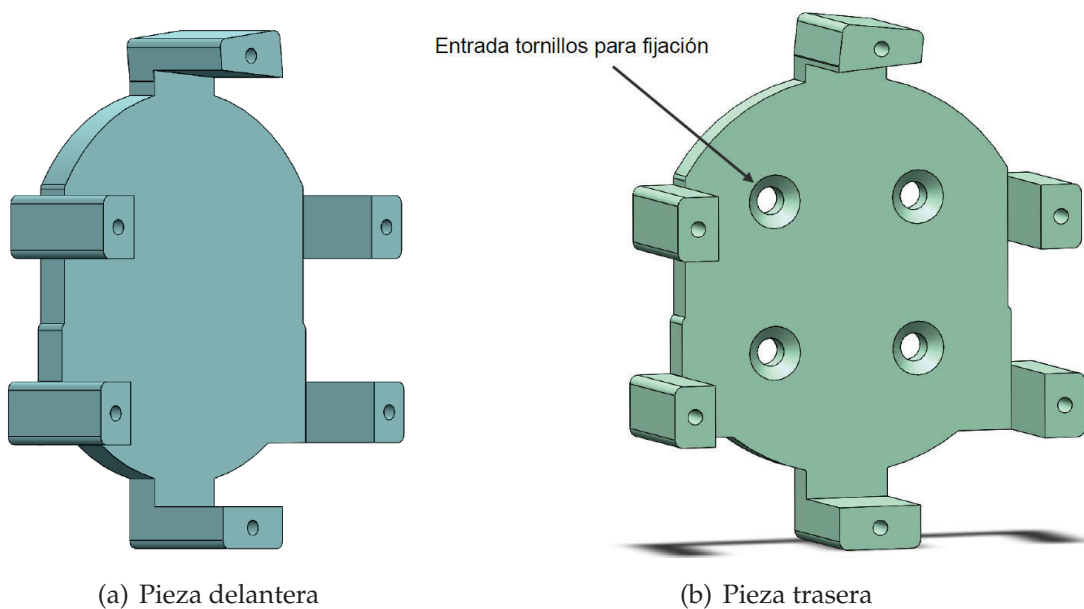


Figura 2.7: Mecanismo de acople entre la caja de motores y el robot

El acople final de la herramienta en el brazo robótico presentado en dicho proyecto resulta en la figura 2.8.

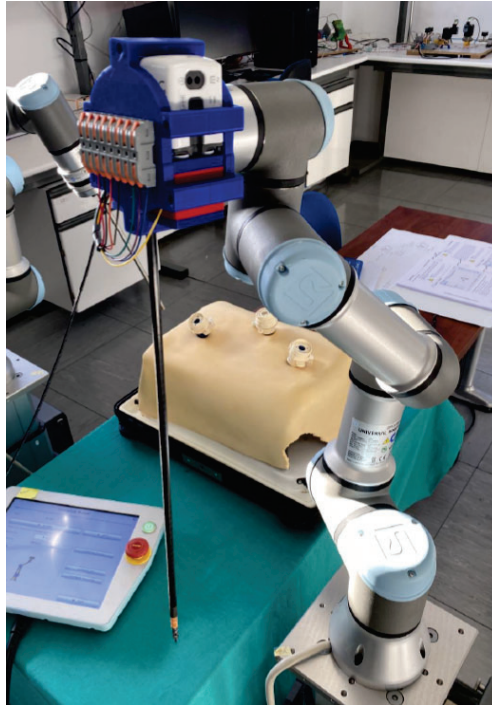


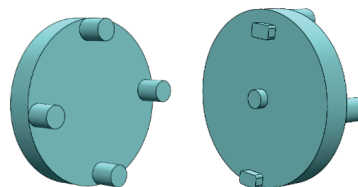
Figura 2.8: Acople final entre la herramienta y el brazo robótico

En el presente proyecto, se utilizará el mismo mecanismo presentado, pero con algunas mejoras en su diseño para mayor robustez de la unión.

En primer lugar, se han sustituido los discos de conexión mostrados en la figura 2.6 por un conjunto de dos piezas más sólidas: un acople metálico que va unido al motor, y un nuevo disco de conexión (figura 2.9). De esta forma, el nuevo sistema de acople sería la pieza metálica encajada en el motor, y el disco de conexión encajado en el acople metálico por un lado y en la herramienta por el otro (figura 2.10).



(a) Acople metálico



(b) Disco de conexión

Figura 2.9: Nuevas piezas de acople para la herramienta

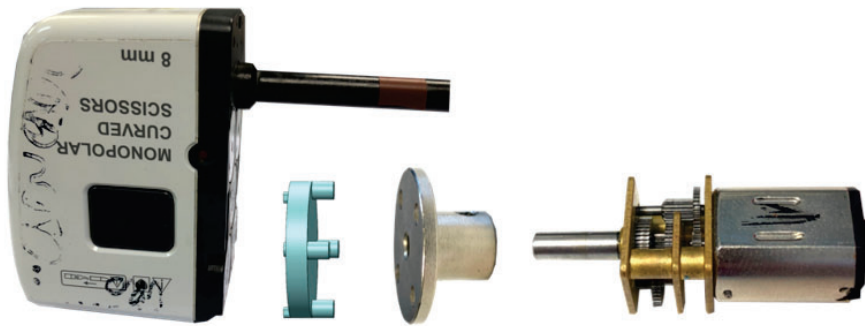
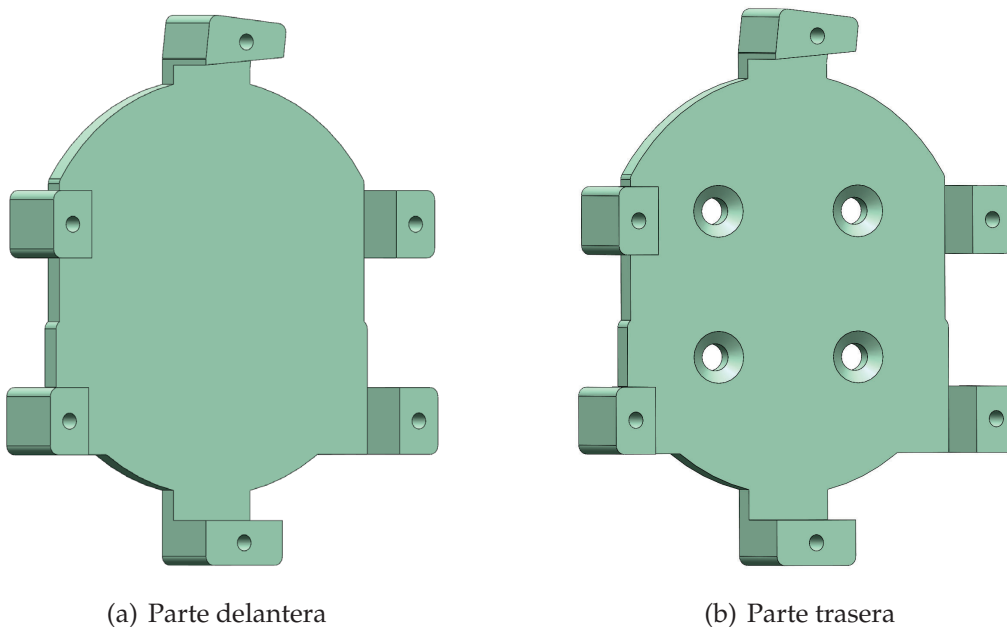


Figura 2.10: Ensamblado de las piezas para el montaje final

Así, estarán unidos el motor y la herramienta mediante estas dos piezas y el giro del motor se transmitirá de forma más precisa hacia el disco de la herramienta.

Por otro lado, al añadir una pieza nueva, el acople de la herramienta con el robot (figura 2.7) debe ser más grande para que encaje la herramienta con los motores. Por ello, se ha realizado un escalado a la pieza original para que mida 4 centímetros más (figura 2.11).



(a) Parte delantera

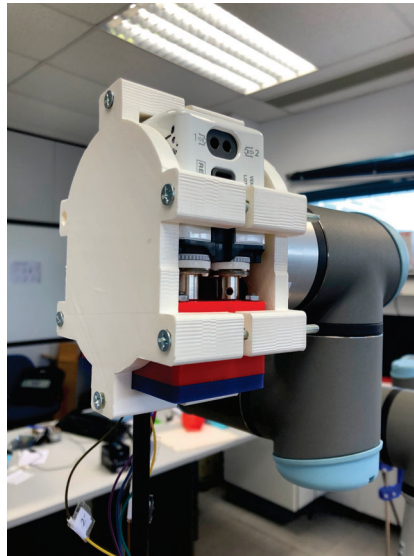
(b) Parte trasera

Figura 2.11: Nuevo mecanismo de acople entre la caja de motores y el robot

El acople final con las modificaciones realizadas se muestra en la figura 2.12.



(a) Unión de los motores con la herramienta



(b) Unión de la herramienta con el robot

Figura 2.12: Acople final del sistema

2.4 DISPOSITIVO HÁPTICO

El dispositivo háptico utilizado es el *Phantom Omni* [5] (figura 2.13). El *Phantom Omni* es una herramienta de retroalimentación táctil utilizada en aplicaciones de realidad virtual y simulación. Fue desarrollado por *Sensible Technologies* y ha encontrado amplio uso en campos como la medicina, la ingeniería y la investigación.

Este dispositivo se compone de un brazo robótico articulado y un lápiz háptico que permite a los usuarios interactuar con objetos virtuales y sentir sensaciones táctiles en tiempo real. El dispositivo está diseñado para replicar la resistencia y las fuerzas de contacto que se encontrarían al tocar objetos reales, brindando una experiencia háptica realista.

La principal característica del *Phantom Omni* es su capacidad de proporcionar retroalimentación háptica de seis grados de libertad, lo que significa que permite movimientos en todas las direcciones: traslación (movimiento lineal) y rotación (movimiento angular). Esto permite a los usuarios manipular objetos virtuales con mayor precisión y

realismo. En el presente proyecto, la traslación permitirá mover el brazo robótico, y la orientación permitirá orientar la herramienta *EndoWrist*.



Figura 2.13: Phantom Omni

2.4.1 Twee

El Twee [6] es un dispositivo que se acopla al *Phantom Omni* para proporcionarle un grado de libertad adicional, el cual se utilizará para comandar la apertura y cierre de la pinza (figura 2.14).

Este dispositivo ofrece un ángulo de apertura de 30° , con una resolución de 3° . También tiene un botón digital adicional situado en la parte superior.



Figura 2.14: Twee

2.5 CONCLUSIONES

Una vez presentado los elementos que componen nuestro sistema, se abordarán las fases del proyecto. En primer lugar, se va a realizar un estudio sobre la herramienta, sus grados de libertad y su relación con el movimiento de los discos. Tras esto, se diseñará el esquema de control de movimiento para el brazo robótico y los grados de libertad de la herramienta. Una vez se haya completado el esquema de control, se implementará en software, y finalmente se realizarán los ensayos y pruebas necesarias para completar el experimento.

CONTROL DE MOVIMIENTO DE LA HERRAMIENTA

Para poder controlar los grados de libertad que ofrece la herramienta, se debe antes estudiar su movimiento, ya que se trata de una herramienta compleja en cuanto a su control. Por ello, en este capítulo se exponen una serie de experimentos realizados a cada uno de los grados de libertad y se propone una forma de controlarlos.

3.1 ESTUDIO PREVIO DE LA HERRAMIENTA

La herramienta *EndoWrist* tiene cuatro grados de libertad: *roll*, *pitch*, *yaw* y *jaw* [7] (figura 3.1(a)), y son controlados por cuatro discos diferentes e interconectados entre sí (figura 3.1(b)).

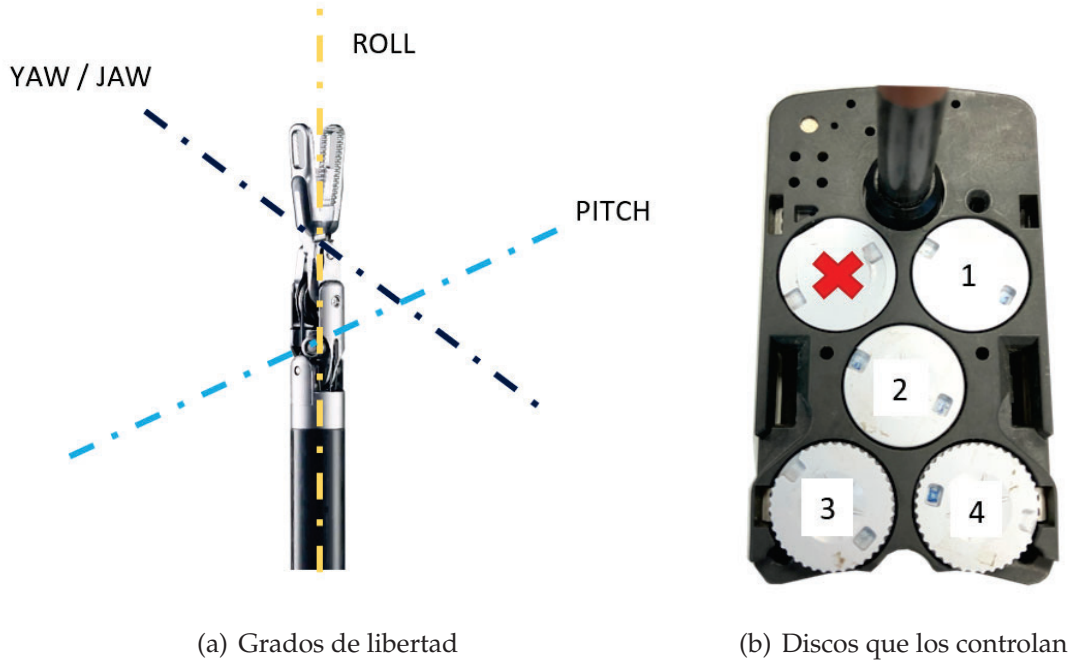


Figura 3.1: Movimiento de la herramienta

Cada grado de libertad es controlado por uno o más discos, de tal forma que un mismo disco puede mover más de un grado de libertad en función de cómo se mueva y junto con qué otros discos se combine. De esta forma, el grado de libertad *roll* lo controla el disco 1; el *pitch* lo controlan los discos 2, 3 y 4; y el *yaw* y *jaw* lo controlan los discos 3 y 4. La relación entre el giro de los discos y el movimiento de cada grado de libertad se muestra en la tabla 3.1.

GRADOS DE LIBERTAD DE LA HERRAMIENTA				
Grado de libertad	Discos que lo controlan	Giro de los discos	Relación de movimiento	Valor de PWM
Roll	Disco 1	Sentido Opuesto	1	150
Pitch	Disco 2	Mismo Sentido	1:1:2	200
	Disco 3	Mismo sentido		100
	Disco 4	Sentido opuesto		150
Yaw	Disco 3	Mismo Sentido	1:1	200
	Disco 4	Sentido opuesto		150
Jaw	Disco 3	Mismo Sentido	1:1	180
	Disco 4	Mismo sentido		150

Cuadro 3.1: Relación entre los discos y los grados de libertad de la herramienta

La relación de movimiento indica la proporción de movimiento del disco frente al grado de libertad. Por ejemplo, para mover el *pitch*, la relación 1:1:2 indica que, por cada giro del disco 2, se debe girar una vez el disco 3 y dos veces el disco 4.

Por otro lado, el valor de PWM viene dado tras realizar una pequeña prueba de movimiento de cada uno de los grados de libertad con el nuevo acople de los motores. Un valor menor haría que el motor no tuviera suficiente fuerza para mover el disco.

3.2 PRIMITIVAS DE MOVIMIENTO DE CADA GRADO DE LIBERTAD

Para saber cuánto se mueve cada grado de libertad según el tiempo en el que se aplica el PWM a cada motor, se muestran varios experimentos realizados para cada uno de estos grados de libertad.

Se han aplicado progresivamente pulsos de PWM con el valor indicado en la tabla 3.1 a cada motor cada 50 o 25 milisegundos (dependiendo de cada grado de libertad), y se ha registrado cuánto se mueve cada uno. Con estos datos, se pueden hacer unas gráficas que muestren el movimiento de cada grado de libertad con el tiempo.

3.2.1 Roll

Para el grado de libertad *roll* se ha considerado la referencia de 0° con la herramienta alineada en su eje Z, y las pinzas de forma lateral, tal y como muestra la figura 3.2. La herramienta tiene un alcance de 360° tanto para el lado positivo como para el negativo.



Figura 3.2: Movimiento del grado de libertad *roll*

Se han realizado cinco experimentos diferentes, en los cuales se ha recogido el ángulo de movimiento del grado de libertad tras aplicar el PWM cada 50 milisegundos. La gráfica resultante se muestra en la figura 3.3, y se representan los cinco experimentos realizados y su media, tanto para el *roll* positivo como para el negativo.

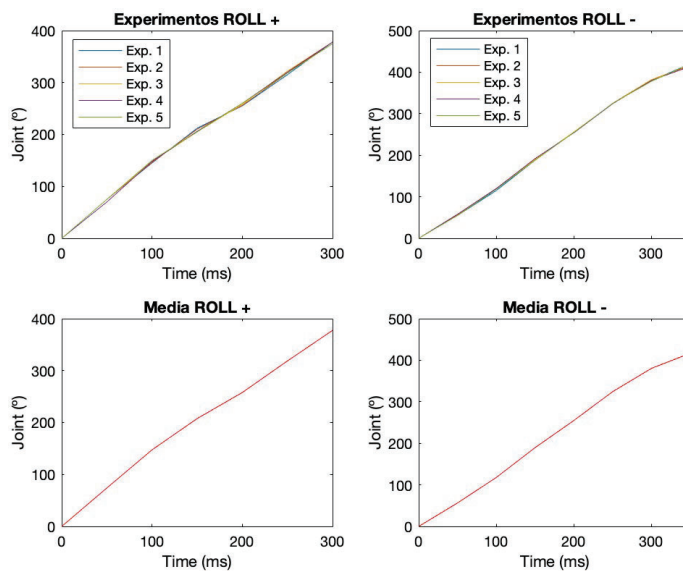


Figura 3.3: Gráfica de los experimentos para el *roll*

Si se toma únicamente la media y se realiza una aproximación polinomial, se obtiene una ecuación que se ajusta a todos los puntos (figura 3.4). Se utilizará esta ecuación para interpolar los diferentes puntos a los que se quiere llegar durante el movimiento.

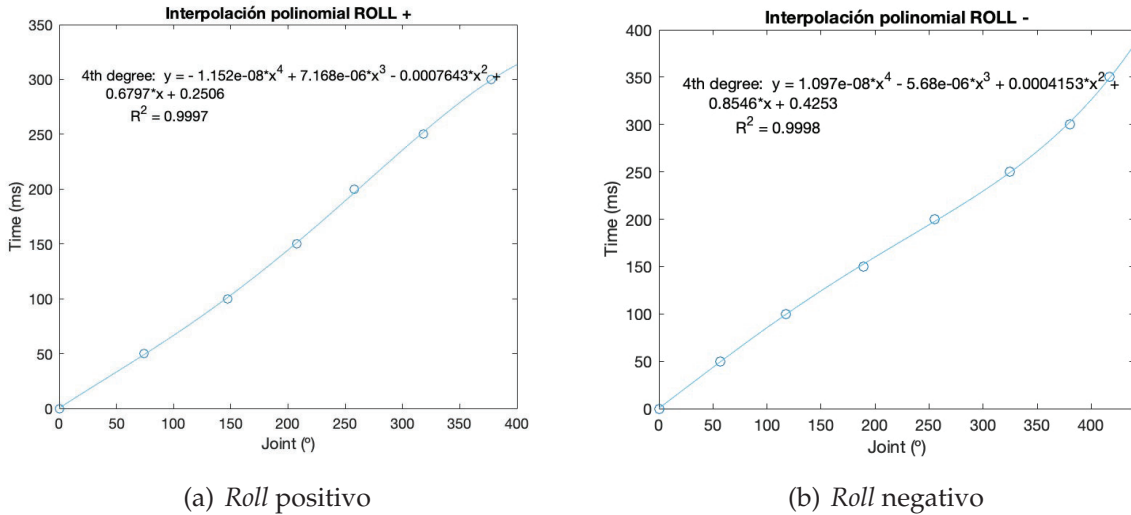


Figura 3.4: Interpolación polinomial para el roll

Con un polinomio de cuarto grado se han obtenido resultados exitosos tanto para el movimiento positivo como para el negativo, por lo que se toman estas aproximaciones.

La ecuación que modela el movimiento del *roll* positivo es:

$$y = -1,152 \cdot 10^{-8} \cdot x^4 + 7,168 \cdot 10^{-6} \cdot x^3 - 0,0007643 \cdot x^2 + 0,6797 \cdot x + 0,2506 \quad (3.1)$$

Y para el roll negativo:

$$y = 1,097 \cdot 10^{-8} \cdot x^4 - 5,68 \cdot 10^{-6} \cdot x^3 + 0,0004153 \cdot x^2 + 0,8546 \cdot x + 0,4253 \quad (3.2)$$

Siendo x los grados que se quiere mover e y el tiempo resultante.

Para el resto de grados de libertad se procede de la misma forma.

3.2.2 Pitch

El ángulo máximo del grado de libertad *pitch* es 80° hacia cada lado, suponiendo que los 0° se encuentran alineados con el eje Z de la herramienta (figura 3.5).



Figura 3.5: Movimiento del grado de libertad *pitch*

Se realizan de la misma forma cinco experimentos, aplicando el PWM correspondiente cada 25 milisegundos y anotando cada valor de ángulo.

La gráfica resultante de su movimiento positivo y negativo se muestra en la figura 3.6, donde de nuevo se representan los cinco experimentos y su media.

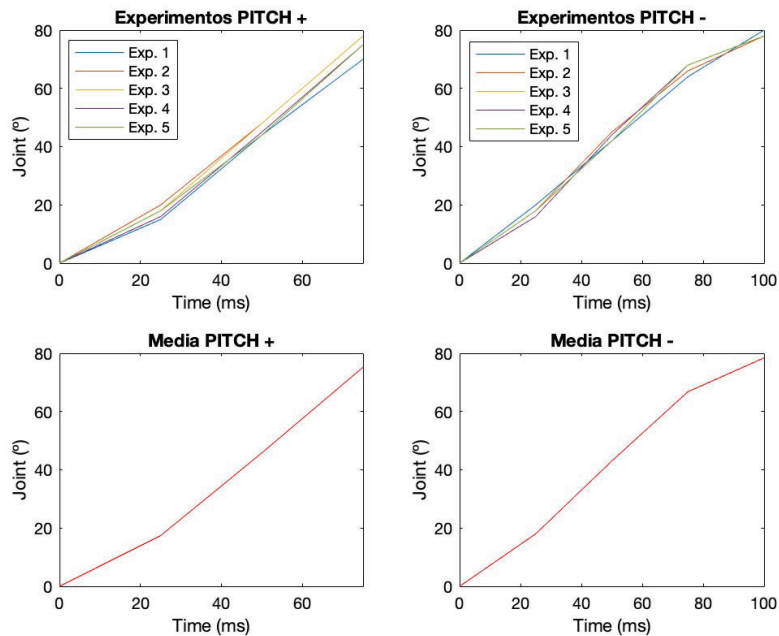


Figura 3.6: Gráfica de los experimentos para el *pitch*

Se realiza su aproximación polinomial de la misma forma que las anteriores, y se obtiene su ecuación (figura 3.7).

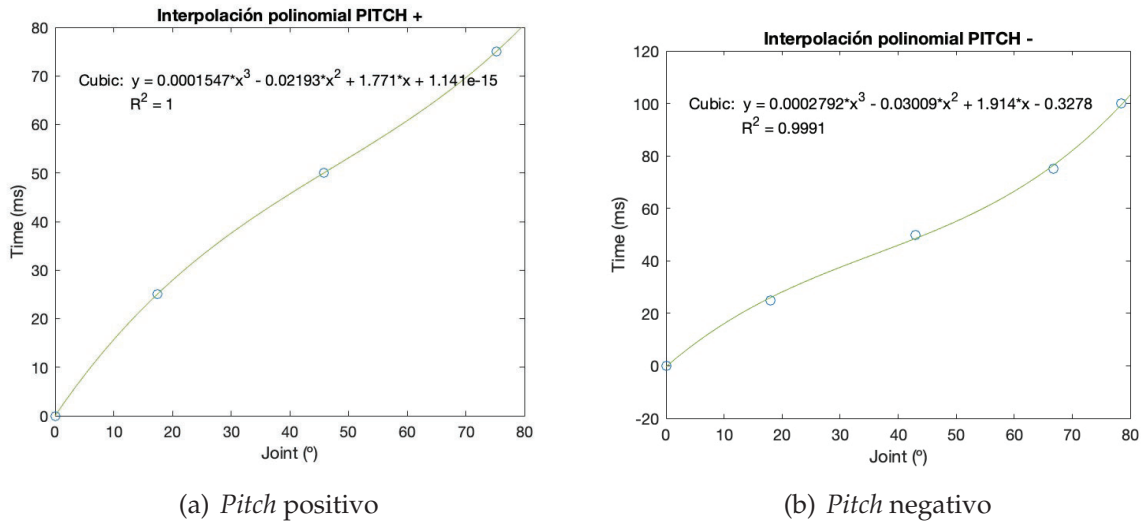


Figura 3.7: Interpolación polinomial para el *pitch*

En este caso, con un polinomio de grado tres ha sido suficiente para ajustarse casi a la perfección a ambos movimientos positivo y negativo.

La ecuación que modela el movimiento del *pitch* positivo es:

$$y = 0,0001547 \cdot x^3 - 0,02193 \cdot x^2 + 1,771 \cdot x + 1,141 \cdot 10^{-15} \quad (3.3)$$

Y para el *pitch* negativo:

$$y = 0,0002792 \cdot x^3 - 0,03009 \cdot x^2 + 1,914 \cdot x - 0,3278 \quad (3.4)$$

Siendo x los grados que se quiere mover este grado de libertad e y el tiempo resultante.

3.2.3 Yaw

Considerando de nuevo que la referencia de 0° se encuentra alineada con el eje Z de la herramienta, el grado de libertad *yaw* es capaz de moverse 120° hacia cada lado (figura 3.8).

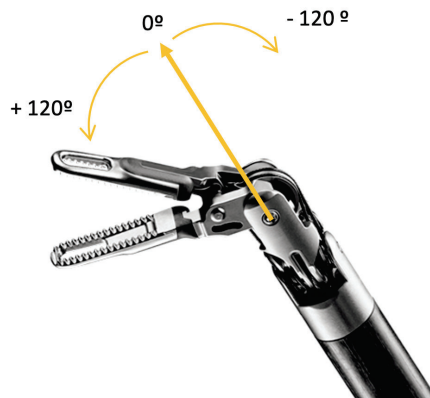


Figura 3.8: Movimiento del grado de libertad *yaw*

Se han realizado de nuevo los experimentos, y se muestran junto con su media en la figura 3.9.

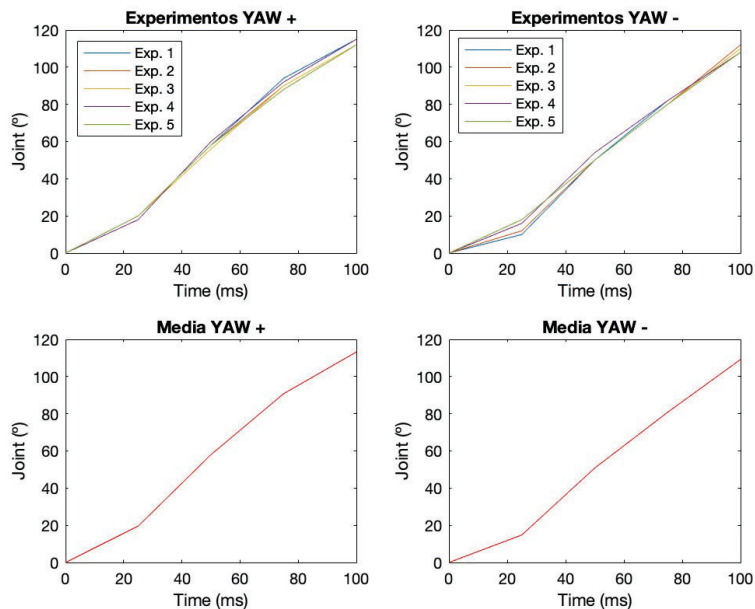


Figura 3.9: Gráfica para los experimentos del *yaw*

El ajuste polinomial con su ecuación se muestra en la figura 3.10.

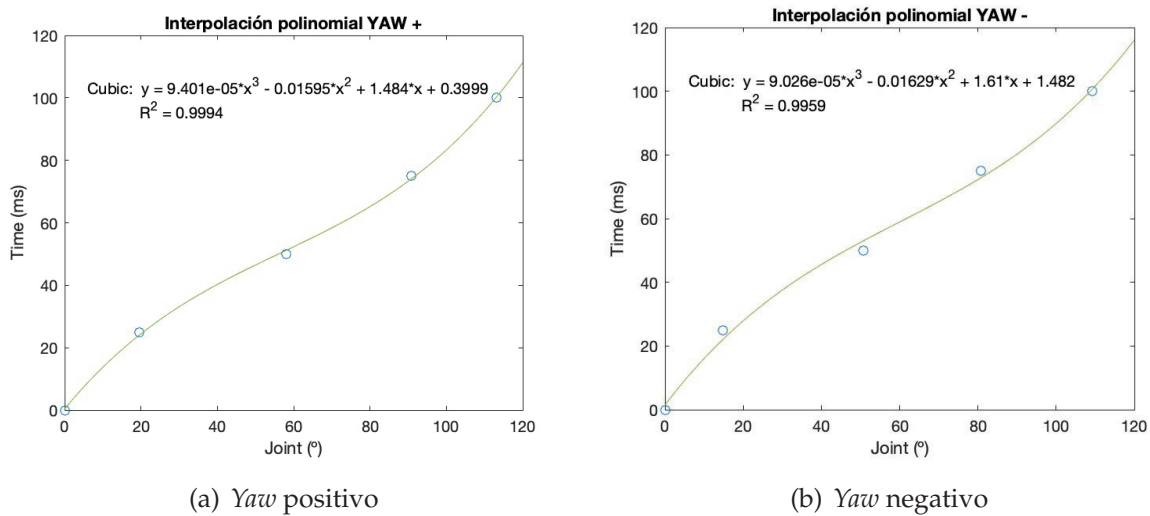


Figura 3.10: Interpolación polinomial para el *yaw*

En este caso, se ha conseguido conseguido un ajuste casi perfecto con un polinomio de tercer grado para el movimiento positivo y un polinomio de cuarto grado para el movimiento negativo.

La ecuación que modela el movimiento del *yaw* positivo es:

$$y = 9,401 \cdot 10^{-5} \cdot x^3 - 0,01595 \cdot x^2 + 1,484 \cdot x + 0,3999 \quad (3.5)$$

Y para el *yaw* negativo:

$$y = 9,026 \cdot 10^{-5} \cdot x^3 - 0,01629 \cdot x^2 + 1,61 \cdot x + 1,482 \quad (3.6)$$

Siendo x los grados que se quiere mover este grado de libertad e y el tiempo resultante.

3.2.4 Jaw

El grado de libertad *jaw* hace referencia a la apertura y cierre de la pinza. Se considera de nuevo que la referencia de 0° se encuentra alineada con el eje Z de la herramienta, de forma idéntica al *yaw* (figura 3.11). Este grado de libertad es capaz de moverse 230° en su apertura y cerrarse hasta los 0° de nuevo.



Figura 3.11: Movimiento del grado de libertad *jaw*

Para la apertura y cierre de la pinza se muestran los datos de los cinco experimentos y su media en la figura 3.12

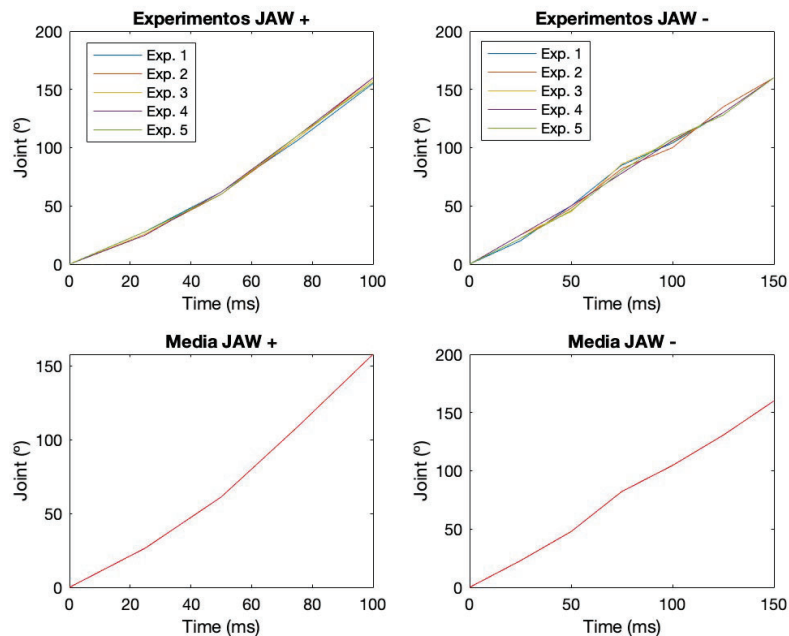


Figura 3.12: Gráfica de los experimentos del *jaw*

El ajuste polinomial con su ecuación se muestra en la figura 3.13.

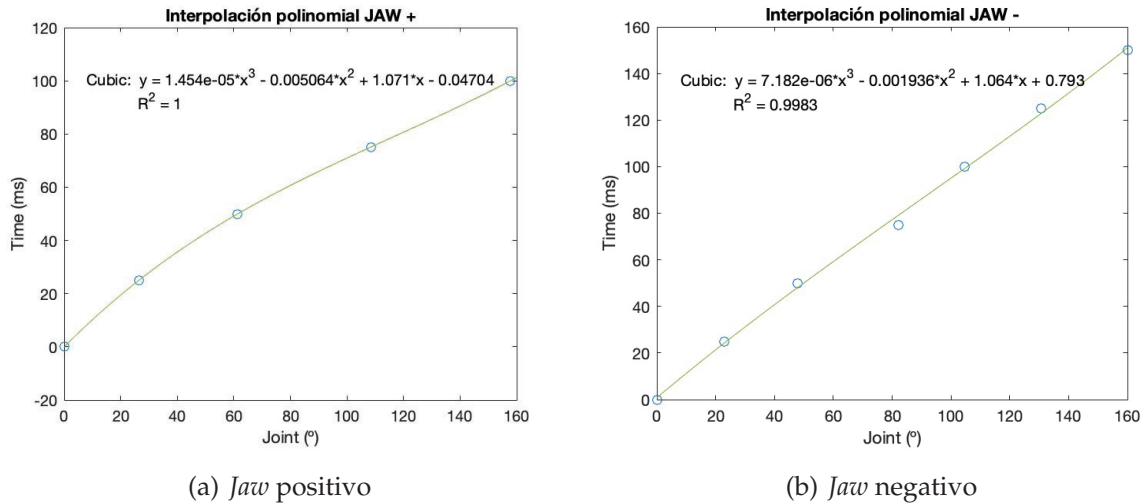


Figura 3.13: Interpolación polinomial para el *jaw*

Se puede ver que un polinomio de tercer grado se ajusta casi a la perfección tanto a la apertura como al cierre de la pinza.

La ecuación que modela la apertura de la pinza es:

$$y = 1,454 \cdot 10^{-5} \cdot x^3 - 0,005064 \cdot x^2 + 1,071 \cdot x - 0,04704 \quad (3.7)$$

Y para el cierre de la pinza:

$$y = 7,182 \cdot 10^{-6} \cdot x^3 - 0,001936 \cdot x^2 + 1,064 \cdot x + 0,793 \quad (3.8)$$

Siendo x los grados que se quiere mover este grado de libertad y y el tiempo resultante.

3.3 PRUEBAS Y CÁLCULO DE ERRORES

El objetivo de esta prueba es comprobar que cada grado de libertad de la herramienta responde correctamente a su comando de movimiento. Por ejemplo, si se comanda que el grado de libertad *roll* se mueva 90° , se deberá ver que la herramienta ha girado esos 90° .

Este control está implementado en la plataforma Arduino IDE, en la que se ha programado el cálculo del tiempo en el que se debe aplicar el PWM correspondiente gracias a las ecuaciones (3.1), (3.2), (3.3), (3.4), (3.5), (3.6), (3.7) y (3.8). Para hacer las pruebas convenientes, se ha diseñado una pequeña interfaz en la que se puede introducir el grado de libertad deseado y los grados que se quiere mover (figura 3.14).

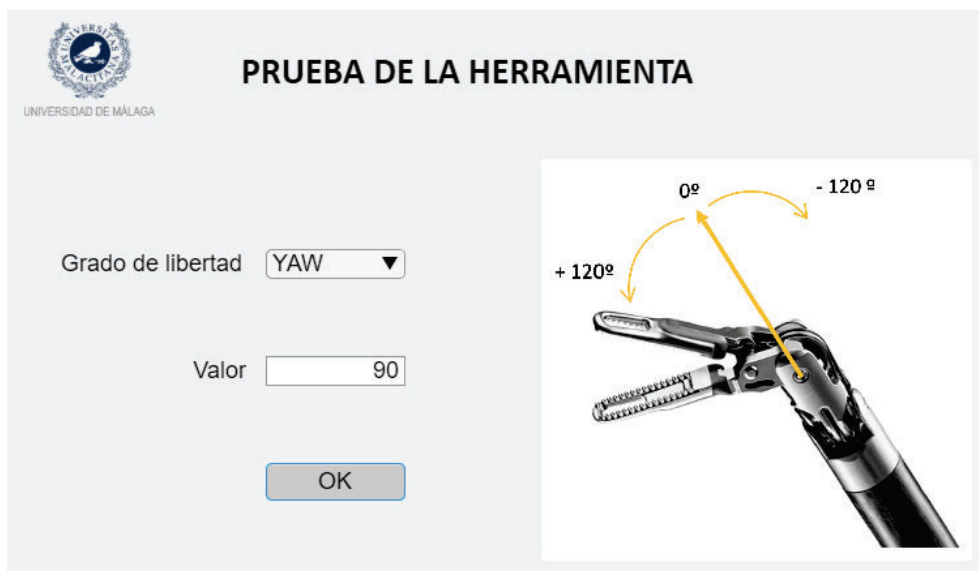


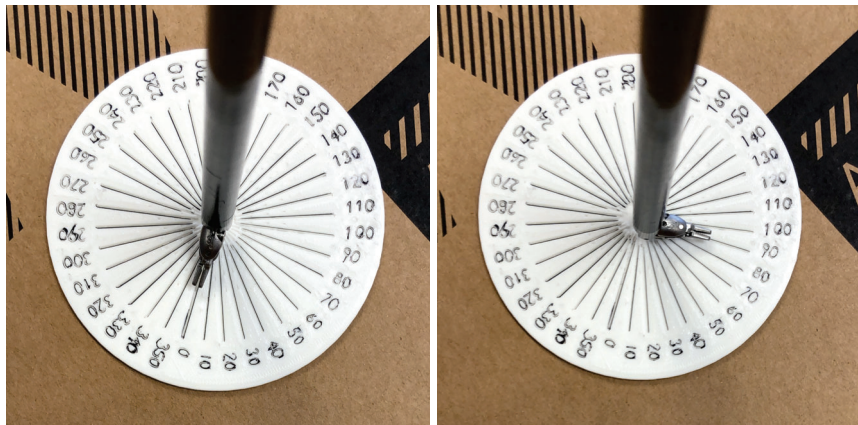
Figura 3.14: Interfaz para las pruebas de la herramienta

Al pulsar el botón 'OK', se enviará el grado de libertad seleccionado y su valor a la plataforma Arduino IDE, la cual se encargará de enviar la orden a cada motor.

3.3.1 Pruebas

A continuación se realizan diversas pruebas con el resultado real del movimiento de la herramienta, y posteriormente se realizará un cálculo de los errores del movimiento.

- **Prueba 1: mover 90° el *roll*.** Se introduce en la interfaz el valor de 90° escogiendo el grado de libertad *roll*. El resultado se muestra en la figura 3.15.

(a) Interfaz para el *roll*

(b) Movimiento 0°

(c) Movimiento 90°

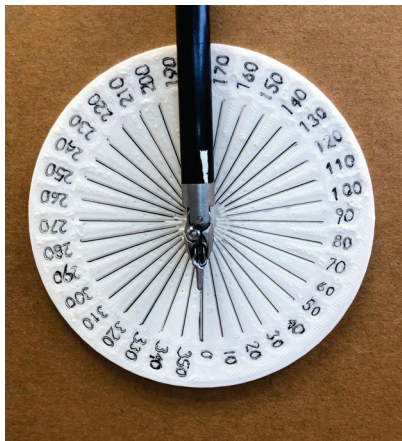
Figura 3.15: Movimiento de 90° del *roll*

Efectivamente, la herramienta se ha movido hasta los 94° aproximadamente, por lo que ha tenido un error puntual de 4°. El balance total del error del *roll* se muestran en el siguiente apartado.

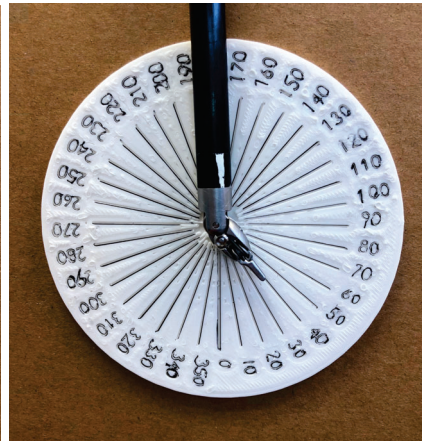
- Prueba 2: mover 40° el *pitch*.** De la misma forma que para el *roll*, se introduce un valor de 40° en la interfaz, seleccionando en este caso el grado de libertad *pitch*. El resultado del movimiento se muestra en la figura 3.16.



(a) Interfaz para el *pitch*



(b) Movimiento 0°

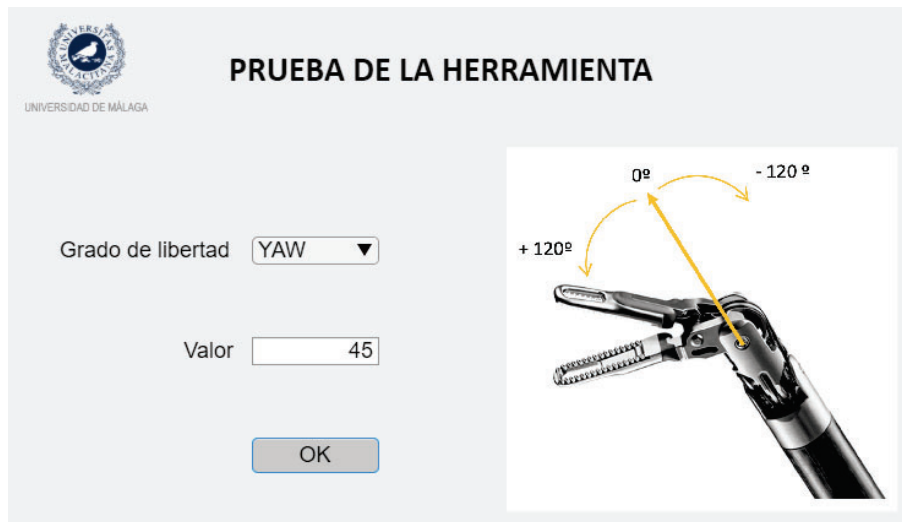
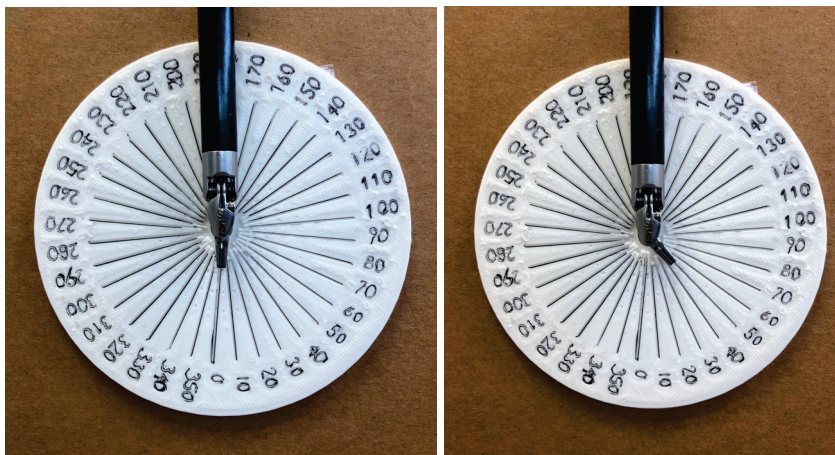


(c) Movimiento 40°

Figura 3.16: Movimiento de 40° del *pitch*

En este caso, se ha movido un total de 38° aproximadamente, por lo que ha presentado un error de 2° grados. De nuevo, el balance de los errores se presenta en el siguiente apartado.

- **Prueba 3: mover 45° el *yaw*.** Para la prueba del *yaw* se procede de la misma forma, introduciendo un valor de 45° en la interfaz y seleccionando el grado de libertad *yaw*. El resultado de su movimiento se muestra en la figura 3.17.

(a) Interfaz para el *yaw*

(b) Movimiento 0°

(c) Movimiento 45°

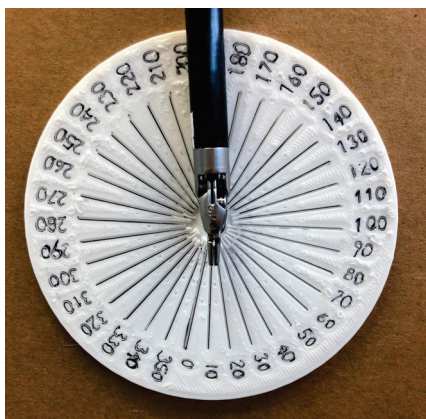
Figura 3.17: Movimiento de 45° del *yaw*

En este caso se ha movido un total de 46° aproximadamente, presentando un error puntual de 1°. El balance total del error para el *yaw* se muestra en el siguiente apartado.

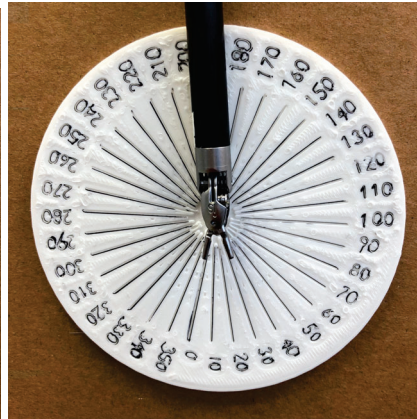
- **Prueba 4: mover 30° el *jaw*.** Finalmente, para el *jaw* se procede de la misma forma, introduciendo 30° como valor y seleccionando el *jaw* como grado de libertad. El resultado del movimiento se muestra en la figura 3.18.



(a) Interfaz para el *jaw*



(b) Movimiento 0°



(c) Movimiento 30°

Figura 3.18: Movimiento de 30° del *jaw*

El *jaw* se ha movido un total de 30°, por lo que ha realizado un movimiento perfecto con un error de 0°. De nuevo, el total del error medido para el *jaw* se estudia en el siguiente apartado.

3.3.2 Cálculo de errores

Para tener una idea de cómo de exacto es nuestro sistema, se debe realizar un estudio de los errores que comete cada uno de los grados de libertad en su movimiento. Por ello, se ha utilizado la anterior interfaz para comandar movimientos progresivos en cada uno de los grados de libertad y se ha comparado el valor introducido con el valor real del movimiento.

- Roll.** Se realiza un experimento comandando valores de ángulo al *roll* entre 0° y 360° de forma progresiva, y se recogen los valores que devuelve la herramienta real ante dichos impulsos (figura 3.19). De esta forma, se calcula el error cometido y se estima el valor máximo de este.

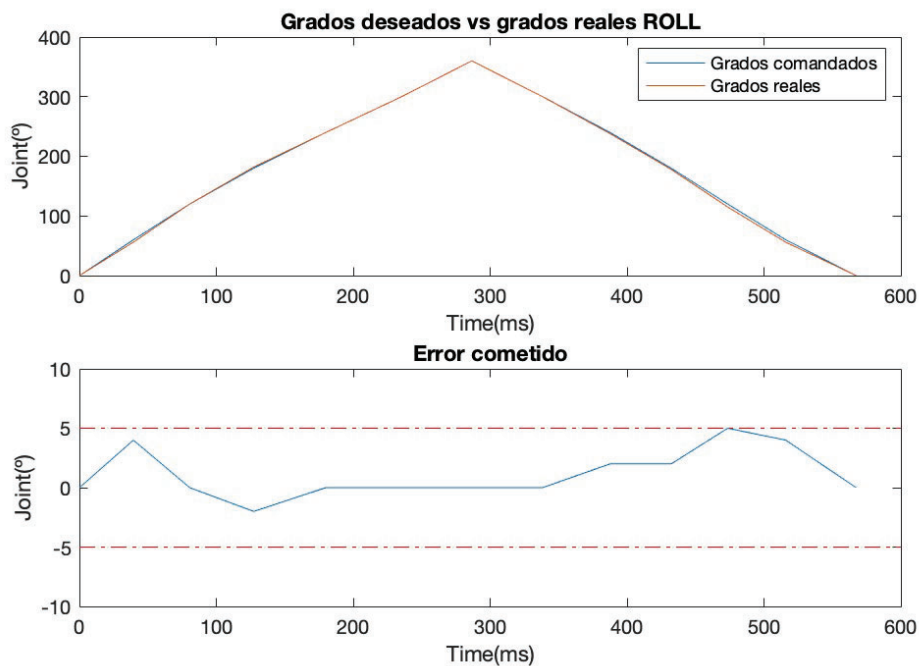


Figura 3.19: Error para el *roll*.

Se observa cómo la curva de valores se sigue de forma bastante aceptable, con un error máximo de 5° .

- Pitch.** Se realiza el mismo experimento, pero para valores de grados entre 0° y 80° , y se comparan los valores comandados con los valores reales que se mueve el *pitch* (figura 3.20). De esta forma, se recoge una gráfica con los errores y se estima el valor del error máximo.

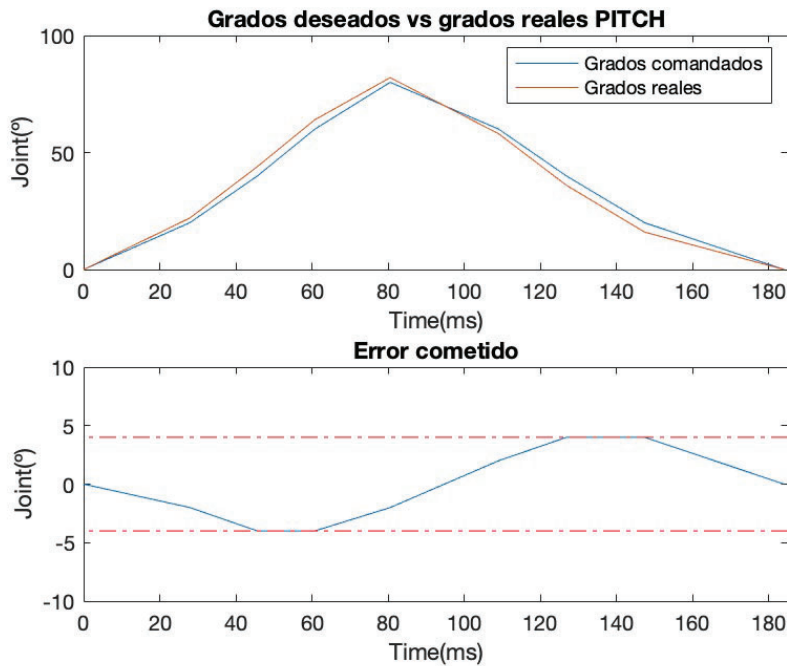


Figura 3.20: Error para el *pitch*.

Se observa cómo la curva de valores reales se acerca bastante a la curva de valores comandados, lo que ofrece un error máximo de 4° .

- Yaw.** En este caso, los ángulos comandados van entre 0° y 120° de forma progresiva, y se comparan estos valores con los valores reales que la herramienta se mueve en el *yaw* (figura 3.21). De esta forma, se calculan los errores y se obtiene el error máximo que tiene este movimiento.

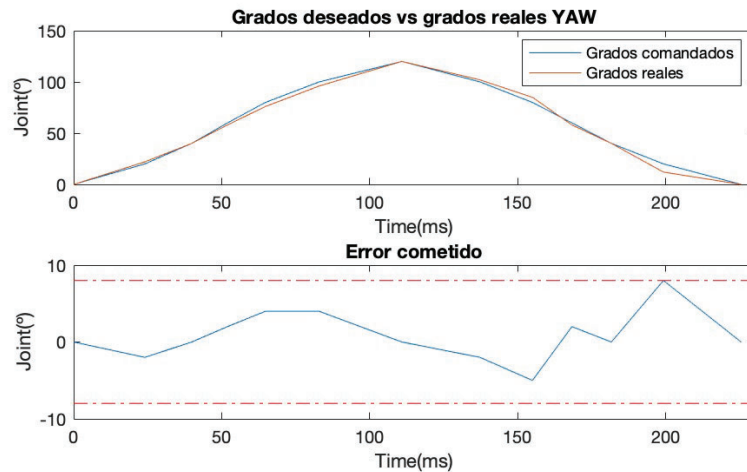


Figura 3.21: Error para el *yaw*.

Se observa que la curva de grados reales acompaña a la de los grados comandados, y el error máximo es de 8° .

- **Jaw.** Finalmente, para el *jaw* se procede comandando esta vez entre 0° y 160° . Se compara la curva de grados reales con los comandados y se realiza una gráfica con los errores del movimiento y el error máximo (figura 3.22).

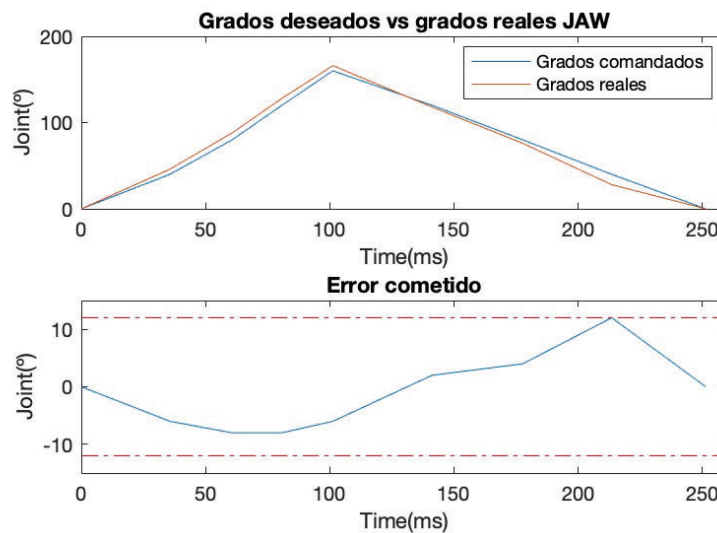


Figura 3.22: Error para el *jaw*.

Se observa una curva de valores reales muy cercana a los valores comandados, con un error máximo de 12° .

3.4 CONCLUSIONES

Se ha visto cómo con unos experimentos sencillos se puede modelar el movimiento de cualquier grado de libertad de nuestra herramienta. Además, se ha modelado en forma de polinomios, lo que facilitará la programación del algoritmo de control.

Puesto que son experimentos realizados en tiempo real en el laboratorio, se tienen errores de movimiento debido al desgaste de los motores, al desgaste de la propia herramienta y a pequeños desfases en la medida. Con esto, se consigue un error de movimiento de entre 4° y 12° para los cuatro grados de libertad.

Una vez realizado el estudio del movimiento de los grados de libertad de la herramienta, se debe diseñar un esquema de control capaz de calcular los ángulos correspondientes para cada grado de libertad, y la posición y orientación del efector final del robot, ambos según el movimiento del dispositivo háptico.

ESQUEMA DE CONTROL DE MOVIMIENTO

En este apartado se pretende plantear el sistema completo, establecer los sistemas de referencia de cada elemento y relacionarlos entre sí mediante un esquema de control.

Con este esquema de control se podrá de calcular dónde se tiene que mover el robot para que la herramienta se mueva en sintonía con el movimiento del dispositivo háptico.

4.1 MODELO DE LA TAREA

El sistema con el que se trabaja consta de dos elementos bien diferenciados: el dispositivo háptico y el conjunto del brazo robótico con la herramienta. Ambos se muestran en la figura 4.1, con sus sistemas de referencia bien definidos.

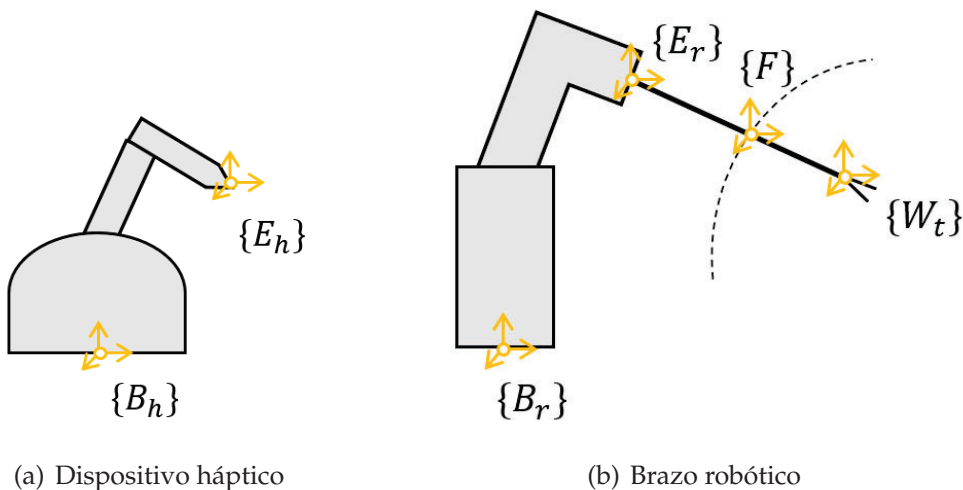


Figura 4.1: Dibujo del sistema simplificado

Cada sistema de referencia corresponde a un elemento del conjunto. En la tabla 4.1 se muestran dichos sistemas de referencia con su elemento correspondiente.

SISTEMAS DE REFERENCIA		
Sistema de referencia	Elemento al que pertenece	Descripción
$\{B_h\}$ $\{E_h\}$	Dispositivo háptico	Base del dispositivo Punta del lápiz
$\{B_r\}$ $\{E_r\}$	Robot	Base del robot Efector final del robot
$\{F\}$ $\{W_t\}$	Herramienta <i>EndoWrist</i>	Punto de fulcro Muñeca de la herramienta

Cuadro 4.1: Sistemas de referencia del conjunto

En robótica quirúrgica, un punto de fulcro se refiere al punto en el que el instrumento quirúrgico se encuentra anclado o pivotado durante una cirugía. Este es crítico para el funcionamiento del robot quirúrgico y es el punto de referencia desde el cual el robot realiza movimientos y maniobras precisas. Es por ello que, al controlar el robot con el dispositivo háptico, se debe controlar su posición para que la muñeca alcance el lugar deseado teniendo en cuenta este punto de fulcro.

Este punto de fulcro limita los grados de libertad del robot a cuatro: desplazamiento en Z y rotaciones en X, Y y Z. Es por ello que se utiliza la herramienta *EndoWrist*, la cual proporciona cuatro grados de libertad extra (*roll*, *pitch*, *yaw* y *jaw*, a partir de ahora q_r, q_p, q_y y q_j).

Los sistemas de referencia en los dispositivos del laboratorio se pueden ver en la figura 4.2.

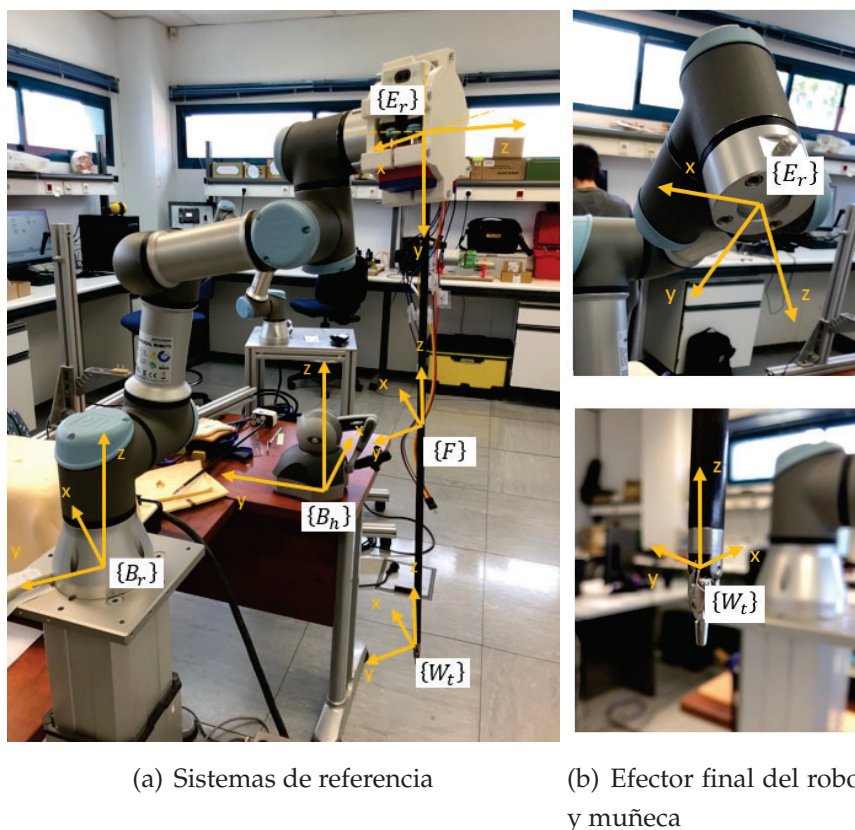


Figura 4.2: Conjunto de sistemas de referencia en los dispositivos del laboratorio

Dichos sistemas de referencia se pueden relacionar entre sí mediante relaciones de transformación (figura 4.3).

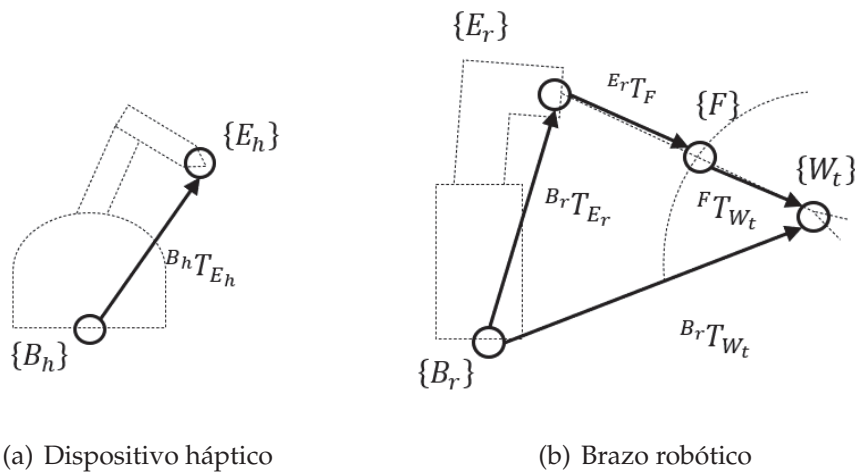


Figura 4.3: Relaciones de transformación del sistema

Cada una de las relaciones se puede describir y ver cómo se obtendría su valor:

- $B_h T_{E_h}$ la proporciona directamente el dispositivo háptico. Representa la posición y orientación de la punta del lápiz con respecto a la base.
- $B_r T_{E_r}$ representa la posición y orientación del efector final del robot con respecto a la base del mismo. Es el objetivo que se debe calcular para saber dónde debe moverse el robot para situarse en la posición deseada, teniendo en cuenta la rotación alrededor del punto de fulcro.
- $E_r T_F$ representa la posición y orientación del punto de fulcro con respecto al efector final del robot.

En cuanto a la posición, es un desplazamiento en el eje Z positivo del efector final del robot de z centímetros, un desplazamiento en el eje X negativo de x centímetros y un desplazamiento en el eje Y positivo de $y - \rho_0$ centímetros, donde ρ_0 representa la distancia del punto de fulcro a la muñeca de la herramienta. Este desplazamiento se debe al acople de la herramienta y sus dimensiones (figura 4.4).

En cuanto a la orientación, se trata de una rotación en el eje X de -90° sobre la orientación que tenga el punto de fulcro.

De esta forma:

$${}^E_r T_F = R_x(90) \cdot D \left(\begin{bmatrix} -x \\ y - \rho_0 \\ z \end{bmatrix} \right) \quad (4.1)$$

- ${}^F T_{W_t}$ representa la posición y orientación de la muñeca de la herramienta con respecto al punto de fulcro. La orientación coincide con la del punto de fulcro, y la posición es un desplazamiento en el eje Z positivo de la herramienta de ρ_0 centímetros.
- ${}^{B_r} T_{W_t}$ representa la posición y orientación de la muñeca de la herramienta con respecto a la base del robot. Esta posición y orientación vienen dadas por la posición del lápiz del dispositivo háptico, la cual necesitará una serie de transformaciones y escalado para adecuar su movimiento al de la herramienta.

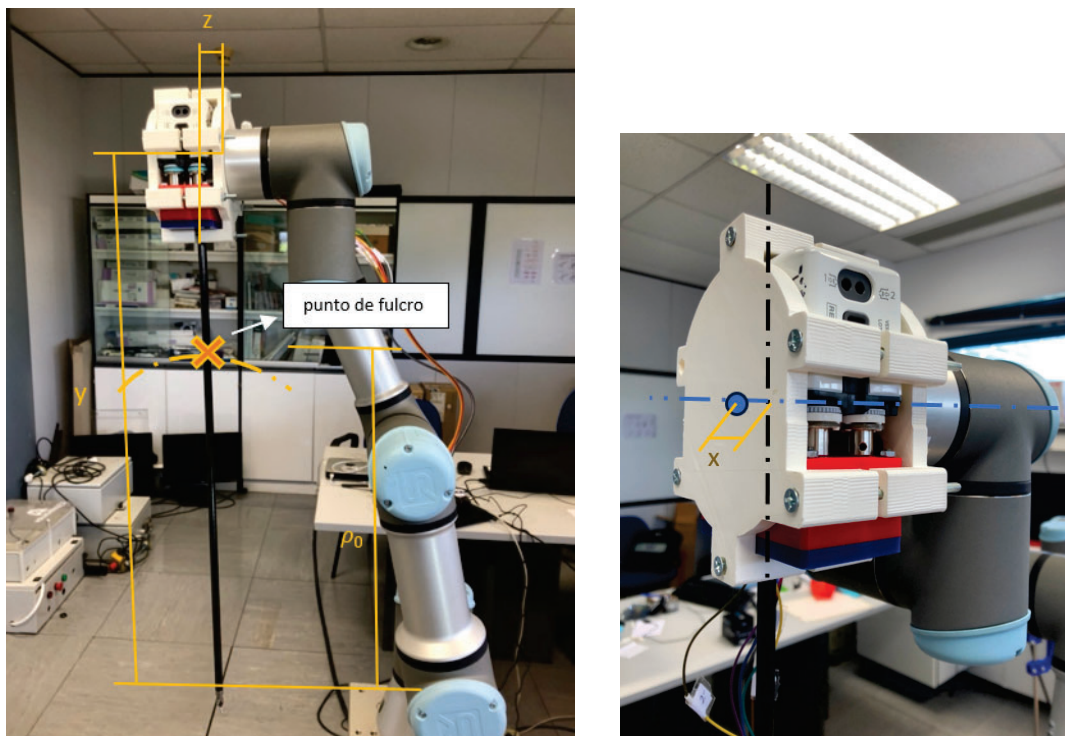


Figura 4.4: Medidas de la herramienta con respecto al efector final del robot

4.2 ESQUEMA DE CONTROL

Una vez que se han establecido todas las relaciones, se presenta el esquema de control utilizado (figura 4.5). El objetivo es que, con el movimiento del lápiz del dispositivo háptico, se pueda mover por un lado el robot (según su posición) y por otro lado la punta de la herramienta (según su orientación).

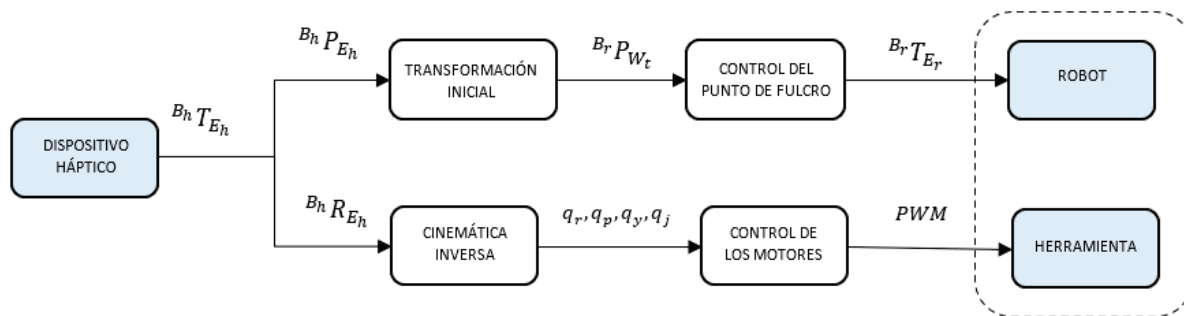


Figura 4.5: Esquema de control

Se han definido cuatro bloques de trabajo, los cuales calculan la variable deseada con conversiones y matrices de transformación. Estos bloques son los siguientes:

- **Transformación Inicial.** Trata de calcular la posición de la muñeca de la herramienta en función de la posición del lápiz del dispositivo háptico.
- **Control del Punto de Fulcro.** Calcula la posición y orientación del efector final del robot según la posición de la muñeca de la herramienta (calculada en el anterior bloque) y la posición del punto de fulcro, definida con el parámetro ρ_0 .
- **Cinemática Inversa.** Calcula el valor del ángulo de cada grado de libertad en función de la orientación del lápiz del dispositivo háptico.
- **Control de los Motores.** Calcula el tiempo que se deben estimular los motores en función del ángulo que se desea mover con cada una de las ecuaciones (3.1), (3.2), ..., (3.8) presentadas en el capítulo anterior.

Las entidades mencionadas se desarrollarán en profundidad a continuación.

4.2.1 Transformación Inicial

Este bloque consiste en calcular la posición de la muñeca de la herramienta a partir de la posición que obtenemos del dispositivo háptico. Para ello, se debe conocer la relación entre ambos sistemas de referencia, que variará según dónde esté colocado el dispositivo háptico respecto a la herramienta.

El dato de entrada en este caso es la posición del lápiz, y para realizar los movimientos, se trabaja con incrementos de posición, ΔP_h . Se pretende que este incremento se traslade directamente a la muñeca de la herramienta, de tal forma que si el lápiz se mueve a la derecha, la muñeca se mueva también a la derecha.

Se debe realizar la transformación entre el sistema de referencia del dispositivo háptico y el de la muñeca. Ambos sistemas de referencia están relacionados entre sí mediante una rotación de -45° con respecto al eje Z (figura 4.6), por lo que:

$${}^{W_t}R_{B_h} = R_z(-45^\circ) \quad (4.2)$$

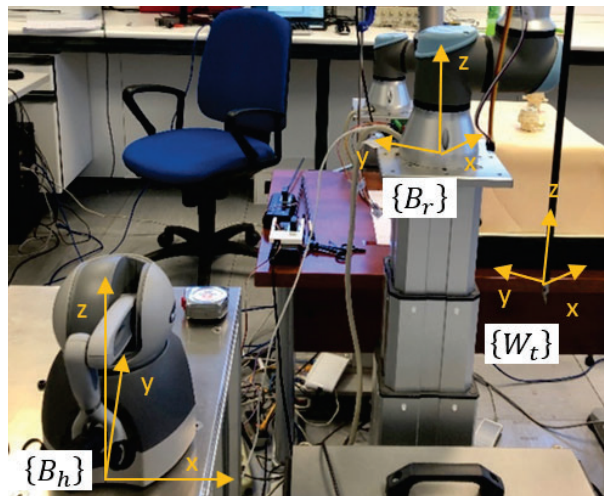


Figura 4.6: Relación entre la base del dispositivo háptico y la muñeca de la herramienta

De esta forma, para obtener el incremento de posición en el sistema de referencia de la muñeca, se realiza la siguiente operación:

$$\Delta P_{W_t} = {}^{W_t}R_{B_h} \cdot \Delta P_h \quad (4.3)$$

Se debe tener este incremento de posición referido a la base del robot, ya que este se comanda de esa forma. Para ello:

$${}^{Br}T_{W_{nueva}} = {}^{Br}T_{W_{ant}} \cdot D(\Delta P_{W_t}) \quad (4.4)$$

De esta manera se obtiene la nueva posición de la muñeca de la herramienta, la posición deseada, referida a la base del robot para un incremento de posición medido con el dispositivo háptico.

4.2.2 Control del Punto de Fulcro

El objetivo de este bloque es obtener la posición y orientación del efector final del robot en función de la posición de la muñeca (calculada en el anterior bloque) y del punto de fulcro.

Para calcular la posición del efector final en función de la nueva posición de la muñeca, hay que tener en cuenta el punto de fulcro. Este será fijo, pero en una primera instancia se debe calcular su posición:

$$P_f = P + (D_T - \rho_0) \cdot \bar{z} \quad (4.5)$$

Siendo P_f la posición del punto de fulcro, P la posición del efector final en ese instante, D_T la longitud total de la herramienta, ρ_0 la distancia desde la muñeca al punto de fulcro y \bar{z} el eje Z del efector final.

Una vez se tiene localizado el punto de fulcro, se debe calcular el nuevo eje de la herramienta tras el movimiento de la muñeca, que será el que una la nueva posición de la misma con el punto de fulcro, fijo:

$$\bar{z} = P_f - {}^{Br}P_{W_{nueva}} \quad (4.6)$$

Una vez se tiene el nuevo eje, la posición del efector final del robot será un desplazamiento en el mismo:

$${}^{B_r}P_{E_r} = (P_f + (D_T - \rho_0) \cdot \bar{z}) \cdot D \begin{pmatrix} 0 \\ 0 \\ z \end{pmatrix} \quad (4.7)$$

Siendo $D \begin{pmatrix} 0 \\ 0 \\ z \end{pmatrix}$ el desplazamiento correspondiente en el eje Z del efector final debido a las dimensiones del acople de la herramienta (figura 4.4).

Una vez calculada la posición del efector final del robot, el siguiente objetivo es conocer la orientación del mismo, ${}^{B_r}R_{E_r}$. Para ello, se trabaja con el valor del eje Z calculado anteriormente, \bar{z} .

Primero se normaliza el eje:

$$m = \sqrt{Z_x^2 + Z_y^2 + Z_z^2} \quad (4.8)$$

$${}^{B_r}Z_{E_{rnorm}} = \left(\frac{Z_x}{m}, \frac{Z_y}{m}, \frac{Z_z}{m} \right) \quad (4.9)$$

Para mayor simplicidad, el eje ${}^{B_r}Z_{E_{rnorm}}$ se denominará a partir de ahora Z_e .

Para formar la matriz de rotación que defina la orientación ${}^{B_r}R_{E_r}$, se utilizan los ángulos de Euler ZYZ, ya que se obtienen los ángulos β y γ con la información del eje Z, y el ángulo α no se considera (representa la rotación en el eje X y no es de interés).

La matriz de rotación según los ángulos de Euler ZYZ es:

$$R_{YZZ}(\alpha, \beta, \gamma) = \begin{pmatrix} c_\alpha c_\beta c_\gamma - s_\alpha s_\gamma & -c_\alpha c_\beta s_\gamma - s_\alpha c_\gamma & c_\alpha s_\beta \\ s_\alpha c_\beta c_\gamma + c_\alpha s_\gamma & -s_\alpha c_\beta s_\gamma + c_\alpha c_\gamma & s_\alpha s_\beta \\ -s_\beta c_\gamma & s_\beta s_\gamma & c_\beta \end{pmatrix} \quad (4.10)$$

De esta forma, los ángulos β y γ se calculan:

$$\beta = \operatorname{atan} \left(\frac{\pm \sqrt{Z_{ex}^2 + Z_{ey}^2}}{Z_{ez}} \right) \quad (4.11)$$

$$\gamma = \operatorname{atan} \left(\frac{Z_{ex} / \sin(\beta)}{-Z_{ey} / (\sin(\beta))} \right) \quad (4.12)$$

Imponemos que $\alpha = 0$.

Por tanto, se construye la matriz de rotación:

$${}^{B_r}R_{E_r} = \begin{pmatrix} c_\beta c_\gamma & -c_\beta s_\gamma & s_\beta \\ s_\gamma & c_\gamma & 0 \\ -s_\beta c_\gamma & s_\beta s_\gamma & c_\beta \end{pmatrix} \quad (4.13)$$

Con esta, ya se ha calculado tanto la posición como la orientación del efector final del robot, por lo que:

$${}^{B_r}T_{E_r} = \begin{pmatrix} {}^{B_r}R_{E_r} & {}^{B_r}P_{E_r} \\ [0,0,0] & 1 \end{pmatrix} \quad (4.14)$$

4.2.3 Cinemática Inversa

El objetivo de este bloque es obtener los ángulos de los cuatro grados de libertad de la herramienta (*roll*, *pitch*, *yaw* y *jaw*) a partir de la orientación del lápiz del dispositivo háptico [8]. Para ello, se relacionan los movimientos del lápiz con los grados de libertad.

Se puede decir que la rotación en el eje 4 se corresponde con el *yaw*, en el eje 5 se corresponde con el *pitch* y en el eje 6 con el *roll* (figura 4.7). De esta forma, estos movimientos son directos.

Se utilizará el Twee como grado de libertad adicional en el dispositivo háptico, disponiendo de la apertura y cierre del mismo para controlar el grado de libertad *jaw*.

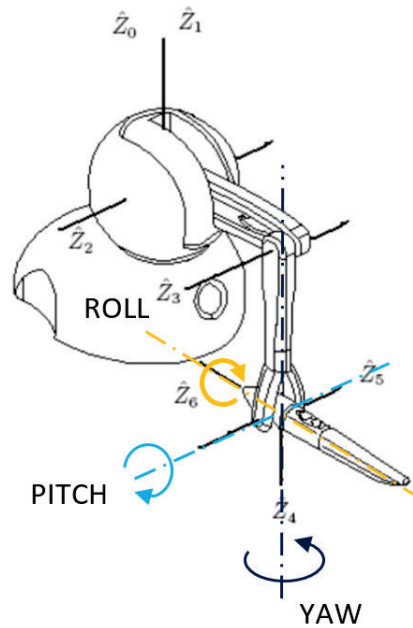


Figura 4.7: Grados de libertad relacionados con el Phantom

Aunque el movimiento es directo (si rotamos el lápiz en el eje 6, se rotará el grado de libertad *roll*), hay que aplicar un escalado a cada grado de libertad para adecuar su movimiento al del lápiz. Por tanto:

$$q_r = R_{eje6} \cdot K_r \quad q_p = R_{eje5} \cdot K_p \quad q_y = R_{eje4} \cdot K_y \quad q_j = T_{wee_opening} \cdot K_j \quad (4.15)$$

Siendo K_r, K_p, K_y y K_j los factores de escala para el *roll*, *pitch*, *yaw* y *jaw* respectivamente.

4.2.4 Control de los Motores

Conocido el ángulo que se quiere mover cada uno de los cuatro grados de libertad de la herramienta, hay que calcular el valor del tiempo en el que se debe aplicar el PWM a cada uno de los motores. Es por ello que hay que acudir a las ecuaciones calculadas de forma experimental en el capítulo 3, (3.1), (3.2), ..., (3.8), en las que, mediante un polinomio, se calcula dicho tiempo en función del grado deseado para cada movimiento. Se muestra una recopilación de estas ecuaciones en la tabla 4.2.

RECOPIACIÓN DE ECUACIONES		
Grado de libertad	Sentido	Ecuación que lo modela
Roll	Positivo	$y = -1,152 \cdot 10^{-8} \cdot x^4 + 7,168 \cdot 10^{-6} \cdot x^3 - 0,0007643 \cdot x^2 + 0,6797 \cdot x + 0,2506$
	Negativo	$y = 1,097 \cdot 10^{-8} \cdot x^4 - 5,68 \cdot 10^{-6} \cdot x^3 + 0,0004153 \cdot x^2 + 0,8546 \cdot x + 0,4253$
Pitch	Positivo	$y = 0,0001547 \cdot x^3 - 0,02193 \cdot x^2 + 1,771 \cdot x + 1,141 \cdot 10^{-15}$
	Negativo	$y = 0,0002792 \cdot x^3 - 0,03009 \cdot x^2 + 1,914 \cdot x - 0,3278$
Yaw	Positivo	$y = 9,401 \cdot 10^{-5} \cdot x^3 - 0,01595 \cdot x^2 + 1,484 \cdot x + 0,39996$
	Negativo	$y = 9,026 \cdot 10^{-5} \cdot x^3 - 0,01629 \cdot x^2 + 1,61 \cdot x + 1,482$
Jaw	Positivo	$y = 1,454 \cdot 10^{-5} \cdot x^3 - 0,005064 \cdot x^2 + 1,071 \cdot x - 0,04704$
	Negativo	$y = 7,182 \cdot 10^{-6} \cdot x^3 - 0,001936 \cdot x^2 + 1,064 \cdot x + 0,793$

Cuadro 4.2: Recopilación de ecuaciones de movimiento para el *roll*, *pitch*, *yaw* y *jaw*

El valor de PWM que se debe aplicar durante dicho tiempo depende de cada grado de libertad y su sentido, y se emplearán los mismos que se aplicaron durante el experimento (tabla 3.1).

4.3 CONCLUSIONES

Se ha visto que el esquema de control del sistema tiene dos partes bien diferenciadas: el control del robot y el control de la herramienta. Cada uno está formado por dos bloques de trabajo, entre los cuales consiguen calcular el valor necesario para mover adecuadamente el robot y los cuatro grados de libertad de la herramienta respectivamente.

El siguiente paso es crear el algoritmo que permita calcular todas estas variables y que consiga mover tanto el robot como la herramienta con el dispositivo háptico en tiempo real a través el ordenador.

IMPLEMENTACIÓN DEL ALGORITMO DE CONTROL

Ya diseñado el esquema de control de movimiento, para poder aplicarlo al robot real se debe implementar en forma de algoritmo. Para ello, se han utilizado tres plataformas diferentes: MATLAB para realizar el algoritmo de control de la herramienta, Arduino IDE para realizar el control de los motores, y ROS para realizar el control del robot y la conexión entre los dos dispositivos (el brazo robótico y el dispositivo háptico) y el resto de plataformas.

5.1 INTRODUCCIÓN A ROS

ROS (*Robot Operating System*) [9] es un meta sistema operativo capaz de controlar dispositivos a bajo nivel, comunicar procesos entre sí, manejar paquetes, etc. Se estructura de forma que el procesamiento se realiza a través de un conjunto de nodos, los cuales se encargan de enviar y recibir información a través de mensajes, llamados topics. Estos topics se pueden entender como unos buses en los que se intercambia información.

En nuestro sistema se trabaja con nodos y topics para comunicar los diferentes elementos. Concretamente tenemos cinco elementos diferenciados que hay que comunicar entre sí, los cuales se dividen en elementos reales (el brazo robótico y el dispositivo háptico), y en algoritmos de control (el algoritmo de control del robot, el algoritmo de control de la herramienta (MATLAB) y el control de motores (Arduino)).

En cuanto a los elementos reales, hay que comunicarlos con el ordenador a través de nodos. Estos elementos son:

- **Dispositivo háptico.** Para el dispositivo háptico se cuenta con el nodo *omni_common*, el cual establece la comunicación con el dispositivo háptico y además ofrece información de utilidad a través de distintos topics:
 - */phantom/pose* muestra la posición y orientación de la punta del lápiz.
 - */phantom/joint_states* muestra las posiciones articulares de cada articulación del dispositivo háptico.
 - */phantom/twee* ofrece información sobre la apertura y cierre del Twee, además del valor del botón (1 si está pulsado y 0 si no lo está).

- **Brazo robótico.** Para comunicar el brazo robótico con el ordenador se trabaja con dos nodos diferentes. El primer nodo es *uma_ur_launch*, el cual se encarga de establecer la comunicación con el robot y permite obtener información del mismo a través de dos topics:
 - */tf* muestra la posición y orientación del efector final del robot.
 - */joint_states* ofrece la posición articular de cada una de las articulaciones del robot.

El segundo nodo relacionado con el robot es *ur3_moveit_config*, el cual permite mover el robot gracias a una serie de comandos enviados a través del topic */ur_hardware_interface/script_command*. El comando utilizado en el proyecto es "*speedl(xd,a,t)*" [10], donde *xd* representa la velocidad en espacio cartesiano (velocidad en *x*, *y*, *z*, *rx*, *ry*, *rz*), *a* representa la aceleración, y *t* representa el tiempo durante el cual se está enviando el comando.

- */ur_hardware_interface/script_command* envía al robot la acción indicada según el comando utilizado y sus variables.

En cuanto a los algoritmos de control, estos deben ser comunicados entre sí y con los anteriores dispositivos. Los algoritmos utilizados son los siguientes:

- **Algoritmo de control del robot.** Este algoritmo está programado en lenguaje C a través de un nodo de ROS, llamado *force_position_control*. Este nodo se suscribe a los topics */phantom/pose* y */phantom/twee* para obtener las diferencias de posición del lápiz del dispositivo háptico y la lectura del botón del Twee.

Una vez calculada la posición y orientación del efector final del robot (con las operaciones (4.7) y (4.12) vistas en el capítulo 4, apartado 4.2.2), el nodo debe comandar al robot para que se mueva a dicha posición con su orientación correspondiente. Para ello, publica por el topic */ur_hardware_interface/script_command* con el comando de velocidad "*speedlz* los valores de posición y orientación calculados.

- **Algoritmo de control de la herramienta.** Este algoritmo se ha programado en MATLAB, y necesita cierta información acerca del dispositivo háptico para poder calcular los valores del *roll*, *pitch*, *yaw* y *jaw*. Concretamente, el algoritmo se suscribe a los siguientes topics: */phantom/pose*, */phantom/joint_states* y */phantom/twee*. Con la información de estos topics, calcula el valor de cada grado de libertad y los publica a través de cuatro topics:

- */EndoWrist/roll*. Contiene el valor del ángulo del *roll*.
- */EndoWrist/pitch*. Contiene el valor del ángulo del *pitch*.
- */EndoWrist/yaw*. Contiene el valor del ángulo del *yaw*.
- */EndoWrist/jaw*. Contiene el valor del ángulo del *jaw*.

- Algoritmo de control de los motores.** El control de los motores se programa en la plataforma Arduino IDE, en lenguaje C. Para conocer los valores de los ángulos deseados para cada grado de libertad, este código debe subscribirse a los topics */EndoWrist/<grado de libertad>* presentados anteriormente.

Con estos valores, calcula el tiempo en el que se debe aplicar el PWM a cada motor con las ecuaciones vistas en la tabla 4.2.

En la figura 5.1 se muestra un esquema con los nodos y los topics y su relación con el sistema completo.

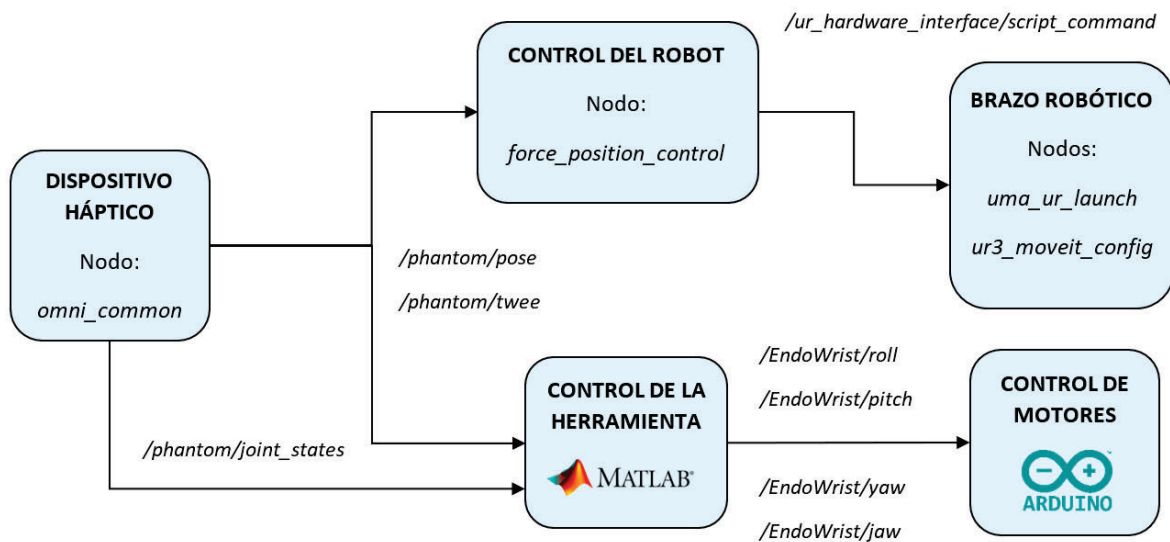


Figura 5.1: Nodos y topics del sistema

A continuación se desarrolla en profundidad cada uno de los nodos de los dispositivos y los algoritmos de control del sistema.

5.2 DESARROLLO POR BLOQUES

Tras ver una breve presentación sobre la estructura en software, se desarrolla cada uno de los elementos presentes para conocer su lenguaje y su comunicación con el resto de elementos.

5.2.1 Dispositivo háptico

El dispositivo háptico se comunica con el ordenador a través de ROS gracias al nodo *omni_common*, cuyos topics se describieron anteriormente. Estos topics son */phantom/pose*, */phantom/joint_states* y */phantom/twee*.

Se puede ver qué información envía cada uno de estos topics en la terminal del ordenador a través del comando *rostopic echo <topic>*:

- */phantom/pose* muestra la posición y orientación de la punta del lápiz (figura 5.2).

```
labrob@labrob-All-Series:~$ rostopic echo /phantom/pose
header:
  seq: 68100
  stamp:
    secs: 1690533183
    nsecs: 133451429
  frame_id: "/map"
pose:
  position:
    x: 0.0
    y: 0.0881141967773
    z: -0.0655107116699
  orientation:
    x: -0.152067265506
    y: -0.145365150339
    z: -0.675662779927
    w: 0.706558085121
```

Figura 5.2: Comando *rostopic echo /phantom/pose* en la terminal del ordenador

Se puede ver que, además de otros datos, nos ofrece la posición cartesiana (x , y , z) y la orientación en cuaternios (x , y , z , w) del lápiz.

- */phantom/joint_states* muestra las posiciones articulares del dispositivo háptico (figura 5.3).

```
labrob@labrob-All-Series:~$ rostopic echo /phantom/joint_states
header:
  seq: 141009
  stamp:
    secs: 1690533256
    nsecs: 45167207
  frame_id: ""
name:
  - waist
  - shoulder
  - elbow
  - yaw
  - pitch
  - roll
position: [-0.0, 0.26888954639434814, -0.6397024393081665, 3.1414862
448197134, -3.150913888214043, -3.0969561358266553]
velocity: []
effort: []
```

Figura 5.3: Comando *rostopic echo /phantom/joint_states* en la terminal del ordenador

Se observa cómo dicho topic envía las posiciones articulares de cada uno de los 6 grados de libertad del dispositivo háptico: *waist*, *shoulder*, *elbow*, *yaw*, *pitch* y *roll*, mostrados con sus ejes en la figura 4.7 del apartado 4.2.3.

- */phantom/twee* muestra el grado de apertura del Twee y si el botón está pulsado o no (figura 5.4).

```
labrob@labrob-All-Series:~$ rostopic echo /phantom/twee
twee_button: 0
twee_angle: 41
```

Figura 5.4: Comando *rostopic echo /phantom/twee* en la terminal del ordenador

Se observa que, en este caso, el botón no está pulsado (*twee_button* : 0) y que el ángulo de apertura es de 41 (entre 0 y 100).

5.2.2 Brazo robótico

El brazo robótico se comunica de igual forma con el ordenador, a través de ROS. Este trabaja con dos nodos diferenciados: *uma_ur_launch* para la comunicación del robot con el ordenador, y *ur3_moveit_config* para poder manipularlo desde el mismo.

Para el primer nodo, *uma_ur_launch*, tenemos los siguientes topics que nos ofrecen información del robot y que podemos leer con el comando *rostopic echo <topic>*:

- */tf* ofrece la posición y orientación del efector final del robot (figura 5.5).

```
labrob@labrob-All-Series:~$ rostopic echo /tf
transforms:
-
  header:
    seq: 0
    stamp:
      secs: 1690534775
      nsecs: 458206961
    frame_id: "upper_arm_link"
  child_frame_id: "forearm_link"
  transform:
    translation:
      x: -0.244125293179
      y: 0.0
      z: 0.0
    rotation:
      x: 0.00433593755987
      y: 0.0013045295167
      z: -0.043416126814
      w: 0.999046814609
```

Figura 5.5: Comando *rostopic echo /tf* en la terminal del ordenador

Se tiene la posición cartesiana (x, y, z) y la orientación en cuaternios (x, y, z, w) del efector final del robot dentro del grupo 'transform'.

- `/joint_states` ofrece las posiciones articulares del brazo robótico (figura 5.6).

```
labrob@labrob-All-Series:~$ rostopic echo /joint_states
header:
  seq: 8480
  stamp:
    secs: 1690534817
    nsecs: 824400478
  frame_id: ''
name:
- elbow_joint
- shoulder_lift_joint
- shoulder_pan_joint
- wrist_1_joint
- wrist_2_joint
- wrist_3_joint
position: [-0.08683234850038701, -0.7611649672137659, 1.0474613904953003, 3.9684
00478363037, -1.8699286619769495, 6.340178162102596]
velocity: [-0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
effort: [-0.6222168207168579, -1.0587159395217896, -0.0403747595846653, -0.07049
561129713058, 0.022379877045750618, 0.03245082125067711]
```

Figura 5.6: Comando `rostopic echo /joint_states` en la terminal del ordenador

Se puede ver la posición articular de cada uno de los seis grados de libertad del robot: `elbow_joint`, `shoulder_lift_joint`, `shoulder_pan_joint`, `wrist_1_joint`, `wrist_2_joint` y `wrist_3_joint`.

Para el segundo nodo, `ur3_moveit_config`, tenemos el topic `/ur_hardware_interface/script_command`, el cual es un topic de publicación por el cual se publican los comandos vistos anteriormente. Si se publica un comando de velocidad como los que utilizaremos en nuestro proyecto, este se vería tal y como muestra la figura 5.7:

```
labrob@labrob-All-Series:~$ rostopic pub /ur_hardware_interface/script_command
std_msgs/String "speedl([0.5,0.4,0,0,1.57,0],0.5,0.008)"
```

Figura 5.7: Comando `rostopic echo /ur_hardware_interface/script_command` en la terminal

En dicho comando se indica que se mueva un incremento en X de 0,5 m, en Y de 0,4 m, y una rotación en Y de 90° ($\pi/2$ radianes), con una aceleración de 0,5 m/s y durante un tiempo de 0,008 segundos (equivalente a la frecuencia del robot, 125 Hz).

5.2.3 Algoritmo de control

El algoritmo de control se ha implementado en tres partes bien diferenciadas: una para el control del movimiento del robot (parte superior del esquema de control, figura 4.5), otra para el control de la punta de la herramienta, y otra para el control de los motores de la misma (parte inferior del esquema de control, figura 4.5).

- **Algoritmo de control del robot.** El algoritmo de control para el movimiento del robot alrededor del punto de fulcro ha sido proporcionado por el equipo de investigación de robótica médica de la Universidad de Málaga. Está implementado en lenguaje C y todo el código conforma un nodo llamado *force_position_control*, como se vió anteriormente.

En él se implementa la transformación inicial entre el dispositivo háptico y la herramienta (bloque *Initial Transform* del esquema de control), y el control alrededor del punto de fulcro (bloque *Fulcrum Control* del esquema de control). Se tiene como resultado la posición y orientación del efector final del robot, el cual se comanda al mismo con el comando visto en el apartado 5.2.2, '*speedl*'.

- **Algoritmo de control de la herramienta.** En cuanto al algoritmo de control de la punta de la herramienta, este se ha programado en MATLAB. Leyendo el valor de los ejes del dispositivo háptico (figura 4.7), calcula el valor de cada grado de libertad (*roll*, *pitch*, *yaw* y *jaw*) con su correspondiente escalado (bloque *Inverse Kinematic for Orientation* del esquema de control) y lo publica a través de los topics correspondientes, */EndoWrist/<grado de libertad>*.

- **Algoritmo de control de los motores.** El control de los motores comprende el último bloque del esquema de control, *Motor Control*. Está formado por un código escrito en C en la plataforma Arduino IDE. Tiene como misión comandar a los motores su movimiento a través de pulsos PWM según el grado de libertad escogido. Por tanto, los valores de entrada serán los grados que se quiere mover cada grado de libertad (*roll*, *pitch*, *yaw* y *jaw*) y como salida se tendrá el tiempo en el que hay que aplicar el pulso PWM a cada uno de los cuatro motores que controlan la herramienta.

Si recordamos la tabla 3.1 del apartado 3.1, cada grado de libertad se mueve con una combinación distinta de motores, por lo que se debe programar adecuadamente esta combinación y su valor de PWM correspondiente.

Para saber cuánto tiempo se debe enviar este pulso PWM, acudimos a las interpolaciones polinómicas desarrolladas en el capítulo 3 (resumidas en la tabla 4.2, y así se calcula el tiempo necesario para alcanzar los grados deseados.

En la figura 5.8 se muestra una parte del código que muestra la implementación del control del movimiento para el *roll* en Arduino IDE.

```
void endowristxiService::move_roll(int angle){
    int pwm = 100;
    char dirf[10] = "forward";
    char dirb[10] = "backward";

    Serial.print(angle);

    if (angle>=0) {
        double time = 9*pow(10,-12)*pow(angle,6)-1*pow(10,-8)*pow(angle,5)
        +4*pow(10,-6)*pow(angle,4)-0.0008*pow(angle,3)+0.0544*pow(angle,2)+1.9606*angle;
        motor1.move(dirf,pwm);
        delay(time);
    }
    else if (angle < 0) {
        angle = -angle;
        double time = 6*pow(10,-12)*pow(angle,6)-7*pow(10,-9)*pow(angle,5)
        +3*pow(10,-6)*pow(angle,4)-0.0006*pow(angle,3)+0.0392*pow(angle,2)+2.2205*angle;
        motor1.move(dirb,pwm);
        delay(time);
    }
}
```

Figura 5.8: Código correspondiente al *roll* en Arduino IDE

De esta forma, la herramienta se moverá en sus cuatro grados de libertad la cantidad indicada por la orientación del dispositivo háptico.

5.3 CONCLUSIONES

La implementación del algoritmo de control es esencial para poder ejecutar el control del sistema. Es por ello que ha sido necesario dividir su implementación en tres códigos diferenciados: uno para el control del brazo robótico alrededor del punto de fulcro con el dispositivo háptico, otro para el control de los grados de libertad de la herramienta de nuevo con el dispositivo háptico, y otro para el control de los motores que mueven estos grados de libertad.

En el siguiente capítulo se exponen los resultados de aplicar estos algoritmos al sistema real y la demostración de su buen funcionamiento.

PUESTA EN MARCHA Y RESULTADOS

En este capítulo se presentan los resultados de llevar a cabo las pruebas mostradas durante este documento con el fin de evaluar la solidez del proyecto realizado, mostrando si se logran de manera satisfactoria los objetivos establecidos.

Inicialmente, en el capítulo 3, se realizaron las pruebas convenientes para conocer la herramienta *EndoWrist* y poder controlarla adecuadamente, por lo que se mostrará su correcto funcionamiento. Tras el diseño del esquema de control y su desarrollo e implementación, se verá también qué resultados ofrece el movimiento del robot con el dispositivo háptico. Por último, se comprobará la implementación de todo el proyecto en conjunto.

6.1 MOVIMIENTO DEL INSTRUMENTO

Se debe probar el buen funcionamiento de la herramienta controlada a través del dispositivo háptico. Ya se vió en el capítulo 3 cómo la herramienta mueve cada uno de sus cuatro grados de libertad con un error de entre 4° y 12° .

En este caso, será el dispositivo háptico el que enviará el valor de los grados que se tiene que mover cada uno de ellos, como se vió en el apartado 4.2.3. Se realizarán diversas pruebas para ver cómo se mueve la herramienta con cada uno de los cuatro movimientos que ofrece el dispositivo háptico.

- **ROLL.** Para mover el *roll*, se debe mover el lápiz alrededor del eje 6, es decir, rotando el lápiz sobre sí mismo (figura 6.1).

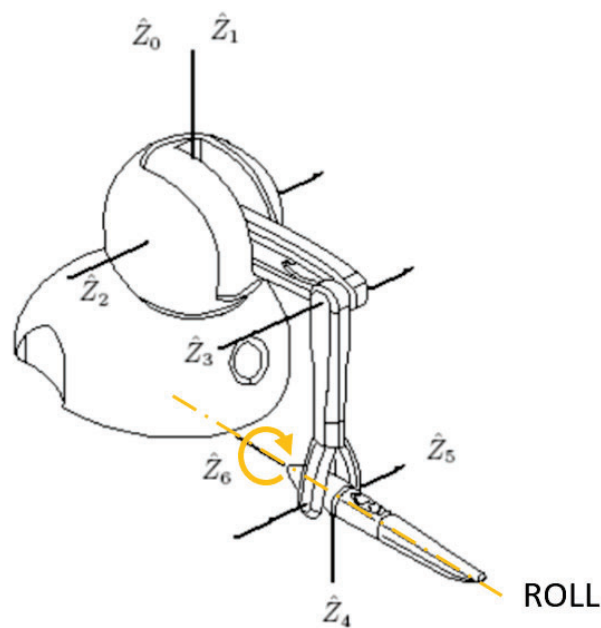


Figura 6.1: Eje correspondiente al *roll* en el dispositivo háptico

El resultado que arroja este movimiento se visualiza en la figura 6.2.



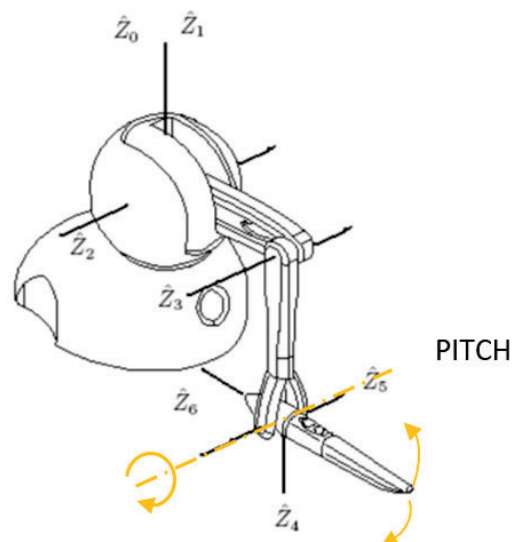
(a) Posición inicial

(b) Posición tras el movimiento

Figura 6.2: Movimiento del *roll* con el lápiz

Se aprecia cómo la herramienta ha girado el *roll* tras girar el lápiz.

- **PITCH.** Para mover el *pitch*, se debe mover el lápiz en el eje 5, es decir, rotando hacia arriba y hacia abajo (figura 6.3).

Figura 6.3: Movimiento del *pitch* en el dispositivo háptico

El resultado que ofrece este movimiento en la herramienta real se muestra en la figura 6.4.



(a) Posición inicial

(b) Posición tras el movimiento

Figura 6.4: Movimiento del *pitch* con el lápiz

Se observa cómo, al rotar el lápiz hacia arriba, el *pitch* se mueve en la misma dirección.

- **YAW.** El *yaw* se mueve alrededor del eje 4 del dispositivo háptico, es decir, moviendo el lápiz de izquierda a derecha y viceversa (figura 6.5).

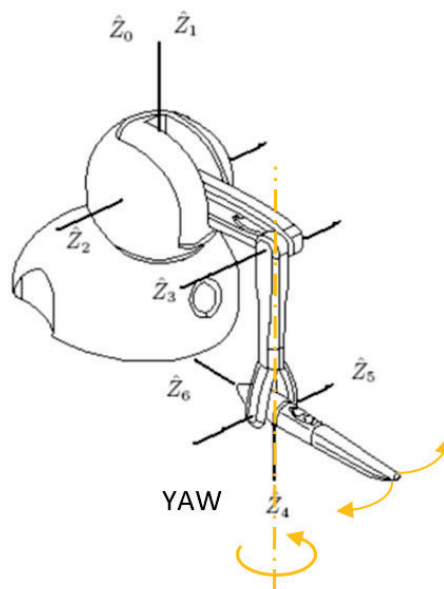
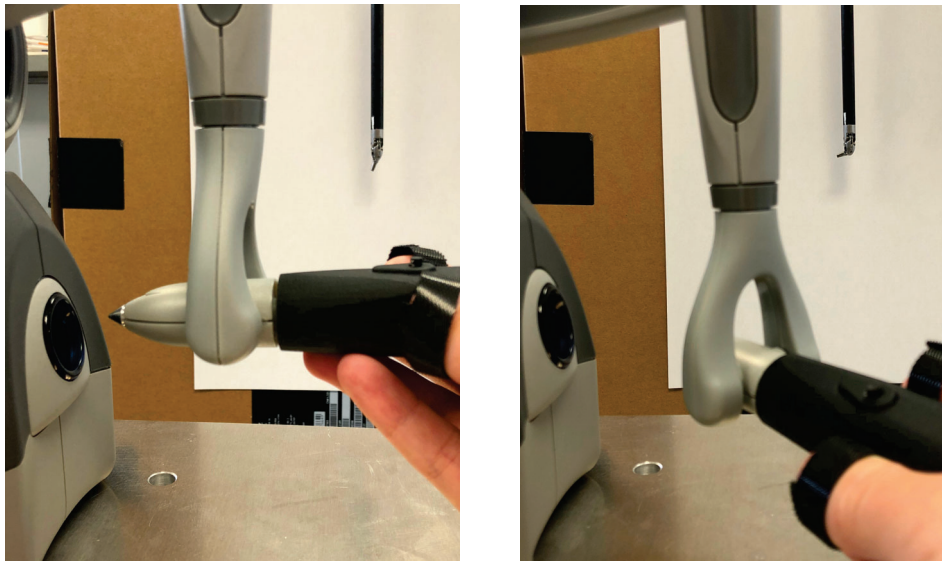


Figura 6.5: Movimiento del *yaw* en el dispositivo háptico

El movimiento real de la herramienta ante este giro se muestra en la figura 6.6.



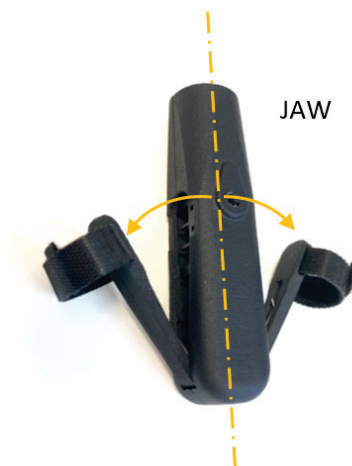
(a) Posición inicial

(b) Posición tras el movimiento

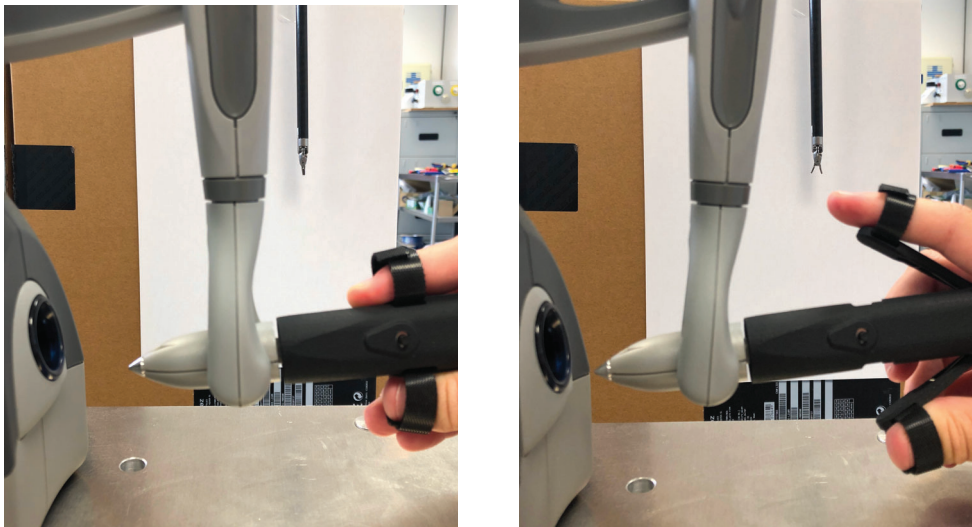
Figura 6.6: Movimiento del *yaw* con el lápiz

Se observa cómo, al girar el lápiz hacia la derecha, el *yaw* gira el valor correspondiente.

- **JAW.** Para mover el *jaw* se utiliza el Twee, y por tanto se sincroniza la apertura del Twee con la apertura de la pinza, y lo mismo para el cierre (figura 6.7).

Figura 6.7: Movimiento del *jaw* en el Twee

La apertura de la pinza real tras este movimiento se muestra en la figura 6.8.



(a) Posición inicial

(b) Posición tras el movimiento

Figura 6.8: Movimiento del *jaw* con el lápiz

Se puede ver cómo la pinza se abre cuando se abre el Twee.

6.2 MOVIMIENTO DEL ROBOT

Esta prueba consiste en mover el robot con el dispositivo háptico y ver que, efectivamente, este se mueve en sintonía con el movimiento del lápiz, rotando alrededor del punto de fulcro. Se comprobará este movimiento en dos fases diferenciadas: primero en una simulación, dibujando el valor del efector final del robot calculado y viendo si pivota alrededor del fulcro; y más tarde en el robot real, en el entorno de prácticas del laboratorio de investigación de robótica de la Universidad de Málaga.

6.2.1 Prueba en simulación

Para la prueba en simulación se dispone de un programa realizado en MATLAB que trata de visualizar el movimiento del efector final del robot y dibujar su trayectoria. En él se verá el eje de la herramienta y si pivota alrededor del punto de fulcro.

Se utilizará la *ToolBox Robotic Systems* de MATLAB para simular el robot real. Al lanzar el código, se abrirá una gráfica en 3D en la cual veremos la simulación (figura

6.9), inicialmente estando colocado el robot en la posición indicada en el código. Una vez se muestre la gráfica, se puede mover el dispositivo háptico y visualizar en tiempo real cómo se mueve el robot.

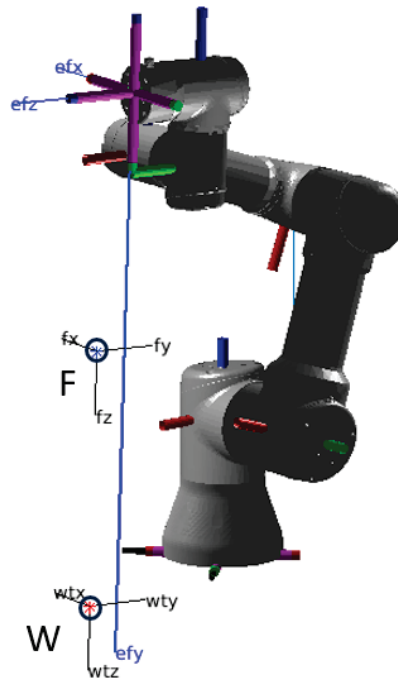


Figura 6.9: Posición inicial del robot en simulación

Se visualiza en azul el eje Y del efector final, el mismo que el eje Z de la herramienta, pero sin el desplazamiento provocado por las dimensiones del acople. Se muestra también el punto de fulcro con su orientación correspondiente (punto F) y la muñeca de la herramienta con su orientación (punto W), las cuales están separadas del eje Y de la herramienta por el mismo motivo explicado anteriormente, las dimensiones del acople.

Si, por ejemplo, se mueve el lápiz a la izquierda, la muñeca se moverá a la izquierda, el punto de fulcro se mantendrá, y el efector final del robot se moverá a la derecha para conseguir el movimiento deseado. El resultado de la simulación se muestra en la figura 6.10:

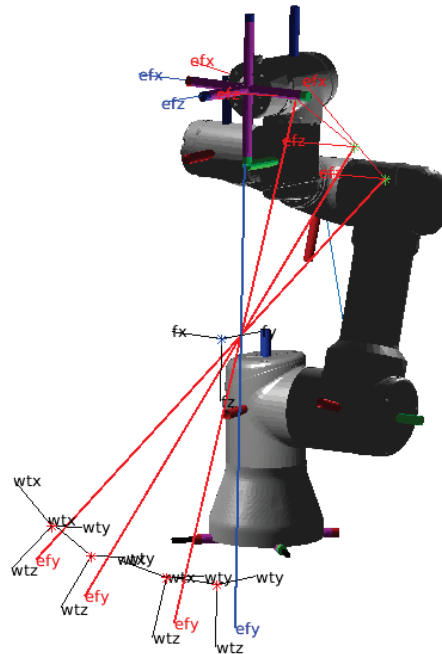


Figura 6.10: Simulación del movimiento del robot

Se puede observar cómo, al mover el lápiz hacia la izquierda, la muñeca de la herramienta se mueve correctamente a la izquierda. Se muestran los puntos de la muñeca y su orientación, la cual siempre acompaña al eje de la herramienta.

Se visualiza cómo el efector final se mueve de tal forma que la muñeca se coloca donde debe, rotando siempre alrededor del punto de fulcro. De esta forma, todos los ejes del efector final deben cortar en el punto de fulcro (líneas rojas), pero desplazado de nuevo por las dimensiones del acople.

Se observa que el sistema funciona correctamente, y está listo para ser probado en el robot real.

6.2.2 Prueba en el robot real

Para probar el experimento en el robot real, se conectará el mismo con los algoritmos de control a través del entorno de ROS. Por ello, se debe lanzar el nodo con el algoritmo del control del robot (*force_position_control*) junto con los nodos del robot (*uma_ur_launch* y *ur3_moveit_config*) y el nodo del Phantom Omni (*omni_common*). Cuando estén todos operativos, el robot estará listo para moverse.

El robot, por seguridad, nunca funcionará si no se tiene pulsado el botón del Twee en el dispositivo háptico. De esta forma, teniendo pulsado el botón, se moverá el robot en sintonía con el movimiento del lápiz. Si el lápiz se mueve a la derecha, la muñeca de la herramienta deberá ir a la derecha, por lo que el robot se moverá de tal forma que mueva la muñeca al sitio correspondiente respetando el punto de fulcro.

Se muestra una prueba de su funcionamiento en la figura 6.11.



(a) Reposo



(b) Movimiento hacia adelante

Figura 6.11: Movimiento real del robot

Se puede ver cómo la muñeca se mueve hacia adelante, al igual que el lápiz, y por consiguiente el efector final del robot se mueve hacia atrás, respetando así la rotación alrededor del punto de fulcro.

6.3 MOVIMIENTO DEL CONJUNTO DEL ROBOT CON LA HERRAMIENTA

Para que el experimento esté completo, se debe probar que funciona todo el sistema a mismo tiempo: el movimiento del robot alrededor del punto de fulcro y el movimiento de la punta de la herramienta, ambos con el dispositivo háptico.

Como ambos deben estar controlados por el mismo dispositivo, se ha programado la tarea de la siguiente forma:

- Si se tiene pulsado el botón del Twee, se moverá el robot, tal y como se hizo en la prueba del apartado 6.2.2.
- Si no se tiene pulsado el botón, se controlarán los grados de libertad de la herramienta como se hizo en la prueba del apartado 6.1.

El experimento completo se ha realizado en el entorno de prácticas del laboratorio de investigación de robótica médica de la Universidad de Málaga. En dicho entorno se encuentra un espacio dedicado a la simulación de un paciente, con una prótesis simulando el abdomen del mismo (figura 6.12). En esta última prueba se trabajará sobre dicho espacio para mayor realismo del experimento.

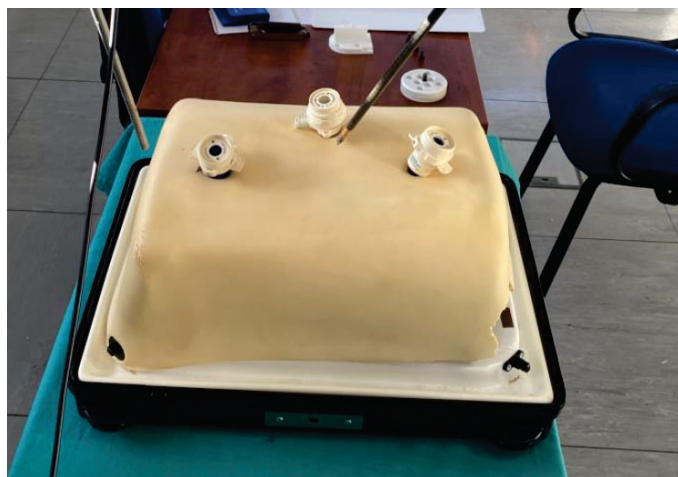
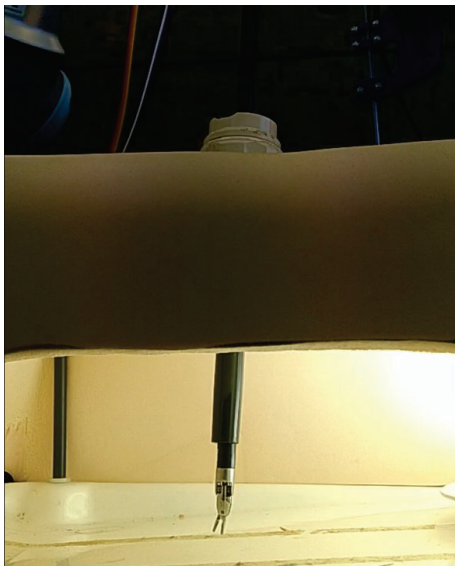


Figura 6.12: Falso abdomen del paciente

Se introduce la herramienta dentro del 'abdomen' y se procede a evaluar el experimento. Primero se moverá el robot dentro del mismo para colocarlo en la posición deseada, y deberá pivotar sobre el punto de fulcro, en este caso el borde del abdomen (figura 6.13). Una vez colocado el robot, se procederá a mover la punta de la herramienta según el movimiento que se desee (figura 6.14).

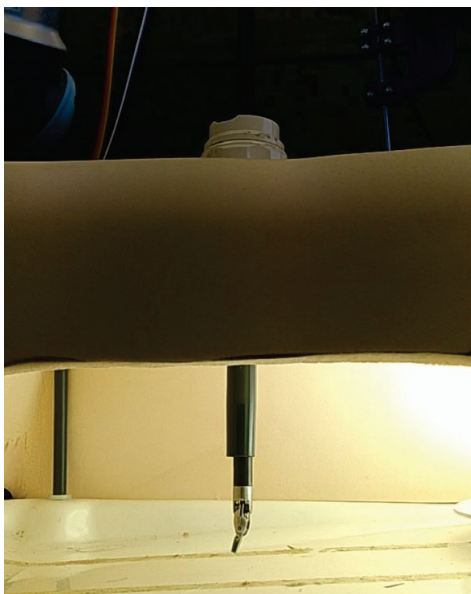


(a) Reposo

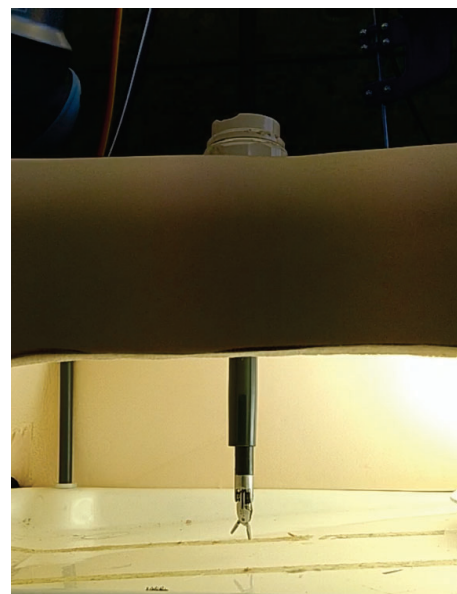


(b) Movimiento hacia la derecha

Figura 6.13: Movimiento del robot en el abdomen



(a) Reposo



(b) Apertura de la pinza

Figura 6.14: Movimiento de la pinza en el abdomen

Se observa cómo el robot pivota correctamente y se mueve ligeramente a la derecha, y por otro lado se ve cómo se abre la pinza dentro del abdomen.

Se puede decir que el resultado del experimento final es exitoso.

6.4 CONCLUSIONES

Tras realizar pruebas por separado de cada una de las partes del experimento y comprobar su buen funcionamiento, se ha realizado el experimento completo y se puede concluir con un buen resultado.

Hay ciertos aspectos del experimento que se pueden mejorar para un mejor rendimiento, y se comentarán en el siguiente apartado.

CONCLUSIONES FINALES Y FUTURAS LÍNEAS DE TRABAJO

Una vez finalizado el experimento y comprobado su buen funcionamiento, se recopilan las conclusiones del proyecto y se plantean futuras líneas de trabajo para mejorar el conjunto del experimento.

7.1 CONCLUSIONES FINALES

Concluidas todas las fases del proyecto, se puede comprobar el buen funcionamiento del sistema y la consecución del objetivo propuesto. Durante el presente documento se han ido exponiendo los pasos para ir completando cada fase del proyecto, los inconvenientes y su solución en cada momento. Además, se han realizado experimentos para comprobar los resultados y mejorar los errores que proporcionan. En general, se han conseguido los dos experimentos individuales:

- Por un lado, se ha controlado el movimiento de cada uno de los cuatro grados de libertad que ofrece la herramienta *EndoWrist* con un error de entre 4° y 12°.
- Por otro lado, ha sido posible el control del robot con el dispositivo háptico alrededor del punto de fulcro.

Finalmente, en la integración de los dos experimentos en un único sistema que sea capaz de controlar tanto el robot como la herramienta con el mismo dispositivo háptico, se ha conseguido un resultado satisfactorio, siendo el sistema capaz de controlar por un lado el robot si el botón del dispositivo está pulsado, y la herramienta por otro lado si el botón no está pulsado.

Sin embargo, el experimento final tiene márgenes de mejora, pudiendo añadir otros elementos que ayuden a un mejor control del sistema. Dichas mejoras se explican en el siguiente apartado.

7.2 FUTURAS LÍNEAS DE TRABAJO

Una vez obtenido un buen resultado final, es conveniente plantear futuras líneas de trabajo diseñadas para un mejor funcionamiento del sistema y un crecimiento en esta área de trabajo.

Por ello, se plantean dos mejoras para el proyecto: una sustitución de motores y un nuevo elemento para el sistema.

En cuanto al primer aspecto, el movimiento de los grados de libertad de la herramientas están condicionados por un control en bucle abierto, lo que no permite conocer su estado real en ningún momento. Eso se debe a la falta de codificadores en los motores para poder leer la posición. Es por ello que se propone cambiar los cuatro motores

DC por unos motores servo con codificadores que permitan controlar la posición de cada grado de libertad en cada momento y, de esta forma, poder hacer un control más completo y con menos errores.

Por otro lado, dado que el sistema sólo ofrece dos posibles estados al tener un único botón, sólo se tienen dos movimientos: control del brazo robótico cuando se pulsa el botón, y control de la herramienta cuando no se pulsa. Esto implica que no existe un modo de reposo ni de seguridad en el que no se realice ningún movimiento, lo que nos abre un camino a introducir un nuevo elemento que permita añadir un nuevo estado.

Es por ello que se propone añadir al sistema un pedal, el cual añadirá dos estados nuevos en los que repartir los movimientos con mayor seguridad. De esta forma, se tendrían cuatro estados: botón sin pulsar y pedal sin pisar (00), botón pulsado y pedal sin pisar (01), botón sin pulsar y pedal pisado (10), y botón pulsado y pedal pisado (11). Se podría mover el robot en uno de los estados, la herramienta en otro estado, un estado de reposo y seguridad, y un estado sobrante que se podría utilizar para cualquier acción, como puede ser iniciaciones o movimientos a puntos de reposo.

BIBLIOGRAFÍA

- [1] “Da Vinci surgical systems,” <https://www.intuitive.com/en-us/products-and-services/da-vinci/systems>. (page 2).
- [2] “Da Vinci instruments,” <https://www.intuitive.com/en-gb/products-and-services/da-vinci/instruments>. (page 2).
- [3] J. W. P. W. S. H. W. K. H. K. H. R. Dong Hoon Koh, Won Sik Jang and Y. D. Choi, “Efficacy and safety of robotic procedures performed using the da Vinci robotic surgical system at a single institute in Korea: Experience with 10,000 cases,” September 2018. (page 3).
- [4] H. T. Galindo, *Trabajo fin de grado. Diseño e implementación de un mecanismo de control de un instrumento quirúrgico EndoWrist*. Universidad de Málaga, 2022. (page 3).
- [5] O. A. D.-R. P. O. A. Jarrillo-Silva, V. Parra-Vega, *PHANTOM OMNI Haptic Device: Kinematic and Manipulability*, 2009. (page 14).
- [6] “Twee,” <https://www.bbzsrl.com/index.php/materialis/services/#twee>. (page 15).
- [7] J. M. M. Sérgio S. Afonso, “Development of a gestural master interface for tele-surgery applications,” Mechanical Engineering Department, Instituto Superior Técnico, Universidade de Lisboa, Tech. Rep., November 2014. (page 18).
- [8] J. A. Beckman, *The PHANTOM Omni as an under-actuated robot*. Iowa State University, 2007. (page 46).
- [9] “Ros – robot operating system,” <https://www.ros.org>. (page 50).
- [10] U. Robots, *The URScript Programming Language*, 2019. (page 51).