UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

Diseño e implementación de un mecanismo en configuración de muñeca esférica para experimentos de control de fuerza en superficies virtuales

TESIS

Que para obtener el título de

INGENIERO ELÉCTRICO ELECTRÓNICO

PRESENTA

Enrique Martínez Fernández

DIRECTOR DE TESIS

Dr. Marco Antonio Arteaga Pérez

Ciudad Universitaria, Cd. Mx., 2018

JURADO:

PRESIDENTE: Dr. Paul Rolando Maya Ortiz

VOCAL: M.I. José Daniel Castro Díaz

SECRETARIO: Dr. Marcos Ángel González Olvera

1ER. SUPLENTE: Dr. Hoover Mujica Ortega

2DO. SUPLENTE: M.I. Mauro Gilberto López Rodríguez

Asesor:

Dr. Marco Antonio Arteaga Pérez

Agradecimientos

A mi querida Universidad Nacional Autónoma de México por brindarme una excelente educación profesional.

A mis padres por su apoyo incondicional y confianza.

A mis hermanas por estar conmigo siempre y hacer cada día especial.

A todos los profesores que participaron en mi formación como ingeniero.

Al Dr. Marco Arteaga por recibirme y darme la oportunidad de realizar este trabajo.

A mi asesor Daniel Castro por compartir conmigo su tiempo y conocimientos, por su dedicación y por ser un gran amigo.

A mis compañeros del Laboratorio de Robótica por apoyarme en momentos difíciles y ayudarme siempre que lo necesité.

A mis amigos por acompañarme en este largo viaje.

A la UNAM por su apoyo a través del proyecto PAPIIT IN114617.

Dedicatoria

A mis padres que con su inmenso amor, dedicación, esfuerzo y sabias enseñanzas me permitieron llegar hasta aquí. Gracias por nunca dejar de creer en mí.

Índice general

Capítulo 1

Introducción

Los seres vivos están rodeados de una enorme cantidad información constantemente. Para poder subsistir en su entorno, requieren de medios que les permitan procesarla y responder en consecuencia, lo cual se logra a través de los órganos que conforman los sentidos. Gracias a ellos, simples fenómenos físicos se pueden transformar en sensaciones: una vibración en el aire se puede percibir como sonido, una molécula como un olor y una onda electromagnética como un color o calidez en la piel. Teniendo en cuenta la interacción individuo-ambiente que se acaba de describir, se desea definir el término *realidad*. En esencia, esta palabra significa la existencia efectiva de algo. ¿Cómo sabe un ser vivo que algo *en verdad existe*? La forma más natural de hacerlo es a través de sus órganos sensoriales y los efectos que provocan. Por ello se definirá *realidad* como el conjunto de objetos animados o inanimados que están en el ambiente e interactúan con cada ser vivo.

Debido a que la forma en que un individuo puede conocer su entorno es a través de los sentidos, se puede plantear la pregunta: ¿todo lo que se percibe a través de ellos es *real*? Antes de contestar, sería conveniente describir un breve experimento. Si una persona junta ambas manos, una al lado de la otra, con los puños cerrados y las palmas paralelas al piso, después extiende los brazos y levanta los dedos índices y los coloca de forma tal que se puedan ver de frente cerrando el ojo derecho y ubicando la vista en el dedo del mismo lado para posteriormente mover lentamente el brazo izquierdo hacia el costado, entonces desde su perspectiva, en algún punto del recorrido el dedo desaparecerá. ¿Quiere decir que este no existe? Evidentemente no, simplemente se está pasando por una zona de la retina conocida como punto ciego, probando así que los sentidos no son infalibles.

Otro ejemplo se muestra en la Figura 1.1. Parece que en algunos momentos hay puntos negros, pero luego desaparecen. Entonces, ¿existen o no? En realidad no están ahí, pero para el individuo que los percibe *parecen reales*, al menos momentáneamente. La ilusión óptica creada es muy simple, pero demuestra que si se excita a los sentidos de forma adecuada, es posible inducir sensaciones de forma *artificial*.

Pero, ¿para qué sirve hacer creer a una persona que está sucediendo algo? Hay una gran cantidad de situaciones en las que puede resultar útil hacerlo. Un ejemplo simple es

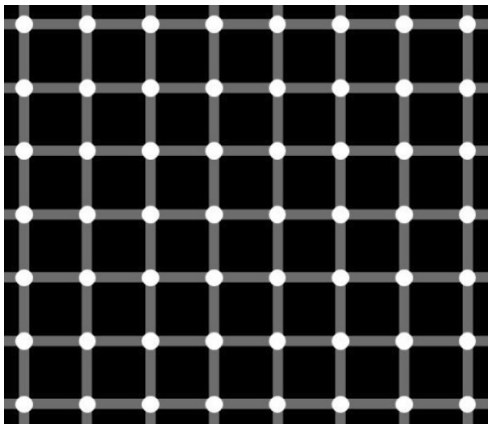


Figura 1.1: Ilusión óptica.

en el entretenimiento. La experiencia de ver una película puede ser mucho más placentera si las imágenes y el sonido son lo suficientemente convincentes, es decir, si logran crear una experiencia similar a la que la persona percibiría si estuviera viviendo la situación ella misma.

Si al usuario le es difícil o incluso imposible distinguir entre los estímulos provenientes de la realidad y los que se generan artificialmente, entonces se puede decir que la persona se encuentra *inmersa* en ese entorno. Más formalmente, se puede definir *inmersión* como la *sensación de estar presente* en un ambiente (Sherman and Craig, 2002). Es posible agrupar a las técnicas que tratan de lograr dicho objetivo con el término *realidad virtual*, lo que a continuación se estudiará con mayor detalle: qué es, para qué sirve y su importancia.

1.1. Estado del arte

Antes de dar una definición formal de realidad virtual resulta conveniente estudiar brevemente las aplicaciones que tiene y los componentes que la conforman. Una vez que se conozcan ambas cuestiones, será más fácil definir el término de forma concreta.

La importancia de estudiar la realidad virtual radica en sus aplicaciones. Una que viene a la mente casi de forma natural es el entretenimiento. Entre más tecnologías haya que involucren a los sentidos, este será cada vez más inmersivo. Si es más realista, más se involucrará el usuario con lo que se le quiere transmitir. Incluso se podría evocar sentimientos: miedo, tristeza o felicidad. Pero esta tecnología no está limitada a esta área, por lo que a continuación se da una pequeña lista de ejemplos donde se ha usado realidad virtual en aplicaciones reales.

- Un estudio hecho en (Rothbaum et al., 1995) utiliza un dispositivo de realidad virtual para estudiar la posibilidad de usarla al tratar acrofobia (miedo a las alturas).

- En (Lange et al., 2009), se investiga la efectividad de los videojuegos para lograr rehabilitación motriz en pacientes con lesión de médula espinal o que sufrieron un derrame cerebral. Se usan varias consolas y dispositivos relacionados, entre ellos un robot paralelo llamado Novint Falcon.
- Otro ejemplo es el trabajo presentado en (Luciano et al., 2009), donde se utiliza realimentación visual y háptica para desarrollar un simulador que sirve para entrenar a futuros odontólogos.
- En (Sánchez and Tadres, 2010) se desarrolló y probó un juego que usa realimentación auditiva y táctil mediante el robot Novint Falcon para evaluar qué tan precisa es la representación mental del ambiente virtual que logran obtener pacientes invidentes.
- En (Linda and Manic, 2011), se usa al mismo robot para dar realimentación de fuerza a un operador que controla un grupo de robots trabajando colaborativamente.

Aunque aquí se citan sólo algunos ejemplos, la idea es clara: la realidad virtual puede ser extremadamente útil ya sea para aplicaciones de entretenimiento, de ingeniería o médicas. Por lo tanto, definitivamente es conveniente estudiar y desarrollar este tipo de tecnologías.

1.1.1. Componentes de la Realidad Virtual

De acuerdo con (Sherman and Craig, 2002), hay cuatro elementos esenciales que conforman un sistema de realidad virtual, los cuales son:

- Ambiente virtual. Se puede definir como el conjunto de objetos en un mundo y las relaciones que los gobiernan. Generalmente este entorno se crea a través de una computadora. Las leyes que rigen este *universo* deben ser similares a las del mundo real para poder crear sensaciones convincentes.
- Inmersión. Corresponde con la definición que se ha dado anteriormente: hacer sentir al usuario que está presente en un ambiente.
- Realimentación sensorial. Se refiere a el uso de dispositivos para enviar información de forma artificial a los sentidos. Generalmente se estimula la vista y el oído, pero también se puede hacer con otros sentidos.
- Interactividad. Se logra cuando el ambiente responde a las acciones del usuario. Para que esto sea posible, debe existir una representación de la persona en el entorno virtual, la cual es llamada *avatar*.

Sin alguno de estos componentes el sistema de realidad virtual está incompleto. Por ejemplo, sin realimentación sensorial el usuario jamás sabrá qué está pasando en el ambiente virtual y no podrá haber interacción.

1.1.2. Definición de Realidad Virtual

Es complejo definir con precisión la combinación de palabras *realidad virtual*. En un ambiente cotidiano incluso podrían entenderse como antónimos. Sin embargo, en el contexto que se está estudiando en este trabajo coexisten y se les debe dar una interpretación. Anteriormente se dio una definición de la palabra *realidad*, así que ahora se pretende definir el adjetivo que la acompaña.

Algunos posibles significados de *virtual* son:

- Que tiene virtud para producir un efecto.
- Que no se concreta en la realidad aunque reúne las condiciones para ello.
- Que existe de manera aparente pero no es real.

Al juntar ambas definiciones, *realidad virtual* es el conjunto de objetos *aparentemente reales pero inexistentes* que están en el ambiente e interactúan con el individuo. Esta definición no es del todo adecuada porque es demasiado amplia y abstracta, pero gracias a que ya se expuso para qué sirve la realidad virtual y los componentes que la conforman, es posible desarrollar la primer definición y dar una más concreta: *realidad virtual* es (Sherman and Craig, 2002) un medio compuesto de simulaciones computacionales que miden la posición y acciones de un usuario y reemplazan o aumentan la información que llega a uno o más sentidos para lograr una sensación de presencia en la simulación (un mundo virtual).

Gracias a la segunda definición es posible identificar qué tipo de tecnología se necesitaría para desarrollar realidad virtual. En primer lugar, una computadora es necesaria porque así se generará el ambiente y se logrará la interacción con el usuario. Pero ahora surge la pregunta de cómo lograr medir las acciones del usuario y reemplazar o aumentar la información que llega a los órganos sensoriales. A continuación se estudiarán los dispositivos que cumplen este propósito.

1.1.3. Dispositivos para Realidad Virtual

Los dispositivos diseñados para lograr realimentación sensorial han ido evolucionando con el tiempo y cada vez son más sofisticados. Un ejemplo de un aparato diseñado para la vista son los lentes de realidad virtual como los que se muestran en la Figura 1.2. Su objetivo es crear imágenes estereoscópicas que abarquen todo el campo visual del usuario, lo cual además se complementa con el rastreo de la posición de su cabeza, de forma tal que las imágenes generadas sigan sus movimientos. Debido a que los humanos dependen fuertemente de la vista, se puede lograr un grado de inmersión muy alto.

Figura 1.2: Lentes de realidad virtual.

En cuanto al oído se pueden usar audífonos que combinado con técnicas de sonido tridimensional logran hacer creer al usuario que hay una fuente sonora proveniente de algún lugar específico de los alrededores. La mayoría de los dispositivos actuales se enfocan a la vista y audición, pero esas no son las únicas posibilidades.

Otro sentido que tiene un gran impacto en la forma en que un ser humano percibe su entorno y al cual se enfocará el estudio de este trabajo de aquí en adelante es el tacto. Se pueden percibir varias propiedades de un objeto a través de dicho sentido, como lo son: forma, peso, inercia, rigidez, temperatura, textura, posición y orientación (Mihelj and Podobnik, 2012). Es difícil tratar de recrear todas estas propiedades en un solo dispositivo, por lo que generalmente se enfoca la atención a ciertas sensaciones específicas. Es por ello que existe una gran variedad de artefactos, por ejemplo, un dispositivo para transmitir temperatura es muy diferente a uno usado para explorar la forma de un objeto. Se utiliza la palabra *háptica* para describir artefactos o aplicaciones que estimulan al tacto.

Una forma muy natural de explorar un objeto es sujetándolo y manipulándolo con las manos, lo cual permite conocer varias de sus propiedades. Un ejemplo de un dispositivo háptico que cumple este propósito es el guante mostrado en la Figura 1.3. Su función es medir la posición de la mano del usuario y transmitir los movimientos a un ambiente virtual y dependiendo de la interacción que se dé con los objetos que haya, se realimentará fuerza para que la persona los pueda sentir. Con él se puede explorar la forma, rigidez, posición y orientación de un objeto virtual.

Figura 1.3: Guante háptico.

Otro dispositivo que cumple una función muy similar es el que se muestra en la Figura 1.4. Usa un arreglo de actuadores que mediante ultrasonido crean formas y texturas. La gran ventaja es que no es necesario que el usuario use o sujete algo, sino que simplemente extienda el brazo y toque el aire, pudiendo recrear las mismas propiedades que en el guante háptico.

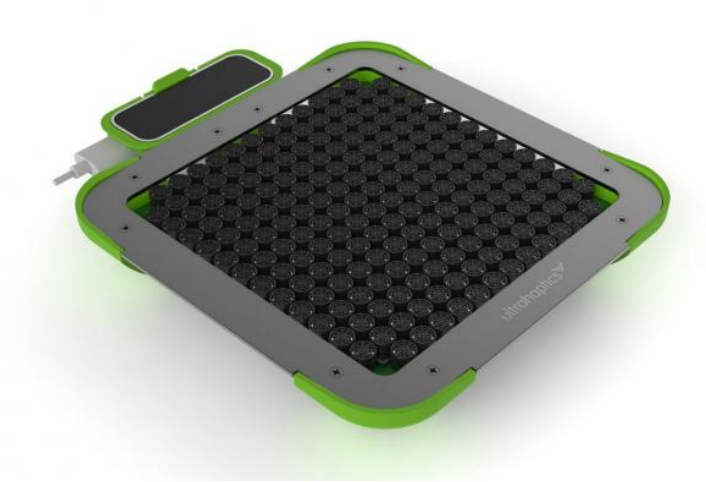


Figura 1.4: Cama de aire.

Se pueden desarrollar aparatos para una aplicación muy específica, como es el caso del volante mostrado en la Figura 1.5. Está pensado para simular el manejo de un automóvil, e incluye un sistema de realimentación para imitar la respuesta que tendría el volante al manejar por diferentes tipos de terrenos a distintas velocidades.

Figura 1.5: Volante y pedales.

Finalmente, también es posible usar robots manipuladores para aplicaciones hápticas. Un ejemplo es el robot Novint Falcon que aparece en la Figura 1.6. El usuario sujeta la esfera al final del mecanismo y lo manipula libremente. En respuesta, el robot puede aplicar fuerzas de reacción para lograr interacción con objetos virtuales.

Figura 1.6: Robot Falcon de la empresa Novint.

Una de las características más importantes que se le piden a un robot diseñado para aplicaciones hápticas es que la resistencia que oponga sea muy baja para que los movimientos del brazo y muñeca que realice el usuario sean naturales y se realicen mediante el contacto de la mano del usuario con la herramienta que tiene el robot al final del mecanismo llamada *efector final*.

El robot Falcon cuenta con tres *grados de libertad*. De forma intuitiva, se puede entender este término como los posibles movimientos que puede realizar un mecanismo. Al

Figura 1.7: Robot Geomagic Touch.

tener tres grados de libertad, el robot Falcon es capaz moverse en todas las direcciones espaciales: ancho, largo y profundidad, siendo el *espacio de trabajo* el conjunto de todos los puntos que puede alcanzar el efector final del robot.

El robot Novint Falcon no es el único robot háptico existente, en realidad hay gran variedad de características y fabricantes. Otro dispositivo muy usado es el Geomagic Touch de 3D Systems, que se muestra en la Figura 1.7, el cual cuenta con seis grados de libertad.

Anteriormente se ha dicho que con tres grados de libertad el robot Falcon puede alcanzar cualquier punto en el espacio tridimensional. Entonces, ¿para qué sirve tener seis como en el caso del Touch? La diferencia radica en que, aunque el robot paralelo puede mover su efector final a cualquier punto, no puede girarlo u orientarlo. Con los tres grados de libertad adicionales, el robot Touch puede ubicar la pluma y además girarla a cualquier posición, lo que le permite al usuario realizar un rango de movimientos más amplio al darle mayor destreza al mecanismo, entendiéndose este término como la capacidad de cambiar la posición y orientación de un objeto desde una configuración a otra escogida arbitrariamente (Bicchi, 2000).

El robot Novint Falcon fue lanzado como un producto de costo accesible, pensado para su uso en videojuegos por usuarios domésticos. La serie Touch de 3D Systems está pensada para el ámbito profesional y su arquitectura es serial. La serie Omega de la empresa Force Dimension tiene estructura paralela, igual que el Falcon, y también están orientados a ámbitos profesionales. Es por ello que el robot Falcon resulta muy atractivo: tiene características similares a los mecanismos de alto desempeño y la diferencia de precio

Dispositivo	Fabricante	Grados de libertad	Fuerza máxima [N]	Resolución [mm]	Precio [USD]
Falcon	Novint	3	9	0.063	250
Touch	3D Systems	6	3.3	0.055	2,800
Touch X	3D Systems	6	7.9	0.023	4,400
Omega 3	Force Dimension	3	12	<0.01	21,800
Omega 6	Force Dimension	6	12	<0.01	28,000

Tabla 1.1: Resumen de dispositivos hápticos comerciales.

con ellos es muy grande como se puede observar en la Tabla 1.1. Con esto en mente, se busca desarrollar aplicaciones recortando significativamente el costo y manteniendo la alta calidad.

1.1.4. Control de fuerza

Para que una persona pueda reconocer propiedades relacionadas con el tacto (forma, peso, inercia, etc.) se requiere de una exploración física del objeto. Si se desea simular estos atributos en un sistema virtual, será necesario crear reacciones que se opongan a las fuerzas que el usuario imprima con sus manos o brazos, es decir, se requiere controlar las fuerzas que aparezcan en la interacción del usuario con un dispositivo háptico.

Existen principalmente dos formas de hacerlo: directamente e indirectamente (Siciliano et al., 2009). La primera requiere de un sensor que haga mediciones y las compare contra una referencia, mientras que la segunda se basa en regularla a través de un controlador de posición. Estas técnicas son esenciales, ya que en ausencia de control de fuerza, podría llegar a suceder alguna de las siguientes situaciones:

- Que se dañe al dispositivo háptico en alguno de sus componentes mecánicos o electrónicos.
- Que se lesione al usuario al imprimirse una fuerza repentina o demasiado fuerte.
- Que las propiedades que se intentan recrear sean simuladas deficientemente, perdiéndose la inmersión que se haya logrado.

Evidentemente, nada de esto es deseable en un sistema de realidad virtual, de ahí la importancia de su estudio.

1.2. Planteamiento del problema

El desarrollo de aplicaciones de realidad virtual que incluyan háptica se ha visto limitado debido al alto costo de los dispositivos que permiten realizarlas. Esta tesis pretende atacar el problema mediante el uso y modificación de un robot de bajo costo.

La primera etapa del trabajo consistió en adaptar un sensor de fuerza a un robot Touch. Los objetivos de trabajar con este dispositivo se enlistan a continuación:

- Conocer las bases teóricas del control de fuerza.
- Implementar un controlador de fuerza con el método directo.

En la segunda etapa se trabajó con el robot Novint Falcon, el cual cuenta con tres grados de libertad en su estado actual. Esto le permite alcanzar cualquier punto contenido en su espacio de trabajo, pero no es posible cambiar la orientación del efector final. Ya se ha mencionado que el tener al menos seis grados de libertad permite al usuario interactuar de forma mucho más natural con el robot. Considerando lo anterior, los objetivos de la segunda etapa son:

- Dotar al robot con tres grados de libertad adicionales, los cuales serán actuados, mediante la adaptación de una muñeca esférica.
- Realizar la instrumentación requerida para poder controlar el nuevo mecanismo.
- Implementar un controlador de posición.
- Crear una aplicación háptica donde se muestre el funcionamiento mejorado del robot.

Con estas modificaciones el robot tendrá mayor destreza y esto ayudará a mejorar la inmersión que se logre en el ambiente virtual. Además, se obtendrá un prototipo de seis grados de libertad actuados con un costo mucho menor al de los robots disponibles actualmente.

1.3. Contribuciones

Las contribuciones de esta tesis son:

- Habilitar al robot Geomagic Touch para implementar controladores de fuerza.
- Implementación de un mecanismo de bajo costo, haciendo así al robot Falcon un dispositivo de seis grados de libertad.
- Diseño de la electrónica y programación necesarias para controlar dicho mecanismo.
- Creación de una aplicación de realidad virtual que sirva como plataforma de aprendizaje en el uso del robot Falcon y como base para en el futuro desarrollar aplicaciones más sofisticadas.

1.4. Organización de la tesis

En el Capítulo 2 se revisan las bases matemáticas necesarias para realizar el análisis y control de los robots utilizados. En el Capítulo 3 se estudia a fondo las técnicas de control de fuerza y se describe el trabajo realizado con el robot Geomagic Touch. En el Capítulo 4 se reporta el diseño mecánico y electrónico de una muñeca esférica para el robot Novint Falcon. En el Capítulo 5 se presentan conclusiones y el trabajo que queda por realizar.

Capítulo 2

Preliminares

En este capítulo se establecerán las bases matemáticas y físicas necesarias para poder analizar y controlar robots utilizados en este trabajo. Primero se introducen las nociones básicas de los mecanismos para luego aplicarlas al caso específico de robótica. Después se estudian las técnicas para analizarlos, incluyendo cinemática directa, inversa y diferencial, para finalmente llegar al modelo dinámico. Por último se estudian algunas técnicas de control.

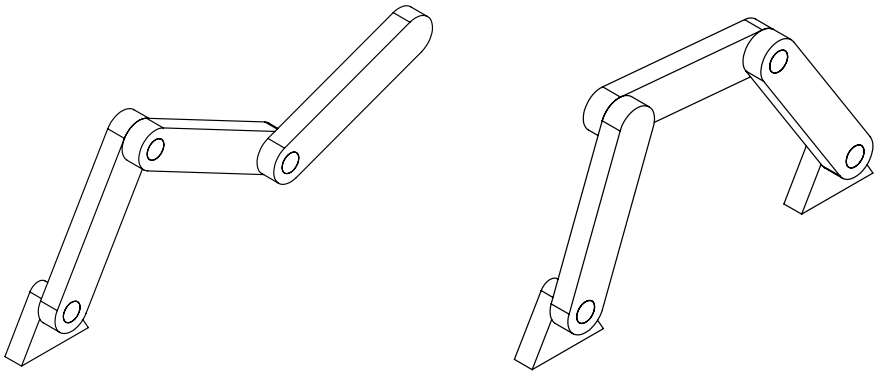
2.1. Mecanismos

Un *mecanismo* (Norton, 2016) es una cadena cinemática en la cual al menos un eslabón está fijo a tierra. A continuación se enlistan las definiciones para entender este concepto y que serán necesarias para el estudio de los robots.

- Un *eslabón* es un cuerpo que posee al menos dos nodos (puntos de unión con otros cuerpos). La mayoría de los robots actuales están construidos con eslabones rígidos, lo cual permite realizar análisis geométricos con los parámetros fijos. Si los eslabones son flexibles, la dificultad para estudiar matemáticamente al robot se incrementa.

- Una *articulación* (también llamada junta o par cinemático) es la unión de dos o más eslabones en sus nodos. Existen varios tipos de juntas, y una forma de clasificarlas es por los movimientos que permiten realizar a los cuerpos que unen. Las más importantes para robótica son la prismática y la de revolución. La primera únicamente permite la rotación de ambos cuerpos alrededor de un eje fijo, mientras que la segunda sólo permite traslación. Pares cinemáticos más complejos son equivalentes a una sucesión de varias juntas de revolución y prismáticas.

- Una *cadena cinemática* se obtiene al unir un conjunto de eslabones con juntas.



(a) Cadena cinemática abierta (b) Cadena cinemática cerrada

Figura 2.1: Tipos de cadenas cinemáticas.

- Los *grados de libertad* son el número de parámetros independientes necesarios para definir la posición y orientación de un cuerpo de forma única en cualquier instante de tiempo. Dependiendo del número de eslabones y el tipo de juntas con las que se unan, un mecanismo tendrá mayor o menor número de grados de libertad, este hecho afecta directamente el tipo de tareas que podrá realizar.

El objetivo de estudiar mecanismos es crear máquinas que puedan realizar una tarea de forma controlada y precisa.

2.2. Robots hápticos

Ahora que se han planteado las nociones básicas, en esta sección se revisan algunas definiciones relativas a robots hápticos que se usarán durante el desarrollo de la tesis.

2.2.1. Arquitectura

Existen dos tipos de arquitecturas para un robot, las cuales se muestran en la Figura 2.1. El inciso (a) corresponde con una cadena cinemática abierta y el (b) con una cadena cinemática cerrada. Es posible distinguir una de otra con un ejercicio simple. Si partiendo de un punto cualquiera y recorriendo la secuencia de eslabones es posible regresar al lugar de donde se partió, entonces se trata de una cadena cerrada, de lo contrario será abierta. En la Tabla 2.1 (Pandilov and Dukovski, 2014) se resumen las características que distinguen a ambos tipos de arquitecturas.

Un robot serial tendrá un buen desempeño cuando alguno o varios de los siguientes puntos se cumpla:

- Se requiere un amplio espacio de trabajo.

Característica	Serial	Paralelo
Tamaño del espacio de trabajo	Grande	Chico
Cinemática directa	Sencilla	Complicada
Cinemática inversa	Complicada	Sencilla
Errores de posición	Acumulativos	Se promedian
Errores de fuerza	Se promedian	Acumulativos
Rigidez	Baja	Alta
Inercia	Alta	Baja
Velocidad/aceleración	Lenta	Rápida
Precisión	Baja	Alta
Calibración	Sencilla	Complicada

Tabla 2.1: Características de las arquitecturas paralela y serial.

- Los movimientos que realizará el robot no son muy rápidos o bruscos.
- No importa si el manipulador tiene inercia muy grande.
- La precisión de posición no es tan crítica.
- No habrán fuerzas muy grandes, o si las hay, se cuenta con el espacio suficiente para alojar un robot de mayor tamaño.

Por otro lado se podrá usar un robot paralelo cuando haya uno o más de los siguientes criterios:

- Tener un amplio espacio de trabajo no es tan relevante.
- Se requiere que el robot se mueva rápidamente.
- Se desea baja inercia del manipulador.
- Se requiere mayor precisión de posición.
- Existirán grandes fuerzas y se desea un mecanismo compacto.

Aunque los robots seriales sean usados más comúnmente, eso no quiere decir que la arquitectura paralela sea inferior, sino que dependerá del uso que se le dé el elegir cuál es la mejor solución.

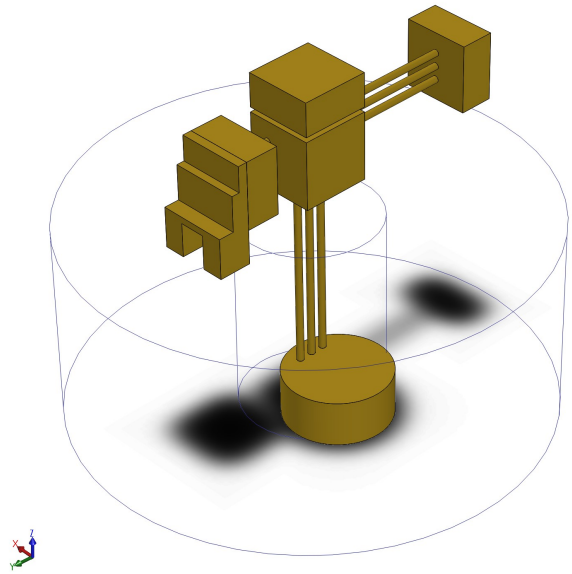


Figura 2.2: Espacio de trabajo del robot cilíndrico TP-801.

2.2.2. Espacio de trabajo

El *espacio de trabajo* de un robot es el volumen total formado por todos los puntos que puede alcanzar la herramienta que tenga el robot al final de la estructura (llamada *efector final*). La forma del espacio de trabajo se ve afectada por diversos factores, como puede ser: dimensión de los eslabones, tipo y número de juntas, límites mecánicos, etc.

A manera de ejemplo, en la Figura 2.2 se muestra en forma esquemática el espacio de trabajo del robot TP-801. Se trata de un mecanismo con articulaciones RPP (revolución-prismática-prismática) por lo que su espacio de trabajo adquiere una forma cilíndrica. Otras configuraciones comunes son RRR (revolución-revolución-revolución) y PPP (prismática-prismática-prismática) las cuales que generan un espacio de trabajo esférico y un prisma respectivamente.

2.2.3. Grados de libertad

Los grados de libertad de un robot junto con su estructura mecánica definirán el tipo de movimientos que puede realizar. Para que el mecanismo pueda alcanzar cualquier punto dentro de un espacio tridimensional, debe poseer al menos tres grados de libertad. Esto le permitirá posicionar su efector final en cualquier punto dentro de su espacio de trabajo, pero si se desea que además pueda lograr cualquier orientación, entonces se debe añadir tres más. Aunque es posible usar un robot de tres grados de libertad para

aplicaciones hápticas, lo cierto es que el tener seis mejora significativamente la interacción entre el usuario y el ambiente. En el primer caso los movimientos se limitan a los que se puedan realizar con el brazo, pero en el segundo es posible usar también la muñeca. El costo a pagar es un mecanismo y control más complejos, pero la inmersión será mayor. Algunos dispositivos como el Omega 7 de Force Dimension incluyen un séptimo grado de libertad. Aunque esto no añade destreza adicional al robot, en aplicaciones de háptica puede resultar útil ya que generalmente se usa para simular el uso de alguna herramienta, como por ejemplo unas tijeras.

2.3. Análisis de robots

El análisis de robots consta esencialmente de dos partes: cinemática y dinámica. La primera se encarga de estudiar el movimiento del robot basándose únicamente en su geometría. El estudio dinámico además toma en consideración los efectos de las fuerzas involucradas y las características físicas del manipulador como lo son inercia, fricción en las articulaciones, etc.

2.3.1. Cinemática directa

El objetivo de la *cinemática directa* es obtener la posición y orientación del efector final del robot con respecto al sistema de referencia base a partir de las variables articulares, representadas con un vector de la forma

$$\boldsymbol{q} = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{bmatrix} \tag{2.1}$$

donde n representa el número de articulaciones y q_i puede ser un ángulo o una distancia si se trata de una articulación de revolución o prismática respectivamente.

Para resolver el problema de forma sistemática, es posible utilizar la convención de Denavit-Hartenberg (Spong et al., 2006). Se trata de un algoritmo para seleccionar los sistemas de referencia en cada articulación de forma tal que se pueda obtener la posición y orientación del efector final respecto al sistema base por medio de multiplicar matrices, las cuales adquieren la forma de la ecuación siguiente:

$${}^{i-1}\boldsymbol{A}_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\alpha_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.2}$$

donde ${}^{i-1}\boldsymbol{A}_i$ denota la transformación de coordenadas del sistema de referencia i al sistema de referencia $i-1$. Los parámetros α_i, θ_i, a_i y d_i se definen en el algoritmo Denavit-Hartenberg (consultar (Spong et al., 2006)). Es importante notar que tres de los cuatro parámetros serán constantes dependiendo del tipo de articulación. Si se trata de una articulación de revolución, entonces θ_i será variable. Si se trata de una articulación prismática, entonces d_i será variable.

Al realizar la multiplicación

$$\boldsymbol{T} = {}^{0}\boldsymbol{A}_1 {}^{1}\boldsymbol{A}_2 \cdots {}^{\text{n-1}}\boldsymbol{A}_n \tag{2.3}$$

se obtiene la matriz $\boldsymbol{T}$ que representa la acumulación de transformaciones necesarias para obtener la posición y orientación del efector final del robot respecto a un sistema de referencia base.

2.3.2. Cinemática inversa

La cinemática inversa es la operación contraria a la directa, ya que su objetivo es hallar las variables articulares conociendo la orientación y posición del efector final. Matemáticamente, el problema se puede plantear como encontrar los valores de q_i que cumplan la igualdad

$$\boldsymbol{A}_1(q_1)\boldsymbol{A}_2(q_2)\ldots\boldsymbol{A}_n(q_n) = \boldsymbol{T}$$

donde:

- $\boldsymbol{T} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{d} \\ 0 & 1 \end{bmatrix}$ es un matriz de transformación homogénea que representa la posición y orientación del efector final deseadas.
- $\boldsymbol{R}$ es la matriz de rotación del efector final.
- $\boldsymbol{d}$ es el vector de posición del efector final.

Para este problema se puede usar un método analítico, pero se requiere resolver un sistema de ecuaciones no lineales. Es por ello que se prefiere usar la geometría del robot para relacionar las variables.

2.3.3. Cinemática diferencial

La cinemática diferencial se encarga de mapear las velocidades articulares con las velocidades en el espacio cartesiano. Esto se hace a través del *Jacobiano* del manipulador tal como se muestra en la ecuación:

$$\dot{\boldsymbol{x}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{2.4}$$

donde $\boldsymbol{J} \in \Re^{6\times n}$ es el Jacobiano y $\dot{\boldsymbol{x}} \in \Re^{n\times 1}$ es el vector de velocidad en el espacio cartesiano. El Jacobiano sirve para planear y ejecutar trayectorias suaves, determinar las singularidades del robot, obtener el modelo dinámico y para la transformación entre fuerza y par. Esto último se logra a través de la ecuación

$$\boldsymbol{\tau} = \boldsymbol{J}(\boldsymbol{q})^T \boldsymbol{F} \tag{2.5}$$

Jacobiano geométrico

El Jacobiano geométrico consta de dos partes: una superior y una inferior. La primera parte representa las velocidades lineales y tiene la forma

$$J_v = \begin{bmatrix} J_{v_1} & \cdots & J_{v_n} \end{bmatrix}$$

donde la i-ésima columna J_{v_i} está dada por

$$J_{v_i} = \begin{cases} z_{i-1} \times (o_n - o_{i-1}) & \text{para articulación de revolución} \\ z_{i-1} & \text{para articulación prismática} \end{cases}$$

z_{i-1} representa el eje de rotación de la articulación i-ésima y o_i es el vector de posición del sistema de referencia i-éismo. La mitad inferior del Jacobiano representa las velocidades angulares y tiene la forma

$$J_\omega = \begin{bmatrix} J_{\omega_1} & \cdots & J_{\omega_n} \end{bmatrix}$$

donde la i-ésima columna J_{ω_i} está dada por

$$J_{\omega_i} = \begin{cases} z_{i-1} & \text{para articulación de revolución} \\ 0 & \text{para articulación prismática} \end{cases}$$

Finalmente, al juntar ambas partes se obtiene el Jacobiano geométrico

$$\boldsymbol{J} = \begin{bmatrix} J_1 & J_2 & \cdots & J_n \end{bmatrix}$$

donde la i-ésima columna J_i está dada por

$$J_i = \begin{cases} \begin{bmatrix} z_{i-1} \times (o_n - o_{i-1}) \\ z_{i-1} \end{bmatrix} & \text{para articulación de revolución} \\ \begin{bmatrix} z_{i-1} \\ 0 \end{bmatrix} & \text{para articulación prismática} \end{cases} \tag{2.6}$$

Jacobiano analítico

Si la orientación del efector final se ha representado con un número mínimo de parámetros, entonces es posible calcular el Jacobiano al derivar la cinemática directa del robot respecto al tiempo.

$$\dot{\boldsymbol{x}} = \frac{\partial f_i}{\partial q_1}\dot{q}_1 + \frac{\partial f_i}{\partial q_2}\dot{q}_2 + \cdots + \frac{\partial f_i}{\partial q_n}\dot{q}_n \tag{2.7}$$

Expresada en forma matricial, la ecuación (2.7) se convierte en

$$\dot{\boldsymbol{x}} = \begin{bmatrix} {}^0\dot{d}_n \\ {}^0\dot{\omega}_n \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial q_1} & \frac{\partial f_1}{\partial q_2} & \cdots & \frac{\partial f_1}{\partial q_n} \\ \frac{\partial f_2}{\partial q_1} & \frac{\partial f_2}{\partial q_2} & \cdots & \frac{\partial f_2}{\partial q_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial q_1} & \frac{\partial f_m}{\partial q_2} & \cdots & \frac{\partial f_m}{\partial q_n} \end{bmatrix} \dot{\boldsymbol{q}} \tag{2.8}$$

donde ${}^0\dot{d}_n \in \Re^3$ es la velocidad lineal y ${}^0\dot{\omega}_n \in \Re^3$ es la velocidad angular.

2.3.4. Modelo dinámico

El modelo dinámico de un sistema físico es un conjunto de ecuaciones diferenciales que describe su comportamiento, las cuales se obtienen mediante las leyes físicas que rigen los elementos que conforman al sistema y la interacción que hay entre ellos.

Un sistema cuyo modelo es una ecuación diferencial de orden cero es un sistema *estático*, y se caracteriza por responder instantáneamente a cualquier estímulo, por otro lado, un sistema *dinámico* tiene como modelo ecuaciones de orden mayor o igual a uno, a diferencia de un sistema estático, sus variables evolucionan con el tiempo.

Si el modelo está formado por un conjunto de ecuaciones lineales, entonces se dice que el sistema es lineal. Un sistema es *no lineal* si al menos una de las ecuaciones es no lineal.

Si un sistema cuenta con una sola entrada y una sola salida, será llamado SISO (Single Input - Single Output), mientras que si cuenta con múltiples entradas y salidas, entonces será llamado MIMO (Multiple Input - Multiple Output).

Con todo lo anterior en mente, se puede clasificar a un robot como un sistema dinámico no lineal de múltiples entradas y salidas.

El modelo dinámico de un robot manipulador está dado por la ecuación

$$\boldsymbol{H}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{D}\,\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau} \tag{2.9}$$

donde:

- $\boldsymbol{H}(\boldsymbol{q}) \in \Re^{\mathrm{nxn}}$ es la matriz de inercia.
- $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} \in \Re^{\mathrm{n}}$ es el vector de fuerzas centrífugas y de Coriolis.

- $\boldsymbol{D} \in \Re^{\mathrm{nxn}}$ es la matriz de coeficientes de fricción viscosa en las articulaciones del robot.
- $\boldsymbol{g}(\boldsymbol{q})$ es el vector de fuerzas gravitacionales.
- $\boldsymbol{\tau} \in \Re^{\mathrm{n}}$ es el vector de pares de entrada en las articulaciones.

Este modelo tiene una propiedad llamada *linealidad respecto a los parámetros*. Esto quiere decir que existe una función $\boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) \in \Re^{n \times l}$ y un vector $\boldsymbol{\Theta} \in \Re^{l}$ tal que se cumpla la igualdad

$$\boldsymbol{H}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{D}\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\boldsymbol{\Theta} = \boldsymbol{\tau} \tag{2.10}$$

La función $\boldsymbol{Y}$ es llamada *regresor* y $\boldsymbol{\Theta}$ es el vector de parámetros. La dimensión de $\boldsymbol{\Theta}$, es decir, el número de parámetros necesarios para expresar el modelo en esta forma no es única.

2.3.5. Restricciones holonómicas

Cuando el manipulador entra en contacto con un objeto en su espacio de trabajo, se generan fuerzas de reacción que pueden afectar la forma en que se mueve, se dice entonces que el robot está en movimiento restringido. Es posible representar estas restricciones como un conjunto de ecuaciones algebraicas que deben satisfacer las variables de posición y orientación del efector final. Si es posible que estas ecuaciones adquieran la forma

$$\boldsymbol{\varphi}(\mathbf{q}) = 0 \tag{2.11}$$

entonces se dice que la restricción es *holonómica* (Siciliano et al., 2009). El vector $\boldsymbol{\varphi}$ tiene dimensión m, con $m < n$. La derivada de esta ecuación será

$$\boldsymbol{J}_{\varphi}(\boldsymbol{q})\dot{\boldsymbol{q}} = 0 \tag{2.12}$$

donde $\boldsymbol{J}_{\varphi}(\boldsymbol{q}) = \frac{\partial \varphi}{\partial q} \in \Re^{m \times 6}$ es llamado *Jacobiano de la restricción*. La ecuación (2.11) establece claramente que la restricción está en función de las variables articulares, pero en caso de que se plantee en coordenadas cartesianas, es posible relacionarlas por medio de la ecuación siguiente:

$$\boldsymbol{J}_{\varphi}(\boldsymbol{q}) = \boldsymbol{J}_{\varphi}(\boldsymbol{x})\boldsymbol{J}(\boldsymbol{q}) \tag{2.13}$$

donde $\boldsymbol{J}_{\varphi}(\boldsymbol{q})$ y $\boldsymbol{J}_{\varphi}(\boldsymbol{x})$ son los Jacobianos de la restricción en coordenadas articulares y cartesianas respectivamente, y $\boldsymbol{J}(\boldsymbol{q})$ es el Jacobiano del manipulador.

Cuando el robot se encuentra en movimiento restringido, el modelo matemático se ve afectado y queda como la ecuación:

$$\boldsymbol{H}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{D}\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau} + \boldsymbol{J}_{\varphi}(\boldsymbol{q})^{T}\boldsymbol{\lambda} \tag{2.14}$$

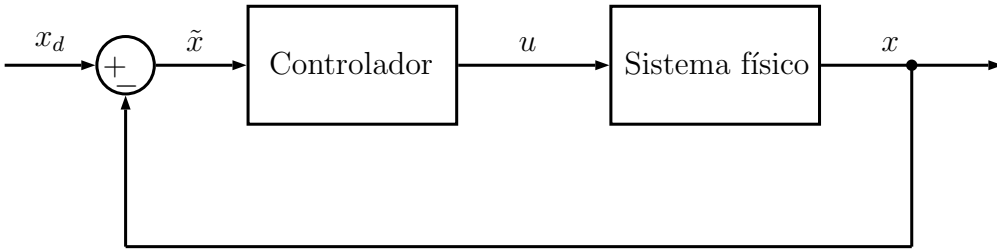


Figura 2.3: Esquema de control clásico.

donde $\boldsymbol{\lambda}$ es un multiplicador de Lagrange que representa la fuerza aplicada en el punto de contacto con la superficie y que puede deberse a objetos reales o, en el caso de los dispositivos hápticos, a la interacción del robot con el ambiente virtual.

2.4. Control de robots

Ahora que se conocen las herramientas matemáticas para análisis, es posible dar un paso más: hacer que los robots se comporten conforme se requiera. Este proceso involucra el medir variables, compararlas contra lo que se desee y mover el mecanismo para compensar los errores que existan a lo largo del tiempo. En la Figura 2.3 se presenta un esquema clásico de control aplicable a cualquier sistema físico en donde:

- x_d es la variable de referencia, es decir, lo que se desea que haga el sistema físico y puede ser constante o variable con el tiempo.
- x representa el valor de la(s) salida(s) del sistema físico.
- $\tilde{x}$ es la señal de error, y resulta de restar el valor real de las variables del sistema a la referencia.
- u representa la entrada que se aplicará al sistema para compensar el error.

Aplicando este esquema al caso específico de un robot, es posible identificar qué representa físicamente cada variable.

- Las salidas $\boldsymbol{x}$ usualmente son las variables articulares $\boldsymbol{q}$, las cuales se miden con sensores integrados en cada articulación. Adicionalmente, puede ser de interés medir las fuerzas existentes en el efector final debido a la interacción ambiente-robot.
- Las entradas $\boldsymbol{u}$ dependen del tipo de actuador que tenga el robot. Lo más común es que tengan motores eléctricos de corriente directa, por lo que las entradas serán los pares aplicados en cada articulación. Si se añade un bloque al diagrama que involucre la dinámica del actuador, entonces las entradas serán los voltajes aplicados a cada motor.

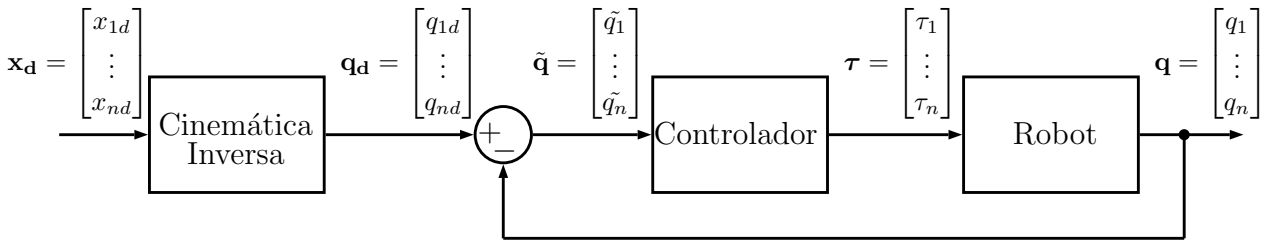


Figura 2.4: Esquema de control de posición básico para un robot.

- La referencia $\boldsymbol{x_d}$ es un valor numérico que representa par o voltaje dependiendo del tipo de entrada.

Si $\boldsymbol{x_d}$ permanece constante, entonces el problema de control adquiere el nombre de *regulación*. Si $\boldsymbol{x_d}$ varía con el tiempo, entonces el problema es de *seguimiento*. Ambos tipos son importantes para un robot. Por ejemplo, si se desea que el manipulador ejerza una fuerza constante en una superficie plana, se está lidiando con un problema de regulación; por otro lado, si se deseara que el robot se mueva a lo largo de una trayectoria específica en su espacio de trabajo, entonces el problema es de seguimiento.

2.4.1. Controladores

Un tipo de control básico implica hacer que el robot se mueva de acuerdo con una referencia. Si esta es constante, se le está pidiendo al mecanismo que permanezca en una posición fija, lo cual podría ser útil en casos aislados, pero en general se desea que el manipulador se mueva de un lado a otro constantemente. Esto significa que habrá que crear una o varias trayectorias que deberá seguir el robot. Estas suelen diseñarse en el espacio cartesiano, pero como el robot trabaja en el espacio articular, es necesario realizar la transformación entre ambos espacios mediante la cinemática inversa. En la Figura 2.4 se muestra un diagrama de bloques más completo donde además se ha indicado explícitamente la forma de las señales en cada etapa.

La principal tarea consiste en diseñar el bloque del controlador. Un buen diseño deberá ser capaz de:

- Hacer que el robot siga a la referencia.
- Combatir perturbaciones que provengan del exterior.
- Compensar dinámicas que no se hayan considerado (por ejemplo, fricción seca en articulaciones).
- Cumplir con cualquier otro criterio que se le pida, como puede ser: ahorro energético, respuesta rápida, etc.

Control PID

Uno de los controladores más sencillos usa el error para hacer que el robot se mueva de acuerdo a la referencia. El acrónimo PID significa Proporcional-Integral-Derivativo, lo que quiere decir que se aplicarán acciones de control con dichos operadores matemáticos.

Un control proporcional puro está dado por la ecuación 2.15

$$\boldsymbol{\tau} = \boldsymbol{K_p}\tilde{\boldsymbol{q}} \tag{2.15}$$

donde $\boldsymbol{K_p} \in \Re^{n\times n}$ es una matriz diagonal positiva definida de ganancias proporcionales. Este controlador hará que la acción de control sea mayor entre más grande sea el error, pero no logra un seguimiento perfecto de la trayectoria: siempre habrá un error.

Un controlador Proporcional-Derivativo añade un término diferencial:

$$\boldsymbol{\tau} = \boldsymbol{K_p}\tilde{\boldsymbol{q}} + \boldsymbol{K_d}\dot{\tilde{\boldsymbol{q}}} \tag{2.16}$$

donde $\boldsymbol{K_d} \in \Re^{n\times n}$ es una matriz diagonal positiva semidefinida de ganancias derivativas. Esta ley de control además de actuar en forma proporcional al error, intenta predecir su comportamiento y compensarlo. Tampoco logra hacer que el error llegue a cero.

Para que un controlador logre hacer nulo al error en estado permanente, es necesario añadirle un término integral. Al juntar las tres acciones, se obtiene la ley de control de la ecuación (2.17).

$$\boldsymbol{\tau} = \boldsymbol{K_p}\tilde{\boldsymbol{q}} + \boldsymbol{K_i}\int_0^t \tilde{\boldsymbol{q}}\, dt + \boldsymbol{K_d}\dot{\tilde{\boldsymbol{q}}} \tag{2.17}$$

donde $\boldsymbol{K_i} \in \Re^{n\times n}$ es una matriz diagonal positiva semidefinida de ganancias integrales. El sumar la integral del error garantiza que este tenderá a cero de forma asintótica, es decir:

$$t \to \infty \implies \tilde{\boldsymbol{q}} \to 0$$

También es posible tener un controlador PI, pero nunca se usará un control integral o derivativo puro, ni tampoco uno I-D debido a problemas de estabilidad.

Las principales ventajas de un control PID son:

- No requiere del modelo matemático del robot para su implementación, sino que se basa en el comportamiento del error. Sin embargo, es importante recalcar que si se conoce el modelo del sistema, es posible lograr una mejor sintonización del controlador.
- Las operaciones son sencillas y no demandan muchos recursos computacionales.
- Con una buena sintonización, el desempeño también lo es.

Esto ha hecho que este tipo de controlador sea usado ampliamente en una gran gama de situaciones. En particular, es posible aplicarlo a robótica industrial y logrará un desempeño adecuado.

Controladores no lineales

Los controladores que se basan en funciones no lineales pueden tener aún más ventajas que un controlador lineal como el PID. Algunos intentan eliminar las características dinámicas del robot, mientras que otros pretenden aprovecharlas. Incluso hay controladores que intentan hacer que el manipulador se comporte como un sistema lineal.

Un ejemplo es la ley de control que se muestra a continuación:

$$\boldsymbol{\tau} = \boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\boldsymbol{\Theta} + \boldsymbol{u} \tag{2.18}$$

El primer sumando pretende anular la dinámica del robot, ya que la multiplicación del regresor por el vector de parámetros es equivalente al modelo matemático. El principal problema radica en obtener una buena aproximación de los parámetros, ya que algunos de ellos suelen ser difíciles de calcular o medir y siempre existirá incertidumbre. El controlador se encarga de anular los efectos de la fricción viscosa, inercias, fuerzas centrífugas y gravedad, de forma tal que sólo resta diseñar $\boldsymbol{u}$, la cual puede ser cualquier ley de control.

El sistema en lazo cerrado será:

$$\boldsymbol{H}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{D}\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\boldsymbol{\Theta} + \boldsymbol{u}$$

Este controlador se puede ver como dos lazos que en conjunto actúan como uno solo. El término u representa una ley de control cualquiera, como podría ser un PID, y el resto se puede ver como un lazo de linealización.

Es importante mencionar que también hay controladores lineales que pueden hacer uso del modelo matemático del sistema, y por ende también requerirán de los parámetros del robot, por lo que ese problema no es exclusivo a controladores no lineales. A modo de resumen, un controlador no lineal tiene las siguientes ventajas y desventajas:

- Puede anular las características dinámicas de un robot.
- Puede hacer que un sistema no lineal se comporte como lineal.
- Es capaz de lograr un desempeño mejor al de un controlador lineal.
- Las operaciones necesarias demandan más recursos computacionales.
- Suelen requerir un conocimiento profundo del sistema.

2.4.2. Sintonización del controlador

Uno de los aspectos de mayor importancia al diseñar un controlador PID es el proceso de sintonización, es decir, encontrar el valor de las ganancias que se usarán en las matrices $\boldsymbol{K_p}, \boldsymbol{K_d}, \boldsymbol{K_i}$, ya que estas afectan directamente el comportamiento del sistema en lazo cerrado. En general existen dos métodos para realizar la sintonización: experimentales y

basados en el modelo matemático del sistema físico. Los métodos experimentales son adecuados cuando no se tiene el modelo dinámico del robot. Los métodos basados en modelo suelen dar los mejores resultados, ya que al tenerse conocimiento de la dinámica del manipulador, es posible desarrollar la teoría que garantice estabilidad, realizar simulaciones y hacer ajustes hasta encontrar el comportamiento deseado. Un ejemplo de estos últimos es el que se describe en (Kelly et al., 2006). El procedimiento está diseñado específicamente para un robot manipulador y permite calcular las ganancias de un controlador PID de posición que garanticen estabilidad asintótica de forma local teniendo en cuenta las siguientes consideraciones:

- El robot cuenta únicamente con articulaciones de revolución.
- La posición deseada es constante.
- Las condiciones iniciales del robot son lo suficientemente cercanas a la referencia.

Los cálculos a realizar son:

1. $\lambda_{max}(\boldsymbol{K_i}) \geq \lambda_{min}(\boldsymbol{K_i}) > 0$
2. $\lambda_{max}(\boldsymbol{K_p}) \geq \lambda_{min}(\boldsymbol{K_p}) > k_g$
3. $\lambda_{max}(\boldsymbol{K_d}) \geq \lambda_{min}(\boldsymbol{K_d}) > \frac{\lambda_{max}(\boldsymbol{K_i})}{\lambda_{min}(\boldsymbol{K_p})-k_g} \frac{\lambda^2_{max}(\boldsymbol{H})}{\lambda_{min}(\boldsymbol{H})}$

donde:

- $k_g = n(Max_{i,j,q}|\frac{\partial g_i(q)}{\partial q_j}|)$
- $\boldsymbol{g}$ es el vector de gravedad del robot.
- $\boldsymbol{H}$ es la matriz de inercia del robot.

Para poder usar este método de sintonización, es necesario conocer al menos parcialmente el modelo dinámico del robot: como mínimo se deben conocer la matriz de inercia y el vector de gravedad.

Capítulo 3

Control de fuerza en superficies virtuales

Un robot está en contacto frecuente con objetos dentro de su espacio de trabajo que producen de forma natural fuerzas de reacción que hacen que el manipulador entre en movimiento restringido. En contraste, para las aplicaciones que usan robots hápticos para interactuar con ambientes virtuales es necesario simular dicho contacto para hacer creer al usuario que está tocando objetos reales. En ambos casos el controlar la fuerza existente entre el efector final del robot y el objeto es de gran importancia. Si el objeto es real y el manipulador imprime una fuerza grande, podría llegar a dañarse dicho objeto o algún motor o eslabón del robot. En el caso de un objeto virtual, un mal control de fuerza podría representar erróneamente sus propiedades y mermar la calidad de la simulación. Es por ello que además del controlador de posición que usualmente acompaña a un robot es necesario adicionar control de fuerza. Existen diferentes técnicas para realizar tal tarea, pero el principal factor que las distingue es si usan un sensor para medir fuerza, o bien, si la estiman y controlan indirectamente. A continuación se estudiarán ambos métodos.

3.1. Control indirecto

Un primer enfoque para el control de fuerza es hacerlo indirectamente a través de un controlador de posición. Este acercamiento permite ahorrar costo en la instrumentación del robot al no requerir un sensor de fuerza. Pero este hecho también puede interpretarse como una desventaja, ya que no hay forma de asegurar que la fuerza real corresponde con la que se establece en el controlador. Para aplicaciones donde no es crítico mantener una referencia de fuerza específica (por ejemplo, para mover piezas pequeñas de un lado a otro), un control indirecto puede ser suficiente.

Un ejemplo de un controlador de esta categoría es el control por impedancias que se presenta en el diagrama de bloques de la Figura 3.1 (Mihelj and Podobnik, 2012) en donde:

- $\boldsymbol{x}$ es el vector de posición del efector final del robot.
- $\boldsymbol{F}$ es el vector de fuerzas del efector final del robot.
- $\boldsymbol{\tau}$ es el vector de par de cada articulación del robot.
- $\boldsymbol{q}$ es el vector de variables articulares.
- $\boldsymbol{F_d}$ es el vector de fuerzas deseadas.
- $\boldsymbol{\tau_d}$ es el vector de par deseado.
- $\boldsymbol{J}^T$ es el Jacobiano transpuesto del manipulador.
- El bloque de modelo de impedancias representa el modelo matemático del entorno que incluye términos de fuerza elástica, viscosa e incercial.

Se recuperan las variables articulares $\boldsymbol{q}$ a través de los encoders, para después transformarlas a coordenadas cartesianas mediante la cinemática directa. Usando el modelo del entorno virtual y la posición del efector final, se calculan las fuerzas de reacción deseadas $\boldsymbol{F_d}$. Posteriormente estas se transforman a los pares deseados $\boldsymbol{\tau_e}$ mediante el Jacobiano transpuesto con la expresión descrita en la ecuación (2.5) del Capítulo 2.

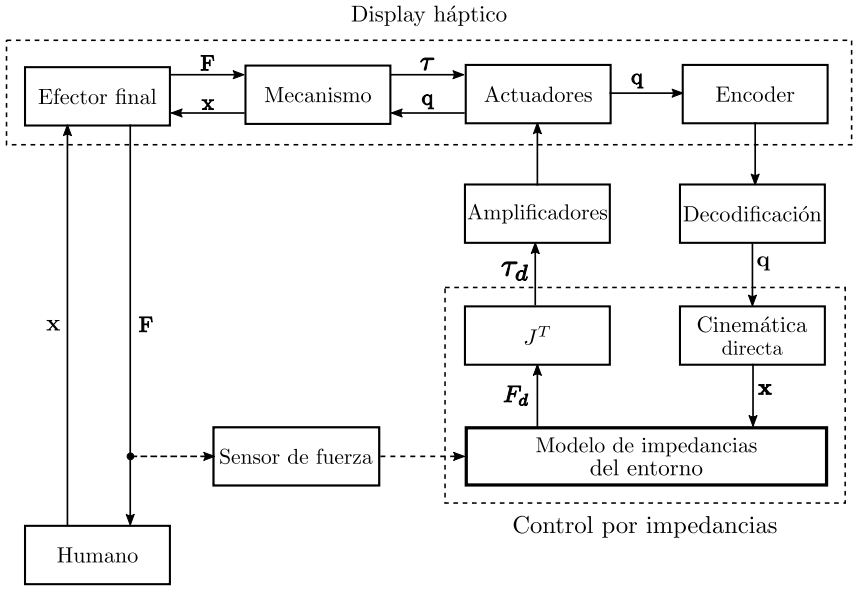


Figura 3.1: Control por impedancias (Mihelj and Podobnik, 2012). Las flechas punteadas representan información complementaria.

3.2. Control directo

Una segunda forma de hacer control de fuerza es mediante la adaptación de un sensor de fuerza en el efector final del robot y realimentándola en un lazo. De esta forma se puede lograr un controlador preciso, ya que se puede comparar una señal de referencia con las fuerzas que se miden directamente en el manipulador sin tener que recurrir a estimaciones, asegurando que el error de seguimiento se encuentre dentro de los límites que se hayan establecido. Un esquema que implementa esta idea es el que se muestra en la Figura 3.2. Consta de dos acciones de control: una para compensar el error de posición y otra para el de fuerza. Con la acción conjunta de ambas señales se puede lograr que el robot siga una trayectoria y una referencia de fuerza una vez que entra en contacto con objetos dentro de su espacio de trabajo, siempre y cuando ambos objetivos sean alcanzables simultáneamente.

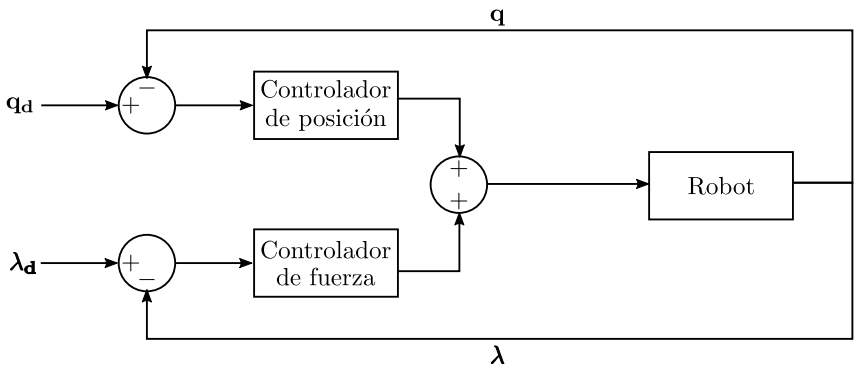


Figura 3.2: Esquema de control de fuerza y posición.

Realizar este tipo de control suele ser más costoso debido a la inclusión del sensor, pero habrá aplicaciones en las que sea crítico conocer el error de fuerza para evitar daños al objeto manipulado o al robot mismo en el proceso que se esté llevando a cabo. Si tal es el caso, un control de fuerza directo es indispensable.

3.3. Medición de fuerza

Para poder hacer control de fuerza directo, primero es necesario definir qué es lo que se desea medir. En robótica existe una gran variedad de fuerzas en todo momento: fuerzas de reacción en el efector final y en los eslabones, pares de fuerzas en cada motor, peso del robot y los objetos que manipula, etc. Pero la interacción de un manipulador con el ambiente siempre se dará a través del efector final, lo cual también es cierto en aplicaciones hápticas ya que el usuario sujetará al mecanismo en la herramienta que tenga al final de la cadena cinemática y ahí se renderizarán las fuerzas de interacción con los objetos virtuales,

Figura 3.3: Celda de carga modelo LLB300 marca Futek.

es por ello que el sensor se colocará ahí. Existen principalmente tres tipos de sensores de fuerza usados en robótica (Spong et al., 2006):

- Un sensor de fuerza en la muñeca que permita realizar mediciones a lo largo de tres ejes de referencia.
- Galgas extensiométricas colocadas en los ejes de cada motor.
- Sensores táctiles, generalmente usados para medir fuerza de agarre en la pinza del robot.

El primer tipo de sensor es el más adecuado para aplicaciones hápticas porque permite conocer las componentes de la fuerza existente en el efector final. Idealmente se desea medir a lo largo de los tres ejes, pero dado que en el Laboratorio de Robótica se cuenta por el momento con una celda de carga como la que se muestra en la Figura 3.3, los experimentos se realizaron mediante ella. Este sensor únicamente funciona en la dirección perpendicular a él, por lo que no sería posible conocer las dos componentes restantes.

La celda genera un voltaje proporcional a la carga aplicada en el botón. Esta señal es lineal y se sabe que en la carga máxima de 10 libras la salida es de 2.6466 [mV/V]. Se desea medir la fuerza en Newtons y no en libras, por lo que se calcula la conversión dada por la ecuación

$$F = (LS)\left(\frac{10}{2.6466}\right)\left(\frac{44.482216}{10}\right)[\text{N}] \tag{3.1}$$

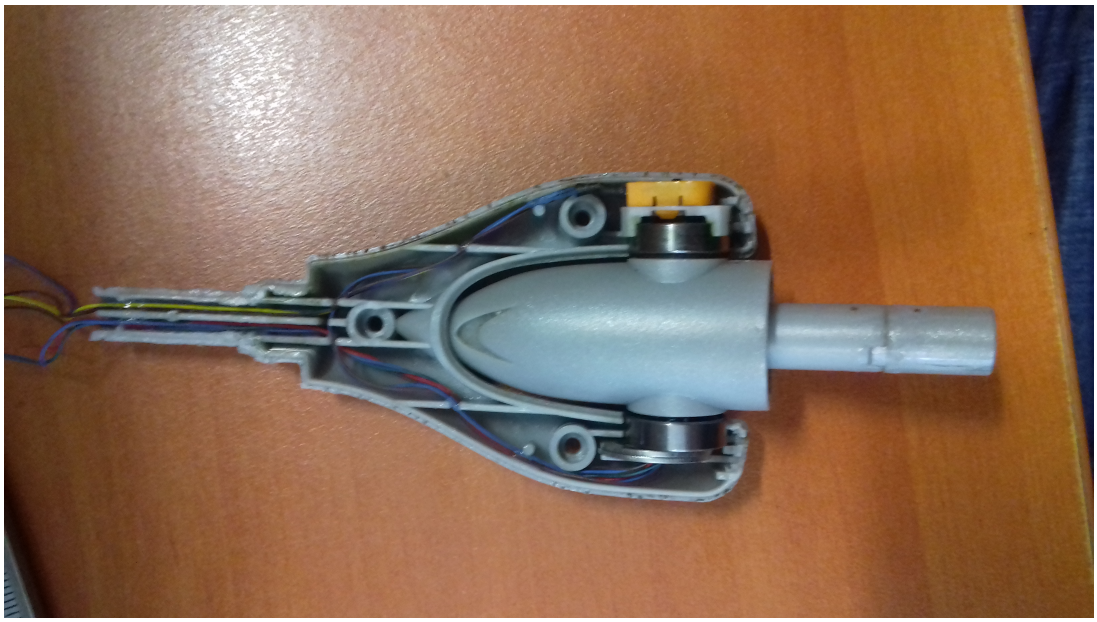


Figura 3.4: Muñeca esférica del robot Geomagic Touch.

donde LS es la lectura entregada por el sensor, el segundo factor es la pendiente de la curva característica del sensor y el último factor es la conversión de libras a newtons.

3.3.1. Adaptación del sensor de fuerza

En la fotografía del robot Touch de la Figura 1.7 del Capítulo 1 se puede apreciar que el efector final que el fabricante diseñó tiene forma de bolígrafo. Su geometría hace muy complicado el poder adaptar la celda de carga de forma que resulte cómodo para el usuario sujetar al robot a la vez que se puedan realizar las lecturas. Es por ello que fue necesario rediseñar algunas piezas de la muñeca del robot para acomodar al sensor. Para hacerlo se desmontó dicho mecanismo, mostrándose en la Figura 3.4 su construcción interna.

En la Figura 3.5 se puede ver la posición a la que se debe llevar al robot para realizar su calibración. Como se puede apreciar, la segunda pieza de la muñeca es esencial porque en la base hay un receptáculo donde debe entrar para poder calibrar al dispositivo, es por ello que fue necesario recrear dicha pieza mediante CAD.

Una vez que se había recreado la punta del bolígrafo, en el otro extremo se podía diseñar la pieza libremente para poder adaptar el sensor de fuerza. El objetivo era diseñarla de tal forma que permitiera al usuario manipular al robot fácilmente y al mismo tiempo tocar con su dedo índice el botón de la celda de carga para poder tomar lecturas. Se intentaron varias geometrías distintas, pero la mayoría falló debido a que no sujetaban firmemente al dedo o eran frágiles. Finalmente se optó por diseñar una pieza que sujeta al

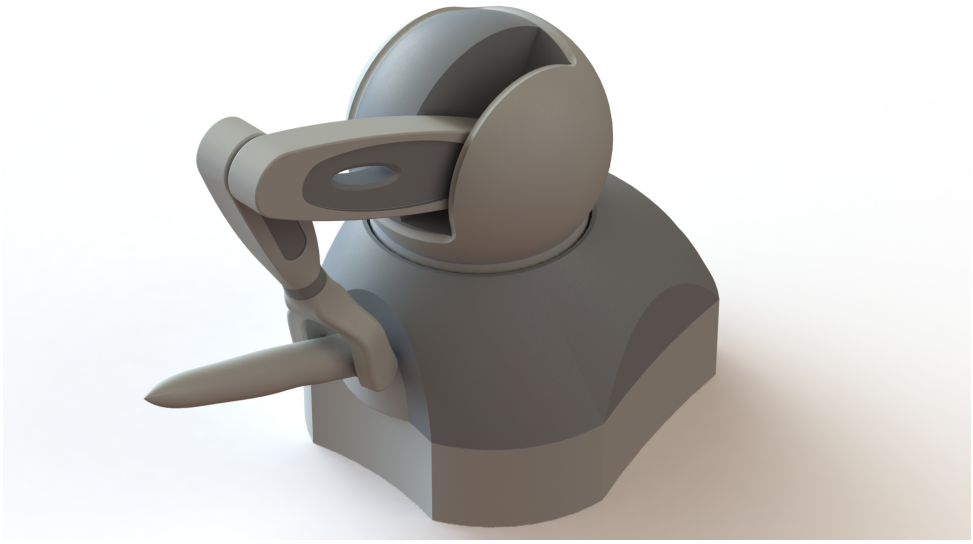


Figura 3.5: Posición para calibración.

dedo mediante una cinta ajustable, logrando así una buena sujeción del robot y contacto con la celda de carga. En la Figura 3.6 se muestran algunas vistas de la pieza diseñada. La celda de carga se inserta en el hueco redondo que se ve en la vista inferior. El robot con la nueva pieza instalada se muestra en la Figura 3.7.

La ventaja de la cinta ajustable es que cualquier usuario podrá usar el robot fácilmente sin importar el tamaño de su dedo. La calibración funciona correctamente, lo que significa que se recreó la punta del bolígrafo de forma adecuada. La manufactura se hizo con una impresora 3D con filamento ABS de 1.75[mm] de diámetro, lo cual implica un costo de producción muy bajo. Basándose en el diseño de esta pieza, si en un futuro se desea usar algún sensor distinto al que se presenta en este trabajo, las modificaciones a realizar al dibujo serán muy sencillas y el costo de manufactura volverá a ser bajo.

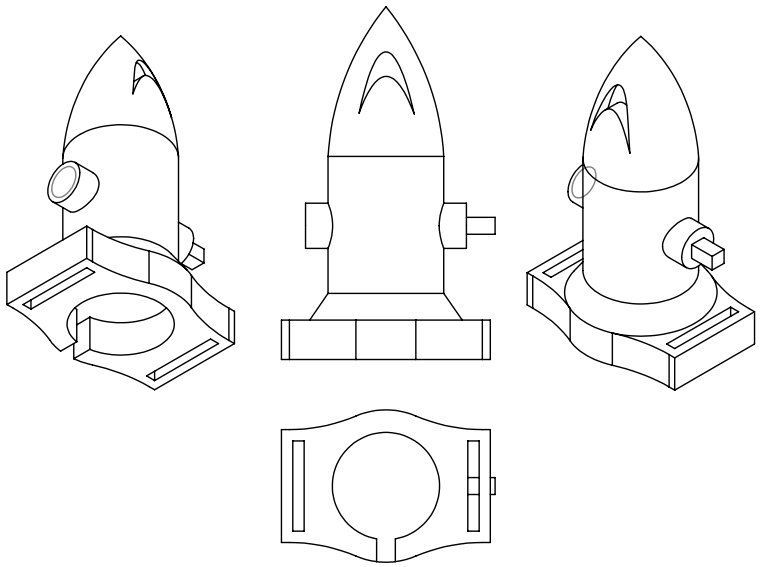


Figura 3.6: Dibujo de la pieza para adaptar el sensor de fuerza.

Figura 3.7: Robot con el sensor de fuerza adaptado.

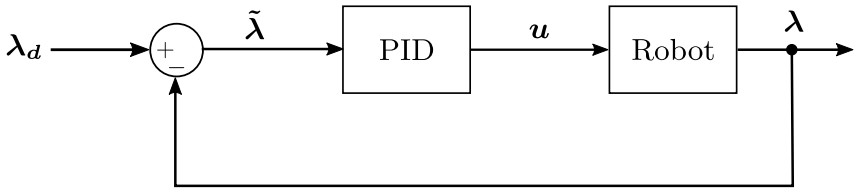


Figura 3.8: Diagrama de bloques del controlador.

3.4. Resultados experimentales

El primer controlador que se implementó para probar el funcionamiento del sensor fue el que se muestra en la Figura 3.8. Se genera una fuerza de referencia constante en dirección de uno de los ejes coordenados, lo cual hace que el robot empuje constantemente tratando de alcanzar el valor establecido. El usuario opone resistencia al movimiento para generar una lectura y debe tratar de orientar al efector final de forma que quede lo más perpendicular posible a la dirección de la fuerza de referencia o de lo contrario la lectura será inferior a la realidad. Para que esto no fuera necesario se requeriría medir las componentes de la fuerza en los tres ejes de referencia y calcular la resultante. La ecuación del controlador está dada por:

$$\boldsymbol{u} = \boldsymbol{K_p}\tilde{\boldsymbol{\lambda}} + \boldsymbol{K_i}\int_0^t \tilde{\boldsymbol{\lambda}}\, dt + \boldsymbol{K_d}\dot{\tilde{\boldsymbol{\lambda}}}$$

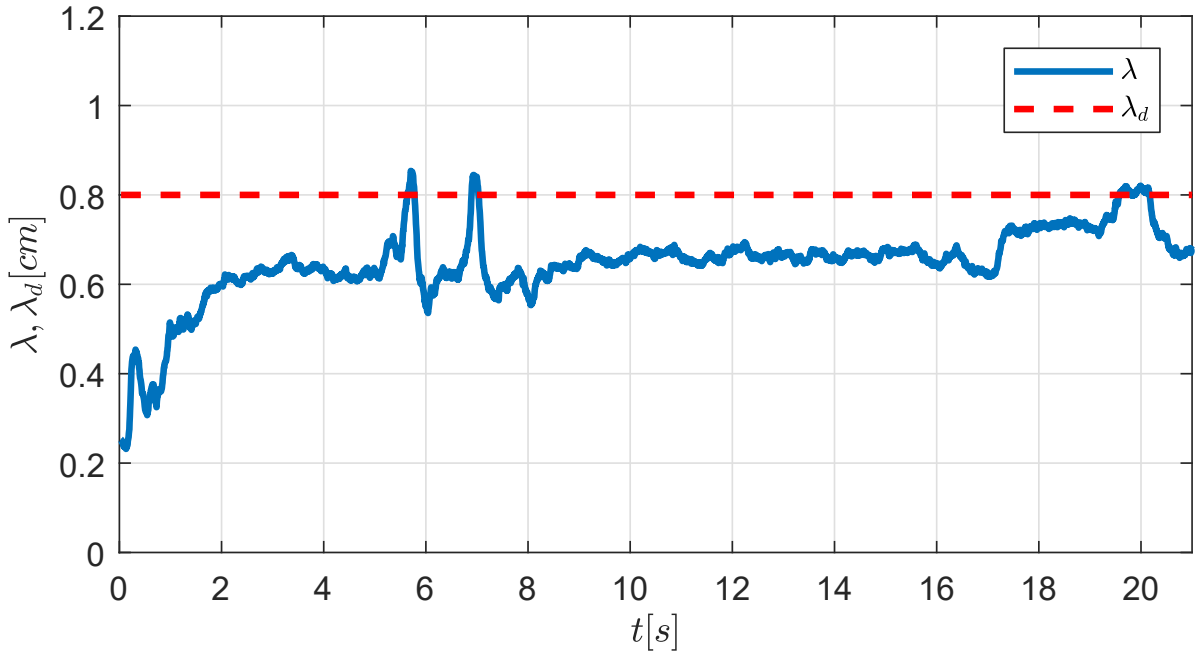
La derivada y la integral de la señal de error se calculan numéricamente mediante las ecuaciones siguientes.

$$\int_0^t \tilde{\boldsymbol{\lambda}}\, dt \approx (\tilde{\boldsymbol{\lambda}}_{\boldsymbol{act}} + \tilde{\boldsymbol{\lambda}}_{\boldsymbol{ant}})\Delta t \quad \dot{\tilde{\boldsymbol{\lambda}}} \approx \frac{\tilde{\boldsymbol{\lambda}}_{\boldsymbol{act}} - \tilde{\boldsymbol{\lambda}}_{\boldsymbol{ant}}}{\Delta t} \tag{3.2}$$

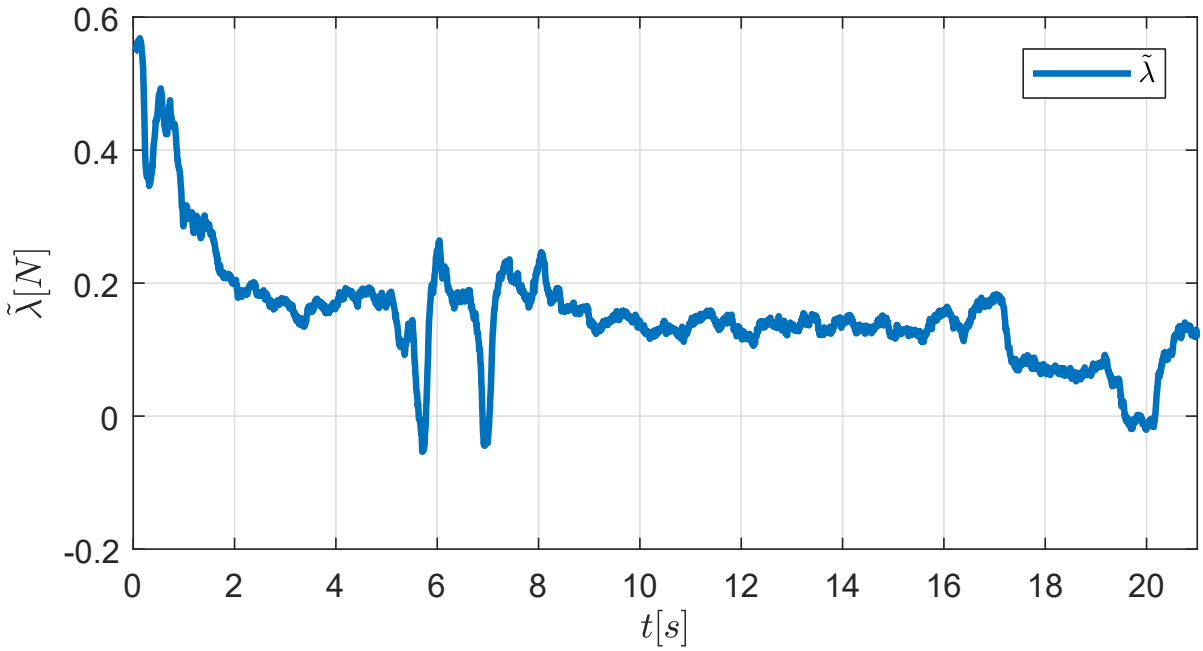
donde:

- $\tilde{\boldsymbol{\lambda}}_{\boldsymbol{act}}$ es el error de fuerza en la iteración actual.
- $\tilde{\boldsymbol{\lambda}}_{\boldsymbol{ant}}$ es el error de fuerza en la iteración anterior.
- Δt es el tiempo transcurrido entre la iteración actual y la anterior.

En la Figura 3.9 se pueden ver las gráficas del experimento. En la parte superior se aprecia la fuerza de referencia (línea punteada) y la real en el efector final. En la parte inferior se puede apreciar el error de fuerza. Se puede ver que durante todo el experimento el error de fuerza fue cercano a cero, que es justamente lo que se desea. Las variaciones bruscas corresponden con perturbaciones que se añadieron durante el experimento sujetando al efector final del robot y presionando el botón de la celda de carga para generar una lectura mayor a la realidad, y se puede ver que el controlador fue capaz de compensarlas.



(a) Señal de referencia y real.



(b) Señal de error.

Figura 3.9: Regulación de fuerza.

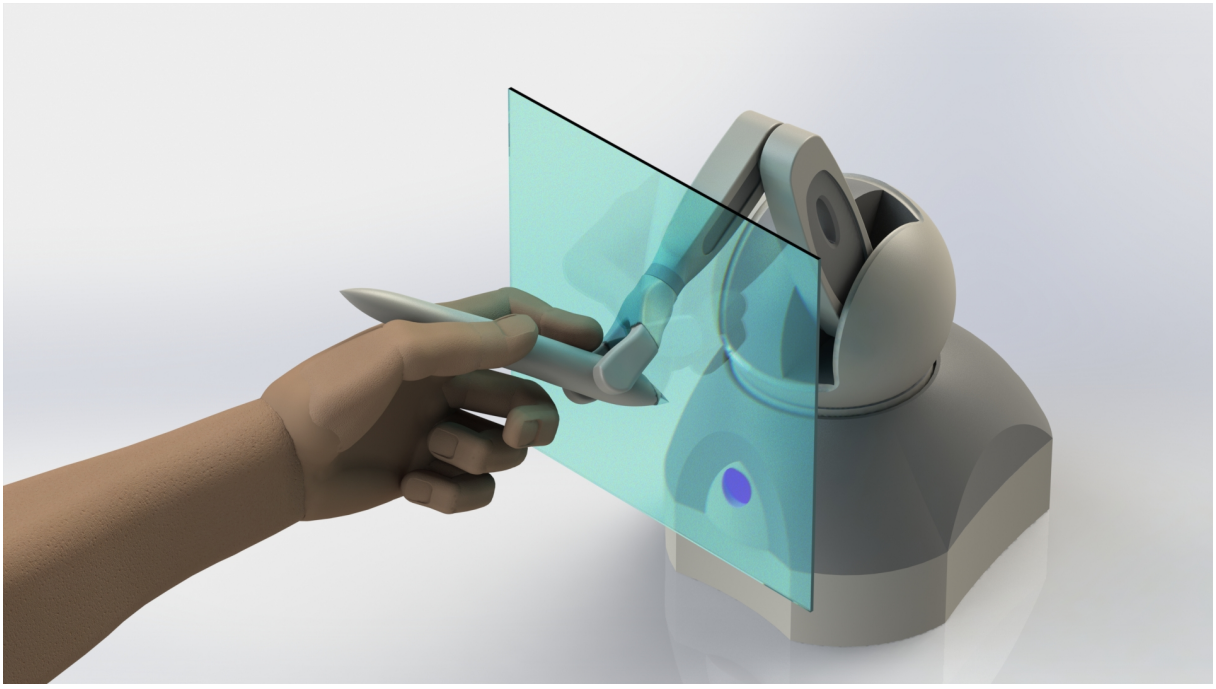


Figura 3.10: Segundo experimento realizado.

Se hizo un segundo experimento como el que se muestra en la Figura 3.10. El objetivo es que el usuario sienta que hay un plano infinito dentro del espacio de trabajo del robot cuya ecuación es

$$\varphi(x, y, z) = x - 0.1 = 0 \tag{3.3}$$

y la dirección de la fuerza de reacción que debe tener el robot se calcula mediante su gradiente.

$$\nabla\varphi = \begin{bmatrix} \frac{\partial\varphi}{\partial x} \\ \frac{\partial\varphi}{\partial y} \\ \frac{\partial\varphi}{\partial z} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \tag{3.4}$$

La magnitud de la fuerza se calcula mediante la ley de control. Si se usa la ecuación de una identidad geométrica distinta y su gradiente para calcular la dirección de la fuerza de reacción, entonces se pueden simular distintos lugares geométricos dentro del espacio de trabajo del robot.

Mientras no se toque al plano virtual, la fuerza generada es nula. En cuanto hay contacto, se generan dos señales de referencia. La primera le indica al robot que su efector final debe estar posicionado sobre el plano (es decir, se controla la posición), mientras que la segunda lo obliga mantener una fuerza de contacto constante. El robot intentará

alcanzar ambos objetivos simultáneamente mediante la ley de control de la ecuación 3.2. Se trata de dos controladores PID, uno para cada variable.

$$u = K_p\tilde{\mathbf{q}} + K_i\int \tilde{\mathbf{q}}dt + K_d\dot{\tilde{\mathbf{q}}} + K_{pf}\tilde{\boldsymbol{\lambda}} + K_{if}\int \tilde{\boldsymbol{\lambda}}dt + K_{df}\dot{\tilde{\boldsymbol{\lambda}}} \tag{3.5}$$

En donde:

- $\tilde{\boldsymbol{q}} = \boldsymbol{q_d} - \boldsymbol{q}$ es el vector de errores de posición.
- $\tilde{\boldsymbol{\lambda}} = \boldsymbol{\lambda_d} - \boldsymbol{\lambda}$ es el vector de errores de fuerza.
- $\dot{\tilde{\boldsymbol{q}}} = \dot{\boldsymbol{q}}_{\boldsymbol{d}} - \dot{\boldsymbol{q}}$ es la derivada del vector de errores de posición.
- $\dot{\tilde{\boldsymbol{\lambda}}} = \dot{\boldsymbol{\lambda}}_{\boldsymbol{d}} - \dot{\boldsymbol{\lambda}}$ es la derivada del vector de errores de fuerza.
- $\boldsymbol{K_p}, \boldsymbol{K_{pf}}$ son matrices diagonales positivas definidas de ganancias proporcionales para posición y fuerza respectivamente.
- $\boldsymbol{K_d}, \boldsymbol{K_{df}}$ son matrices diagonales positivas definidas de ganancias derivativas para posición y fuerza respectivamente.
- $\boldsymbol{K_i}, \boldsymbol{K_{if}}$ son matrices diagonales positivas definidas de ganancias integrales para posición y fuerza respectivamente.

Para sintonizar el controlador de posición es posible usar la metodología descrita en el Capítulo 2. Para ello es necesario conocer la matriz de inercias y el vector de gravedad del robot, los cuales se reportan en (Rodríguez, 2014):

$$\boldsymbol{H} = \begin{bmatrix} \theta_1 c_3^2 + 2\theta_2 c_2 c_3 + \theta_3 c_2^2 & 0 & 0 \\ 0 & 2\theta_2 s_2 s_3 + 2\theta_2 c_2 c_3 + \theta_1 + \theta_3 & \theta_2 s_2 s_3 + \theta_2 c_2 c_3 + \theta_1 \\ 0 & \theta_2 s_2 s_3 + \theta_2 c_2 c_3 + \theta_1 & \theta_1 \end{bmatrix}$$

$$\boldsymbol{g} = \begin{bmatrix} 0 \\ \theta_7 g c_2 \\ \theta_8 g c_3 \end{bmatrix}$$

donde el valor de los parámetros $\theta_1, \ldots, \theta_8$ fue estimado en el mismo documento a través de un controlador adaptable:

- $\theta_1 = m_3 l_{c3} + I_3 = 1.4 \times 10^{-5}$
- $\theta_2 = a_2 m_3 l_{c3} = -5.5 \times 10^{-6}$
- $\theta_3 = m_2 l_{c2} + a_2^2 m_3 + I_2 = 4 \times 10^{-5}$

- $\theta_7 = m_2 l_{c2} + a_3 m_3 = 5.2 \times 10^{-3}$
- $\theta_8 = m_3 l_{c3} = 5.27 \times 10^{-3}$

Siguiendo el procedimiento de sintonización, primero es necesario calcular la constante k_g:

$$k_g = n(Max_{i,j,q}|\frac{\partial g_i(q)}{\partial q_j}|)$$

y las derivadas parciales del vector de gravedad son

$$\begin{aligned}
\frac{\partial g_1}{\partial q_1} = \frac{\partial g_1}{\partial q_2} = \frac{\partial g_1}{\partial q_3} = 0 \\
\frac{\partial g_2}{\partial q_1} = \frac{\partial g_2}{\partial q_3} = 0, \frac{\partial g_2}{\partial q_2} = -\theta_7 g \sin q_2 \\
\frac{\partial g_3}{\partial q_1} = \frac{\partial g_3}{\partial q_2} = 0, \frac{\partial g_3}{\partial q_3} = -\theta_8 g \sin q_3
\end{aligned}$$

de donde se obtiene

$$\begin{aligned}
k_g = n(5.27 \times 10^{-3}\theta_8) \\
k_g = 0.1551
\end{aligned}$$

El siguiente paso consiste en calcular los valores característicos máximos y mínimos de la matriz de inercia del robot. Dicha matriz depende de funciones no lineales de las variables q_1, q_2, q_3, por lo que encontrar una solución analítica no es sencillo, y es por ello que se creó un programa que la evalúa numéricamente en incrementos de 0.1 en el rango $[0, 2\pi]$ para cada una de las tres variables. Siguiendo tal procedimiento se encontró que:

$$\begin{aligned}
\lambda_{min} = 3.354 \times 10^{-9} \\
\lambda_{max} = 53.31 \times 10^{-6}
\end{aligned}$$

Los cálculos a realizar son:

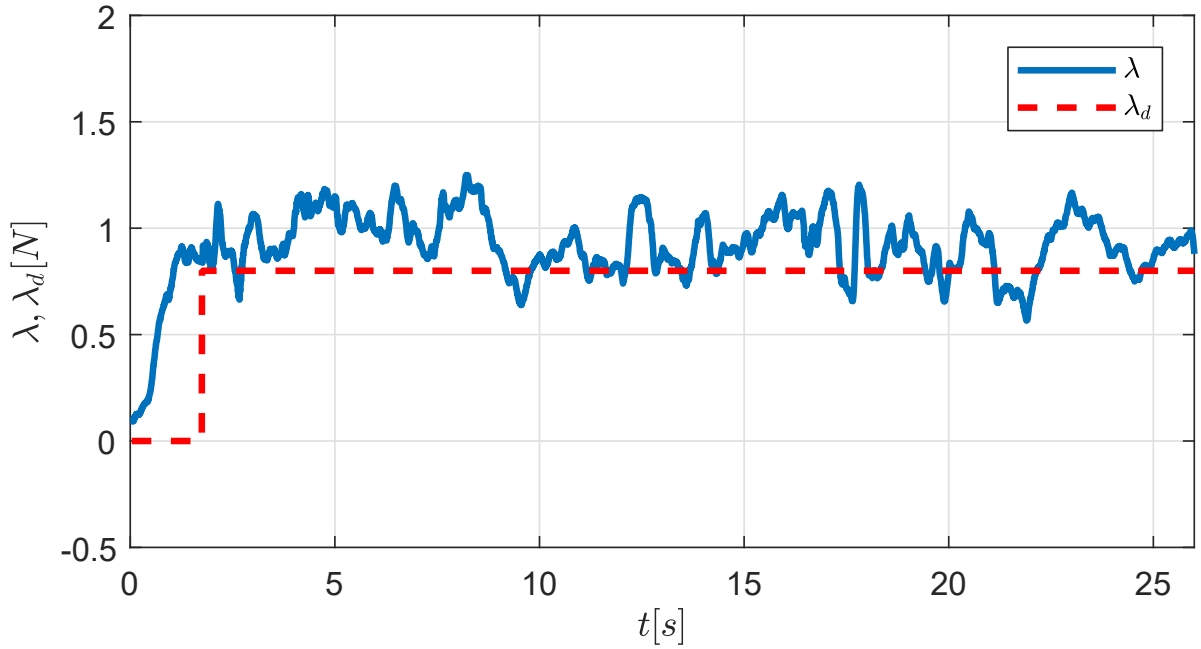
- $\lambda_{max}(\boldsymbol{K_i}) \geq \lambda_{min}(\boldsymbol{K_i}) > 0$
- $\lambda_{max}(\boldsymbol{K_p}) \geq \lambda_{min}(\boldsymbol{K_p}) > 0.1551$
- $\lambda_{max}(\boldsymbol{K_d}) \geq \lambda_{min}(\boldsymbol{K_d}) > \frac{\lambda_{max}(\boldsymbol{K_i})}{\lambda_{min}(\boldsymbol{K_p}) - k_g} \frac{\lambda^2_{max}(\boldsymbol{H})}{\lambda_{min}(\boldsymbol{H})}$

con lo que finalmente se propusieron las matrices:

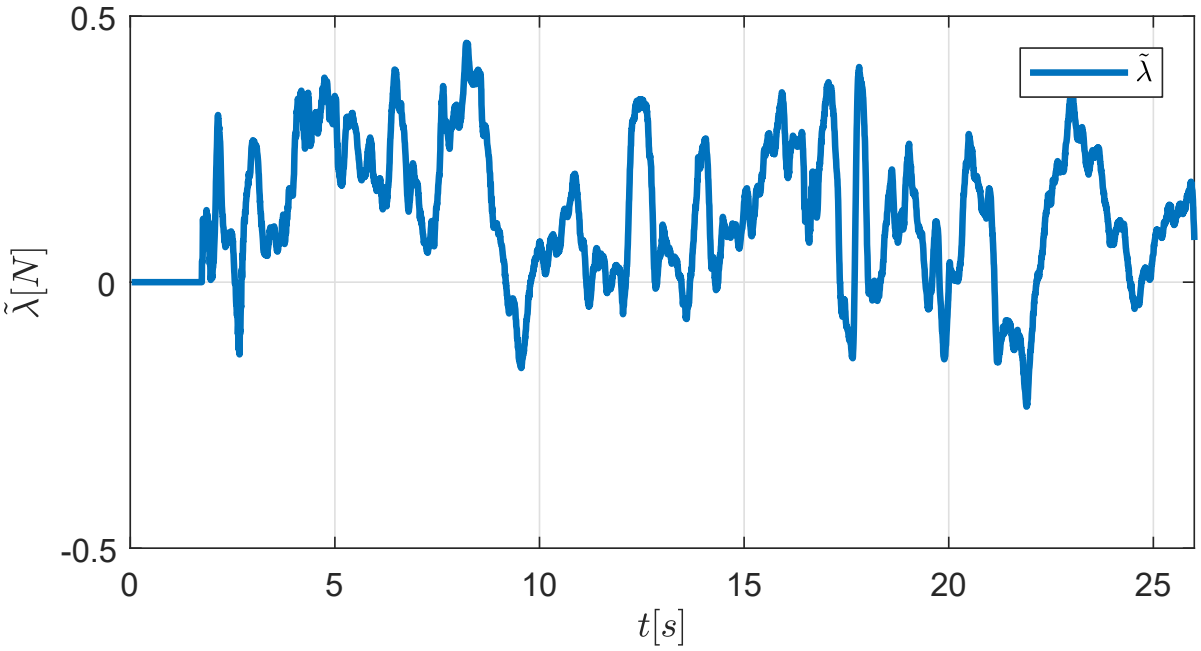
$$\boldsymbol{K_i} = \begin{bmatrix} 1.0 & 0 & 0 \\ 0 & 1.0 & 0 \\ 0 & 0 & 1.0 \end{bmatrix}$$
$$\boldsymbol{K_p} = \begin{bmatrix} 5.0 & 0 & 0 \\ 0 & 5.0 & 0 \\ 0 & 0 & 5.0 \end{bmatrix}$$
$$\boldsymbol{K_d} = \begin{bmatrix} 0.3 & 0 & 0 \\ 0 & 0.3 & 0 \\ 0 & 0 & 0.3 \end{bmatrix}$$

En la Figura 3.11 se muestran las gráficas de fuerza y en la Figura 3.12 las de posición del experimento. Con este controlador el usuario podrá sentir el plano y además el robot intentará mantener la fuerza de contacto constante. Tanto el error de fuerza como el de posición se mantienen cercanos al cero en todo momento, por lo que se puede concluir que el experimento fue exitoso y el robot alcanzó ambos objetivos de control.

Por último es importante recalcar la diferencias entre ambos controladores. En el primero no se regula la posición del robot, y un control de fuerza por sí solo no es de gran utilidad porque el robot no sabrá cómo moverse dentro del ambiente en el que está, lo cual puede llevar a un comportamiento errático del sistema. En este caso particular, el robot intentará empujar indefinidamente en un solo sentido si no registra fuerza en el efector final. Este controlador no es adecuado para una aplicación háptica porque no sería capaz de recrear las propiedades de un objeto virtual debido a la ausencia de un control de posición, pero gracias a él se demuestra que ya es posible regular fuerza. En contraste, el controlador del segundo experimento es capaz de recrear una forma geométrica mediante la posición del mecanismo y además permite hacer regulación de fuerza, lo cual lo convierte en una aplicación háptica porque cuenta con todos los elementos necesarios.

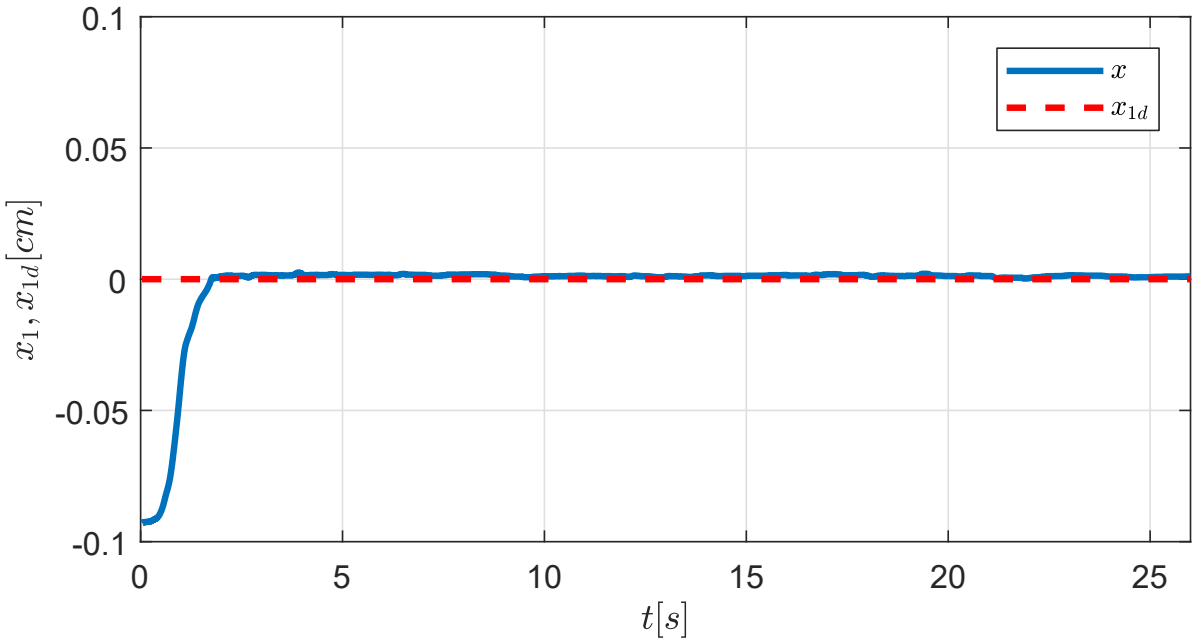


(a) Señal de referencia y real.

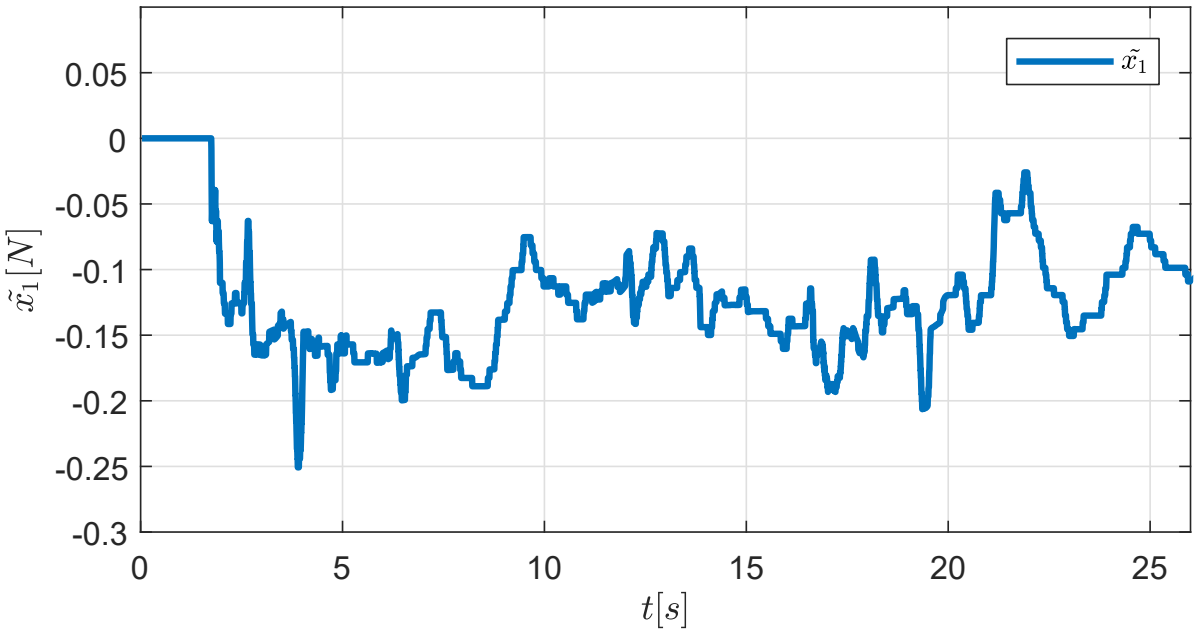


(b) Señal de error.

Figura 3.11: Gráficas de fuerza.



(a) Señal de referencia y real.



(b) Señal de error.

Figura 3.12: Gráficas de posición.

Capítulo 4

Diseño de la muñeca esférica

El mecanismo paralelo en el que está basado el robot Novint Falcon es capaz de mover su efector final en un espacio tridimensional. En la Figura 4.1 se visualiza los movimientos que se pueden realizar y como se puede apreciar, la esfera siempre permanece con una orientación fija. El usuario puede usar su brazo para operar al robot, pero si pudiera además usar la muñeca, la manipulación del efector final sería más natural, lo cual es indispensable en aplicaciones inmersivas. Para ello se recurre a una muñeca esférica, que es un mecanismo con las siguientes características:

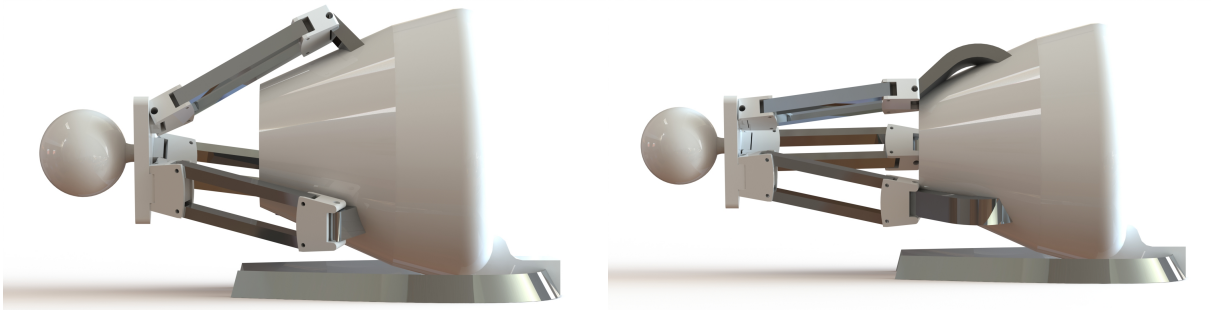
- Posee tres grados de libertad.
- Permite rotación en torno a tres ejes perpendiculares entre sí, los cuales se intersectan en un punto común llamado *centro de la muñeca*.

Dotar a un robot con esta cadena cinemática tiene una gran ventaja: permite desacoplar (Spong et al., 2006) los problemas de posición y orientación, es decir, es posible analizar el posicionamiento y la orientación del efector final por separado y después obtener la solución final al problema. Esto se debe a que la posición del centro de la muñeca depende únicamente de los primeros tres grados de libertad.

4.1. Diseño mecánico

Es posible crear varios mecanismos que cumplan con las características de una muñeca esférica. Cada uno tendrá ciertas propiedades, ventajas y desventajas, por lo que es necesario establecer criterios de selección:

- Deberá ser de bajo costo. Esto está en conformidad con uno de los objetivos principales de la tesis: desarrollar aplicaciones hápticas a un costo inferior.
- El mecanismo deberá ser ligero y por ende de inercia pequeña para que el robot pueda moverlo de forma ágil y además resulte cómodo para el usuario.



(a) Extensión.

(b) Movimiento lateral.

(c) Movimiento vertical.

Figura 4.1: Movimientos que puede realizar el robot Falcon.

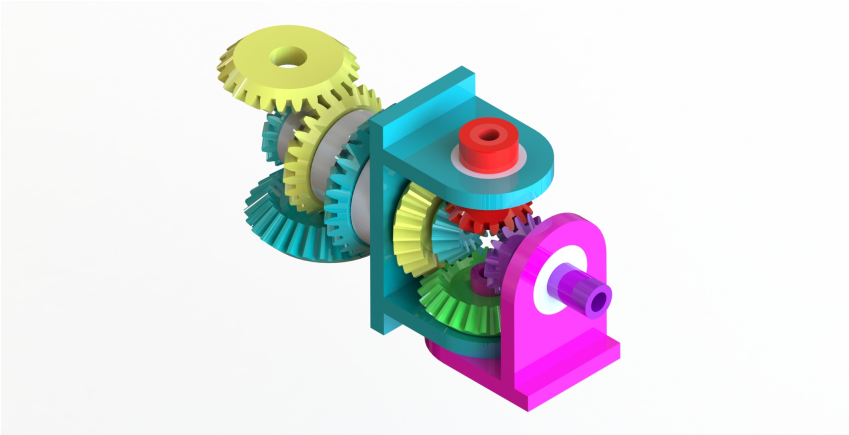


Figura 4.2: Muñeca esférica basada en trenes de engranes planetarios.

- Encontrar equilibrio entre simplicidad y funcionamiento adecuado.

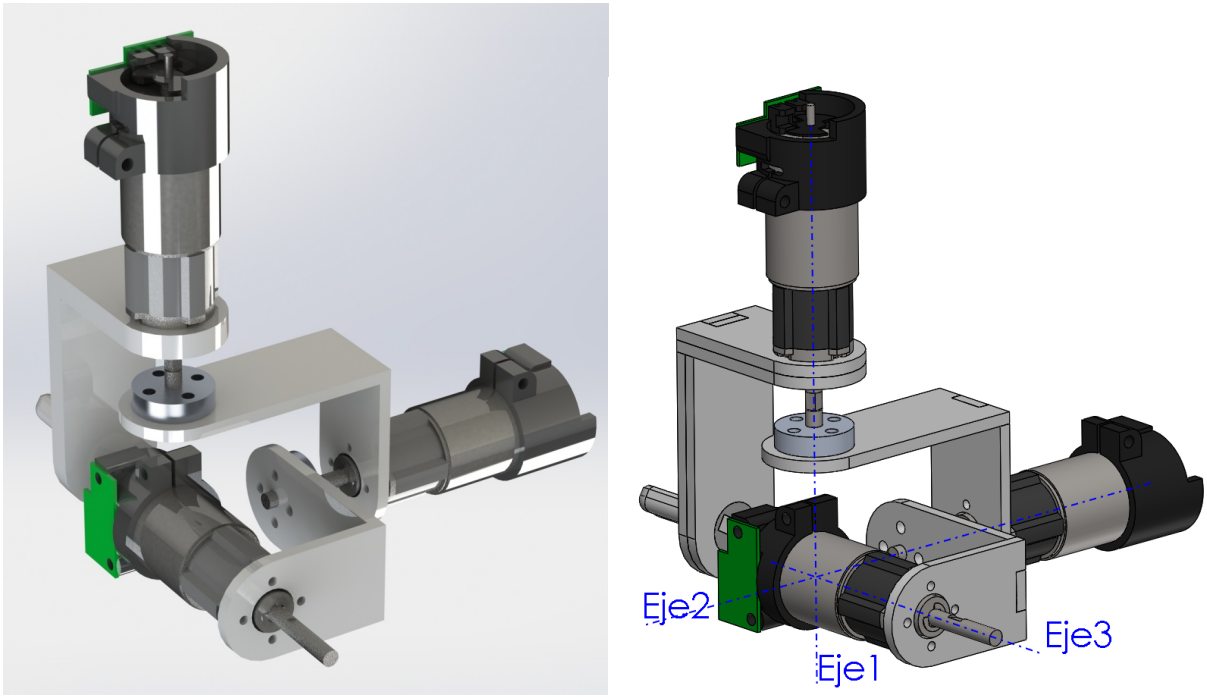
En la literatura se reportan algunos mecanismos (Tsai, 1999) (Tsai, 2000) (Hsu et al., 1999) que cumplen la función requerida similares al de la Figura 4.2. Su diseño se basa en extraer las propiedades de trenes de engranes planetarios cónicos, representándolos en forma de gráficas para encontrar todas las posibles configuraciones que tengan las características de una muñeca esférica. Este tipo de mecanismos tienen algunas grandes ventajas:

- Cuentan con tres entradas coaxiales que permiten montar los motores que mueven las piezas en la parte posterior de la muñeca, lo cual a su vez minimiza los efectos debidos a la inercia.

- Como todos los actuadores quedan montados en la plataforma móvil del robot, no existe el problema de que se enrede el cableado.

- Algunas configuraciones permiten rotación continua en torno a uno, dos o incluso los tres ejes de la muñeca.

A pesar de todas estas excelentes propiedades es evidente un problema muy grande: el mecanismo es sumamente complejo. Se fabricaron las piezas con una impresora 3D, pero la calidad no fue suficiente y el mecanismo se atascaba. Para lograr un funcionamiento correcto sería necesario un proceso de manufactura muy preciso y materiales de alta calidad, por lo que el costo se incrementaría significativamente. Es por ello que se decidió cambiar el diseño por uno mucho más simple que aún cuando no tenga todas las ventajas que se mencionaron, cumple con la función requerida y además se alinea mejor a los criterios de selección establecidos.



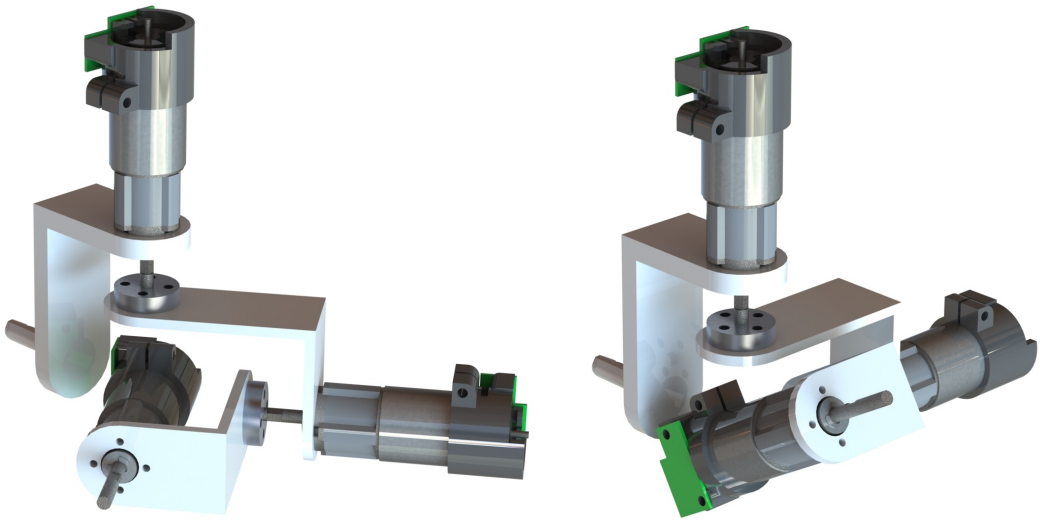
(a) Renderizado del mecanismo.

(b) Centro de la muñeca

Figura 4.3: Diseño de la muñeca esférica.

La estructura que se eligió para el mecanismo es como se describe a continuación: se monta en el eje del motor anterior un eslabón que asegura que el rotor del actuador siguiente quede perpendicular al eje de rotación anterior. En la Figura 4.3 (a) se muestra un renderizado del mecanismo y en la Figura 4.3 (b) se extendieron líneas imaginarias para indicar el centro de la muñeca. En la Figura 4.4 se muestran los movimientos que puede realizar el mecanismo en torno a los dos primeros ejes. Se omitió el movimiento del tercer eje porque no se notaría la rotación en una imagen debido a la simetría del rotor.

En la Figura 4.5 se muestra una fotografía del robot con la muñeca acoplada. En el eje del último motor de la muñeca se acopló el efector final, que es una pieza con forma similar al de un bolígrafo para que el usuario pueda sujetar y manipular al robot. Además en la fotografía se puede apreciar que dicha pieza tiene una rotación en torno a cada uno de los tres ejes, demostrando que el mecanismo cumple con la función para la cual fue diseñado: permitir el giro del efector final para dar mayor destreza al robot. Las piezas se fabricaron con corte láser en MDF de 3 milímetros, el cual es una buena opción debido a su bajo costo y fácil manejo. Es importante mencionar que se trata de un primer prototipo y no un producto final, lo cual se puede ver, por ejemplo, en que no hay orificios o rutas para manejar el cableado de los motores y sensores. Además el material usado es blando, lo cual podría resultar en una falla con el uso del mecanismo. A pesar de ello, tiene la ventaja del costo y demuestra bien el funcionamiento del mecanismo.



(a) Rotación respecto al primer eje. (b) Rotación respecto al segundo eje.

Figura 4.4: Movimientos de la muñeca.

La Tabla 4.1 presenta el precio aproximado de los componentes y procesos necesarios para desarrollar el mecanismo. Es importante señalar que todos los componentes se compraron en menudeo, lo que incrementa el costo del mecanismo. Si se fabricara en masa, el precio por unidad disminuiría significativamente.

El último aspecto importante a considerar es que el mecanismo adaptado añade un peso adicional en el efector final. Dicho peso es de aproximadamente 4.31[N], el cual es menor a la fuerza máxima que puede generar el robot, por lo que será posible compensarlo cuando se implemente un controlador. Gracias a su arquitectura paralela, el manipulador Falcon es capaz de cargar el peso añadido. Si se deseara lograr lo mismo con un robot serial como el Touch, entonces se requeriría de una mayor capacidad de carga, lo cual incrementaría su tamaño.

Característica	Costo[MXN]
Material	50
Manufactura piezas	30
Microcontrolador	400
Manufactura PCB	150
Componentes electrónicos	300
Motores	300
Total	1230

Tabla 4.1: Resumen de los costos de fabricación del mecanismo.

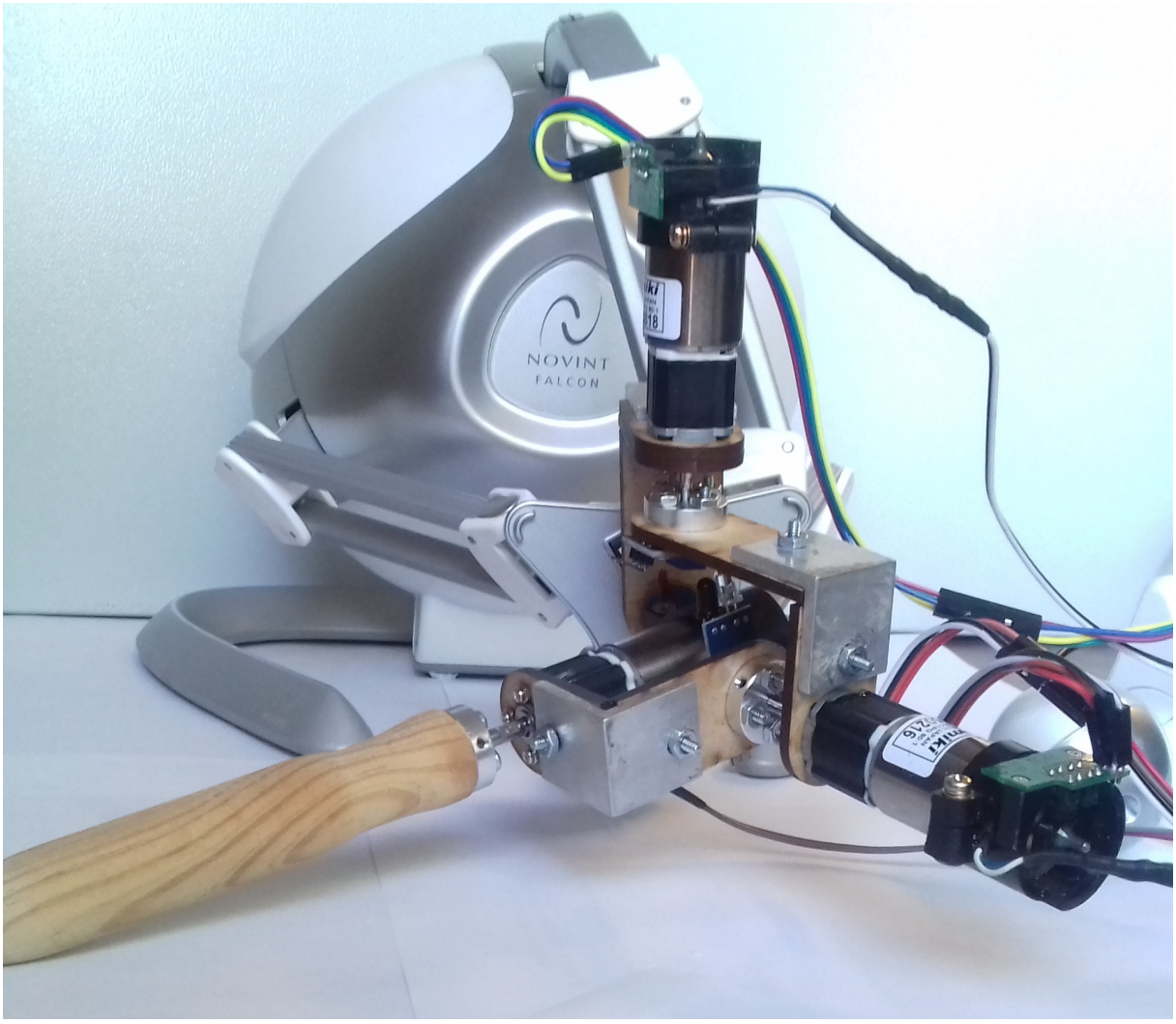


Figura 4.5: Robot con la muñeca adaptada.

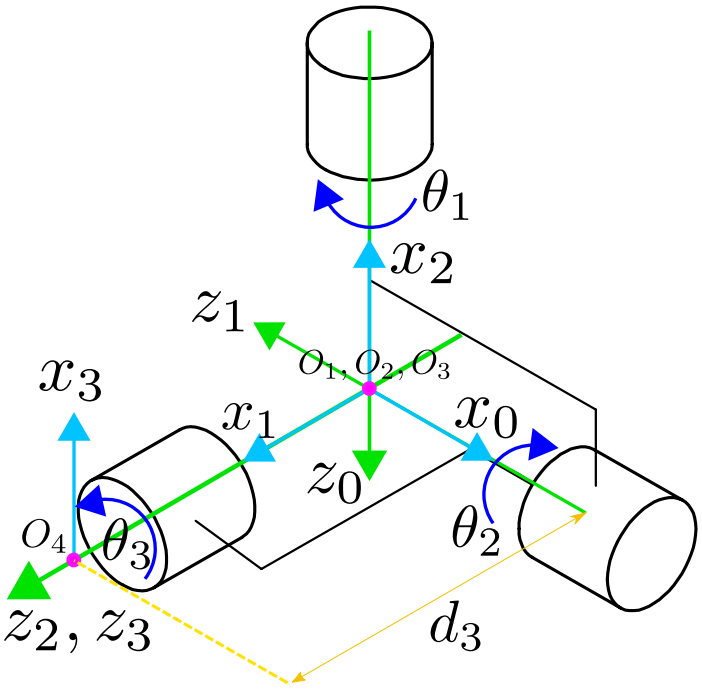


Figura 4.6: Uso del algoritmo Denavit-Hartenberg en la muñeca esférica.

Eslabón	$a_i[mm]$	$d_i[mm]$	$\alpha_i[^\circ]$	$\theta_i[^\circ]$
1	0	0	-90	θ_1
2	0	0	90	θ_2
3	0	54	0	θ_2

Tabla 4.2: Parámetros Denavit-Hartenberg de la muñeca esférica

4.1.1. Cinemática directa

La cinemática directa de la muñeca se obtuvo a través de la convención de Denavit-Hartenberg. Para ello en la Figura 4.6 se muestra en forma esquemática la muñeca y se han seguido los pasos del algoritmo para asignar los sistemas de referencia de cada articulación. En la Tabla 4.2 se muestra el valor de cada uno de los cuatro parámetros, y de acuerdo con ellos se construyen las matrices de transformación homogénea $A_i, i = 1 \ldots 3$, las cuales representan la transformación necesaria para llegar del sistema $i-1$ al sistema i.

$$A_1 = \begin{bmatrix} \cos(\theta_1) & 0 & -\sin(\theta_1) & 0 \\ \sin(\theta_1) & 0 & \cos(\theta_1) & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.1}$$

$$A_2 = \begin{bmatrix} \cos(\theta_2) & 0 & \sin(\theta_2) & 0 \\ \sin(\theta_{21}) & 0 & -\cos(\theta_2) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.2}$$

$$A_3 = \begin{bmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & 0 \\ \sin(\theta_3) & \cos(\theta_3) & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.3}$$

Al multiplicar todas las matrices se obtiene ${}^0T_3$, que representa la acumulación de todas las transformaciones necesarias para conocer la orientación del efector final del robot.

$${}^0T_3 = \begin{bmatrix} c\theta_1 c\theta_2 c\theta_3 - s\theta_1 s\theta_3 & -c\theta_3 s\theta_1 - c\theta_1 c\theta_2 s\theta_3 & c\theta_1 s\theta_2 & d_3 c\theta_1 s\theta_2 \\ c\theta_2 c\theta_3 s\theta_1 + c\theta_1 s\theta_3 & c\theta_1 c\theta_3 - c\theta_2 s\theta_1 s\theta_3 & s\theta_1 s\theta_2 & d_3 s\theta_1 s\theta_2 \\ -c\theta_3 s\theta_{21} & s\theta_3 s\theta_{21} & c\theta_2 & d_3 c\theta_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.4}$$

Para el proyecto que se realizó no es necesario estudiar la cinemática directa del robot porque el fabricante proporciona una librería dinámica, la cual resuelve internamente dicho problema. En el programa únicamente se ingresen las coordenadas a las que se desea que llegue el efector final en una función incluida en la librería e internamente se realizan los cálculos necesarios. Para un estudio completo de la solución consultar (Rodríguez, 2017).

4.2. Diseño electrónico

Una vez diseñado el mecanismo fue necesario crear la electrónica que permitiría controlarlo, la cual debe ser capaz de realizar las siguientes funciones:

- Leer el sensor del eje de cada motor de la muñeca.
- Realizar las operaciones necesarias para transformar las lecturas de cada sensor en posición angular.
- Generar la lógica para poder controlar el sentido y velocidad de giro de cada motor.
- Proveer la potencia necesaria para mover a cada motor.
- Proteger a los circuitos lógicos de los de potencia.

Esto se hizo mediante un circuito impreso cuyo proceso de diseño e implementación se detalla en esta sección.

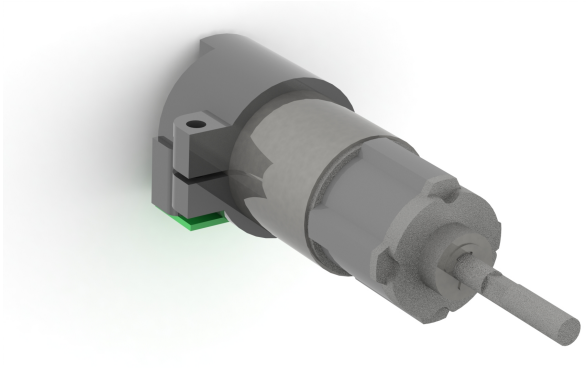


Figura 4.7: Motor Namiki 22CL-3501PG.

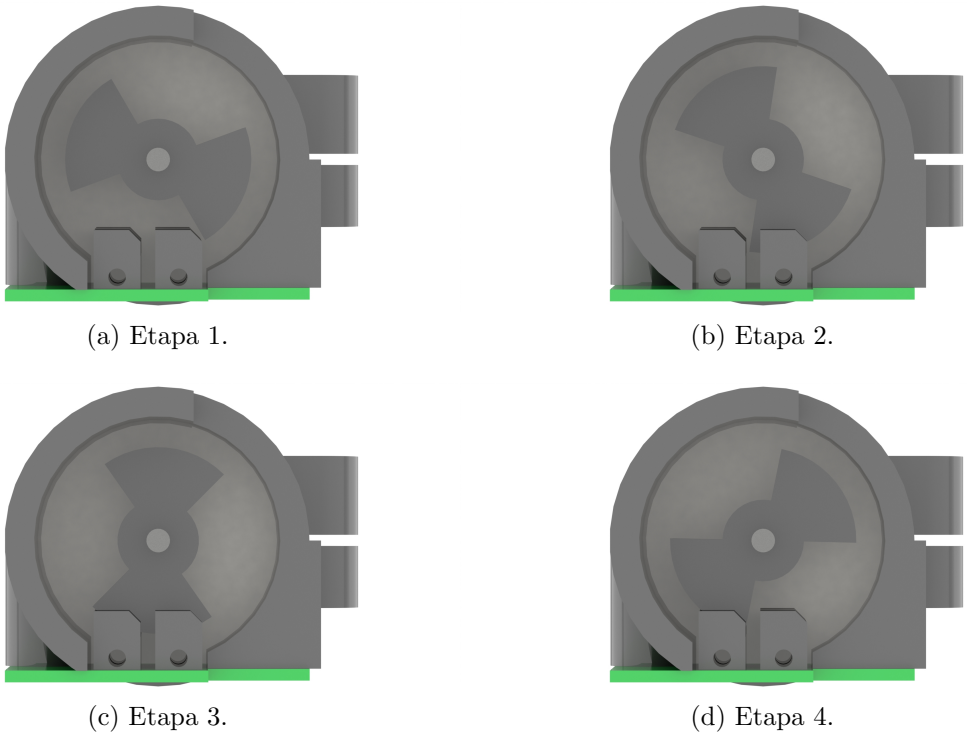
4.2.1. Lectura de posición angular

En la Figura 4.7 se muestra un motor Namiki 22CL-3501PG, que es el que se usó en la muñeca esférica. Consta básicamente de tres partes: un motor de corriente directa, una caja de reducción y un encoder. Este último es el sensor que permite leer la posición angular del rotor. Una vez montada la muñeca en el robot, dichas lecturas corresponden con las variables articulares $\boldsymbol{q}$.

Para explicar el funcionamiento de un encoder, en la Figura 4.8 se muestra la vista trasera del motor. El sensor consta de dos partes: un disco y dos sensores infrarrojos. El disco gira junto con el rotor y en los incisos (a) al (d) se muestran las posiciones claves para poder calcular la rotación del motor.

En el inciso (a) el disco no activa ninguno de los dos sensores infrarrojos, por lo que ambas señales permanecen en 1 lógico. Cuando el rotor gira y alcanza la posición de (b), el canal A cambia de estado a un 0 lógico. En (c) el disco está bloqueando la luz de ambos sensores, por lo que el canal B cambia a 0 lógico. Finalmente, en el inciso (d) el canal A regresa a 1 lógico y el ciclo se vuelve a repetir. En la Figura 4.9 se puede ver gráficamente las formas de onda y como se puede apreciar, se generan dos señales desfasadas 90° y es por ello que este tipo de sensor recibe el nombre de encoder de *cuadratura*.

Una característica de gran importancia de un encoder es su resolución, es decir, el número de pulsos que genera en una vuelta completa del rotor. Entre más pulsos genere, se podrá medir y controlar la posición angular con mayor exactitud. El encoder del motor Namiki cuenta únicamente con dos divisiones, por lo que podría pensarse que sólo se pueden realizar lecturas cada 180°, sin embargo, hay dos técnicas para mejorar este valor y este motor cuenta con ambas. La primer forma de hacerlo es cuando se usan dos sensores infrarrojos en vez de uno solo, ya que se generan dos pulsos más por cada división del disco. Además esto permite conocer el sentido de giro del rotor si se sigue la secuencia en la que cambian de valor las señales. La segunda forma de mejorar la resolución es a través de una caja de engranes cuyo eje gire más lentamente que el del rotor. El motor Namiki tiene una caja con una reducción de 80:1, lo que quiere decir que su rotor debe



(a) Etapa 1. (b) Etapa 2.

(c) Etapa 3. (d) Etapa 4.

Figura 4.8: Funcionamiento de un encoder de cuadratura.

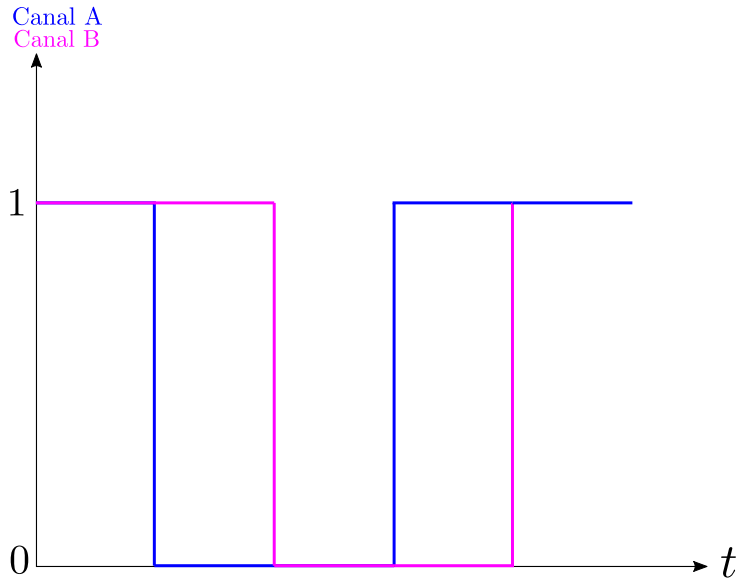


Figura 4.9: Forma de onda de las señales de un encoder.

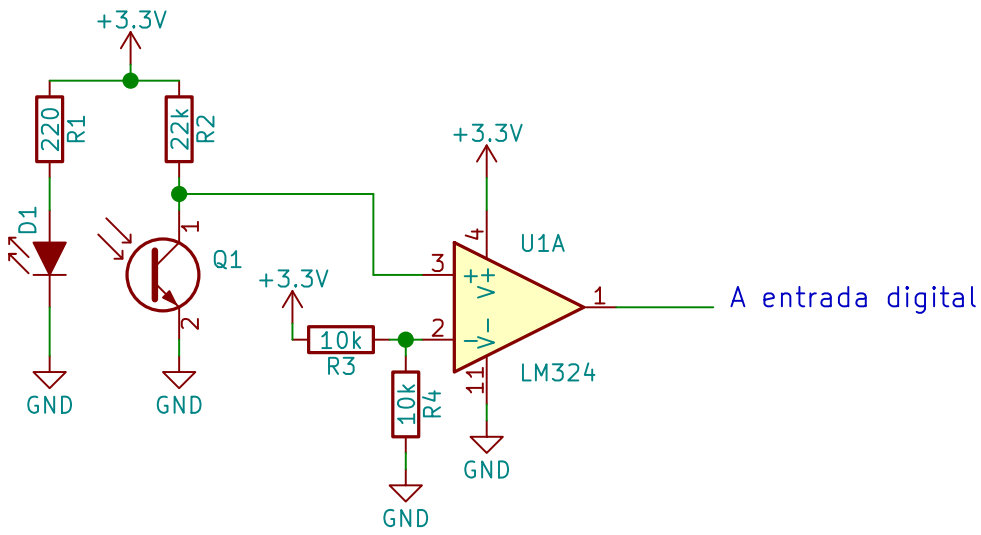


Figura 4.10: Circuito para cada canal del encoder.

girar 80 veces para que el de la caja de engranes dé una vuelta completa. Con todas estas consideraciones en mente, la posición angular se puede calcular a través de la ecuación

$$\theta_i = \frac{360 \times \text{Cuenta}}{4 \times 2 \times 80}[^\circ] \tag{4.5}$$

donde Cuenta representa la suma del número de pulsos que han ocurrido. Si el sentido de giro del rotor es positivo la cuenta incrementa y si es negativa se decrementa. Como se puede apreciar en el denominador, la resolución mejoró de $360^\circ/2 = 180^\circ$ a $360^\circ/(4 \times 2 \times 80) = 0.5625^\circ$. El factor de 80 se debe a la reducción de la caja de engranes y el factor de 2×4 debido a que el disco cuenta con dos divisiones y los sensores infrarrojos generan cuatro pulsos por cada división.

La última cuestión de gran importancia es acondicionar la señal del sensor. Se debe asegurar que los cambios sean limpios para evitar activaciones erráticas en las entradas digitales del microcontrolador. Es por ello que se usó un amplificador operacional en configuración de comparador no inversor para cada canal de cada encoder. Su función es comparar la señal del sensor contra el valor $V_{cc}/2$, saturándola a V_{cc} si está por encima o llevándola a cero si está por debajo de dicho valor. En la Figura 4.10 se muestra el circuito completo para leer posición angular. Matemáticamente se puede definir la función que realiza el comparador con la ecuación

$$V_{salida} = \begin{cases} V_{cc} & V_{entrada} > V_{cc}/2 \\ 0 & V_{entrada} < V_{cc}/2 \end{cases} \tag{4.6}$$

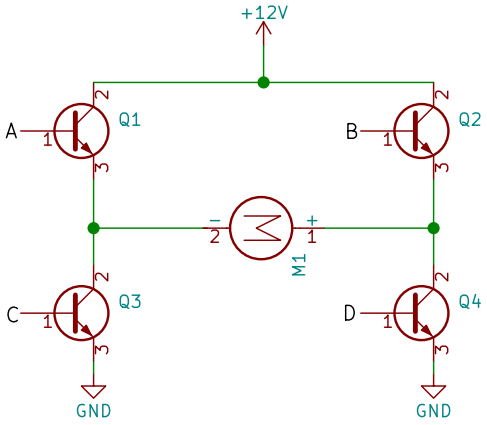


Figura 4.11: Puente H.

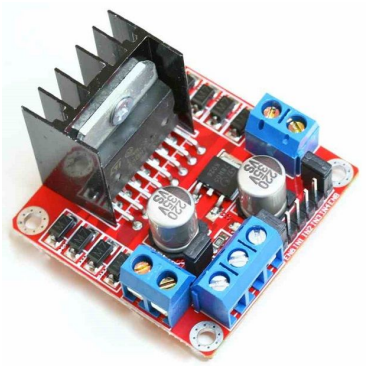


Figura 4.12: Módulo de puente H basado en el integrado L298.

4.2.2. Etapa de potencia

Una vez que se puede realizar la lectura de posición angular, es necesario un circuito que permita que los motores se muevan. El microcontrolador por sí solo es incapaz de generar la potencia necesaria para los actuadores, y es por ello que se requiere una electrónica de potencia para lograrlo. Se usó un puente H para lograr tal fin, el cual es un circuito de cuatro transistores que permite controlar el sentido y velocidad de giro de un motor. En la Figura 4.11 se muestra una versión simplificada de este dispositivo donde A, B, C y D representan cuatro señales de control digitales. Si se enciende A y D al mismo tiempo, el motor girará en un sentido y si se encienden B y C girará en sentido contrario. Nunca se debe encender A y C o B y D al mismo tiempo porque se crearía un corto circuito.

Se decidió usar un módulo prefabricado para realizar esta función con el fin de disminuir la complejidad del circuito impreso y ahorrar tiempo de diseño. Se usó el que se muestra en la Figura 4.12, el cual está basado en el circuito integrado L298. Se trata de un chip de alta potencia que incluye dos puentes H en un solo empaquetado, por lo que sirve para controlar hasta dos motores. Además cuenta con un pin de habilitación para

cada canal que se usa para controlar la velocidad de giro de cada motor mediante una señal modulada por ancho de pulso (PWM - Pulse Width Modulation) proveniente del microcontrolador. Dicha señal es cuadrada y oscila a una frecuencia constante, y la técnica consiste en variar su ciclo de trabajo (es decir, el tiempo que permanece encendida) para que el voltaje promedio que se aplica al motor sea variable. El ciclo de trabajo está definido por la ecuación

$$\delta = \frac{t_{on}}{T} \tag{4.7}$$

donde t_{on} representa el tiempo que la señal permanece en nivel alto y T su periodo. En la Figura 4.13 se muestran tres ejemplos de señales moduladas por ancho de pulso. Debido a que el motor Namiki tiene un voltaje nominal de 12[V], se usó ese valor de amplitud para las gráficas. Como se puede apreciar, el voltaje promedio que el motor recibe aumenta o disminuye dependiendo de si el ciclo de trabajo es mayor o menor respectivamente.

El último aspecto importante a cubrir es el aislamiento de los circuitos lógicos de los de potencia. Esto es necesario porque un motor cuenta con varios embobinados, los cuales pueden generar altos voltajes cuando hay cambios bruscos de velocidad o sentido de giro o cuando el usuario está manipulando al robot. Si se da cualquiera de estos casos, se podría dañar alguna entrada o salida del microcontrolador o incluso destruirlo por completo, y el daño se puede extender hasta el puerto de la computadora que controla todo el dispositivo. Es por ello que se usó un optoacoplador para cada señal que va hacia o viene desde los circuitos de potencia. Este dispositivo consiste en un fototransistor y un diodo emisor de luz infrarroja en un solo chip. Las dos funciones principales que realiza un optoacoplador son:

- Aislar eléctricamente dos señales.
- Servir como convertidor de nivel para que dos dispositivos que funcionen con voltajes diferentes se puedan comunicar.

En la Figura 4.14 se muestra el circuito que se utilizó para optoacoplar cada una de las señales que van hacia el puente H. El microcontrolador activará o desactivará al diodo emisor de luz según sea necesario, lo cual a su vez hará lo análogo con el fototransistor. Después se recupera la señal del colector del transistor para llevarla al puente H. En total se requirieron 9 de estos circuitos: dos señales para sentido y una para velocidad de giro para cada motor.

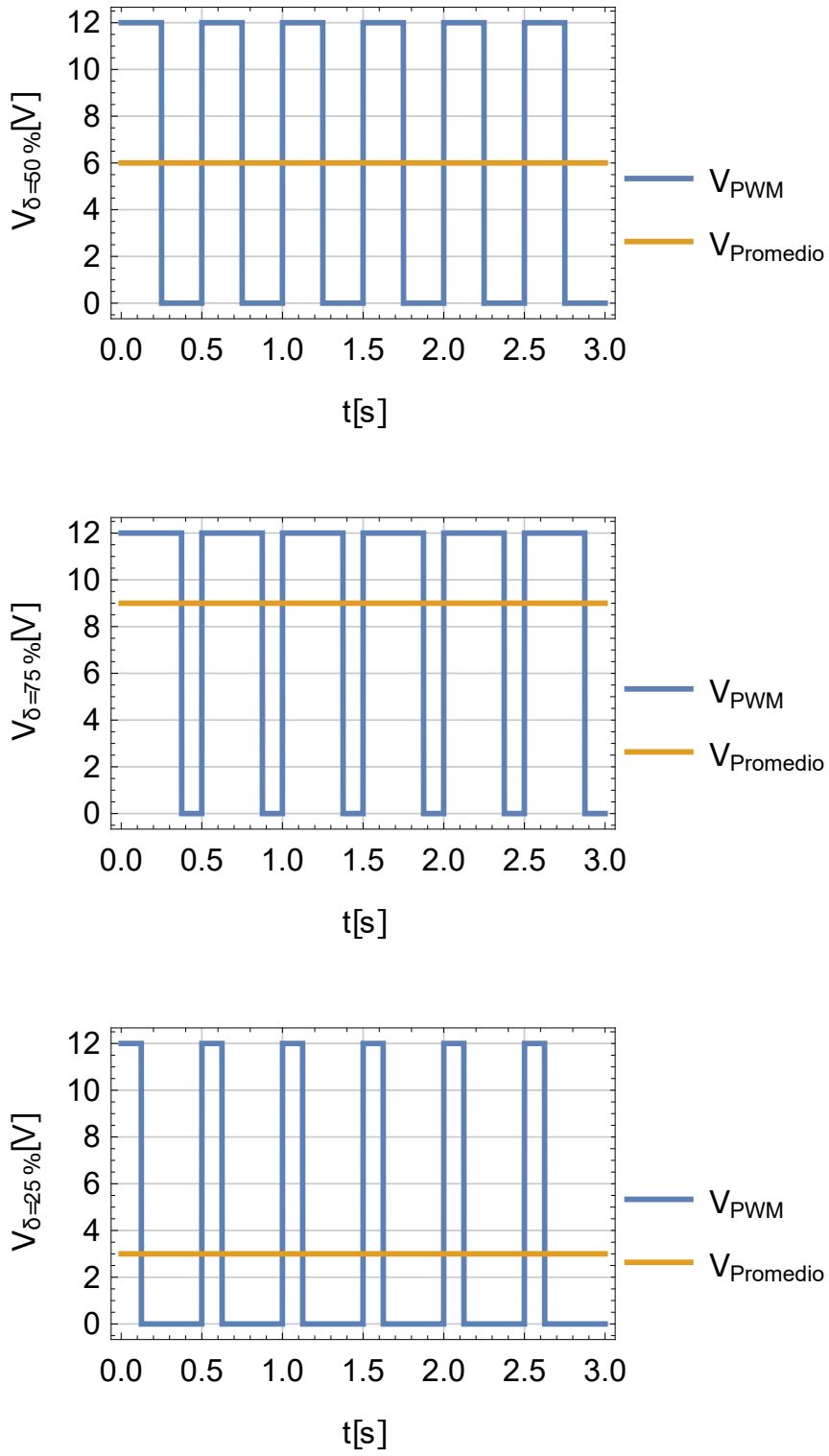


Figura 4.13: Modulación por ancho de pulso.

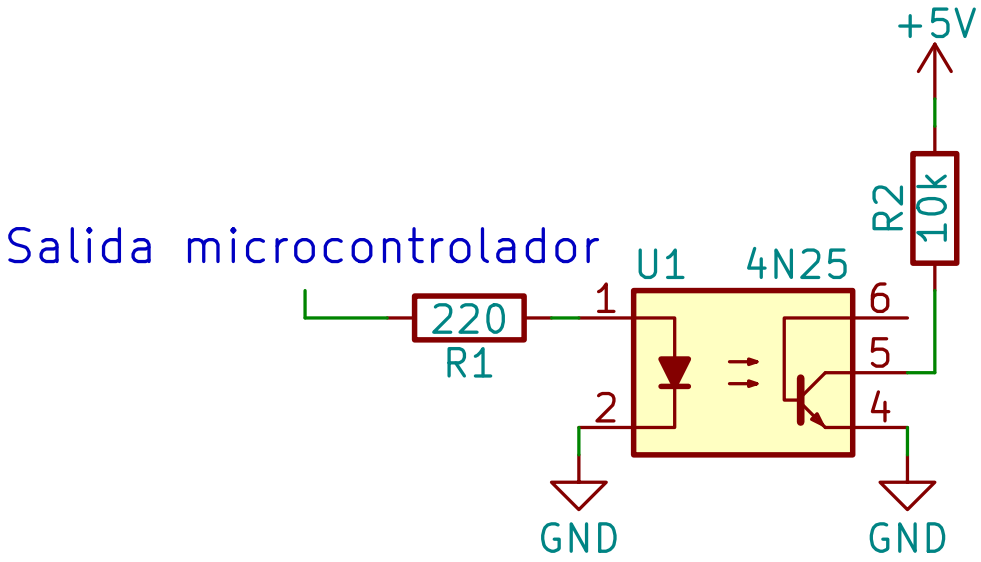


Figura 4.14: Circuito de optoacoplado.

4.2.3. Microcontrolador

El componente que hace funcionar la muñeca esférica es el microcontrolador. Este se encarga de leer los encodres y realizar los cálculos para determinar la posición y velocidad angular, comunicarse con la computadora que controlará todo el sistema, mandar las señales necesarias a los puentes H y generar la modulación por ancho de pulso. Algunos de los factores que se tomaron en cuenta para la selección del microcontrolador fueron:

- Frecuencia del reloj.
- Número de entradas y salidas digitales.
- Número de salidas que pueden generar modulación por ancho de pulso.
- Número de salidas analógicas.
- Número de entradas digitales que pueden funcionar para interrupciones externas.

Para la muñeca esférica son particularmente críticas la frecuencia del reloj y el número de entradas digitales con función de interrupción externa. Esto se debe a que es indispensable registrar cada pulso que generen los encoders ya que si se pierden algunos, la posición angular que se calcule será errónea, lo que a su vez podría generar inestabilidad en el controlador del mecanismo. Si se hace uso de las interrupciones es posible evitar el problema siempre y cuando el procesador sea lo suficientemente rápido para realizar todas las operaciones necesarias antes de que se genere otro pulso.

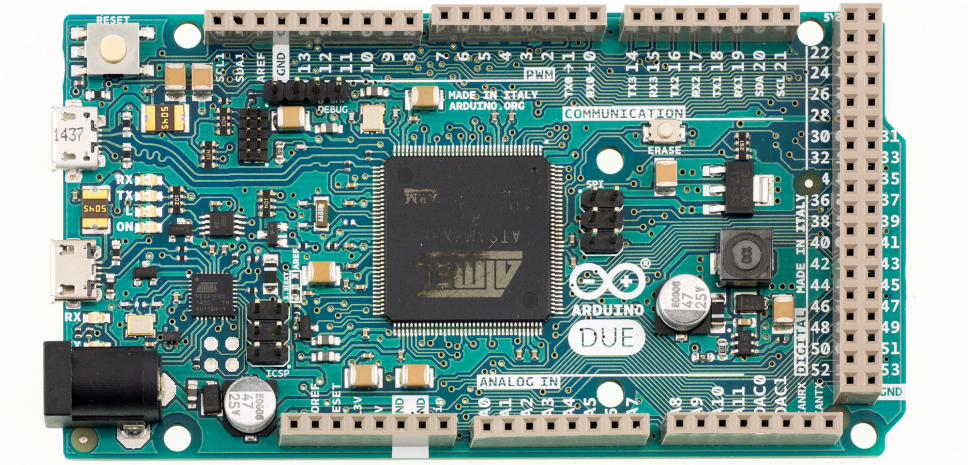


Figura 4.15: Arduino Due.

Característica	Valor
Frecuencia del reloj	84 MHz
Entradas y salidas digitales	54
Salidas que proveen PWM	12
Salidas analógicas	2
Entradas con interrupción	54 (todos los pines digitales)
Voltaje de operación	3.3V
Entradas analógicas	12
Arquitectura	32 bits

Tabla 4.3: Características del Arduino Due.

Los microcontroladores de la familia Arduino cuentan con varias características que los hacen una buena opción para este prototipo: entorno de desarrollo sencillo, gran cantidad de librerías para cubrir variedad de funciones y dispositivos, amplia documentación, etc. Existen muchos modelos con diferentes características, pero no todos fueron aptos para esta aplicación ya que algunos no contaban con suficientes pines para interrupción, mientras que otros tenían un reloj de frecuencia baja, etc. De todos ellos se escogió el modelo Due (Figura 4.15) por ser uno de los más potentes. En la Tabla 4.3 se muestran algunas de sus características.

Aunque en este trabajo no se hizo uso de las salidas analógicas, es importante mencionar que pueden resultar útiles a futuro. Como el microcontrolador cuenta con dos de ellas, sería posible controlar dos de los grados de libertad y el restante usando varios pines digitales junto con un convertidor digital-analógico de forma totalmente analógica.

4.3. Desarrollo de una aplicación

Una vez completado todo lo anterior, el siguiente paso fue realizar una aplicación sencilla para demostrar el funcionamiento del dispositivo. Recuérdense los elementos esenciales de un sistema de realidad virtual:

- Ambiente virtual.
- Realimentación sensorial.
- Interactividad.
- Inmersión.

Hasta este punto únicamente se cuenta con un dispositivo de realimentación sensorial: el robot Falcon con la muñeca eférica adaptada. Es necesario desarrollar el ambiente virtual y la interactividad (lo cual a su vez generará inmersión) para tener un sistema de realidad virtual completo.

Existen muchas formas de generar ambos elementos, pero el elegir una opción adecuada puede simplificar significativamente el trabajo que hay que realizar. Es por ello que una de las herramientas que destacó inmediatamente fue un motor de videojuegos. Estos programas integran una serie de características de gran utilidad para un sistema de realidad virtual:

- Manejo de objetos gráficos: renderizado, transformaciones (rotación, posición), iluminación, cámaras, sombras, texturas, etc.
- Motor de física: detección de colisiones entre objetos, simulación de leyes físicas reales (particularmente mecánicas), etc.
- Manejo de dispositivos de entrada como teclados, ratones o controles.

Estas características permitirían agilizar el desarrollo de la aplicación de realidad virtual y fue por ello que se decidió usar uno de estos programas. El siguiente paso es decidir cuál motor usar, ya que hay una gran cantidad disponibles. De entre todos ellos hay dos que destacan porque se han usado para desarrollar videojuegos de muy alta popularidad y éxito: Unity y Unreal Engine. Ambas son herramientas de nivel profesional que se pueden usar libremente mientras no sea de manera comercial (en cuyo caso las empresas que los desarrollan establecen cuotas). Finalmente se decidió usar Unity porque se disponía de documentación sobre cómo hacer funcionar al robot Falcon con dicho motor.

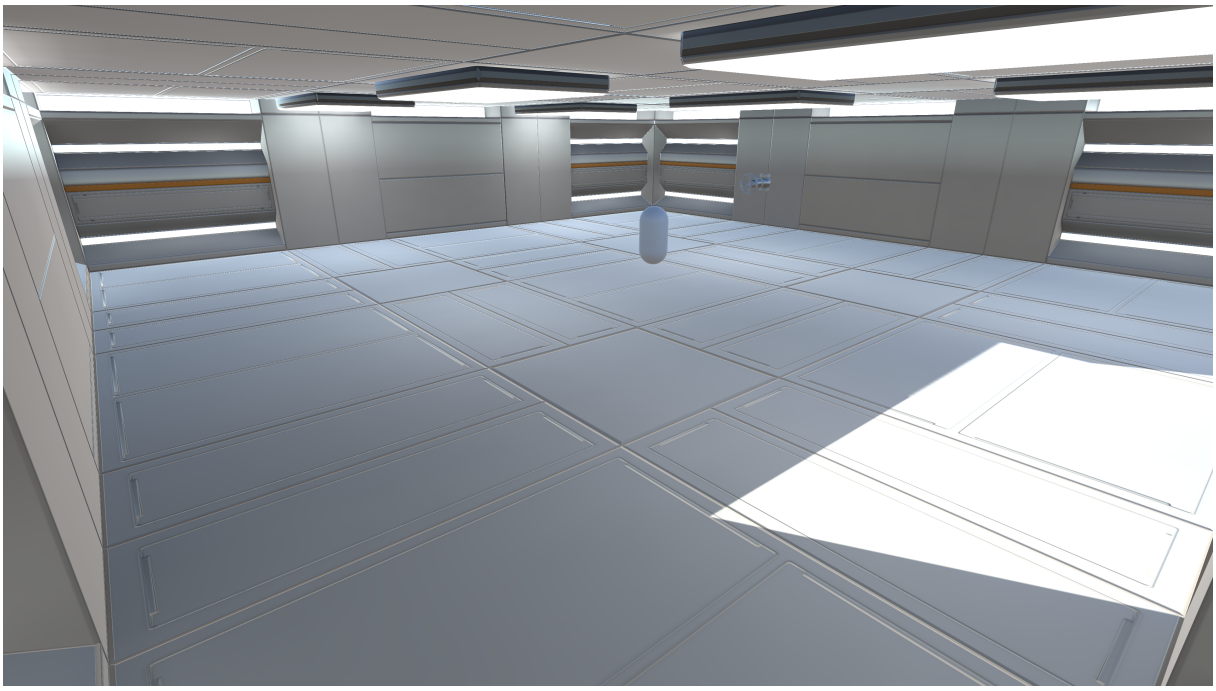


Figura 4.16: Mundo virtual.

4.3.1. El ambiente virtual interactivo

El mundo virtual que se usó para la aplicación se muestra en la Figura 4.16. Es un cuarto sencillo donde el usuario puede navegar desde una perspectiva en primera persona. En la misma imagen se puede ver un objeto con forma de cápsula, el cual representa al usuario. En la Figura 4.17 se muestra una captura de pantalla de la vista del cuarto desde la perspectiva este y como se puede ver, se añadió una retícula para que identifique el centro de la pantalla en todo momento.

Para que el mapa sea interactivo, Unity debe ser capaz de leer las entradas que haga el usuario y hacer modificaciones en el entorno. Para hardware estándar como el ratón, teclado, controles de consolas, entre algunos otros, Unity cuenta con soporte nativo, por lo que basta con unas cuantas líneas de código para realizar esa función, pero ese no es el caso del robot Falcon.

Para que el robot funcionara con Unity fue necesario usar la librería dinámica del fabricante para permitir la comunicación y así leer posiciones, enviar comandos, etc. Esto hizo que surgiera un inconveniente: la librería desarrollada por Novint tiene una arquitectura de 32 bits pero todas las computadoras modernas son de 64 bits. Este hecho limita a usar la versión de 32 bits de Unity, la cual oficialmente dejó de desarrollarse en diciembre del 2017. Otra desventaja es que las aplicaciones desarrolladas en esta arquitectura están limitadas a usar un máximo de 4GB de memoria RAM, lo cual podría volverse un problema conforme más compleja se vuelva la aplicación. Las funciones que se pueden realizar

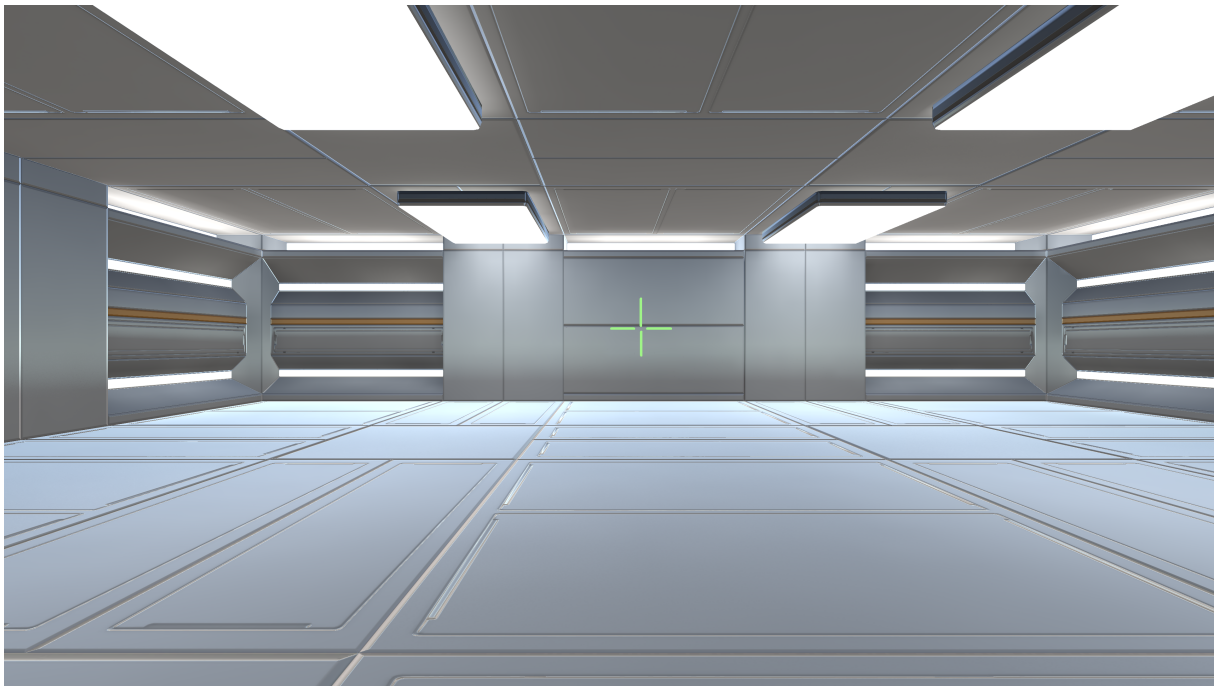


Figura 4.17: Vista en primera persona del usuario.

con la librería son:

- Abrir y cerrar canal de comunicación con el dispositivo.
- Leer la posición del efector final.
- Enviar instrucciones para activar uno o varios motores.

Con estas funciones básicas es suficiente para lograr interacción con el ambiente virtual. La primera de ellas es la navegación, que consiste en que el usuario se pueda mover y mirar sus alrededores. Para ello se realizan transformaciones de traslación y rotación en el avatar. Se asignaron las teclas "w" y "s" del teclado para trasladar la cápsula hacia adelante y hacia atrás respectivamente, mientras que para el movimiento en el eje z se usan las teclas "a" y "d" para izquierda y derecha respectivamente.

La imagen que ve el usuario en el monitor proviene de una cámara que está fija en la parte superior de la cápsula. Con las transformaciones de traslación el jugador podrá navegar por todo el mapa pero siempre verá a una dirección fija. Es por ello que es necesario permitir que la cámara rote horizontal y verticalmente, lo cual normalmente se haría con el ratón, pero el objetivo fue sustituirlo por el robot. Es por ello que para lograr la rotación se recupera la posición en el eje x y y del robot y se usan para hacer las transformaciones necesarias. Si el usuario mueve el efector final del robot como en la Figura 4.1-(b), la cámara rotará en torno al eje y y el usuario podrá mirar a su alrededor horizontalmente.

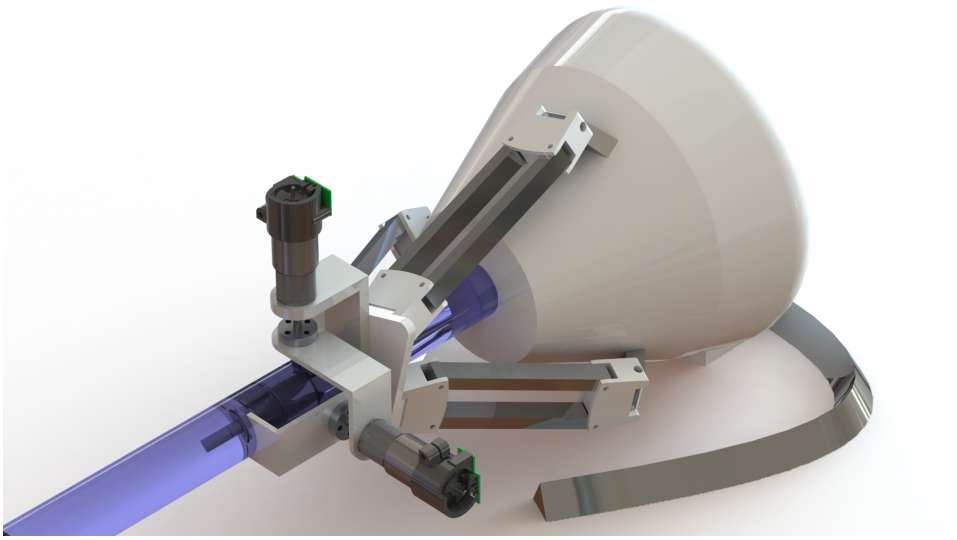


Figura 4.18: Representación gráfica de la zona muerta del controlador.

Si mueve el efector final como en la Figura 4.1-(c), entonces la vista se rotará alrededor del eje z y el usuario podrá ver hacia arriba y abajo. Para detener la rotación de la cámara es necesario llevar al robot exactamente a las coordenadas $(0, 0, z)$, pero como es muy complicado posicionar el efector final exactamente en dichas coordenadas, se estableció una zona muerta alrededor del origen del sistema de referencia del robot. Gráficamente se vería como en la Figura 4.18, donde el cilindro azul representa dicha zona. Así se provee al usuario una zona cómoda para trabajar y ser más preciso con los movimientos que se desean realizar.

4.3.2. Comunicación y control

Una vez que la navegación básica fue implementada se procedió a hacer una escena simple que demostrara algunas ventajas de tener seis grados de libertad. Para ello se colocó en el mapa de la aplicación el modelo de una válvula con el objetivo de que el usuario pueda interactuar con ella, girándola, a la vez que se le brinda realimentación háptica mediante la simulación de fricción viscosa. Esto se logra a través dos acciones:

- Se inicia un controlador que regula la posición del efector final en el origen del sistema de referencia, mientras que las primeras dos articulaciones de la muñeca también se regulan a cero. El objetivo es que el robot "pierda" cinco grados de libertad y el único que quede libre sea el que le permite al efector final girar en torno de su propio eje.

- Una vez realizado el paso anterior, el usuario únicamente puede mover el último grado de libertad, el cual opondrá cierta resistencia a la rotación del efector final

realizada por la persona, logrando así simular fricción.

Antes de poder realizar estas acciones, primero es necesario identificar cuándo desea el usuario manipular la válvula. Mientras esté en modo navegación, en cada ciclo de actualización se traza un rayo perpendicular al plano de la cámara que pasa justo por el centro de la imagen. Con ayuda de la retícula el usuario puede identificar hacia dónde apunta dicho rayo para así poder interactuar con objetos. Si se apunta la mira a la válvula y presiona la tecla "e", entonces Unity sabrá que se desea pasar a la escena de la válvula y se realizarán varias acciones:

1. Se traslada al avatar justo en frente de la válvula y se impide su movimiento.
2. La rotación de la cámara se bloquea de forma que la vista quede enfocada de frente a la válvula.
3. Se inicia un controlador que regula la posición del efector final en las coordenadas $(0, 0, 0)$ para posición y $(0, 0, \gamma)$ para rotación. En otras palabras, únicamente se le permite al efector final girar en torno a su propio eje. Se escogieron estas referencias porque corresponden con la posición de *home* del robot y se desea que permanezca ahí mientras la escena esté en curso.
4. Se inicia un controlador que se opone al giro del efector final descrito en el punto anterior para simular fricción viscosa en la válvula.

La escena se ve como en la Figura 4.19. El usuario podrá girar el efector final en torno al último grado de libertad, lo cual en cambio rotará a la válvula en el ambiente virtual. Además se pueden simular diferentes grados de fricción viscosa, presentando menor a mayor resistencia al movimiento, presionando las teclas "0", "1", "2", "3" y "4" en el teclado numérico. La ecuación que rige este comportamiento es

$$\boldsymbol{u_{fv}} = \boldsymbol{K_{fv}}\dot{\boldsymbol{q}} \tag{4.8}$$

donde $\mathbf{K_{fv}}$ es una matriz diagonal positiva semidefinida que representa los coeficientes de fricción viscosa en cada articulación, que en esta escena únicamente es diferente de cero para la última:

$$\boldsymbol{K_{fv}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & K_{fv6} \end{bmatrix}$$

El controlador de posición calcula una segunda entrada para el sistema dada por un PID de ecuación

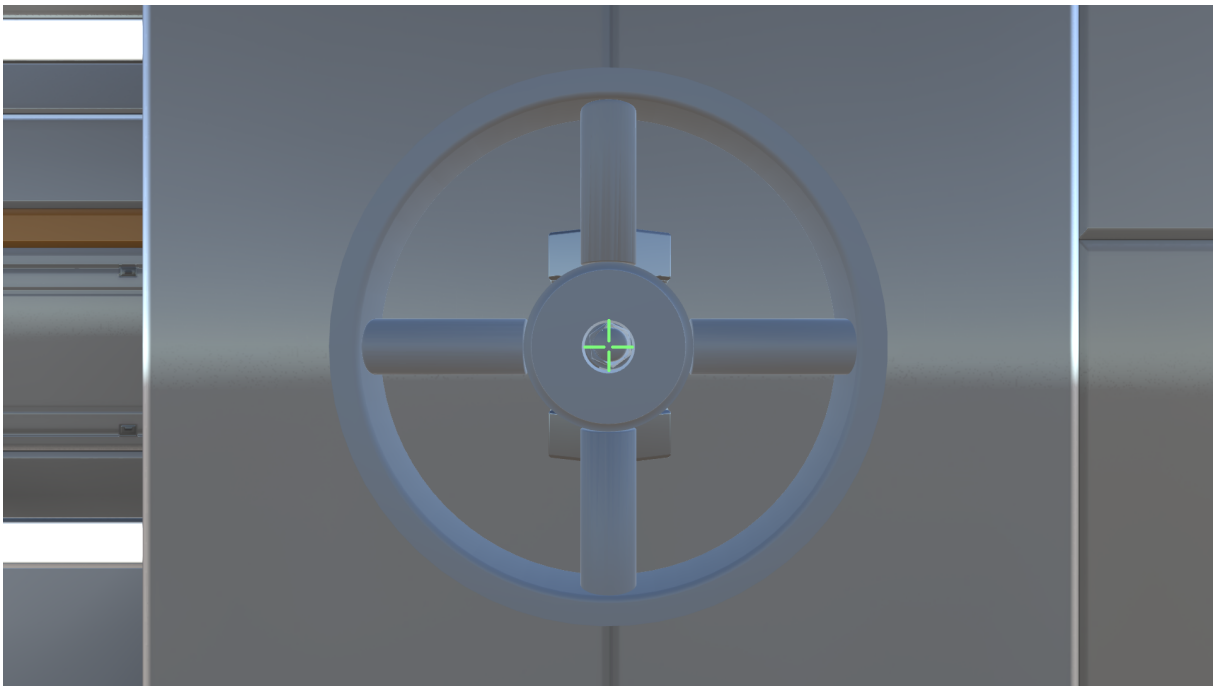


Figura 4.19: Escena de la válvula.

$$\boldsymbol{u_{pos}} = \boldsymbol{K_p}\tilde{\boldsymbol{q}} + \boldsymbol{K_d}\dot{\tilde{\boldsymbol{q}}} + \boldsymbol{K_i}\int_0^t \tilde{\boldsymbol{q}}\, dt \tag{4.9}$$

en donde $K_{p6} = K_{d6} = K_{i6} = 0$ porque no se regula posición en la última articulación. La señal de entrada total está dada por

$$\boldsymbol{u} = \boldsymbol{u_{fv}} + \boldsymbol{u_{pos}} \tag{4.10}$$

La sintonización del controlador es más complicada para el robot Falcon que para el Touch, lo cual se debe a que el modelo matemático del primero es más complejo. En (Rodríguez, 2017) se estudia a fondo dicho modelo, y aquí se rescatan las partes necesarias para seguir el procedimiento:

$$\boldsymbol{H} = \underbrace{I_r + \frac{1}{3}m_a a^2 + m_b a^2}_{I_A} \begin{bmatrix} \ddot{\theta}_{11} \\ \ddot{\theta}_{12} \\ \ddot{\theta}_{13} \end{bmatrix} + (\boldsymbol{J}^T)^{-1} \underbrace{3m_b + m_c}_{m} \boldsymbol{J}^{-1} \tag{4.11}$$

$$\boldsymbol{g} = -ag(\frac{1}{2}m_a + m_b) \begin{bmatrix} \sin\varphi_1 \sin\theta_{11} \\ \sin\varphi_2 \sin\theta_{12} \\ \sin\varphi_3 \sin\theta_{13} \end{bmatrix} - (\boldsymbol{J}^T)^{-1} m \begin{bmatrix} 0 \\ -g \\ 0 \end{bmatrix} \tag{4.12}$$

donde:

- I_r es el momento de inercia del rotor.
- m_a es la masa del eslabón de entrada.
- m_b es la masa de una biela del paralelogramo.
- a es la longitud del eslabón de entrada.
- m_c es la masa de la plataforma móvil.
- $(\varphi_1, \varphi_2, \varphi_3) = (105, 345, 225)[^\circ]$
- g es la constante de gravedad.

Tanto el vector de gravedad como la matriz de inercia involucran a la matriz Jacobiana, la cual introduce aún más variables que las tres que aparecen explícitamente en las ecuaciones (4.11) y (4.12). Ambos dependen de funciones no lineales de varias variables, por lo que realizar los cálculos que requiere el procedimiento de sintonización propuesto se vuelve una tarea extremadamente compleja. Es por ello que no se empleó tal método y las ganancias se ajustaron experimentalmente.

Para lograr el control es necesario comunicarse con el robot y la muñeca esférica. En el caso del primero, se usan las funciones predefinidas en la librería dinámica provista por Novint. Para el segundo caso, las instrucciones se envían mediante un puerto de comunicación serial. Unity envía una cadena de tres números flotantes separados con comas con la forma:

$$u_4, u_5, u_6$$

donde u_i representa el valor de la entrada que se debe ingresar a cada actuador de la muñeca esférica con $|u_i| \leq 12$, que es el voltaje nominal de los motores Namiki. El signo de u_i determina el sentido de giro de la articulación i. En respuesta, el Arduino envía a Unity una segunda cadena de números flotantes con el formato:

$$q_4, q_5, q_6$$

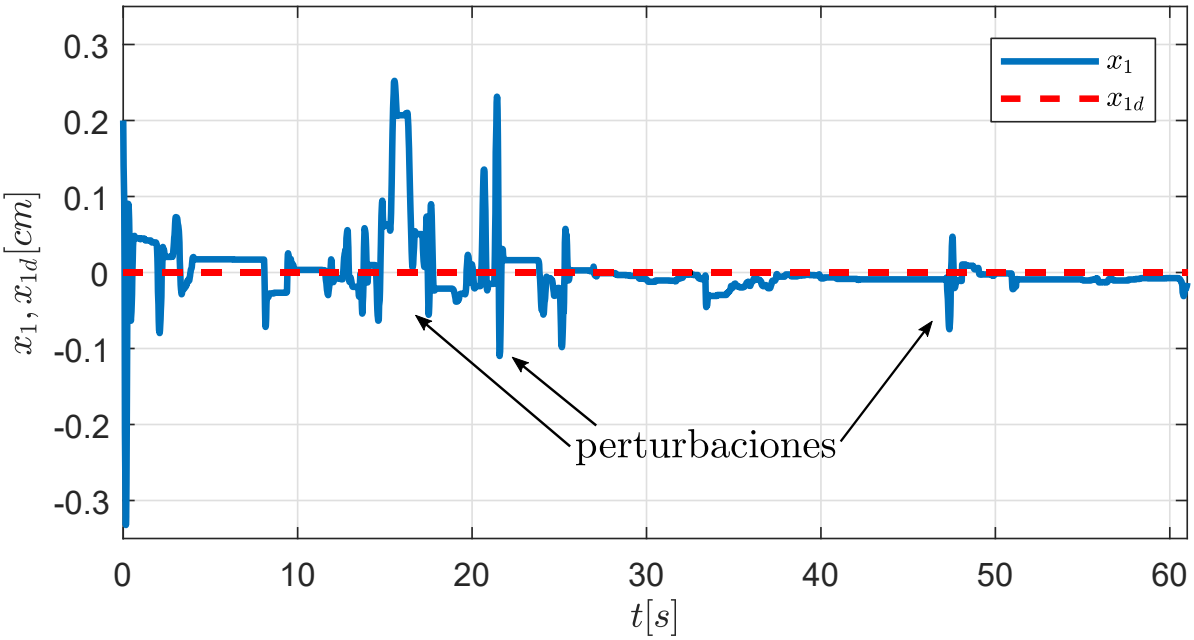
que corresponde con el valor de las variables articulares q_i de la muñeca esférica. Este proceso se repite cada que hay un ciclo de actualización, por lo que todo se traduce en una experiencia interactiva. Finalmente, el usuario puede regresar al modo de navegación al presionar la tecla “Esc”.

4.4. Resultados

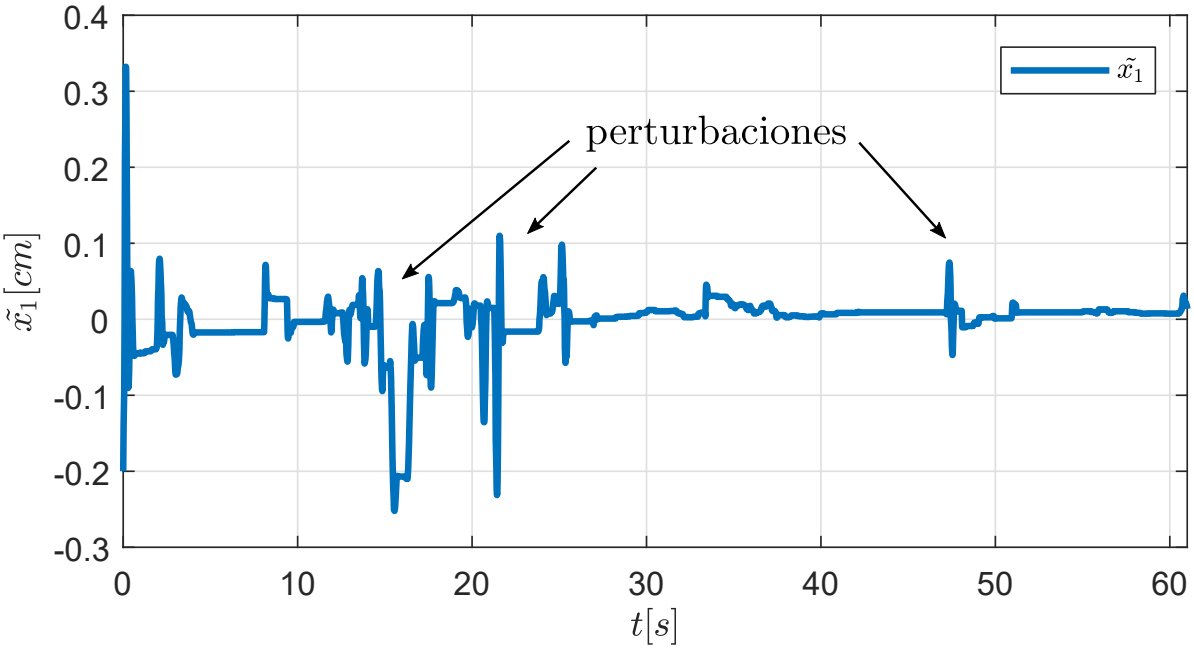
En esta sección se presentan algunas gráficas de la escena de la válvula, es decir, representan el comportamiento del robot una vez que se ha iniciado el controlador de

posición y el que simula fricción viscosa. Para la navegación no son relevantes las gráficas ya que no se hace ningún tipo de control sino únicamente lecturas. En las Figuras 4.20 a 4.25 se presentan las gráficas. El inciso (a) de cada Figura muestra la evolución de las señales de referencia y real durante el experimento. El inciso (b) de cada Figura es la señal de error. Todas las señales de referencia son cero porque se regula al robot a la posición de *home*. Cabe señalar que no se incluyó gráfica de error para la última articulación de la muñeca esférica porque no sigue una referencia de posición sino que únicamente simula fricción viscosa.

Se puede ver que en ciertos instantes de tiempo se añadieron perturbaciones al sistema empujando al mecanismo, los cuales se han señalado en cada gráfica. Un ejemplo de ello es en la Figura 4.20 alrededor de $t = 16$ y $t = 22$, en donde se pueden ver cambios abruptos que alejan a la articulación de la referencia, pero fue capaz de recuperarse y acercar el error nuevamente a cero. Por ello es posible concluir que el controlador cumple con su función adecuadamente.

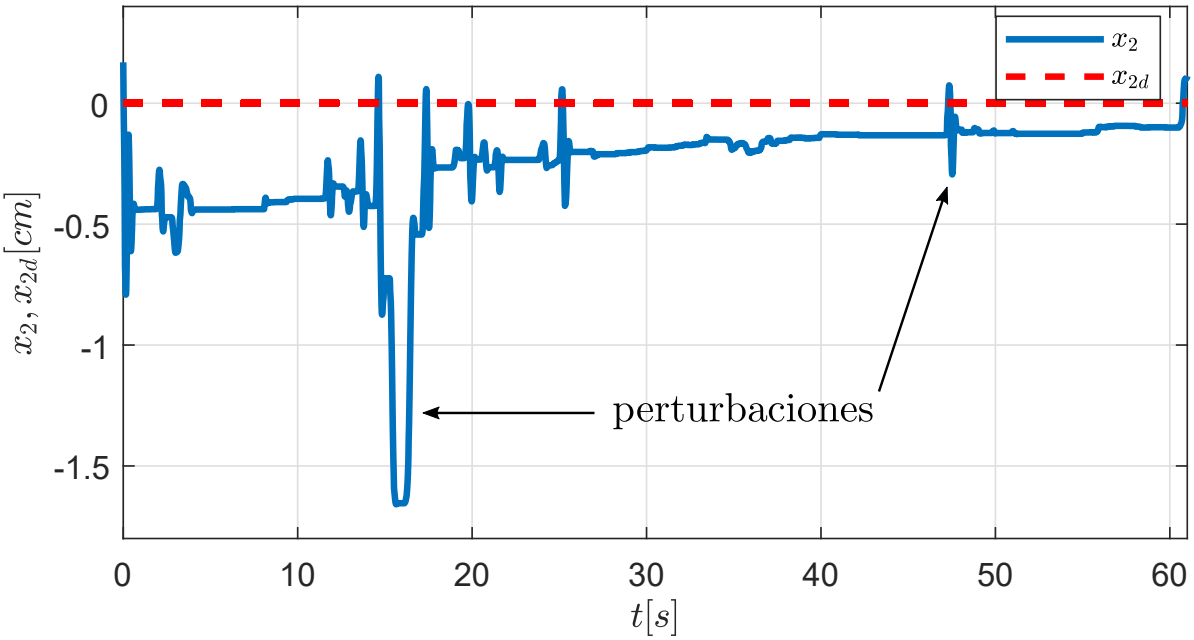


(a) Señal de referencia y real.

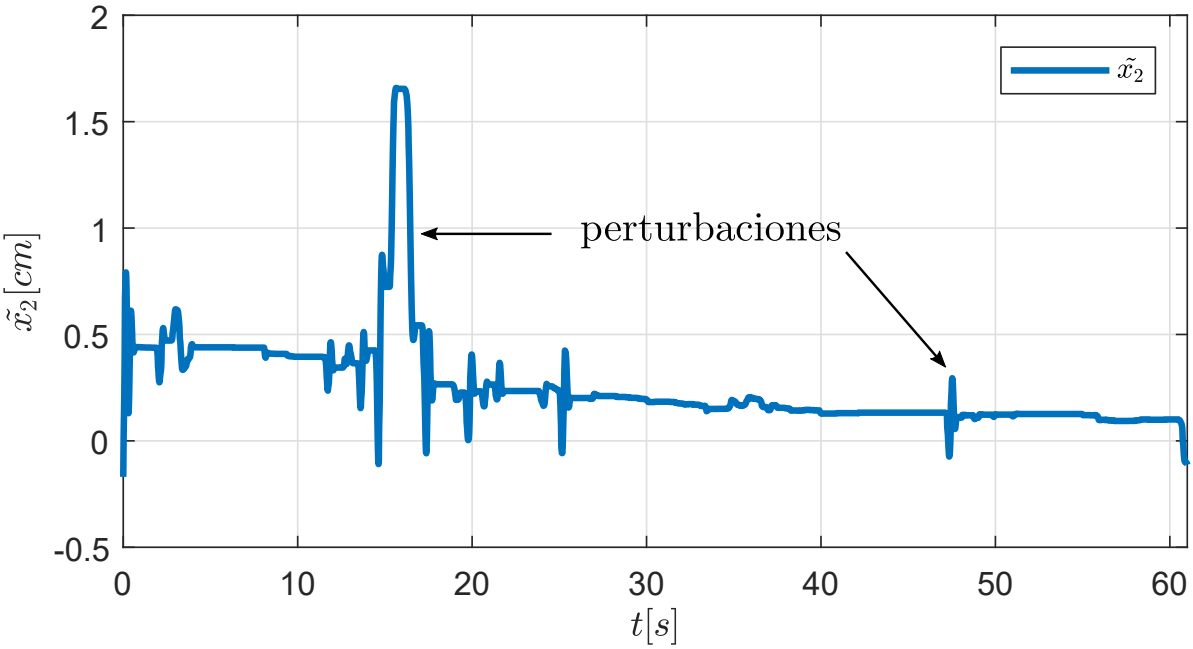


(b) Error de seguimiento.

Figura 4.20: Gráficas de posición x.

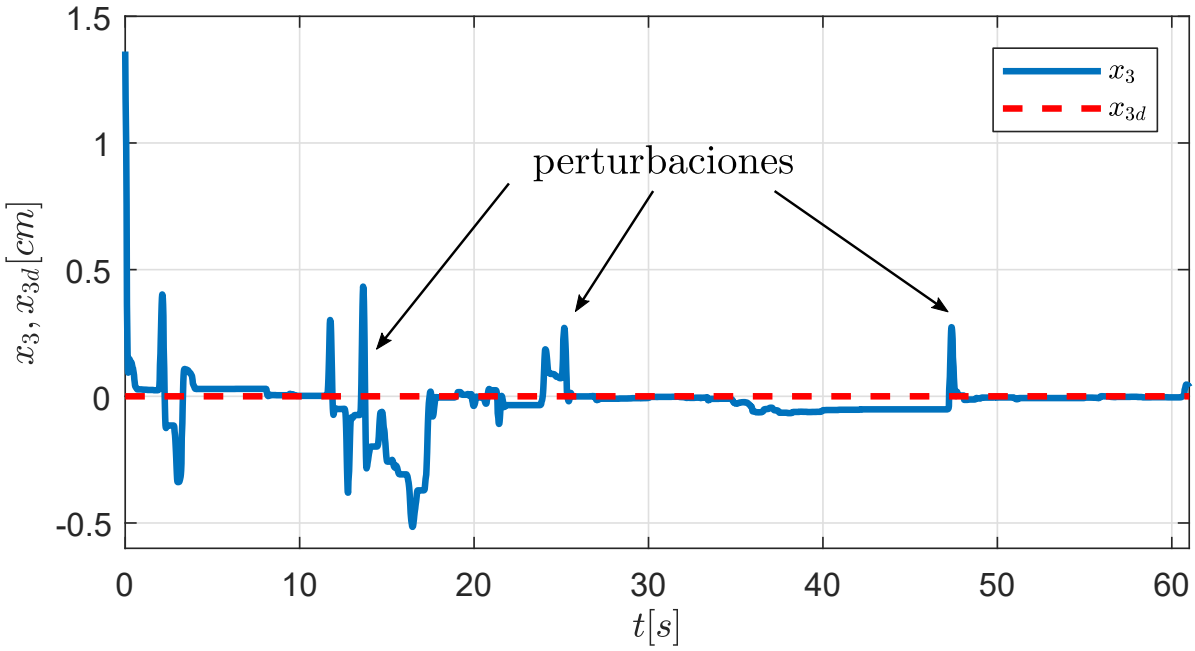


(a) Señal de referencia y real.

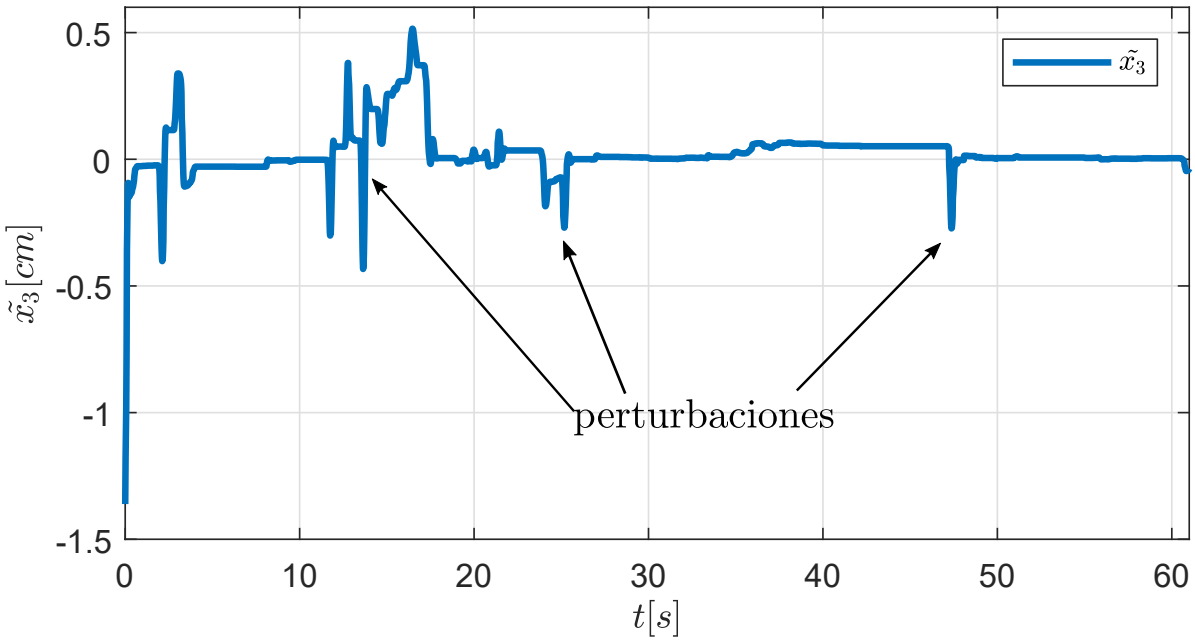


(b) Error de seguimiento.

Figura 4.21: Gráficas de posición y.

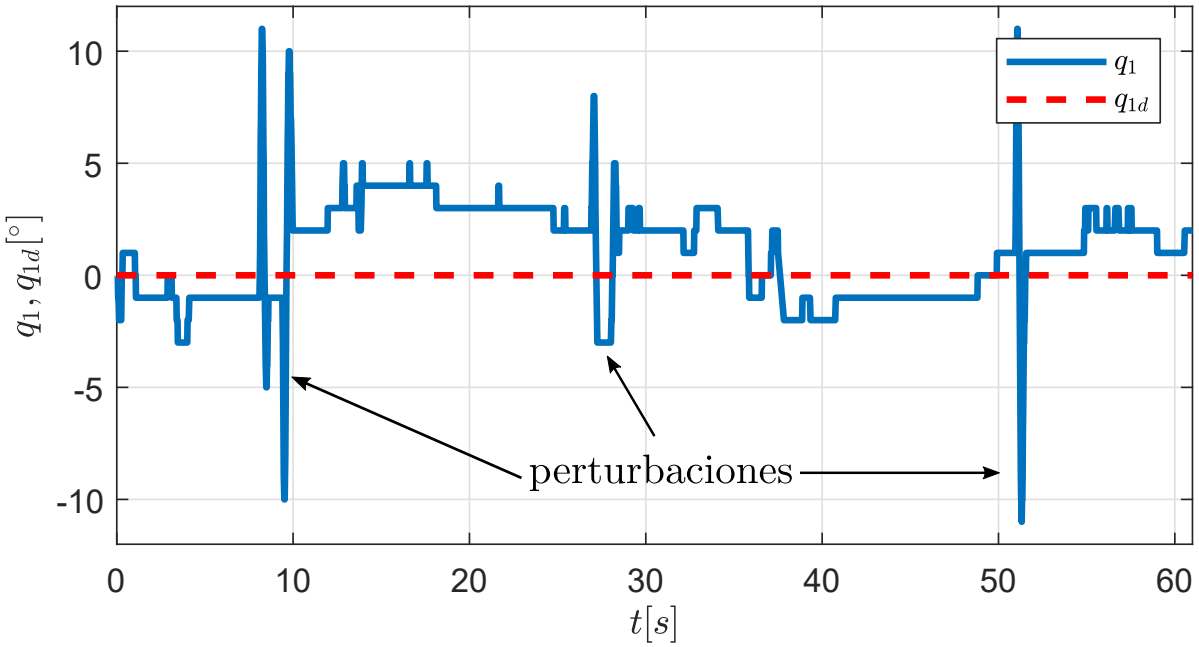


(a) Señal de referencia y real.

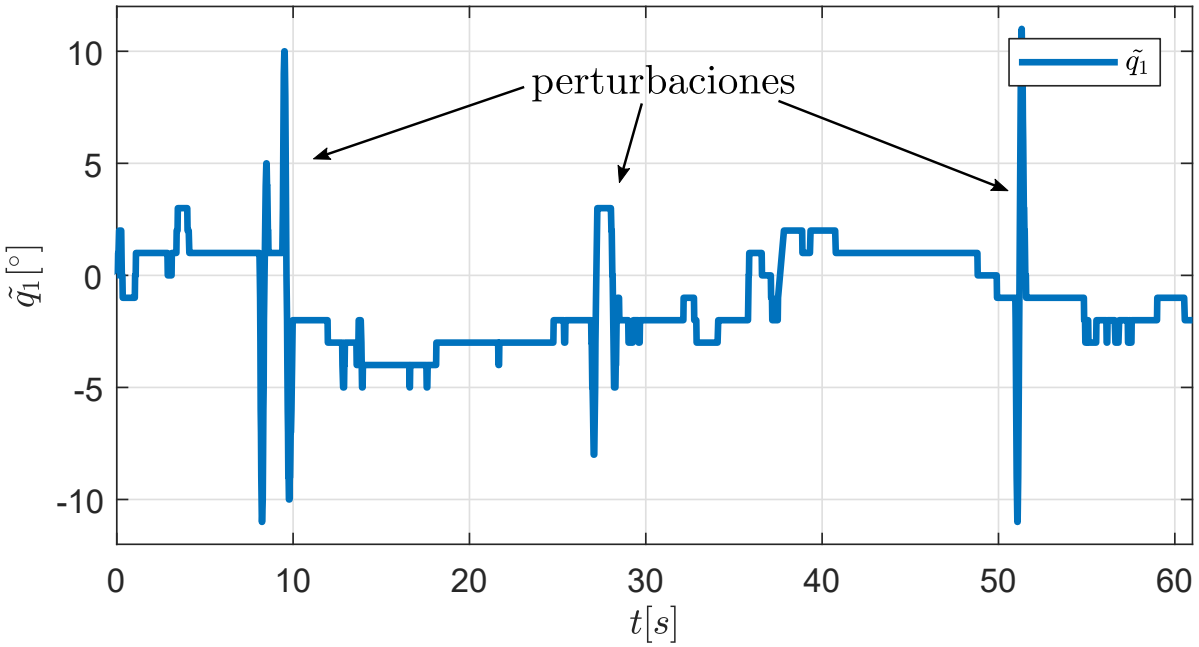


(b) Error de seguimiento.

Figura 4.22: Gráficas de posición z.

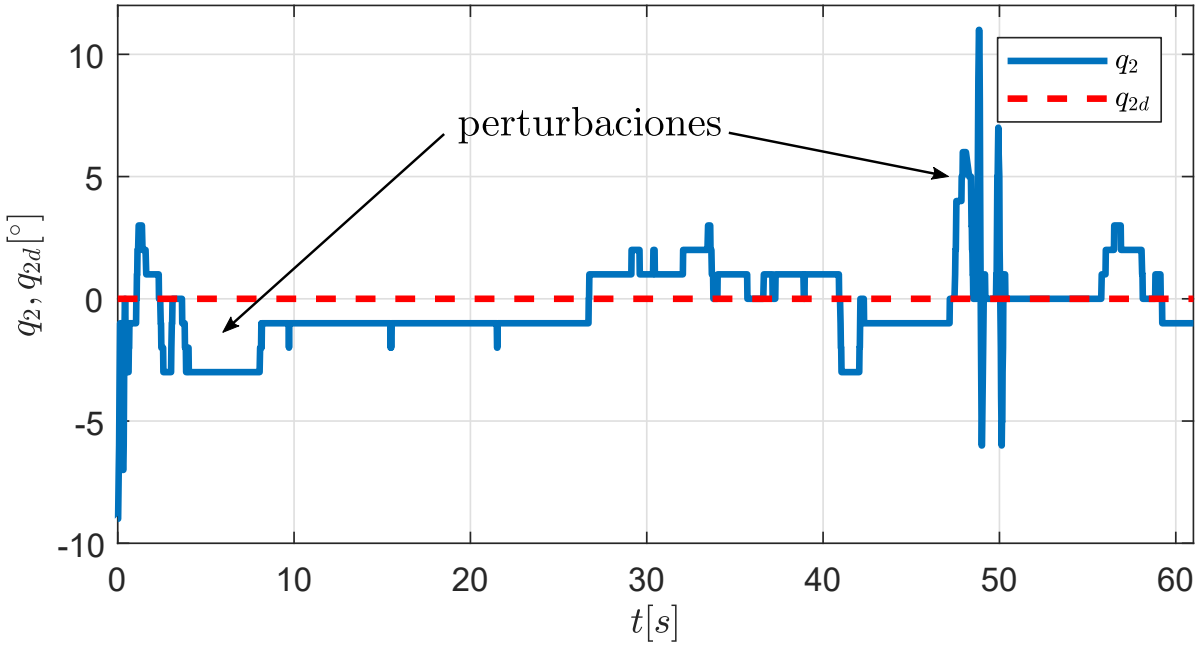


(a) Señal de referencia y real.

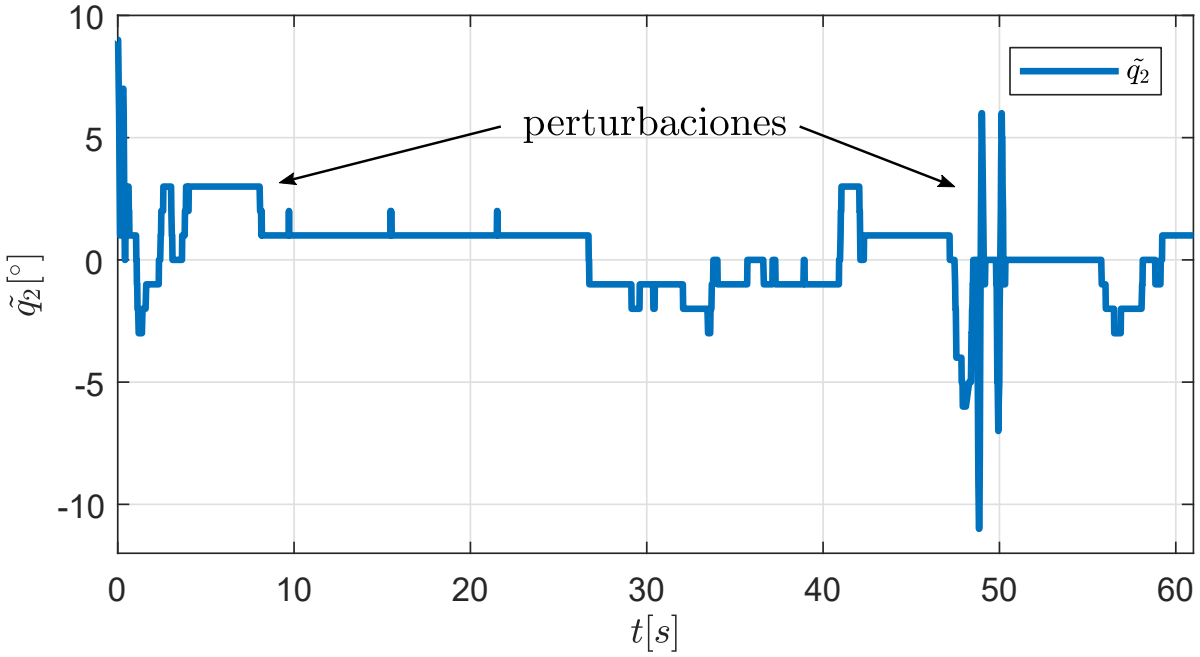


(b) Error de seguimiento.

Figura 4.23: Gráficas de la primer articulación de la muñeca.



(a) Señal de referencia y real.



(b) Error de seguimiento.

Figura 4.24: Gráficas de la segunda articulación de la muñeca.

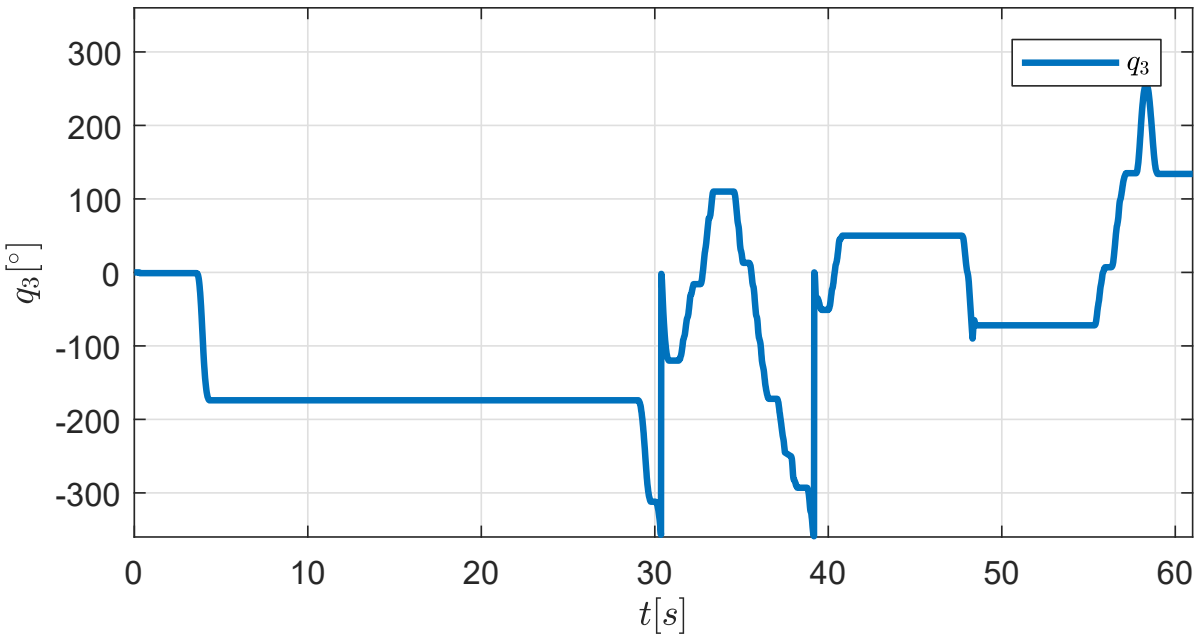


Figura 4.25: Gráfica de la tercer articulación de la muñeca.

Capítulo 5

Conclusiones

Con el trabajo hecho en los robots fue posible alcanzar los objetivos planteados al inicio de la tesis. En primer lugar gracias a la adaptación del sensor que se hizo al robot Touch, será posible realizar experimentos de control de fuerza en el Laboratorio de Robótica. Esto a su vez permitirá desarrollar aplicaciones hápticas con mayor precisión en el regulamiento de las fuerzas desplegadas en el robot. En cuanto al robot Falcon, se logró obtener el prototipo de seis grados de libertad que se deseaba. Esto permitirá desarrollar aplicaciones de realidad virtual con interacciones usuario-ambiente más complejas y realistas. Mediante la documentación generada, cualquier estudiante o investigador que haga uso de este prototipo podrá entender a fondo su funcionamiento e incluso hacer modificaciones si así se requiriera.

El mecanismo desarrollado es de bajo costo y simple, lo cual cumple otro de los objetivos del trabajo, pero es importante recordar que se trata de un primer prototipo, por lo que es posible pensar en mejoras que se le pueden hacer en el futuro. En primer lugar se deben diseñar rutas para guiar al cableado, ya que actualmente se encuentra suelto y pudiera llegar a enredarse además de que no es estético a la vista. Otra posible mejora es el uso de un material más resistente, ya que está fabricado con MDF de 3 milímetros, el cual es blando y con el tiempo y uso podría fallar. Una mejora más podría ser la modificación del mecanismo para poder hacer uso del efector final con el que cuenta el robot, ya que tiene la ventaja de contar con cuatro botones, los cuales pueden ser útiles para lograr interacciones en un ambiente virtual. Se requeriría cableado adicional pero se expandirían las opciones de interacción con las que se cuentan actualmente.

La aplicación desarrollada en Unity, aunque sencilla, establece una base sobre la cual se puede aprender las nociones básicas del uso del robot Falcon, y se espera que basándose en las ideas que en ella se presentan se puedan desarrollar en el aplicaciones hápticas cada vez de mejor calidad con un tiempo de desarrollo y curva de aprendizaje inferiores. El uso de un motor para videojuegos permite simplificar el desarrollo en áreas que no son de interés primordial para el trabajo que se realiza en el Laboratorio de Robótica y enfocar la atención a las que sí lo son, como la implementación de controladores avanzados y una

realimentación visual y háptica más realista.

5.1. Trabajo a futuro

Ahora que el robot Touch cuenta con un sensor de fuerza montado en el efector final, algunas de las tareas que se pueden realizar a futuro son:

- Implementación de controladores directos de fuerza más sofisticados.
- Estudio de las ventajas del control de fuerza directo en aplicaciones hápticas.
- Sustitución del sensor de fuerza por uno que pueda medir en más ejes, como por ejemplo el sensor ATI Nano 17 (que es de seis) de reciente adquisición en el Laboratorio de Robótica.

En cuanto a la segunda parte del trabajo realizado, ya que se cuenta con un prototipo de un robot paralelo con seis grados de libertad, será posible explorar las capacidades y limitaciones de la plataforma para poder hacer mejoras en cualquiera de las áreas involucradas: programación, mecánica o electrónica. Algunas de las cuestiones que quedan por hacer son:

- Análisis matemático y obtención de los modelos cinemático y dinámico del mecanismo de seis grados de libertad resultante de este trabajo.
- Uso y pruebas del mecanismo en los simuladores de cirugía desarrollados en el Laboratorio de Análisis de Imagenología Biomédica del Instituto de Ciencias Aplicadas y Tecnología con el que el Laboratorio de Robótica tiene colaboración.
- Desarrollar una aplicación de realidad virtual más avanzada que muestre mejor las ventajas de los seis grados de libertad.
- Adaptación al mecanismo de un sensor de fuerza ATI Nano 17 de seis ejes para poder hacer control de fuerza directo.

Cabe señalar que el dispositivo desarrollado no se limita únicamente al área académica sino que es posible continuar su desarrollo y en algún punto lograr un producto completo que pueda competir en el mercado. Para ello se requeriría generar un mecanismo para los primeros tres grados de libertad y afinar los defectos que se identifiquen en este primer prototipo. Se espera que este trabajo provea las bases para poder lograr tales objetivos.

Bibliografía

Bicchi, A. (2000). Hands for dexterous manipulation and robust grasping: A difficult road toward simplicity. *IEEE Transactions on robotics and automation*, 16(6):652–662.

Castillo, J. A. V. (2015). Display háptico de sistemas dinámicos virtuales y teleoperados sujetos a restricciones holonómicas. Tesis de licenciatura, UNAM.

de Llergo, C. G. L. (2014). Control de fuerza sobre superficies virtuales. Tesis de maestría, UNAM.

Hsu, C.-H., Chang, C.-C., and Hsu, J.-J. (1999). Structural synthesis of bevel-gear robotic wrist mechanisms. *PROCEEDINGS-NATIONAL SCIENCE COUNCIL REPUBLIC OF CHINA PART A PHYSICAL SCIENCE AND ENGINEERING*, 23:518–525.

Kelly, R., Davila, V. S., and Perez, J. A. L. (2006). *Control of robot manipulators in joint space*. Springer Science & Business Media.

Lange, B., Flynn, S., and Rizzo, A. (2009). Initial usability assessment of off-the-shelf video game consoles for clinical game-based motor rehabilitation. *Physical Therapy Reviews*, 14(5):355–363.

Linda, O. and Manic, M. (2011). Fuzzy force-feedback augmentation for manual control of multirobot system. *IEEE Transactions on Industrial Electronics*, 58(8):3213–3220.

Luciano, C., Banerjee, P., and DeFanti, T. (2009). Haptics-based virtual reality periodontal training simulator. *Virtual reality*, 13(2):69–85.

Mihelj, M. and Podobnik, J. (2012). *Haptics for virtual reality and teleoperation*, volume 67. Springer Science & Business Media.

Norton, R. L. (2016). *Diseño de maquinaria*.

Pandilov, Z. and Dukovski, V. (2014). Comparison of the characteristics between serial and parallel robots. *Acta Technica Corviniensis-Bulletin of Engineering*, 7(1):143.

Rodríguez, I. T. (2017). Puesta en funcionamiento del robot paralelo novint falcon. Tesis de licenciatura, UNAM.

Rodríguez, M. G. L. (2014). Puesta en marcha del robot omni phantom de sensable. Tesis de licenciatura, UNAM.

Rothbaum, B. O., Hodges, L. F., Kooper, R., Opdyke, D., et al. (1995). Effectiveness of computer-generated (virtual reality) graded exposure in the treatment of acrophobia. *The American journal of psychiatry*, 152(4):626.

Sánchez, J. and Tadres, A. (2010). Audio and haptic based virtual environments for orientation and mobility in people who are blind. In *Proceedings of the 12th international ACM SIGACCESS conference on Computers and accessibility*, pages 237–238. ACM.

Sherman, W. R. and Craig, A. B. (2002). *Understanding virtual reality: Interface, application, and design*. Elsevier.

Siciliano, B., Sciavicco, L., Villani, L., and Oriolo, G. (2009). Robotics–modelling, planning and control. advanced textbooks in control and signal processing series.

Spong, M. W., Hutchinson, S., and Vidyasagar, M. (2006). *Robot modeling and control*, volume 3. Wiley New York.

Tsai, L.-W. (1999). *Robot analysis: the mechanics of serial and parallel manipulators*. John Wiley & Sons.

Tsai, L.-W. (2000). *Mechanism design: enumeration of kinematic structures according to function*. CRC press.

Apéndice A

Hoja de datos del sensor

FUTEK
ADVANCED SENSOR TECHNOLOGY, INC.

MODEL LLB300

Miniature Load Button

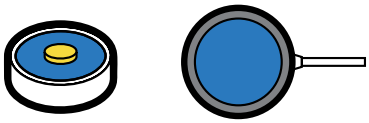
FEATURES

- Ultra fast response
- Robust strain relief
- Low deflection
- 17-4 stainless-steel construction
- For use in compression
- Adheres to RoHS directive 2011/65/EU

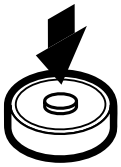
Non-loading surface, do not contact

Active End

Fixed End



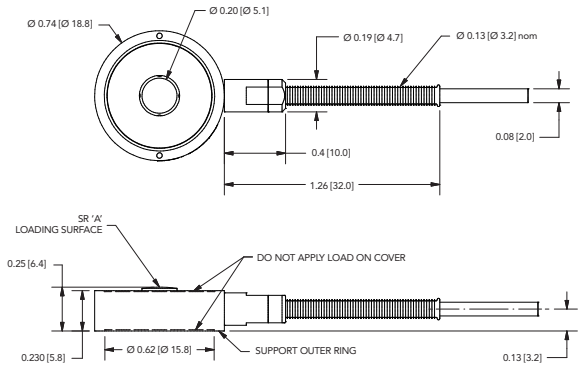
Top view Bottom view



+ Output (compression)

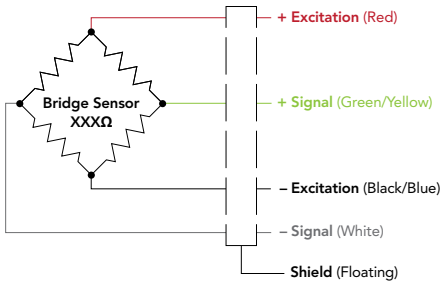
SPECIFICATIONS	
PERFORMANCE	
Nonlinearity	±0.2% of RO
Hysteresis	±0.2% of RO
Nonrepeatability	±0.1% of RO
ELECTRICAL	
Rated Output (RO)	2.5 mV/V (1000 lb) 1 mV/V (50, 250 lb) 2 mV/V (25, 100, 500 lb)
Excitation (VDC or VAC)	18 max
Bridge Resistance	700 Ohm nom
Insulation Resistance	≥500 MOhm @ 50 VDC
Connection	#29 AWG, 4 conductor, spiral shielded Teflon cable 10-ft [3 m] long
Wiring/Connector Code	WC1
MECHANICAL	
Weight	1.2 oz [34 g]
Safe Overload	150% of RO
Deflection	See chart on next page
Material	17-4 PH stainless-steel
IP Rating	IP64
TEMPERATURE	
Operating Temperature	–60 to 250°F (–50 to 121°C)
Compensated Temperature	60 to 250°F (15 to 121°C)
Temperature Shift Zero	±0.01% of RO/°F (±0.018% of RO/°C)
Temperature Shift Span	±0.02% of Load/°F (±0.036% of Load/°C)
CALIBRATION	
Calibration Test Excitation	10 VDC
Calibration (std)	5-pt Compression
Shunt Calibration Value	100 kOhm
CONFORMITY	
RoHS	2011/65/EU
CE	EN55011:2009; EN61326-1:2006

DIMENSIONS inches [mm]



WIRING CODE (WC1 WITH SHIELD)

RED	+ EXCITATION
BLACK	– EXCITATION
GREEN	+ SIGNAL
WHITE	– SIGNAL
SHIELD	FLOATING



CAPACITIES

ITEM #	lb	N	Deflection (in)	Natural Frequency (kHz)
FSH03939	25	111	0.0007	21
FSH03954	50	222	0.0003	34
FSH03955	100	445	0.0007	34
FSH03938	250	1112	0.0004	52
FSH03953	500	2224	0.0007	52
FSH03937	1000	4448	0.001	58

Drawing Number: FI1119-F

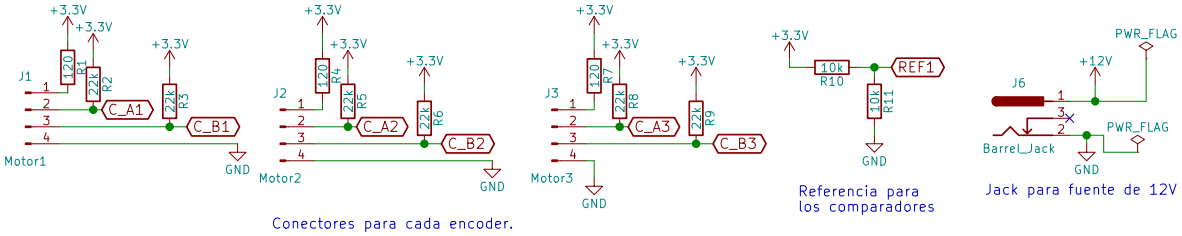
10 Thomas, Irvine, CA 92618 USA
Tel: (949) 465-0900
Fax: (949) 465-0905

www.futek.com

CE RoHS ANSI Z540-1 ISO 17025 ISO 9001 ISO 13485 U.S. Manufacturer

Apéndice B

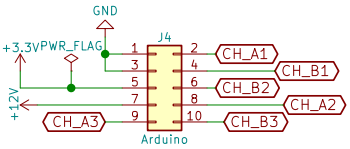
Diagrama eléctrico de la muñeca



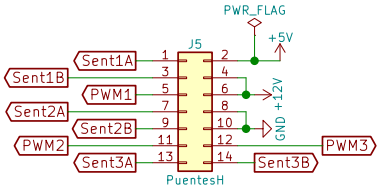
Conectores para cada encoder.

Referencia para los comparadores

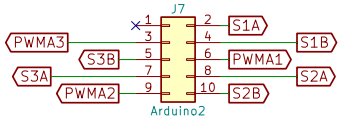
Jack para fuente de 12V



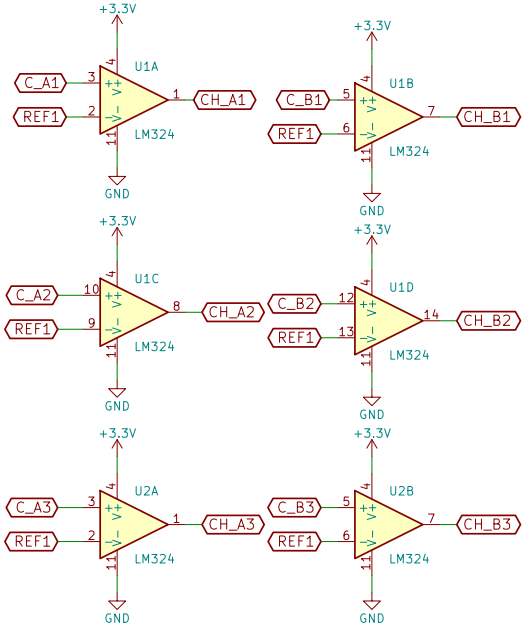
Conector para entradas del Arduino



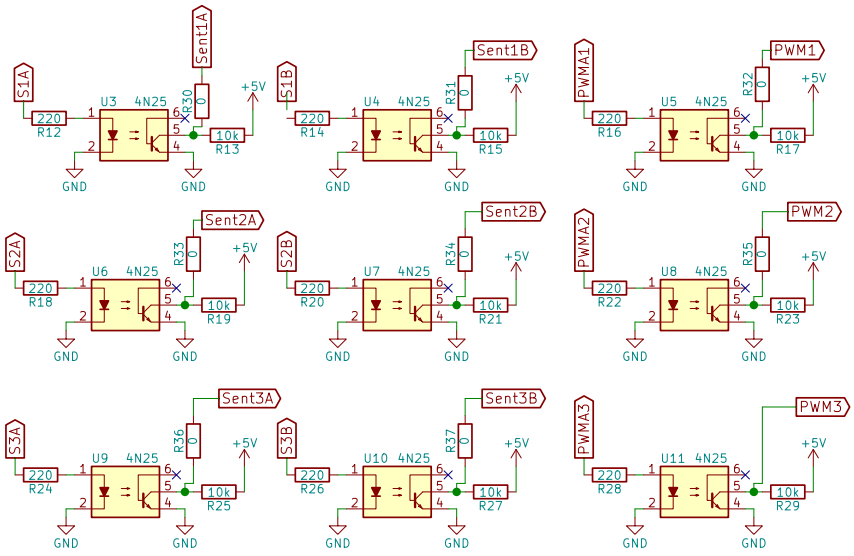
Conector para señales de los puentes H



Conector para salidas del Arduino



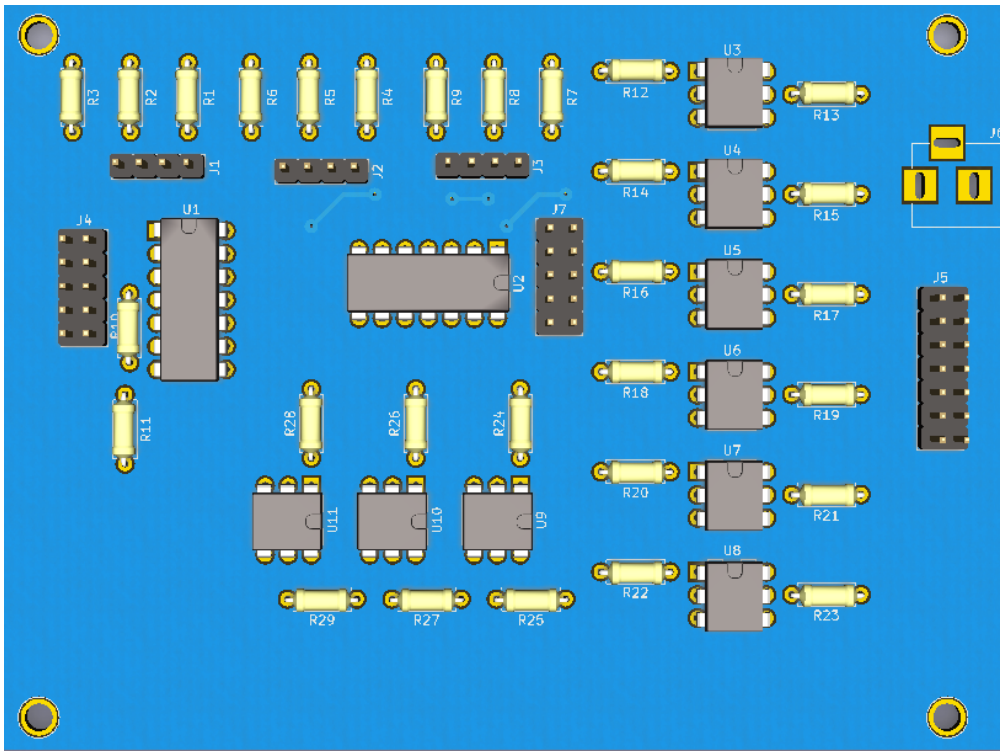
Comparadores

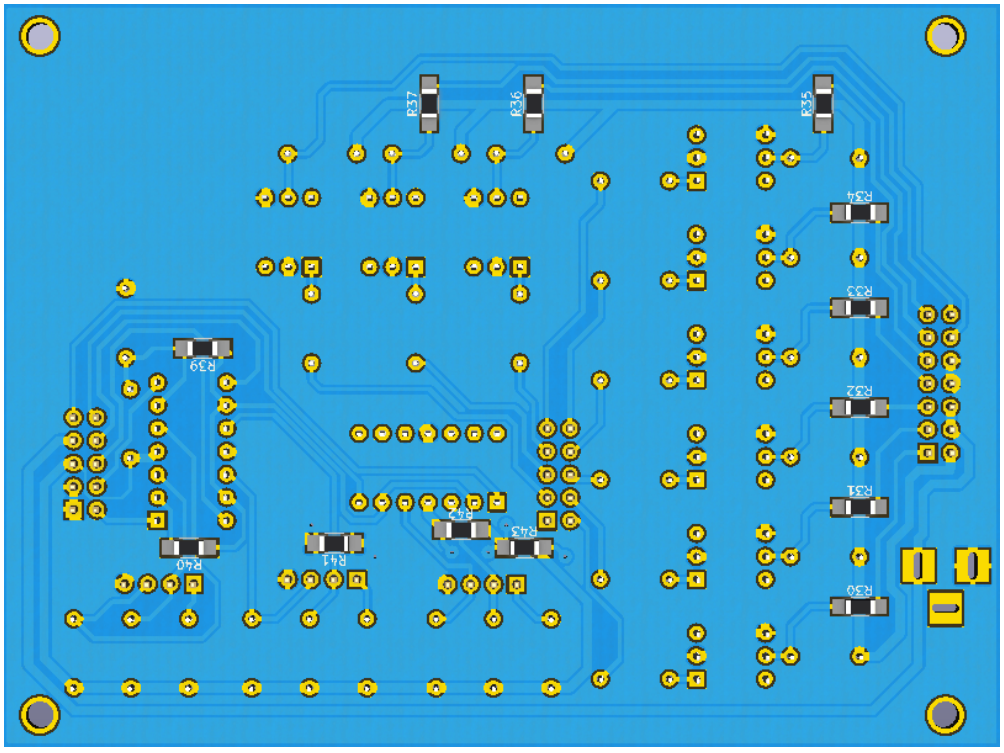


Optoacopladores para cada señal de los puentes H

Apéndice C

Fotografías del circuito impreso


R3
R2
R1
R6
R5
R4
R9
R8
R7
J1
J2
J3
J4
U1
U2
J7
R10
R11
R28
R26
R24
U11
U10
U9
R29
R27
R25
U3
R12
R13
U4
R14
R15
U5
R16
R17
U6
R18
R19
U7
R20
R21
U8
R22
R23
J5


R37
R36
R35
R34
R33
R32
R31
R30
R39
R40
R41
R42
R43