



UNIVERSIDAD
DE MÁLAGA



ESCUELA DE INGENIERÍAS INDUSTRIALES

Departamento de Ingeniería de Sistemas y Automática

Área de Conocimiento de Ingeniería de Sistemas y Automática

TRABAJO FIN DE GRADO

SISTEMA ROBUSTO DE SEGUIMIENTO DE HERRAMIENTAS QUIRÚRGICAS

Grado en

Ingeniería Electrónica, Robótica y Mecatrónica

Autor: PAULA LUQUE CONTRERAS

Tutor: ISABEL GARCÍA MORALES

Cotutor: VÍCTOR FERNANDO MUÑOZ MARTÍNEZ

MÁLAGA, Septiembre de 2.023

SISTEMA ROBUSTO DE SEGUIMIENTO DE HERRAMIENTAS QUIRÚRGICAS

Autor: Paula Luque Contreras

Tutor: Isabel García Morales

Cotutor: Víctor Fernando Muñoz Martínez

Departamento: Ingeniería de Sistemas y Automática

Titulación: Grado en Ingeniería Electrónica, Robótica y Mecatrónica

Palabras clave: Cirugía laparoscópica, filtro de Kalman, cambio de sistema de referencia, sensor Polaris Spectra, instrumental quirúrgico.

Resumen

La curva de aprendizaje para nuevos cirujanos en operaciones de laparoscopia junto con la falta de cirujanos expertos para enseñar y corregir a los mismos, hace necesario que se creen simuladores de realidad virtual capaces de sustituir a estos cirujanos expertos. Además, en los últimos años, el uso de la cirugía laparoscópica robotizada ha estado en constante crecimiento, lo que aumenta la colaboración humano-robot en el quirófano. Partiendo de estas necesidades, surge este trabajo de fin de grado (TFG), que se enfoca en desarrollar un sistema robusto y preciso que estime la posición de los instrumentos quirúrgicos durante una operación de laparoscopia mediante el uso de sensores de localización. Se comienza con la toma de medidas de las posiciones del instrumental quirúrgico y tras esto se diseña e implementa el filtro de Kalman, capaz de predecir el movimiento del instrumental mediante un modelo matemático del movimiento que sigue el mismo durante la operación y fusiona las medidas para corregir dicha predicción. Se comprueba su efectividad mediante el uso de sensores Polaris Spectra y del robot UR3. Los resultados confirman una reducción significativa de errores en la estimación de la posición del instrumental.

ROBUST SURGICAL TOOL TRACKING SYSTEM

Author: Paula Luque Contreras

Supervisor: Isabel García Morales

Co-supervisor: Víctor Fernando Muñoz Martínez

Department: Ingeniería de Sistemas y Automática

Degree: Grado en Ingeniería Electrónica, Robótica y Mecatrónica

Keywords: Laparoscopic surgery, Kalman filter, change of reference frame, Polaris Spectra sensor, surgical instruments.

Abstract

The learning curve for new surgeons in laparoscopic surgeries, coupled with the lack of expert surgeons to train and correct them, necessitates the creation of virtual reality simulators capable of replacing these expert surgeons. Furthermore, in recent years, the use of robotic laparoscopic surgery has been steadily increasing, thereby enhancing human-robot collaboration in the operating room. Stemming from these needs, this undergraduate thesis work focuses on developing a robust and accurate system to estimate the position of surgical instruments during a laparoscopic operation using localization sensors. The process begins with measuring the positions of the surgical instruments, followed by the design and implementation of the Kalman filter. This filter has the capability to predict the movement of the instruments by using a mathematical model of their motion throughout the operation. It then integrates these measurements to correct the prediction. The effectiveness of the system is validated using Polaris Spectra sensors and the UR3 robot. The results confirm a significant reduction in errors in estimating the position of the instruments.

Agradecimientos

A mis tutores, Isabel y Víctor, por su ayuda y por ser mis guías durante este trabajo.

A mis compañeros de laboratorio y amigos, Juan María, Álvaro y Marcos, por hacer más ameno este proyecto, por su apoyo y por aconsejarme siempre que lo he necesitado.

Por último, a mis padres y amigos, por escucharme y acompañarme siempre, sin su respaldo no hubiese llegado hasta aquí.

Índice

Resumen	III
Abstract	V
Agradecimientos	IX
1 Introducción y visión general	1
1.1 Motivación	2
1.2 Marco de realización	2
1.3 Objetivo del trabajo	6
1.4 Principales aportaciones	7
1.5 Estructura del documento	8
2 Estado de la cuestión	11
2.1 Cirugía laparoscópica robotizada	12
2.2 Localización del instrumental laparoscópico dentro del campo operatorio	14
2.3 Casos de uso de los datos recopilados en la localización del instrumental laparoscópico	15
2.4 Conclusiones	18
3 Estimación de la posición del instrumental quirúrgico mediante el filtro de Kalman	21
3.1 Introducción	22
3.2 Planteamiento del problema	23

3.3	Diseño del filtro de Kalman	24
3.4	Fusión de medidas de la estimación de la posición mediante el filtro de Kalman	31
3.5	Conclusiones	34
4	Desarrollo del sistema de localización y resultados	35
4.1	Introducción	36
4.2	Descripción y diseño del entorno experimental	36
4.3	Calibración de los dispositivos del entorno de trabajo	38
4.4	Descripción del flujo del programa	44
4.5	Variables medidas y análisis estadístico	49
4.6	Conclusiones	56
5	Conclusiones y líneas futuras	59
5.1	Conclusiones	60
5.2	Líneas futuras	60
	Bibliografía	66

Índice de figuras

1.1	Sensores Polaris Spectra de NDI.	4
1.2	Herramientas que captan los sensores Polaris.	5
1.3	Universal Robot UR3.	6
1.4	Esquema del funcionamiento del entorno de trabajo.	8
2.1	Robot PUMA 560.	12
2.2	Ejemplos de robots usados en cirugías laparoscópicas.	13
2.3	Empleo de métodos de <i>tracking</i> en quirófano.	14
2.4	Empleo de métodos basados en imágenes en quirófano.	15
2.5	Ejemplo de la colaboración humano-robot.	16
2.6	Simuladores de RV para la formación en operaciones de laparoscopia.	18
3.1	Herramienta 39 del sensor Polaris Spectra con las partes de la misma referenciadas.	23
3.2	Entorno de trabajo.	24
3.3	Esquema de los pasos a seguir en el filtro de Kalman.	25
3.4	Esquema de los pasos a seguir en el filtro de Kalman con sus correspondientes ecuaciones.	31
3.5	Esquema del filtro de Kalman usando el método 1 para corregir.	32
3.6	Esquema del filtro de Kalman usando el método 2 para corregir.	33
4.1	Fotografía del entorno experimental.	37
4.2	Fotografías del robot en movimiento.	38

4.3	Esquema del montaje experimental con las matrices de paso entre los sistemas de coordenadas implicados.	39
4.4	Montaje experimental para la calibración de sensores.	40
4.5	Herramienta del sensor y el soporte para las mismas indicando sus partes correspondientes.	41
4.6	Zoom de la herramientas del Polaris (H39) unida al soporte que se acopla sobre la punta de la herramienta del robot (TTP).	42
4.7	Algoritmo de sincronización de tiempos.	44
4.8	Fases del sistema de seguimiento de instrumental quirúrgico. . . .	45
4.9	Uso de la función de inicialización para los dispositivos Polaris Spectra de NDI en un dispositivo con sistema operativo Ubuntu. . .	46
4.10	Uso de la función de lectura y escritura por el puerto serie para los dispositivos Polaris Spectra de NDI en un dispositivo con sistema operativo Ubuntu.	46
4.11	Uso de la función de lectura para los dispositivos Polaris Spectra de NDI en un dispositivo con sistema operativo Ubuntu.	47
4.12	Uso de la función de fin de comunicación para los dispositivos Polaris Spectra de NDI en un dispositivo con sistema operativo Ubuntu.	47
4.13	Uso de las funciones del filtro de Kalman con los dos métodos distintos de corrección.	48
4.14	Coordenada Z de los tres dispositivos con sus respectivas marcas de tiempo para el movimiento paralelo al eje X de la base del robot.	50
4.15	Zoom de la gráfica de la coordenada Z de los tres dispositivos con sus respectivas marcas de tiempo para el movimiento paralelo al eje X de la base del robot.	51
4.16	Zoom de la coordenada Z de los tres dispositivos y las correcciones de Kalman con sus respectivas marcas de tiempo.	53
4.17	Gráfica de la coordenada Z de los tres dispositivos y las correcciones de Kalman calculadas usando las funciones dadas por MATLAB con sus respectivas marcas de tiempo.	55

Índice de Tablas

4.1	Tabla que recoge los errores máximos en posición entre los distintos dispositivos para los distintos movimientos.	52
4.2	Tabla que recoge los errores máximos en posición entre los resultados obtenidos mediante el filtro de Kalman implementado con las funciones desarrolladas en este trabajo y el robot.	54
4.3	Tabla que recoge los errores máximos en posición entre los resultados obtenidos mediante el filtro de Kalman implementado con las funciones desarrolladas por MATLAB y el robot.	56

Capítulo 1

Introducción y visión general

Contenido

1.1 Motivación	2
1.2 Marco de realización	2
1.3 Objetivo del trabajo	6
1.4 Principales aportaciones	7
1.5 Estructura del documento	8

1.1. Motivación

La cirugía laparoscópica es un método de cirugía mínimamente invasiva común en el campo abdominal y pélvico. Consiste en la introducción de instrumental quirúrgico específico y de una cámara a través de unas pequeñas incisiones en el cuerpo del paciente. Este proceso, además de reducir riesgos, acorta el tiempo de postoperatorio y lo hace menos doloroso [1].

Sin embargo, y a pesar de estos beneficios, es más complicado para el cirujano en comparación con una operación abierta ya que no cuenta con visión 3D directa, requiere de habilidades técnicas especializadas, coordinación y los instrumentos quirúrgicos utilizados presentan limitaciones de movilidad.

Para solventar estos inconvenientes se estudian los movimientos y acciones que realiza un profesional durante la operación de laparoscopia mediante métodos de seguimiento (*tracking*) [1, 2] o métodos de *Deep Learning* [3] y se emplea ese conjunto de datos recopilados en la creación de simuladores virtuales capaces de enseñar y evaluar las habilidades y destrezas necesarias para llevar a cabo este tipo de abordaje [4], así como en el entrenamiento de robots para asistir en intervenciones quirúrgicas [5, 6].

En este TFG, la adquisición de dicha información se realiza mediante *tracking* usando dos sensores comerciales de infrarrojo capaces de localizar marcas situadas sobre el instrumental quirúrgico. Este método de obtención de datos conlleva la existencia de errores de medición, sombras sobre las marcas de referencia y pérdidas de la herramienta durante el proceso de medición, por lo tanto y con el objetivo de minimizar los errores presentados, existen soluciones como el filtro de Kalman [7, 8], un algoritmo matemático capaz de mejorar la precisión de los datos recopilados mediante la predicción y la fusión de medidas de distintos sensores [9].

1.2. Marco de realización

Este TFG ha sido desarrollado en el Laboratorio de Robótica Médica del Departamento de Ingeniería de Sistemas y Automática de la Universidad de Málaga, ubicado en el edificio de Institutos Universitarios en el Parque Tecnológico de Andalucía. A continuación se detallarán las líneas de investigación en las que está implicado, los proyectos desarrollados y la infraestructura que se ha usado en este trabajo.

Comenzando con las líneas de investigación, este grupo de investigación

cuenta con las siguientes:

- **Diseño de Robots Quirúrgicos.** Investigación enfocada en desarrollar y mejorar dispositivos robóticos para asistir a cirujanos en procedimientos médicos, buscando mayor precisión y mínima invasión al paciente.
- **Control de Movimiento del Instrumental.** Se llevan a cabo varios proyectos que se enfocan en mejorar la precisión y estabilidad de los movimientos del instrumental quirúrgico controlado por robots.
- **Planificación Automatizada.** Desarrollo de algoritmos que automatizan la planificación de procedimientos quirúrgicos, optimizando la trayectoria del instrumental y tomando decisiones informadas.
- **Arquitecturas de Control Tolerante a Fallos.** Investigación que busca diseñar arquitecturas de control capaces de detectar y mitigar fallos o errores durante cirugías asistidas por robots, garantizando la seguridad del paciente.
- **Interfaces Persona-Máquina.** Se busca la mejora de la comunicación humano-robot en quirófano para facilitar el trabajo de ambos.
- **Telecirugía.** Se enfoca en desarrollar tecnologías para permitir a los cirujanos realizar operaciones a distancia, controlando robots quirúrgicos sin estar en contacto directo con el paciente.
- **Ensayos Clínicos.** Experimentación in-vitro, con animales y humanos para testear los proyectos desarrollados.

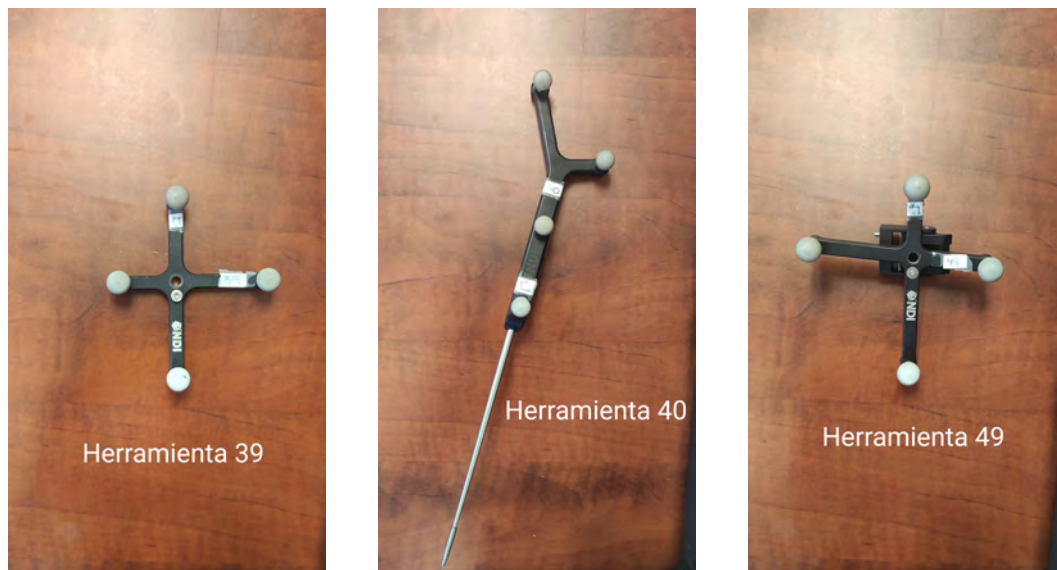
En la actualidad, en el Laboratorio de Robótica Médica, se están desarrollando los proyectos RACE e iSUTURE, financiados por el Ministerio de Ciencia, Innovación y Universidades del Gobierno de España, y por la Junta de Andalucía, respectivamente. El proyecto RACE pretende formalizar y evaluar distintos tipos de sutura realizados durante una cistectomía laparoscópica, así como realizar un análisis y aplicación de la asistencia más eficaz que puede ofrecer una plataforma robótica quirúrgica. Por otro lado, el proyecto iSUTURE tiene como objetivo el desarrollo de una nueva herramienta quirúrgica, basada en una herramienta comercial que pueda acoplarse a uno de los brazos del sistema Da Vinci y manejarse desde su consola de teleoperación sin necesidad de realizar modificaciones en este sistema robótico.

Finalmente, se describe la infraestructura presente en el laboratorio que ha sido utilizada durante este trabajo:

- **Sensores Polaris Spectra.** Fabricados por la empresa Northern Digital Inc. (NDI), se trata de un sistema de medición óptico preciso utilizado en diversas aplicaciones, incluyendo la robótica médica, la navegación quirúrgica y la investigación biomecánica. Permiten la captura de datos en 3D de la posición y orientación de herramientas quirúrgicas mediante la localización de marcadores, que pueden ser pasivos (marcas reflectantes) o activos (luces LED). Pueden ser programados a través del puerto serie gracias al USB integrado que proporcionan. Se pueden ver los sensores en la figura 1.1 y las herramientas que pueden localizar en la figura 1.2.



Figura 1.1: Sensores Polaris Spectra de NDI.



(a) Herramienta 39.

(b) Herramienta 40.

(c) Herramienta 49.

Figura 1.2: Herramientas que captan los sensores Polaris.

- **Universal Robot UR3.** Brazo robot diseñado para la colaboración humano-robot por la empresa Universal Robots. Dispone de 6 grados de libertad y permite un giro infinito de la última articulación, otorgándole unas capacidades únicas para la manipulación de herramientas laparoscópicas de forma remota. El robot dispone de una caja de control manejable mediante pistola de programación con pantalla táctil, aunque también puede accederse al estado del robot así como mandar órdenes de control a través de cable ethernet. Aparece en la figura 1.3.

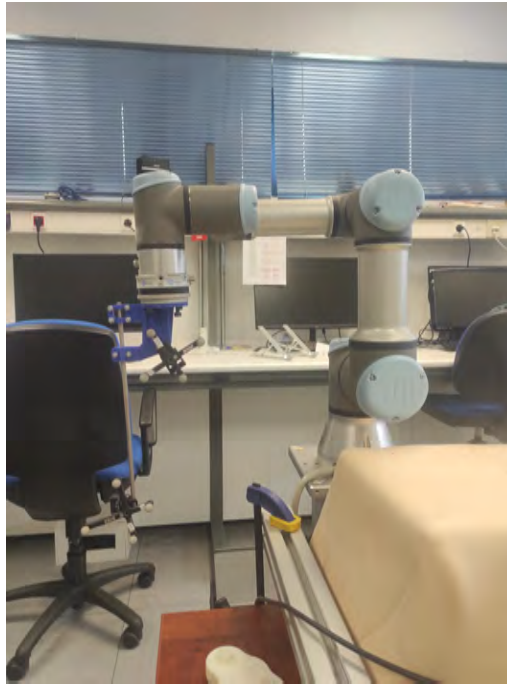


Figura 1.3: Universal Robot UR3.

1.3. Objetivo del trabajo

El objetivo general de este TFG es la realización de un sistema fiable y preciso de seguimiento de instrumental de laparoscopia. Para conseguir el objetivo establecido se usará el material especificado en la sección 1.2, es decir, los sensores Polaris Spectra para obtener la posición de los marcadores pasivos y el robot UR3 para tener movimientos controlados y repetitivos. Finalmente se usa el filtro de Kalman para minimizar los errores producidos durante la obtención de datos.

Partiendo de este objetivo general, se establecen los siguientes objetivos específicos:

- **Determinación de las características funcionales del dispositivo.** Estudio del tipo de comunicación que usa y del funcionamiento de los Polaris Spectra. Acotación del error en las medidas y la latencia en la toma simultánea de posiciones de marcas usando dos sensores.
- **Desarrollo de algoritmos de fusión de medidas.** Realización de procedimientos y algoritmos para proporcionar un comportamiento robusto y preci-

so capaz de suministrar localizaciones exactas del instrumental de laparoscopia a partir de las medidas que aportan cada uno de los dos dispositivos y la unión de las mismas.

- **Validación de los algoritmos.** Fase de experimentación para comprobar el correcto funcionamiento de los sensores y los errores en la localización de las marcas.

Tras alcanzar los objetivos establecidos, se determinará la utilidad de este trabajo y las posibles mejoras que podría introducir en el campo de la cirugía laparoscópica.

1.4. Principales aportaciones

Este TFG presenta un método robusto y preciso para estimar la posición de las herramientas quirúrgicas durante una operación de laparoscopia. Este enfoque tiene el potencial de permitir la recopilación y análisis de datos de cirugías reales, lo que podría mejorar la colaboración humano-robot en un quirófano y facilitar la creación de simuladores de este tipo de intervención que ayuden en la formación de nuevos profesionales.

Para ubicarlo en un contexto real, se puede usar como ejemplo el proyecto PY18-4510 “Sistema Quirúrgico Ciberfísico Para Abordaje Laparoscópico” de la Junta de Andalucía, en el que se desarrolla un sistema ciber-físico donde trabajan de forma conjunta personas y robots en aplicaciones de cirugía laparoscópica. Este proyecto se ha llevado a cabo en el Laboratorio de Robótica Médica del Departamento de Ingeniería de Sistemas y Automática de la Universidad de Málaga [10].

Se muestra el entorno de trabajo en la figura 1.4, extraída del trabajo [10]. En ella se muestra cómo el cirujano utiliza el simulador de laparoscopia a través del manejo de las herramientas, pudiendo seguir la intervención a través de la pantalla de simulación, que presenta las imágenes que estaría captando la cámara, movida por el brazo robot.

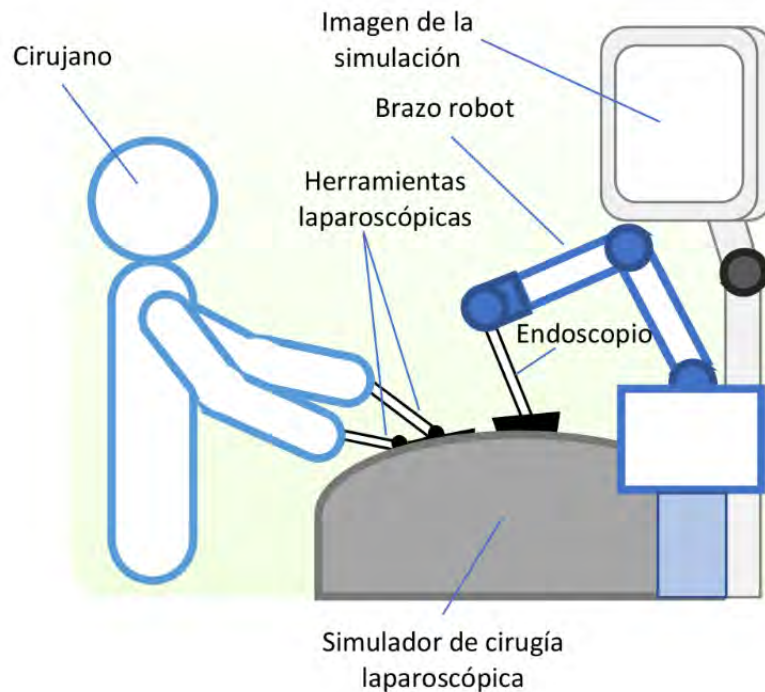


Figura 1.4: Esquema del funcionamiento del entorno de trabajo.

Una posible aplicación sería usar los sensores Polaris Spectra para localizar el instrumental que usa el cirujano. Tras recopilar los datos, se aplicaría el filtro de Kalman para mejorarlos y finalmente se podrían usar estas medidas para mejorar la colaboración humano-robot en futuras intervenciones y estudiar los gestos que se siguen durante las distintas operaciones.

1.5. Estructura del documento

Este documento consta de cinco capítulos, un anexo y referencias bibliográficas. Se detallan a continuación:

- **Capítulo 1. Introducción y visión general.** Se presenta el propósito de realizar este TFG, se indica donde se va a llevar a cabo, la infraestructura utilizada, los objetivos que se pretenden lograr y las principales aportaciones.
- **Capítulo 2. Estado de la cuestión.** se investiga el contexto actual y el entorno que rodea este trabajo. Se explica que es la cirugía laparoscópica

robotizada y en que consiste la localización de instrumental quirúrgico. Tras esto, se detallan las aplicaciones de estos datos obtenidos en la colaboración humano-robot y la creación de simuladores de realidad virtual para ayudar en la formación de nuevos cirujanos.

- **Capítulo 3. Estimación de la posición del instrumental quirúrgico mediante el filtro de Kalman.** Se aborda la importancia de los sistemas de estimación de estados en la localización del instrumental en el campo quirúrgico, concretamente se explica la creación y diseño del filtro de Kalman, usado en este trabajo para para fundir las medidas de ambos sensores con dos métodos distintos y para predecir la localización de los dispositivos ante la falta de medidas al combinar medidas anteriores con el modelo del tipo de movimiento que realiza un cirujano durante una operación de laparoscopia.
- **Capítulo 4. Desarrollo del sistema de localización y resultados.** Se describen los experimentos realizados y la calibración de los dispositivos del entorno de trabajo, es decir, los cambios de sistema de referencia para trabajar en una base común. Seguidamente se describe el flujo de programa, se presentan los resultados obtenidos y finalmente se dan unas conclusiones.
- **Capítulo 5. Conclusiones y líneas futuras.** Concluye el trabajo recopilando las conclusiones a las que se puede llegar tras su realización y se dan posibles líneas futuras.

Capítulo 2

Estado de la cuestión

Contenido

2.1	Cirugía laparoscópica robotizada	12
2.2	Localización del instrumental laparoscópico dentro del campo operatorio	14
2.3	Casos de uso de los datos recopilados en la localización del instrumental laparoscópico	15
2.4	Conclusiones	18

2.1. Cirugía laparoscópica robotizada

La cirugía laparoscópica robotizada es una técnica quirúrgica avanzada en la que se utiliza un sistema robótico para realizar la intervención a través de pequeñas incisiones en el cuerpo del paciente. Para llevarla a cabo el cirujano se posiciona en una consola de mandos y controla los movimientos del robot, que utiliza instrumentos quirúrgicos miniaturizados y una cámara de alta definición al ejecutar la operación.

El concepto de este tipo de cirugías surgió en el año 1985 con el robot PUMA 560 (Programmable Universal Machine for Assembly) [11], que aparece en la figura 2.1, obtenida de la web [12]. Es un brazo robótico desarrollado por la empresa Unimation en la década de 1970 para aplicaciones industriales, sin embargo, por su estructura y capacidad de movimiento, se comenzó a utilizar en biopsias cerebrales mediante punción con el fin de aumentar la precisión. Además, solo eran necesarias pequeñas incisiones en lugar de grandes cortes, lo que conllevaba una recuperación más rápida y menos dolorosa para los pacientes. El uso de este robot en el campo operatorio, fue el primer paso para confiar en este tipo de dispositivos en la medicina.



Figura 2.1: Robot PUMA 560.

En la actualidad, los sistemas robóticos quirúrgicos son ampliamente utilizados gracias a su capacidad para mejorar significativamente las destrezas y capacidades de los cirujanos, ayudando a superar los desafíos asociados con la reducción de libertad de movimiento y la pérdida de la sensación de profundidad en la visión 2D. De este modo, permiten a los especialistas realizar gestos precisos y delicados al escalar los movimientos que este realiza, además de eliminar

temblores fisiológicos, lo que resulta en una reducción de los riesgos y la mejora de la calidad de los resultados [13].

Algunos de los robots más usados en la actualidad para este fin son el sistema Da Vinci [14] y el sistema Zeus [15] que se pueden ver en las figuras 2.2a y 2.2b respectivamente, ambas proceden de la web [16]. El dispositivo Da Vinci es uno de los robots quirúrgicos más conocidos y utilizados en la medicina moderna, fue desarrollado por Intuitive Surgical Inc y está en uso desde el año 2000. Fue diseñado para ayudar a los cirujanos a realizar procedimientos de manera precisa y menos invasiva. Consiste en una consola en la que se sitúa el cirujano para comandar los movimientos deseados y unos brazos robóticos que realizan los movimientos dentro del cuerpo del paciente. Con respecto al robot Zeus, fue desarrollado antes que el Da Vinci por Computer Motion Inc. para realizar procedimientos mínimamente invasivos y asistir a los cirujanos durante las operaciones, de nuevo funciona con una consola y brazos robóticos. A este proceso de comandar al robot desde una consola en lugar de estar directamente sobre el paciente se le denomina telecirugía.



(a) *Da Vinci Surgical System.*



(b) *ZEUS.*

Figura 2.2: Ejemplos de robots usados en cirugías laparoscópicas.

La cirugía laparoscópica robotizada sigue avanzando constantemente, con perspectivas de desarrollar sistemas más precisos y sofisticados en el futuro, lo que ampliará las capacidades para realizar procedimientos cada vez más complejos. En este sistema, la colaboración efectiva entre humanos y robots es fundamental, ya que trabajan de forma conjunta en el quirófano al reproducir el robot lo que el cirujano le comanda.

Un papel crucial en este tipo de colaboración es la localización del instrumental laparoscópico, pues con estos datos se podrán evaluar destrezas quirúrgicas y mejorar las condiciones de realización de este tipo de cirugías [17]. Para lograr

esto, se captan los movimientos que realiza un profesional durante la intervención utilizando técnicas de *tracking* o métodos de *Deep Learning* que se detallarán en el siguiente apartado, 2.2.

2.2. Localización del instrumental laparoscópico dentro del campo operatorio

La medición precisa de la posición y orientación de los instrumentos en laparoscopia es esencial para el desarrollo de simuladores y la investigación de mejoras en este tipo de intervención. Esta se puede llevar a cabo de dos formas distintas: en tiempo real (en línea) o captar todos los movimientos y tras finalizar la intervención, usarlos (fuera de línea). Además, se pueden recopilar los datos mediante dos métodos distintos: métodos de seguimiento (*tracking*) [2] y métodos basados en imágenes o vídeos [18].

Los métodos de *tracking* implican el uso de marcadores que se adhieren a los instrumentos y sensores que los captan y transmiten información sobre la posición y orientación de los mismos. Este sistema de seguimiento procesa la información y proporciona una representación tridimensional de la posición de los instrumentos. Aparece un ejemplo en la figura 2.3 proporcionada por la marca NDI en la web [19].



Figura 2.3: Empleo de métodos de *tracking* en quirófano.

Por otro lado, los métodos basados en imágenes procesan los datos mediante algoritmos de aprendizaje automático, como las redes neuronales y son capaces de clasificarlos. Las técnicas de aprendizaje automático pueden ser entrenadas con imágenes y vídeos previamente etiquetados para aprender a reconocer la posición de los instrumentos en diferentes situaciones. Se puede ver un ejemplo de uso en quirófano en la figura 2.4 procedente de la web [20].



Figura 2.4: Empleo de métodos basados en imágenes en quirófano.

Ambos procedimientos tienen sus ventajas y desventajas. Los métodos de *tracking* proporcionan mediciones precisas de la posición de los instrumentos, pero pueden ser costosos y requieren la adhesión de sensores a las herramientas quirúrgicas, mientras que los métodos basados en imágenes y redes neuronales son más económicos y no requieren la fijación de sensores a los instrumentos, pero pueden tener limitaciones en cuanto a la precisión de las mediciones y la necesidad de un gran volumen de datos para entrenar los algoritmos de aprendizaje automático.

En conclusión, tanto los métodos de *tracking* como los métodos basados en imágenes y redes neuronales son útiles para medir la posición de los instrumentos en laparoscopia, la elección de uno de ellos dependerá de las necesidades y recursos del cirujano y del equipo médico. Estos datos recopilados se usarán con posterioridad para facilitar la colaboración humano-robot y para la creación de simuladores virtuales que contribuyan en la formación de nuevos profesionales en el campo de la cirugía laparoscópica.

2.3. Casos de uso de los datos recopilados en la localización del instrumental laparoscópico

La cirugía laparoscópica ha experimentado avances significativos con la implementación de métodos de *tracking* y redes neuronales para localizar el instrumental en quirófano. Dos áreas de interés destacadas son la colaboración humano-robot y la formación con simuladores virtuales. La colaboración entre cirujanos y robots busca mejorar la eficiencia, mientras que los simuladores virtuales ofrecen un entorno seguro para el entrenamiento. A continuación se deta-

llarán estos dos casos de uso.

Colaboración humano-robot en cirugía laparoscópica

La colaboración humano-robot consiste en la interacción y trabajo conjunto entre humanos y robots en un entorno determinado, con el objetivo de mejorar la eficiencia, la productividad y la seguridad en el lugar de trabajo. Esta colaboración se ha convertido en una tendencia cada vez más popular en diversos campos, pero este TFG en concreto, se centra en la colaboración en la medicina. Se puede ver un ejemplo en la figura 2.5 obtenida del artículo [21], en la que aparece una simulación de intervención laparoscópica donde intervienen al paciente un cirujano y un robot y se localiza todo el instrumental mediante un dispositivo *tracker*.

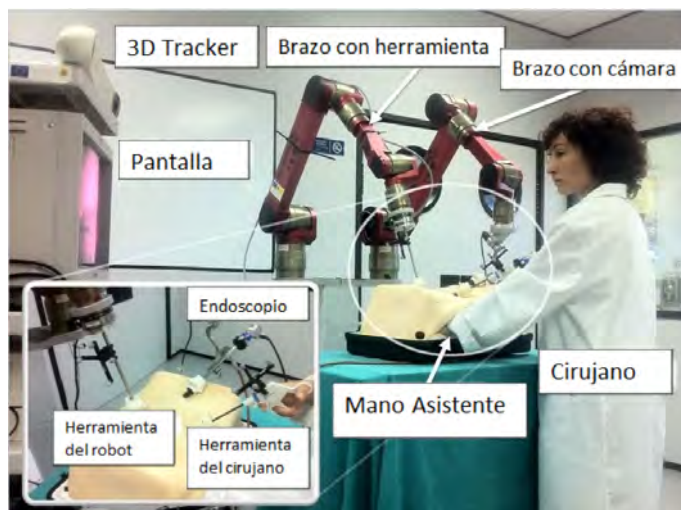


Figura 2.5: Ejemplo de la colaboración humano-robot.

En la actualidad, el uso de robots quirúrgicos ha evolucionado significativamente, y la atención se ha centrado en mejorar la capacidad de los mismos para trabajar de manera autónoma y adaptarse a situaciones cambiantes en el quirófano. Además, se ha prestado atención a la interacción del robot con el entorno [22] y la capacidad del robot para seguir órdenes dadas por el cirujano, requiriendo por tanto un entrenamiento previo de los mismos [21]. Esto ha llevado al desarrollo de nuevas técnicas de control, como el reconocimiento de voz, el movimiento con la cabeza y la detección de la mirada del cirujano [23], para mejorar la comunicación cirujano-robot.

Formación en cirugía laparoscópica

La formación de nuevos cirujanos en el campo de la cirugía laparoscópica conlleva una curva de aprendizaje pronunciada y una práctica repetitiva. Para desarrollar habilidades quirúrgicas, existen varios tipos de métodos de entrenamiento, como laboratorio seco (utiliza modelos de órganos humanos hechos de silicona o plástico), laboratorio húmedo (usa órganos de animales cadavéricos) y simulador de realidad virtual (RV). Además de estos métodos de aprendizaje es crucial entrenar con animales vivos o cadáveres humanos con la guía de un cirujano experto que evite la adquisición de hábitos inapropiados y corrija al alumno [1].

Sin embargo, existen problemas como el bienestar animal, la falta de oportunidades de entrenamiento, la escasez de cirujanos expertos y las restricciones de horas de trabajo. Por estos motivos, hay una demanda urgente de desarrollar un método objetivo de evaluación de habilidades quirúrgicas capaz de mejorar la eficiencia de la transmisión de habilidades [1].

Es así como surgen los simuladores virtuales, capaces de proporcionar un entorno seguro y controlado que permite obtener horas de prácticas, coordinación mano-ojo y evaluación de habilidades de nuevos profesionales [4], lo que facilita la corrección de posibles errores. Esto mejora la curva de aprendizaje y garantiza una transición más fluida hacia la práctica clínica.

Aparecen algunos ejemplos de dichos simuladores de Realidad Virtual (VR) en las figuras 2.6a, 2.6b y 2.6c, concretamente el simulador *Laparoscopy VR* de la empresa *CAE Healthcare*, el *LAP Mentor III* de *Simbionix* y el *LapSim* de *SurgicalScience* respectivamente.



(a) *Laparoscopy VR*. [24].

(b) *LAP Mentor III*. [25].

(c) *LapSim*. [26].

Figura 2.6: Simuladores de RV para la formación en operaciones de laparoscopia.

2.4. Conclusiones

La cirugía laparoscópica robotizada ha experimentado un notable auge en la actualidad debido a la mejora que aporta al escalar movimientos y eliminar temblores, lo que hace que el procedimiento sea más preciso y seguro.

Para continuar con la mejora de esta técnica se recurre a la localización del instrumental laparoscópico, ya sea usando métodos de seguimiento o métodos basados en imágenes.

Con los datos captados se han desarrollado nuevas estrategias de formación para los cirujanos, usándolos en la creación de simuladores virtuales y siendo estos uno de los métodos más relevantes al permitir practicar y perfeccionar las habilidades quirúrgicas en un entorno virtual antes de realizar procedimientos en pacientes reales.

Además de su aplicación en la formación, el seguimiento de instrumental también ha demostrado ser esencial para mejorar la colaboración humano-robot en quirófano. Al medir las localizaciones del cirujano y sabiendo las posiciones del robot, se facilita la coordinación entre ambas partes.

En conjunto, estos avances tecnológicos en cirugía laparoscópica robotizada y el seguimiento de instrumental están transformando la práctica quirúrgica al hacerla más precisa y segura, al tiempo que contribuyen a la formación de futuros cirujanos y mejoran la colaboración entre humanos y robots en el quirófano.

Capítulo 3

Estimación de la posición del instrumental quirúrgico mediante el filtro de Kalman

Contenido

3.1	Introducción	22
3.2	Planteamiento del problema	23
3.3	Diseño del filtro de Kalman	24
3.4	Fusión de medidas de la estimación de la posición mediante el filtro de Kalman	31
3.5	Conclusiones	34

3.1. Introducción

El avance y desarrollo de técnicas de estimación de estados precisos ha sido de utilidad en diversos campos científicos y tecnológicos. Estas técnicas se basan en un modelo matemático del sistema a estimar, y permiten la extracción de información a partir de datos limitados y ruidosos.

Uno de los métodos más utilizados en la actualidad para este fin es el filtro de Kalman [7] gracias a su rapidez y eficiencia junto a la capacidad para lidiar con el ruido y la incertidumbre en las medidas [8]. Utiliza un modelo matemático que describe la dinámica del sistema, complementado por una matriz de covarianza que representa la incertidumbre asociada al mismo.

Dentro del filtro de Kalman existen distintos tipos:

- **Filtro de Kalman básico [7]**. También conocido como filtro de Kalman lineal, es el filtro de Kalman estándar y se utiliza cuando los modelos del sistema y las mediciones son lineales y las incertidumbres se distribuyen normalmente. Es adecuado para problemas con ruido aditivo gaussiano.
- **Filtro de Kalman extendido (EKF) [27]**. Es una extensión del filtro de Kalman básico que permite abordar sistemas no lineales. Utiliza una aproximación de primer orden de la serie de Taylor para linealizar el modelo del sistema y las mediciones en torno a la estimación actual. Aunque el EKF es más flexible que el filtro de Kalman básico, puede ser computacionalmente más costoso debido a la necesidad de calcular jacobianos y matrices de covarianza extendidas.
- **Filtro de Kalman unscented (UKF) [28]**. Es una extensión del filtro de Kalman básico para sistemas no lineales. Este evita la necesidad de calcular jacobianos y puede proporcionar una mejor aproximación del error de propagación usando una técnica llamada transformación *unscented*. UKF es más preciso que EKF en muchas situaciones y puede ser más eficiente computacionalmente.
- **Filtro de Kalman de partículas (PF) [29]**. También conocido como filtro de Monte Carlo, es una variante del filtro de Kalman que se utiliza para estimar el estado en sistemas no lineales y no gaussianos. Utiliza un enfoque basado en un conjunto de partículas (muestras) para estimar la distribución posterior del estado y actualiza estas partículas a través de la propagación del sistema y la ponderación basada en las mediciones. Pueden ser computacionalmente intensivos debido al gran número de partículas necesarias para obtener una estimación precisa.

Se busca crear un sistema de seguimiento robusto y preciso y este filtro es capaz de predecir usando el modelo matemático ante la falta de medidas, fusionar las mismas (provenientes de distintos dispositivos) para disminuir el ruido de medición y combinar las medidas y el modelo matemático para mejorar las estimaciones que devuelve. Al trabajar con un sistema lineal en este trabajo, se ha optado por el filtro de Kalman básico.

3.2. Planteamiento del problema

Como se ha mencionado en la sección 3.1, este TFG aparece con el objetivo de crear un sistema de estimación de la posición de instrumental quirúrgico laparoscópico, es decir, se busca que sea capaz de trabajar en quirófano y que los resultados sean precisos.

Para localizar el instrumental, se hace uso de dos sensores Polaris Spectra de la marca NDI y las herramientas que estos aportan (aparecen en las figuras 1.1 y 1.2 respectivamente). Estos sensores de infrarrojo son capaces de localizar unos marcadores esféricos cubiertos de un material reflectante situados sobre una estructura rígida que asegura que la distancia entre los mismos es constante, como se puede ver en la figura 3.1, donde se especifican las partes de la herramienta 39 (H39). El sensor localiza dichas marcas y por triangulación calcula el centro de la herramienta, aportando además de la posición y la orientación del mismo, el error que se está cometiendo en la medida.

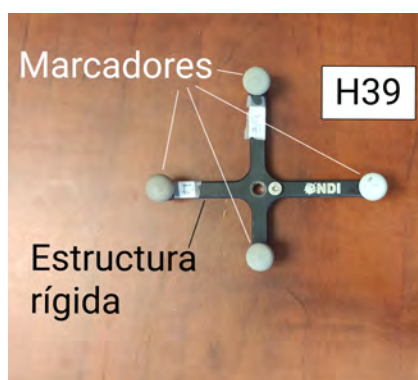


Figura 3.1: Herramienta 39 del sensor Polaris Spectra con las partes de la misma referenciadas.

Partiendo de estos sensores, se desarrolla el entorno de trabajo, que consiste en usar los mismos para localizar el instrumental quirúrgico que se utiliza durante

una operación de laparoscopia. Para ello, se coloca sobre dicho instrumental la H39 buscando saber qué movimientos realiza un cirujano durante este tipo de intervención. Se puede ver un esquema en la figura 3.2, donde aparecen los dos sensores Polaris Spectra al fondo, la H39 sobre el instrumental quirúrgico en el centro de la imagen y este usándose en el simulador de laparoscopia.

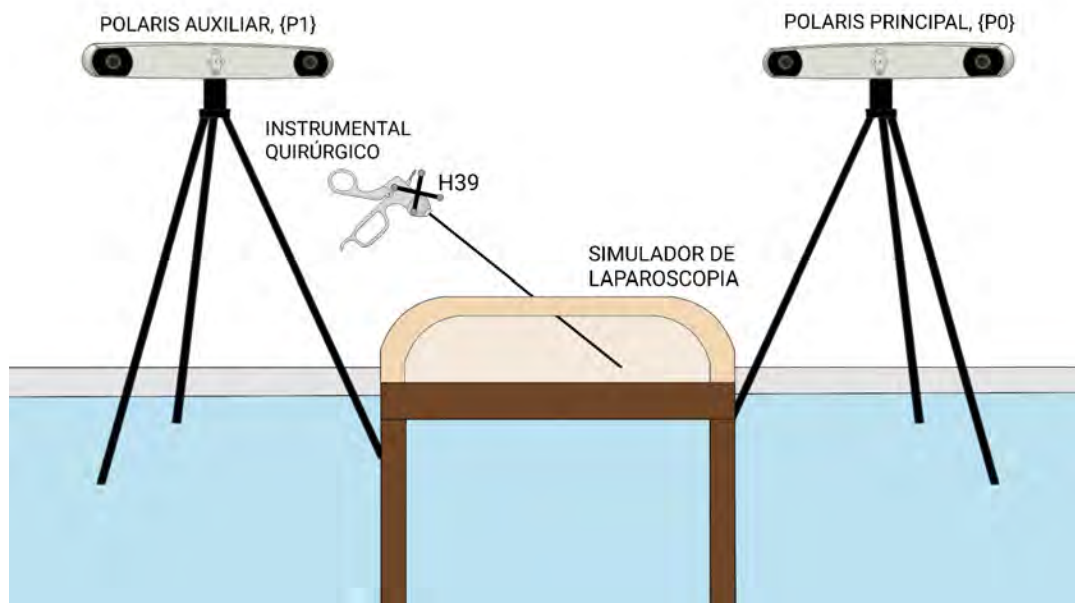


Figura 3.2: Entorno de trabajo.

Es importante tener en cuenta que pueden surgir obstáculos como sombras u otras interferencias que dificulten la correcta detección de los marcadores, lo cual puede ocasionar pérdida momentánea de la herramienta por parte del sensor. Además existen errores de medición. Por ello, tras la recolección de datos es necesario recurrir a métodos de predicción, y siendo el filtro de Kalman una opción reconocida por su eficacia en la estimación de estados y su habilidad para combinar mediciones afectadas por ruido con modelos dinámicos, se implementará el mismo y se comprobará su eficacia durante este trabajo.

3.3. Diseño del filtro de Kalman

El filtro de Kalman es un algoritmo matemático de estimación desarrollado por Rudolf Kalman en la década de 1960 [7]. Se basa en la combinación de un

modelo matemático del sistema a estimar con mediciones y señales de entrada ruidosas para obtener una estimación precisa del estado actual. Se puede simplificar en tres pasos, inicialización, predicción y corrección. Como se muestra en la figura 3.3, el primer paso es la inicialización, a continuación la predicción y tras este, si hay medidas de los sensores, se continúa con la corrección y si no es así se vuelve a predecir.

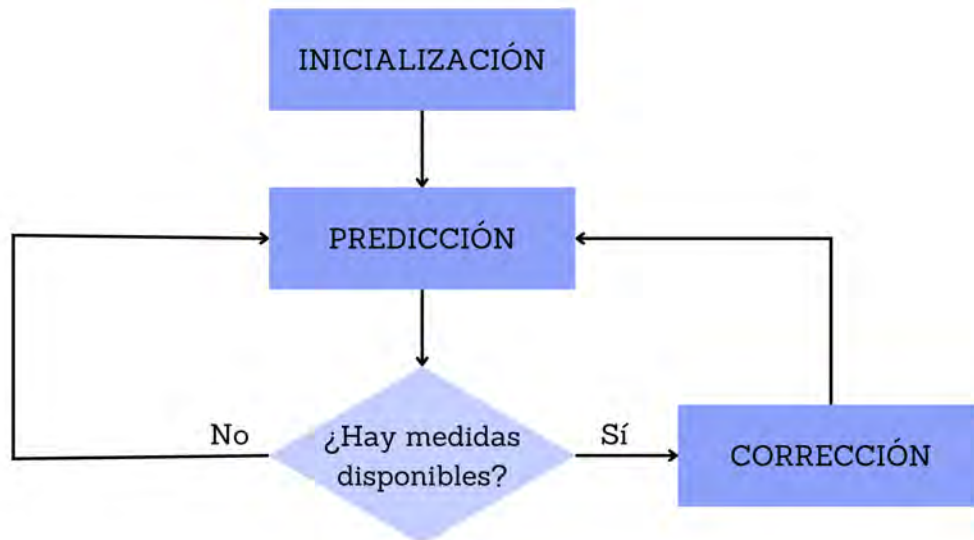


Figura 3.3: Esquema de los pasos a seguir en el filtro de Kalman.

Para implementar el filtro de Kalman, en primer lugar se crea un modelo matemático del sistema que después se usará en los pasos descritos. Para este trabajo, se quiere modelar el movimiento que realiza un cirujano durante una operación de laparoscopia, según la tesis [30], este consiste en un modelo lineal en variables de estados, que considera que el objeto se mueve a velocidad constante con perturbaciones de aceleración.

Para crear dicho modelo se parte de la ecuación de predicción 3.1:

$$\hat{x}_{n+1,n} = F * \hat{x}_{n,n} + G * u_n + w_n \quad (3.1)$$

En ella $\hat{x}_{n+1,n}$ es el vector de estados que se predice para el tiempo $n + 1$ (se usa el acento circunflejo ' \hat{x} ' para diferenciar el vector de estados de la posición en el eje X, ' x '). También aparece $\hat{x}_{n,n}$, vector de estados que se estima para el tiempo n . Los vectores de estado mencionados están compuestos por la posición

y la velocidad en las direcciones X, Y, Z de un sistema de coordenadas global en el entorno de trabajo y aparecen en las ecuaciones 3.2 y 3.3 respectivamente. También aparece $u_{n,n}$ que es el vector de entrada del sistema, en este caso consta de aceleración en las direcciones X, Y, Z, del mismo sistema de referencia global (ecuación 3.4) y w_n que es el ruido de proceso y se toma como cero en este trabajo porque se considera que se ajusta al tipo de movimiento descrito.

$$\hat{x}_{n+1,n} = \begin{pmatrix} x_{(t+1)} \\ y_{(t+1)} \\ z_{(t+1)} \\ v_{x,(t+1)} \\ v_{y,(t+1)} \\ v_{z,(t+1)} \end{pmatrix} \quad (3.2)$$

$$\hat{x}_{n,n} = \begin{pmatrix} x \\ y \\ z \\ v_x \\ v_y \\ v_z \end{pmatrix} \quad (3.3)$$

$$u_{n,n} = \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} \quad (3.4)$$

Continuando con las matrices que aparecen en la misma ecuación, están F, matriz de transición de estado que relaciona el estado actual con el estado siguiente y G, matriz de control que modela cómo afecta la existencia de un control externo aplicado al sistema. El método habitual para su obtención es la resolución del sistema de ecuaciones diferenciales para un sistema dinámico lineal en el dominio del tiempo que aparecen en 3.5, donde \hat{x} es el vector de estados, $\dot{\hat{x}}$ la derivada del mismo, u el vector de entrada, y el vector de salida, A la matriz de estado, B la matriz de entrada, C la matriz de salida y D la matriz de transmisión directa. Tras esto, se calcularían las mencionadas matrices F y G partiendo de la ecuación 3.6 y extrayendo las expresiones 3.7 y 3.8 como se detalla en [31].

$$\left. \begin{aligned} \dot{\hat{x}}(t) &= A * \hat{x}(t) + B * u(t) \\ y(t) &= C * \hat{x}(t) + D * u(t) \end{aligned} \right\} \quad (3.5)$$

$$x(t + \Delta t) = e^{A*\Delta t} * x(t) + B * \left(\int_{\Delta t}^0 (e^{A*t}) dt \right) * u(t) \quad (3.6)$$

$$F = e^{A*\Delta t} \quad (3.7)$$

$$G = B * \left(\int_{\Delta t}^0 (e^{A*t}) dt \right) \quad (3.8)$$

Sin embargo, en este trabajo, para la obtención de las mismas, se resuelve la ecuación 3.1 que aparece como resultado de sustituir los vectores de estado y el vector de entrada en la ecuación 3.9 y donde el subíndice $t + 1$ indica el tiempo actual y t el tiempo anterior y la aplicación de las fórmulas de un movimiento uniformemente acelerado, que indican como se desarrolla la posición y la velocidad en el tiempo. Se pueden ver en las ecuaciones 3.10 y 3.11 donde x_0 representa la posición inicial, x la posición final, t el tiempo, a la aceleración, v_0 la velocidad inicial y v la velocidad final. Finalmente quedan las matrices F y G que aparecen en 3.12 y 3.13 respectivamente donde Δt indica el tiempo entre muestras.

$$\begin{pmatrix} x_{(t+1)} \\ y_{(t+1)} \\ z_{(t+1)} \\ v_{x,(t+1)} \\ v_{y,(t+1)} \\ v_{z,(t+1)} \end{pmatrix} = F * \begin{pmatrix} x_t \\ y_t \\ z_t \\ v_{x,t} \\ v_{y,t} \\ v_{z,t} \end{pmatrix} + G * \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} + w_n \quad (3.9)$$

$$x = x_0 + v_0 * t + a * t^2 / 2 \quad (3.10)$$

$$v = v_0 + a * t \quad (3.11)$$

$$F = \begin{pmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.12)$$

$$G = \begin{pmatrix} \Delta t^2/2 & 0 & 0 \\ 0 & \Delta t^2/2 & 0 \\ 0 & 0 & \Delta t^2/2 \\ \Delta t & 0 & 0 \\ 0 & \Delta t & 0 \\ 0 & 0 & \Delta t \end{pmatrix} \quad (3.13)$$

Se continúa con la ecuación de salida 3.14, compuesta por $\hat{x}_{n,n}$ que corresponde a los valores que toma el vector de estados, v_n el ruido en la medida y la matriz H, matriz de observación, que aparece en la ecuación 3.15, obtenida de la resolución de la ecuación 3.16, que aparece al introducir el vector de estados y la salida que se quiere obtener en 3.14.

$$z_{n,n+1} = H * \hat{x}_{n,n} + v_n \quad (3.14)$$

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \quad (3.15)$$

$$\begin{pmatrix} x_t \\ y_t \\ z_t \end{pmatrix} = H * \begin{pmatrix} x_{(t-1)} \\ y_{(t-1)} \\ z_{(t-1)} \\ v_{x,(t-1)} \\ v_{y,(t-1)} \\ v_{z,(t-1)} \end{pmatrix} + v_n \quad (3.16)$$

Ya obtenidos todos los elementos que aparecen en la ecuación de estados (3.1) y la ecuación de salida (3.14), se ha obtenido el modelo matemático y se puede continuar con la implementación del filtro de Kalman, se tienen que obtener todas las matrices necesarias para las fases de **predicción** y de **corrección**.

Predicción

En la etapa de predicción se utiliza el modelo matemático y las estimaciones anteriores para predecir el estado futuro del sistema y su incertidumbre asociada.

Las ecuaciones usadas son las 3.1 y 3.17, donde F vuelve a ser la matriz de transición de estado, el sufijo T indica que se traspone la matriz, $P_{n,n}$ es la matriz de varianza de la estimación del estado actual, $P_{n,n}$ la matriz de varianza de la predicción del estado anterior y Q la matriz de ruido del proceso.

$$P_{n+1,n} = F * P_{n,n} * F^T + Q \quad (3.17)$$

La matriz $P_{n,n}$ controlan el peso que tienen las medidas tomadas y la estimación del filtro de Kalman, haciendo que sea más o menos importante cada parte en cada iteración. Se inicializa antes de comenzar la fase de predicción, en la inicialización, se usan valores de la diagonal altos si las medidas de los sensores son más fiables que el modelo matemático y valores bajos para lo contrario. El resto de la matriz es cero si los ruidos no están correlados.

Aparece la matriz de inicialización para $P_{n,n}$ que se ha usado en este trabajo en la expresión 3.18. Se puede ver que toma valores altos en la diagonal, ya que los fabricantes de los sensores indican que estos comenten un error máximo de 0.3 milímetros y se considera que las mediciones son más fiables que el modelo matemático. El resto de valores en la matriz son cero porque los ruidos de posición y velocidad en las distintas direcciones no están relacionados. Se continúa con el cálculo de $P_{n+1,n}$ según la ecuación 3.17 y en la siguiente iteración del filtro se actualiza $P_{n,n}$ con el valor de $P_{n+1,n}$ en la iteración anterior.

$$P = \begin{pmatrix} 100 & 0 & 0 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100 \end{pmatrix} \quad (3.18)$$

Por otro lado, la matriz Q representa la incertidumbre asociada al modelo del sistema y cómo se propaga a lo largo del tiempo en ausencia de mediciones. Para su cálculo, se usa la ecuación 3.19 si hay entradas al sistema y la ecuación 3.20 si no las hay. Dentro de estas ecuaciones, σ_a es la desviación estándar del ruido de la entrada al sistema y se obtiene de manera experimental con las aceleraciones que aparecen durante el movimiento.

$$Q = G * \sigma_a^2 * G^T \quad (3.19)$$

$$Q = F * \sigma_a^2 * F^T \quad (3.20)$$

Cuando se acaba esta fase, se actualizan los valores de $P_{n+1,n}$ con el de $P_{n,n}$, y de $\hat{x}_{n,n+1}$ con el de $\hat{x}_{n,n}$.

Corrección

Una vez acabada la fase de predicción y si hay medidas disponibles, se continúa con la fase de corrección, donde se incorporan las mediciones reales del sistema para mejorar la precisión de la estimación y reducir la incertidumbre. El

El filtro de Kalman usa las matrices de covarianza para ponderar la confiabilidad de las mediciones y las predicciones, y ajustar el estado estimado. Estas estimaciones que se han corregido con las medidas reales se utilizarán como punto de partida en la siguiente iteración del filtro, donde se repetirán los pasos de predicción y corrección para disminuir errores a medida que se reciben nuevas mediciones.

Las ecuaciones usadas son 3.21, 3.22 y 3.23, donde aparecen las matrices K_n , conocida como ganancia de Kalman, $P_{n,n}$, actualizada en la fase de predicción, H , matriz de observación, I que representa la matriz identidad y R , matriz de covarianza del ruido de medida.

$$K_n = P_{n,n} * H^T * (H * P_{n,n} * H^T + R_n)^{-1} \quad (3.21)$$

$$\hat{x}_{n+1,n} = \hat{x}_{n,n} + K_n * (z_n - H * \hat{x}_{n,n}) \quad (3.22)$$

$$P_{n+1,n} = (I - K_n * H) * P_{n,n} * (I - K_n * H)^T + K_n * R_n * K_n^T \quad (3.23)$$

La matriz R captura la precisión y la incertidumbre de los sensores utilizados para obtener las mediciones reales del sistema basándose en los errores de las medidas usadas y la relación entre los mismos. Puede ser dada por el fabricante o calculada de forma experimental.

En este trabajo se ha usado la matriz R que aparece en la expresión 3.24, donde $0,3^2$ representa la varianza en los datos del sensor, y es dada por el fabricante (está en milímetros, pues es la medida con la que se trabaja durante los experimentos buscando tener una mayor precisión en los datos).

$$R = \begin{pmatrix} 0,3^2 & 0 & 0 \\ 0 & 0,3^2 & 0 \\ 0 & 0 & 0,3^2 \end{pmatrix} \quad (3.24)$$

Por lo tanto, tras la obtención e inicialización de todas las matrices implicadas en el filtro de Kalman, se puede proceder a su implementación, quedando finalmente, como se muestra en la figura 3.4 el primer paso de inicialización de la matriz P y del vector de estados inicial, este se puede tomar como el primer valor que se obtiene de posición de cualquiera de los dos sensores y cero para los valores de velocidad o como cero en todo el vector si no hay medidas disponibles en la primera iteración. Se continúa con el proceso de predicción que implica dos

ecuaciones, y si hay medidas, con la fase de corrección, donde se aplican tres ecuaciones. Se continúa el proceso de predecir y corregir hasta que el usuario lo decida.

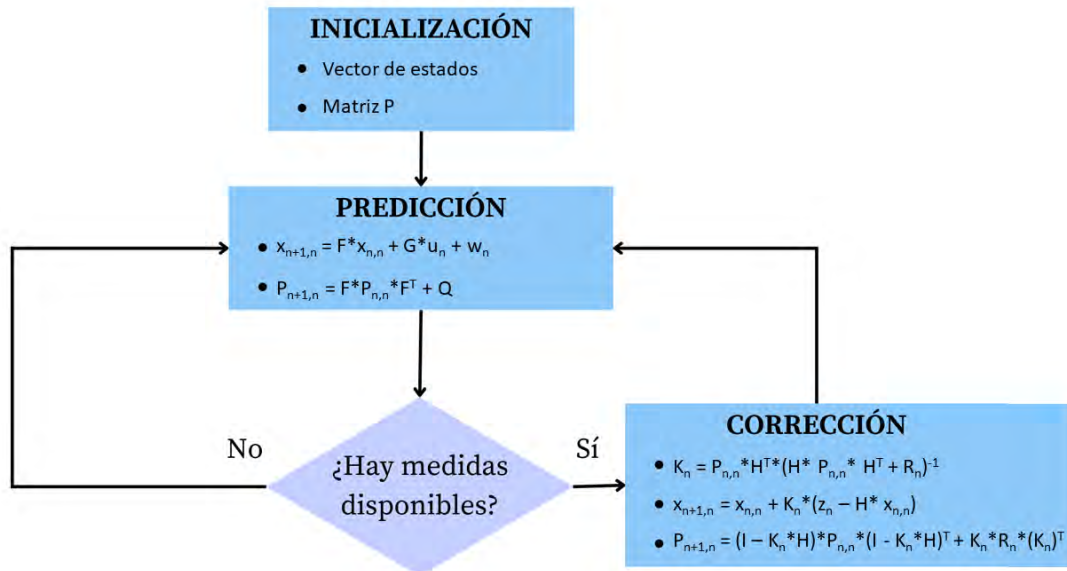


Figura 3.4: Esquema de los pasos a seguir en el filtro de Kalman con sus correspondientes ecuaciones.

3.4. Fusión de medidas de la estimación de la posición mediante el filtro de Kalman

Ya se ha explicado el diseño del filtro de Kalman cuando se reciben medidas de un único sensor en la sección 3.3, a continuación se explicará como usarlo si hay medidas provenientes de dos sensores, donde uno de ellos se considera el principal y otro el auxiliar.

Como ya se mencionado en el apartado anterior, Kalman se basa en dos partes diferenciadas, la predicción y la corrección, pero las medidas solo intervienen en la fase de corrección, por ello, se desarrollan dos métodos distintos para dicha fase durante este trabajo, mientras que la fase de predicción no varía y se continúan usando las ecuaciones 3.1 y 3.17.

Los dos métodos de corrección distintos para la fusión de medidas con dos sensores se presentan a continuación:

- Método 1.** Este método sigue el diagrama de flujo de la figura 3.5, se comienza con la inicialización, se sigue con la predicción y en la parte de corrección (engloba todo el recuadro celeste), si hay medidas del sensor principal se corrige usando únicamente este sensor, si no las hay y tampoco del sensor auxiliar, no se corrige y se vuelve a la fase de predicción y si sí hay medidas del sensor auxiliar se corrige usando únicamente este sensor.

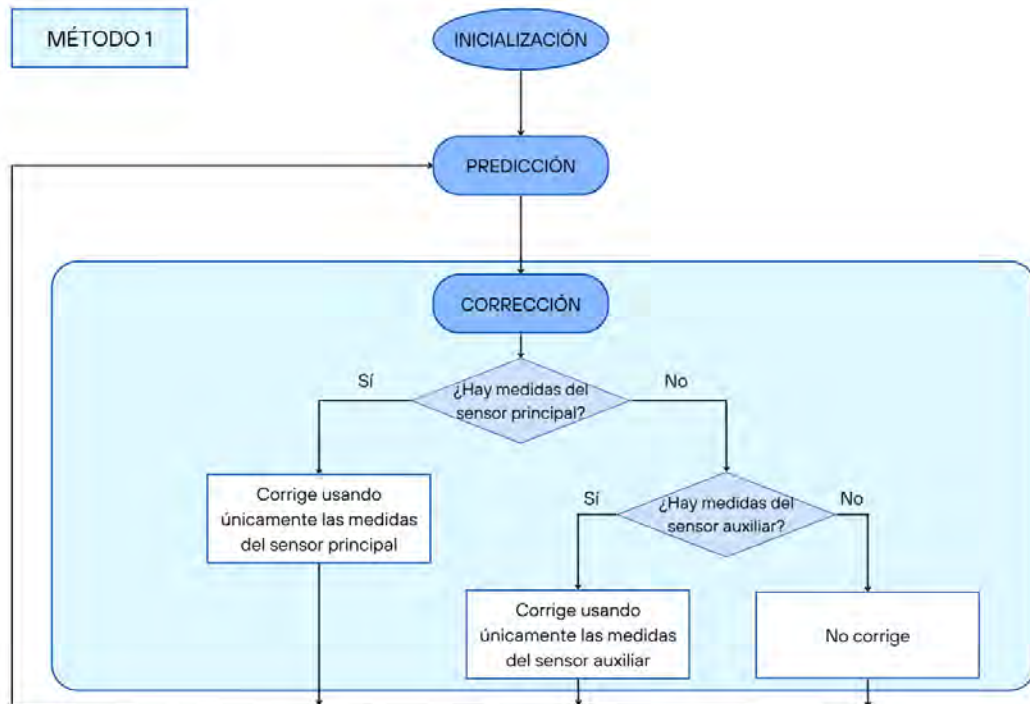


Figura 3.5: Esquema del filtro de Kalman usando el método 1 para corregir.

Se siguen usando las ecuaciones 3.21, 3.22 y 3.23 para la fase de corrección con cada sensor solo si se escoge ese. Es decir, el proceso sigue siendo el mismo que el que se usa cuando hay un único sensor, pero con la diferencia de usar un sensor u otro.

- Método 2.** En el segundo método se sigue el diagrama de la figura 3.6, de nuevo se comienza con la inicialización y se sigue con la predicción y la corrección. Para la fase de corrección se siguen todos los pasos dentro del recuadro celeste, es decir, si hay medidas disponibles de ambos sensores, se corrige dos veces en cada iteración con la información que devuelven ambos, si hay solo de un sensor se usa ese para la corrección y si no hay medidas disponibles, se realiza únicamente el paso de predicción.

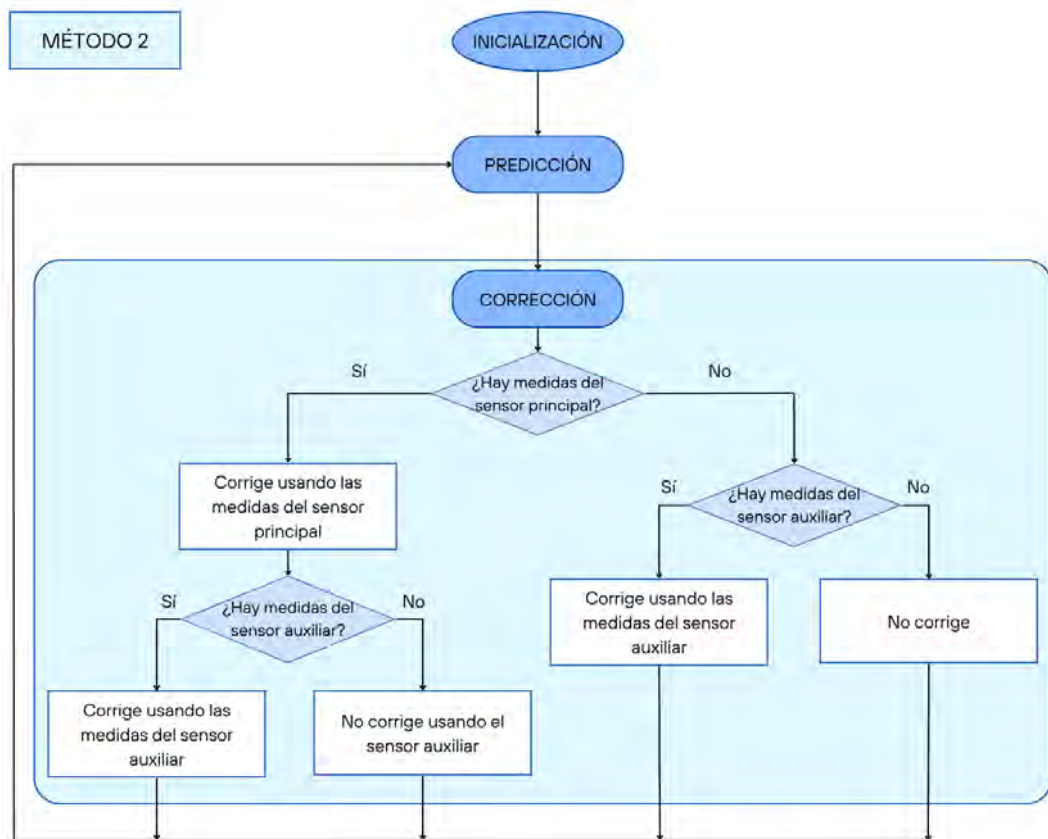


Figura 3.6: Esquema del filtro de Kalman usando el método 2 para corregir.

En este método, de nuevo se usan las ecuaciones 3.21, 3.22 y 3.23 en la fase de corrección para cada sensor si se escoge ese y si hay medida de ambos, se usan con el principal, se actualizan los valores del vector de estados, la ganancia de Kalman K y la matriz P y se vuelven a usar las ecuaciones con el sensor auxiliar.

La elección entre los dos métodos de corrección del filtro de Kalman dependerá de las características de los dispositivos de medición utilizados. Si uno de los dispositivos es más preciso o proporciona un mejor campo de visión del experimento, podría ser preferible utilizar ese dispositivo exclusivamente y recurrir al otro solo en caso de falta de medidas, es decir, utilizar el método 1. No obstante, puede que ambos sensores sean igual de fiables y se quiera usar ambos en cada iteración, optando por el método de corrección 2.

3.5. Conclusiones

En el ámbito de la estimación, la búsqueda de métodos que ofrezcan un alto nivel de precisión es un objetivo fundamental. En este contexto, el filtro de Kalman se destaca como una opción reconocida por su eficacia en diversas aplicaciones.

Como se ha descrito durante este capítulo, el proceso comienza con la inicialización del filtro, donde se establecen las condiciones iniciales y las incertidumbres asociadas a las variables del sistema. A continuación, se procede a la fase de predicción, mediante la cual se estima el estado futuro del sistema en función de su estado actual y el comportamiento previamente modelado, y la fase de corrección que combina las predicciones del sistema con las mediciones reales disponibles.

Es importante mencionar que la fase de corrección aparecen distintos métodos, como se describe en el apartado 3.4. La elección de uno de ellos penderá de las características del sistema en cuestión y de los dispositivos de medición utilizados.

En definitiva, el filtro de Kalman se destaca como una herramienta valiosa para la estimación precisa en numerosos campos, desde sistemas de navegación hasta la robótica médica. Al combinar el diseño adecuado del modelo del sistema y la elección adecuada del método de corrección, se obtiene un sistema robusto y preciso, capaz de adaptarse a cambios y de proporcionar estimaciones confiables incluso en ausencia de mediciones.

Capítulo 4

Desarrollo del sistema de localización y resultados

Contenido

4.1	Introducción	36
4.2	Descripción y diseño del entorno experimental	36
4.3	Calibración de los dispositivos del entorno de trabajo	38
4.4	Descripción del flujo del programa	44
4.5	Variables medidas y análisis estadístico	49
4.6	Conclusiones	56

4.1. Introducción

Para resolver los objetivos planteados en este TFG, se ha diseñado un *set-up* experimental que incluye dos sensores Polaris Spectra para la recolección de datos, un filtro de Kalman para mejorar la precisión de las medidas tomadas y un robot UR3 que garantiza movimientos controlados y conocidos para comprobar la efectividad del filtro y llegar a unas conclusiones.

En términos de la implementación práctica de esta solución, se ha optado por el programa MATLAB debido a su flexibilidad y capacidad para facilitar la comunicación con los sensores. MATLAB ha demostrado ser una herramienta eficiente para la manipulación y análisis de datos, lo que resulta especialmente beneficioso en el contexto de este trabajo.

Además, para establecer la comunicación con el robot UR3, se ha utilizado el entorno de trabajo ROS (Robot Operating System) en Ubuntu, que proporciona una infraestructura sólida y modular para la comunicación entre componentes de robots. Esta comunicación ha sido desarrollada por investigadores del Laboratorio de Robótica Médica del Departamento de Ingeniería de Sistemas y Automática de la Universidad de Málaga.

4.2. Descripción y diseño del entorno experimental

Partiendo del objetivo principal de este trabajo, se comienza el diseño de un experimento que sea capaz de probar la efectividad del sistema desarrollado. Este debe ser sencillo pero realista, por ello, se usan dos sensores Polaris Spectra, diseñados para ser usados en quirófano y el robot Universal Robot UR3 que se describe en la sección 1.2 y es utilizado en el Laboratorio de Robótica Médica del Departamento de Ingeniería de Sistemas y Automática de la Universidad de Málaga para operaciones de laparoscopia por su precisión, flexibilidad y facilidad para la colaboración humano-robot.

Se usan por tanto dos sensores para captar el movimiento del instrumental quirúrgico y un robot para realizar de forma controlada la trayectoria de la herramienta quirúrgica y comparar esta con las estimaciones proporcionadas por el filtro de Kalman con el uso de las medidas de los Polaris buscando saber la efectividad del diseño planteado.

Se ha tomado una fotografía del entorno experimental en el laboratorio que se puede ver en la figura 4.1 donde aparecen todos los componentes implicados, se indican los sensores, tanto el principal como el auxiliar, el robot, su base y su

punta, nombrada como TTP (*Tool Tip Position*), la herramienta quirúrgica y la H39 que usan los Polaris para localizar el instrumental.

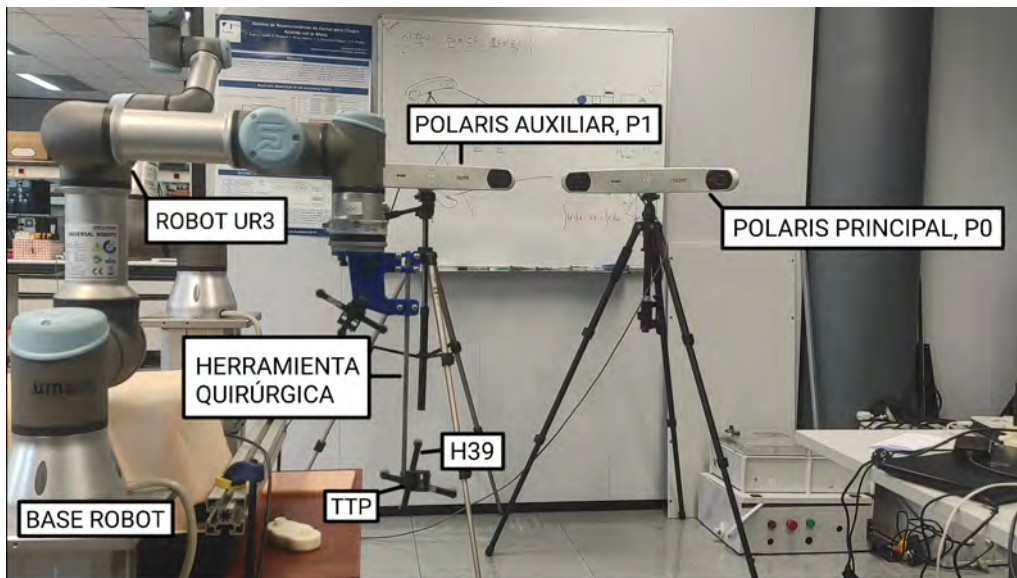


Figura 4.1: Fotografía del entorno experimental.

Partiendo de estos dispositivos se realizan tres experimentos distintos, en cada uno se mueve el robot paralelo a uno de los ejes (X, Y o Z) de la base del robot. Se llevan a cabo dichos movimientos controlados, comandando al robot moverse una distancia mediante ROS.

En la figura 4.2 se muestra un ejemplo del movimiento que realiza el robot de su TTP paralelo al eje Z de su base (aparece en verde). También se pueden ver en el fondo los dos sensores Polaris Spectra para tomar medidas de las marcas de su herramienta situada lo más próxima posible al TTP. La secuencia de movimiento comienza con la fotografía 1, seguida por las 2, 3 y 4, en todas se ha dibujado en color naranja la trayectoria que se sigue y con asteriscos la posición del TTP. Las líneas azules indican donde comienza el movimiento y las rojas donde acaban. Cuando el TTP llega a la línea roja, realiza el movimiento inverso y vuelve a la azul, repitiendo este movimiento en tres ocasiones para cada experimento y trabajando con 2000 medidas en cada uno de ellos.

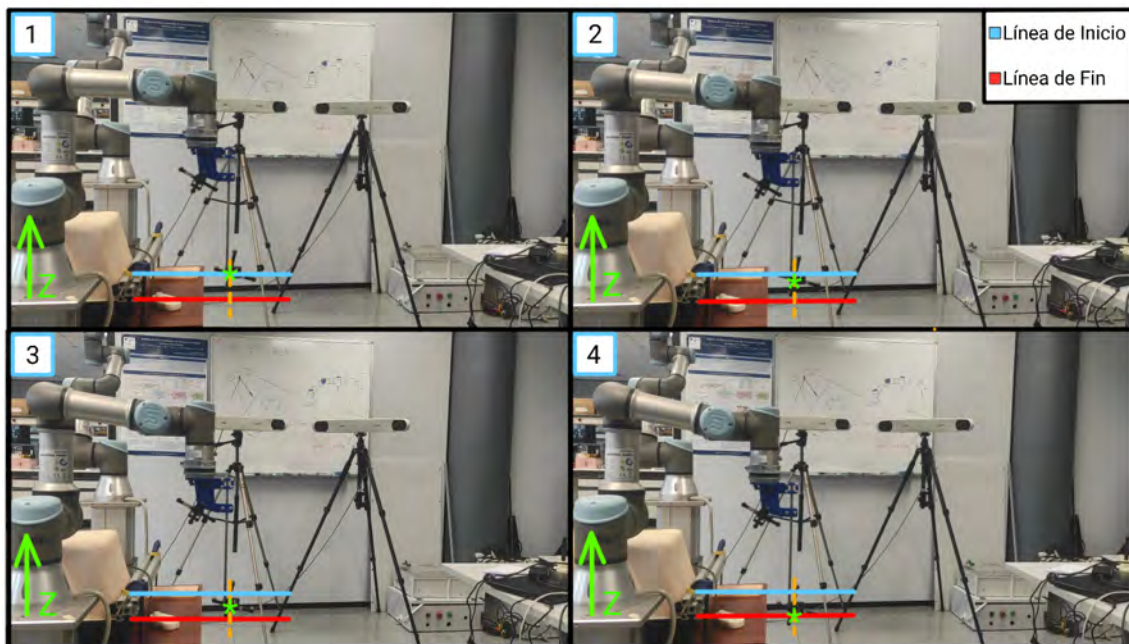


Figura 4.2: Fotografías del robot en movimiento.

Por lo tanto, se mueve el TTP en movimientos controlados, el robot devuelve la posición del mismo y se coloca la herramienta de los Polaris en una posición conocida con respecto al TTP para poder saber esa posición desde los sensores.

Con las medidas de estos tres aparatos, se pasa todo a un sistema de referencia común y se comprueba la efectividad del filtro de Kalman al combinar el modelo que sigue el movimiento de un cirujano durante una operación de laparoscopia con las medidas que aportan los dos sensores Polaris Spectra.

4.3. Calibración de los dispositivos del entorno de trabajo

Para comprobar la efectividad del filtro de Kalman se realiza el montaje experimental que se detalla en la sección 4.2, pero para poder comparar las medidas que devuelven los distintos dispositivos, es necesario que dichas medidas se expresen en la misma base. La base global escogida durante este trabajo y en la que se deberán transformar todas las medidas es el sistema de referencia del Polaris principal.

Hay cinco sistemas de referencia implicados, el del final de la punta de la herramienta del robot $\{TTP\}$, el de su base $\{BR\}$, el de los Polaris principal $\{P0\}$ y auxiliar $\{P1\}$ y el de la H39 $\{H39\}$. Para cambiar las medidas de una base a otra, se busca una relación entre los sistemas de referencia implicados, es decir, una matriz de transformación homogénea que permita realizar el cambio de un sistema de referencia a otro. La notación para estas matrices será aT_b y significa que es una matriz de transformación que indica la posición y orientación del sistema b respecto al sistema de referencia a .

En la figura 4.3 se detallan los sistemas de referencia implicados y las matrices de transformación entre los mismos. Se muestran los Polaris, el robot y abajo a la derecha un *zoom* donde la H39 está acoplada al TTP del robot buscando que se vea con mayor claridad la matriz de transformación entre ambos sistemas de referencia.

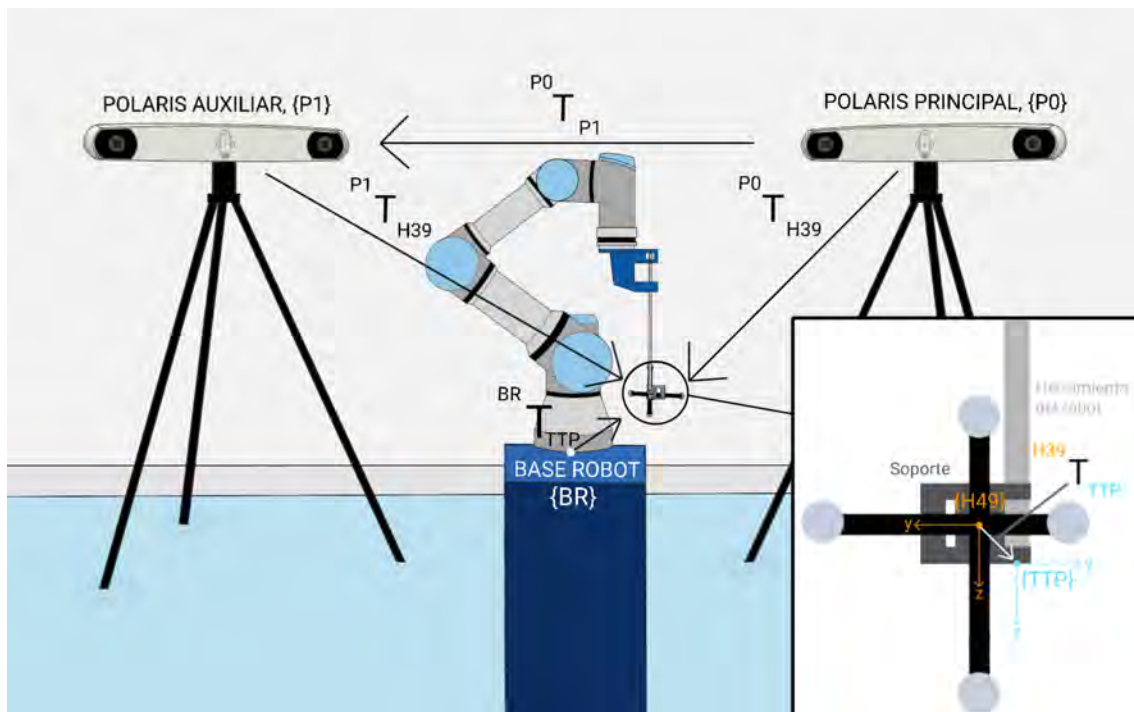


Figura 4.3: Esquema del montaje experimental con las matrices de paso entre los sistemas de coordenadas implicados.

Para las matrices de transformación intermedias que van a cambiar de una base a otra se han usado distintos métodos de obtención y se describen a continuación:

- Matriz ${}^{P0}T_{P1}$. Los métodos para conseguir dicha matriz pueden ser varia-

dos, pero debido a la complejidad para tomar medidas en tres dimensiones manualmente, se ha optado por el procedimiento que se muestra en la figura 4.4. Consiste en situar los sensores en una posición en la que ambos sean capaces de localizar los marcadores de su herramienta (esté dentro de su campo de visión) y tomar 100 medidas de la misma desde ambos sensores con la H39 estática. Por la configuración en la que se han programado, los Polaris devuelven las matrices homogéneas que definen la posición y orientación de la H39 respecto al sistema de referencia correspondiente a cada sensor (${}^{P0}T_{H39}$ y ${}^{P1}T_{H39}$).

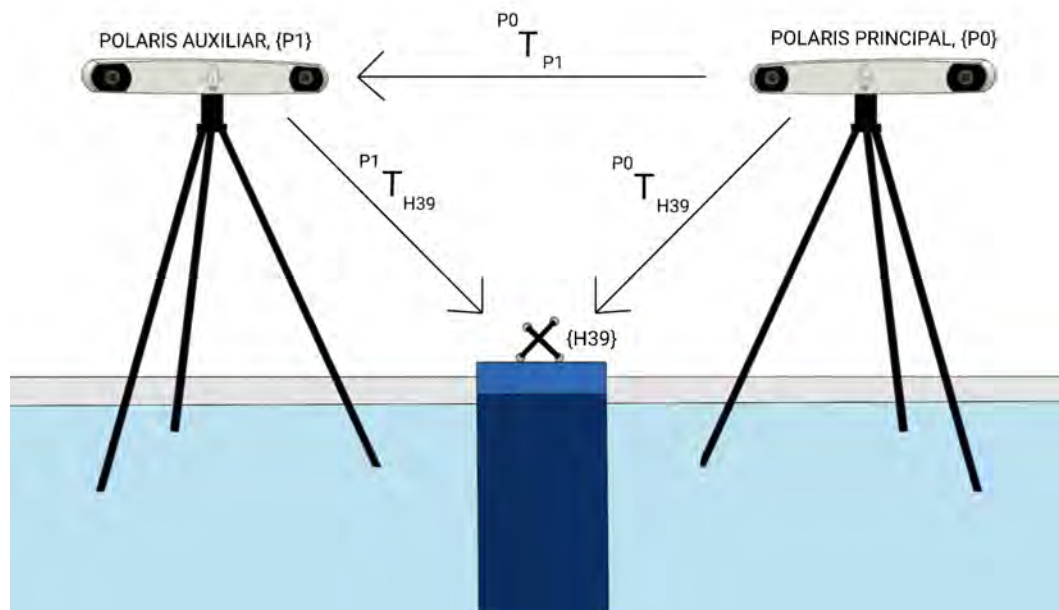


Figura 4.4: Montaje experimental para la calibración de sensores.

Buscando obtener un valor preciso y para minimizar el ruido de medición, tras la obtención de medidas, se hace una media de las mismas y seguidamente se procede al cálculo de la matriz de paso que relaciona un sensor con respecto al otro. Para la obtención de dicha matriz (${}^{P0}T_{P1}$), se parte de la figura 4.4 y se sigue una secuencia específica, que va desde el Polaris 0 a la Herramienta 39 (${}^{P0}T_{H39}$) y desde la Herramienta 39 al Polaris 1 (${}^{H39}T_{P1}$). El sensor devuelve ${}^{P1}T_{H39}$ por lo tanto, empleando la matriz inversa, se obtendrán las matrices necesarias para obtener ${}^{P0}T_{P1}$, quedando finalmente la ecuación 4.1.

$${}^{P0}T_{P1} = {}^{P0}T_{H39} * ({}^{P1}T_{H39})^{-1} \quad (4.1)$$

- Matriz ${}^{TTP}T_{H39}$. Para el experimento se usa un soporte que acopla la H39 al TTP, este permite alinear el eje Z del TTP con el eje Z de la herramienta, en concreto se encaja el agujero de la herramienta con el *Side Locating Pin* del soporte. Esto se puede ver en la figura 4.5, proveniente del manual de los sensores Polaris Spectra [32] donde aparece la herramienta a la izquierda y el soporte a la derecha, además de las partes de la que consta cada uno.

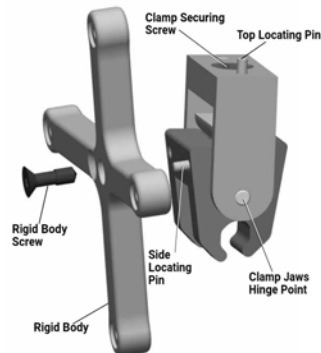


Figura 4.5: Herramienta del sensor y el soporte para las mismas indicando sus partes correspondientes.

El mencionado soporte introduce un desplazamiento y una rotación entre el TTP y el centro de la herramienta. Se puede apreciar en la figura 4.6 donde aparece la herramienta del sensor, el soporte y la herramienta del robot. Se puede ver en color naranja el sistema de referencia $\{H39\}$ y en azul el $\{TTP\}$. En concreto el desplazamiento es de -15 mm en el eje X, -16 mm en el Y y -8 mm en el Z (medido manualmente) y el giro es de π radianes en el eje Z ya que los ejes Y son contrarios. Quedando finalmente el cálculo de la matriz como la expresión 4.2.

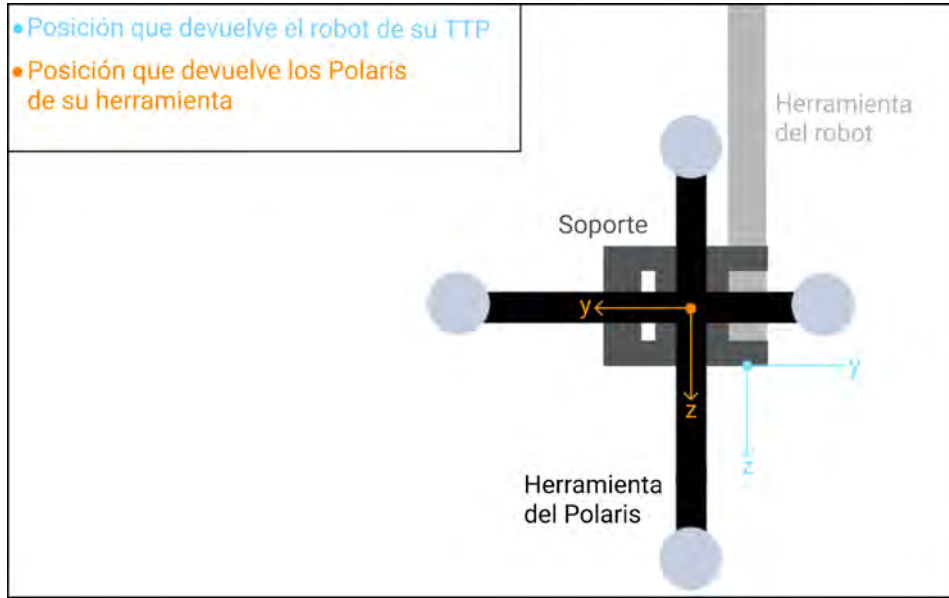


Figura 4.6: Zoom de la herramientas del Polaris (H39) unida al soporte que se acopla sobre la punta de la herramienta del robot (TTP).

$${}^{TTP}T_{H39} = \text{desp}([-15 \ -16 \ -8]) * \text{rot}_Z(\pi) \quad (4.2)$$

- Matriz ${}^{P0}T_{BR}$. Partiendo del esquema de la figura 4.3 se puede ver que la secuencia a seguir es: $P0 \rightarrow H39 \rightarrow TTP \rightarrow BR$. Se usan las matrices ya calculadas (si se sigue el camino en sentido contrario al que marca la flecha en el esquemático, se usa la matriz inversa) y queda finalmente la ecuación 4.3.

$${}^{P0}T_{BR} = {}^{P0}T_{H39} * ({}^{TTP}T_{H39})^{-1} * ({}^{BR}T_{TTP})^{-1} \quad (4.3)$$

Ya calculadas todas las matrices que intervienen en el proceso, para pasar las medidas del Polaris auxiliar y las que devuelve el robot al sistema de referencia del Polaris principal se siguen las ecuaciones 4.4 y 4.5 respectivamente que permiten saber la posición de la herramienta del sensor en el sistema de referencia $\{P0\}$ independientemente del dispositivo que devuelve las medidas.

$${}^{P0}T_{H39} = {}^{P0}T_{P1} * {}^{P1}T_{H39} \quad (4.4)$$

$${}^{P0}T_{H39} = {}^{P0}T_{BR} * {}^{BR}T_{TTP} * {}^{TTP}T_{H39} \quad (4.5)$$

Una vez se tienen las matrices de paso entre Polaris (${}^{P0}T_{P1}$) y entre el $P0$ y el robot (${}^{P0}T_{BR}$) se puede dejar de considerar el experimento estático, se realizan movimientos controlados comandados al robot mediante ROS y se usan las mencionadas matrices para cambiar todas las medidas al sistema de referencia $\{P0\}$.

Durante los experimentos realizados, se ha medido la posición del TTP desde la base del robot, así como la posición de la H39 de los Polaris desde sus respectivos sistemas de referencia. Tras esto, y como se empieza con el experimento estático, se mide el mismo punto durante un periodo de tiempo desde los 3 dispositivos, por lo que se pueden calcular las matrices de transformación.

Sin embargo, cuando el experimento comienza a ser dinámico, se puede ver que se necesita una sincronización de tiempo entre los dispositivos implicados, es decir, para comparar las posiciones medidas cuando se está moviendo el robot, se necesita que las medidas correspondan al mismo instante de tiempo o se estarían comparando distintas posiciones.

Para esta sincronización de tiempo, en primer lugar se pasan todas las medidas al sistema de referencia $\{P0\}$. El siguiente paso es colocar una marca de tiempo para cada medida. Cada aparato comienza un contador interno cuando empieza a medir, por lo tanto se tiene una marca de tiempo para cada medida desde cada dispositivo.

Teniendo todas las medidas en el mismo sistema de referencia y sus marcas de tiempo, se busca el punto en el que se empieza a mover el robot mediante un algoritmo que detecta dicho punto buscando cuando se supera un incremento de posición de un milímetro. Se muestra un ejemplo de funcionamiento de este algoritmo en la figura 4.7, donde se ven dos curvas, que representan posición frente a tiempo, si el color verde es uno de los Polaris y el rojo el robot, ambos comienzan en cero y después empiezan a crecer simulando que se empieza a mover el robot, sin embargo están desfasadas en tiempo. Cuando se mide que la posición x es equivalente a 1 mm con respecto a la posición inicial (en naranja en la gráfica), se captura en qué tiempo ocurren, en este caso, $t1$ y $t2$. Tras esto se restan estos tiempos y se suman a las marcas de tiempo de la curva verde consiguiendo que ambos sistemas estén sincronizados en tiempo y pudiendo comparar, tras esto, las medidas de los distintos aparatos.

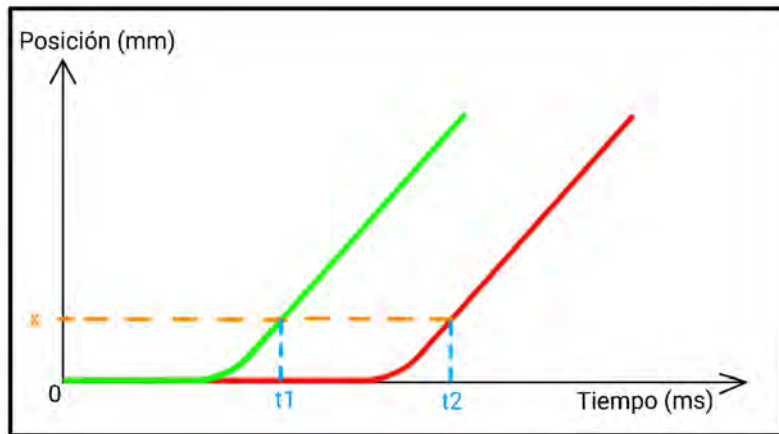


Figura 4.7: Algoritmo de sincronización de tiempos.

Con este procedimiento se obtiene para el movimiento paralelo al eje X una diferencia en el tiempo inicial de **3163.7 ms**, para el paralelo al eje Y **1752.8 ms** y para el paralelo al eje Z, **2506.1 ms**.

Ya calculadas estas diferencias de tiempos, para los datos con cada tipo de movimiento, se suman a las marcas de tiempo del robot (que es el que comienza antes sus mediciones) y ya se habrían sincronizado las medidas en tiempo puesto que los sensores ya miden de forma simultánea porque así se ha implementado en MATLAB (se explicará con posterioridad en el apartado 4.4), por ello, no es necesario cambiar sus marcas de tiempo.

Seguidamente, se realiza la interpolación de los datos ya que aunque ahora sí comienzan a la vez el tiempo de muestreo es distinto para cada dispositivo, y con estos datos se procede al uso del filtro de Kalman, en el que se introduce el modelo matemático y se usan los datos de los dos sensores para corregir.

Finalmente, se considera la posición proporcionada por el robot como la referencia correcta, y se compara con los resultados obtenidos tanto de los sensores como del filtro de Kalman después de la corrección buscando saber la precisión de los sensores al realizar distintos movimientos con el robot y la mejora que puede introducir o no el filtro de Kalman.

4.4. Descripción del flujo del programa

A continuación se describe la secuencia a seguir para usar el sistema de seguimiento de herramientas quirúrgicas. Se puede ver en la figura 4.8 y consiste en un primer paso de toma de medidas desde los sensores, llevándose a cabo

mediante tres fases, la inicialización, la lectura y el cierre de la comunicación, aparece en azul en la imagen. Cuando se acaba el proceso de toma de medidas comienza el segundo paso que consiste en el uso del filtro de Kalman, pudiendo elegir el usuario entre los distintos métodos de corrección que se detallan en la sección 3.4, aparece en verde en la imagen. Por último, en el tercer paso se realiza una interpolación lineal de las medidas y se compararan para medir errores entre los sensores, los datos del robot y los resultados del filtro de Kalman, partiendo de que se consideran como correctas las posiciones que devuelve el robot. Este último paso aparece en naranja.

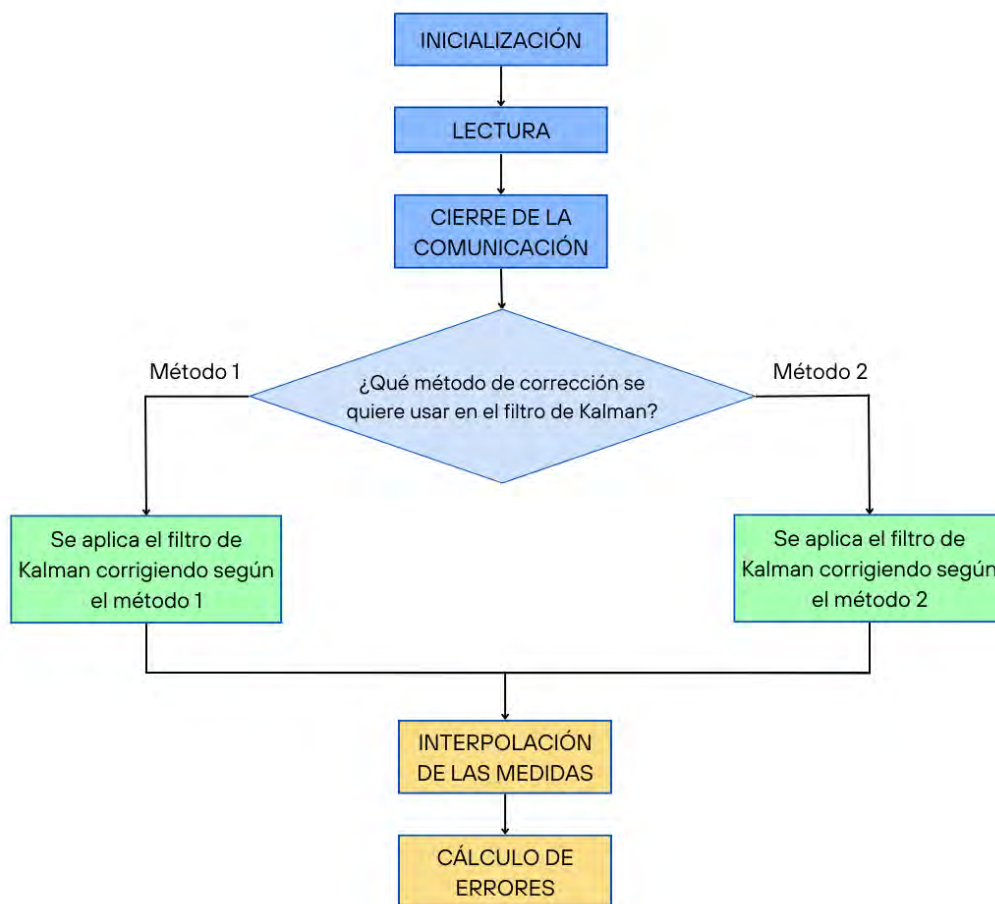


Figura 4.8: Fases del sistema de seguimiento de instrumental quirúrgico.

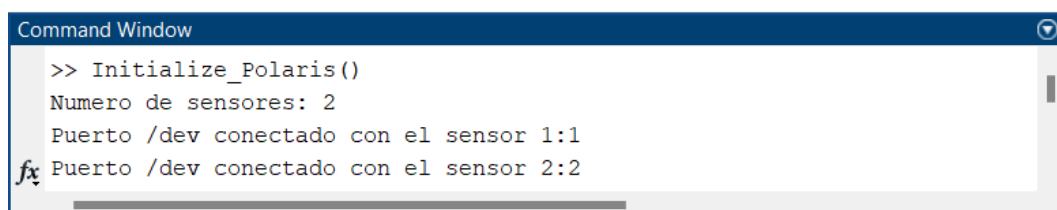
Toda la implementación para trabajar con los sensores, el filtro de Kalman y la interpolación y medida de errores se ha llevado a cabo en el programa MATLAB.

A continuación se detallan los pasos seguidos:

PASO 1. Obtención de medidas

Para trabajar con los Polaris se realiza la inicialización, la lectura y el fin de la comunicación:

- **Inicialización.** Se ha creado una función que permite al usuario conectarse a uno o a dos sensores Polaris Spectra al mismo tiempo. El código está adaptado a Windows y a Ubuntu. La sintaxis aparece en la figura 4.9, donde se llama a la función sin parámetros de entrada y esta pregunta al usuario el número de sensores a utilizar y los puertos serie en los que están conectados.



```
Command Window
>> Initialize_Polaris()
Numero de sensores: 2
Puerto /dev conectado con el sensor 1:1
fx Puerto /dev conectado con el sensor 2:2
```

Figura 4.9: Uso de la función de inicialización para los dispositivos Polaris Spectra de NDI en un dispositivo con sistema operativo Ubuntu.

Esta función abre los puertos serie seleccionados para empezar la comunicación, resetea los sensores, establece la máxima velocidad que permiten para la comunicación a través del puerto serie, los inicializa, selecciona la máxima velocidad del iluminador de infrarrojos y manda la información necesaria para que los Polaris sean capaces de distinguir entre las distintas herramientas que aporta.

Dentro de esta función, se usa otra, cuya sintaxis aparece en la figura 4.10. Los parámetros de entrada son el mensaje en formato *string* ('str') y el número de sensores que se está utilizando simultáneamente ('nPolaris') y devuelve la lectura del puerto serie en el mismo formato, *string* en 'OUT'. Se encarga de escribir y leer del puerto serie y está adaptada al modo de funcionamiento para uno o dos sensores, ya que para tener simultaneidad en tiempo de las medidas de los dos dispositivos es necesario mandar los comandos de inicialización y lectura a la vez. Esta función se usa en el resto de funciones para la comunicación por el puerto serie.

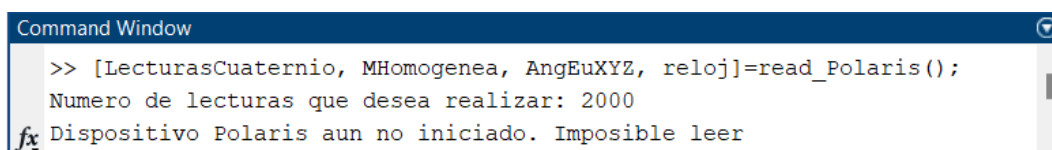


```
Command Window
fx >> OUT = pol_send( str , nPolaris)
```

Figura 4.10: Uso de la función de lectura y escritura por el puerto serie para los dispositivos Polaris Spectra de NDI en un dispositivo con sistema operativo Ubuntu.

La necesidad de tener simultaneidad en las medidas de ambos sensores se detalla en el apartado 4.3.

- **Lectura.** Se ha creado una función que se usa sin parámetros de entrada y devuelve la lectura de todas las herramientas. Pregunta al usuario el número de medidas que desea realizar, se toman todas esas medidas y se guardan en una matriz. Finalmente se cambia el formato, los sensores devuelven cuaternios y esta función devuelve los datos, además de en cuaternios, en ángulos de Euler y como matrices homogéneas, añadiendo otro campo con las marcas de tiempo de las medidas tomadas. Aparece un ejemplo de uso en la figura 4.11 donde se puede ver que se devuelven los datos como cuaternios (LecturasCuaternio), matrices homogéneas (MHomogenea), ángulos de Euler (AngEuXYZ) y las marcas de tiempo para cada medida (reloj).



```
Command Window
>> [LecturasCuaternio, MHomogenea, AngEuXYZ, reloj]=read_Polaris();
Numero de lecturas que desea realizar: 2000
fx Dispositivo Polaris aun no iniciado. Imposible leer
```

Figura 4.11: Uso de la función de lectura para los dispositivos Polaris Spectra de NDI en un dispositivo con sistema operativo Ubuntu.

- **Fin de la comunicación.** Se implementa en una función el fin de la toma de medidas, no tiene parámetros de entrada. En ella se cierra el puerto serie indicando que se ha terminado la comunicación. Aparece un ejemplo de uso en la figura 4.12.



```
Command Window
fx >> end_Polaris()
```

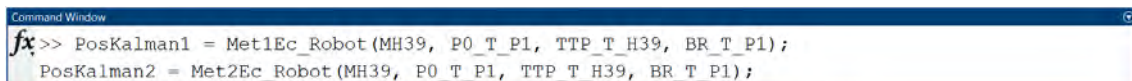
Figura 4.12: Uso de la función de fin de comunicación para los dispositivos Polaris Spectra de NDI en un dispositivo con sistema operativo Ubuntu.

Ya obtenidas las medidas pertinentes se separan los datos devueltos por los sensores agrupándolos por herramienta y sensor que la ha captado. Además, se calcula la velocidad y las aceleraciones entre puntos usando las marcas de tiempo y las posiciones.

PASO 2. Filtro de Kalman

Tras la obtención de la posición de la herramienta de los Polaris desde los sensores, así como de realizar los ajustes necesarios para trabajar en un sistema de referencia común, se aplica el filtro de Kalman usando dichas medidas en la fase de corrección.

Para ello se han creado dos funciones distintas, una con cada método de corrección especificado en el apartado 3.4. La sintaxis utilizada aparece en la figura 4.13, en la primera línea se usa el método de corrección 1 y en la segunda el método 2. Ambas tienen como entradas las medidas de H39 que captan los sensores en 'MH39' y las matrices de paso de un sistema de referencia al otro, en concreto 'P0_T_P1' hace referencia a la matriz ${}^{P0}T_{P1}$, 'TTP_T_H39' a la matriz ${}^{TTP}T_{H39}$ y 'BR_T_P0' a la matriz ${}^{BR}T_{P0}$. Como salida se devuelven las estimaciones en posición que calcula el filtro en el vector 'PosKalman1' para el método 1 y 'PosKalman2' para el método 2.



```
Command Window
fx >> PosKalman1 = Met1Ec_Robot(MH39, P0_T_P1, TTP_T_H39, BR_T_P1);
PosKalman2 = Met2Ec_Robot(MH39, P0_T_P1, TTP_T_H39, BR_T_P1);
```

Figura 4.13: Uso de las funciones del filtro de Kalman con los dos métodos distintos de corrección.

PASO 3. Interpolación y obtención de errores

En un *script* se usan los datos que devuelven los Polaris, los que devuelve el robot y los obtenidos tras aplicar el filtro de Kalman, como se quieren comparar pero no corresponden a los mismos instantes de tiempo, en primer lugar se sincronizan en tiempo y tras esto se recurre a la interpolación lineal entre puntos para comparar exactamente la misma posición desde los distintos dispositivos.

Una vez realizadas las interpolaciones, además de hacer una gráfica para facilitar la comprensión de los resultados obtenidos, se calculan los errores entre los dos sensores, entre cada sensor y el robot y entre los resultados que devuelve el filtro de Kalman y el robot.

Finalmente, se puede recoger el uso de todas las funciones en el pseudocódigo que aparece a continuación. En el paso 1, se toman los datos, los pasos son inicializar los sensores, tomar las medidas, cerrar la comunicación y calcular las matrices de transformación necesarias. Tras la obtención de medidas se aplica el filtro de Kalman usando los dos métodos de corrección y se finaliza con el paso 3, donde se hace una interpolación de los datos y se calculan los errores

en posición entre el Polaris principal y auxiliar (e_P0P1), el error entre el Polaris principal y el robot (e_P0R), el error entre el Polaris auxiliar y el robot y los errores entre las posiciones obtenidas al aplicar el filtro de Kalman con los dos métodos de corrección distintos y el robot (e_K1R para el método 1 y e_K2R para el método2).

Pseudocódigo de los pasos a seguir para el uso del sistema de seguimiento implementado en este trabajo

```

1  % PASO 1 -----
2  % Toma de medidas
3  Inicializacion_de_Sensores()
4  [Medidas, Marcas_de_tiempo] = Toma_de_Medidas()
5  Cierre_de_la_Comunicacion
6  % Calculo de matrices de transformacion
7  P0_T_P1 = Calculo_matriz_transformacion_entre_Polaris(Medidas)
8  TTP_T_H39 = Matriz_transformacion_TTP_H39(Deplazamiento, ...
      Angulo_de_Rotacion, Eje_de_Rotacion)
9  BR_T_P0 = Matriz_transformacion_BR_P0(Medidas)
10
11 %% PASO 2 -----
12 % Calculo de las posiciones con el filtro de Kalman usando el ...
      metodo de correccion 1
13 PosKalman1 = KalmanM1(Medidas, Matrices_Transformacion);
14
15 % Filtro de Kalman usando el metodo de correccion 2
16 PosKalman2 = KalmanM2(Medidas, Matrices_Transformacion);
17
18 %% PASO 3 -----
19 Interpolacion_de_los_Datos
20 % Calculo de errores
21 [e_P0P1, e_P0R, e_P1R, e_K1R, e_K2R] =
22 Calcular_Errores_En_Posicion(Medidas, PosKalman1, PosKalman2, ...
      Matrices_Transformacion);

```

4.5. Variables medidas y análisis estadístico

Tras los experimentos realizados y para comprobar la precisión del método de seguimiento desarrollado, se presentan en este apartado tres partes diferenciadas. En la primera se comparan las medidas de los sensores con las del robot, en la segunda se comparan todas las mediciones de los 3 dispositivos (dos sensores y robot) con los datos que devuelve el filtro de Kalman al implementarse con las funciones desarrolladas durante este trabajo y en el tercero se vuelven a

comparar todos los datos con los que se obtienen al usar el filtro de Kalman pero esta vez con las funciones desarrolladas por MATLAB.

PARTE 1. Comparación de las mediciones de los sensores y el robot.

Tras obtener la posición de la H39 desde el sensor principal y auxiliar y la del TTP del robot desde la base del mismo, se hacen los cambios de sistema de referencia para localizar la herramienta de los sensores en el sistema de referencia $\{P0\}$, se sincronizan las medidas en tiempo, se interpolan y se grafican para poder ver la diferencia entre los tres dispositivos.

Aparece un ejemplo ilustrativo de las medidas de los tres dispositivos en el sistema de referencia $\{P0\}$ para la coordenada Z durante el experimento en el que el TTP del robot se mueve paralelo al eje X de la base del mismo en la figura 4.14. En ella aparece en color verde el Polaris auxiliar, en rojo el principal y en azul el robot. Se puede ver como al principio el experimento es estático y después se realiza tres veces el mismo movimiento hasta acabar de nuevo de forma estática pero en una posición distinta.

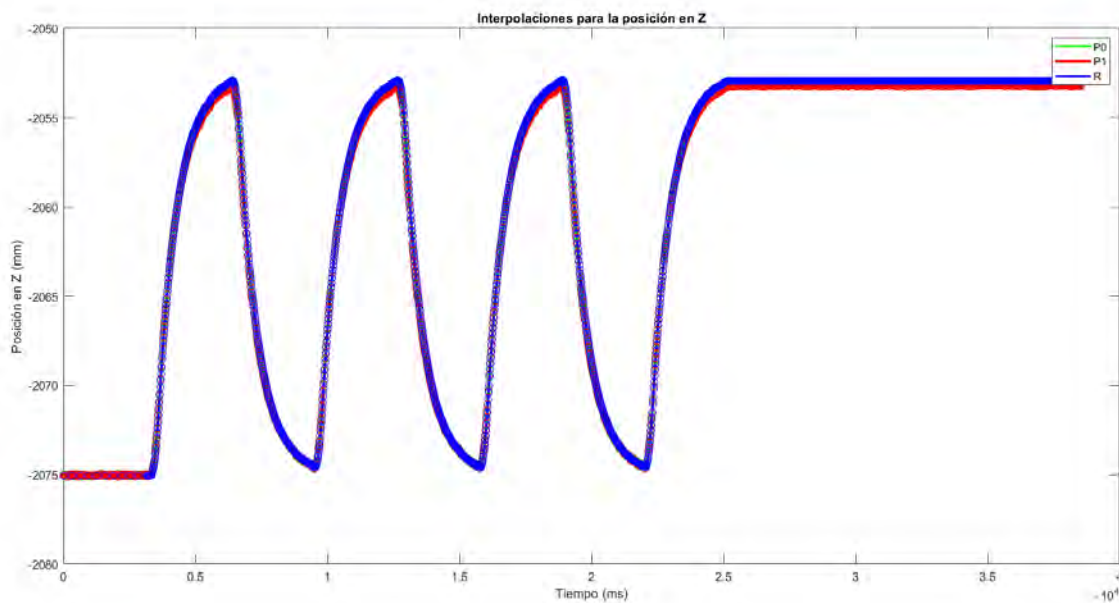


Figura 4.14: Coordenada Z de los tres dispositivos con sus respectivas marcas de tiempo para el movimiento paralelo al eje X de la base del robot.

Como se trabaja con 2000 medidas, la gráfica no es muy clara y por eso se

adjunta un *zoom* en la figura 4.15 donde se ve que el tiempo, que se encuentra en el eje X de la gráfica en milisegundos, no es exactamente el mismo para los tres dispositivos, lo que demuestra la necesidad de la interpolación.

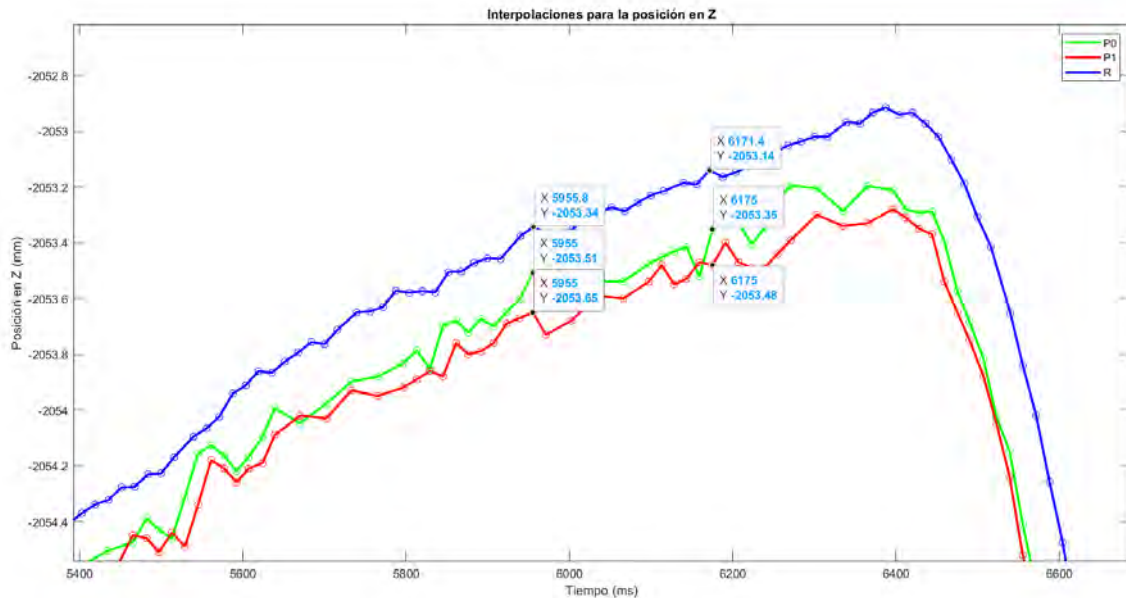


Figura 4.15: Zoom de la gráfica de la coordenada Z de los tres dispositivos con sus respectivas marcas de tiempo para el movimiento paralelo al eje X de la base del robot.

Ya interpoladas las medidas se continúa con la comparación de las mismas. Se restan los valores que devuelven los sensores entre ellos y entre cada uno y el robot para saber los errores máximos que aparecen. Los resultados de estas comparaciones se presentan en la tabla 4.1 donde se muestra en la primera columna el tipo de experimento realizado, es decir, indica si se ha movido el robot en la dirección X, Y o Z con respecto a la base del mismo. En la segunda columna se especifica el eje en el que se ha obtenido el error con respecto al sistema de referencia del sensor principal {P0}. En la tercera, cuarta y quinta columna aparecen los errores entre el Polaris principal y el auxiliar, entre el Polaris principal y el robot y entre el Polaris auxiliar y el robot respectivamente.

De los resultados obtenido se puede extraer que el error máximo **entre sensores** se observa en el eje Z, alcanzando un valor de **0.1285 (mm)**, mientras que entre el **Polaris principal y el robot** y el entre el **Polaris auxiliar y el robot** los valores máximos son **0.6858 (mm)** y **0.7490 (mm)** respectivamente, y ambos se producen en el eje Y. Además, estos errores máximos ocurren durante el movimiento paralelo al eje X de la base del robot.

Dirección del movimiento	Eje {P0}	Error entre Polaris (mm)	Error Polaris principal-robot (mm)	Error Polaris auxiliar-robot (mm)
Movimiento paralelo al eje X de la base del robot:	X	0,0585	0,5455	0,5154
	Y	0,1093	0,6543	0,7490
	Z	0,1285	0,2985	0,3744
Movimiento paralelo al eje Y de la base del robot	X	0,0577	0,2949	0,2431
	Y	0,0870	0,6858	0,1956
	Z	0,0967	0,1209	0,1587
Movimiento paralelo al eje Z de la base del robot	X	0,0916	0,1880	0,1866
	Y	0,0797	0,1363	0,1917
	Z	0,0688	0,3307	0,3225

Tabla 4.1: Tabla que recoge los errores máximos en posición entre los distintos dispositivos para los distintos movimientos.

PARTE 2. Comparación de las mediciones de los sensores y el robot con el filtro de Kalman usando las funciones desarrolladas durante este trabajo.

Durante este trabajo se ha desarrollado la implementación del filtro de Kalman haciendo uso del diseño que se presenta en la sección 3.3 en el programa MATLAB.

Tras el uso de las funciones desarrolladas, se grafican de nuevo las mediciones de los tres dispositivos y se incluyen los resultados del filtro de Kalman con los dos métodos utilizados. Para mayor claridad, se aplica *zoom* y se muestra en la figura 4.16. En esta imagen, el color verde corresponde a las mediciones del Polaris principal, el rojo a las mediciones del Polaris auxiliar y el azul a las mediciones del robot en las interpolaciones de la coordenada Z durante el movimiento paralelo al eje X del robot. También se presentan las correcciones efectuadas por el filtro de Kalman utilizando el primer método en color magenta, y el segundo método en color cian.

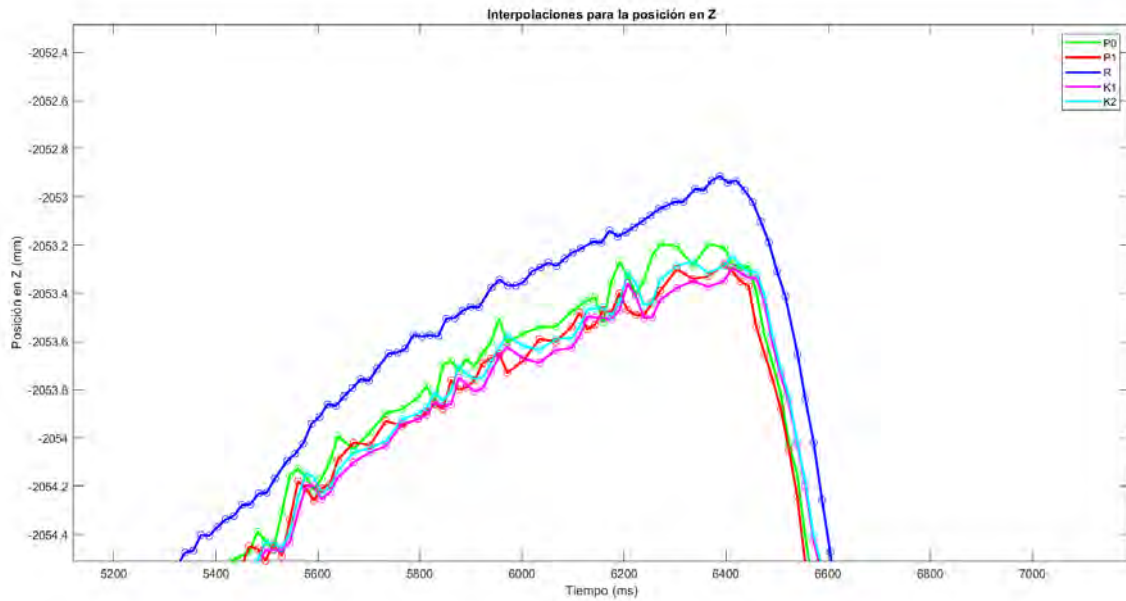


Figura 4.16: Zoom de la coordenada Z de los tres dispositivos y las correcciones de Kalman con sus respectivas marcas de tiempo.

Los errores máximos obtenidos en cada experimento para cada dirección de los ejes del sistema de referencia del sensor principal se muestra en la tabla 4.2. Las columnas 1 y 2 indican lo mismo que la tabla 4.1, la tercera el error entre los resultados obtenidos del filtro de Kalman con el método 1 y los resultados que devuelve el robot e igual con la columna 4 pero con el método 2.

Al aplicar dichas funciones (una para cada método de corrección) se obtienen un error máximo de **0.2743 (mm)** para el **método 1** en la dirección del eje X del sistema de referencia {P0} con el movimiento paralelo al eje Z de la base del robot, mientras que para el **método 2** se alcanza un valor de **0.2801 (mm)** en la misma dirección y para el mismo tipo de movimiento.

Dirección del movimiento	Eje {P0}	Error entre el filtro de Kalman (método 1) y el robot (mm)	Error entre el filtro de Kalman (método 2) y el robot (mm)
Movimiento paralelo al eje X de la base del robot	X	0,0326	0,0421
	Y	0,2056	0,2156
	Z	0,2344	0,2408
Movimiento paralelo al eje Y de la base del robot	X	0,0315	0,0423
	Y	0,2102	0,2110
	Z	0,1951	0,1932
Movimiento paralelo al eje Z de la base del robot	X	0,2743	0,2801
	Y	0,0681	0,0781
	Z	0,0639	0,0654

Tabla 4.2: Tabla que recoge los errores máximos en posición entre los resultados obtenidos mediante el filtro de Kalman implementado con las funciones desarrolladas en este trabajo y el robot.

PARTE 3. Comparación de las mediciones de los sensores y el robot con el filtro de Kalman usando las funciones desarrolladas por MATLAB.

MATLAB ha desarrollado varias funciones para implementar el filtro de Kalman, en concreto: *trackingKF*, *predict* y *correct*. Para su uso, se escoge un modelo a velocidad constante en tres dimensiones y se usan los datos de los sensores Polaris Spectra. Se comparan estos resultados con las mediciones que devuelve el robot como se puede ver en la figura 4.17. Aparece en verde los datos del Polaris auxiliar, en rojo los del Polaris principal, en azul los del robot, en magenta los datos que devuelve Kalman con las funciones desarrolladas durante este trabajo para el método de corrección 1 y en cian para el método 2. Finalmente, aparece en negro los resultados que devuelve el filtro de Kalman usando para su implementación las funciones desarrolladas por MATLAB y corrigiendo según el método 1 y en amarillo para el método 2.

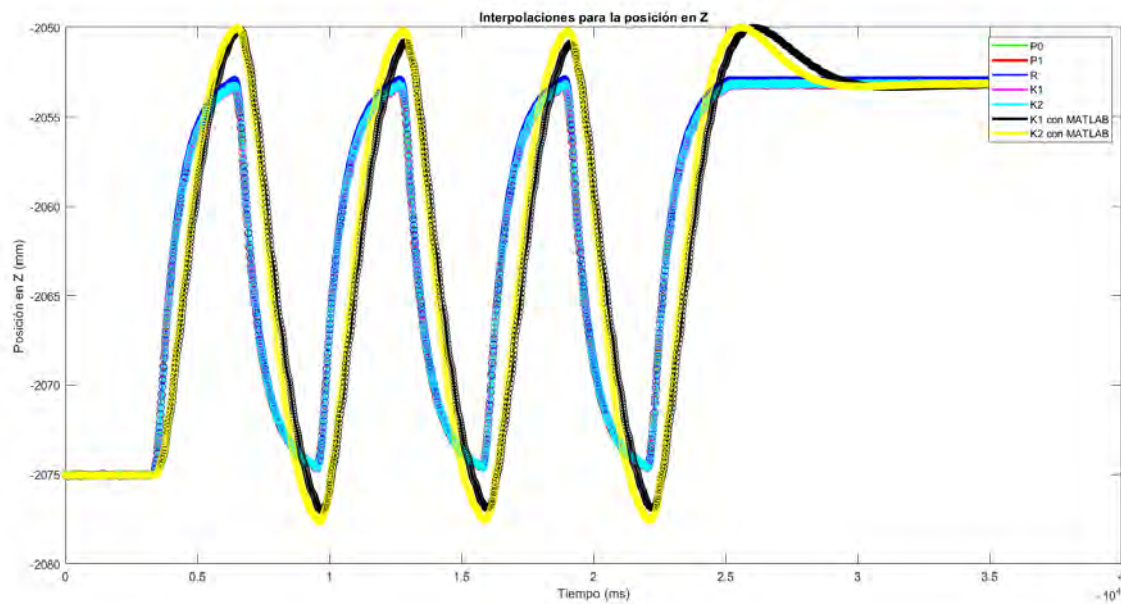


Figura 4.17: Gráfica de la coordenada Z de los tres dispositivos y las correcciones de Kalman calculadas usando las funciones dadas por MATLAB con sus respectivas marcas de tiempo.

Se vuelven a recoger los datos obtenidos en una tabla, en concreto en la tabla 4.3, que representa los mismos datos que la tabla 4.2 pero usando para el filtro de Kalman las funciones dadas por MATLAB.

Con dichas funciones se obtienen un error máximo de **6.6398 (mm)** para el **método 1**, mientras que para el **método 2** se alcanza un valor de **5.7739 (mm)** ambos se obtienen para el movimiento paralelo al eje X de la base del robot en el eje Z del sistema de referencia {P0}.

Dirección del movimiento	Eje {P0}	Error entre el filtro de Kalman (método 1) y el robot (mm)	Error entre el filtro de Kalman (método 2) y el robot (mm)
Movimiento paralelo al eje X de la base del robot	X	0,4748	0,4031
	Y	6,2282	5,3807
	Z	6,6398	5,7739
Movimiento paralelo al eje Y de la base del robot	X	0,3094	0,2793
	Y	3,4967	3,0201
	Z	3,2131	2,7584
Movimiento paralelo al eje Z de la base del robot	X	4,9644	4,5518
	Y	0,0702	0,0811
	Z	0,3766	0,3766

Tabla 4.3: Tabla que recoge los errores máximos en posición entre los resultados obtenidos mediante el filtro de Kalman implementado con las funciones desarrolladas por MATLAB y el robot.

El nuevo error máximo es de **6.6398 (mm)**, frente a los **0.2801 (mm)** que se obtiene con las funciones desarrolladas en este trabajo, lo que demuestra la precisión del método desarrollado que tiene en cuenta las perturbaciones de aceleración para este tipo de movimiento, el ajuste de los parámetros como el error máximo de los sensores y el hacer más fiable el modelo matemático o las medidas.

4.6. Conclusiones

En este capítulo se ha detallado el procedimiento seguido para la creación de los distintos experimentos, los cambios de sistemas de referencia implicados, el proceso seguido para crear el sistema de seguimiento y se han presentado los resultados obtenidos.

Si se analizan los resultados, se puede comprobar que los errores máximos de 0.6543 mm y 0.7390 mm entre los sensores y el robot (antes de aplicar Kalman) se reducen a 0.2743 mm y 0.2801 mm tras la aplicación del filtro implementado con las funciones desarrolladas durante este TFG. Lo cual demuestra la efectividad del filtro de Kalman al lograr una disminución del error en un rango del 57 al 63%.

Estos resultados son indicativos del éxito alcanzado en la mejora de la precisión del sistema de medición de movimiento del cirujano durante una operación

de laparoscopia, respaldando la eficacia y la utilidad práctica de la solución propuesta para crear un sistema robusto de seguimiento de herramientas quirúrgicas.

Capítulo 5

Conclusiones y líneas futuras

Contenido

5.1 Conclusiones	60
5.2 Líneas futuras	60

5.1. Conclusiones

Durante el desarrollo de este TFG, se ha abordado la creación de un sistema robusto de seguimiento de herramientas quirúrgicas. Mediante la utilización de los sensores Polaris Spectra y la implementación del filtro de Kalman, se ha buscado crear un método preciso capaz de funcionar incluso ante la falta de medidas de los sensores.

Tras la investigación del contexto y el entorno actual en el campo del seguimiento quirúrgico, se diseñó un filtro de Kalman y se llevaron a cabo experimentos que permitieron comprobar la efectividad del trabajo desarrollado.

Para el diseño del filtro, el modelo matemático se basó en el propuesto por María Belén Estebanez en su tesis [30], el cual describe los movimientos que realiza un cirujano durante una operación de laparoscopia. Se ajustaron parámetros, como la fiabilidad de las medidas, el error que se podía producir en las mismas (dado por el fabricante) y la desviación típica de las aceleraciones durante el movimiento, obtenida a partir de los experimentos realizados.

Los errores máximos obtenidos con los datos experimentales y la aplicación del filtro de Kalman con dos implementaciones distintas para cada método de corrección desarrollado en este trabajo, se recogen en las tablas 4.1 y 4.2. El análisis de los mismos ha permitido comprobar la notable mejora que introduce el filtro en este procedimiento, logrando reducir el error máximo en posición hasta en un 63 %.

En resumen la implementación del filtro de Kalman y la experimentación han demostrado la utilidad de dicho filtro, capaz de mejorar las medidas al eliminar ruidos, ser capaz de fusionarlas y dar predicciones incluso ante la falta de las mismas.

5.2. Líneas futuras

La disminución del error en posición que ha aportado el filtro de Kalman en este trabajo abre el camino para su desarrollo en cuanto a la estimación de la orientación. Esto permitiría una precisión aún mayor en el seguimiento de herramientas quirúrgicas y aumentaría la utilidad y confiabilidad del sistema en aplicaciones clínicas. Otra posible tarea a implementar partiendo de este TFG sería conseguir que el sistema de seguimiento trabaje en tiempo real.

Además, los resultados obtenidos en este trabajo pueden tener un impacto significativo en aplicaciones prácticas como la creación de simuladores de reali-

dad virtual, que a su vez podrían ser empleados para el entrenamiento de nuevos cirujanos, mejorando así sus habilidades y conocimientos antes de realizar intervenciones reales.

También se podrían detectar fallos de movimiento del robot en el quirófano y gestos que realicen los profesionales durante la intervención, para mejorar la colaboración humano-robot o emplear esta recolección de datos en el aprendizaje por demostración, que consiste en que una máquina aprenda a realizar una tarea observando ejemplos de cómo lo hace un experto (en lugar de depender solo de reglas o datos), al imitar las acciones y decisiones del profesional para llevar a cabo la tarea de manera efectiva.

En conclusión, el uso de los sensores Polaris Spectra junto con la implementación del filtro de Kalman, ha demostrado una mejora de los datos recopilados dando lugar a un sistema de seguimiento de herramientas quirúrgicas preciso y robusto.

Bibliografía

- [1] K. Ebina, T. Abe, K. Hotta, M. Higuchi, J. Furumido, N. Iwahara, M. Kon, S. Komizunai, Y. Kurashima, H. Kikuchi, R. Matsumoto, T. Osawa, S. Murai, T. Tsujita, K. Sase, X. Chen, N. Shinohara, and A. Konno, “Development and validation of a measurement system for laparoscopic surgical procedures in practical surgery training,” in *2023 IEEE/SICE International Symposium on System Integration (SII)*, Atlanta, GA, USA, 2023, pp. 1–6.
- [2] J. L. Grantner, A. H. Kurdi, M. AlGailani, I. Abdel-Qader, R. G. Sawyer, and S. Shebrain, “Multi-thread implementation of tool tip tracking for laparoscopic surgical box-trainer intelligent performance assessment system,” in *2019 IEEE 23rd International Conference on Intelligent Engineering Systems (INES)*, 2019, pp. 000 131–000 136.
- [3] A. Boonkong, D. Hormdee, S. Sonsilphong, and K. Khampitak, “Surgical instrument detection for laparoscopic surgery using deep learning,” in *2022 19th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, 2022, pp. 1–4.
- [4] B. Baby, M. Chasmai, T. Banerjee, A. Suri, S. Banerjee, and C. Arora, “Representation learning using rank loss for robust neurosurgical skills evaluation,” in *2022 IEEE International Conference on Image Processing (ICIP)*, Bordeaux, France, 2022, pp. 4048–4052.
- [5] J. Neumann, A. Uciteli, T. Meschke, R. Bieck, S. Franke, H. Herre, and T. Neumuth, “Ontology-based surgical workflow recognition and prediction,” in *2022, Journal of Biomedical Informatics*, vol. 136, 2022.
- [6] H. Huynhnguyen and U. A. Buy, “Toward gesture recognition in robot-assisted surgical procedures,” in *2020 2nd International Conference on Societal Automation (SA)*, Funchal, Portugal, 2021, pp. 1–4.

- [7] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME—Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.
- [8] Q. Li, R. Li, K. Ji, and W. Dai, "Kalman filter and its application," in *2015 8th International Conference on Intelligent Networks and Intelligent Systems (ICINIS)*, 2015, pp. 74–77.
- [9] A. Pozo-Ruz, M. García-Alegre, A. Ribeiro, L. García, D. Guinea, and F. Sandoval, "Localización de vehículos: Fusión de medidas gps y odometría," *Mundo electrónico*, 06 2023.
- [10] J. M. Herrera López, "Sistema de detección y clasificación de gestos del cirujano para la automatización de maniobras en cirugía laparoscópica," Ph.D. dissertation, Universidad de Málaga, Málaga, 2023.
- [11] P. Corke and B. Armstrong-Helouvy, "A search for consensus among model parameters reported for the puma 560 robot," in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, 1994, pp. 1608–1613 vol.2.
- [12] H.-I. Lin and C. Lee, "Measurement of the robot motor capability of a robot motor system: A fitts's-law-inspired approach," *Sensors*, vol. 13, no. 7, p. 8412–8430, 2013. [Online]. Available: <http://dx.doi.org/10.3390/s130708412>
- [13] C. Martínez Ramos, "Robótica y cirugía laparoscópica," *Cirugía Española*, vol. 80, no. 4, pp. 189–194, 2006. [Online]. Available: <https://www.elsevier.es/es-revista-cirugia-espanola-36-articulo-robotica-cirugia-laparoscopica-13093226>
- [14] P. L. Anderson, R. A. Lathrop, S. D. Herrell, and R. J. Webster, "Comparing a mechanical analogue with the da vinci user interface: Suturing at challenging angles," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 1060–1065, 2016.
- [15] M. Ghodoussi, S. Butner, and Y. Wang, "Robotic surgery - the transatlantic case," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, vol. 2, 2002, pp. 1882–1888 vol.2.
- [16] Álvaro García Sánchez, "Zeus y da vinci: Presente y futuro de la cirugía," Sitio web, 2018, [En línea]. Accedido el: 06/07/23. [Online]. Available: <https://robotics2017site.wordpress.com/2018/02/27/zeus-y-da-vinci-presente-y-futuro-de-la-cirugia/>

- [17] J. A. Sánchez-Margallo, F. M. Sánchez-Margallo, J. B. Pagador Carrasco, I. Oropesa García, and J. Gómez Aguilera, Enrique J. and Moreno del Pozo, "Utilidad de un sistema de seguimiento óptico de instrumental en cirugía laparoscópica para evaluación de destrezas motoras," *Cirugía Española*, vol. 92, no. 6, pp. 421–428, 2014.
- [18] A. González, F. Gayá Moreno, P. Orden, F. Del Pozo Guerrero, F. Sánchez-Margallo, and E. Gómez Aguilera, "Método de seguimiento 3d del instrumental quirúrgico mediante análisis de vídeo laparoscópico," *Actas del XXVI Congreso Anual de la Sociedad Española de Ingeniería Biomédica (CASEIB 2008)*, 01 2008.
- [19] NDI, "Ndi's premier optical tracker for oem robotic surgery systems," Sitio web, [En línea]. Accedido el: 06/07/23. [Online]. Available: <https://www.ndigital.com/optical-measurement-technology/polaris-vega-xt/>
- [20] STERIS, "Cámara quirúrgica free5™," Sitio web, [En línea]. Accedido el: 06/07/23. [Online]. Available: <https://www.medicalexpo.es/prod/steris/product-70182-444718.html>
- [21] E. Bauzano, B. Estebanez, I. Garcia-Morales, and V. F. Muñoz, "Collaborative human–robot system for hals suture procedures," *IEEE Systems Journal*, vol. 10, no. 3, pp. 957–966, 2016.
- [22] S.-Y. Ko, J. Kim, D.-S. Kwon, and W.-J. Lee, "Intelligent interaction between surgeon and laparoscopic assistant robot system," in *ROMAN 2005. IEEE International Workshop on Robot and Human Interactive Communication, 2005.*, 2005, pp. 60–65.
- [23] S. Li, J. Zhang, L. Xue, F. J. Kim, and X. Zhang, "Attention-aware robotic laparoscope for human-robot cooperative surgery," in *2013 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2013, pp. 792–797.
- [24] "Da vinci simnow," [En línea]. Accedido el: 18/05/23. [Online]. Available: <https://www.intuitive.com/en-us/products-and-services/da-vinci/education/simnow>
- [25] M. T. Simulators, "Simulador de laparoscopia – lap mentor iii," Sitio web, [En línea]. Accedido el: 06/07/23. [Online]. Available: <https://morethansimulators.com/simulador/simulador-laparoscopia/>
- [26] L. C. y Tecnología, "Simulador de cirugía laparoscópica- lapsim®," Sitio web, [En línea]. Accedido el: 06/07/23. [Online]. Available: <https://labtop.pe/es/products/simulador-de-cirugia-laparoscopica-lapsim>

-
- [27] H. Khazraj, F. Faria da Silva, and C. L. Bak, "A performance comparison between extended kalman filter and unscented kalman filter in power system dynamic state estimation," in *2016 51st International Universities Power Engineering Conference (UPEC)*, 2016, pp. 1–6.
- [28] E. Wan and R. Van Der Merwe, "The unscented kalman filter for nonlinear estimation," in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*, 2000, pp. 153–158.
- [29] Y. Xu, K. Xu, J. Wan, Z. Xiong, and Y. Li, "Research on particle filter tracking method based on kalman filter," in *2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, 2018, pp. 1564–1568.
- [30] B. Campos, "Diseño e implantación de un sistema multimodal para un asistente robótico," Ph.D. dissertation, Universidad de Málaga, Málaga, 2013.
- [31] A. Becker, "Kalman filter - tutorials and resources," Sitio web, 2017, [En línea]. Accedido el: 13/04/23. [Online]. Available: <https://www.kalmanfilter.net/default.aspx>
- [32] N. D. Inc., *Polaris Spectra Tool Kit Guide*, Northern Digital Inc., 2006.

