



**Universidad Autónoma de San Luis Potosí**  
**Facultad de Ingeniería**  
**Centro de Investigación y Estudios de Posgrado**

**Sistema de teleoperación sobre superficies  
arbitrarias con visión no calibrada**

**T E S I S**

Que para obtener el grado de:

Maestro en Ingeniería Eléctrica

Presenta:

Ing. Alejandro Ramírez Quirino

Asesor:

Dr. Emilio Jorge González Galván

San Luis Potosí, S. L. P.

Febrero 2018







**UASLP**

Universidad Autónoma  
de San Luis Potosí

26 de octubre de 2017

**ING. ALEJANDRO RAMÍREZ QUIRINO  
P R E S E N T E.**

En atención a su solicitud de Temario, presentada por el **Dr. Emilio Jorge González Galván**, Asesor de la Tesis que desarrollará Usted con el objeto de obtener el Grado de **Maestro en Ingeniería Eléctrica**, me es grato comunicarle que en la Sesión del H. Consejo Técnico Consultivo celebrada el día 26 de octubre del presente, fue aprobado el Temario propuesto:

**TEMARIO:**

**"Sistema de Teleoperación sobre superficies arbitrarias con visión no calibrada"**

Introducción.

1. Fundamentos de teleoperación.
2. Método de reconstrucción de superficies arbitrarias.
3. Implementación del sistema de teleoperación.
4. Resultados experimentales.

Conclusiones.

Anexos.

Bibliografía.

**"MODOS ET CUNCTARUM RERUM MENSURAS AUDEBO"**

**A T E N T A M E N T E**

  
**M. I. JORGE ALBERTO PÉREZ GONZÁLEZ**  
**DIRECTOR.** DE SAN LUIS POTOSÍ  
FACULTAD DE INGENIERÍA  
DIRECCION



**FACULTAD DE  
INGENIERÍA**

Av. Manuel Nava 8  
Zona Universitaria - CP 78290  
San Luis Potosí, S.L.P.  
tel. (444) 826 2330 al39  
fax (444) 826 2336  
www.uasp.mx

Copia. Archivo.  
\*etn.



**UNIVERSIDAD AUTÓNOMA DE SAN LUIS POTOSÍ**  
**FACULTAD DE INGENIERÍA**  
**CENTRO DE INVESTIGACIÓN Y ESTUDIOS DE POSGRADO**  
**MAESTRÍA EN INGENIERÍA ELÉCTRICA**

Tesis:

**“Sistema de teleoperación sobre superficies arbitrarias con visión no calibrada”**

Presenta:

---

Ing. Alejandro Ramírez Quirino

Sinodales:

---

Dr. Emilio Jorge González Galván

Asesor

---

Dr. Mauro Eduardo Maya Méndez

Revisor

---

Dr. Juan Segundo Ramírez

Revisor

---

Dr. Jorge Alberto Morales Saldaña

Suplente



## *Agradecimientos*

*A mis padres, por todo el apoyo que me han dado desde que comencé mis estudios, por siempre estar ahí cuando más los necesito y creer en mí en todo momento.*

*A mis hermanos por compartir su vida conmigo.*

*A mi asesor Dr. Emilio Jorge González, por el gran apoyo recibido desde la licenciatura y la oportunidad de formar parte su investigación. Por ser tan paciente, compartirme su tiempo y conocimiento en todo momento. Gracias por sus consejos y facilitarme el equipo necesario para el desarrollo de este trabajo.*

*A mis amigos y compañeros de vida, Tony, Marcelleño, Gustavo, y Juan José, por su amistad, confianza y apoyo.*

*A mis compañeros de generación, en especial a Ramón, David, Alfredo y Julio, que me ofrecieron su amistad y me apoyaron en todo momento. Por los grandes momentos que pasamos juntos durante esta etapa y hacerla más llevadera.*

*A mis compañeros del Laboratorio de Robótica, por la ayuda recibida y aceptarme de la mejor manera como parte del equipo de trabajo. Especialmente a Ubaldo y Jorge que estuvieron siempre para brindarme su apoyo y conocimientos.*

*Al Maestro Luis Murillo, Doctor Mauro Maya, Ingeniero Efraín y al Doctor Antonio Cárdenas, por la oportunidad de trabajar con ellos cuando no tenía beca y por sus consejos.*

*A los maestros y administrativos del CIEP, por las clases y soporte brindado, porque sin ellos esto no habría sido posible.*

*A CONACyT por la beca otorgada para realizar mis estudios de maestría.*





# Índice general

---

<b>Índice de figuras</b>	<b>v</b>
<b>Índice de tablas</b>	<b>vii</b>
<b>Introducción</b>	<b>1</b>
<b>1. Fundamentos de teleoperación</b>	<b>5</b>
1.1. Teleoperación . . . . .	5
1.1.1. Características de los sistemas de teleoperación . . . . .	7
1.2. Elementos de un sistema de teleoperación . . . . .	8
1.3. Teleoperación en la medicina . . . . .	10
1.4. Control de sistemas teleoperados . . . . .	12
1.4.1. Control unilateral . . . . .	13
1.4.2. Control bilateral . . . . .	14
1.4.3. Control de impedancia . . . . .	14
<b>2. Método de reconstrucción de superficies arbitrarias</b>	<b>17</b>
2.1. Visión por computadora . . . . .	17
2.2. Coordenadas homogéneas y transformaciones elementales . . . . .	19
2.3. Modelo de cámara . . . . .	20
2.3.1. Proyección perspectiva . . . . .	20
2.3.2. Modelo ortográfico de cámara . . . . .	22
2.4. Método de manipulación de espacio de cámara . . . . .	23
2.4.1. Estimación no lineal por método de mínimos cuadrados . . . . .	24
2.4.2. Determinación de la posición física de un punto observado . . . . .	26
2.5. Correspondencia entre imágenes . . . . .	28
<b>3. Implementación del sistema de teleoperación</b>	<b>31</b>
3.1. Elementos del sistema de teleoperación . . . . .	31
3.2. Controlador de impedancia . . . . .	32
3.3. Asimetría cinemática . . . . .	35
3.3.1. Mapeo entre superficies de trabajo . . . . .	36
3.3.2. Movimiento perpendicular a la superficie estimada . . . . .	37

---

3.4. Cálculo de la orientación de la herramienta . . . . .	38
3.5. Integración de los elementos del sistema de teleoperación . . . . .	42
<b>4. Resultados experimentales</b>	<b>47</b>
4.1. Elementos del sistema de teleoperación . . . . .	47
4.2. Estimación de la superficie arbitraria . . . . .	48
4.2.1. Estimación de los parámetros de visión . . . . .	48
4.2.2. Prueba de correspondencia entre imágenes . . . . .	51
4.2.3. Pruebas de la metodología propuesta para asimetría cinemática	53
4.3. Pruebas del sistema de teleoperación . . . . .	57
4.3.1. Pruebas sobre un objeto plano . . . . .	59
4.3.2. Pruebas sobre una sección esférica . . . . .	63
4.3.3. Pruebas sobre casco . . . . .	65
<b>Conclusiones</b>	<b>69</b>
<b>A. Anexos</b>	<b>71</b>
A.1. Comunicación con las cámaras UEye . . . . .	71
A.2. Comunicación con el robot Fanuc . . . . .	73
A.3. Estimación de los parámetros de visión . . . . .	75
A.4. Estimación de la trayectoria del esclavo . . . . .	78
<b>Bibliografía</b>	<b>81</b>

# Índice de figuras

---

1.1. Raymond Goertz, a la izquierda con el mecanismo maestro esclavo M1, a la derecha con el E1, con accionamiento eléctrico y servocontrol, Argonne National Laboratory. . . . .	6
1.2. MASCOT, sistema teleoperado con servo manipuladores, Argonne National Laboratory. . . . .	6
1.3. Esquema típico de un sistema teleoperado dividido por zonas. . . . .	9
1.4. Sistema de telecirugía ZEUS <sup>®</sup> Surgical System. . . . .	11
1.5. Sistema de telecirugía DaVinci. . . . .	12
1.6. Sistema teleoperado TER, . . . . .	12
1.7. Esquema típico de un sistema teleoperado con control unilateral. . . . .	13
2.1. Proyección perspectiva de una escena tridimensional en una bidimensional.	21
2.2. Puntos vistos por la cámara 1 y cámara 2 respectivamente. . . . .	29
2.3. Imágenes de la proyección de la matriz de puntos láser vistos por el sistema de visión. . . . .	30
2.4. Resultado de la correspondencia entre las coordenadas de los puntos objetivos detectados. . . . .	30
3.1. Diagrama del sistema de teleoperación. . . . .	32
3.2. Diagrama de bloques del control de impedancia del sistema de teleoperación. . . . .	34
3.3. Respuesta del controlador de impedancia en el eje X. . . . .	35
3.4. Espacios de trabajo típicos definidos en una relación maestro-esclavo. . . . .	36
3.5. Definición de los espacios de trabajo a partir de nueve puntos. . . . .	37
3.6. Sistemas coordenados de los espacios de trabajo. . . . .	38
3.7. Relación entre los sistemas coordenados. . . . .	39
3.8. Componentes del vector normal a la superficie estimada. . . . .	41
3.9. Diagrama de actividad del sistema de teleoperación. . . . .	43
3.10. Botón 1: inicia el mapeo de puntos del maestro al esclavo. . . . .	44
3.11. Diagrama de clases del sistema de teleoperación. . . . .	44
3.12. Ejecución del sistema de teleoperación. . . . .	45
4.1. Led montado en el efector final del robot Fanuc LR Mate 200iC. . . . .	49

---

4.2. Puntos muestra que definen el espacio de estimación paramétrica. . . . .	49
4.3. Resultado de la estimación de parámetros de visión en el espacio de cámara. . . . .	50
4.4. Comparación entre los puntos muestra y los puntos estimados por los parámetros calculados. . . . .	51
4.5. Correspondencia de puntos objetivo en cada una de las cámaras. . . . .	52
4.6. Estimación de las coordenadas $(X, Y, Z)$ de los puntos objetivo correspondidos. . . . .	52
4.7. Correspondencia de puntos objetivo en cada una de las cámaras. . . . .	53
4.8. Estimación de las coordenadas $(X, Y, Z)$ de los puntos objetivo correspondidos. . . . .	54
4.9. Puntos que definen los espacios de trabajo para la asimetría cinemática. . . . .	54
4.10. Comparativa entre los puntos que definen el plano esclavo y su estimación. . . . .	55
4.11. Estimación de una línea trazada en el plano maestro y trasladada al esclavo. . . . .	55
4.12. Comparación entre la trayectoria deseada a seguir por el esclavo y la estimada. . . . .	56
4.13. Error relativo de la estimación de la trayectoria en cada uno de los ejes. . . . .	56
4.14. Puntos que definen los espacios de trabajo para la sección cilíndrica. . . . .	57
4.15. Trayectorias descritas por el dispositivo maestro y esclavo. . . . .	58
4.16. Comparativa entre las trayectoria estimada por el método de asimetría y la real descrita por el robot. . . . .	58
4.17. Trayectorias propuestas para el seguimiento. . . . .	59
4.18. Trayectoria descrita por el dispositivo maestro y esclavo sobre un plano estimado. . . . .	60
4.19. Comparativa entre las trayectoria estimada y la real trazada sobre la superficie plana estimada. . . . .	61
4.20. Fuerzas percibidas por el sensor durante la interacción. . . . .	61
4.21. Trayectorias descritas por el maestro y el esclavo sobre un plano propuesto. . . . .	62
4.22. Comparativa entre las trayectoria estimada y la real trazada sobre la superficie plana propuesta. . . . .	63
4.23. Fuerzas percibidas por el sensor durante la interacción. . . . .	63
4.24. Trayectorias propuestas para el seguimiento. . . . .	64
4.25. Trayectorias descritas por el dispositivo maestro y esclavo sobre una sección esférica. . . . .	64
4.26. Comparativa entre las trayectoria estimada y la real trazada sobre la superficie semiesférica estimada. . . . .	65
4.27. Fuerzas percibidas por el sensor durante la interacción. . . . .	66
4.28. Trayectorias propuestas para el seguimiento. . . . .	66
4.29. Trayectorias descritas por el dispositivo maestro y esclavo sobre una superficie estimada. . . . .	67
4.30. Comparativa entre las trayectoria estimada y la real trazada sobre la superficie del casco. . . . .	67
4.31. Fuerzas percibidas por el sensor durante la interacción. . . . .	68

# Índice de tablas

---

3.1. Tipos de respuesta de un sistema de segundo orden. . . . .	34
3.2. Valores de los parámetros y condiciones iniciales para el controlador de impedancia. . . . .	35
4.1. Valores de los parámetros de visión estimados para cada una de las cámaras.	50
4.2. Error medio en la estimación paramétrica . . . . .	51
4.3. Valores de los parámetros del control de impedancia . . . . .	59
A.1. Funciones miembro de la clase cUeye. . . . .	72
A.2. Funciones miembro de la clase cFanuc. . . . .	73
A.3. Funciones miembro de la clase cCSM. . . . .	75



# Introducción

---

## Motivación

El presente proyecto se centra en la continuación del desarrollo de un sistema de teleoperación con percepción de fuerza hacia el usuario, donde existe paridad entre superficies de diferente geometría. El proyecto tendrá una posible aplicación en el área de telemedicina, buscando proveer un servicio médico a distancia, usando un robot industrial, medios electrónicos y de telecomunicaciones. La motivación principal para su uso ha sido la de facilitar el acceso a los servicios sanitarios en lugares remotos y aislados. Las personas que viven en las zonas rurales o en poblaciones pobres con escasos recursos, tienen problemas para obtener una adecuada atención médica. Por otro lado, en los hospitales de las grandes ciudades se presentan situaciones en las que el número de personas en espera de un servicio médico excede las capacidades de las instalaciones y en ocasiones, estas personas no necesitan realmente una visita al hospital. En este trabajo se pretende desarrollar un sistema de teleoperación que permita detectar, usando un medio no invasivo como la visión por computadora, la geometría de la superficie sobre la que el robot remoto realizará un contacto. Este proyecto, como se mencionó anteriormente, tiene posible aplicación al telediagnóstico, específicamente usando ultrasonido, en el que un experto proporcione instrucciones de cómo realizar esta prueba a una persona que no tenga el conocimiento, o que él mismo la realice a distancia. Esto debido a que el sensor del ultrasonido debe ser capaz de colocarse y orientarse de una forma adecuada, requiriendo de seis grados de libertad para lograr este objetivo. El proyecto contempla la solución del problema de asimetría cinemática entre el sistema local y el remoto, mediante el mapeo desde una superficie plana, a una superficie arbitraria, que es la que será detectada por medio de visión. Se buscará validar el hecho de que el conocimiento de la geometría de la superficie sobre la que actuará el manipulador, mejorará el desempeño del controlador de impedancia que modula la fuerza de contacto entre el robot y la superficie.

## Objetivo general

El presente trabajo de tesis tiene por objeto el desarrollo un sistema de teleoperación que permita la reconstrucción tridimensional de superficies arbitrarias usando visión no calibrada y validar el desempeño de un controlador de impedancia cinemático integrado a un robot industrial, ante las incertidumbres en la geometría de la superficie sobre la que interactuará el dispositivo.

## Objetivos específicos

Los objetivos particulares planteados en el desarrollo de este proyecto son:

- Desarrollo de un sistema de escaneo de superficies tridimensionales basado en el uso de un método que usa visión libre de calibración.
- Desarrollo de metodología para la solución del problema de asimetría cinemática entre dispositivos local y remoto, a través del mapeo entre superficies.
- Integración al sistema de teleoperación con modulación de fuerza y reflexión de fuerza al usuario.
- Validación experimental de sistema en pruebas que involucren objetos inanimados.

## Planteamiento del problema

Una de las principales dificultades al implementar un controlador de impedancia en sistemas de teleoperación, es el hecho de que se debe tener conocimiento de la superficie sobre la que se realizará el mapeo de trayectorias definidas por un operador. La necesidad de conocer la superficie de objetos cuya superficie es arbitraria es muy importante y más aún el uso de métodos no invasivos para determinarla.

## Metodología

Para cumplir con los objetivos planteados del trabajo de tesis, se trabaja con base en la siguiente metodología. Primero se realiza una revisión bibliográfica para conocer los diferentes desarrollos de sistemas teleoperados, aplicaciones y los tipos de controles que emplean para la interacción con ambientes remotos, así como metodologías empleadas para realizar la reconstrucción de superficies tridimensionales basadas en visión por computadora. El segundo paso es el desarrollo e implementación de una técnica para la estimación de superficies arbitrarias que sea libre de un proceso de calibración, y



comprobar la precisión del método desarrollado. El siguiente paso es el estudio del sistema de teleoperación implementado hasta la fecha y la integración del sistema de visión al mismo; se propondrá una metodología para realizar el mapeo de trayectorias descritas en el área local y trasladarlas a la superficie estimada. Por último, se realizarán experimentos del sistema de teleoperación integrado en su totalidad y se validará el mejoramiento del controlador de impedancia propuesto en trabajos previos, teniendo en cuenta que se tendrá un conocimiento de la superficie arbitraria sobre la que se interactúa.

## Estructura de la tesis

La organización del presente trabajo es como sigue:

- En el capítulo 1 se da una breve introducción a los sistemas de teleoperación, los antecedentes del desarrollo de estos sistemas, mostrando las características y elementos más importantes a considerar en su desarrollo. Se muestran las aplicaciones en el área de medicina, que es una posible aplicación del presente trabajo. Por último se presentan las diferentes arquitecturas propuestas para estos sistemas y los controladores más comunes que se han utilizado para la retroalimentación de fuerzas de interacción con el entorno.
- En el capítulo 2 se presenta una introducción a los sistemas de visión por computadora para obtener coordenadas tridimensionales de objetivos a partir de imágenes bidimensionales adquiridas de una escena. Se describen los elementos que componen un sistema de visión por computadora, el proceso de adquisición y procesamiento de imágenes para la detección de objetivos. Se muestran los modelos matemáticos más utilizados para la reconstrucción tridimensional. Después, se presenta el método de manipulación en espacio de cámara (CSM) para la estimación de coordenadas tridimensionales de objetivos perseguidos en el espacio de cámara. Por último se presenta un método para realizar la correspondencia entre puntos detectados por diferentes cámaras, con el objetivo de obtener las coordenadas tridimensionales de la escena.
- En el capítulo 3 se detalla la integración del sistema de visión explicado en el capítulo 2, con los otros elementos que conforman el sistema de teleoperación. Se detalla el desarrollo de la metodología seguida para resolver el problema de asimetría cinemática que existe entre el robot esclavo y maestro con los que se cuenta en el Laboratorio de Robótica de la Facultad de Ingeniería, así como el cálculo de la orientación que debe tener la herramienta de trabajo para asegurar un movimiento perpendicular a la superficie reconstruida.
- En el capítulo 4 se muestran los resultados experimentales de las pruebas realizadas al sistema, donde se busca validar la estimación de la superficie arbitraria, el

seguimiento de trayectorias definidas por el operador y el desempeño del controlador de impedancia cuando interactúa con el objeto. Se muestran las pruebas con diferentes características para probar el funcionamiento del sistema de visión, la precisión en la estimación y evaluación de la correspondencia entre imágenes de puntos objetivo sobre la superficie arbitraria. También se evalúa el seguimiento correcto de trayectorias estimadas por el método de asimetría cinemática planteado y el comportamiento del controlador de impedancia con el entorno, esto con la ayuda de diferentes objetos de geometría y superficie diferente.

- Por último, al final de esta tesis, se presentan las conclusiones a las que se llegó a partir del desarrollo del presente trabajo y se indica el trabajo a futuro a realizar para continuar con el proyecto de investigación.

# Fundamentos de teleoperación

---

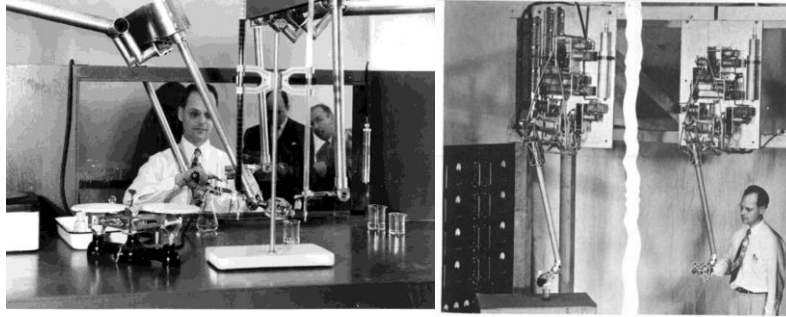
## 1.1. Teleoperación

Los sistemas teleoperados se pueden definir como sistemas que permiten a personas realizar tareas de forma remota, en lugares dónde podría haber un ambiente hostil o el acceso es imposible o desaconsejable. Esto a través del manejo de un robot maestro, el cual se localiza en una zona local de trabajo, que permite controlar el movimiento de un robot esclavo ubicado en una zona remota. Dichas tareas pueden requerir la ejecución de operaciones complejas, ya sea de una precisión muy elevada, requerimientos de fuerzas elevadas para la manipulación de objetos o porque deben ser realizadas en lugares alejados o de difícil acceso. Es por ello que se recurre al uso de robots que permiten expandir las habilidades de un operador para la ejecución de trabajos complejos.

Según la Asociación de Industriales en Robótica (RIA por sus siglas en inglés), un robot se puede definir como “manipulador funcional reprogramable, capaz de mover material, piezas, herramientas o dispositivos especializados mediante movimientos variables programados, con el fin de realizar tareas diversas”. El uso de robots para aumentar las habilidades de un operador, es debido a que son máquinas precisas, soportan ambientes hostiles para el hombre, son reprogramables, permitiendo una gran versatilidad y adaptabilidad.

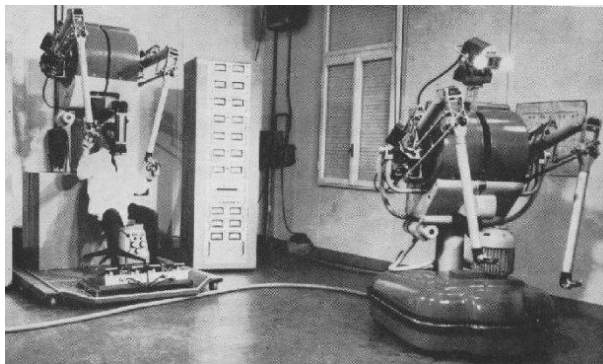
Las primeras investigaciones enfocadas en el desarrollo de sistemas teleoperados fueron realizadas por Raymond Goertz en el Argonne National Laboratory, Estados Unidos, 1947, cuyo objetivo era construir un manipulador que pudiera ser manejado a distancia. El primer desarrollo de estas investigaciones fue la creación de un manipulador esclavo que imitaba los movimientos que un operador realizaba con un brazo unido al manipulador maestro, el cuál fue nombrado M1 y era puramente mecánico; los movimientos se transmitían eje a eje al manipulador esclavo. Para la siguiente década las investigaciones se encaminaron al desarrollo de mecanismos actuados con motores, para los manipuladores maestro y esclavo. No fue hasta 1954 cuando Goertz presentó el pri-

mer manipulador maestro-esclavo con accionamiento eléctrico y servocontrol en ambos manipuladores, el cual fue llamado E1. En la Figura (1.2) se muestran los mecanismos desarrollados por Goertz.



**Figura 1.1:** Raymond Goertz, a la izquierda con el mecanismo maestro esclavo M1, a la derecha con el E1, con accionamiento eléctrico y servocontrol, Argonne National Laboratory.

Para mediados de los 50's del siglo pasado la Agencia Atómica Francesa (Commissariat à l'Energie Atomique), Pesanti y Cherel comienzan el desarrollo de un sistema maestro esclavo mecánico [25]. En el mismo año se desarrolló el sistema teleoperado Mascot, entre un equipo Italiano y el Argonne National Laboratory, el cual era un servo manipulador utilizado en entorno de radiación. Versiones posteriores desarrolladas sirvieron en plantas nucleares e incluían retroalimentación de fuerza hacia el usuario.



**Figura 1.2:** MASCOT, sistema teleoperado con servo manipuladores, Argonne National Laboratory.

Para el año de 1965 Carl Flatau del Brookhaven National Laboratory del Departamento de Energía de EE. UU., desarrolló un telemanipulador que, por vez primera, utilizaba sensores de fuerza y motores de CD. En estos trabajos se experimentaron los primeros problemas debidos a los retardos en la comunicación y se observó por primera

vez la inestabilidad en la reflexión de fuerzas en los sistemas teleoperados. Como resultado de la carrera espacial entre EE. UU. y la URSS, se tuvieron grandes avances en el desarrollo de manipuladores teleoperados. El Surveyor III fue la segunda nave de la serie en realizar el alunizaje y poseía brazos manipuladores para recolectar muestras de la tierra lunar y permitía mediar las fuerzas requeridas, convirtiéndose en el primer sistema teleoperado en el espacio.

En la actualidad el desarrollo de sistemas teleoperados se ha visto ampliado en campos como la minería, cirugía, educación, militar, vigilancia, construcción, rescate, entre otros. Los avances en estos sistemas han estado fuertemente ligados a la evolución de la robótica y la informática, de manera que los objetivos buscados se centran en mejorar la estabilidad, la transparencia, reducir los efectos de los retardos de comunicación y aumentar la seguridad de los sistemas. El uso de sistemas hápticos para la retroalimentación de fuerza hacia el usuario empleando controladores de impedancia, sigue siendo caso de estudio. Por ejemplo, en [21] se construye un modelo de control de fuerza del dispositivo háptico para evaluar los efectos dinámicos de la mano del usuario. Donde se analizan los movimientos de las manos como perturbaciones de fuerza en el sistema de control de la unidad háptica, con el fin de mejorar la transparencia. En [12] se presenta un método de análisis de estabilidad robusto geométrico en 3D basado en las nociones de variables de onda y parámetros de dispersión, desarrollado para determinar la estabilidad en teleoperación y sistemas hápticos.

### 1.1.1. Características de los sistemas de teleoperación

La mayoría de las características de los sistemas teleoperados se conservan hasta la fecha, sin embargo, los avances modernos en actuadores, sensores, sistemas de procesamiento de datos, en telecomunicaciones y diseño de robots, han incrementado las capacidades y áreas de uso de los sistemas teleoperados. Estos avances permiten expandir con mayor precisión y fidelidad las habilidades humanas, permiten incrementar la complejidad y proponer gran variedad de arquitecturas a los sistemas de teleoperación, lo que fomenta realizar investigaciones en este campo para mejorar la estabilidad y aumentar la transparencia de los sistemas.

Al aumentar la complejidad de los sistemas también surge la necesidad de asegurar la estabilidad en los mismos. El concepto de estabilidad es muy común y es una propiedad de los sistemas, en el que es estable si responde con una variación finita a variaciones finitas de sus señales de entrada. En este caso, es necesario que los robots respondan de manera estable durante la teleoperación, ya que puede dañar al entorno de no ser así. El ruido eléctrico y los retrasos en la comunicación son causantes de inestabilidades. En este segundo caso, es debido a los retrasos en el protocolo de comunicación utilizado y la distancia a la que estén separados los dispositivos.

Otra característica importante en los sistema de teleoperación, aparte de la estabilidad, es la transparencia y se considera un parámetro importante para medir el

desempeño de la teleoperación. El concepto de transparencia tiene varias definiciones, aunque una acepción aceptada es la capacidad de la teleoperación para hacer que un operador perciba el ambiente en que interactúa el robot esclavo como si lo estuviera manipulando directamente.

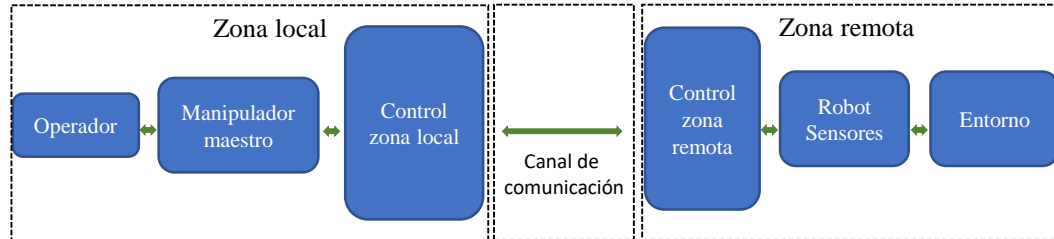
Existen varios estudios que aseguran que la transparencia perfecta puede ser lograda en teoría, sin embargo es muy complicado realizarla en la práctica. Los trabajos realizados por Lawrence y Yokokohji [18] demostraron que la transparencia perfecta es realizable con el uso de la fuerza y la retroalimentación de posición del esclavo al maestro (conocida como arquitectura de cuatro canales). Uno de los requisitos es la medición de aceleraciones para la implementación de los algoritmos, algo que es complicado en la práctica. Los trabajos de Hashtrudi-Zaad y Salcudean [13] se oponen ligeramente a esta afirmación y muestran como una transparencia perfecta es obtenible con una arquitectura de tan solo tres canales. El algoritmo requiere que un elemento de retroalimentación de fuerza local debe ser incluido en el controlador. Estos estudios muestran como estas dos características (estabilidad y transparencia) compiten en objetivos de diseño de sistemas teleoperados.

Otra característica importante es el método de comunicación entre los dispositivos esclavo y maestro en el sistema. El uso y expansión de protocolos de comunicación por internet, han permitido la creación de sistemas que se comunican a muy grandes distancias, algo que no era realizable con protocolos seriales como el RS-232 y expandir los UDP/IP, más allá de las áreas de conexión locales. Si bien, esto permitió desarrollar sistemas teleoperados que se comunican a grandes distancias a través de internet, los retardos en la comunicación fueron más notorios con este tipo de señales. Se han realizado investigaciones de los efectos provocados por los retardos en comunicación. Por ejemplo, en [7] y [1] se presenta un estudio de los retardos en el sistema de transmisión de datos entre dos puntos, en sistemas con control bilateral implementado, los cuales utilizan una red de internet.

## 1.2. Elementos de un sistema de teleoperación

Una manera sencilla de describir a un sistema teleoperado es mediante un operador que describe trayectorias que debe realizar un robot esclavo en una área remota [3], el cual realiza las operaciones indicadas y devuelve la información de su estado y del entorno sobre el que interactúa. Esta retroalimentación de información es lo que permite cerrar el lazo de control en el sistema. Existen varios elementos presentes en el ciclo de control y que son esenciales para el funcionamiento del sistema teleoperado. Además se pueden clasificar dependiendo del espacio en que este ubicados, en la Figura (1.3) se muestra un esquema de teleoperación dividido en tres zonas, la zona local, la remota y el canal de comunicación entre ellas.

Cada una de las zonas cuentan con canales de comunicación bidireccionales para el envío y recepción de información. La descripción de cada una de las zonas se detalla continuación:



**Figura 1.3:** Esquema típico de un sistema teleoperado dividido por zonas.

- **Zona local:** En esta zona se encuentra el operador humano, quien realiza la descripción de las tareas a realizar. Se cuenta con un dispositivo de actuación (manipulador maestro) para generar las trayectorias descritas por el operador para ser mapeadas al robot de la zona remota. Para que el operador pueda percibir el entorno sobre el que está trabajando, es necesario dotar a la interfaz del sistema teleoperado, de sensores para la retroalimentación para que el operador pueda tener información de la ejecución de la tarea que se esté realizando.
- **Zona remota:** Es el espacio en que se encuentra el manipulador esclavo y el ambiente u objetos sobre los que interactúa, denominado entorno remoto de trabajo. Es importante resaltar que a diferencia del manipulador de la zona local en el que el ambiente de trabajo es conocido por el operador, el entorno de trabajo del robot esclavo es no estructurado, variable o desconocido. Es necesario dotar al sistema de elementos que permitan conocer o determinar la geometría del entorno remoto o de la superficie de los objetos con que se interactúa.
- **Canal de comunicación:** Es el medio a través del cual se comunican las zonas local y remota. Estos canales pueden emplear diversos protocolos de comunicación, los cuales son elegidos dependiendo de la distancia entre ambas zonas. Desde protocolos de comunicación serial para distancias cortas, hasta protocolos que utilizan redes locales LAN o inclusive por redes de Internet para distancias largas de comunicación. Existen dos características para la elección del canal de comunicación, el ancho de banda y el retardo de comunicación en la transmisión de información. Ya que de no contar con las especificaciones necesarias el sistema puede volverse inestable.

La descripción de cada uno de los elementos presentes en un sistema de teleoperación es la siguiente:

- **Operador:** es la persona encargada de realizar las tareas o trayectorias que debe seguir el robot esclavo. Es quien tiene conocimiento de cómo debe manejarse el

entorno remoto o los objetos que estén dentro de él.

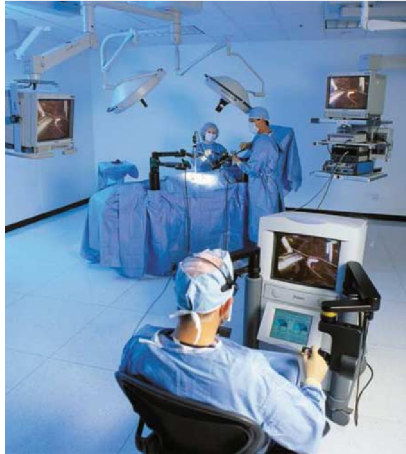
- Manipulador maestro: es el dispositivo encargado de capturar las tareas descritas por el operador, adquiere los parámetros de entrada al controlador del sistema. Se encarga de retroalimentar la información que percibe el manipulador remoto durante la interacción con el entorno. El uso de robots manipuladores es muy común con estos fines, los cuales deben contar con actuadores capaces de ejercer fuerzas hacia el usuario.
- Control zonal local: Es utilizado para mejorar la calidad de la operación, el desempeño o para disminuir efectos producidos por ruido, ya sea eléctrico, de medición o retardos en la comunicación. Se encarga de realizar el mapeo de las trayectorias descritas por el operador y enviarlas al controlador remoto.
- Controlador zona remota: Recibe las trayectorias deseadas a seguir en el entorno y las ejecuta. Es el encargado de determinar las posiciones, velocidades y pares que debe realizar el robot para alcanzar las posiciones deseadas. Si existe una retroalimentación de fuerza percibida del entorno, se encarga de realizar las correcciones a las posiciones deseadas recibidas.
- Manipulador esclavo: También conocido como robot esclavo, es quien lleva a cabo las tareas de seguimiento e interacción con el entorno que le son enviadas. Típicamente suelen emplearse robots manipuladores seriales de seis grados de libertad, para alcanzar las posiciones deseadas dentro de un espacio de trabajo limitado con una orientación deseada.
- Sensores: Permiten adquirir los estados de los robots, del entorno y las tareas realizadas. Ayudan al operador a percibir el entorno de trabajo y realizar acciones o correcciones de las tareas. Existen diversos sensores que permiten medir las variables presentes en el sistema, por ejemplo, la adquisición de imágenes del entorno, fuerzas y pares presentes en el contacto de la herramienta del robot esclavo y el entorno.

### 1.3. Teleoperación en la medicina

El uso de sistemas de teleoperación en el área médica ha sido recientemente fortalecido de forma importante [6, 8]. Desde los desarrollos de prótesis o dispositivos de asistencia a discapacitados, hasta novedosos sistemas de telecirugía o teliagnóstico. Uno de los casos más importantes registrados, fue la primera cirugía asistida por teleoperación, en la cual el sistema robótico ZEUS se utilizó con este fin (ver Figura (1.4)), la parte maestra (un cirujano especializado) se encontraba ubicado en la ciudad de Nueva York, en EE. UU., mientras que el esclavo se encontraba en Estrasburgo, Francia. El tipo de cirugía realizada fue una colecistomía laparoscópica y la operación Lindbergh, siendo un éxito. Donde la comunicación de los dispositivos se realizó a través



de un canal privado de Internet, con un protocolo TCP UDP/IP y un retardo en la comunicación de 224 *ms*.



**Figura 1.4:** Sistema de telecirugía ZEUZ ®Surgical System.

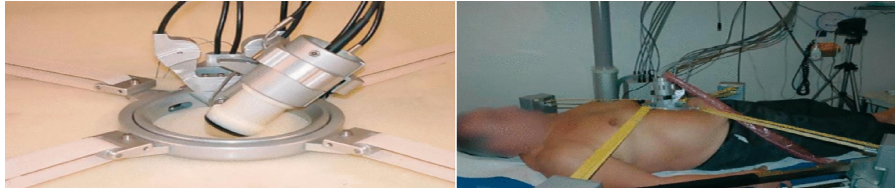
Otro caso de cirugía teleoperada es el sistema desarrollado Intuitive Surgical Inc. llamado DaVinci, el cual consiste en tres partes separadas. En una de ellas el cirujano se coloca en una posición ergonómica y ventajosa dentro de la estación de trabajo. Con sus manos toma los dispositivos maestros que actúan como interfaz con la computadora. La computadora y un sistema de imágenes en 3D ocupan el resto del espacio en la estación de trabajo. La tercera parte consiste en un robot que tiene tres brazos manipuladores. El brazo central es el encargado de sostener la cámara y los restantes están adaptados con los instrumentos quirúrgicos que reproducen el movimiento del cirujano. Los instrumentos quirúrgicos se mueven como una muñeca, éstos tienen 7 grados de libertad y 2 grados de libertad de rotaciones adicionales. El robot se posiciona sobre una mesa de operaciones. En 1999 se reportó la primera intervención quirúrgica exitosa con Da Vinci en un bypass de una arteria coronaria.

El área de telediagnóstico es la aplicación de las técnicas de telemedicina para hacer posible la comunicación e interacción entre los profesionales de la salud [14], accediendo a la opinión especializada o estableciendo un diagnóstico cooperativo a partir del intercambio de información clínica del paciente. Esta interacción se puede realizar en tiempo real. Un trabajo realizado cuyo objetivo es su funcionamiento en tiempo real es el llamado TER por sus siglas en inglés. El objetivo de este proyecto es proporcionar una solución fiable para llevar a cabo los exámenes de ultrasonido de expertos, en ubicaciones lejanas. TER es un sistema de telerobótica diseñado y desarrollado por un consorcio francés integrado por universidades, hospitales y empresas industriales. Una originalidad de TER es el desarrollo de un robot esclavo compatible accionado por los músculos. Este robot esclavo es teleoperado por un médico experto que realiza de forma remota el examen, donde cabe mencionar que el robot esclavo es de diseño específico



**Figura 1.5:** Sistema de telecirugía DaVinci.

para la tarea.



**Figura 1.6:** Sistema teleoperado TER, .

## 1.4. Control de sistemas teleoperados

Existen dos estrategias de control para un sistema de teleoperación, en el que se requiere controlar un robot esclavo por medio de un robot maestro. Éstas dependen de la aplicación y los parámetros del sistema y pueden ser:

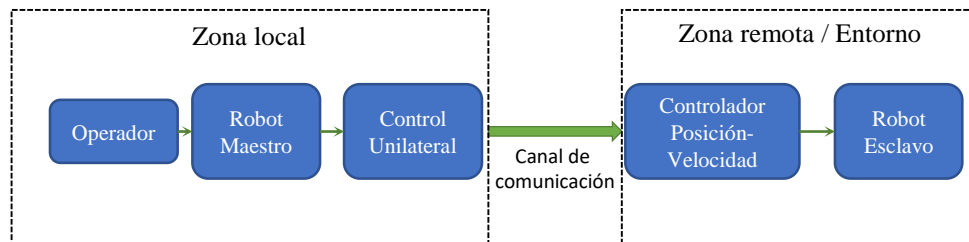
- Control unilateral: Esta estrategia consiste en generar una trayectoria con el robot maestro, la cual sirve como referencia, ya sea en posición o velocidad y es transmitida al robot esclavo.
- Control bilateral: Además de ser mapeadas trayectorias descritas por el robot maestro, se busca una retroalimentación o reflexión de fuerza hacia el dispositivo maestro de la interacción del robot esclavo con el ambiente.

Como se menciona, una característica importante del control bilateral es que el control se hace en posición-velocidad y fuerzas pares de las articulaciones de los manipuladores, para lo cual es necesario contar con robots de arquitectura abierta, es decir que permita un control en estas variables. Si solo se cuenta con el control de posición o velocidad por separado, no puede ser considerado un control bilateral. El robot empleado durante el desarrollo de este proyecto es de arquitectura cerrada, ya que solo permite controlar la posición. No se tiene acceso a las velocidades, aceleraciones, fuerzas o pares del manipulador. El movimiento del robot se logra a partir del envío de posiciones deseadas y el controlador interno del robot se encarga de realizar los procesos para alcanzar dicha posición, por lo que el control de pares está oculto al usuario.

#### 1.4.1. Control unilateral

Un sistema teleoperado con control unilateral consiste en un robot maestro que controla a un robot esclavo, el cual esta colocado en un ambiente remoto. Cuando los principales parámetros que sirven de referencia para trazar las trayectorias del esclavo son posición y velocidad, se describe un sistema unilateral [28].

En la Figura (1.7) se muestra un esquema típico de los sistemas teleoperados con control unilateral. Los componentes del esquema presentado son: Un operador humano (describe las trayectorias a seguir), robot maestro (genera las posiciones y velocidades articulares a mapear), controlador maestro y esclavo (encargados de realizar el mapeo y reproducir la trayectoria, respectivamente), canal de comunicación y el espacio de trabajo de la zona remota (entorno de interacción). Este tipo de esquema no cuenta con retroalimentación del ambiente remoto y las señales fluyen en un solo sentido, aunque existen casos en que, al usar un robot industrial que tiene un sensor de fuerza en su herramienta y posee retroalimentación de fuerzas de contacto, sin embargo el control sigue siendo unilateral.



**Figura 1.7:** Esquema típico de un sistema teleoperado con control unilateral.

### 1.4.2. Control bilateral

Además del control de posición-velocidad o fuerza-par, otra característica importante de los controles bilaterales es que permiten tener retroalimentación de información hacia el usuario. Existen diversas arquitecturas de control de teleoperación que se han clasificado en función de los parámetros que se intercambian entre los dispositivos maestro y esclavo. Las categorías en que se dividen son:

- **Posición-posición:** La posición del robot esclavo es determinada directamente de la posición del maestro, sin necesidad de sensores de fuerza. Es el esquema de control más sencillo, cuenta con una buena estabilidad pero ofrece poca transparencia.
- **Fuerza-posición:** La posición del robot esclavo es determinada a partir de la trayectoria descrita por el maestro y las fuerzas de interacción que percibe el esclavo del entorno son retroalimentadas y reproducidas por el maestro. Por lo que es necesario contar con un sensor de fuerza en la herramienta del manipulador remoto. Esta arquitectura es la utilizada en el presente trabajo.
- **Fuerza-Fuerza:** La trayectorias del robot maestro y esclavo se determinan a partir de las lecturas de fuerzas de ambos dispositivos. Aunque también existe un control local de posición en cada uno de los manipuladores. Esta arquitectura cuenta con una buena transparencia, sin embargo tienden a ser inestables [24].
- **Posición-fuerza:** El robot maestro sigue la posición del robot esclavo, mientras que el esclavo es controlado a partir de la fuerza ejercida por el operador sobre el manipulador maestro.

Las arquitecturas descritas anteriormente se generalizan en un controlador de cuatro canales [17], donde las posiciones y fuerzas son los parámetros intercambiados en los canales de comunicación de los manipuladores maestro y esclavo. Esta clasificación se basa en la dualidad que existe en las posibles tareas que pueden realizar los robots del sistema. Se observa la importancia de contar con controladores de posición que consigan ubicar los manipuladores en las posiciones deseadas con precisión y repetibilidad, además de que proporcionen un control en fuerza para la ejecución de tareas que requieren contacto con el entorno.

### 1.4.3. Control de impedancia

Como se mencionó anteriormente, la retroalimentación de fuerzas que percibe el robot esclavo hacia el robot maestro mejora la transparencia en el sistema de teleoperación. De manera tal que un control de interacción física entre la herramienta del robot esclavo y el entorno donde opera es de suma importancia para la correcta ejecución de tareas que requieren contacto con la superficie de objetos en dicho entorno. Durante el contacto, las superficies pueden contener restricciones dentro de la trayectoria a seguir

por la herramienta. Si el robot esclavo presenta restricciones para acceder a las variables de control, posición-velocidad o fuerza-par, las arquitecturas descritas en la sección (1.4.2) resultan insuficientes. Una arquitectura propuesta para realizar retroalimentación de fuerzas cuando se tienen este tipo de restricciones es el control de impedancia. El control de impedancia pretende que la interacción dinámica entre el manipulador y su entorno sea modulada y controlada por la variación de dicha impedancia. Esto se realiza a partir de un sensor de fuerza colocado en el efector final.

En los trabajos desarrollados por Hogan [15] se propone un modelo de interacción que consiste en imponer una dinámica de un modelo generalizado masa, resorte amortiguador al robot maestro, donde la relación entre la trayectoria de la herramienta del esclavo, la trayectoria deseada y las fuerzas de contacto estén dadas por la expresión (3.1).

Lawrence [17], presenta un enfoque de control de impedancia basado en posición, donde las fuerzas y los pares son explícitamente medidos usando un sensor de fuerza en la herramienta del robot. Con base en las fuerzas y pares de contacto, se obtiene un vector de posición con el que se ajusta la posición del robot y la nueva configuración se envía al lazo de control interno.

En [21] se analiza un modelo de control de fuerza de impedancia adaptativa de un dispositivo háptico para teleoperación. Se propone un nuevo modelo de control de fuerza del dispositivo háptico para investigar los efectos dinámicos de la mano del usuario donde los movimientos de la mano del usuario pueden funcionar como una perturbación de la fuerza en el sistema de control de la fuerza háptica. Por lo que, en controladores de impedancia, es posible obtener un buen grado de transparencia. Hoy en día los trabajos en arquitecturas de control para sistemas teleoperados y controladores de impedancia, siguen siendo objeto de estudio.



---

## Capítulo 2

# Método de reconstrucción de superficies arbitrarias

---

En este capítulo se presenta una introducción a los sistemas de visión por computadora para obtener coordenadas tridimensionales de objetivos a partir de imágenes bidimensionales adquiridas de una escena. Se describen brevemente los elementos que componen un sistema de visión por computadora, el proceso de adquisición y procesamiento de imágenes para la detección de objetivos, se presentan los modelos matemáticos utilizados para la reconstrucción tridimensional. A continuación, se presenta el método de manipulación en espacio de cámara (CSM) para la estimación de coordenadas tridimensionales de objetivos perseguidos en el espacio de cámara, el cual ha sido probado en tareas de posicionamiento y seguimiento [10]. En este trabajo se proyectan puntos de luz láser sobre un objeto de trabajo, que funcionan como marcas visuales detectadas por las cámaras. Para realizar la estimación de posición de las marcas visuales proyectadas se propone un método para resolver la correspondencia entre objetivos vistos con dos cámaras y realizar un empate entre ellos.

### 2.1. Visión por computadora

La visión por computadora, también denominada visión artificial, es un proceso de extracción de información de una escena o ambiente a partir de imágenes adquiridas por sensores y procesadas por una computadora. Un sistema de visión realiza alguna de las tareas que el sistema de visión humano es capaz de realizar [22], por ejemplo pueden de extraer la estructura y propiedades de una escena tridimensional a partir de una o más imágenes bidimensionales. Uno de los principales problemas es la pérdida de una dimensión durante el proceso de adquisición de la imagen [20], ya que el procesamiento se hace a partir de proyecciones de dos dimensiones, compuestas por una matriz de puntos denominados píxeles. Cada pixel puede almacenar valores de un tipo en concreto dependiendo de la representación de la imagen.

El proceso de visión por computadora se suele dividir en etapas clasificadas en dos grupos, el primero se encarga de ejecutar los métodos de bajo nivel. Esto es obtener las características más básicas de una imagen (bordes, regiones, etc.). El segundo se encarga del procesamiento de alto nivel, esto es, a partir de la información obtenida del nivel bajo, construir una descripción de la escena lo más precisa posible. Dicho proceso involucra las siguientes etapas:

- **Adquisición de la imagen:** Se realiza la captura de una escena con un sensor de imagen, como lo es una cámara.
- **Preprocesamiento:** Incluye métodos de eliminación de ruido, suavizado y realce de la imagen de la etapa anterior.
- **Segmentación:** Permite separar los elementos de una escena, ya sea con detección de bordes o regiones.
- **Extracción de características:** Se obtiene una representación formal de los elementos segmentados.
- **Reconocimiento:** mediante técnicas se localizan objetos en la escena y se reconstruyen sus coordenadas tridimensionales con métodos como la triangulación.
- **Interpretación:** Se realiza la interpretación de la escena y se toman las acciones necesarias.

Existen una gran variedad de técnicas de reconstrucción de superficies en tres dimensiones, las cuales se dividen [27] en dos grandes grupos: adquisición por contacto y sin contacto. Sin embargo, las técnicas de contacto son invasivas con el objeto, algo no deseable en la aplicación planteada. Dentro de las técnicas sin contacto existen técnicas no ópticas, como el radar y el sonar, sin embargo no cuentan con una buena precisión, comparadas con las técnicas ópticas. Estas técnicas son utilizadas por la visión por computadora para la reconstrucción o estimación de superficies dentro de un espacio de trabajo.

A su vez, las técnicas ópticas se pueden clasificar en dos grupos según las fuentes de iluminación utilizadas para iluminar la escena [2], técnicas ópticas pasivas y técnicas ópticas activas. En las técnicas pasivas la geometría de las fuentes de iluminación no influye en la reconstrucción, mientras que las técnicas activas utilizan fuentes de luz con características específicas, cuya posición influye directamente en la determinación de las coordenadas tridimensionales. Se han observado mejores resultados en las técnicas activas a la hora de realizar reconstrucciones tridimensionales de precisión, por ello es que en este trabajo se emplea una de estas técnicas.



## 2.2. Coordenadas homogéneas y transformaciones elementales

La proyección de la luz reflejada por elementos de una escena en el espacio de la imagen se realiza mediante una transformación perspectiva. Esta transformación genera la formación de la imagen a partir de la proyección de un espacio en tres dimensiones a uno en dos dimensiones. Para ello es necesario conocer cómo se obtienen las coordenadas homogéneas, las transformaciones básicas de puntos y sistemas coordenados.

En geometría proyectiva, las coordenadas homogéneas de un punto en tres dimensiones cuyas coordenadas cartesianas son  $(X, Y, Z)$ , se definen con el punto tetradiimensional mostrado en (2.1), donde  $k$  es una constante arbitraria distinta de cero.

$$(kX, kY, kZ, k) \quad , \quad k \neq 0 \quad (2.1)$$

De esa manera un punto  $P$  del espacio expresado de manera vectorial en coordenadas cartesianas, queda definido como en (2.2).

$$P = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (2.2)$$

Y su representación en coordenadas homogéneas esta dado por (2.3)

$$P_h = \begin{bmatrix} kX \\ kY \\ kZ \\ k \end{bmatrix} \quad (2.3)$$

Esta representación es muy importante, ya que para aplicar operaciones de transformación sobre un punto en el espacio, debe estar expresado en sus coordenadas homogéneas. En la mayoría de los casos, por simplicidad el valor de  $k$  es igual a uno.

La traslación de un punto  $P = [X \ Y \ Z]$  [5] expresado en sus coordenadas homogéneas se muestra en la ecuación (2.4), donde el vector  $[X_0 \ Y_0 \ Z_0]^T$  es el vector de traslación en cada uno de sus ejes.

$$\begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & X_0 \\ 0 & 1 & 0 & Y_0 \\ 0 & 0 & 1 & Z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (2.4)$$

Para realizar el escalado de un punto en el espacio  $P = [X \ Y \ Z]$ , con factores de escala  $s_x$ ,  $s_y$  y  $s_z$  es cada uno de los ejes cartesianos  $X$ ,  $Y$  y  $Z$  respectivamente,

expresando en sus coordenadas homogéneas se muestra en la ecuación (2.5). Un caso particular es el de realizar la misma escala en cada uno de los ejes, para esto  $s_x = s_y = s_z$ .

$$\begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (2.5)$$

Si se desea realizar una rotación de un punto  $P$  respecto a cada uno de sus ejes cartesianos, de manera que el ángulo  $\alpha$  se rota sobre el eje  $X$ ,  $\beta$  sobre el eje  $Y$  y  $\gamma$  un ángulo en el eje  $Z$ , expresada en coordenadas homogéneas, se definen las matrices de rotación  $R_{x,\alpha}$ ,  $R_{y,\beta}$  y  $R_{z,\gamma}$  como se muestra en (2.6), (2.7) y (2.8).

$$R_{x,\alpha} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & \sin \alpha & 0 \\ 0 & -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.6)$$

$$R_{y,\beta} = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.7)$$

$$R_{z,\gamma} = \begin{bmatrix} \cos(\gamma) & \sin(\gamma) & 0 & 0 \\ -\sin(\gamma) & \cos(\gamma) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.8)$$

## 2.3. Modelo de cámara

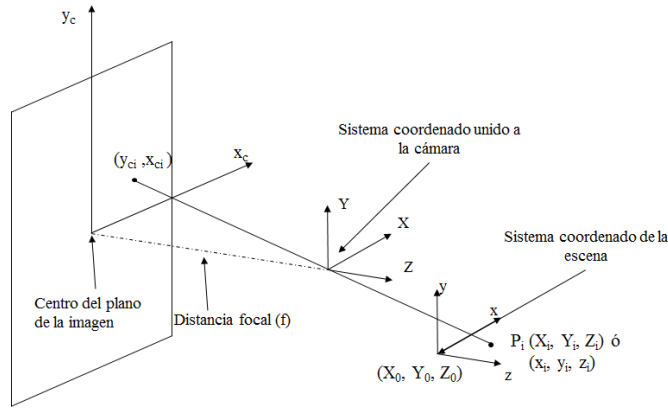
### 2.3.1. Proyección perspectiva

Uno de los modelos más utilizados de transformación que no considera deformaciones en las imágenes causadas por las distorsiones no lineales de la óptica de una cámara, es el modelo de cámara de orificio.

Este modelo considera la óptica del sistema formada por una única lente representada por un punto infinitesimal (denominado foco), a través del cual pasan los rayos de luz de una escena hacia el sensor de la cámara. Por lo que todos los puntos de la escena estarán enfocados en la imagen. Este modelo equivale en una transformación perspectiva, cuyo principal parámetro es la distancia focal.

Por lo tanto, la proyección en perspectiva es una proyección de puntos tridimensionales en un espacio (de una escena) a puntos bidimensionales en una imagen, a través de líneas de proyección que pasan por un punto (pinhole), conocido como centro óptico. En la figura 2.1 se ilustra la formación de la imagen mediante la proyección en perspectiva, donde  $(X, Y, Z)$  es el sistema de coordenadas unido a la cámara, desplazado a lo largo del eje focal de plano  $(x_c, y_c)$  de la imagen. Se aprecia que el punto  $P_i$ , con coordenadas  $(X_i, Y_i, Z_i)$ , referido al marco de referencia de la escena, tiene coordenadas del punto correspondiente en el plano de la imagen  $(x_{ci}, y_{ci})$ , las cuales pueden modelarse por la ecuación (2.9).

$$x_{ci} = f \frac{X_i}{Z_i}, y_{ci} = f \frac{Y_i}{Z_i} \quad (2.9)$$



**Figura 2.1:** Proyección perspectiva de una escena tridimensional en una bidimensional.

La relación de transformación que se define entre el sistema de coordenadas unido a la cámara y el marco de referencia de la escena se puede expresar en la ecuación (2.10)

$$\begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix} = \mathbf{H}_{xyz}^{XYZ} \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} \quad (2.10)$$

donde la matriz de transformación  $\mathbf{H}_{xyz}^{XYZ}$  puede ser representada por una matriz de rotación a partir de los parámetros de Euler, por lo que la expresión (2.10) puede definirse como lo muestra la ecuación (2.11).

$$\begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix} = \begin{bmatrix} e_1^2 + e_2^2 - e_3^2 - e_4^2 & 2(e_2e_3 + e_1e_4) & 2(e_2e_4 - e_1e_3) & X_0 \\ 2(e_2e_3 - e_1e_4) & e_1^2 - e_2^2 + e_3^2 - e_4^2 & 2(e_3e_4 + e_1e_2) & Y_0 \\ 2(e_2e_4 + e_1e_3) & 2(e_3e_4 - e_1e_2) & e_1^2 - e_2^2 - e_3^2 + e_4^2 & Z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} \quad (2.11)$$

donde los términos  $e_1, e_2, e_3, e_4$  representan los cuatro parámetros de Euler y satisfacen la restricción  $e_1^2 + e_2^2 + e_3^2 + e_4^2 = 1$  y el vector de traslación se localiza en el origen del sistema de referencia de la escena  $(X_0, Y_0, Z_0)$  relativo al sistema de coordenadas de la cámara. Se observa que el modelo de cámara de proyección perspectiva contiene siete parámetros independientes:  $X_0, Y_0, Z_0, f$  y tres de los cuatro parámetros de Euler.

### 2.3.2. Modelo ortográfico de cámara

El modelo ortográfico de cámara es una simplificación de la proyección perspectiva descrita en la sección anterior [11]. Si se considera que el origen en el eje  $Z$  del sistema de referencia de la escena  $(Z_0, \text{ver figura 2.1})$  tiene un límite asintótico mucho mayor que las cantidades  $x_i, y_i, z_i, X_0, Y_0$ , se obtiene la simplificación del modelo, por lo que las coordenadas del plano de la imagen  $(x_{ci}, y_{ci})$  obtenidas en la ecuación (2.9) se reescriben como lo muestra en (2.12)

$$\begin{aligned} x_{ci} &= f \frac{X_i}{Z_i} = \frac{N_x}{D + \frac{1}{\epsilon}} = \Psi_x(\epsilon) \\ y_{ci} &= f \frac{Y_i}{Z_i} = \frac{N_y}{D + \frac{1}{\epsilon}} = \Psi_y(\epsilon) \end{aligned} \quad (2.12)$$

donde los términos  $N_x, N_y$  y  $D$  se expresan por la ecuación (2.13).

$$\begin{aligned} N_x &= f [(e_1^2 + e_2^2 - e_3^2 - e_4^2)x_i + 2(e_2e_3 + e_1e_4)y_i + 2(e_2e_3 - e_1e_2)z_i + X_0] \\ N_y &= f [2(e_2e_3 - e_1e_4)x_i + (e_1^2 - e_2^2 + e_3^2 - e_4^2)y_i + 2(e_3e_4 + e_1e_2)z_i + Y_0] \\ D &= [2(e_2e_4 + e_1e_3)x_i + 2(e_3e_4 - e_1e_2)y_i + (e_1^2 - e_2^2 - e_3^2 + e_4^2)z_i] \\ \epsilon &= \frac{1}{Z_0} \end{aligned} \quad (2.13)$$

Si se expande el valor de  $\Psi_x$  usando una serie de Taylor para cuando  $\epsilon = 0$  se obtiene la ecuación (2.14). De manera similar para  $y_c$ .

$$\Psi_x(\epsilon) = \Psi_x|_{\epsilon=0} + \frac{\partial \Psi_x}{\partial \epsilon}|_{\epsilon=0} \epsilon + \frac{\partial^2 \Psi_x}{\partial \epsilon^2}|_{\epsilon=0} \frac{\epsilon^2}{2} + \dots \quad (2.14)$$

Considerando solo el término de primer orden en (2.14), entonces  $\frac{f}{Z_0}$  puede combinarse con el resto de los parámetros para formar un nuevo vector de parámetros  $\mathbf{C} = [C_1, \dots, C_6]^T$ . Por lo que el modelo de cámara ortográfico esta dado por la las ecuaciones mostradas en (2.15).

$$\begin{aligned} f_x(x, y, z, \mathbf{C}) &= b_1(\mathbf{C})x_i + b_2(\mathbf{C})y_i + b_3(\mathbf{C})z_i + b_4(\mathbf{C}) \\ f_y(x, y, z, \mathbf{C}) &= b_5(\mathbf{C})x_i + b_6(\mathbf{C})y_i + b_7(\mathbf{C})z_i + b_8(\mathbf{C}) \end{aligned} \quad (2.15)$$

donde,

$$\begin{aligned}
 b_1(\mathbf{C}) &= (C_1^2 + C_2^2 - C_3^2 - C_4^2) \\
 b_2(\mathbf{C}) &= 2(C_2C_3 + C_1C_4) \\
 b_3(\mathbf{C}) &= 2(C_2C_4 - C_1C_3) \\
 b_4(\mathbf{C}) &= C_5 \\
 b_5(\mathbf{C}) &= 2(C_2C_3 - C_1C_4) \\
 b_6(\mathbf{C}) &= (C_1^2 - C_2^2 + C_3^2 - C_4^2) \\
 b_7(\mathbf{C}) &= 2(C_3C_4 + C_1C_2) \\
 b_8(\mathbf{C}) &= C_6
 \end{aligned} \tag{2.16}$$

Por lo que, las coordenadas  $(x_{ci}, y_{ci})$  de un punto  $P$  en el sistema de referencia de la escena pueden obtenerse a partir del modelo ortográfico, si se conoce la coordenada  $(x_i, y_i, z_i)$  y los seis parámetros de visión contenidos en  $\mathbf{C}$ , los cuales definen la relación entre la localización física del punto tridimensional  $P$  y su localización en el plano de la imagen. La determinación de los parámetros de visión se realiza por algún método de estimación, en este trabajo se realiza por el método de mínimos cuadrados mostrado en la sección (2.4)

## 2.4. Método de manipulación de espacio de cámara

Como se observa en los modelos descritos en las secciones anteriores, el lente de la cámara del sistema de visión aplica una transformación de perspectiva a la luz que incide sobre ella, la cual procede de la escena. Esta transformación relaciona el sistema coordenado de la escena con el sistema coordenado de dos dimensiones de la imagen. Para reconstruir una superficie tridimensional en la escena a partir de una imagen de dos dimensiones es necesario calcular la transformación inversa que permita calcular las coordenadas tridimensionales de cada punto objetivo proyectado sobre la superficie. Por lo tanto, es necesario conocer con exactitud la transformación que realiza la cámara, para realizar el modelo matemático que permita realizar la transformación inversa. El proceso conocido como calibración de cámara, permite determinar los parámetros de la transformación entre puntos objetivo en la escena y puntos en la imagen, de la proyección perspectiva. Existen dos tipos de parámetros de esta transformación:

- Parámetros intrínsecos: Son las propiedades físicas de una cámara y de su lente. La distancia focal, el desplazamiento del centro de la imagen y el coeficiente de distorsión radial, conforman estos parámetros, que en ocasiones, son proporcionados por el fabricante.
- Parámetros extrínsecos: Definen la posición y orientación de la cámara con respecto al sistema de coordenadas de la escena. Las traslaciones  $T_x$ ,  $T_y$ ,  $T_z$  y las rotaciones en ángulos  $\alpha$ ,  $\beta$ ,  $\gamma$ .

Sin embargo, para este trabajo se propone realizar un sistema que requiera de un proceso libre de calibración, de manera que el usuario final no tenga que realizar un proceso de calibración de cámara, y que la reconstrucción de la superficie sea lo más sencilla posible, sin perder la precisión y fidelidad en la reconstrucción.

El método de manipulación en espacio de cámara es una técnica que usa visión por computadora que no requiere de un proceso de calibración del sistema de visión ni del robot [10],[11]. Este método ha sido utilizado satisfactoriamente para tareas de posicionamiento y orientación de manera precisa en un espacio tridimensional [10],[9],[4],[19]. Es posible manipular objetos y colocarlos en posiciones y orientaciones deseadas con respecto a objetos fijos.

Una característica importante del método es que los objetivos se definen y persiguen en los marcos de referencia de las imágenes obtenidas por las cámaras del sistema de visión. Los objetivos se estiman usando seis parámetros de visión, los cuales son estimados a partir de un proceso no lineal. Así mismo, definen una relación no lineal entre la configuración interna del robot empleado y su localización correspondiente en el plano de imagen (del espacio de cámara), de un número de observaciones realizadas. Esta relación se logra por medio del modelo cinemático del robot y el modelo de cámara utilizado.

### 2.4.1. Estimación no lineal por método de mínimos cuadrados

A partir del modelo de cámara ortográfico, se definen seis parámetros de visión contenidos en un vector  $\mathbf{C}$  que aparecen en la ecuación (2.17),

$$\mathbf{C} = [C_1, \dots, C_6]^T \quad (2.17)$$

Los parámetros se determinan independientemente para cada cámara al minimizar la función  $J(C)$ , definido en la ecuación (2.18).

$$J(C) = \sum_i^m \left\{ [x_{ci} - f_x(x_i, y_i, z_i; \mathbf{C})]^2 + [y_{ci} - f_y(x_i, y_i, z_i; \mathbf{C})]^2 \right\} W_i \quad (2.18)$$

donde:

- $m$  := número de observaciones para determinar los parámetros de visión.
- $(x_{ci}, y_{ci})$  := coordenadas de la observación en la imagen de la cámara.
- $(x_i, y_i, z_i)$  := coordenadas tridimensionales de la posición real de las observaciones referidas al marco fijo del robot manipulador.

- $\mathbf{W}_i$  := Peso relativo positivo dado a la medición.

Las relaciones  $f_x(x_i, y_i, z_i; \mathbf{C})$  y  $f_y(x_i, y_i, z_i; \mathbf{C})$  para el modelo ortográfico de cámara, se definen como:

$$f_x(x_i, y_i, z_i; \mathbf{C}) \equiv b_1(\mathbf{C})x_i + b_2(\mathbf{C})y_i + b_3(\mathbf{C})z_i + b_4(\mathbf{C}) \quad (2.19)$$

$$f_y(x_i, y_i, z_i; \mathbf{C}) \equiv b_5(\mathbf{C})x_i + b_6(\mathbf{C})y_i + b_7(\mathbf{C})z_i + b_8(\mathbf{C}) \quad (2.20)$$

donde los términos  $b_1, \dots, b_8$  están determinados por una expresión no lineal que depende de los parámetros de visión:

$$\begin{aligned} b_1(\mathbf{C}) &= C_1^2 + C_2^2 - C_3^2 - C_4^2 \\ b_2(\mathbf{C}) &= 2(C_2C_3 + C_1C_4) \\ b_3(\mathbf{C}) &= 2(C_2C_4 - C_1C_3) \\ b_4(\mathbf{C}) &= C_5 \\ b_5(\mathbf{C}) &= 2(C_2C_3 - C_1C_4) \\ b_6(\mathbf{C}) &= C_1^2 - C_2^2 + C_3^2 - C_4^2 \\ b_7(\mathbf{C}) &= 2(C_3C_4 + C_1C_2) \\ b_8(\mathbf{C}) &= C_6 \end{aligned} \quad (2.21)$$

Si se expresa el equivalente matricial de la ecuación (2.22), queda dada por:

$$J(\mathbf{C}) = \Delta \mathbf{Y}^T \mathbf{W} \Delta \mathbf{Y} \quad (2.22)$$

donde los residuos  $\Delta Y$  se expresan como:

$$\Delta Y = \begin{bmatrix} x_{c1} - f_x(x_i, y_i, z_i; \mathbf{C}) \\ y_{c1} - f_y(x_i, y_i, z_i; \mathbf{C}) \\ \vdots \\ x_{cm} - f_x(x_i, y_i, z_i; \mathbf{C}) \\ y_{cm} - f_y(x_i, y_i, z_i; \mathbf{C}) \end{bmatrix} \quad (2.23)$$

Dado que las ecuaciones para estimar los parámetros de visión son no lineales, el proceso de estimación es iterativo, para lo que puede emplearse un procedimiento de estimación no lineal de corrección de mínimos cuadrados.

Iniciando en un valor inicial para los parámetros  $\mathbf{C}$  y una corrección de los mismos  $\Delta \mathbf{C}$  se evalúa al considerar una linealización del vector  $\Delta \mathbf{Y}$  alrededor de  $\mathbf{C}$  usando una expansión en series de Taylor como:

$$\Delta \mathbf{Y}^p = \Delta \mathbf{Y}^c - \mathbf{A} \Delta \mathbf{C} \quad (2.24)$$

$\Delta \mathbf{Y}^p$  puede considerarse como un vector de predicción de residuales al aplicar la corrección  $\Delta \mathbf{C}$ .

La matriz  $\mathbf{A}$  (de dimensiones  $2m \times 6$ ), puede obtenerse como:

$$A = \begin{bmatrix} x_1 & y_1 & z_1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x_1 & y_1 & z_1 & 1 \\ & & & f & & & & \\ x_m & y_m & z_m & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x_m & y_m & z_m & 1 \end{bmatrix} \begin{bmatrix} \frac{\partial b_1}{\partial C_1} & \cdots & \frac{\partial b_1}{\partial C_6} \\ \vdots & \ddots & \vdots \\ \frac{\partial b_8}{\partial C_1} & \cdots & \frac{\partial b_8}{\partial C_6} \end{bmatrix} \quad (2.25)$$

La corrección  $\Delta \mathbf{C}$  se obtiene al minimizar:

$$J'(\Delta \mathbf{C}) = \Delta \mathbf{Y}^p T \mathbf{W} \Delta \mathbf{Y}^p \quad (2.26)$$

Desarrollando la expresión, considerando  $\Delta \mathbf{Y}^p$ , la corrección esta dada por:

$$\Delta \mathbf{C} = [\mathbf{A}^T \mathbf{W} \mathbf{A}]^{-1} [\mathbf{A}^T \mathbf{W} \Delta \mathbf{Y}^C] \quad (2.27)$$

Una vez que se obtiene una minimización de (2.26), en el incremento  $\Delta \mathbf{C}$  es muy pequeño, se dice que los parámetros de visión han convergido, por lo que el método iterativo termina.

El algoritmo 1 muestra como determinar los parámetros de visión del sistema, el cual termina cuando el error de convergencia es menor a  $1 \times 10^{-5}$ .

---

**Algoritmo 1:** Determinación de los parámetros de visión.

---

**Data:**  $\mathbf{X}_c \mathbf{Y}_c \in \mathbb{R}^m$ ,  $\mathbf{XYZ} \in \mathbb{R}^{m \times 3}$ ,  $\mathbf{W} \in \mathbb{R}^{2m \times m}$ .

**Result:**  $\mathbf{C} \in \mathbb{R}^6$

- 1  $Y = \text{cargar\_observaciones}(\mathbf{X}_c \mathbf{Y}_c)$ ;
- 2  $Error = 1 \times 10^{-5}$ ;
- 3 **while**  $absoluto(Jc_{anterior} - Jc_{nuevo}) > Error$  **do**
- 4      $Y_{modelo} = \text{calcular\_modelo\_camara}(\mathbf{X}_c \mathbf{Y}_c)$ ;
- 5      $\mathbf{A} = \text{matriz\_jacobiana\_modelo}(\mathbf{C})$ ;
- 6      $\Delta Y = Y - Y_{modelo}$ ;
- 7      $Jc_{nuevo} = \Delta Y' \mathbf{W} \Delta Y$ ;
- 8      $\Delta \mathbf{C} = [\mathbf{A}' \mathbf{W} \mathbf{A}]^{-1} [\mathbf{A}' \mathbf{W} \Delta Y]$ ;
- 9      $\mathbf{C} = \mathbf{C} + \Delta \mathbf{C}$ ;

10 **return**  $\mathbf{C}$ ;

---

### 2.4.2. Determinación de la posición física de un punto observado

Una vez determinados los parámetros de visión de las cámaras utilizadas en el sistema de visión, es posible estimar la posición tridimensional de un punto observado en los planos de imagen de las cámaras. Es decir, se puede realizar una transformación del espacio de cámara al espacio de referencia tridimensional.

Para determinar la posición de uno o varios puntos objetivo  $P_i$ , es posible emplear la



metodología de estimación no lineal por mínimos cuadrados, de manera similar a como se realizó al principio de este apartado. A continuación se presenta la metodología para un sistema de visión con dos cámaras con parámetros de visión independientes  $\mathbf{C}_1$  y  $\mathbf{C}_2$ .

El vector de estimación está dado por la posición tridimensional  $(x_i, y_i, z_i)$  definida como:

$$\mathbf{XYZ} = [x_i, y_i, z_i]^T \quad (2.28)$$

las cuales se determinan al minimizar la función:

$$\begin{aligned} J(\mathbf{XYZ}) = & \sum_i^4 \left\{ [x_{c1} - f_x(\mathbf{XYZ}; \mathbf{C}_1)]^2 + [y_{c1} - f_y(\mathbf{XYZ}; \mathbf{C}_1)]^2 \right. \\ & \left. + [x_{c2} - f_x(\mathbf{XYZ}; \mathbf{C}_2)]^2 + [y_{c2} - f_y(\mathbf{XYZ}; \mathbf{C}_2)]^2 \right\} W_i \end{aligned} \quad (2.29)$$

donde:

- 4 := La sumatoria es de 1 a 4 debido a que se estima a partir de dos pares de coordenadas.
- $(x_{ci}, y_{ci})$  := coordenadas de la observación en la imagen de la cámara.
- $\mathbf{C}_1, \mathbf{C}_2$  := parámetros de visión de la cámara 1 y cámara 2, respectivamente.
- $W_i$  := Peso relativo positivo dado a la medición.

Las relaciones  $f_x(\mathbf{XYZ}; \mathbf{C}_1, \mathbf{C}_2)$  y  $f_y(\mathbf{XYZ}, z_i; \mathbf{C}_1, \mathbf{C}_2)$  para el modelo ortográfico de cámara se toman de las ecuaciones (2.19) y (2.20). En este caso el vector de residuos esta dado por:

$$\Delta Y = \begin{bmatrix} x_{c1} - f_x(\mathbf{XYZ}; \mathbf{C}_1) \\ y_{c1} - f_y(\mathbf{XYZ}; \mathbf{C}_1) \\ x_{c2} - f_x(\mathbf{XYZ}; \mathbf{C}_2) \\ y_{c2} - f_y(\mathbf{XYZ}; \mathbf{C}_2) \end{bmatrix} \quad (2.30)$$

Iniciando en una posición inicial, la cual debe ser coherente con el sistema de referencia de la escena, para el vector de posición  $\mathbf{XYZ}$  y una corrección de la misma  $\Delta \mathbf{XYZ}$ , se evalúa al considerar una linealización del vector  $\Delta Y$  alrededor de  $\mathbf{XYZ}$  como:

$$\Delta Y^p = \Delta Y^c - \mathbf{A} \Delta \mathbf{XYZ} \quad (2.31)$$

La matriz  $\mathbf{A}$  (de dimensiones 4x3), puede obtenerse a partir de los términos  $b_1, \dots, b_8$  de cada cámara como:

$$A = \begin{bmatrix} b_{11} & b_{21} & b_{31} \\ b_{51} & b_{61} & b_{71} \\ b_{12} & b_{22} & b_{32} \\ b_{52} & b_{62} & b_{72} \end{bmatrix}_{C_1, C_2} \quad (2.32)$$

La corrección  $\Delta\mathbf{XYZ}$  esta expresada como:

$$\Delta\mathbf{XYZ} = [A^T W A]^{-1} [A^T W \Delta Y^C] \quad (2.33)$$

El algoritmo 2 es utilizado para determinar la posición tridimensional para un punto  $P_i$ , se deben elegir condiciones iniciales para comenzar la estimación, éstas deben representar un punto que este dentro del espacio de trabajo del robot, definido por las muestras de la estimación de los parámetros de visión.

---

**Algoritmo 2:** Determinación de la posición tridimensional del punto  $P_i$ .

---

**Data:**  $\mathbf{X}_c, \mathbf{Y}_c \in \mathbb{R}^m$ ,  $\mathbf{C}_1, \mathbf{C}_2 \in \mathbb{R}^6$ ,  $\mathbf{W} \in \mathbb{R}^{4 \times 4}$ .

**Result:**  $\mathbf{XYZ} \in \mathbb{R}^6$

```

1  $Y = \text{cargar\_observaciones}(x_{c1}, y_{c1}, x_{c2}, y_{c2});$ 
2  $Error = 1 \times 10^{-5};$ 
3 while  $absoluto(Jc_{anterior} - Jc_{nuevo}) > Error$  do
4    $Y_{modelo} = \text{calcular\_modelo\_camara}(\mathbf{XYZ}, \mathbf{C}_1, \mathbf{C}_2);$ 
5    $\mathbf{A} = \text{matriz\_jacobiana\_modelo}(\mathbf{C}_1, \mathbf{C}_2);$ 
6    $\Delta Y = Y - Y_{modelo};$ 
7    $Jc_{nuevo} = \Delta Y' \mathbf{W} \Delta Y;$ 
8    $\Delta\mathbf{XYZ} = [\mathbf{A}' \mathbf{W} \mathbf{A}]^{-1} [\mathbf{A} \mathbf{W} \Delta Y];$ 
9    $\mathbf{XYZ} = \mathbf{XYZ} + \Delta\mathbf{C};$ 
10 return  $\mathbf{XYZ};$ 

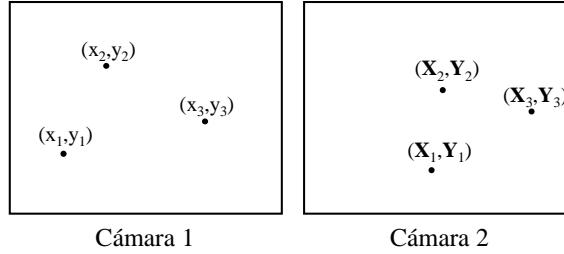
```

---

## 2.5. Correspondencia entre imágenes

Como se muestra en el capítulo anterior, se propone realizar la reconstrucción de superficies arbitrarias a través del método de manipulación de espacio de cámara. Empleando un modelo no lineal de cámara para determinar la posición tridimensional, de puntos objetivo en el espacio de cámara, con respecto al marco de referencia de un robot industrial. Sin embargo, el procedimiento mostrado hasta el momento, realiza el cálculo de puntos tridimensionales individuales, lo cual no es óptimo. Para ello se proyecta una matriz de puntos láser sobre la superficie a reconstruir. Así, la detección y reconstrucción de puntos se realiza en un solo muestreo. Para obtener las coordenadas tridimensionales de cada punto de la matriz láser en el marco del robot, es necesario hacer un etiquetado de cada punto detectado en el espacio de cámara, de manera tal que exista una relación uno a uno entre las coordenadas de cada punto, obtenidas por

las cámaras del sistema. Este problema es conocido como correspondencia entre imágenes, es decir, encontrar una relación entre las coordenadas de un punto visto por dos o más cámaras. Considerando la Fig. (2.2), se puede determinar una relación cuadrática



**Figura 2.2:** Puntos vistos por la cámara 1 y cámara 2 respectivamente.

entre las coordenadas  $(x_i, y_i)$  de la cámara 1 y las coordenadas  $(X_i, Y_i)$  de la cámara 2 como,

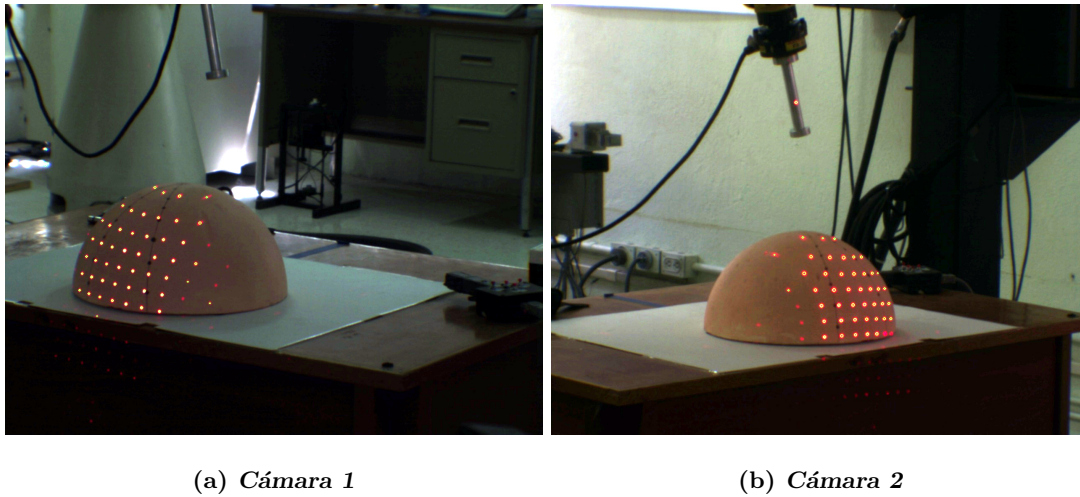
$$\begin{aligned} X_i &= ax_i^2 + by_i^2 + cx_i + dy_i + f \\ Y_i &= gx_i^2 + hy_i^2 + ix_i + jy_i + k \end{aligned} \quad (2.34)$$

donde,  $(x_i, y_i)$  y  $(X_i, Y_i)$  son las coordenadas de un punto en el espacio de cada una de las cámaras y los parámetros  $a, \dots, k$  describen la relación de correspondencia entre puntos de la imagen. Los parámetros de correspondencia se determinan al minimizar el funcional  $\Phi_c$ ,

$$\begin{aligned} \Phi_c &= \sum_{i=1}^N \left\{ (X_i - ax_i^2 - by_i^2 - cx_i - dy_i - f)^2 \right. \\ &\quad \left. + (Y_i - gx_i^2 - hy_i^2 - ix_i - jy_i - k)^2 \right\} \end{aligned} \quad (2.35)$$

donde,  $N$  es el número de puntos muestra para definir la relación entre coordenadas. La minimización de  $\Phi_c$  se realiza a través del método de mínimos cuadrados. En la Fig. (2.3) se muestra una proyección de una matriz de puntos sobre una sección de superficie esférica, en la cual se realiza el etiquetado con el método descrito en esta sección. En la Fig. (2.4) se observa el etiquetado de puntos de un experimento realizado. Los cuadros azules simbolizan puntos detectados en cada uno de los espacios de cámara del sistema, las etiquetas rojas determinan los puntos cuyas coordenadas tienen una correspondencia positiva a partir del método descrito.

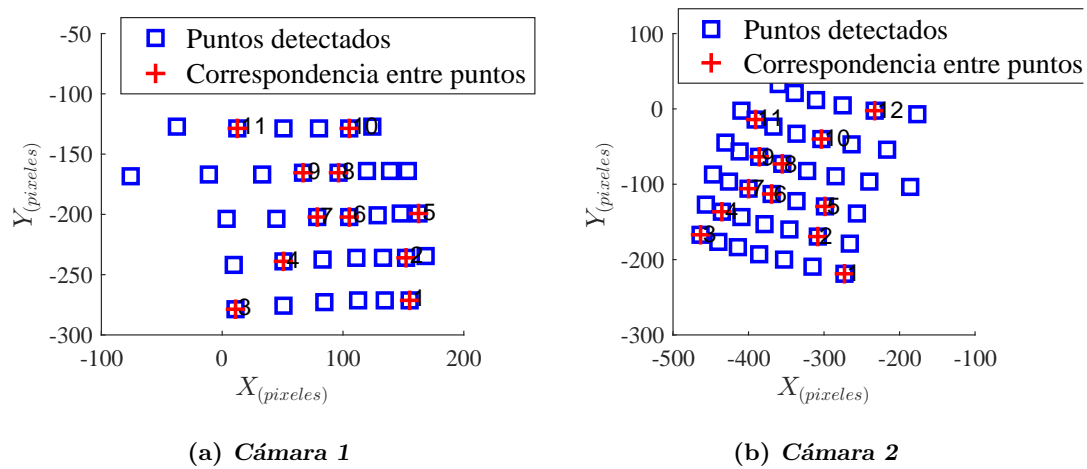
Una vez realizada la correspondencia entre los puntos detectados por el sistema de visión, es posible determinar la posición tridimensional de cada pareja de puntos empatados como se indica en la sección (2.4.2).



(a) Cámara 1

(b) Cámara 2

**Figura 2.3:** Imágenes de la proyección de la matriz de puntos láser vistos por el sistema de visión.



(a) Cámara 1

(b) Cámara 2

**Figura 2.4:** Resultado de la correspondencia entre las coordenadas de los puntos objetivos detectados.

---

## Capítulo 3

# Implementación del sistema de teleoperación

---

Como ya se ha mencionado, el objetivo principal de este trabajo es desarrollar un sistema de teleoperación que interactúe sobre superficies arbitrarias de objetos de trabajo que están dentro del espacio del robot usando una técnica de visión por computadora no calibrada para estimar dicha superficie. El sistema hará uso de un robot industrial de seis grados de libertad como dispositivo esclavo y una unidad háptica como dispositivo maestro.

En este capítulo se explica la integración del sistema de visión explicado en el capítulo (2), con los otros elementos que conforman el sistema de teleoperación. Se detalla el desarrollo de la metodología seguida para resolver el problema de asimetría cinemática que existe entre el robot esclavo y maestro, así como el cálculo de la orientación que debe tener la herramienta de trabajo para asegurar un movimiento perpendicular a la superficie reconstruida.

### 3.1. Elementos del sistema de teleoperación

En la Figura (3.1) se muestran los elementos que componen el sistema de teleoperación implementado en el Laboratorio de Robótica. Cada uno de los elementos de izquierda a derecha se describen a continuación:

- Unidad háptica: Phantom Omni, es un robot de seis grados de libertad, con retroalimentación de fuerza en tres de sus ejes y es utilizado como dispositivo o robot maestro.
- Computadora: Dell Optiplex, se encarga de realizar el procesamiento del sistema de teleoperación, realiza las estimaciones paramétricas, de posición y orientación. Se encarga de la comunicación entre cada uno de los elementos y el registro de acciones realizadas.

- Controlador del robot: modelo R-30iA, se encarga del control de posición del robot esclavo, tiene implementado el controlador de impedancia y recolecta toda la información del robot, así como la captada por el sensor de fuerza. Se comunica con un protocolo TCP/IP por medio de la red local del laboratorio.
- Robot Fanuc LR Mate 200iC: Es un manipulador serial de seis grados de libertad, que tiene montado en el efector final un sensor de fuerza, que a su vez funciona como herramienta, es el dispositivo esclavo en el sistema de teleoperación. Es de arquitectura cerrada, por lo que solo se tiene acceso al control por posición.
- Sensor de Fuerza: modelo FS-15iA, es un sensor de fuerza/par que funciona como herramienta del robot, el cual interacciona directamente con la superficie de un objeto de trabajo.
- Cámaras UEye: El sistema de visión esta integrado por dos cámaras uEye modelo UI-1540-C, a color. Son utilizadas para la adquisición de imágenes de la escena, las cuales se procesan para realizar la estimación de los parámetros de visión y la caracterización de la superficie arbitraria.

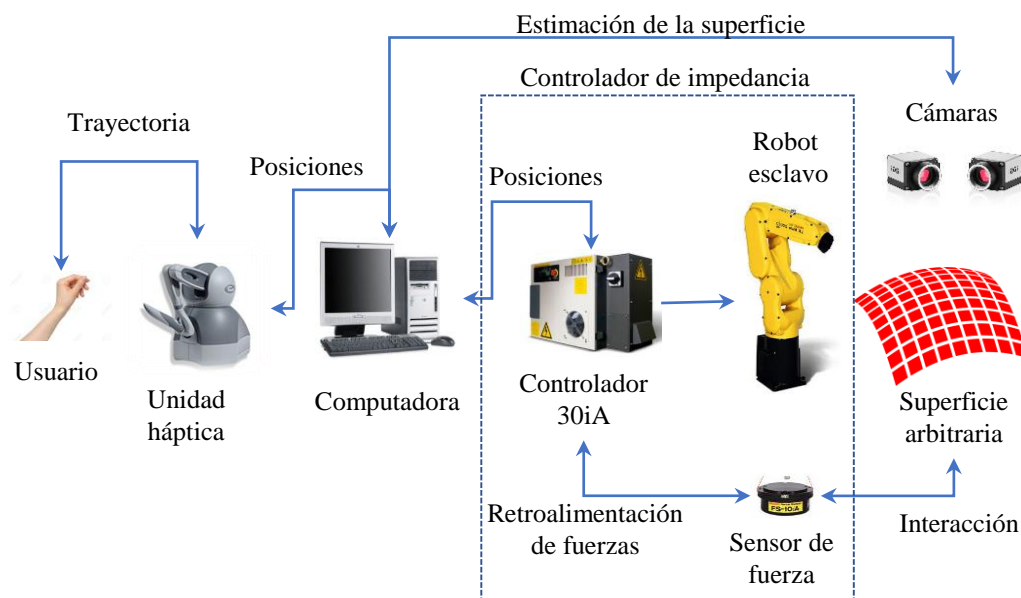


Figura 3.1: Diagrama del sistema de teleoperación.

### 3.2. Controlador de impedancia

Cuando el sistema de teleoperación realiza una tarea de seguimiento, donde el robot interactúa sobre la superficie de trabajo del objeto, el control clásico de posición y

velocidad del robot resulta insuficiente. Esto debido a que, además de los errores en la reconstrucción de la superficie y que no son conocidos en todo punto, las tareas de contacto pueden contener características inerciales, disipativas o elásticas, por lo que el control de movimiento puro es propenso a fallas. Una alternativa es controlar la dinámica del robot manipulador, además de la posición y velocidad.

El control de impedancia [16], es un modelo de interacción dinámica que impone al sistema un comportamiento de un sistema generalizado masa-resorte-amortiguador dado por,

$$\mathbf{F}_e = \mathbf{K}_d \mathbf{x} + \mathbf{B}_d \dot{\mathbf{x}} + \mathbf{M}_d \ddot{\mathbf{x}} \quad (3.1)$$

donde,  $\mathbf{F}_e \in \mathfrak{R}^m$  representa las fuerzas de interacción,  $\mathbf{M}_d \in \mathfrak{R}^{m \times m}$ ,  $\mathbf{K}_d \in \mathfrak{R}^{m \times m}$  y  $\mathbf{B}_d \in \mathfrak{R}^{m \times m}$  son las matrices de inercia, rigidez y amortiguamiento respectivamente, las cuales son diagonales definidas positivas. Los vectores  $\mathbf{x} \in \mathfrak{R}^m$ ,  $\dot{\mathbf{x}} \in \mathfrak{R}^m$  y  $\ddot{\mathbf{x}} \in \mathfrak{R}^m$  representan la posición, velocidad y aceleración respectivamente, respecto al marco de coordenadas del robot.

Con el propósito de implementar un controlador de impedancia en un robot manipulador industrial de arquitectura cerrada, se desarrollan algoritmos de control cinemático de impedancia. Lawrence [17], presenta un enfoque de control de impedancia basado en posición, donde las fuerzas y los pares son explícitamente medidos usando un sensor de fuerza en la herramienta del robot. Con base en las fuerzas y pares de contacto, se obtiene un vector de posición con el que se ajusta la posición del robot y la nueva configuración se envía al lazo de control interno. Al considerar  $\mathbf{x}$  en (3.1), como un vector de ajuste de posición obtenido por un filtro de interacción de fuerzas, entonces,

$$\mathbf{x} = [\mathbf{K}_d(s) + \mathbf{B}_d(s) + \mathbf{M}_d(s)]^{-1} \mathbf{F}_e(s) \quad (3.2)$$

Así, la nueva posición  $\mathbf{x}_r \in \mathfrak{R}^m$  está dada por,

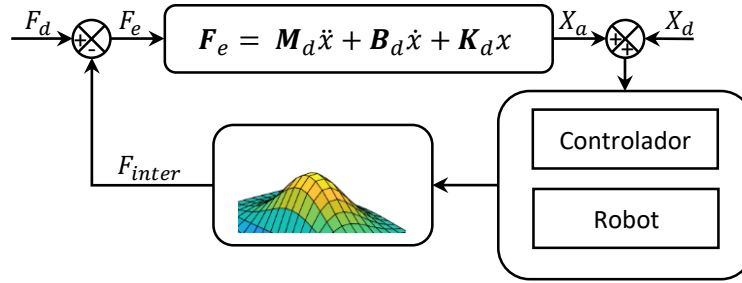
$$\mathbf{x}_r = \mathbf{x}_d - \mathbf{x} \quad (3.3)$$

donde,  $\mathbf{x}_d \in \mathfrak{R}^m$  representa la trayectoria deseada, la cual es calculada a partir de la trayectoria definida por el operador que se traslada a la superficie arbitraria a partir de las ecuaciones (3.7) que incluyen a los denominados parámetros de asimetría  $a, \dots, r$  definidos en la sección 3.3.1. Finalmente, la trayectoria de referencia es seguida por el efector final del manipulador usando el control de posición implementado por el fabricante dentro del robot industrial utilizado. Cabe mencionar que la entrada de fuerza al sistema  $F_e$  esta definida por la fuerza de referencia o fuerza deseada ( $F_d$ ) menos la fuerza de interacción con el entorno ( $F_{inter}$ ) como se puede apreciar en la ecuación (3.4).

$$F_e = F_d - F_{inter} \quad (3.4)$$

En la Fig. (3.2) se muestra el diagrama a bloques del sistema de teleoperación implementado con el controlador de impedancia.

Como se mencionó antes, se trata de imponer una dinámica de un sistema masa-resorte-amortiguador al robot esclavo, este comportamiento se basa en la selección de



**Figura 3.2:** Diagrama de bloques del control de impedancia del sistema de teleoperación.

los parámetros del controlador de impedancia. Al ser un sistema de segundo orden, el factor de amortiguamiento  $\zeta$  define el tipo de respuesta que tendrá el sistema. El valor de  $\zeta$  se determina con la ecuación (3.5) y los tipos de respuesta que se presentan se describen en la Tabla (3.1).

$$\zeta = \sqrt{\frac{B_d^2}{4M_dK_d}} \tag{3.5}$$

Caso	Tipo de respuesta
$\zeta > 1$	Respuesta sobre amortiguada, se distingue por no presentar oscilaciones ante una entrada de impulso o un escalón.
$\zeta = 1$	Respuesta críticamente amortiguada, no presenta oscilaciones al igual que el tipo anterior, sin embargo su implementación es complicada debido a los errores numéricos.
$\zeta < 0$	Respuesta subamortiguada, presenta oscilaciones al ser sometido a entradas del tipo impulso o escalón.
$\zeta = 0$	Se presentan oscilaciones permanentes en el sistema.

**Tabla 3.1:** Tipos de respuesta de un sistema de segundo orden.

La determinación de parámetros se explica con detalle en [14], donde los valores de parámetros de masa, amortiguamiento, rigidez y factor de amortiguamiento, además de las condiciones iniciales asignadas para la solución, se muestran en la Tabla (3.2). Dichos valores son propuestos a partir del comportamiento del sistema durante pruebas

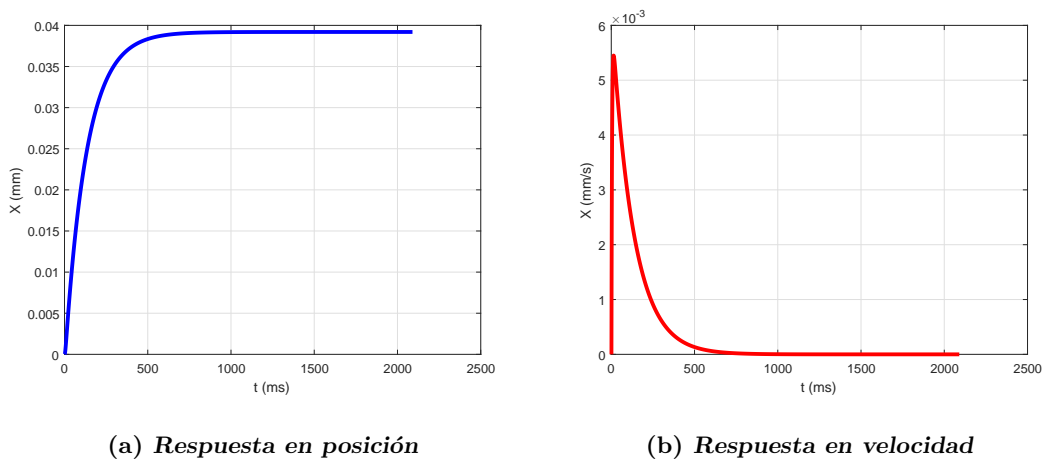


$M_d$	$B_d$	$K_d$	$\zeta$	$x_a(0)$	$\dot{x}_a(0)$
286	1666.7	250	3.11123	0	0

**Tabla 3.2:** Valores de los parámetros y condiciones iniciales para el controlador de impedancia.

experimentales del controlador de impedancia.

La respuesta del controlador, con los parámetros mencionados, ante una entrada de fuerza de 1 N se muestran en la Figura (3.3). Se aprecia del lado izquierdo de la figura, la respuesta del cambio de posición y velocidad de la herramienta en el eje  $X$  ante la fuerza propuesta. Se comprueba que la respuesta es sobreamortiguada, dado el valor de  $\zeta$ . En el capítulo 4 se muestra la respuesta del controlador interactuando con superficies.



**Figura 3.3:** Respuesta del controlador de impedancia en el eje  $X$ .

### 3.3. Asimetría cinemática

En esta sección se muestra una solución al problema de asimetría cinemática el cual es debido a que los dispositivos a utilizar como maestro y esclavo tiene una estructura mecánica diferente, por lo que las trayectorias definidas en el maestro no son mapeadas de manera directa al esclavo. Esta diferencia en la estructura mecánica de los dispositivos hace que los espacios de trabajo y los sistemas coordenados sean diferentes, por lo que es necesario definir un mapeo que estime las trayectorias que tiene que seguir el robot esclavo a partir de trayectorias trazadas por un usuario en el robot maestro.

Una vez realizada la caracterización de la superficie, se definen las superficies de trabajo para el dispositivo maestro (unidad háptica) y el esclavo (robot), estableciendo un mapeo entre ellos. Para la unidad, a través de la cual se define una trayectoria, la superficie de trabajo es un plano que sirve de referencia. En el caso de la superficie de trabajo del robot esclavo, ésta se define acotando una región de la superficie arbitraria caracterizada como se describe en el capítulo 2. Así, el mapeo es de una superficie plana a una geometría arbitraria.

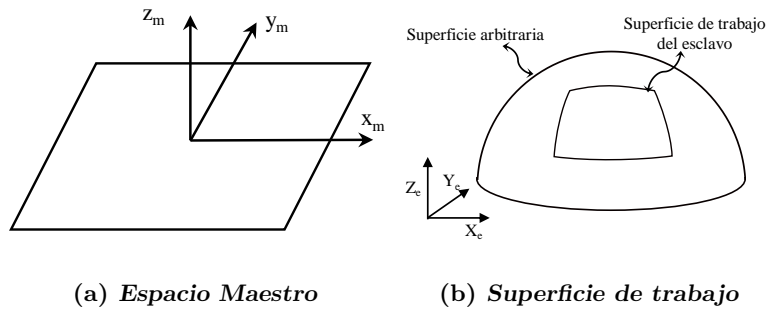
### 3.3.1. Mapeo entre superficies de trabajo

La Fig. [3.4] muestra los sistemas de coordenadas  $(x_m, y_m, z_m)$  del plano maestro y  $(X_e, Y_e, Z_e)$  del esclavo. Es posible definir una relación cuadrática entre ambos sistemas como,

$$\begin{aligned} X_{e_i} &= f_{x_a}(a, b, c, d, e, f) \\ Y_{e_i} &= f_{y_a}(g, h, i, j, k, l) \\ Z_{e_i} &= f_{z_a}(m, n, o, p, q, r) \end{aligned} \quad (3.6)$$

con,

$$\begin{aligned} f_{x_a} &= ax_{m_i}^2 + by_{m_i}^2 + cx_{m_i}y_{m_i} + dx_{m_i} + ey_{m_i} + f \\ f_{y_a} &= gx_{m_i}^2 + hy_{m_i}^2 + ix_{m_i}y_{m_i} + jx_{m_i} + ky_{m_i} + l \\ f_{z_a} &= mx_{m_i}^2 + ny_{m_i}^2 + ox_{m_i}y_{m_i} + px_{m_i} + qy_{m_i} + r \end{aligned} \quad (3.7)$$



**Figura 3.4:** Espacios de trabajo típicos definidos en una relación maestro-esclavo.

donde,  $(x_{m_i}, y_{m_i}, z_{m_i})$  y  $(X_{e_i}, Y_{e_i}, Z_{e_i})$  son las coordenadas de un punto en plano maestro y esclavo, respectivamente. Los parámetros de asimetría  $a, \dots, r$  describen la relación entre coordenadas de los planos. La determinación de los parámetros de asimetría se realiza al minimizar el funcional  $\Phi_a$ ,

$$\Phi_a = \sum_i^{N_a} \left\{ [X_{e_i} - f_{x_a}]^2 + [Y_{e_i} - f_{y_a}]^2 + [Z_{e_i} - f_{z_a}]^2 \right\} \quad (3.8)$$

donde  $N_a$  es el número de muestras que definen el área de trabajo en los planos. Las funciones  $(f_{x_a}, f_{y_a}, f_{z_a})$  se representan en (3.7). La minimización de  $\Phi_a$  se realiza a partir del método de mínimos cuadrados, de manera análoga a la obtención de los parámetros de visión, de forma tal que el resultado de la minimización son los vectores de parámetros de asimetría cinemática, los cuales definen el mapeo descrito al inicio de esta sección y se representan como:

$$\mathbf{P}_x = \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix} \quad \mathbf{P}_y = \begin{bmatrix} g \\ h \\ i \\ j \\ k \\ l \end{bmatrix} \quad \mathbf{P}_z = \begin{bmatrix} m \\ n \\ o \\ p \\ q \\ r \end{bmatrix} \quad (3.9)$$

La selección de las  $N_a$  muestras que definen el mapeo, se proponen de manera tal que se limiten los espacios de trabajo a partir de nueve puntos y que estén relacionados uno a uno en ambos espacios. En la Figura (3.5) se muestra la definición y relación uno a uno de nueve puntos.

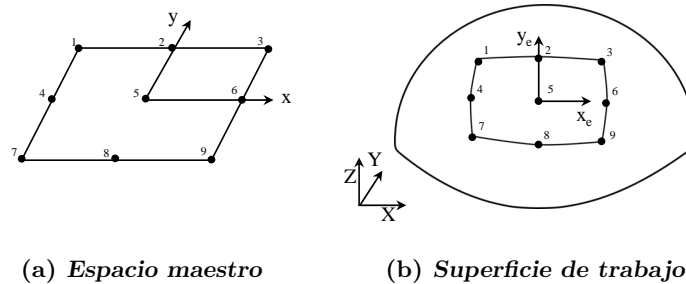


Figura 3.5: Definición de los espacios de trabajo a partir de nueve puntos.

### 3.3.2. Movimiento perpendicular a la superficie estimada

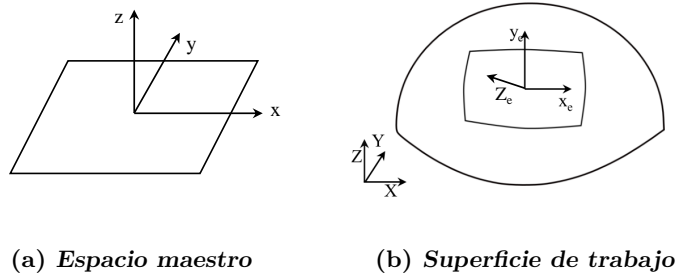
El mapeo descrito transforma trayectorias descritas sobre el plano maestro hasta su correspondiente trayectoria en el plano esclavo. Cuando el dispositivo háptico se mueve en dirección del eje  $z_m$ , el mapeo no considera este desplazamiento. Se observa que el desplazamiento en  $z_m$  es perpendicular al plano maestro por lo que, para propósitos de establecer una correspondencia entre los espacios de trabajo, se considera un desplazamiento perpendicular a la superficie reconstruida. Al ser una superficie arbitraria, se

considera un plano que se ajuste a los puntos muestra usados para la reconstrucción (ver la sección 3.4). El plano queda definido como,

$$Ax_e + By_e + Cz_e + D = 0 \tag{3.10}$$

donde  $A, B, C, D$  son los parámetros que definen al plano de ajuste y  $(x_e, y_e, z_e)$  representan las coordenadas de los puntos muestra de la superficie reconstruida (ver Figura (3.6)). Así, el vector normal a este plano está dado por,

$$v = [ A \quad B \quad C ] \tag{3.11}$$



**Figura 3.6:** Sistemas coordenados de los espacios de trabajo.

Cuando el maestro se desplace en dirección del eje  $z_m$ , el movimiento del esclavo en dirección al eje  $Z_e$  es  $k$  veces la coordenada  $z_{m_i}$ . Por lo que un punto  $i$ , con coordenadas  $(x_{m_i}, y_{m_i}, z_{m_i})$  se mapea como,

$$\begin{aligned} Z_{e_i} &= kz_{m_i} \\ X_{e_i} &= x_{m_i} + AZ_{e_i} \\ Y_{e_i} &= y_{m_i} + BZ_{e_i} \\ Z_{e_i} &= z_{m_i} + CZ_{e_i} \end{aligned} \tag{3.12}$$

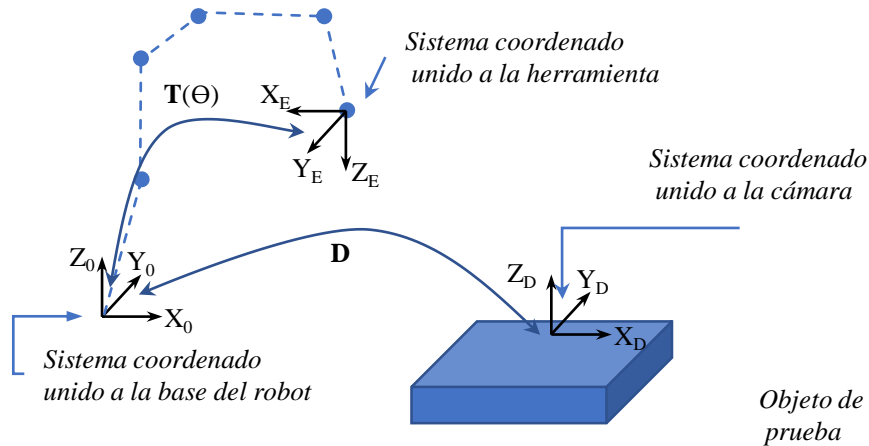
donde  $(X_{e_i}, Y_{e_i}, Z_{e_i})$  son las coordenadas del  $i$ -ésimo punto descrito por en una trayectoria en el plano maestro. De esta manera, el movimiento en dirección al eje  $z_m$  se mapea en un movimiento perpendicular al plano de ajuste de la superficie reconstruida.

### 3.4. Cálculo de la orientación de la herramienta

Para determinar la orientación de la herramienta del robot, se consideran tres sistemas coordenados como se muestra en la Figura (3.7). Uno de ellos es el sistema asociado a la base del robot, otro es el sistema unido a la herramienta y el tercero es el sistema asociado a la superficie del objeto de trabajo. Donde  $D$  es una transformación entre

la posición y orientación de la herramienta sobre la superficie de trabajo, en la que se considera la misma orientación de la superficie de interés para el posicionamiento de la herramienta del robot. Como se muestra en el capítulo 2, es posible realizar la estimación de posición de la herramienta hasta este punto, sin embargo, es necesario realizar el cálculo de orientación de la herramienta para que se posicione perpendicular a la superficie de trabajo. Para ello se define una matriz  $\mathbf{D}$  que describe la orientación de la herramienta con respecto a la superficie de trabajo.

Como se menciona, para esta aplicación se requiere que la herramienta se mueva perpendicular a la superficie estimada, por lo que el eje  $Z_e$  del sistema coordenado de la herramienta (ver Figura (3.7)) es coincidente con el eje  $Z_D$  del sistema coordenado asociado a la superficie. Considerando que el eje  $Z_D$  es en realidad el vector normal a la superficie estimada, el eje  $Z_E$  es también coincidente con el vector  $\hat{N}$ .



**Figura 3.7:** Relación entre los sistemas coordenados.

Para determinar el vector normal a la superficie estimada, se considera la ecuación del plano, la cual es definida en (3.13).

$$Ax + By + Cz + D' = 0 \quad (3.13)$$

Para simplificar el cálculo del vector normal se considera que  $D' = -1$ . Así el vector normal al plano se define como (3.14).

$$\hat{N} = \frac{1}{\sqrt{A^2 + B^2 + C^2}}(A\hat{i} + B\hat{j} + C\hat{k}) \quad (3.14)$$

Para determinar los valores  $A$ ,  $B$  y  $C$  que definen al plano en la ecuación (3.13), se modela a partir un plano sobre la estimación de la superficie de trabajo obtenida en

el capítulo (2). De manera que para obtener los parámetros del plano se minimiza la función escalar  $\phi_P$  como se muestra en la ecuación (3.15).

$$\phi_P = \sum_i^{N_P} (Ax_i + By_i + Cz_i - 1)^2 \quad (3.15)$$

donde  $N_P$  es el número de puntos que definen el plano de la superficie. Para realizar la minimización se deben cumplir las siguientes condiciones (3.16).

$$\frac{\partial \phi_P}{\partial A} = 0 \quad \frac{\partial \phi_P}{\partial B} = 0 \quad \frac{\partial \phi_P}{\partial C} = 0 \quad (3.16)$$

desarrollando, se tiene que la estimación de los parámetros que definen el plano se obtiene de la ecuación (3.17).

$$\begin{bmatrix} A \\ B \\ C \end{bmatrix} = \mathbf{M}_P^{-1} \mathbf{S} \quad (3.17)$$

donde  $\mathbf{M}_P$  se considera una matriz no singular definida como:

$$\mathbf{M}_P = \begin{bmatrix} \sum x_i^2 & \sum x_i y_i & \sum x_i z_i \\ \sum x_i y_i & \sum y_i^2 & \sum y_i z_i \\ \sum x_i z_i & \sum y_i z_i & \sum z_i^2 \end{bmatrix} \quad (3.18)$$

y el vector  $\mathbf{S}$  definido por:

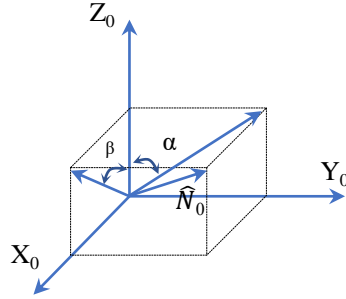
$$\mathbf{S} = \begin{bmatrix} \sum x_i \\ \sum y_i \\ \sum z_i \end{bmatrix} \quad (3.19)$$

A partir de los valores de  $A$ ,  $B$  y  $C$  en la ecuación (3.17), se pueden obtener el vector normal a la superficie estimada  $\hat{N}$  a partir de la ecuación (3.14). Sin embargo la dirección del vector  $\hat{N}$  no esta definida, con la restricción de que  $Z_e$  debe ser coincidente con el vector normal, se debe satisfacer que  $V_n \cdot \hat{N} < 0$ , de no ser así  $\hat{N} = -\hat{N}$ . Donde el vector  $V_n$  depende de los parámetros de visión, y representa a un vector que sale del origen del sistema asociado a la cámara, el cual se define en la ecuación (3.20) en función de los ángulos de Euler.

$$V_n = \begin{bmatrix} -2(e_2 e_4 + e_1 e_3) \\ -2(e_3 e_4 + e_1 e_2) \\ -e_1^2 + e_2^2 + e_3^2 + e_4^2 \end{bmatrix} \quad (3.20)$$

Para determinar la matriz de rotación que asegura que el vector  $\hat{N}$  coincide con el eje  $Z_e$ , se muestra en la Figura (3.8) los ángulos  $\alpha$  y  $\beta$  que debe ser rotado el sistema.

De esta manera la matriz de rotación que asegura esta orientación está definida como:



**Figura 3.8:** Componentes del vector normal a la superficie estimada.

$$Rot_{x,-\alpha}Rot_{y,\beta} = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) & 0 \\ -\sin(\alpha)\sin(\beta) & \cos(\alpha) & \sin(\alpha)\cos(\beta) & 0 \\ -\cos(\alpha)\sin(\beta) & -\sin(\alpha) & \cos(\alpha)\cos(\beta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.21)$$

donde los ángulos  $\alpha$  y  $\beta$  se calculan con:

$$\begin{aligned} \alpha &= \arctan\left(\frac{N_y}{N_z}\right) \\ \beta &= \arctan\left(\frac{N_x}{N_z}\right) \end{aligned} \quad (3.22)$$

donde  $N_x$ ,  $N_y$  y  $N_z$  son las componentes del vector  $\hat{N}$ . Así, la matriz  $\mathbf{D}$ , queda definida por:

$$\mathbf{D} = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) & X_e \\ -\sin(\alpha)\sin(\beta) & \cos(\alpha) & \sin(\alpha)\cos(\beta) & Y_e \\ -\cos(\alpha)\sin(\beta) & -\sin(\alpha) & \cos(\alpha)\cos(\beta) & Z_e \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.23)$$

Los puntos objetivo deben estar definidos con respecto al sistema coordenado dado por  $D$ , y son calculados como:

$$\begin{bmatrix} x'_i \\ y'_i \\ z'_i \\ 1 \end{bmatrix} = \mathbf{D}^{-1} \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} \quad (3.24)$$

donde los puntos  $(x'_i, y'_i, z'_i)$  están definidos con respecto a la base del robot (que son los puntos estimados en (2)) y  $(x_i, y_i, z_i)$  con respecto al sistema definido en la superficie de trabajo.

Sin embargo, la orientación que debe ser enviada al robot, debe estar especificada en la convención conocida como ángulos de balanceo, inclinación y orientación (en inglés *roll*, *pitch*, *yaw* o *wpr*). Esta convención está definida por la rotación (3.25).

$$Rot_{z,w}Rot_{y,p}Rot_{x,r} = \begin{bmatrix} cwcp & cwspsr - swcr & cwspr + swsr \\ swcp & swspsr + cwcr & swspr - cwsr \\ -sp & cp & cpcr \end{bmatrix} \quad (3.25)$$

La determinación de los valores de los ángulos *wpr* se realiza al igualar la matriz de rotación (3.25) con la matriz de rotación dada por  $D^{-1}$ .

### 3.5. Integración de los elementos del sistema de teleoperación

El desarrollo del sistema de teleoperación se basó en el paradigma de programación orientada a objetos y el diseño de interfaces gráficas de usuario en Visual C++. En la Fig. 3.9 se muestra la manera en la que interactúa un usuario (operador) con cada uno de los componentes del sistema de teleoperación (sistema de visión, unidad háptica y robot Fanuc) para realizar una tarea de seguimiento de trayectoria sobre una superficie objetivo. Este diagrama muestra las actividades que realiza el sistema para un funcionamiento correcto. Cada una de las tareas que realiza cada uno de los elementos está bien definida y especificada en el programa a partir de cada una de las clases desarrolladas (ver la definición de cada una en los Anexos A).

El proceso para realizar una tarea de teleoperación en la cual se interactúa con un objeto de trabajo se realiza en las siguientes etapas:

- Inicia sistema de teleoperación: El usuario inicia el programa para comenzar el seguimiento de una trayectoria sobre un objeto.
- Inicialización del sistema de visión: Se inicia la comunicación con las cámaras uEye, se define el espacio de trabajo para determinar los parámetros de visión (en caso de que la estimación haya sido realizada anteriormente este paso se elimina), se coloca un objeto en el espacio de trabajo definido y se inicia la estimación de la superficie arbitraria del objeto.
- Asimetría cinemática: Una vez caracterizada, la superficie es enviada para la estimación de los parámetros de asimetría, los cuales mapean cada uno de los puntos recibidos al espacio del esclavo.
- Generación de trayectoria: El usuario presiona el Botón 1 de la unidad háptica (ver Figura (3.10)) y comienza a trazar una trayectoria dentro del espacio del maestro, si el Botón 1 no es presionado, no es trasladada al robot esclavo. Esto como un medio de incrementar la seguridad del sistema.



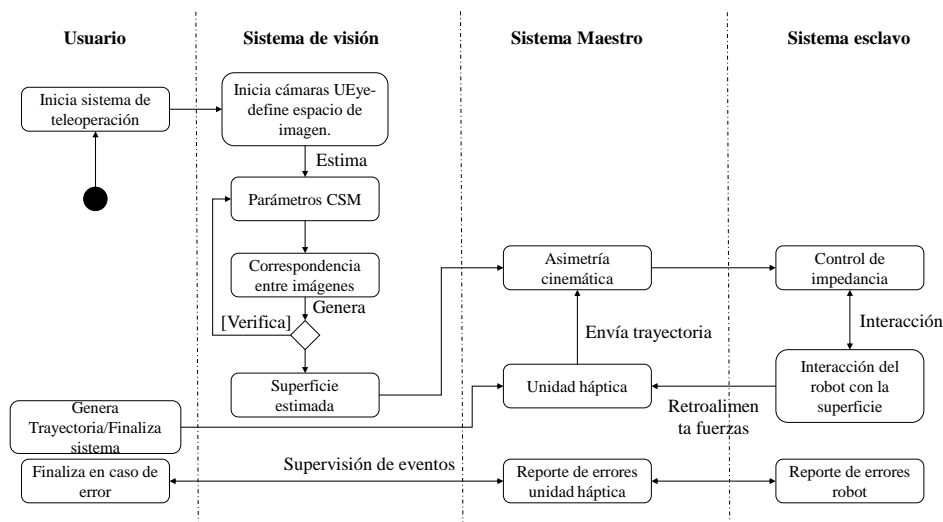


Figura 3.9: Diagrama de actividad del sistema de teleoperación.

- Mapeo de trayectoria: El mapeo de un punto, definido en espacio de trabajo del maestro, es realizado uno a uno si existe un cambio de posición del robot maestro en un lapso de 50 ms, es decir, el periodo de muestreo de posición y mapeo es de 50 ms.
- Interacción con la superficie: Definida la trayectoria deseada a partir del mapeo de la etapa anterior, ésta es enviada al controlador del robot, el cual tiene implementado el controlador de impedancia descrito anteriormente. Las correcciones de posición de la trayectoria estimada se realiza dependiendo de las fuerzas percibidas por la herramienta y medidas por el sensor de fuerza.
- Finalizar sistema: El sistema puede terminar si suceden dos casos, el primero es que el usuario detenga el sistema de teleoperación debido a que no hay más tareas por realizar, y el segundo es en caso de error en alguno de los subsistemas que impida seguir realizando una tarea de seguimiento.

Las clases permiten describir los elementos presentes en el sistema y definir las tareas que deben cumplir. En la Fig. 3.11 se observan las nueve clases elementales que componen al sistema de teleoperación y la manera en la que interaccionan cada una de ellas. Los atributos y métodos de cada una de las clases se describen en los Anexos.

Durante la ejecución del sistema de teleoperación, existen tareas que deben realizarse paralelamente. Una vez que se está realizando una tarea de seguimiento el sistema debe tener la capacidad de leer las posiciones descritas por la unidad háptica, estimar y enviar la posición deseada al robot fanuc y además, estar ejecutando la interfaz gráfica actualizando los datos mostrados y revisando posibles errores durante el tiempo de ejecución.

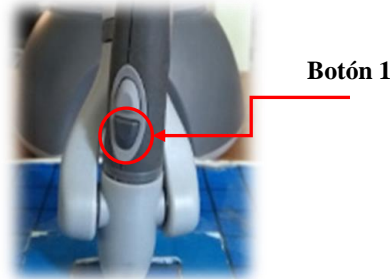


Figura 3.10: Botón 1: inicia el mapeo de puntos del maestro al esclavo.

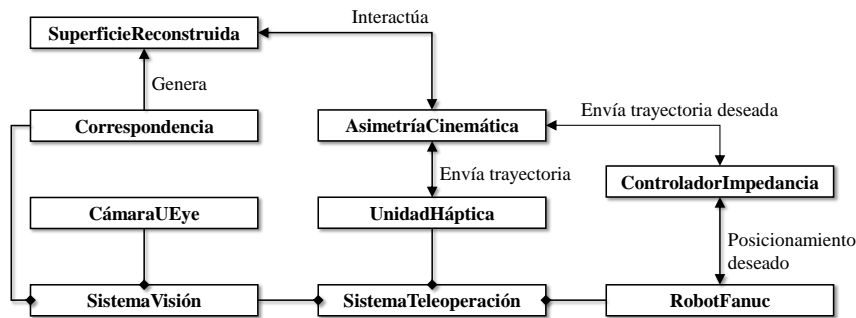


Figura 3.11: Diagrama de clases del sistema de teleoperación.

En la Fig. 3.12 se muestran los hilos (tareas que se realizan en paralelo) que controlan los elementos del sistema. El hilo esclavo se encarga de monitorizar la comunicación con el robot Fanuc, enviar las posiciones deseadas, la lectura de fuerzas y el registro de datos. El hilo maestro se encarga del procesamiento de trayectorias descritas por el operador y la estimación de la trayectoria deseada, además de realizar el registro y monitorización de datos. El hilo principal, se encarga del control y monitorización de la interfaz gráfica de usuario y los otros dos hilos. Es el encargado de controlar los errores ocurridos durante el tiempo de ejecución y comunicarse con el operador.



**Figura 3.12:** Ejecución del sistema de teleoperación.



---

## Capítulo 4

# Resultados experimentales

---

En los capítulos anteriores se presentaron los fundamentos teóricos y los algoritmos desarrollados durante la implementación del sistema de teleoperación. En este capítulo se muestran los resultados experimentales de las pruebas realizadas al sistema, donde se busca validar la estimación de la superficie arbitraria, el seguimiento de trayectorias definidas por el operador y el desempeño del controlador de impedancia cuando interactúa con el objeto. Se plantean pruebas con diferentes características para probar el funcionamiento del sistema de visión, la precisión en la estimación y evaluación de la correspondencia entre imágenes de puntos objetivo sobre la superficie arbitraria. También se evalúa el seguimiento correcto de trayectorias estimadas por el método de asimetría cinemática planteado y el comportamiento del controlador de impedancia con el entorno, esto con la ayuda de diferentes objetos de geometría y superficie diferente.

### 4.1. Elementos del sistema de teleoperación

La plataforma experimental desarrollada para la implementación del sistema de teleoperación descrito en los capítulos anteriores, está compuesta por los elementos que se observan en la Figura (3.1). El equipo mostrado se encuentra ubicado en el Laboratorio de Robótica de la Facultad de Ingeniería. El dispositivo esclavo es un Robot Fanuc LR Mate 200 iC, el cual es un robot manipulador serial de seis grados de libertad montado sobre una estructura que mantiene sus base en posición vertical. El controlador de dicho robot es un Fanuc R-30ia mate, el cual cuenta con un puerto de comunicación Ethernet y soporta el protocolo TCP/IP para el desarrollo de aplicaciones que requieran monitorizar o controlar el robot a través de una red LAN. Como efector final del robot se encuentra montado un sensor de fuerza/par de la marca Fanuc modelo FS 10iA que funciona como herramienta de contacto con el entorno.

El dispositivo maestro es una unidad háptica Phantom Omni desarrollada por la empresa Sensable, al igual que el robot anterior es un manipulador serial de seis grados de libertad. Tiene la capacidad de transmitir fuerza en los tres primeros ejes, ideal para que el operador tenga una retroalimentación de fuerza. El protocolo de comunicación

del dispositivo es FireWire 1394 y cuenta con bibliotecas desarrolladas en C/C++ compatibles con el entorno de desarrollo integrado (IDE) Visual Studio 2010 (VS2010).

El sistema de visión esta compuesto por dos cámaras USB uEye UI-1540-C, cuentan con una resolución de 1280 por 1024 pixeles y están colocadas a una distancia aproximada de 5 metros del espacio de trabajo del robot. Al igual que con la unidad háptica, cuenta con bibliotecas desarrolladas en C/C++ para la adquisición de imágenes. El procesamiento de las imágenes se realiza con OpenCV versión 2.4.13, que es un conjunto de bibliotecas de código abierto para el desarrollo de aplicaciones de visión por computadora.

La implementación de los algoritmos se realizó en una computadora de escritorio Dell Optiplex 745, con un procesador Pentium IV a 3.6 GHz. El proyecto generado se desarrolló en el lenguaje Visual C++, en el IDE Visual Studio 2012, con compatibilidad para las bibliotecas de VS2010 donde se programaron las clases necesarias para la implementación del sistema de teleoperación descrito en capítulos anteriores.

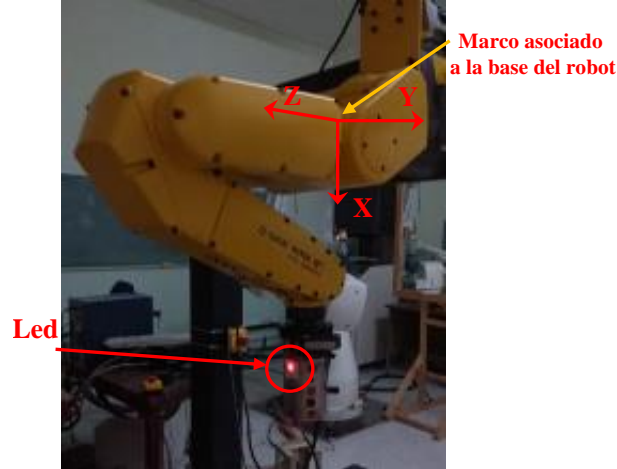
## 4.2. Estimación de la superficie arbitraria

Una vez programados los algoritmos descritos en el capítulo 2 para determinar la estimación de una superficie de un objeto prueba, se determina la precisión del método de reconstrucción de superficie arbitraria. Para ello se realizaron pruebas en la estimación paramétrica de los parámetros de visión descritos en la sección (2.4.1) y posteriormente en la precisión del cálculo de posición de puntos objetivo con respecto al espacio de trabajo del robot. Además, se realizaron pruebas al algoritmo de correspondencia descrito en la sección (4.2.1) para la estimación de múltiples puntos proyectados sobre un objeto, con la finalidad de realizar la estimación de su superficie.

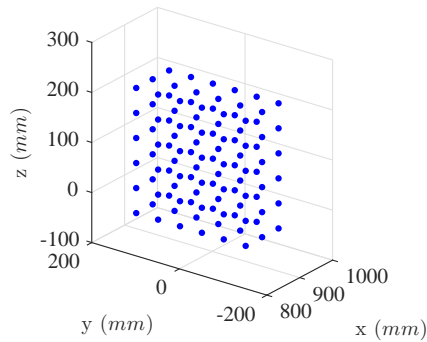
### 4.2.1. Estimación de los parámetros de visión

Para determinar los seis parámetros de visión  $C_1, \dots, C_6$  del método de manipulación de espacio de cámara se implementó el algoritmo descrito en el capítulo 2. Se consideraron un total de cien muestras para la estimación paramétrica y un espacio de trabajo de 300 mm de largo, 300 mm de ancho y 150 mm de alto. Cada una de las muestras consiste en la posición tridimensional  $(X_i, Y_i, Z_i)$ , con respecto al sistema coordinado del robot, de un LED montado en el efector final del robot como se muestra en la Figura (4.1). A partir de un algoritmo de visión por computadora se determinan las coordenadas  $(x_{c_i}, y_{c_i})$  del centroide del LED, en el espacio de cámara. Esto se hace para cada una de las cámaras que conforman el sistema de visión. En la Figura (4.2) se observan los puntos utilizados como muestras para la estimación paramétrica, también se pueden observar las dimensiones del espacio de trabajo del robot a utilizar.

A partir de el espacio definido, se obtienen los centroides de todas las muestras en el espacio de cámara, esto para las dos cámaras que conforman el sistema de visión.



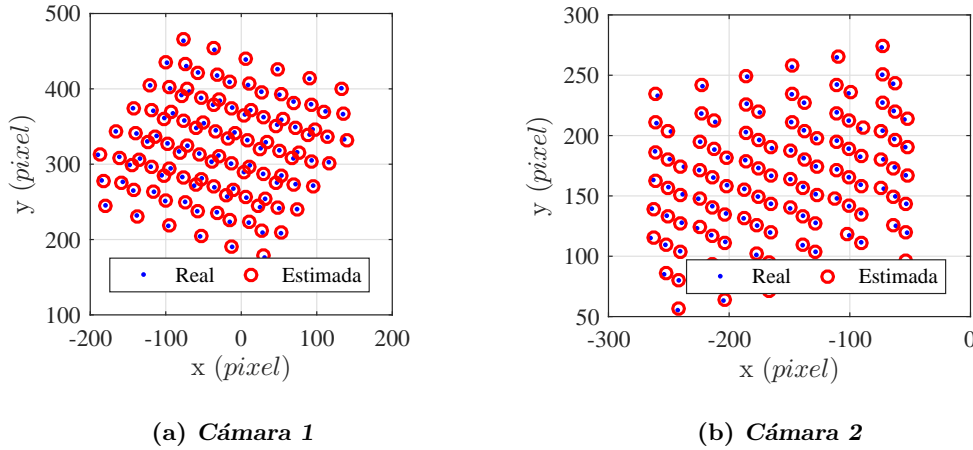
**Figura 4.1:** Led montado en el efector final del robot Fanuc LR Mate 200iC.



**Figura 4.2:** Puntos muestra que definen el espacio de estimación paramétrica.

Una vez que se tienen las coordenadas  $(X_i, Y_i, Z_i)$ ,  $(x_{c1_i}, y_{c1_i})$  y  $(x_{c2_i}, y_{c2_i})$  se realiza la estimación paramétrica de los parámetros de visión a partir del método de mínimos cuadrados. En la Figura (4.3) se muestran en azul los puntos muestra con los que se realizó la estimación y en rojo las estimaciones de los puntos a partir de los parámetros calculados. Los valores numéricos de los parámetros  $C_1, \dots, C_6$  para cada una de las cámaras de muestran en la Tabla (4.1).

Para comprobar que los parámetros de visión estimados son correctos, se realiza el proceso inverso para estimar las posiciones tridimensionales del centroide del LED, con respecto al marco coordenado del robot. Si las posiciones estimadas corresponden con las respectivas coordenadas de cada muestra utilizada para calcularla, quiere decir que los parámetros de visión son válidos para realizar la estimación de posición con respecto al marco del robot. En la Figura (4.4) se muestran los puntos muestra con sus



**Figura 4.3:** Resultado de la estimación de parámetros de visión en el espacio de cámara.

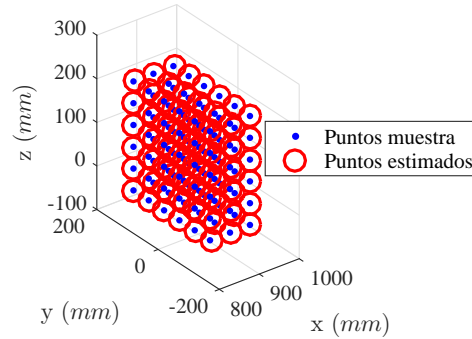
Cámara	$C_1$	$C_2$	$C_3$	$C_4$	$C_5$	$C_6$
1	0.57623329	971.6940	-0.42148474	-0.10001	0.65584629	-42.8051829
2	0.63008	0.3095789	-0.4782483	0.240676	-408.8843	680.129

**Tabla 4.1:** Valores de los parámetros de visión estimados para cada una de las cámaras. -Estos son los valores utilizados para estimar la posición tridimensional de un punto objetivo en el espacio del robot.

respectivas estimaciones, el error promedio entre las coordenadas reales y la estimadas es menor a  $1.3\text{ mm}$ , que se considera apropiado para la aplicación de reconstrucción de superficie planteada en este trabajo.

En la Tabla (4.2) se muestra el error medio de un total de 30 experimentos de estimación de coordenadas tridimensionales de una marca visual. El error es la diferencia entre las coordenadas reales de la marca visual, medida con ayuda del robot, y la estimación realizada por el método de manipulación en espacio de cámara descrito. En este caso, el error medio obtenido durante los experimentos es menor a  $2.3\text{ mm}$  con una desviación estándar de  $0.21\text{ mm}$ .





**Figura 4.4:** Comparación entre los puntos muestra y los puntos estimados por los parámetros calculados.

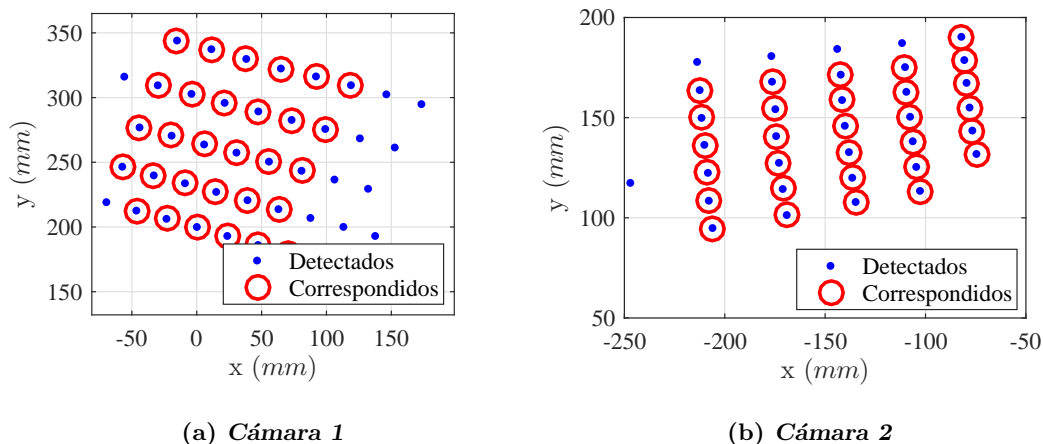
Error medio			
Eje X	Eje Y	Eje Z	Magnitud
1.4 mm	1.4 mm	1.4 mm	2.3 mm

**Tabla 4.2:** Error medio en la estimación paramétrica.

#### 4.2.2. Prueba de correspondencia entre imágenes

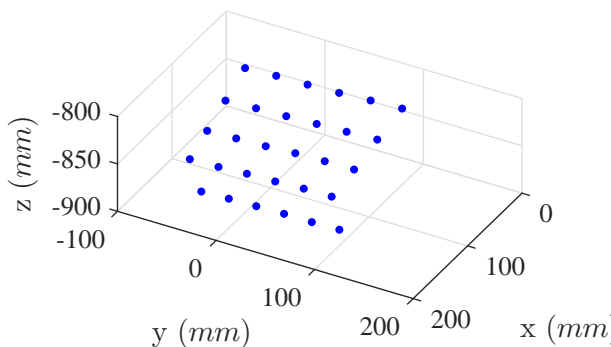
Una vez comprobada la precisión en la estimación de puntos objetivo dentro del espacio de trabajo del robot, se realiza un prueba para garantizar que el método de correspondencia entre imágenes funciona correctamente. Para ello se proyectó una matriz de puntos láser sobre un objeto plano, con la intención de hacer corresponder los puntos detectados en el espacio de las dos cámaras. En la clase `cUeye`, se programó un algoritmo para detectar los puntos láser proyectados sobre la superficie, el cual devuelve las coordenadas de los centroides los objetivos detectados. Para obtener los parámetros de correspondencia, se hacen corresponder manualmente cinco puntos del espacio de la cámara uno, con sus similares en la cámara dos. Estos funcionan como coordenadas muestra para el método de mínimos cuadrados, el cuál estima dichos parámetros. En la Figura (4.5) se muestran en azul los puntos láser detectados en cada una de las cámaras. Los puntos rojos, son los puntos que tienen una coordenada correspondiente en los espacios de cámara, es decir simbolizan las correspondencias positivas que se determinaron a partir de los parámetros estimados.

Teniendo las correspondencias positivas, que relacionan los puntos láser en los dos espacios de cámara, se estima la posición tridimensional de cada uno de ellos. Con estas estimaciones se reconstruye de manera aceptable la superficie del objeto prueba. En la Figura (4.6) se observa la superficie estimada a partir de la correspondencia entre



**Figura 4.5:** Correspondencia de puntos objetivo en cada una de las cámaras.

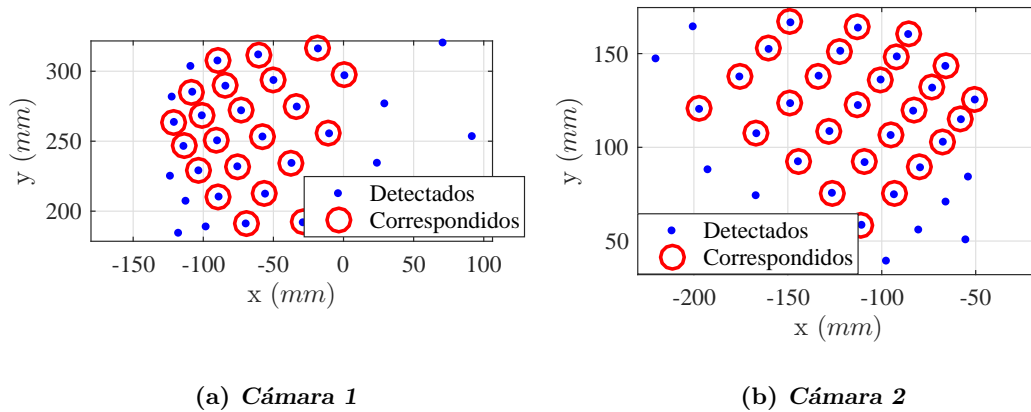
imágenes y la estimación tridimensional de los puntos. En este experimento, el número de puntos correspondidos es de treinta, por lo que la superficie del objeto prueba se estima con la misma cantidad de puntos.



**Figura 4.6:** Estimación de las coordenadas  $(X, Y, Z)$  de los puntos objetivo correspondidos.

Se realiza una segunda prueba para validar que no solo se realizan correspondencias entre objetos planos, ya que se trata de determinar superficies de cualquier geometría con curvas suaves (que es una restricción del método descrito en este trabajo) y que no presente protuberancias que interfieran con la proyección de puntos láser. Esta prueba consiste en proyectar la matriz de puntos láser sobre una sección de una esfera. En la

Figura (4.7) se muestran los puntos detectados sobre la superficie esférica en cada una de las cámaras, donde los puntos rojos representan los puntos correspondidos entre los dos espacios de cámara.

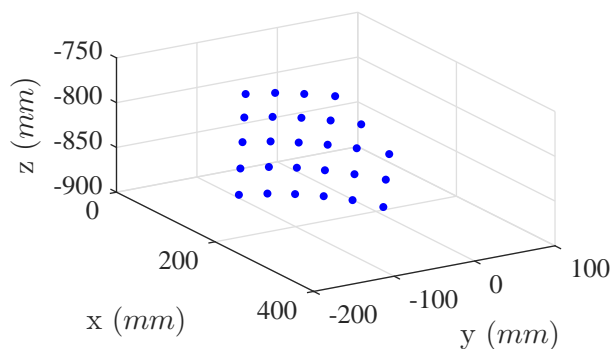


**Figura 4.7:** Correspondencia de puntos objetivo en cada una de las cámaras.

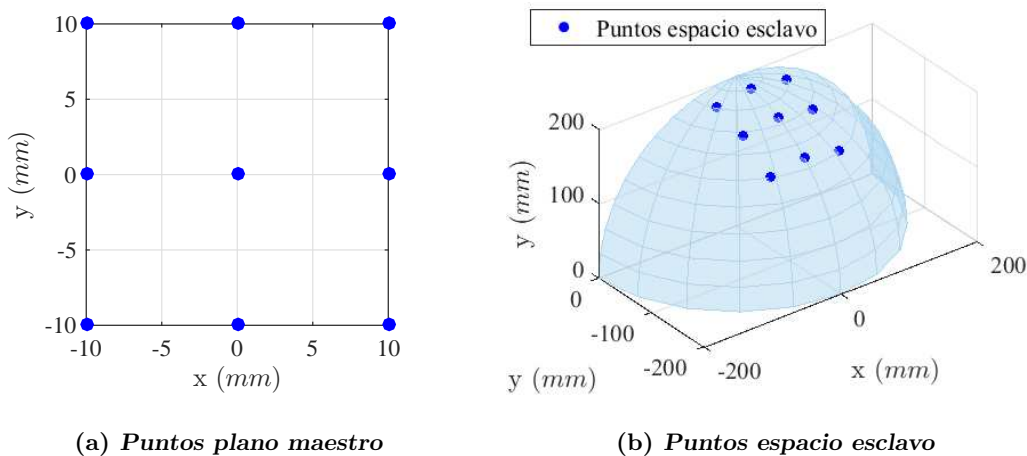
En este experimento se obtuvieron un total de veintisiete puntos con correspondencia positiva. Con las coordenadas de los centroides, de cada una de las cámaras, se estiman las coordenadas tridimensionales con respecto a la base del robot. El resultado de esta estimación se observa en la Figura (4.8), donde se aprecia la forma de una sección esférica.

### 4.2.3. Pruebas de la metodología propuesta para asimetría cinemática

En esta sección se busca probar el método de asimetría cinemática propuesto e implementado en este trabajo, el cual consiste en calcular los parámetros de asimetría que permiten estimar la trayectoria que debe seguir el esclavo, a partir de una trayectoria trazada por un usuario con el dispositivo maestro. Se implementó el algoritmo propuesto en el capítulo 2, para lo cual se proponen nueve puntos que definan los espacios de trabajo, del maestro y el esclavo. En la Figura (4.9) se muestran los espacios de trabajo definidos para el maestro y el esclavo respectivamente. El espacio de trabajo del maestro consiste en un cuadrado de 20 mm de ancho por 20 mm de alto, sobre el plano XY de la unidad háptica. El espacio de trabajo del esclavo es una sección de una esfera, como se observa en la figura mencionada.



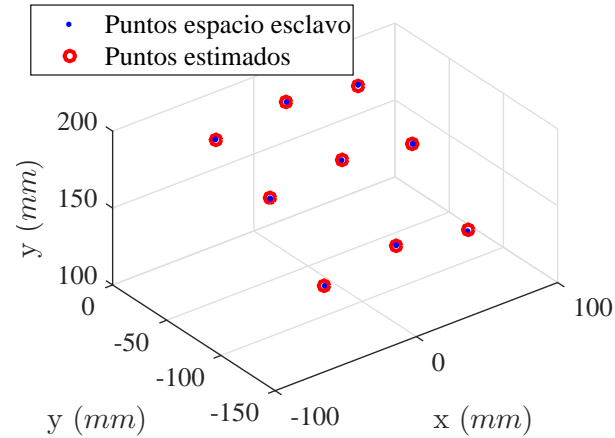
**Figura 4.8:** Estimación de las coordenadas  $(X, Y, Z)$  de los puntos objetivo correspondidos.



**Figura 4.9:** Puntos que definen los espacios de trabajo para la asimetría cinemática.

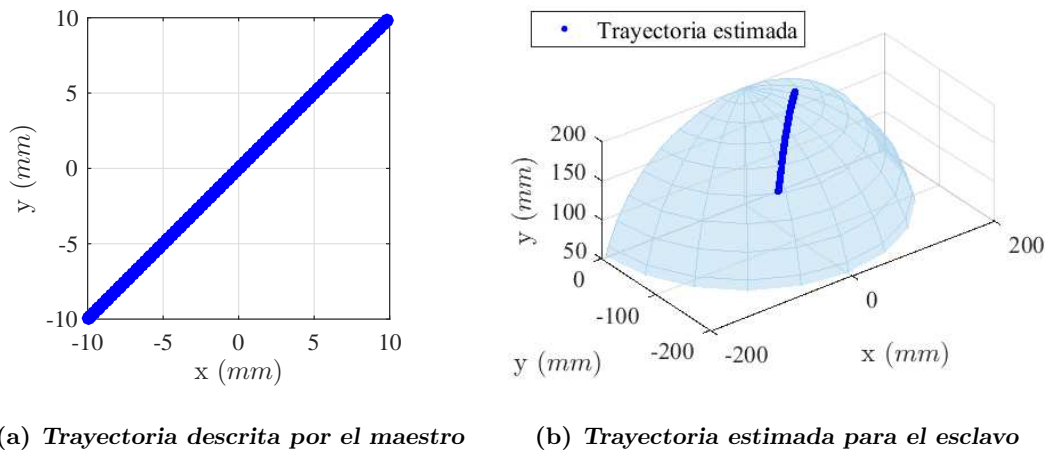
Una vez definidos los espacios de trabajo, se estiman los valores de los parámetros de asimetría con el método de mínimos cuadrados. En esta prueba el error medio de los puntos estimados es de  $1.3 \text{ mm}$ . En la Figura (4.10) se muestran los puntos utilizadas para definir el espacio del esclavo y las estimaciones de esos mismos puntos, calculadas a partir de los parámetros de asimetría obtenidos.

Se propone trazar una línea recta que cruce diagonalmente el cuadrado del plano del maestro propuesto, y estimar la línea que debe ser seguida por el esclavo. En la Figura (4.11) se aprecian las líneas trazadas en cada uno de los espacios de trabajo, se



**Figura 4.10:** Comparativa entre los puntos que definen el plano esclavo y su estimación.

observa que la línea estimada para el esclavo es proyectada sobre la superficie propuesta de forma correcta.



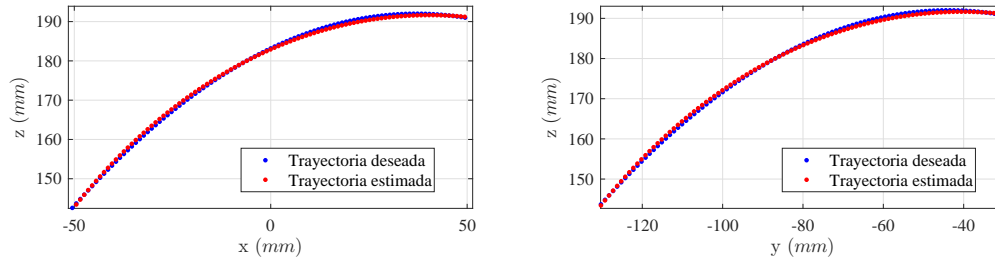
(a) *Trayectoria descrita por el maestro*

(b) *Trayectoria estimada para el esclavo*

**Figura 4.11:** Estimación de una línea trazada en el plano maestro y trasladada al esclavo.

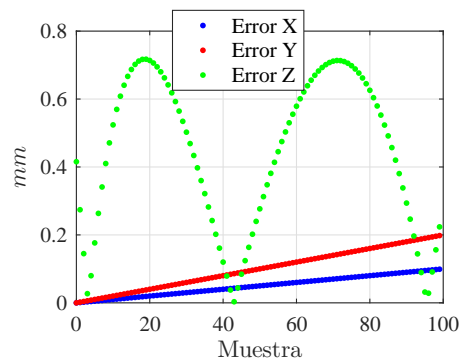
Adicionalmente, la comparación entre la línea estimada y la real que debe seguir el esclavo sobre la superficie, se observa en la Figura (4.12). Se asegura que el mapeo de las trayectorias se realiza de manera correcta, ya que las líneas están prácticamente superpuestas.

El error entre la trayectoria real y la estimada se muestra en la Figura (4.13), dónde se aprecia que el error en ejes Y y Z es despreciable, en el eje X es donde se presenta



**Figura 4.12:** Comparación entre la trayectoria deseada a seguir por el esclavo y la estimada.

el mayor error, sin embargo es muy pequeño para la aplicación que se requiere. Lo que indica que el método para resolver el problema de asimetría cinemática es capaz de estimar trayectorias sobre el espacio de trabajo del esclavo.



**Figura 4.13:** Error relativo de la estimación de la trayectoria en cada uno de los ejes.

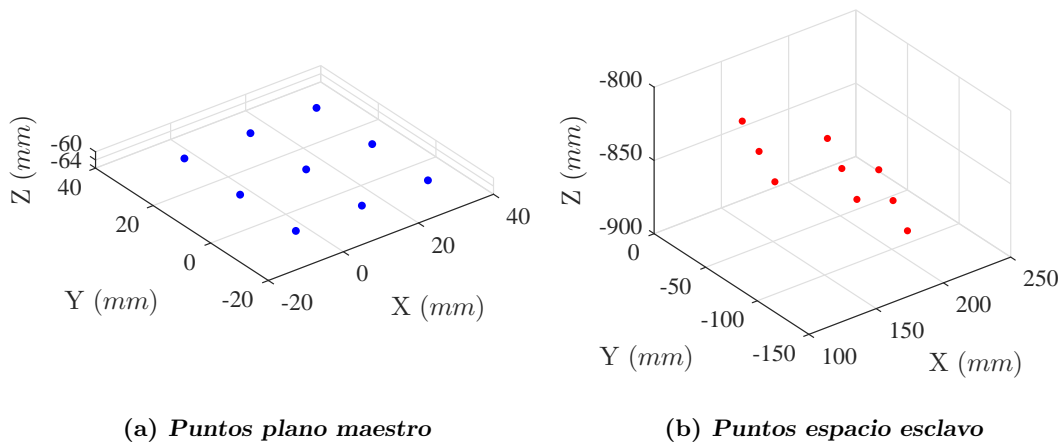
El error mostrado en la Figura (4.13) corresponde al error absoluto entre la trayectoria estimada a partir de los parámetros de asimetría y la real que debería seguir sobre una sección de una esfera perfecta. El error absoluto se puede define en la ecuación (4.1).

$$\varepsilon_r = | Valor_{Real} - Valor_{Estimado} | \tag{4.1}$$

En la Figura (4.13) se observa un comportamiento que asemeja al error absoluto en el eje Z a una función sinusoidal, debido a que la trayectoria estimada pasa sobre y bajo la superficie de trabajo, como se observa en la gráfica izquierda de la Figura (4.12).

Se realiza una segunda prueba, en la que se tiene como objetivo validar que no solo se estima correctamente una trayectoria sobre una superficie conocida, sino que también

los parámetros de asimetría estiman correctamente trayectorias sobre una superficie arbitraria. Para ello se definen los espacios de trabajo del maestro y esclavo con nueve puntos. El espacio maestro es un cuadrado de  $20\text{ mm}$  de lado. Para el espacio esclavo se proyectó una matriz de puntos láser sobre un objeto con forma cilíndrica, se seleccionan el mismo número de puntos, de los objetivos proyectados. En la Figura (4.14) se aprecian los espacios de trabajo definidos para este experimento.



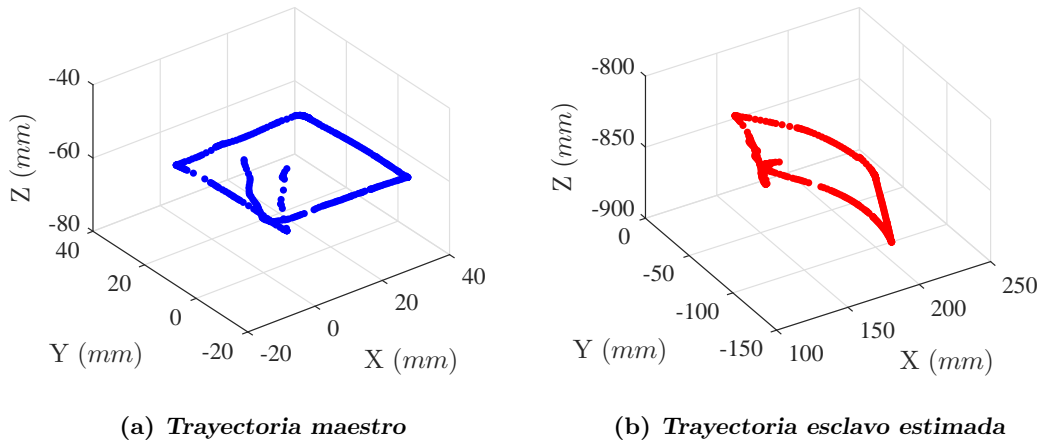
**Figura 4.14:** Puntos que definen los espacios de trabajo para la sección cilíndrica.

Se traza una trayectoria en el dispositivo maestro, en la cual se trata de emular un cuadrado de  $20\text{ mm}$  de lado, sobre el plano maestro definido. En la Figura (4.15) se muestra en azul el trayecto generado en la unidad háptica y en rojo la trayectoria estimada a partir de los parámetros de asimetría calculados para este experimento. Se aprecia que los puntos rojos asemejan un cuadrado dibujado sobre la sección de un cilindro.

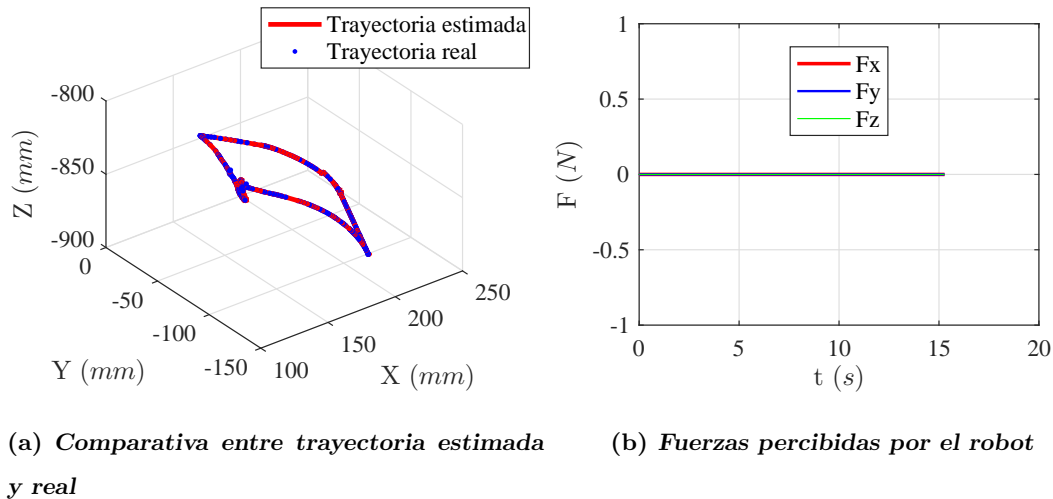
Con el fin de validar que la trayectoria estimada por el método de asimetría es correcto, se envían las coordenadas tridimensionales al robot para que realice esa trayectoria. Los resultados se muestran en la Figura (4.16). La gráfica de la izquierda muestra la trayectoria estimada y la real que siguió el robot durante la prueba, en la derecha se observan las fuerzas percibidas por el sensor de fuerza, las cuales son despreciables por ser muy cercanas a cero. Con esto se valida el funcionamiento del método propuesto para resolver el problema de asimetría cinemática.

### 4.3. Pruebas del sistema de teleoperación

En esta sección se evalúa el controlador de impedancia cuando el robot interactúa con superficies estimadas, las cuales son reconstruidas de diferentes objetos. Con el objetivo de observar su comportamiento y validar el hecho de que estimar la superficie



**Figura 4.15:** Trayectorias descritas por el dispositivo maestro y esclavo.



**Figura 4.16:** Comparativa entre las trayectoria estimada por el método de asimetría y la real descrita por el robot.

lo mejora, en comparación de plantear una superficie conocida para caracterizar un objeto de prueba. El mejoramiento del comportamiento también es posible ajustando los parámetros de impedancia ( $\mathbf{M}_d$ ,  $\mathbf{B}_d$  y  $\mathbf{K}_d$ ), los cuales compensan irregularidades que existen en la superficie reconstruida. La selección de estos parámetros se realiza con base en los trabajos [8], [29] y [14], los cuales presentan resultados mediante simulación y experimentales. A partir de estas referencias se concluye que el mejor comportamiento dinámico impuesto al robot, es la respuesta sobre amortiguada del sistema, la cual



mejora la interacción con la superficie e inclusive en interacciones humano-robot.

$\mathbf{M}_d$	$\mathbf{B}_d$	$\mathbf{K}_d$	$\zeta$
286	1666.7	250	3.1123

**Tabla 4.3:** Valores de los parámetros del control de impedancia.

Los valores utilizados para los parámetros del control de impedancia se muestran en la Tabla [4.3], que son los mismo valores utilizados en [14], los cuales son calculados tomando en cuenta que, para obtener una respuesta sobreamortiguada,  $\zeta$  debe ser mayor a uno.

Con los resultados obtenidos en los trabajos previos, se tiene la certeza del funcionamiento del control de impedancia. Se proponen una serie de pruebas para determinar que se mejora el comportamiento del control de impedancia durante la interacción al estimar la superficie de diferentes objetos de prueba.

#### 4.3.1. Pruebas sobre un objeto plano

En este experimento se busca probar el controlador de impedancia sobre una superficie plana estimada, para ello se colocó una libreta en el espacio de trabajo del robot (cuyos parámetros de visión se estimaron en la sección 4.2.1). Se proyecta la matriz de puntos láser y se obtienen las coordenadas tridimensionales que le corresponden a cada marca visual. Se realiza una trayectoria en el dispositivo maestro, que describe un cuadrado de 20 mm por lado, como se muestra en la Figura (4.17). En esta misma se muestra la trayectoria que se debe seguir sobre la superficie plana, ya que se seleccionan nueve puntos que definen ese espacio de trabajo para el esclavo.



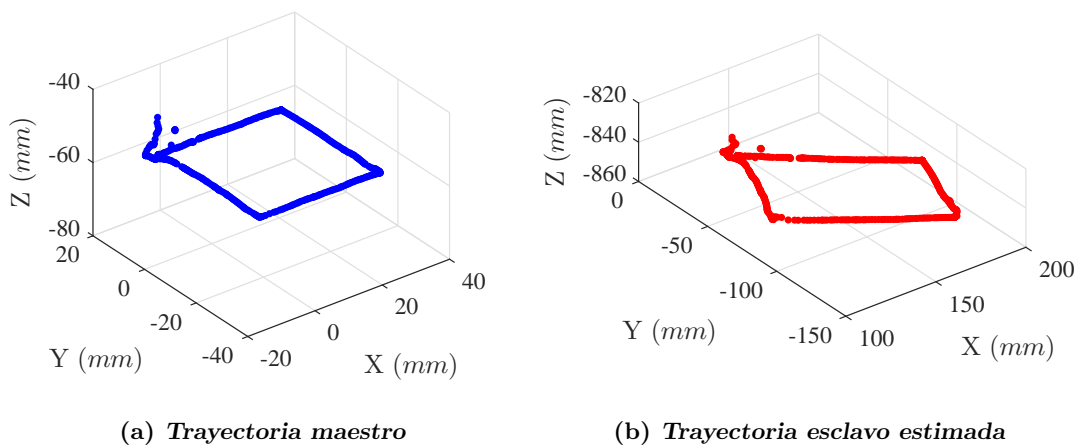
(a) *Plano maestro*

(b) *Espacio esclavo*

**Figura 4.17:** Trayectorias propuestas para el seguimiento.

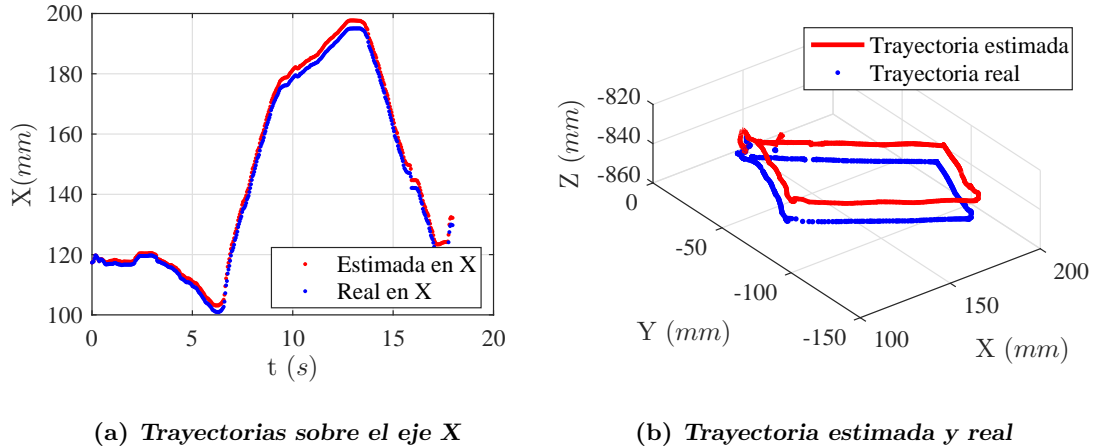
Al realizar el experimento se obtienen las trayectorias descritas por los dispositivos.

En la Figura (4.18) se muestra en azul el trazo realizado por la unidad háptica, a partir del cual se estima el trazo rojo, que es la trayectoria deseada a seguir por el robot. Con el fin de asegurar el contacto en todo momento, a la coordenada del eje  $X$  (referido al marco de coordenadas asociado a la base del robot como se muestra en la Figura (4.1)) estimada se le agregan  $10\text{ mm}$ , con lo que el efector final avanza una profundidad mayor en este eje.



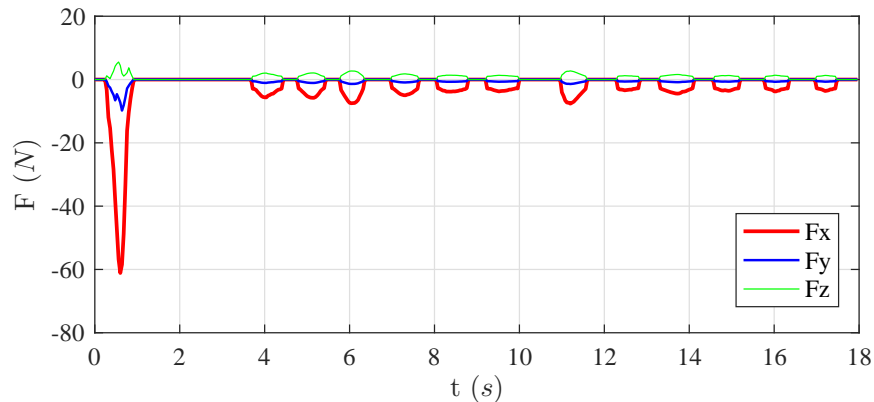
**Figura 4.18:** Trayectoria descrita por el dispositivo maestro y esclavo sobre un plano estimado.

La comparación entre la trayectoria deseada y la descrita por el robot se muestra en la Figura (4.19). En la gráfica de la derecha se comparan los puntos estimados y seguidos por el robot en el eje  $X$ , en la derecha se observa la trayectoria real seguida por el robot, la cual cumple con lo esperado. Las diferencias entre las trayectorias son debidas a la interacción que se tiene entre el robot y el objeto de prueba, por lo que se observa que el control de impedancia esta respondiendo conforme a lo esperado.



**Figura 4.19:** Comparativa entre las trayectoria estimada y la real trazada sobre la superficie plana estimada.

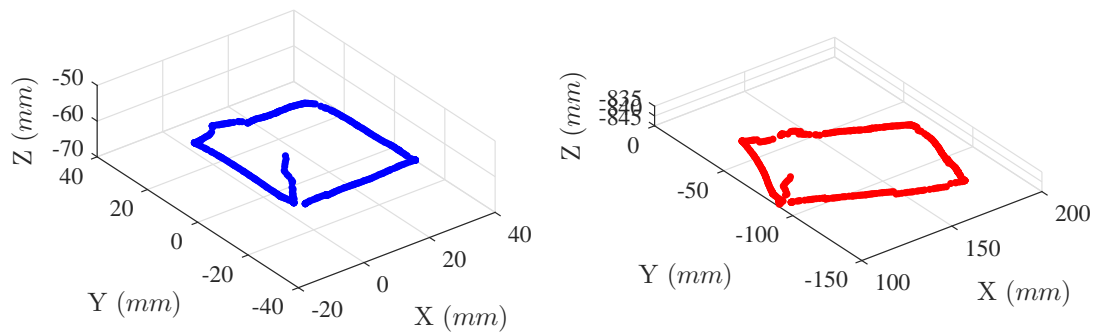
Las fuerzas de interacción que percibe el efector final del robot se aprecian en la Figura (4.20), donde se observa que la respuesta del controlador de impedancia ante esta interacción. Se tiene un pico de fuerza cercano a los  $60\text{ N}$  (el signo es debido a la dirección de la fuerza) en el eje  $X$ , el cual es esperado ya que se le exigió al robot avanzar  $10\text{ mm}$  más de lo esperado en este eje. También se observa que el controlador de impedancia responde conforme a lo esperado, compensando la interacción con el ambiente.



**Figura 4.20:** Fuerzas percibidas por el sensor durante la interacción.

Se propone repetir el experimento mostrado anteriormente, cambiando el espacio de trabajo del dispositivo esclavo con el fin de proponer una superficie que se aproxime al plano formado por la libreta. Esto para observar el hecho de que el controlador

de impedancia responde de mejor manera con la superficie conocida, que simplemente hacer una aproximación del objeto. En la Figura (4.21) se aprecia en azul la trayectoria descrita por el dispositivo maestro y en rojo la estimación correspondiente para el esclavo, las cuales son muy parecidas a las mostradas en la Figura (4.18).

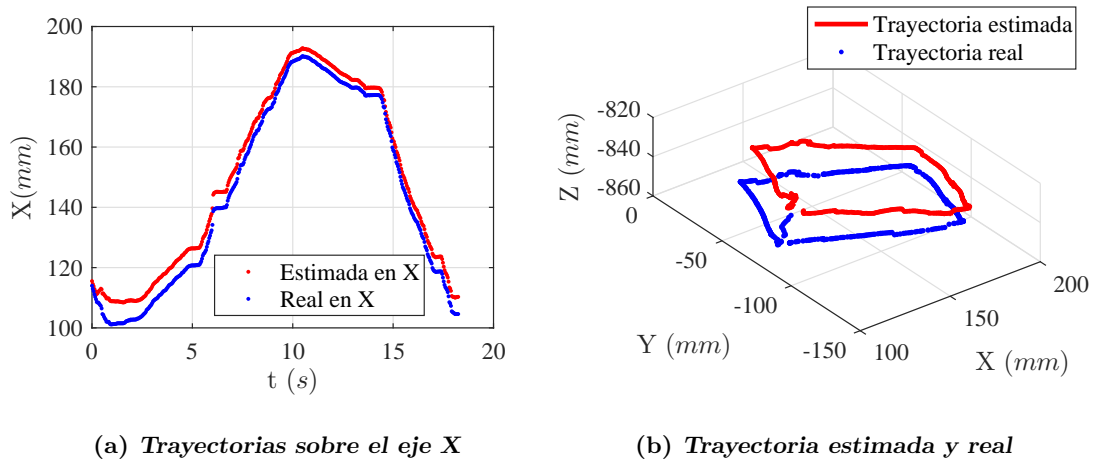


(a) *Trayectoria maestro*

(b) *Trayectoria esclavo estimada*

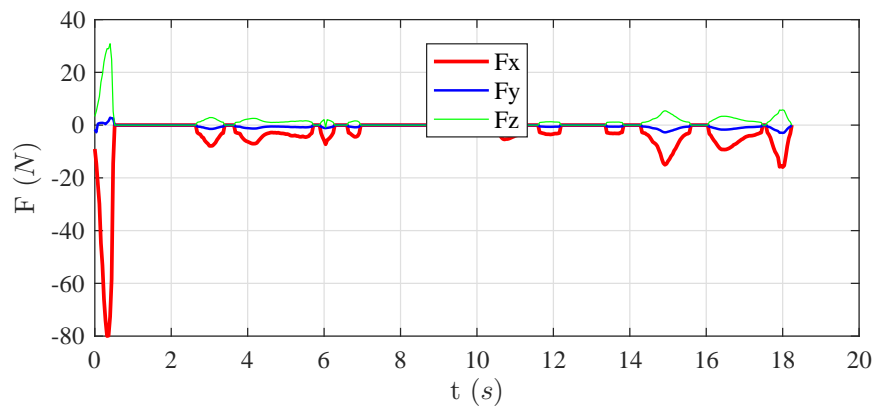
**Figura 4.21:** Trayectorias descritas por el maestro y el esclavo sobre un plano propuesto.

La comparación entre el trazo deseado y el real seguido por el robot se observan en la Figura (4.22). En la izquierda se muestra la comparación entre las trayectorias en el eje  $X$ , en la derecha, las trayectorias en el espacio coordenado del robot. Se aprecia una diferencia mayor entre la estimada y la real, ya que el plano propuesto difiere en la inclinación y posición real del objeto.



**Figura 4.22:** Comparativa entre las trayectoria estimada y la real trazada sobre la superficie plana propuesta.

Las fuerzas de interacción también son diferentes, se alcanza un pico aproximado de  $80\text{ N}$  en el eje  $X$  del robot, pero también uno de casi  $30\text{ N}$  en  $Z$ , lo que indica que el controlador tiene que compensar una mayor contacto con la superficie, debido a la incertidumbre de la geometría. Este experimento comprueba que la reconstrucción de la superficie ayuda a mejorar el comportamiento del controlador de impedancia propuesto en trabajos anteriores.

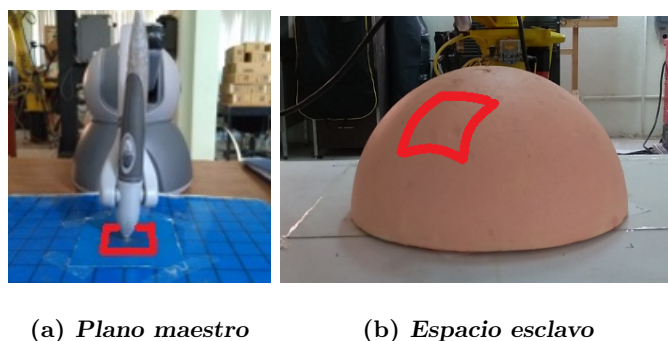


**Figura 4.23:** Fuerzas percibidas por el sensor durante la interacción.

### 4.3.2. Pruebas sobre una sección esférica

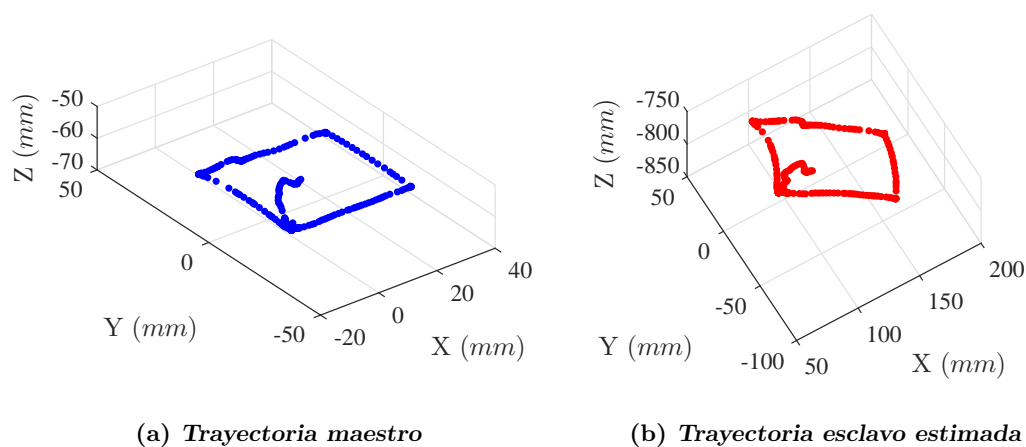
Dado que el propósito es que se mejore la interacción con objetos cuya superficie presente curvas suaves, se propone realizar un seguimiento de trayectoria sobre una

semiesfera de unicel. El radio de la esfera es de  $125\text{ mm}$  y se colocó sobre el espacio de trabajo definido en la sección 4.2.1. En la Figura (4.24) se muestran en rojo la trayectoria que sigue el dispositivo maestro y la esperada que siga el esclavo, esto tomando nueve puntos que definen el espacio de trabajo del esclavo.



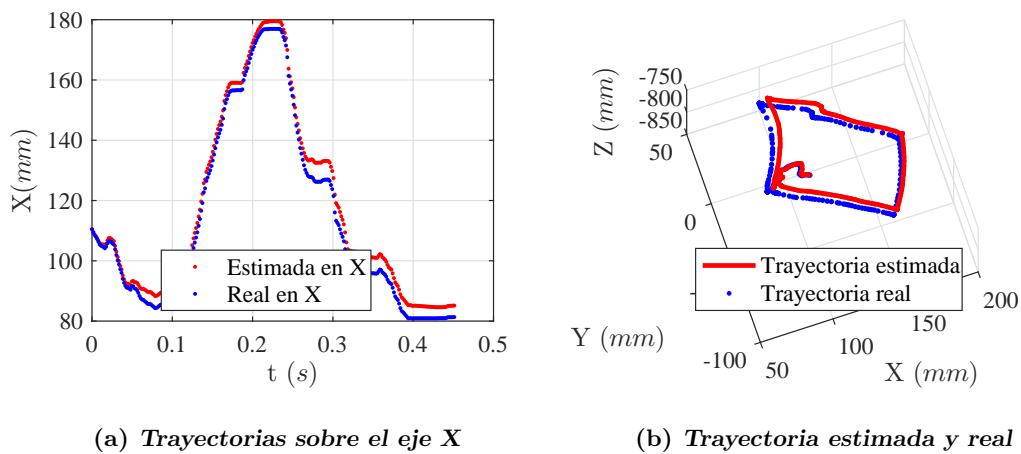
**Figura 4.24:** Trayectorias propuestas para el seguimiento.

En la unidad háptica se traza un cuadrado de  $20\text{ mm}$  por lado para ser mapeado a la esfera. En la Figura (4.25) se muestra en azul el trazo realizado por la unidad háptica, a partir del cual se estima el trazo rojo, que es la trayectoria deseada a seguir por el robot. Con el fin de asegurar el contacto en todo momento, a la coordenada  $X$  estimada se le agregan  $10\text{ mm}$ , con lo que el efector final avanza una profundidad mayor en este eje.



**Figura 4.25:** Trayectorias descritas por el dispositivo maestro y esclavo sobre una sección esférica.

Se compara la trayectoria deseada y la descrita por el robot como se muestra en la Figura (4.19). En la gráfica de la derecha se comparan los puntos estimados y seguidos por el robot en el eje  $X$ , en la derecha se observa la trayectoria real seguida por el robot, la cual se asemeja a lo esperado. Al inicio de la trayectoria, se comienza arriba del plano maestro definido, después el controlador de impedancia va compensando la interacción, que en este caso se propuso  $10\text{ mm}$  mayor en el eje  $X$ . A pesar de esto se observa un buen comportamiento en el seguimiento de la trayectoria.

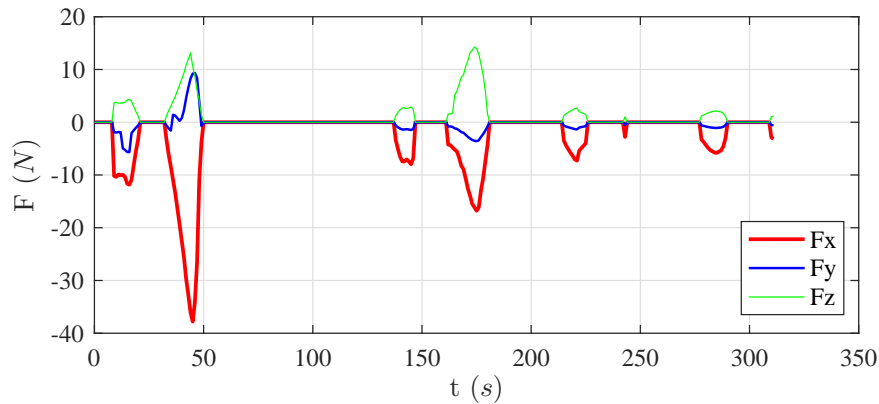


**Figura 4.26:** Comparativa entre las trayectoria estimada y la real trazada sobre la superficie semiesférica estimada.

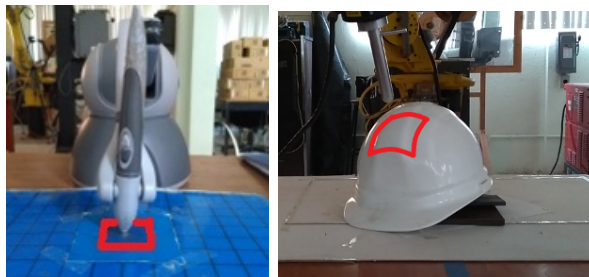
Las fuerzas de interacción que percibe el efector final del robot se aprecian en la Figura (4.27), donde se observa la respuesta del controlador de impedancia ante esta interacción. Se tiene un pico de fuerza cercano a los  $40\text{ N}$  en el eje  $X$ , lo cual es esperado ya que se le exigió al robot avanzar más de lo esperado en este eje. También se observa que el controlador de impedancia responde al contacto con la superficie.

### 4.3.3. Pruebas sobre casco

En este experimento se busca estimar un objeto que contenga una superficie con curvas suaves y con una dureza mayor a la del unicel. Se propone utilizar un casco de plástico el cual, si bien su geometría se asemeja a la de una semiesfera, tiene una superficie que cumple este propósito y el material con el que está fabricado lo hace poco deformable al estar sometido a las fuerzas de interacción que se han observado hasta este punto. Al igual que en las pruebas anteriores, se propone realizar un cuadrado de  $20\text{ mm}$  por lado en el plano de trabajo maestro y estimar su correspondiente trayectoria en el esclavo, como se muestra en la Figura (4.28).



**Figura 4.27:** Fuerzas percibidas por el sensor durante la interacción.



(a) *Plano maestro*

(b) *Espacio esclavo*

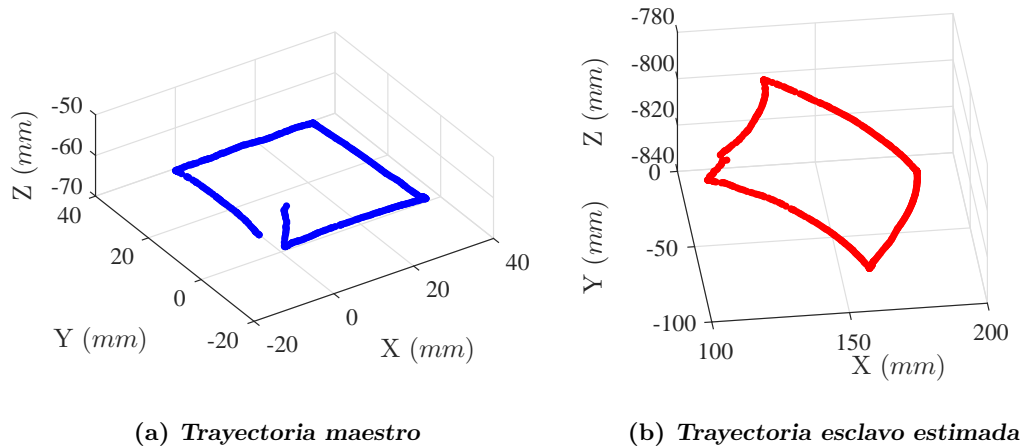
**Figura 4.28:** Trayectorias propuestas para el seguimiento.

Se busca que la trayectoria descrita por la unidad háptica, la cual se aprecia en azul en la Figura (4.29), sea lo más parecida al cuadrado propuesto. Se inicia arriba del plano maestro la trayectoria y se baja hasta alcanzar dicho plano. En rojo se observa la trayectoria estimada para el dispositivo esclavo que debe ser seguida idealmente.

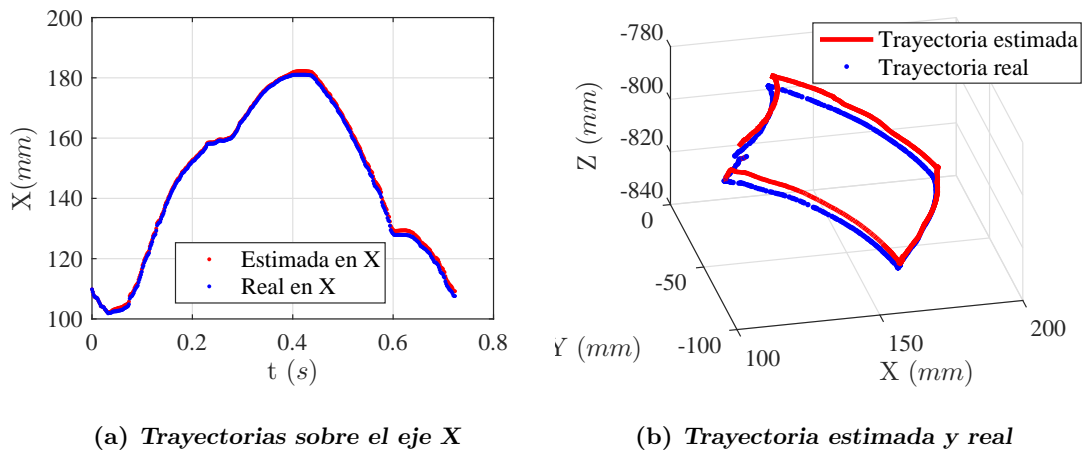
Como en los casos anteriores, se agregan 10 mm a las estimaciones en el eje  $X$  con el objetivo de asegurar contacto con la superficie en todo momento y observar la respuesta en posición del controlador de impedancia. En la Figura (4.30) se muestra la trayectoria estimada y real seguida por el robot en el eje  $X$ , así como la comparación entre la estimada y real en coordenadas tridimensionales. Se observa que se describe la forma esperada sobre el objeto, con la variación esperada.

En la Figura (4.31) se muestran las fuerzas de interacción entre el efector final del robot y la superficie del casco estimada. Se observa, al igual que en casos anteriores, un pico de fuerza, cuyo valor es de aproximadamente 23 N en el eje  $X$ . Este se presenta durante el inicio de la trayectoria debido a que el robot baja de una posición arriba de la superficie y no hay una compensación de posición hasta que la herramienta entra



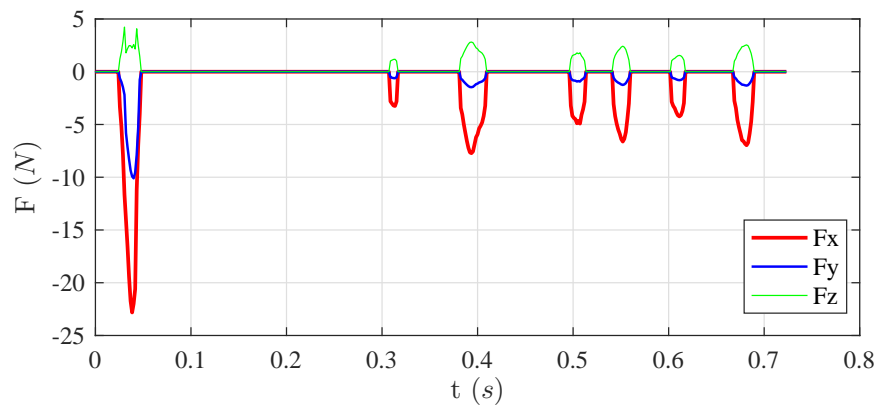


**Figura 4.29:** Trayectorias descritas por el dispositivo maestro y esclavo sobre una superficie estimada.



**Figura 4.30:** Comparativa entre las trayectoria estimada y la real trazada sobre la superficie del casco.

en contacto con la superficie. Esto provoca una mayor fuerza de contacto debido a los retardos de comunicación, la respuesta del sistema y el controlador. También se aprecia que el controlador de impedancia hace una buena compensación de posición.



**Figura 4.31:** Fuerzas percibidas por el sensor durante la interacción.

# Conclusiones

---

En el presente trabajo de tesis se logró la implementación de un sistema de teleoperación que interactúa sobre superficies arbitrarias, reconstruidas por un sistema no invasivo de visión por computadora libre de calibración. En este sistema se busca seguir trayectorias descritas por un operador mediante una unidad háptica como robot maestro, sobre superficies de objetos cuya geometría tenga curvas suaves, y que un robot esclavo interactúe sobre ellas. Se validó el hecho de que el conocimiento aproximado de la superficie sobre la que se interactúa mejora el desempeño del controlador de impedancia propuesto en trabajos anteriores [8],[14] y que compensa los errores debidos a la estimación de las superficies arbitrarias y el mapeo de las trayectorias descritas. Las principales aportaciones del trabajo de tesis desarrollado son:

El desarrollo e implementación de un algoritmo de reconstrucción de superficies arbitrarias dentro del espacio de trabajo del robot esclavo con una buena precisión en la estimación de puntos proyectados sobre la misma, logrando estimar la geometría de superficies suaves. Este algoritmo constituye el desarrollo central de la tesis ya que se logra realizar el mapeo de trayectorias descritas por un robot esclavo, sobre la superficie de objetos de trabajo, lo cual mejora el desempeño del controlador de impedancia desarrollado en trabajos previos.

Para la estimación tridimensional de múltiples puntos proyectados sobre una superficie de trabajo, se implementó un método para hacer corresponder los puntos detectados por dos cámaras ubicadas en distintas posiciones, con lo que se logra realizar una estimación más rápida, optimizando el proceso de reconstrucción tridimensional.

Se desarrolló un mapeo para resolver el problema de asimetría cinemática, debido a las diferentes estructuras de los robots empleados en el sistema de teleoperación, mediante la definición de espacios de trabajo para cada uno de los dispositivos. Haciendo más fácil la identificación de los límites en los que el operador y el robot esclavo se mueven.

Así mismo, se logró la actualización del sistema de teleoperación con los elementos descritos anteriormente e integrando el controlador de impedancia desarrollado en trabajos previos. Esto en una interfaz gráfica de usuario que hace más intuitivo el funcionamiento del sistema, además del desarrollo basado en programación orientada a objetos facilita la implementación de trabajos futuros dentro de este proyecto.

Finalmente, a partir de los resultados obtenidos, no solo se validó el hecho de que el conocimiento de la superficie arbitraria mejora el desempeño del controlador de impedancia, también se observa que el retardo en la comunicación y el muestreo de posiciones del robot maestro (el cual no sobrepasaba los 40 *ms*) no afectó en la estabilidad del sistema.

Como parte del trabajo realizado durante la tesis, se presentó un artículo titulado “Sistema de teleoperación sobre superficies arbitrarias con visión no calibrada” en el XIX Congreso Mexicano de Robótica 2017 (XIX COMRob 2017) el cual se llevó a cabo en las instalaciones de la Torre Académica Mazatlán de la Universidad Autónoma de Sinaloa, Unidad Regional Sur, del 8 al 10 de noviembre de 2017 [26]

## Trabajo a futuro

Con base en el sistema de teleoperación implementado y los resultados obtenidos durante el desarrollo del trabajo de tesis presentado, se presentan las siguientes áreas para trabajo futuro:

- Agregar más cámaras al sistema de visión no calibrada, con el objetivo de ampliar la visibilidad de los objetos y así mejorar la estimación de su superficie. Esto con el fin de ampliar el espacio de trabajo del dispositivo esclavo.
- Mejorar el reconocimiento de puntos láser proyectados sobre la superficie del objeto, considerando los cambios ambientales de iluminación, ya que en aplicaciones prácticas los operadores necesitan tener una buena visibilidad del ambiente.
- Considerar los retardos en la comunicación entre los dispositivos que conforman el sistema de teleoperación, analizar los efectos que causan en la dinámica del sistema e implementar un controlador retardado para mejorar su respuesta.
- Automatizar el proceso de estimación de la superficie para facilitar su manejo, así como mejorar la interfaz humano-máquina del sistema de teleoperación para hacerlo más intuitivo al usuario.
- Implementar un protocolo de seguridad para realizar pruebas sobre personas, respetando los aspectos éticos involucrados en los procesos de experimentación con seres humanos.

---

## Anexo A

# Anexos

---

La programación de los algoritmos descritos en este trabajo, se realizó con base en la programación orientada a objetos (POO) y el lenguaje de programación C++. La metodología para el desarrollo de los programas y características de este lenguaje se describen en [23]. Las principales clases desarrolladas durante la implementación proyecto son: cUEye, cFanuc, cCSM, y cAsimetria. Los elementos y funciones miembro se determinaron a partir de las necesidades planteadas para resolver los algoritmos descritos.

### A.1. Comunicación con las cámaras UEye

La clase cUEye permite la comunicación con cada una de las cámaras, implementa la detección de puntos objetivo y realiza la correspondencia entre espacios de cámara. Los principales elementos miembro de la clase son: ancho, alto y ID de la cámara, coordenadas  $(x, y)$  del centroide de un objeto detectado, una matriz de centroides de objetos detectados y un valor de verificación de conexión. En la Tabla (A.1) se muestran las principales funciones miembro de la clase y en el código (A.1) se implementa la detección del centroide de un LED captado por la cámara.

```
bool cUEye::detectarLed()
{
    Mat original, destino, imLed;
    vector<Mat> canales(3);
    original = getCaptura();
    //Filtro ogaussiano para reducir ruido
    GaussianBlur(original, original, Size(5,5), 1, 1 );
    imLed = original;
    //Se separa el canal rojo de la imagen
```

Función miembro	Descripción
<i>Open(intID);</i>	Abre el puerto de comunicación con la cámara uEye conectada vía USB a la computadora.
<i>getCaptura();</i>	Obtiene una captura de imagen y la almacena para el procesamiento.
<i>detectarLed();</i>	Función que detecta el centroide de un LED en la escena.
<i>getMatrizPuntosRGB();</i>	Detecta el centroide de puntos láser proyectados sobre una superficie, almacena los valores encontrados en un arreglo.
<i>correspondenciaPuntos();</i>	Realiza la correspondencia entre puntos detectados en dos espacios de cámara diferentes, almacena en un archivo las coincidencias de coordenadas encontradas.

**Tabla A.1:** Funciones miembro de la clase cUeye.

```

split(original, canales);
Mat red = canales[2];
cvtColor(red, destino, COLOR_GRAY2BGR );
//Umbralizado de la imagen y operaciones morfologicas
threshold(destino, destino, 70, 255, THRESH_BINARY);
Mat element = getStructuringElement( MORPH_ELLIPSE, Size( 5,
    5 ), Point( 2, 2) );
erode(destino, destino, element);
dilate(destino, destino, Mat(), Point(-1, -1), 1, 1, 1);
//Parametros del detector
SimpleBlobDetector detectorLed(parametros);
vector<KeyPoint> puntos;
detectorLed.detect(destino, puntos);
x_led = puntos[0].pt.x - 1280/2;
y_led = 1024/2 - puntos[0].pt.y;

```

```

        drawKeypoints( destino , puntos , destino , Scalar(17,223,31) ,
            DrawMatchesFlags::DRAW_RICH_KEYPOINTS );
        drawKeypoints( original , puntos , original , Scalar(17,223,31) ,
            DrawMatchesFlags::DRAW_RICH_KEYPOINTS );
        return true;
    }

```

**Código A.1:** Implementación de la detección del centroide de un LED.

## A.2. Comunicación con el robot Fanuc

La clase `cFanuc` permite la comunicación el robot Fanuc LR Mate 200 iC, implementa el posicionamiento en  $XYZWPR$ , la obtención de posición del efector final y captura las fuerzas percibidas por el sensor. Los principales elementos miembro de la clase son: ángulos de junta, vector de posición, furzas y momentos, tipo de movimiento y velocidad. En la Tabla (A.2) se muestran las principales funciones miembro de la clase y en el código (A.1) se implementa la función miembro para el posicionamiento y obtención de fuerzas.

Función miembro	Descripción
<code>ConectarRobot();</code>	Abre el puerto de comunicación con el robot Fanuc.
<code>DesconectarRobot();</code>	Cierra el puerto de comunicación con el robot Fanuc.
<code>GetXYZWPR();</code>	Obtiene la posición tridimensional del efector final y la orientación $WPR$ .
<code>SetXYZWPR();</code>	Envía la posición tridimensional y la orientación al robot Fanuc, y recibe las fuerzas de contacto del efector final.

**Tabla A.2:** Funciones miembro de la clase `cFanuc`.

```

int CFanucClient::SetXYZWPR(double Posicion[6], int move, int speed)
{
    //ColumnVector Th(3);
    char FUERZ[100] = "";

```

```
    char sX[10], sY[10], sZ[10], sW[10], sP[10], sR[10], MOV[10], VEL
        [10];
    char xyzwpr[100] = "";
    int bytesSent, respuesta;
    int bytesRecv = SOCKET_ERROR;
    char recvbuf[10] = "";
    sprintf(sX, "%4.3f", Posicion[0]);
    sprintf(sY, "%4.3f", Posicion[1]);
    sprintf(sZ, "%4.3f", Posicion[2]);
    sprintf(sW, "%4.3f", Posicion[3]);
    sprintf(sP, "%4.3f", Posicion[4]);
    sprintf(sR, "%4.3f", Posicion[5]);
    sprintf(MOV, "%1i", move);
    sprintf(VEL, "%4i", speed);
    sprintf(xyzwpr, "%9s;%9s;%9s;%9s;%9s;%9s;%1s;%4s!", sX, sY, sZ,
        sW, sP, sR, MOV, VEL);
    char sendmsg[10] = "95";
    bytesSent = send(m_socket, sendmsg, strlen(sendmsg), 0);
    bytesRecv = recv(m_socket, recvbuf, 2, 0);
    respuesta = atoi(recvbuf);
    if(respuesta == 59)
        bytesSent = send(m_socket, xyzwpr, strlen(xyzwpr), 0)
            ;
    bytesRecv = recv(m_socket, recvbuf, 2, 0);
    respuesta = atoi(recvbuf);
    if(respuesta == 86)
        bytesRecv = recv(m_socket, FUERZ, 29, 0);
    bytesRecv = recv(m_socket, recvbuf, 2, 0);
    respuesta = atoi(recvbuf);
    double guarda[3];
    char * pEnd[2];
    guarda[0] = strtod(FUERZ, &pEnd[0]);
    guarda[1] = strtod(pEnd[0], &pEnd[1]);
    guarda[2] = strtod(pEnd[1], NULL);
    return 1;
}
```



---

**Código A.2:** Implementación del posicionamiento del robot Fanuc.

### A.3. Estimación de los parámetros de visión

La clase `cCSM` implementa la estimación de los parámetros de visión a partir del método de mínimos cuadrados, como se explica en el capítulo 2. Los principales elementos miembro de la clase son: Los ángulos de junta del robot, los parámetros  $C$  de cámara, el vector de coordenadas estimadas, observaciones, centroides del LED y el número de muestras usadas para la estimación. En la Tabla (A.3) se muestran las principales funciones miembro de la clase y en el código (A.3) se implementa la función miembro la estimación paramétrica.

Función miembro	Descripción
<code>cargaObservaciones();</code>	Carga las observaciones a partir del número de muestras propuestas para determinar la estimación.
<code>estimaParametrosC();</code>	Realiza el cálculo del vector de parámetros $C$ a partir del método de mínimos cuadrados.
<code>agregarObservacion;</code>	Agrega observaciones al método para mejorar las estimaciones de posición, esto al tener mas muestras para el cálculo de los parámetros de visión.

**Tabla A.3:** Funciones miembro de la clase `cCSM`.

```
void CSM::estimaParametrosC()
{
    mat A; //Jacobiano
    mat pA;
    colvec DY; //Delta Y
    colvec DC; //Incremento de C
    mat J_ant(1,1,fill::zeros), J_act(1,1,fill::zeros); //
        Funcion de costos anterior y actual
    double e = 1;
```

```

int i = 1 ;
while (e > E)
{
    // Calcular Y de modelo
    cYmod();
    // Calcular el Jacobiano
    calcMatDerivadaC(); //Derivada con respecto a C
    A = A_xyz * mDC;
    // Se calcula la diferencia entre el modelo y las
        observaciones
    DY = Y - Ymod;
    J_act = DY.t() * W * DY;
    //Se calcula el delta C
    pA = A.t() * W * A;
    DC = inv(pA)*A.t()*W*DY;
    //Actualizar los parametros C
    C = C + DC;
    // se calcula el error de la funcion de costos para
        hacer la minimizacion
    if(i != 1)
    {
        e = abs(J_ant(0,0) - J_act(0,0));
    }
    J_ant = J_act;
    i +=1;
    //Si se supera el numero maximo de iteraciones salir
    if (i > ITERACIONES_MAX) {break;}
}
}

```

**Código A.3:** Implementación del la estimación de parámetros de visión.

Una vez obtenidos los valores de los parámetros es posible estimar posiciones tri-dimensionales con respecto al marco del robot, por lo que se desarrolló una función miembro para este propósito, la cual se muestra en el código (A.3).

```

void EstimadorXYZ::estimaPosXYZ(colvec Yxyz, colvec c1, colvec c2)

```

```

{
    colvec DY(4);
    colvec Dxyz(4);
    mat J_ant(1,1,fill::zeros), J_act(1,1,fill::zeros);
    XYZ(0) = 750.0; //Coordenada de inicio para X
    XYZ(1) = 10.0; //Coordenada de inicio para Y
    XYZ(2) = 10.0; //Coordenada de inicio para Z
    C1 = c1;
    C2 = c2;
    Y = Yxyz;
    int i = 1;
    double e = 1;
    while (e > E)
    {
        // Se calcula el modelo con C1, C2 y XYZ
        fxy_modCam();
        // Diferencia entre el modelo y las observaciones
        DY = Y - Ymod;
        // Calculo del jacobiano 'A'
        fJacobiano();
        J_act = DY.t() * W * DY;
        Dxyz = inv( A.t() * W * A ) * A.t() * W * DY;
        XYZ = XYZ + Dxyz;
        if(i != 1)
        {
            e = abs(J_ant(0,0) - J_act(0,0));
        }
        J_ant = J_act;
        i +=1;
        //Si se supera el numero maximo de iteraciones salir
        if (i > ITERACIONES_MAX) {break;}
    }
    int l = 9;
}

```

Código A.4: Implementación del la estimación de posición.

## A.4. Estimación de la trayectoria del esclavo

La clase *Asimetria* implementa los algoritmos para la estimación de parámetros de asimetría que permiten mapear una trayectoria descrita en el maestro, al dispositivo esclavo. Los principales elementos miembro de la clase son: número de muestras, vectores de posición para los puntos que definen los espacios de trabajo, matrices para realizar las rotaciones necesarias y vectores para la estimación de orientación. En el código (A.4) se implementa la función miembro para la estimación de los parámetros de asimetría y para la estimación de un punto en el esclavo.

```

//Estimacion de los parametros de asimetria
void Asimetria::estimaParametrosAsimetria()
{
    cargarPlanosTrabajo();
    calculaMatA();
    calculaVectoresB();
    calculaPlanoPromedio();
    calculaOrientacion();
    Px = inv(A)*Bx;
    Py = inv(A)*By;
    Pz = inv(A)*Bz;
}

//Estima la posicion del esclavo a partir de una posicion del maestro
rowvec Asimetria::estimaPosXYZ_esclavo(rowvec XYZ)
{
    double Z = XYZ(2);
    double eZ = 1.3;
    XYZ_esclavo(0) = Px(0)*XYZ(0)*XYZ(0) + Px(1)*XYZ(0)*XYZ(1) +
        Px(2)*XYZ(1)*XYZ(1) + Px(3)*XYZ(0) + Px(4)*XYZ(1) + Px(5);
    XYZ_esclavo(1) = Py(0)*XYZ(0)*XYZ(0) + Py(1)*XYZ(0)*XYZ(1) +
        Py(2)*XYZ(1)*XYZ(1) + Py(3)*XYZ(0) + Py(4)*XYZ(1) + Py(5);
    XYZ_esclavo(2) = Pz(0)*XYZ(0)*XYZ(0) + Pz(1)*XYZ(0)*XYZ(1) +
        Pz(2)*XYZ(1)*XYZ(1) + Pz(3)*XYZ(0) + Pz(4)*XYZ(1) + Pz(5);
    if ( abs( Z - zm(0) ) > eZ )
    {
        double Ze = abs(Z-zm(0));
        XYZ_esclavo(0) = XYZ_esclavo(0) - Xp(0)*Ze;
    }
}

```

```

        XYZ_esclavo(1) = XYZ_esclavo(1) - Xp(1)*Ze;
        XYZ_esclavo(2) = XYZ_esclavo(2) + Ze;
    }
    mat Ry(3,3);
    Ry(0,0) = 0; Ry(0,1) = 0; Ry(0,2) = 1;
    Ry(1,0) = 0; Ry(1,1) = 1; Ry(1,2) = 0;
    Ry(2,0) = -1; Ry(2,1) = 0; Ry(2,2) = 0;
    XYZ_esclavo = XYZ_esclavo*Ry;
    return XYZ_esclavo;
}

```

**Código A.5:** Implementación del método de asimetría.

Una vez determinados los parámetros, se implementa un hilo de comunicación que recibe los valores de la trayectoria descrita por el dispositivo maestro y realiza la estimación correspondiente para el plano esclavo. La implementación de este hilo de comunicación se realiza en el código [A.4](#).

```

//Evento del timer del sistema de teleoperacion
System::Void sistemaTeleoperacion::sistemaForm::tTeleoperacion_Tick(
    System::Object^ sender, System::EventArgs^ e)
{
    tiHaptica->Stop(); //Prioridad para el timer del sistema de
        teleoperacion
    if ( gServoDeviceData.Botones == 1)
    {
        double x = gServoDeviceData.m_Position[0];
        double y = gServoDeviceData.m_Position[1];
        double z = gServoDeviceData.m_Position[2];
        String^ Linea = x.ToString("0.000") + " " + y.
            ToString("0.000") + " " + z.ToString("0.000");
        swArchivoTrayectoriaHaptica->WriteLine(Linea);
        rowvec pHap(3);
        pHap(0) = x; pHap(1) = y; pHap(2) = z;
        //Se pasa la coordenada del punto de la haptica
            actual
        rowvec pEstimado = planoHaptica.estimaPosXYZ_escalvo(

```

```
        pHap);  
        //Se escribe en archivo la trayectoria estimada para  
        el dispositivo esclavo  
        Linea = pEstimado(0).ToString("0.000") + " " +  
                pEstimado(1).ToString("0.000") + " " + pEstimado  
                (2).ToString("0.000");  
        swTEstimadaEsclavo->WriteLine(Linea);  
        double Pos[6] = {0};  
        Pos[0] = pEstimado(0);  
        Pos[1] = pEstimado(1);  
        Pos[2] = pEstimado(2);  
        Pos[3] = -170.0;  
        Pos[4] = -70.00;  
        Pos[5] = -0.0;  
        int res = Fanuc.SetXYZWPR(Pos, movLineal, vellLineal);  
    }  
    tiHaptica->Start(); //Se vuelve a iniciar el timer para  
        actualizar datos de la haptica  
}
```

**Código A.6:** Hilo que implementa la teleoperación del robot.

# Bibliografía

---

- [1] Benali, A., Wasiak, V., y Fontaine, J. G. (2001). Remote robot teleoperation via internet. a first approach. En *Proceedings 10th IEEE International Workshop on Robot and Human Interactive Communication. ROMAN 2001 (Cat. No.01TH8591)*, pp. 306–312. [8](#)
- [2] Besl, P. J. (1988). Active, optical range imaging sensors. *Mach. Vision Appl.*, 1(2):127–152. [18](#)
- [3] Bogado Torres, J. R. (2007). Control bilateral de robots teleoperados por convergencia de estados. Tesis de maestría, Universidad Politécnica de Madrid. [8](#)
- [4] Bonilla, I., González-Galván, E. J., Chavez-Olivares, C., Mendoza, M., Loredoflores, A., Reyes, F., y Zhang, B. (2010). A vision-based, impedance control strategy for industrial robot manipulators. En *Automation Science and Engineering (CASE), 2010 IEEE Conference on*, pp. 216–221. IEEE. [24](#)
- [5] Craig John, J. (2006). *Robótica*. Prentice Hall. [19](#)
- [6] Dario, P. y Menciassi, A. (2002). Robotics for surgery. En *Proceedings of the Second Joint 24th Annual Conference and the Annual Fall Meeting of the Biomedical Engineering Society [Engineering in Medicine and Biology]*, volumen 3, pp. 1813–1814 vol.3. [10](#)
- [7] Elhajj, I., Hummert, H., Xi, N., Fung, W. K., y Liu, Y.-H. (2000). Real-time bilateral control of internet-based teleoperation. En *Proceedings of the 3rd World Congress on Intelligent Control and Automation (Cat. No.00EX393)*, volumen 5, pp. 3761–3766 vol.5. [8](#)
- [8] Esparza Gurrola, A. (2013). Implementación de un sistema de teleoperación con percepción de fuerza. Tesis de maestría, Universidad Autónoma de San Luis Potosí, Facultad de Ingeniería. [10](#), [58](#), [69](#)
- [9] González-Galván, E. J., Chávez, C. A., Bonilla, I., Mendoza, M., Raygoza, L. A., Loredoflores, A., y Zhang, B. (2011). Precise industrial robot positioning and path-

- tracking over large surfaces using non-calibrated vision. En *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 5160–5166. IEEE. 24
- [10] Gonzalez-Galvan, E. J. y Skaar, S. B. (1996). Efficient camera-space manipulation using moments. En *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, volumen 4, pp. 3407–3412. IEEE. 17, 24
- [11] González Galván, E. (200). *Introducción al Curso de Robótica y Visión por Computadora*. Universidad Autónoma de San Luis Potosí. Notas del curso impartido en el CIEP-FI. 22, 24
- [12] Haddadi, A. y Hashtrudi-Zaad, K. (2010). Bounded-impedance absolute stability of bilateral teleoperation control systems. *IEEE Transactions on Haptics*, 3(1):15–27. 7
- [13] Hashtrudi-Zaad, K. y Salcudean, S. E. (2002). Transparency in time-delayed systems and the effect of local force feedback for transparent teleoperation. *IEEE Transactions on Robotics and Automation*, 18(1):108–114. 8
- [14] Hernández Marín, J. R. (2014). Desarrollo de un sistema de teleoperación que permita la paridad entre superficies arbitrarias. Tesis de maestría, Universidad Autónoma de San Luis Potosí, Facultad de Ingeniería. 11, 34, 58, 59, 69
- [15] Hogan, N. (1984). Impedance control: An approach to manipulation. En *1984 American Control Conference*, pp. 304–313. 15
- [16] Kronander, K. y Billard, A. (2016). Stability considerations for variable impedance control. *IEEE Transactions on Robotics*, 32(5):1298–1305. 33
- [17] Lawrence, D. A. (1992). Designing teleoperator architectures for transparency. En *Proceedings 1992 IEEE International Conference on Robotics and Automation*, pp. 1406–1411 vol.2. 14, 15, 33
- [18] Lawrence, D. A. (1993). Stability and transparency in bilateral teleoperation. *IEEE Transactions on Robotics and Automation*, 9(5):624–637. 8
- [19] Loredó-Flores, A., González-Galván, E. J., Cervantes-Sánchez, J. J., y Martínez-Soto, A. (2008). Optimization of industrial, vision-based, intuitively generated robot point-allocating tasks using genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(4):600–608. 24
- [20] Molleda Merá, J. (2008). Técnicas de visión por computador para la reconstrucción en tiempo real de la forma 3d de productos laminados. Tesis de maestría, Universidad de Oviedo, Departamento de informática. 17
- [21] Na, U. J. y Vu, M. H. (2012). Adaptive impedance control of a haptic teleoperation system for improved transparency. En *2012 IEEE International Workshop on Haptic Audio Visual Environments and Games (HAVE 2012) Proceedings*, pp. 38–43. 7, 15



- 
- [22] Nevatia, R. (1982). *Machine perception*. Prentice Hall. 17
- [23] P. J. DEITEL, H. M. D. (2009). *C++ Como programar*. PEARSON, Prentice Hall. 71
- [24] Penin, L. F., Ferre, M., Fernandez, J., Aracil, R., y Barrientos, A. (1997). Design fundamentals of master-slave system with a force-position bilateral control scheme. En *Proceedings of the IFAC*. 14
- [25] Pensati, P. y Cherel, G. (1959). French master devices. En *Proceedings of the 7th HLE*. 6
- [26] Ramírez-Quirino, A., González-Galván, E. J., Loredó, A., Martínez, I., González de Alba, A., Cárdenas, J. A., y Maya, M. E. (2017). Sistema de teleoperación sobre superficies arbitrarias con visión no calibrada. En *XIX Congreso Mexicano de Robótica 2017 (XIX COMRob)*. 70
- [27] Rocchini, C., Cignomi, P., Montani, C., Pingi, P., y Scopigno, R. (2001). A low cost 3d scanner based on structured light. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 20(3):299–308. 18
- [28] Rodríguez-Angeles, A., Guzman-Gutierrez, J. L., y Cruz-Villar, C. (2010). User wearable interface based on inertial sensors for unilateral master-slave robot teleoperation. En *2010 7th International Conference on Electrical Engineering Computing Science and Automatic Control*, pp. 458–463. 13
- [29] Soubervielle Montalvo, R. (2013). Sistema para el guiado activo de robots industriales. Tesis de maestría, Universidad Autónoma de San Luis Potosí. 58