



UNIVERSIDAD  
DE MÁLAGA



E.T.S.  
INGENIERÍA  
INFORMÁTICA

UNIVERSIDAD DE MÁLAGA

TRABAJO FIN DE GRADO  
GRADO EN INGENIERÍA DE LA SALUD

---

**DISEÑO DE UNA INTERFAZ BASADA EN LA  
DETECCIÓN DE GESTOS PARA CIRUGÍA  
LAPAROSCÓPICA ASISTIDA POR LA MANO**

**DESIGN OF AN INTERFACE BASED ON GESTURES  
DETECTION FOR HAND-ASSISTED  
LAPAROSCOPIC SURGERY**

---

**Autor:** Arturo de Guzmán Manzano

**Director:** Isabel García Morales

**Codirector:** Enrique Bauzano Núñez

**Departamento:** Ingeniería de Sistemas y Automática

MÁLAGA, Noviembre de 2016



**Resumen:** En el presente trabajo se describe el diseño e implantación de una interfaz de comunicación basada en la detección de gestos del cirujano para cirugía laparoscópica asistida por la mano.

Para el desarrollo del sistema de detección de gestos se va a utilizar el dispositivo hardware Leap Motion, el cual permite registrar secuencias de parámetros físicos de la mano, útiles para la identificación de patrones de movimiento. Esta información es el punto de partida para la implementación de un sistema de reconocimiento de gestos basado en los Modelos Ocultos de Markov. Dicho sistema consta de una fase inicial de entrenamiento de los modelos, y otra fase de reconocimiento que permite distinguir gestos a partir de secuencias de características observables. Además, el sistema desarrollado debe integrarse en la plataforma robótica del grupo de Robótica Médica del departamento de Ingeniería de Sistemas y Automática, la cual dispone de una arquitectura basada en ROS para conectar todos los sistemas y dispositivos de los que disponen. Tanto el hardware necesario como el software desarrollado se describen a lo largo de esta memoria, explicando y justificando cada decisión tomada, para finalmente terminar con una serie de experimentos que validan el funcionamiento sistema de reconocimiento.

**Palabras clave:** interfaz, reconocimiento de maniobras, entrenamiento, gesto, cirugía laparoscópica asistida por la mano (CLAM).

**Abstract:** This work describes a communication interface design and implantation based on surgeon gestures detection for hand-assisted laparoscopy surgery.

Leap Motion will be used for developing the gestures detection system. It consists of a hardware that allows identifying movement patterns by recording sequences of physical parameters of the hand. This information is the starting point to implement a gesture recognition system based on Hidden Markov Models. This system has an initial training phase of the models, and other recognition phase that distinguishes gestures from sequences of observable characteristics. Besides, the system created must be integrated in a robotic platform developed by the Medical Robotic group, which has an architecture based on ROS to connect all its systems and devices. This group belongs to the Systems and Automation Engineering Department at the Malaga University. Both hardware and software architecture developed in this work are described along this memory, explaining and justifying every decision that has been taken. After that, a set of experiments will be conducted to verify the proper functioning of the recognition system.

**Keywords:** interface, maneuvers recognition, training, gesture, hand-assisted laparoscopic surgery (HALS).



# Índice general

<b>Capítulo 1: Introducción</b> .....	<b>1</b>
1.1. Introducción a la robótica médica .....	1
1.2. Motivación del trabajo de fin de grado .....	2
1.3. Objetivos .....	3
1.4. Estructura de la memoria .....	5
<b>Capítulo 2: Técnicas e interfaces en robótica quirúrgica</b> .....	<b>7</b>
2.1. Introducción .....	7
2.2. Descripción de las técnicas de cirugía mínimamente invasiva .....	7
2.3. Interfaces quirúrgicas .....	9
2.5. Conclusiones .....	13
<b>Capítulo 3: Planteamiento del problema</b> .....	<b>15</b>
3.1. Introducción .....	15
3.2. Escenario quirúrgico en HALS .....	15
3.3. Arquitectura del sistema .....	17
3.4. Conclusiones .....	19
<b>Capítulo 4: Algoritmos de entrenamiento y reconocimiento</b> .....	<b>21</b>
4.1. Introducción .....	21
4.2. Esquema general del sistema desarrollado.....	21
4.3. Adquisición de datos .....	24
4.4. Procesado de datos .....	26
4.4.1. Fase de preprocesado de datos .....	26
4.4.2. Fase de codificado de datos .....	29
4.5. Entrenamiento de librería de gestos .....	34
4.6. Reconocimiento de gestos.....	37
4.7. Conclusiones .....	42

<b>Capítulo 5: Implantación y experimentos.....</b>	<b>45</b>
5.1. Introducción .....	45
5.2. Implantación del sistema.....	45
5.2.1. Arquitectura hardware.....	45
5.2.2. Arquitectura software.....	48
5.3. Descripción de los experimentos .....	51
5.3.1. Objetivo.....	51
5.3.2. Gestos registrados .....	52
5.3.3. Batería de experimentos y resultados.....	58
5.3.4. Discusión.....	63
5.4. Conclusiones .....	63
 <b>Capítulo 6: Conclusiones y trabajos futuros .....</b>	<b>65</b>
6.1. Conclusiones .....	65
6.2. Futuras líneas de desarrollo.....	65
 <b>A. Algoritmo K-means.....</b>	<b>67</b>
A.1. Introducción .....	67
A.2. Descripción del algoritmo .....	67
A.3. Implementación del algoritmo .....	67
 <b>B. Modelos Ocultos de Markov.....</b>	<b>69</b>
B.1. Introducción .....	69
B.2. Descripción del modelo.....	69
B.3. Algoritmo Forward-Backward .....	71
B.4. Algoritmo de Baum-Welch .....	73
 <b>Glosario de términos.....</b>	<b>77</b>
 <b>Referencias.....</b>	<b>79</b>

# Capítulo 1: Introducción

## 1.1. Introducción a la robótica médica

En la actualidad la medicina en general, y particularmente la cirugía, dependen cada día más de la tecnología. Con los avances producidos tanto en las técnicas de diagnóstico por imagen como en el instrumental, se está modificando progresivamente la forma de proceder durante las intervenciones quirúrgicas. Así, dichos avances han permitido realizar operaciones complejas que hace unos años eran impracticables, lo que ha supuesto sustanciales beneficios tanto para el paciente como para el cirujano.

Por otro lado, las técnicas quirúrgicas también han evolucionado, dando lugar a especialidades como la cirugía mínimamente invasiva (CMI), en la que se pretenden minimizar las incisiones realizadas sobre el paciente. Dichas incisiones se utilizan para insertar instrumentos especiales, denominados de caña larga, y una óptica que permita la visualización de las estructuras anatómicas internas del paciente durante la operación.

Sin embargo, ésta forma de intervención conlleva una serie de limitaciones intrínsecas que dificultan la ejecución del procedimiento. En primer lugar, la interacción con el cuerpo del paciente exclusivamente mediante el instrumental produce que el cirujano no perciba el tacto o la textura de las estructuras anatómicas. Por otro lado, el uso de un monitor permite únicamente una visión en 2D, lo que produce una pérdida de profundidad del campo de visión, y dificulta la coordinación entre los ojos del cirujano y la actuación realizada por sus manos. Además, la propia metodología de operación, basada en introducir el instrumental quirúrgico a través de incisiones en la pared abdominal, provoca que el abdomen del paciente actúe como fulcro que ejerce un efecto palanca sobre el instrumental, y dificulta la libertad de movimientos.

Además, en cualquier tipo de intervención basada en las técnicas de CMI, se requiere de un ayudante que asista al cirujano en tareas tales como el manejo de la cámara y/o alguna de las herramientas, para lo cual es necesaria una coordinación entre ambos. En dicha colaboración, surgen los problemas de interpretación de la acción a realizar por el ayudante cuando el cirujano manda una orden, la falta de precisión en la colocación de la cámara, así como la inestabilidad de la imagen debido al pulso humano. Ante estos inconvenientes, surge la idea de incorporar nuevos procedimientos que puedan ser abordados con la ayuda de la robótica, gracias a un enfoque basado en el concepto sustituir al ayudante humano por un asistente robótico, permitiendo al cirujano realizar intervenciones en solitario [1]. Así, los asistentes robóticos desarrollados han sido capaces de dirigir la cámara permitiendo observar la zona anatómica de interés [2] o de manejar herramientas adicionales [3], entre otras tareas.

En general, se puede afirmar que la incorporación de los robots dentro del ámbito quirúrgico ha llevado un aumento de precisión, reproducibilidad y eficiencia de los procedimientos, a la par que se ha conseguido disminuir el esfuerzo requerido por la persona que los controla, permitiendo sustituir intervenciones manuales por procedimientos más automatizados, lo que ha supuesto un enorme beneficio tanto para pacientes como para cirujanos.

Como contrapunto, también han aparecido bastantes dificultades, donde cabe destacar la problemática que ha surgido en las prácticas quirúrgicas en la interpretación de consignas de actuación y en el flujo de información entre el cirujano y el asistente robótico. De igual forma, la incorporación de estas tecnologías ha provocado que los cirujanos requieran de una formación más especializada que les permita adaptarse a las nuevas herramientas robóticas.

Para solventar estos problemas, se requiere del desarrollo de interfaces hombre-máquina (HMI's, del inglés, human-machine interfaces) de comunicación entre el cirujano y el robot, dotadas cada vez de mayor inteligencia artificial y que traten de emular la forma de actuación de un ayudante humano en quirófano. Para esto, dichas interfaces deben ser capaces de decidir durante la intervención con el objetivo de asistir de forma autónoma al cirujano en ciertas tareas complejas. Además, su diseño debe de ser lo más intuitivo posible para que el cirujano, como usuario del equipo, se sienta cómodo y entienda como natural la incorporación de robots en cirugía frente a la metodología tradicional.

## **1.2. Motivación del trabajo de fin de grado**

La línea del grupo de investigación de Robótica Médica del Departamento de Ingeniería de Sistemas y Automática de la Universidad de Málaga ha trabajado durante muchos años en el desarrollo de interfaces persona-máquina, llegando a integrar (entre otras) una interfaz de voz en robots quirúrgicos como el "Endoscopic Robot Machine" (ERM) [4], un asistente robótico desarrollado para cirugía asistida por computador.

Actualmente, está trabajando en un proyecto conjunto con la Universidad de Valladolid y con la Universidad Miguel Hernández de Elche de un sistema robotizado colaborativo para cirugía laparoscópica asistida por la mano (CLAM, del inglés hand-assisted laparoscopy surgery o HALS), una técnica quirúrgica mínimamente invasiva, variante de la cirugía laparoscópica convencional, en la cual el cirujano inserta una mano dentro de la cavidad abdominal del paciente a través de una pequeña incisión que se le realiza al mismo con el objetivo de manipular órganos, mientras que con la otra mano utiliza instrumental laparoscópico adicional.

En este proyecto, se pretende que dicho sistema colabore en las maniobras quirúrgicas y aprenda con la práctica. Para ello, incluye una interfaz persona-máquina fundamentada en un guante quirúrgico inteligente (véase figura 1), que se espera que permita reconocer la fase de



la intervención mediante el uso de los gestos de la mano del cirujano en el interior del paciente, el movimiento que se realiza con el instrumental laparoscópico o las lecturas de las señales fisiológicas combinadas con un modelo paciente-intervención [5].



Figura 1. Guante de realidad virtual sensorizado.

Precisamente, la aportación de esta propuesta es diseñar y validar una interfaz de comunicación basada en la detección de gestos de la mano que realiza el cirujano durante intervenciones HALS. Durante la realización de este trabajo no se disponía del guante quirúrgico en la Universidad de Málaga, al encontrarse en desarrollo por parte del equipo de la Universidad de Valladolid. Para paliar ese problema, se ha decidido utilizar Leap Motion, un dispositivo electrónico que nos permite acceder a las posiciones y orientaciones de los dedos, las palmas de las manos, las muñecas, y otros parámetros físicos de interés. Pese a que los dispositivos no son iguales, los parámetros utilizados para el sistema de detección si serán comunes a ambos, por lo que integrar posteriormente dicha interfaz con el guante sensorizado será mucho más sencillo.

### 1.3. Objetivos

El objetivo general del presente trabajo fin de grado se define como el diseño e implantación de una interfaz persona-máquina basada en la detección de gestos, diseñada para intervenciones HALS.

Para ello, será necesario cumplir con los siguientes objetivos específicos:

1. Diseño e implantación de algoritmos de reconocimiento de gestos basados en los Modelos Ocultos de Markov. Este punto es el que engloba la mayor parte del trabajo, por lo que puede desglosarse en las siguientes fases de trabajo:

1.1. Estudio de las técnicas HALS. Se realiza este estudio para determinar qué gestos va a identificar el sistema, y para seleccionar los parámetros físicos que permitan identificar mejor patrones de movimiento. Para poder realizar una elección adecuada de los gestos a identificar, es necesario recopilar información sobre cuáles de estos pueden ser más prácticos para el cirujano durante la intervención. El objetivo es establecer gestos tan simples e intuitivos como sea posible, para no requerir de un entrenamiento muy complejo por parte del cirujano, y por otro lado, que sean lo suficientemente diferentes entre sí como para que el posterior sistema de reconocimiento pueda distinguirlos con facilidad.

1.2. Diseño y desarrollo de sistema de entrenamiento para el reconocimiento de gestos. Tras un estudio inicial del entorno de desarrollo (Leap Motion, Matlab y ROS) se ha de configurar el dispositivo Leap Motion con ROS para la toma de datos en tiempo real, y desarrollar el código que sea necesario para realizar la grabación de datos de cada gesto en C++.

Posteriormente, se debe pasar una fase de procesado que permita que estos datos sean comprensibles para los Modelos Ocultos de Markov. Dicho procesado consistirá en realizar tareas tales como independizar la grabación de cada muestra del tiempo, para así obtener secuencias de movimientos con un número similar de transiciones entre los estados de la red a entrenar.

A continuación, se configurará el sistema de entrenamiento para establecer mediante matrices de probabilidad, cómo evolucionan los parámetros físicos de cada uno de los gestos.

1.3. Diseño y desarrollo del sistema de reconocimiento. Con el desarrollo de esta fase y una vez entrenado el sistema, se permite detectar los gestos realizados a partir de una grabación. Para ello, se devuelve el gesto que es más probable que se haya producido, comparando la evolución de los parámetros con patrones almacenados en el sistema cognitivo.

2. Integración del sistema de reconocimiento desarrollado en una arquitectura basada en ROS. Con esto, se busca una integración funcional completa de esta herramienta dentro de un sistema más amplio que engloba otros componentes desarrollados dentro de las líneas de investigación del departamento.

3. Validación del sistema.

Se validará todo el sistema desarrollado a través de pruebas experimentales que demuestren tanto su funcionamiento de forma aislada, como su correcta integración dentro del sistema global basado en la arquitectura ROS.

## 1.4. Estructura de la memoria

La memoria del presente trabajo de fin de grado se encuentra dividida en seis capítulos, dos anexos y las referencias bibliográficas. Todos los capítulos salvo este primero y el último se inician con una introducción que expone la problemática a resolver, junto con una breve descripción del contenido, y terminan con unas conclusiones que destacan los aspectos más importantes del capítulo. La organización del contenido de dicho trabajo se describe a continuación:

El **capítulo 2, Técnicas e interfaces en robótica quirúrgica**, realiza una descripción breve de las principales técnicas de cirugía mínimamente invasiva, así como de las interfaces hombre-máquina utilizadas en robótica médica, destacando los problemas que éstas pretenden resolver.

El **capítulo 3, Planteamiento del problema**, describe el entorno quirúrgico utilizado en intervenciones HALS, la problemática que supone la falta de interfaces de comunicación en estos procedimientos, a la par que se propone una interfaz basada en la detección de gestos, junto a la arquitectura utilizada para resolver el problema.

El **capítulo 4, Algoritmos de entrenamiento y reconocimiento**, presenta los algoritmos utilizados para el montaje del sistema de reconocimiento, detallando el funcionamiento de cada uno de los módulos que lo componen: adquisición de datos, procesado de datos, entrenamiento de librería de gestos y reconocimiento de gestos.

El **capítulo 5, Implantación y experimentos**, presenta por un lado el entorno hardware y software en el cuál se ha implantado el presente trabajo, detallando cada uno de sus componentes, así como las pruebas experimentales que se han realizado para validar el funcionamiento del sistema de reconocimiento de gestos desarrollado.

Finalmente, en el **capítulo 6, Conclusiones y trabajos futuros** se tratarán los aspectos más relevantes del trabajo, así como las futuras líneas de desarrollo que puedan derivar de este.



# Capítulo 2: Técnicas e interfaces en robótica quirúrgica

## 2.1. Introducción

En este capítulo se va a realizar una descripción de algunas de las técnicas de CMI, haciendo especial hincapié en la cirugía laparoscópica asistida por la mano, destacando sus principales ventajas e inconvenientes frente a otras técnicas convencionales. Asimismo, se va a comentar cómo los sistemas robóticos han permitido paliar algunos problemas surgidos con la aparición de estas nuevas técnicas, además de algunos ejemplos de interfaces de comunicación elaboradas para controlar diferentes asistentes quirúrgicos robotizados, comentando sus principales características.

## 2.2. Descripción de las técnicas de cirugía mínimamente invasiva

De forma resumida, la CMI se puede entender como toda aquella cirugía basada en reducir al mínimo las incisiones realizadas sobre el paciente que se requerirían en el caso de intervenir mediante una cirugía convencional o abierta. Así, se utilizan estas pequeñas incisiones para introducir los instrumentos que maneja el cirujano, bien con sus manos directamente o mediante un sistema robótico.

Entre las técnicas quirúrgicas basadas en este principio, se puede destacar la cirugía laparoscópica, un procedimiento quirúrgico en el que el cirujano inserta tanto instrumentos especiales (de caña larga), como un endoscopio, a través de incisiones realizadas en la cavidad abdominal del paciente. Sobre dichas incisiones se colocan trocares, unos elementos que facilitan la inserción de las herramientas (a través de puntos de fulcro) y disminuyen las lesiones ejercidas sobre la pared abdominal. Esta técnica proporciona una serie de ventajas frente a la cirugía abierta tradicional (laparotomía), entre las que destacan una disminución del tiempo de recuperación del paciente, una reducción de complicaciones post-operatorias y un mejor resultado estético, puesto que las cicatrices son menores [6].

Además, en los últimos años, se ha producido un importante avance en la CMI, debido en gran parte a la aparición y el desarrollo de nuevas técnicas y herramientas tecnológicas. Una de éstas técnicas es la cirugía de puerto único o SILS [7] (de su acrónimo en inglés Single Incision Laparoscopic Surgery), que está basada en la introducción tanto de los instrumentos como del endoscopio a través de una única incisión en el abdomen del paciente, mediante un trocar multipuerto. Otra técnica a destacar es la cirugía endoscópica a través de orificios naturales o NOTES (de su acrónimo en inglés Natural Orifice Transluminal Endoscopic Surgery), que pretende eliminar por completo las incisiones en la pared abdominal, utilizando

únicamente accesos naturales y la perforación de órganos internos. Tanto SILS como NOTES pretenden minimizar el número de puertos de entrada presentes en los procedimientos laparoscópicos, sin embargo, pese a los beneficios que conlleva sobre los pacientes el disminuir el número de incisiones realizadas, surgen grandes desventajas.

En el caso de SILS, surgen dos inconvenientes: la limitación del rango de movimiento de los instrumentos debido a la inserción de todas las herramientas por el mismo puerto, y por otro lado, la pérdida de triangulación producida por la cercanía de las herramientas y el endoscopio, lo que se traduce en una pérdida de sensación de profundidad en la imagen que observa el cirujano. En el caso de NOTES, las principales limitaciones vienen determinadas por el tamaño del puerto de entrada, por lo que al igual que en SILS, se complica la introducción y el manejo del instrumental quirúrgico. Ante esta problemática, se han planteado soluciones tales como incorporar mini-robots en el interior del paciente, entre las que destaca el uso de robots camarógrafos miniaturizados [8], que permitan paliar parcialmente estos problemas.

Otra de estas técnicas es la “hand-assisted laparoscopic surgery” o HALS [9]-[10], una variante de la cirugía laparoscópica tradicional, en la cual el cirujano inserta una mano dentro de la cavidad abdominal del paciente a través de una pequeña incisión que se le realiza al mismo, lo que le permite manipular órganos, mientras que con la otra mano utiliza un instrumental laparoscópico mínimamente invasivo.

Actualmente, la cirugía HALS se aplica en intervenciones en las que, debido a la naturaleza de la operación, no se sabe a priori si se va a tener que realizar una laparotomía una vez empezada la operación con cirugía laparoscópica, y es de especial utilidad en operaciones de esplenectomía [11] (extirpación de bazo) y en colectomías (extirpaciones de colon), aunque tiene muchas otras aplicaciones.

Esta técnica proporciona una serie de ventajas con respecto a la cirugía laparoscópica, entre las que destaca la posibilidad de recuperar el sentido del tacto, lo que permite separar estructuras anatómicas de forma suave y poco traumática, dotando al cirujano de un mayor control sobre la operación. Además, se ha demostrado que el tiempo de recuperación del paciente no resulta superior que en otras metodologías utilizadas (cirugía laparoscópica tradicional y cirugía abierta) [12]-[13].

Desde el punto de vista del paciente, algunos trabajos han demostrado que el uso de técnicas HALS tiene beneficios similares a los obtenidos con procedimientos menos invasivos, como los comentados anteriormente para cirugía laparoscópica [14]. Así, se ha comprobado que, en cierto tipo de intervenciones, como en la extirpación de cáncer colorrectal [15]-[16], los pacientes presentaron una menor pérdida de sangre, menor tiempo de operación, y una estancia más corta en unidades de cuidados intensivos que en el caso de intervenir mediante laparotomías.

El principal inconveniente de este tipo de intervención es que el cirujano solo puede manejar una herramienta laparoscópica adicional, por lo que éste requiere de una profunda colaboración con su ayudante con el fin de coordinar sus movimientos para realizar una maniobra quirúrgica. Con el objetivo de paliar estas dificultades, se plantea la sustitución del asistente humano por nuevos sistemas robotizados (diseñados específicamente para HALS) que permitan trabajar de forma colaborativa con el cirujano y faciliten su tarea mediante el uso de interfaces “persona-máquina”, permitiéndole intervenir en solitario.

## 2.3. Interfaces quirúrgicas

Las interfaces persona-máquina (human-machine interface, HMI) utilizadas en robótica quirúrgica son aquellas que establecen una comunicación directa entre el cirujano y un asistente robótico en ambos sentidos; por un lado, permiten al cirujano comandar órdenes directas al robot utilizado o indicar mediante las maniobras realizadas con el instrumental la fase actual de la intervención. Por otro lado, el asistente robótico es capaz de proporcionar información acerca de parámetros tales como la fuerza ejercida sobre la pared abdominal del paciente, gracias al uso de sensores y otros dispositivos.

Como respuesta a la información enviada por el cirujano, los robots quirúrgicos asisten en ciertas labores realizadas durante la cirugía, facilitando y automatizando gran cantidad de procedimientos. En función del grado de autonomía de dichos asistentes robóticos (es decir, del tipo de control que el cirujano ejerce sobre los mismos) se pueden clasificar en los siguientes grupos:

- **Asistentes con movimientos pre-programados.** En estos asistentes, el cirujano se encarga de determinar el comportamiento de las herramientas que maneja el robot de forma interactiva, gracias al uso de técnicas de diseño asistido por ordenador y fabricación asistida por ordenador (CAD/CAM), que dan lugar a la cirugía integrada por computador (CIS del inglés Computer Integrated Surgery). La filosofía de las técnicas CIS está basada en el uso de sistemas de ayuda al cirujano en el preoperatorio, durante la intervención y en el postoperatorio.

En la fase de preoperatorio, o fase CAD, se obtienen los modelos anatómicos tanto en dos como en tres dimensiones y los datos concretos del paciente, para diseñar un plan operatorio específico para este utilizando navegadores quirúrgicos, que permitan determinar el programa que seguirá el robot durante la intervención, y que se lleva a cabo mediante movimientos pre-programados, como por ejemplo, las trayectorias que debe de seguir un torno en una operación de prótesis de cadera.

Posteriormente, en una fase realizada durante la propia intervención, se lleva a cabo una correspondencia entre las coordenadas virtuales de las tareas y las reales operatorias,

permitiendo que el robot lleve a cabo las tareas quirúrgicas. Además mediante un proceso de registro y la supervisión del cirujano es posible actualizar el plan operatorio en tiempo real. Por otro lado, si el cirujano considera necesario abortar el plan, también puede detener el sistema robótico y continuar manualmente el procedimiento.

Por último, en el postoperatorio se sigue la recuperación del paciente mediante modelos anatómicos del mismo. Entre estos asistentes destacan los diseñados para cirugía ortopédica, como el ROBOCOC, diseñado para la sustitución total de cadera [17], o para reparación ósea, como el CASPAR, que se utiliza principalmente para operaciones de rodillas [18].

- **Asistentes con control teleoperado:** Estos asistentes se basan en el concepto de teleoperación, que puede definirse como el conjunto de tecnologías que comprenden la operación o gobierno a distancia de un dispositivo por parte de un ser humano.

Para el control de dichos asistentes, el cirujano define los movimientos deseados a través de un dispositivo maestro y el robot sigue las referencias que se especifican a través de éste. Como ejemplo de sistema de telecirugía, destaca el da Vinci, sistema quirúrgico utilizado en procedimientos de cirugía mínimamente invasiva, que dispone de tres o cuatro brazos manipuladores, donde uno de ellos maneja la cámara laparoscópica, y los otros se encargan de manejar el resto de instrumental [19]. También dispone de una estación de teleoperación y una unidad de control, además de estar dotado de herramientas con una muñeca con tres ejes capaz de imitar la libertad de movimientos de la muñeca humana.

Otros sistemas teleoperados que merece la pena destacar son el Neurobot, un micromanipulador que permite realizar intervenciones mínimamente invasivas en neurocirugía y que ha permitido reducir el tiempo de las intervenciones quirúrgicas en la base del cráneo, proporcionando resultados satisfactorios [20], o el RAVEN, un prototipo de robot médico teleoperado portable cuyo objetivo es realizar el control remoto de sistemas de cirugía robótica en el espacio o en otros lugares donde se encuentre el paciente [21].

- **Asistentes con control directo o guiado (hands-on):** En estos otros asistentes, el cirujano maneja la herramienta quirúrgica sujeta por el robot, de forma que tanto el cirujano como el asistente robotizado colaboran de forma directa en el desarrollo de una tarea, en la que el computador realiza el papel de mediador entre ambos. Una ventaja de este tipo de control es que el sistema robótico disminuye el temblor de la mano del cirujano, proporcionando un movimiento del instrumental más firme y preciso. Además, ésta técnica permite establecer límites en el área de trabajo, de forma que se impida el acceso a zonas críticas aunque el cirujano trate de acceder a ellas, para así garantizar una mayor seguridad en la operación.



Como ejemplo de este tipo de asistentes cabe destacar aquellos cuya funcionalidad está enfocada hacia la cirugía ortopédica, como el ACROBOT [22], desarrollado para el reemplazo total de rodilla. Dicho asistente permite al cirujano trabajar en una región segura sin entrar en contacto con tejidos blandos y otras estructuras como nervios o ligamentos, gracias al uso de las denominadas “virtual mixures”, unas protecciones virtuales que restringen el acceso por error a determinadas zonas.

Por otro lado, también se han desarrollado sistemas robotizados basados en la forma de manipulación “hand on” para intervenciones tales como la cirugía del oído o cirugía ocular, como el “Steady-hand robot system” de la Universidad Johns Hopkins, para realizar intervenciones de microcirugía de retina [23].

- **Asistentes semiautónomos (colaborativos).** Estos asistentes se caracterizan por disponer de la capacidad de emular al asistente humano que colaboraría con el cirujano durante una intervención, permitiendo realizar ciertas actividades secundarias de forma automática o acorde a los comandos que recibe del cirujano. Como ejemplo, permiten realizar tareas en operaciones de CMI tales como el guiado preciso de la cámara laparoscópica a la región anatómica de interés o el manejo de herramientas quirúrgicas según las órdenes del cirujano. Para ello, se requiere de interfaces persona-máquina que permitan establecer una comunicación directa entre el cirujano y los sistemas robotizados.

En esta línea, el AESOP fue el primer sistema robótico aprobado por la FDA en 1994 para intervenciones quirúrgicas abdominales. Dicho sistema consistía en un brazo robótico que sujetaba una cámara laparoscópica que podía ser controlada mediante comandos de voz [24].

Otros asistentes laparoscópicos son capaces de mover la cámara bien mediante el uso de botoneras acopladas en la palma de la mano del cirujano, como en el caso del robot LAPMAN, de Medsys [25], o mediante interfaces de comunicación basadas en la interpretación de los movimientos de la cabeza del cirujano, gracias a la ayuda de un giróscopo que va acoplado en su cabeza, como ocurre en el robot ENDOASSIST de Armstrong Healthcare [26]. Otros desarrollos se han basado en los movimientos faciales del cirujano para dirigir con precisión el laparoscopio a la región anatómica deseada, gracias a algoritmos basados en imagen [27]-[28], e incluso mediante los gestos específicos de las herramientas laparoscópicas [29].

También hay sistemas robotizados que permiten utilizar interfaces que incorporan más de un medio de comunicación. Este es el caso del asistente ERM [30], un sistema con una interfaz que dispone de dos modos de funcionamiento para dirigir la cámara laparoscópica: uno mediante la voz, comandando órdenes verbales, y otro mediante el

movimiento de un joystick. Dicho asistente ha dado buenos resultados en procedimientos laparoscópicos como la colecistectomía [31].

En general, las HMIs basadas en el envío de órdenes directas añaden tareas nuevas y antinaturales al cirujano que pueden alargar el tiempo de la intervención. Para solucionar este problema, dichas interfaces deben de parecerse lo más posible a la interacción mantenida con el asistente humano durante todo el procedimiento.

Por ello, en la actualidad las HMIs están evolucionando hacia un nuevo modelo que incluya un segundo método de comunicación, basado en el reconocimiento de tareas quirúrgicas, tal y como se aplica en trabajos como [32], donde el robot realiza una actuación dependiendo de la última tarea realizada por el cirujano de forma autónoma. De esta forma, aumenta la capacidad de reacción e inteligencia del asistente robótico, ya que es capaz de interpretar el estado de la operación y puede anticiparse a las necesidades del cirujano, mediante estrategias de acción definidas previamente para cada estado del protocolo quirúrgico.

Otro ejemplo de reconocimiento de las tareas quirúrgicas realizadas puede producirse a partir del seguimiento de los ojos del cirujano durante la operación, ya que la atención visual de éste cambia en función de las diferentes etapas del procedimiento. Así, se ha propuesto el uso de Parallel Layer Perceptor (PLP) [33], para automatizar el reconocimiento de las fases de un protocolo quirúrgico a partir de la mirada del cirujano y el análisis de las imágenes del endoscopio, donde las transiciones de los instrumentos se utilizan para detectar cambios en las fases del protocolo quirúrgico.

Para la identificación de las tareas quirúrgicas, se dispone de diversas metodologías basadas en la detección de patrones, entre las que destacan las técnicas de Dynamic Time Warping (DTW), Redes Neuronales (NN, del inglés Neural Networks), Algoritmos Genéticos o los Modelos Ocultos de Markov (HMM). Hay estudios recientes en los que se han utilizado las técnicas de DTW para identificar tareas quirúrgicas a partir de imágenes de vídeo en cirugías de cataratas [34]. Otros grupos se han basado en las redes neuronales para reconocer tareas en maniobras multitarea a partir de videos laparoscópicos [35]. En otros casos, se han empleado otras herramientas matemáticas para el reconocimiento de los movimientos quirúrgicos elementales, como por ejemplo un clasificador probabilístico de Bayes [36].

De igual forma, los HMM se han utilizado para el entrenamiento y reconocimiento de tareas quirúrgicas, tal como se puede ver en diversos estudios [37], y es precisamente esta metodología la que se ha elegido para el desarrollo del presente trabajo.

## 2.5. Conclusiones

En este capítulo se ha comenzado realizando una descripción de la cirugía mínimamente invasiva, comentando de forma resumida en qué consiste y cuáles son las principales técnicas basadas en este principio que se usan en la actualidad, destacando la cirugía laparoscópica asistida por la mano.

De igual forma, se han comparado estas técnicas con los procedimientos realizados mediante cirugía en abierto, haciendo alusión a diversos artículos con datos experimentales, y se han destacado las ventajas e inconvenientes de las técnicas HALS.

Posteriormente, se ha comentado en qué consisten las interfaces de comunicación persona-máquina utilizadas para controlar diversos asistentes quirúrgicos. Tras esto, se ha realizado una clasificación de los asistentes robotizados en función de su grado de autonomía, describiendo algunos ejemplos y las principales formas de comunicación que se han utilizado: bien mediante órdenes directas comandadas por parte del cirujano o mediante el reconocimiento de tareas quirúrgicas.

Hay que destacar que la mayoría de los avances que se han comentado en las interfaces de comunicación han sido desarrollados para las técnicas tradicionales de cirugía laparoscópica, y poco hay aún escrito sobre avances similares en el ámbito de las técnicas HALS, debido a su reciente aparición en el ámbito quirúrgico, por lo que se hacen necesarias otras soluciones. Por ello, este trabajo pretende aportar la integración de una metodología de comunicación que cubra la necesidad de asistir al cirujano durante estos procedimientos quirúrgicos a partir de la evolución de los movimientos de su mano no diestra, ahorrando tiempo de operación y paradas innecesarias que puedan dificultar el proceso.



# Capítulo 3: Planteamiento del problema

## 3.1. Introducción

Como se ha descrito en capítulos anteriores, el objetivo del presente trabajo fin de grado consiste en el desarrollo de una interfaz persona-máquina para cirugía laparoscópica asistida por la mano, que sea capaz de detectar los gestos del cirujano durante la operación, entendiendo por gesto aquel patrón de movimiento que puede dividirse en acciones básicas o estados y que se puede identificar como una secuencia determinada de estos. Para comprender mejor la problemática a resolver, se van a detallar en este capítulo las principales características y componentes del entorno quirúrgico utilizado en intervenciones HALS, así como los problemas que surgen debido a la falta de interfaces de comunicación en estos procedimientos. Posteriormente, se propone una interfaz basada en la detección de gestos, detallando las hipótesis y simplificaciones utilizadas en la misma. Finalmente, se presenta la arquitectura utilizada para resolver el problema.

## 3.2. Escenario quirúrgico en HALS

En la figura 3.1 se muestra un esquema del escenario quirúrgico HALS utilizado en aquellas intervenciones que disponen de un asistente robotizado:

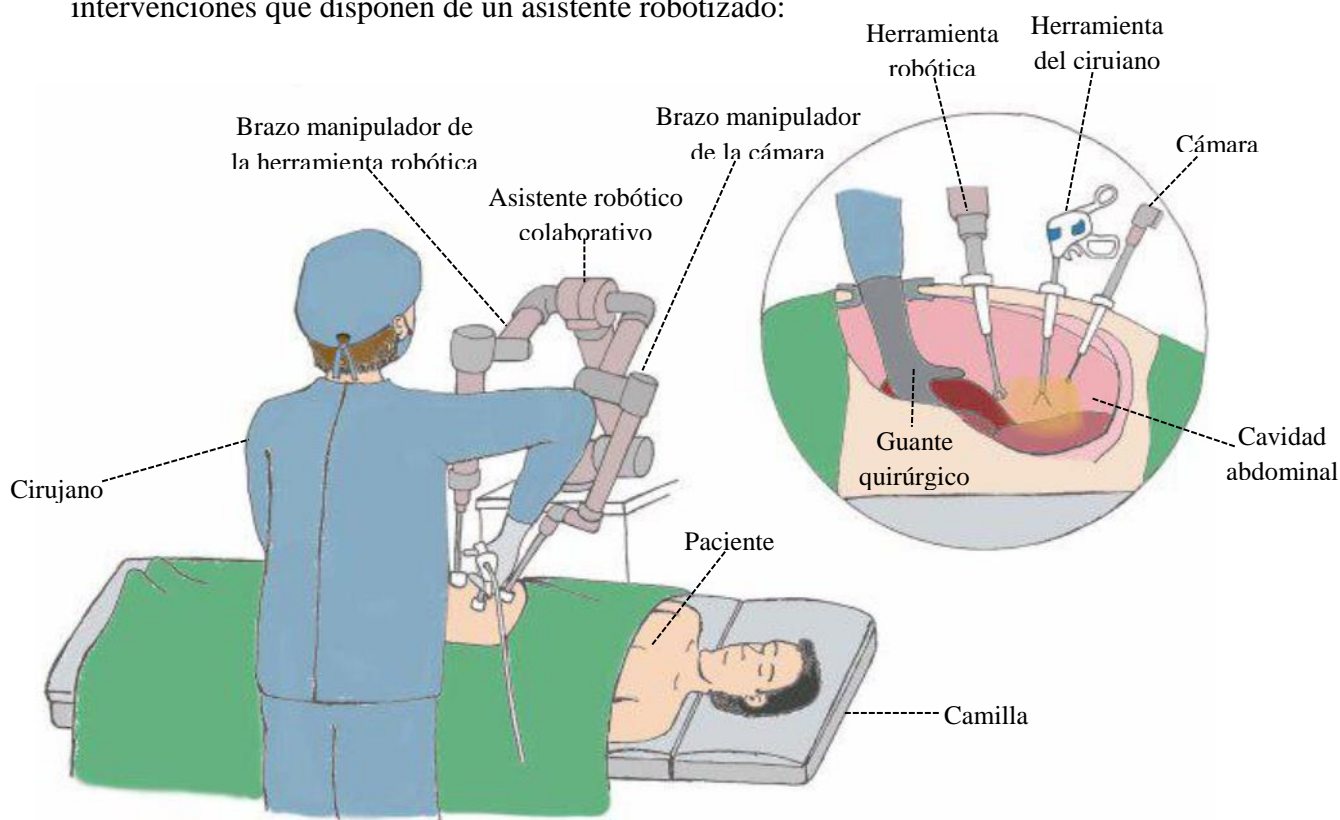


Figura 3.1. Escenario quirúrgico utilizado en intervenciones HALS.

En las técnicas convencionales de cirugía laparoscópica asistida por la mano, el cirujano interactúa con el cuerpo del paciente en un espacio restringido dentro de la cavidad abdominal. Por un lado, la mano no diestra del cirujano se encuentra insertada dentro de la cavidad abdominal a través de una incisión especial realizada sobre paciente (tal y como se muestra en la figura 3.1), con el fin de manipular directamente las estructuras anatómicas de interés. Mientras tanto, la otra mano permanece fuera del paciente, y tiene la función de manejar un instrumento quirúrgico, que en la figura se ha denotado como “Herramienta del cirujano”.

Sin embargo, para la operación se requiere del manejo de herramientas adicionales con el fin de cumplir ciertas tareas quirúrgicas. Por ello, se ha planteado la incorporación de un asistente robótico colaborativo que se encargue del soporte de una cámara y una herramienta robótica, insertadas también en el interior del abdomen del paciente. La cámara tiene la función de proporcionar al cirujano una visión detallada de las estructuras y tejidos internos, mientras que la herramienta robótica tiene el objetivo de permitir la realización de tareas como pueden ser la sutura o la sujeción de órganos.

Con la adición de este asistente, se proporciona al cirujano un mayor control sobre la operación, ya que le permite incorporar instrumental que no podría manejar manualmente por tener su mano no diestra en el interior del paciente durante toda la intervención.

Llegados a este punto, es posible plantearse dos opciones. La primera es que se disponga de un asistente humano encargado de dirigir el sistema robótico utilizado durante la intervención, lo que requeriría de una profunda colaboración en muchas de las tareas quirúrgicas entre cirujano y asistente. La segunda es que sea el propio cirujano en solitario el que pueda coordinarse con dicho sistema. Esta última opción simplifica mucho el proceso, aunque como los asistentes robóticos deben simular a un asistente humano, deben de resolverse dos problemas fundamentales: el desarrollo de una interfaz de comunicación natural que permita la transmisión de información entre cirujano y asistente, y la toma de decisiones autónoma. Para solucionar estos problemas, se requiere de la combinación de tres características:

- **El modelado del protocolo quirúrgico.** Con el desarrollo de este modelo, se consigue establecer una secuencia de comandos a seguir por el robot durante la intervención de CMI. Por ejemplo, Muradore et al propuso la creación de un conjunto de tareas elementales que se pueden combinar para construir el procedimiento quirúrgico completo [38]. También hay estudios de investigación que modelan el protocolo quirúrgico con estados que dependen de la herramienta laparoscópica insertada en la cavidad abdominal [39].
- **La navegación automática de la herramienta robótica en el escenario quirúrgico.** Hay ciertas tareas que ya se han cubierto en la literatura [40], y una de ellas es el control

visual, que consiste en el guiado automático de la cámara laparoscópica. Esta técnica se ha usado para realizar movimientos seguros del endoscopio en cirugía cardíaca [41], o para predecir la localización del efector final [42]-[43]. Otros trabajos han utilizado el reconocimiento de tareas quirúrgicas que inicialmente estaban centradas en la evaluación de las habilidades del cirujano [44] para transferir estas habilidades a un robot asistente. Además, algunas de las principales tareas quirúrgicas han sido completamente automatizadas, tales como el punzado de sutura automático [45] o la atadura de nudos [46] en procedimientos de sutura, así como el agarre y levantamiento en la retracción de tejidos [47].

- **Un sistema de comunicación capaz de reconocer las tareas del cirujano y/o dirigir órdenes.** Dicho sistema requiere del uso de una interfaz capaz de comandar órdenes directas y/o reconocer tareas quirúrgicas mediante una interfaz persona-máquina. Los trabajos iniciales se han centrado en que el cirujano comande órdenes al robot usando diferentes medios de comunicación: dispositivos joystick, movimientos con la cabeza, gestos específicos de las herramientas laparoscópicas o comandos de voz, tal y como se comentó en el apartado 2.2. Por otro lado, el reconocimiento de tareas quirúrgicas se ha aplicado a trabajos donde el robot actúa dependiendo de la última tarea reconocida del cirujano.

Este documento se centra en el diseño y desarrollo de la interfaz de comunicación, basándose en la detección de gestos de la mano no diestra del cirujano, y utilizando dicha detección como medio para comandar órdenes que permitan una asistencia colaborativa activa por medio de la participación de las herramientas laparoscópicas que incorpora el robot.

### 3.3. Arquitectura del sistema

En el contexto de este trabajo fin de grado, el robot colaborativo encargado de reemplazar al asistente humano requiere de las siguientes características:

- La monitorización del estado actual de la intervención a partir del protocolo quirúrgico.
- La navegación segura de la herramienta laparoscópica y el control de fuerzas con restricciones de movimiento.
- Una interfaz de comunicación que permita al sistema robótico generar la referencia de actuación a partir del reconocimiento de los gestos del cirujano.

Para resolver estos problemas, se propone la arquitectura del planificador colaborativo que se puede observar en la figura 3.2.

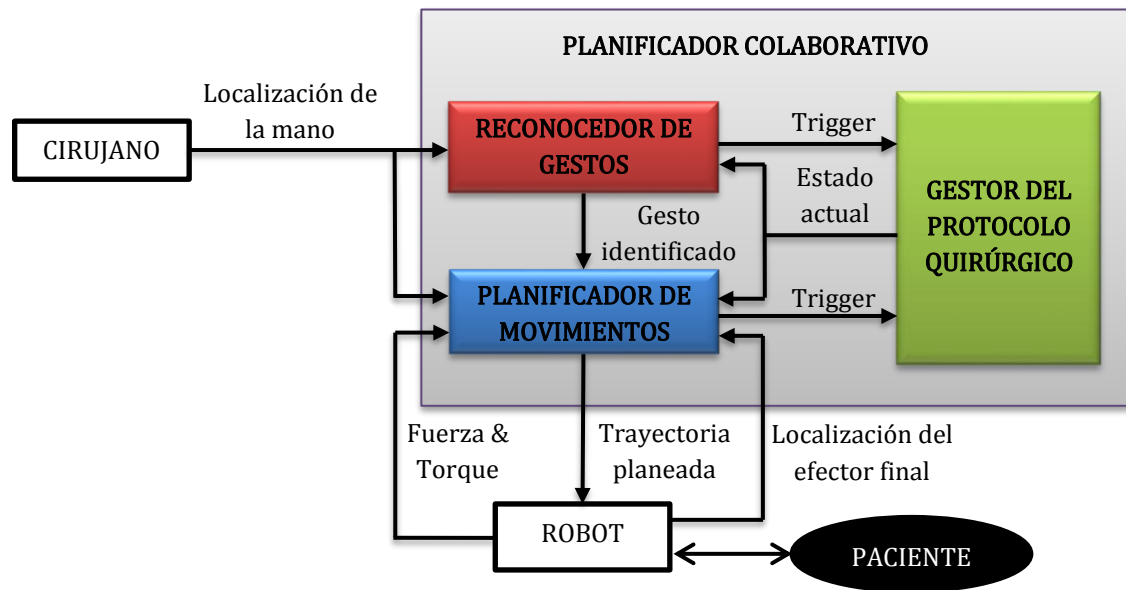


Figura 3.2. Arquitectura propuesta para la asistencia al cirujano en técnicas HALS.

En esta arquitectura se encuentra como entrada inicial la *localización de la mano* del cirujano, a partir de la cual se permite identificar los movimientos que ésta realiza. Para ello, es necesario disponer de algún elemento que proporcione en cada momento tanto la posición de los dedos como su orientación. Este elemento puede ser un guante quirúrgico sensorizado.

Por otro lado, el *reconocedor de gestos* interactúa con la localización de la mano del cirujano para determinar si se ha producido un gesto de entre un conjunto de gestos entrenados previamente. Estos se encuentran asociados a la realización de una tarea quirúrgica sobre el paciente, como puede ser servir de sostén de tejidos, o estirar el hilo de sutura. El conjunto de tareas quirúrgicas posibles se encuentran incluidas dentro del protocolo quirúrgico.

Tras esto, se observa un bloque *planificador de movimientos*, encargado de determinar una trayectoria adecuada para comandar al robot tras recibir como entrada la identificación de un determinado gesto y el estado actual de la intervención, provenientes del reconocedor de gestos y del gestor de protocolo quirúrgico respectivamente. Para ello, recibe otros parámetros tales como las localizaciones del efector final del robot y de la mano del cirujano, con el objetivo de evitar colisiones, así como la fuerza y el torque que producen las herramientas laparoscópicas sobre el paciente, para llegar a la posición deseada minimizando las fuerzas ejercidas sobre el individuo. La trayectoria a seguir se calcula por medio de un controlador híbrido de fuerza y posición.



Otro bloque es el *gestor del protocolo quirúrgico*, que se encarga de determinar el estado actual de la intervención a partir de un diagrama de estados y proporciona una actuación u otra sobre el planificador de movimientos en base al estado actual. Dicho diagrama cambia su estado actual si recibe una señal de activación (*Trigger*) desde el reconocedor de gestos o desde el planificador de movimientos.

Más abajo en la figura encontramos el *robot*, que es el encargado de realizar las actuaciones pertinentes sobre el paciente, ejecutando tareas tales como el guiado de la cámara laparoscópica o el manejo de las herramientas robóticas a partir de la trayectoria diseñada por el bloque planificador de movimientos.

Finalmente se encuentra el *paciente*, que proporciona indirectamente información al robot acerca de los límites de operación de las herramientas quirúrgicas, que vienen determinados por la dimensión de su cavidad abdominal.

### **3.4. Conclusiones**

Por lo general, los asistentes robóticos comerciales están centrados en una interface de teleoperación, por lo tanto, son incapaces de hacer frente a técnicas HALS debido al requisito de contacto directo entre el cirujano y el paciente. Precisamente, en este capítulo se ha presentado un entorno quirúrgico en HALS junto a una arquitectura simplificada que permite la asistencia robótica en este escenario. Durante la descripción de dicha arquitectura, se han identificado tres cuestiones mínimas a cubrir con el objetivo de utilizar sistemas robóticos con esta técnica; primero, se requiere de un gestor del protocolo quirúrgico que controle el estado del procedimiento quirúrgico y permita al robot conocer el estado de la intervención. Segundo, es necesario un planificador de movimientos flexible que permita proporcionar trayectorias auto-guiadas al robot evitando obstáculos y minimizando la fuerza ejercida sobre el paciente. Por último, se requiere de una interfaz de comunicación cirujano-robot que implemente tanto órdenes directas a partir del cirujano como reconocimiento de gestos en tiempo real. Así, este trabajo se centra en el desarrollo de una interfaz basada en la detección de gestos, con lo que se resuelve parte de la problemática.



# Capítulo 4: Algoritmos de entrenamiento y reconocimiento

## 4.1. Introducción

Al igual que ocurre con el resto de metodologías quirúrgicas, el desarrollo de una interfaz de usuario de una plataforma robótica diseñada para asistir en técnicas HALS debe dotar al cirujano de un medio de comunicación eficiente y confortable, que permita simplificar el procedimiento de las operaciones.

Por ello, el objetivo es implementar una interfaz que permita una comunicación cirujano-robot durante las intervenciones HALS que disponen de un asistente quirúrgico semiautónomo. Dicha interfaz formará parte de una plataforma robótica colaborativa, en la cual es necesario el reconocimiento de maniobras para poder obtener una consigna de actuación por parte del asistente robotizado, que dependerá del protocolo quirúrgico. Para esto se plantea un sistema de reconocimiento de gestos de la mano no diestra del cirujano que se encuentra introducida en el interior del paciente durante estas intervenciones, lo que permite combatir el principal inconveniente de las técnicas HALS: disponer de sólo una mano para manejar el instrumental laparoscópico.

Para conseguir dicho objetivo, este capítulo propone la metodología llevada a cabo para completar el sistema de reconocimiento de gestos, incluyendo los procesos y módulos necesarios para obtener una correcta detección de los patrones de movimiento producidos durante una intervención quirúrgica. En primer lugar, se detalla la fase de procesamiento de datos, en la cual se han tratado los datos obtenidos a partir de los movimientos de la mano del cirujano, y se definen los parámetros para entrenar cada uno de los gestos. Posteriormente, se detalla la fase de entrenamiento, y finalmente, se presenta el sistema de reconocimiento propuesto.

## 4.2. Esquema general del sistema desarrollado

Como resumen del funcionamiento del sistema, se puede comentar que éste se divide en dos fases fundamentales. En una primera *Fase de Entrenamiento*, se pretende crear una librería con algunos gestos realizados por el cirujano, a partir del registro y procesado de datos de la localización de la mano del mismo, y siguiendo un proceso de entrenamiento de dichas librerías. Por otro lado, mediante una *Fase de Reconocimiento* se pretende determinar si se ha realizado algún gesto durante la detección de la localización de la mano no diestra del cirujano. Para ello se comparan secuencias de datos (obtenidas a partir del procesado de los

datos registrados durante la evolución cinemática de la mano) con los modelos de los gestos almacenados en las librerías creadas previamente en la *Fase de Entrenamiento*.

Para el reconocimiento de gestos, se debe elegir una representación matemática adecuada que permita comparar los movimientos de la mano que realiza el cirujano durante la intervención con el conjunto de patrones de movimiento almacenados y entrenados previamente. Para esto, se han elegido los Modelos Ocultos de Markov (HMM), cuyo funcionamiento se detalla en el Anexo B.

En la figura 4.1 se muestran los procesos y módulos necesarios para obtener una correcta detección de los gestos realizados durante una intervención:

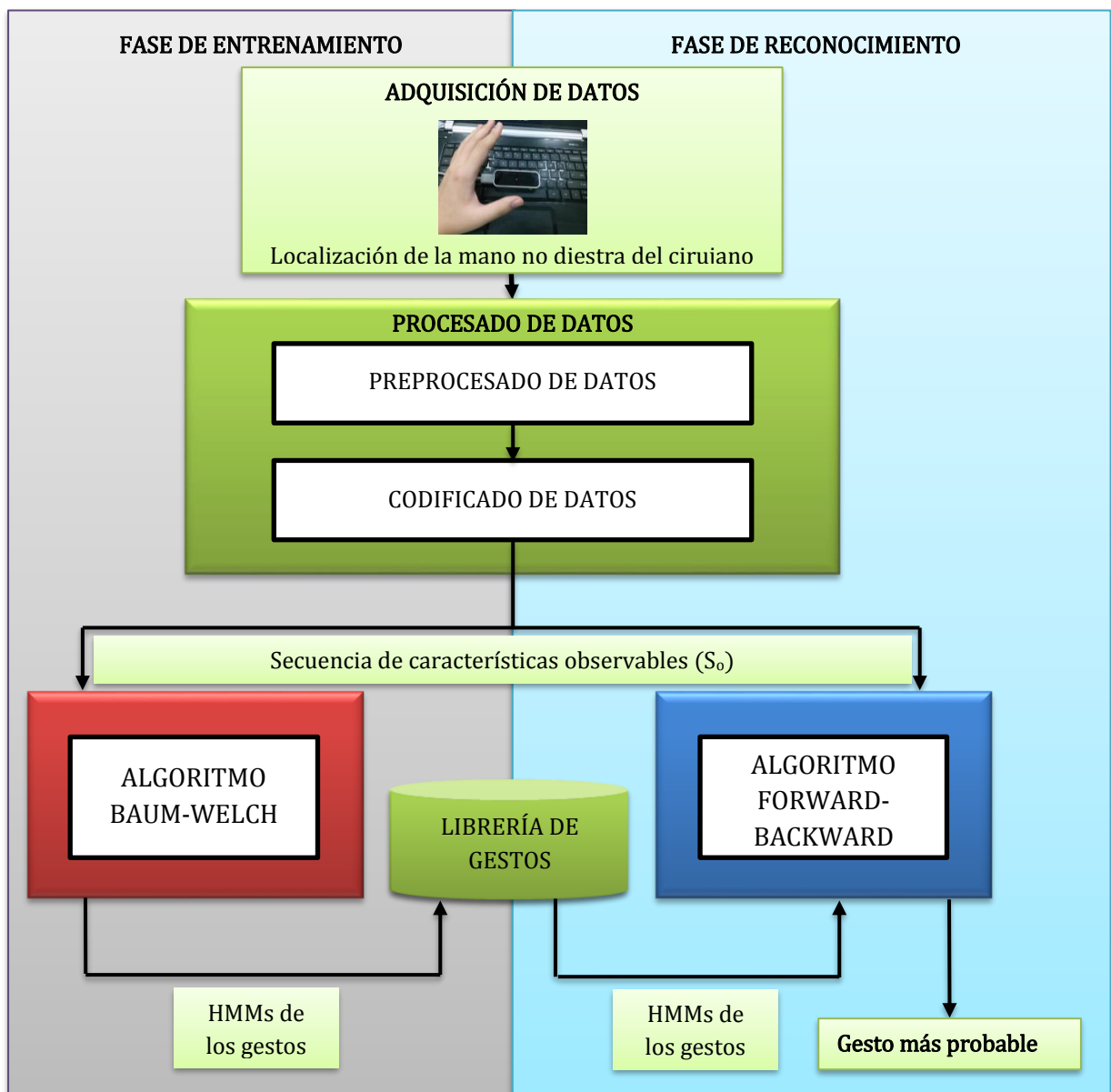


Figura 4.1. Esquema general de los módulos necesarios para la detección de gestos.

Como se ha comentado, el sistema se puede dividir en dos grandes fases, que pueden verse claramente diferenciadas en la figura 4.1. En la parte izquierda de la figura encontramos la primera fase, *Fase de Entrenamiento*, donde se observan los módulos que participan en el procesado e implementación del sistema de entrenamiento de gestos. Por otro lado, en la parte derecha de la imagen se observa la *Fase de Reconocimiento*, donde se muestran los módulos que participan en el procesado e implementación del sistema de reconocimiento. En la zona media de la figura se exponen los módulos comunes a ambas fases: el módulo de *Adquisición de Datos*, el de *Procesado de Datos* y la *Librería de Gestos*. A continuación, se procede a definir cada uno de los módulos que componen la figura 4.1 para comprender el funcionamiento del sistema.

En primer lugar, el módulo de *Adquisición de Datos*, marcado en verde claro en la figura, tiene como objetivo almacenar los valores de localización de los puntos anatómicos de la mano que sean de interés a partir de un sensor de localización de la mano.

La información obtenida mediante el bloque de *Adquisición de Datos* sirve como entrada al siguiente módulo, denominado *Procesado de datos*, marcado en verde oscuro en la figura, cuya finalidad es preparar los datos extraídos previamente para que sean comprensibles para los HMM. Para ello, se lleva a cabo un proceso de codificado de varios *parámetros físicos* ( $P_i$ ), en secuencias ( $S_o$ ) de *características observables* ( $o_i$ ). En este punto, conviene aclarar la nomenclatura utilizada para que el lector pueda seguir correctamente el desarrollo del presente capítulo:

- Se denomina *parámetro físico* a aquella variable que permite relacionar dos sistemas de referencia situados en la mano mediante una magnitud física, como puede ser la distancia o el ángulo entre dos dedos.
- Se ha llamado *característica observable* al parámetro que caracteriza la interacción entre varios sistemas de referencia de la mano y que contiene información extraída de la localización relativa de dichos sistemas de referencia. Estas se obtienen a partir del codificado de varios parámetros físicos ( $P_i$ ) en un único valor discreto o etiqueta.

Inicialmente, el módulo *Preprocesado de Datos* recibe las localizaciones seleccionadas de los dedos de la mano, calcula los *parámetros físicos* a partir de la evolución cinemática que experimenta la mano, y permite su ajuste mediante el recorte de muestras erróneas al inicio y fin de cada grabación y otros procedimientos tales como la sincronización temporal de muestras. Posteriormente, el módulo de *Codificado de Datos* se encarga de clasificar los valores de los *parámetros físicos* en etiquetas discretas, y dichas etiquetas permiten construir *vectores característicos* ( $VC$ ), unas estructuras de datos que almacenan la clasificación de los *parámetros físicos* de cada gesto en un instante de muestreo. Tras esto, se transforman dichos vectores en *características observables*, representadas mediante etiquetas simples que

compactan la información relevante, y se construye una secuencia de estas características ( $S_o$ ) con el objetivo de modelar cada registro de un gesto.

A continuación del bloque *Procesado de Datos* se encuentra una bifurcación que muestra los dos destinos a los que se pueden enviar las secuencias  $S_o$  en función de la fase del sistema. En el caso de encontrarse en la *Fase de Entrenamiento*, las secuencias de características observables se envían al *Algoritmo de Baum-Welch*, el cual permite ajustar los *HMM* que sirven para modelar cada uno de los gestos. Estos modelos vienen definidos por un par de matrices: la matriz A de probabilidades de salto entre estados y la matriz B de distribución de probabilidad de características observables en cada estado. Estas matrices se envían al módulo denominado *Librería de Gestos*, donde se almacenan.

En el caso de encontrarse en la *Fase de Reconocimiento*, las secuencias de características observables se envían al algoritmo de *Forward-Backward*, con el objetivo de interpretar si se ha realizado un gesto de los presentes en la *Librería de Gestos* previamente entrenada. Para ello, además de las secuencias que se reciben desde el módulo de *Procesado de Datos*, recibe datos en forma de HMM desde la *Librería de Gestos*. Dicho algoritmo se encarga de operar dichas secuencias con las matrices de probabilidad de los HMM que modelan los gestos almacenados, lo que permite determinar qué gesto es más probable que se haya producido, es decir, a qué patrón se acerca más la secuencia de datos que recibe como entrada el algoritmo de reconocimiento.

A continuación, se detallarán los algoritmos empleados en cada uno de los módulos expuestos en la figura 4.1.

### 4.3. Adquisición de datos

Los datos que se recogen del cirujano son los movimientos que éste realiza con su mano no diestra, simulando el movimiento que realizaría en una operación en el interior del abdomen del paciente.

En general, el sensor de localización de la mano otorga información en forma de matrices homogéneas que definen la posición y orientación de articulaciones, falanges, palma de la mano, muñeca y otras regiones de interés. Dichas localizaciones quedan definidas mediante transformaciones entre sistemas de referencia  $T_L^D$ , expresando siempre las localizaciones respecto a un sistema de referencia global {L}, y siendo {D} la localización destino, que puede ser, por ejemplo, la punta del dedo índice. Dichas matrices homogéneas están compuestas por una rotación y una translación, tal y como se puede ver en la ecuación 4.1:

$$T_L^D = \begin{bmatrix} R_L^D & P_L^D \\ 0 & 1 \end{bmatrix} \quad (4.1)$$

Donde  $P_L^D$  es un vector de translación en coordenadas cartesianas y  $R_L^D$  es una matriz de rotación.

Precisamente, a partir del estudio de la evolución de estas transformadas, se puede conocer la trayectoria que se realiza durante un gesto, considerando especialmente los cambios que se producen entre pares de sistemas de coordenadas expresados respecto al sistema  $\{L\}$ . Así, los *parámetros físicos* que caracterizan a los gestos vendrán dados por magnitudes tales como distancias ( $d$ ) o ángulos ( $\alpha$ ) entre sistemas de coordenadas de la mano. Las ecuaciones (4.2) y (4.3) permiten obtener los parámetros anteriormente especificados:

$$d = |P_L^D - P_L^O| \quad (4.2)$$

$$\alpha = \tan^{-1}(|\vec{v}_O \times \vec{v}_D|, \vec{v}_O \cdot \vec{v}_D) \quad (4.3)$$

Donde:

- $O$  y  $D$  son un par de sistemas de referencia, de origen  $\{O\}$  y destino  $\{D\}$ , ambos expresados respecto al sistema de referencia global  $\{L\}$ .
- $v_O$  y  $v_D$  son vectores directores asociados al sistema de referencia de origen  $\{O\}$  y otro al de destino  $\{D\}$ , respectivamente.

Como ejemplo ilustrativo de esto, en el caso de añadir un gesto basado en la apertura y cierre de los dedos índice y corazón en forma de tijera, sería interesante almacenar *parámetros físicos* tales como la distancia y el ángulo entre las puntas de dichos dedos.

Respecto a dichos *parámetros físicos* ( $P_i$ ), cabe destacar como hipótesis de partida que la elección de los mismos variará de un gesto a otro, ya que estos dependerán de los patrones de movimiento que se desean analizar, y que su asignación se realizará de forma manual.

## 4.4. Procesado de datos

En la figura 4.2 se muestra el tratamiento que reciben los datos desde su adquisición hasta la obtención de las secuencias de característica observables.

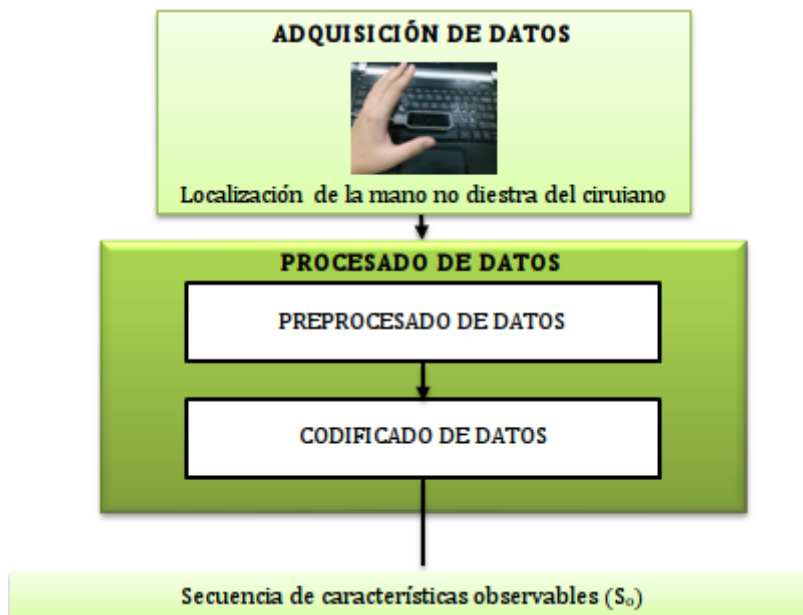


Figura 4.2. Esquema del módulo *Procesado de Datos*.

Como ya se ha comentado en el esquema general, el módulo *Procesado de Datos* tiene la función de preparar los datos de localización extraídos previamente para que sean comprensibles por los HMM y en general, para los algoritmos de entrenamiento o reconocimiento que se utilizarán posteriormente.

Para ello, consta de dos fases o módulos internos a este, el *Preprocesado de Datos* y el *Codificado de Datos*, cuya descripción se detalla en los apartados 4.4.1 y 4.4.2 respectivamente.

### 4.4.1. Fase de preprocesado de datos

En muchos casos, las posiciones y orientaciones facilitadas por el sistema de adquisición de datos están contaminadas por algunas muestras erróneas, debido a movimientos bruscos en el proceso de grabación, o a la superposición de varios dedos, lo que produce áreas de sombras que ocultan la localización de ciertos sistemas de coordenadas. Precisamente, al inicio y fin de cada grabación suele haber variaciones no deseadas en los *parámetros físicos* registrados, debido a movimientos bruscos que se producen como consecuencia de la propia condición establecida para detectar el inicio y fin de cada grabación, basada en la visibilidad de la mano por parte del sensor de localización utilizado. Para solucionar esto, se ha realizado un recorte



de los datos almacenados, de forma que se elimina un porcentaje fijo al inicio y fin de los *parámetros físicos* en todas las grabaciones, con el objetivo de disminuir estas variaciones inesperadas. En la figura 4.3 se muestra el efecto del recorte de inicio y fin realizado sobre la evolución de 3 *parámetros físicos* producidos durante el gesto de apertura y cierre de dedos índice y medio en forma de tijera.

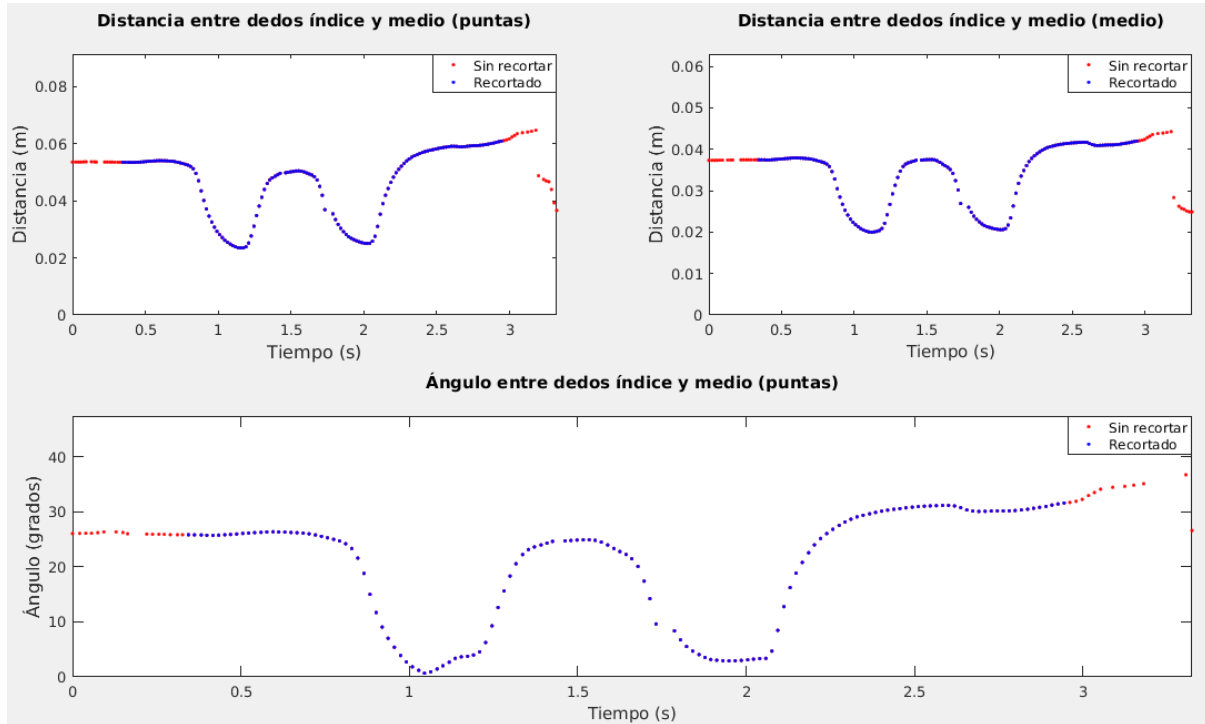


Figura 4.3. Recorte de porcentaje fijo al inicio y fin de los parámetros físicos.

En dicha figura, se observa la evolución de 3 *parámetros físicos* antes y después del recorte de forma superpuesta (en rojo antes, *Sin recortar*, y en azul después, *Recortado*). Mediante dicho recorte, se eliminan ciertas regiones que pueden dar lugar a error en el reconocimiento posterior, ya que pueden diferir mucho entre una grabación y otra pese a que en ambas se produzca el mismo gesto. Tras aplicar este procedimiento, las muestras almacenadas por cada *parámetro físico* se redistribuyen a la duración inicial de la grabación, con un período igual a dicha duración dividida entre el número de muestras que no se han recortado.

Posteriormente, para que los datos extraídos sean comprensibles por los Modelos Ocultos de Markov, es necesario que todas las grabaciones de un mismo gesto tengan igual duración y número de muestras. Para ello, se almacenan dichos parámetros por cada grabación, y se extrae posteriormente la duración y el número de muestras máximos, bien a partir de las propias grabaciones que se están utilizando para crear la *Librería de Gestos*, si se está en la *Fase de Entrenamiento*, o bien de las propias *Librerías de Gestos* directamente, si se está en la *Fase de Reconocimiento*. Esto es posible gracias a que en dichas librerías también se almacenan otros parámetros característicos de cada gesto aparte de los *HMM*, como pueden

ser la *duración máxima* y el *número de muestras máximo*. Con estos datos, se puede realizar una interpolación lineal que permita ajustar todas las grabaciones a una misma duración y misma cantidad de muestras. De esta forma, se consiguen sincronizar los vectores de los datos correspondientes los *parámetros físicos*, estableciendo un período de muestreo igual a la duración máxima dividido entre el número de muestras máximo.

Como ejemplo ilustrativo de esto, se puede ver en la figura 4.4 cómo quedarían dos grabaciones de un *parámetro físico* de un mismo gesto:

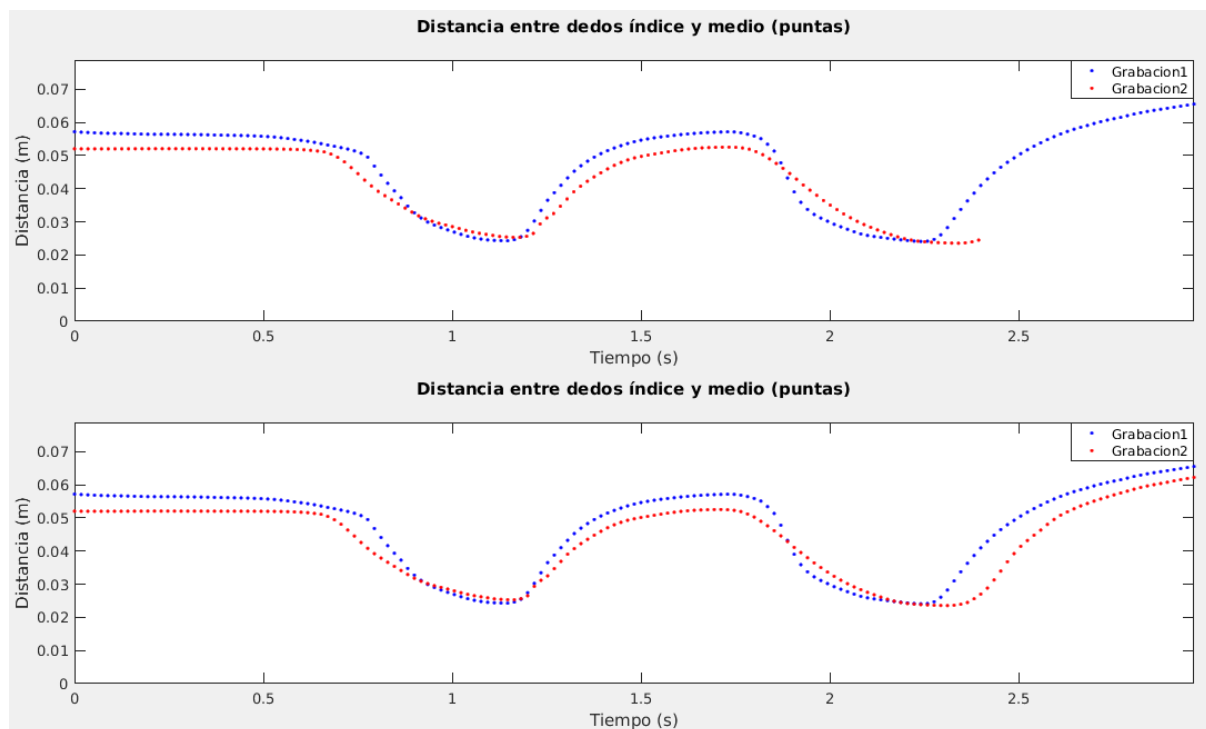


Figura 4.4. Interpolación lineal de muestras para un parámetro físico.

En la parte superior de la figura 4.4 se puede ver como para el *parámetro físico* “Distancia entre dedos índice y medio (puntas)” se dispone de diferente duración en las grabaciones 1 y 2. Gracias al almacenamiento de la duración de cada grabación durante su procesado, se puede determinar la duración máxima, e igualar la duración de todas las grabaciones a este valor, redistribuyendo e interpolando sus muestras. En la parte inferior de dicha figura se muestra el resultado de la interpolación, observando que ambas grabaciones disponen de igual duración. Por otro lado, también se equiparan el número muestras.

Aplicando el recorte comentado previamente y la interpolación lineal a la evolución de los parámetros físicos de todas las grabaciones, se consiguen secuencias de movimiento que darán lugar a un número similar de transiciones entre los estados de la red que se entrenará posteriormente. Además, se obtiene un conjunto de datos adecuado para el proceso de codificado que se detalla a continuación.

## 4.4.2. Fase de codificado de datos

Una vez ajustadas las muestras, el objetivo del codificado es preparar los datos para que sean comprensibles por los HMM. Para ello, se usa el *algoritmo K-means*, que se comporta como un clasificador, agrupando los valores de distancias y ángulos de la mano en un conjunto de categorías determinadas (ver Anexo A). El número de categorías de cada uno de los parámetros se establece a partir de la observación de los datos experimentales obtenidos y de la precisión que se considere para el modelado de los gestos.

Para realizar dicha clasificación, es necesario determinar en primer lugar los *centroides* ( $C_i$ ), es decir, los centros geométricos de estas agrupaciones, lo que servirá para clasificar posteriormente las muestras de cada uno de los *parámetros físicos*. Esto se puede ver en más detalle en la figura 4.5 que se incluye a continuación:

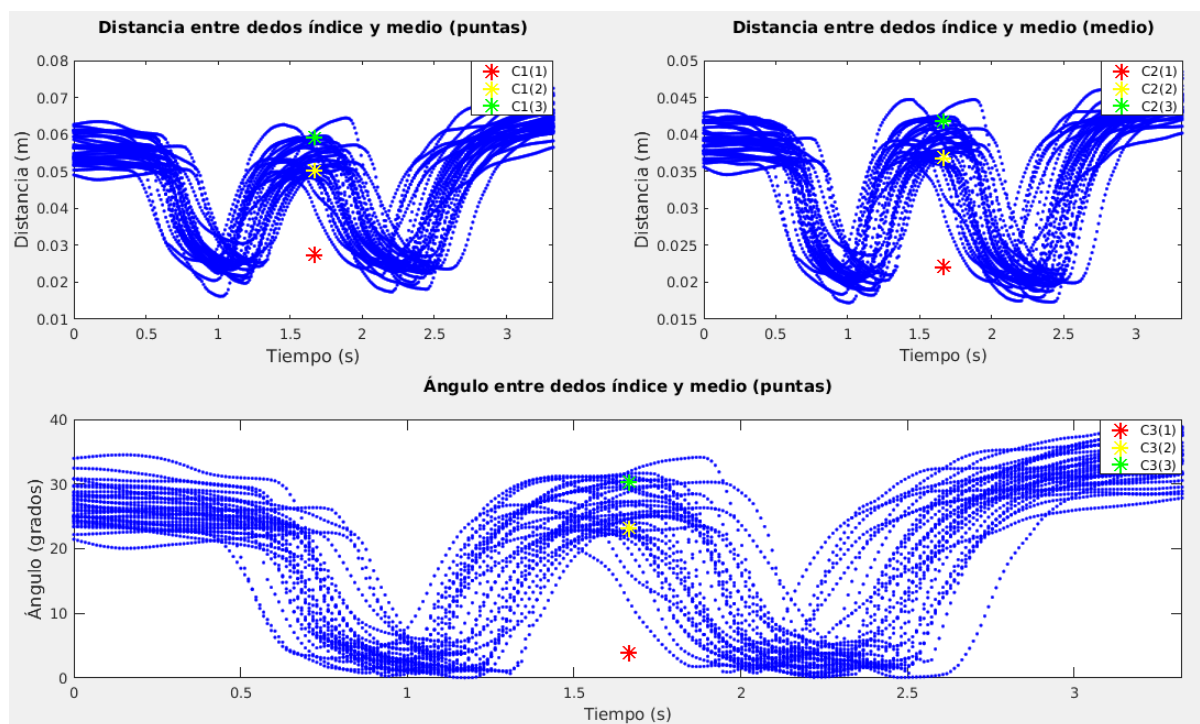


Figura 4.5. Determinación de centroides para los parámetros físicos del gesto de apertura y cierre de dedos índice y medio en forma de tijera.

Como se puede observar, para este gesto se ha decidido realizar tres agrupaciones de datos por cada uno de los tres *parámetros físicos*. Los *centroides* se pueden observar marcados con asteriscos en color rojo, amarillo y verde, y sus valores se almacenan en los vectores  $C1$ ,  $C2$  y  $C3$  respectivamente.

Posteriormente, se procede a clasificar o cuantificar cada una de las muestras de los *parámetros físicos* en una de las categorías establecidas. Para ello, se halla en primer lugar la distancia entre las muestras y sus *centroides* según la ecuación 4.4:

$$d_i(j, k) = |X_i(j) - C_i(k)| \quad i = 1 \dots nP; j = 1 \dots nM; k = 1 \dots nC \quad (4.4)$$

Donde:

*d*: Matriz de distancias entre las muestras del parámetro *i* y sus centroides

*X<sub>i</sub>*: Vector de muestras correspondientes al parámetro *i*

*C<sub>i</sub>*: Vector de centroides correspondientes al parámetro *i*

*nP*: Número de parámetros

*nM* y *nC*: Número de muestras y centroides del parámetro *i* respectivamente

Con esta matriz de distancias se puede determinar el centroide más cercano a cada muestra. Para ello, por cada *parámetro físico* *i* (*P<sub>i</sub>*), se recorre la matriz *d<sub>i</sub>* por filas, y se almacenan aquellos índices en los que se encuentra el valor mínimo de cada fila. Estos índices dan lugar un vector (*S<sub>i</sub>*) que sirve como secuencia discreta representativa de cómo varía el *parámetro físico* *P<sub>i</sub>* a lo largo de la grabación de un gesto. Aplicando el mismo procedimiento para el resto de *parámetros físicos*, se consigue registrar mediante varias secuencias discretizadas la evolución cinemática de la mano producida durante la ejecución de un gesto. Como ejemplo ilustrativo de esto, se puede observar la figura 4.6:

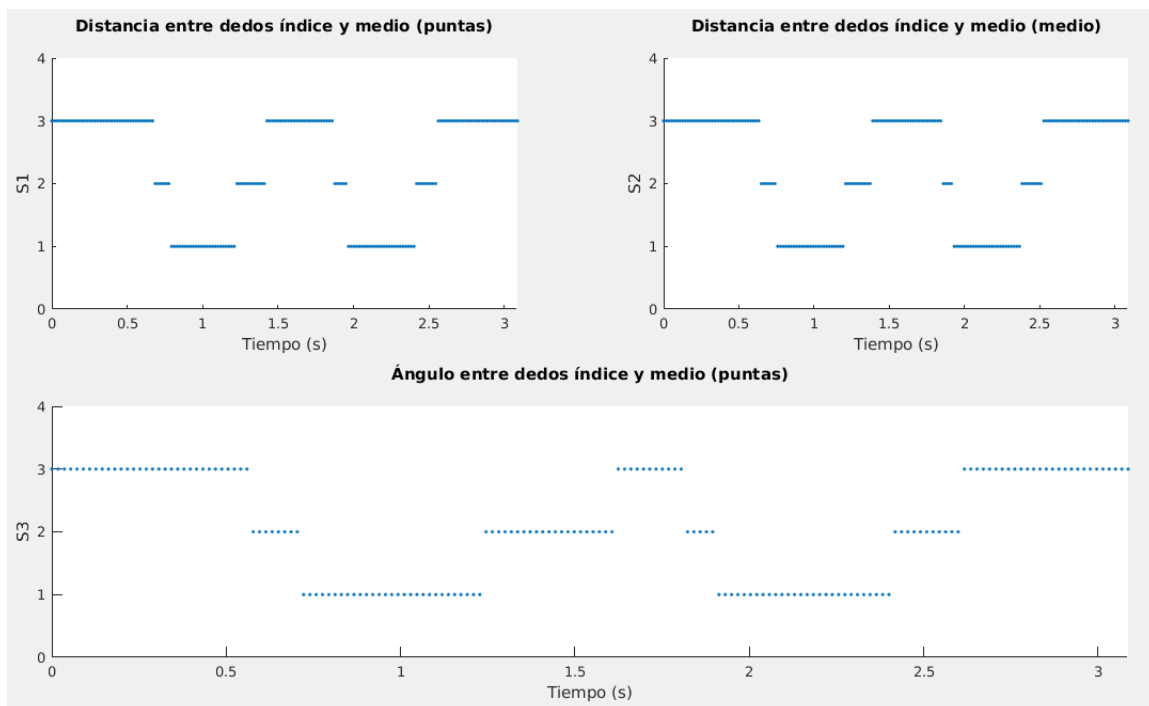


Figura 4.6. Evolución discretizada de los parámetros físicos de una grabación correspondiente al gesto de apertura y cierre de dedos índice y medio en forma de tijera.

Además, para evitar problemas de sobreoscilaciones puntuales (originadas generalmente por ruido en la medición del propio sensor) en las secuencias  $S_i$  obtenidas para cada *parámetro físico*, se incluye una zona de histéresis entre cada par de centroides contiguos, tal como se muestra en la figura 4.7 mediante líneas discontinuas:

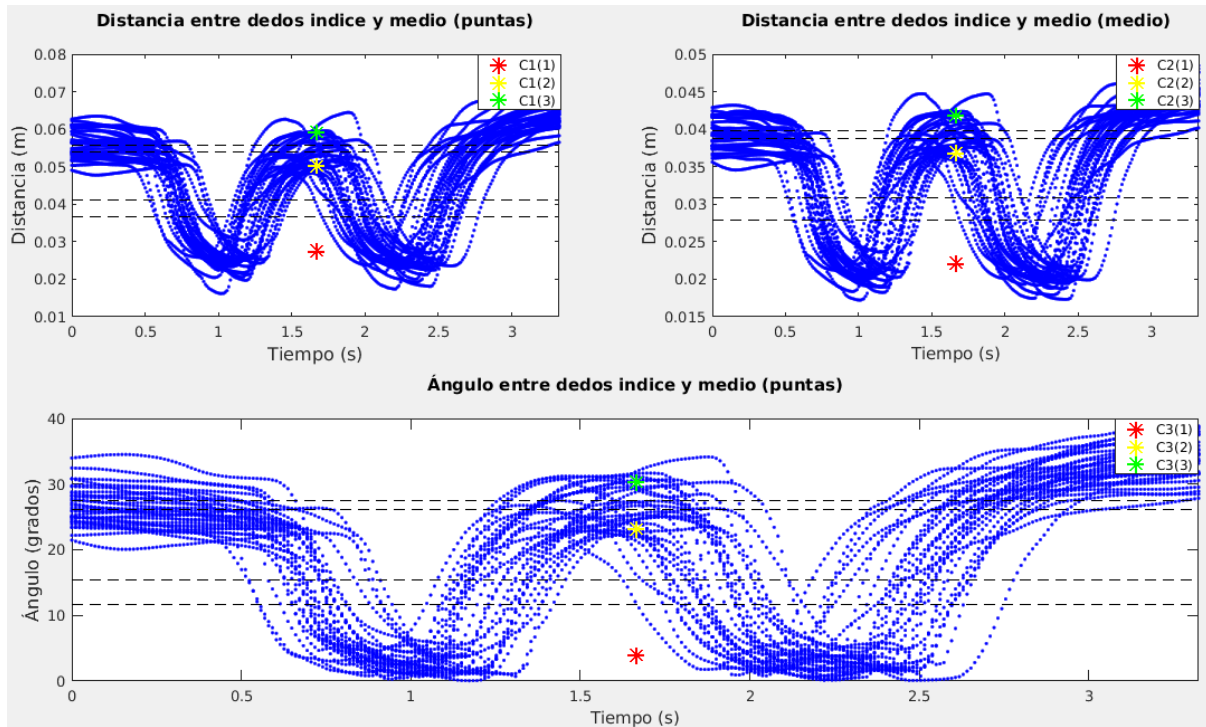


Figura 4.7. Regiones de histéresis para los parámetros físicos correspondientes al gesto de apertura y cierre de dedos índice y medio en forma de tijera.

De esta forma, si el valor de la muestra de un *parámetro físico*  $P_i$  entra en una región de histéresis, se le asigna el valor anterior de la secuencia  $S_i$ , correspondiente al centroide más cercano, y dicho valor se mantendrá hasta el momento en el que una muestra siguiente salga de la región comentada y quede más cerca otro centroide diferente.

Una vez hecho esto, se tiene que en cada instante de muestreo ( $j$ ), la postura de la mano viene definida por una combinación de valores discretos (uno por cada *parámetro físico*). A esta combinación en un instante de muestreo  $j$  se le denomina *vector característico* ( $VC$ ), y tiene la estructura que se observa en la expresión 4.5:

$$VC = (S_1(j), S_2(j), \dots, S_i(j), \dots, S_{nP}(j)) \quad (4.5)$$

De forma genérica, este vector tiene una dimensión de  $nP$  componentes, es decir, igual al número de *parámetros físicos*, donde cada componente  $S_i(j)$  puede tomar  $nC_i$  valores, lo que hace referencia al número de centroides establecidos para el *parámetro físico*  $P_i$ .

Teniendo en cuenta estos datos, y suponiendo que el *número de centroides* ( $nC$ ) se mantiene constante para todos los *parámetros físicos* de un mismo gesto, se puede deducir el *número de vectores característicos diferentes* ( $N$ ) que podrán producirse durante el registro de dicho gesto mediante la expresión 4.6:

$$N = nP^{nC} \quad (4.6)$$

Precisamente, este valor coincide con el número de *características observables* ( $o_i$ ) que describen los *estados o acciones básicas* ( $q_i$ ) que formarán el diagrama de los Modelos Ocultos de Markov (ver Anexo B). Por lo tanto, asignando a cada *vector característico* diferente una *característica observable*  $o_i$ , se consigue codificar cada una de las  $N$  combinaciones de los *parámetros físicos* de la mano en un único valor discreto, lo que permite que los gestos quirúrgicos puedan ser modelados específicamente como *secuencias de características observables* ( $S_o$ ). Esto se puede realizar mediante una tabla de asignación como la que aparece en la figura 4.8:

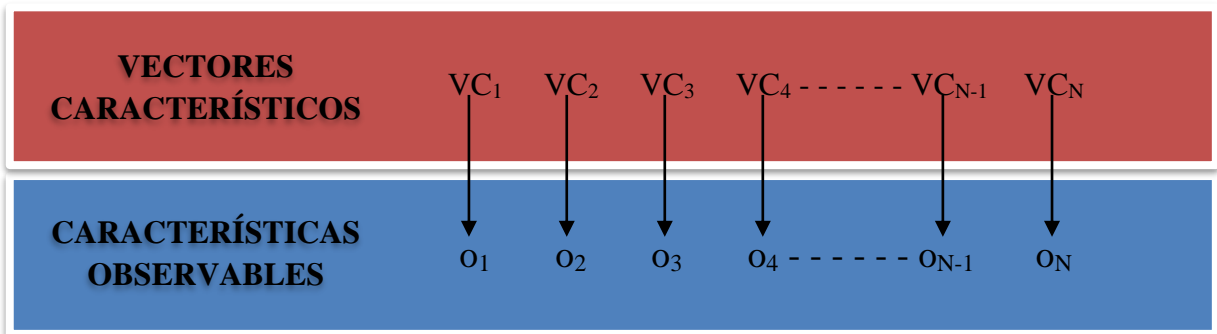


Figura 4.8. Codificado de vectores característicos en características observables.

La correspondencia entre un *vector característico* y su *característica observable* viene dada por la expresión 4.7:

$$o_i = VC_i(1) + nC * (VC_i(2) - 1) + \dots + nC^{nP-1} * (VC_i(nP) - 1) \quad i = 1 \dots N \quad (4.7)$$

Para comprender mejor el procedimiento de codificado, se muestra a continuación su desarrollo para el caso del gesto de apertura y cierre de los dedos índice y medio en forma de tijera, el cual ya se ha comentado previamente para detallar el funcionamiento de otros procedimientos llevados a cabo a lo largo del capítulo.

Puesto que dicho gesto dispone de tres *parámetros físicos* ( $nP=3$ ) y cada uno de ellos se agrupa en función de tres *centroides* ( $nC=3$ ), se consideran  $N = 27$  *vectores característicos* diferentes, resultado de tener en cuenta todas las combinaciones posibles entre las secuencias discretas  $S1$ ,  $S2$  y  $S3$  correspondientes a los *parámetros físicos*  $P1$ ,  $P2$  y  $P3$  de dicho gesto. Partiendo de la figura 4.6, se puede observar que en las primeras muestras de la grabación se dispone del vector característico  $VC = (3, 3, 3)$ .

Para identificar qué característica observable le corresponde, simplemente se aplica la expresión 4.7 citada anteriormente, tal y como se muestra en la expresión 4.8:

$$o_i = 3 + 3 * (3 - 1) + 3^2 * (3 - 1) = 27 \quad (4.8)$$

Aplicando este procedimiento a la secuencia completa de *vectores característicos* que particularizan cada grabación de un gesto, se pueden conseguir *secuencias de características observables*  $S_o$ , lo que permite modelar los gestos quirúrgicos que se deseen. Además, con los *vectores característicos* establecidos, se cubren todas las posibilidades de interacción entre las localizaciones de la mano seleccionadas.

Tras esto, se toman las secuencias resultantes del proceso de codificado y se eliminan los valores contiguos repetidos, lo que permite representar los gestos a partir de las variaciones de movimiento más destacadas, y eliminar la dependencia del tiempo, que es variable de una grabación a otra. Además, sólo se almacenan en una secuencia final aquellas *características observables* que se hayan repetido de forma contigua por encima de un valor umbral. De esta forma, se eliminan posibles transiciones puntuales que no son representativas para ese gesto, y que podrían dificultar el entrenamiento posterior. Como ejemplo ilustrativo de esto, se puede ver la figura 4.9, donde se observan la secuencia codificada obtenida para una grabación del gesto de apertura y cierre de los dedos índice y medio en forma de tijera, y la secuencia sin repeticiones resultante, suponiendo un valor umbral de cinco muestras. Se puede observar cómo se han marcado en verde aquellas subsecuencias que cumplen la condición del valor de umbral impuesto, y en verde más oscuro el primer término de cada subsecuencia, para enfatizar que no se añadirán valores contiguos repetidos.

SECUENCIA DE CARACTERÍSTICAS OBSERVABLES																													
15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	
15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	14	14	14	14	14	14	14	14	14	14	14	14	14	14
14	14	14	14	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	4	13	14	14	14	14	14	14	14	14	14	14	14	14	
14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	
14	14	14	14	14	14	5	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	4	14	14	14	14	14	14	14	14	14	14	14	14	
14	14	14	23	26	26	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	
27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	
27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	

SECUENCIA SIN REPETICIONES						
15	14	1	14	1	14	27

Figura 4.9. Ejemplo de obtención de secuencia codificada sin repeticiones.

Por último, cabe destacar que durante el codificado en la *Fase de Entrenamiento*, es necesario almacenar la longitud mínima de entre todas las secuencias sin repeticiones obtenidas para las grabaciones del gesto que se esté entrenando, y con este valor, limitar la longitud de dichas secuencias, con el objetivo de homogeneizar completamente los datos que se almacenarán en la *Librería de Gestos*.

Un ejemplo de esto se puede ver en la figura 4.10 que se muestra a continuación, para el caso de dos grabaciones de un mismo gesto:

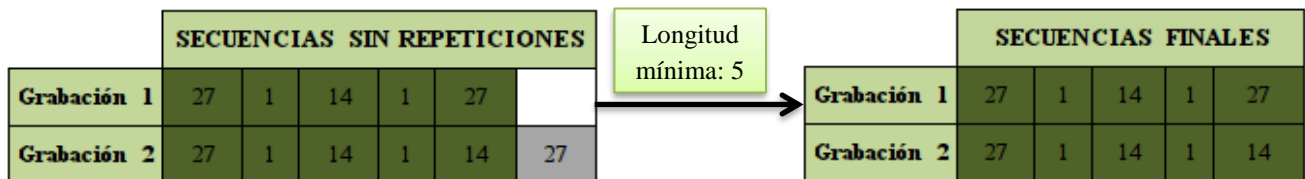


Figura 4.10. Ejemplo de obtención de secuencias de características observables finales.

## 4.5. Entrenamiento de librería de gestos

En la etapa de entrenamiento de gestos quirúrgicos, se obtienen los parámetros correspondientes a los diagramas de los Modelos Ocultos de Markov que describirán cada uno de estos gestos (ver Anexo B). De esta forma, cada uno de los gestos que se desean modelar vendrá caracterizado por un par de matrices A y B de probabilidad: por un lado, la matriz A sirve para representar la distribución de probabilidades de transición entre acciones básicas o estados ( $q_i$ ), y por otro lado, la matriz B almacena la distribución de probabilidades de características observables ( $o_i$ ) en cada acción básica o estado ( $q_i$ ). En la figura 4.11 se puede observar un ejemplo de las dos matrices comentadas con las variables que relacionan escritas en la parte izquierda en vertical, y en la superior horizontal de las mismas:

	q1	q2	q3	q4	q5
q1	0.25	0.25	0.25	0	0.25
q2	0.5	0.1	0.2	0.1	0.1
q3	0.2	0.2	0.2	0.2	0.2
q4	0.4	0.3	0.1	0.1	0.1
q5	0.1	0.2	0.1	0	0.6

Matriz A

	o1	o2	o3	o4	o5	o6	o7	o8	o9	o10
q1	0.2	0.2	0.3	0.3	0	0	0	0	0	0
q2	0.1	0.2	0.1	0.1	0.3	0.1	0.1	0	0	0
q3	0	0.2	0.15	0.15	0.1	0.2	0	0.2	0	0
q4	0	0	0.2	0.3	0.1	0.1	0.1	0.1	0.1	0
q5	0.2	0.1	0	0	0.1	0.1	0	0	0.2	0.3

Matriz B

Figura 4.11. Ejemplo de matrices A y B de probabilidad.

En este caso, las matrices de dicha figura representan un modelo formado por cinco estados o acciones básicas y diez características observables diferentes. Por un lado, los elementos de la matriz A contienen los valores de probabilidad de transición de un estado concreto a otro, y



por otro, los componentes de la matriz B indican la probabilidad de que en un estado se emita una característica observable determinada. Además, ambas matrices deben cumplir la condición de que la suma de sus elementos por filas sea igual a 1.

El número de características observables ( $o_i$ ) se puede obtener de ensayos experimentales una vez que se han recogido todos los gestos que se desean modelar, estableciendo dicho valor mediante la expresión 4.6, en función del número de parámetros físicos elegidos y del número de centroides utilizados, tal y como se comenta en el apartado 4.4.2. El número de estados o acciones básicas ( $q_i$ ) se determina mediante un proceso iterativo de prueba-error en el que se comprueban los resultados que proporciona el algoritmo de reconocimiento *Forward-Backward* (explicado en el Anexo B) para diferente número de estados.

Para el proceso de entrenamiento, es necesario definir un par de matrices A y B iniciales que permitan a los algoritmos establecer cómo puede evolucionar la red de Markov. Mientras que la matriz B puede definirse inicialmente con igual probabilidad para todos sus componentes, la matriz A inicial puede presentar diferentes configuraciones en función de la evolución esperada de los estados del modelo.

En la figura 4.12 se puede ver un ejemplo de matriz B inicial, así como algunas estructuras diferentes de la matriz A, donde las transiciones de color azul son las posibles y las marcadas en gris tienen probabilidad de salto entre estados igual a cero.

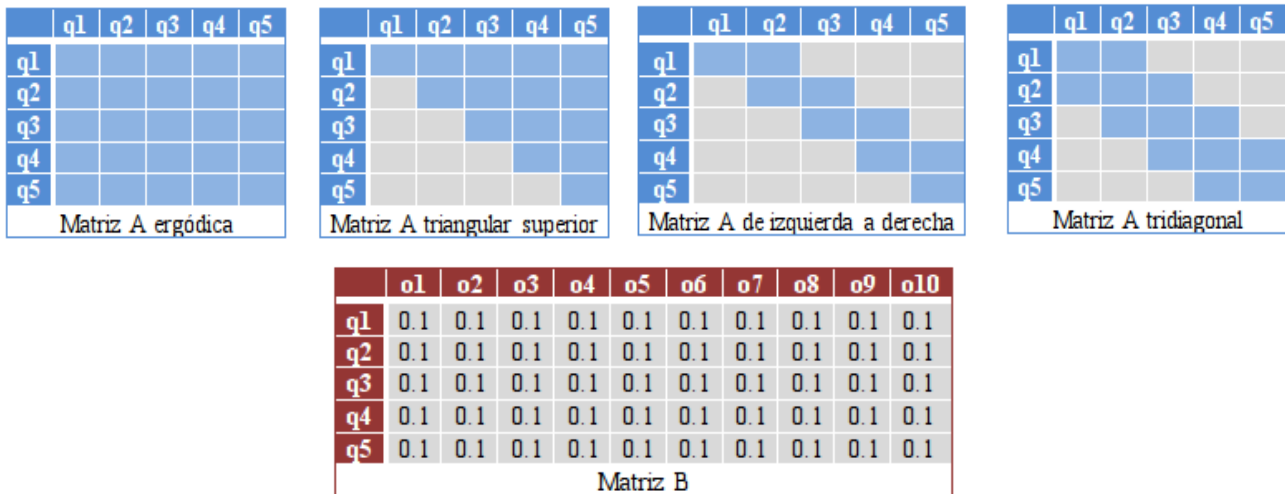


Figura 4.12. Diferentes topologías para la matriz de estados (A) y matriz de características observables (B).

En primer lugar, se muestra una matriz A ergódica, es decir aquella en la que todos los estados se encuentran conectados entre sí. Esto posibilita que cada estado pueda alcanzarse desde cualquier otro tras un número finito de pasos, ya que se permiten transiciones tanto hacia adelante como hacia atrás.

Por otro lado, la matriz A puede adoptar la forma triangular superior, con la que se permite únicamente avanzar hacia cualquier otro estado sin posibilidad de retroceder a estados anteriores. En tercer lugar, se puede observar una topología en la que sólo se permiten las transiciones de un estado a sí mismo o al siguiente estado más contiguo. Por último, se encuentra una configuración en la que se permiten transiciones de un estado a sí mismo, así como de retroceso o avance en el diagrama, aunque sólo a aquellos estados que se encuentran más contiguos.

Una vez declaradas dichas matrices, el procedimiento consiste en utilizar el *Algoritmo de Baum-Welch* para el proceso de entrenamiento (explicado en el Anexo B). Dicho algoritmo permite ajustar adecuadamente los valores de las matrices A y B para modelar cada uno de los gestos. Para ello, además de las matrices iniciales comentadas previamente, se requiere de un conjunto de secuencias que permitan modelar los patrones de movimiento de la mano, y estas son precisamente las *secuencias de características observables* ( $S_o$ ) obtenidas según el procedimiento que se detalla previamente en el apartado 4.4.2.

Como resumen del proceso de entrenamiento, se puede ver el esquema que se muestra en la figura 4.13:

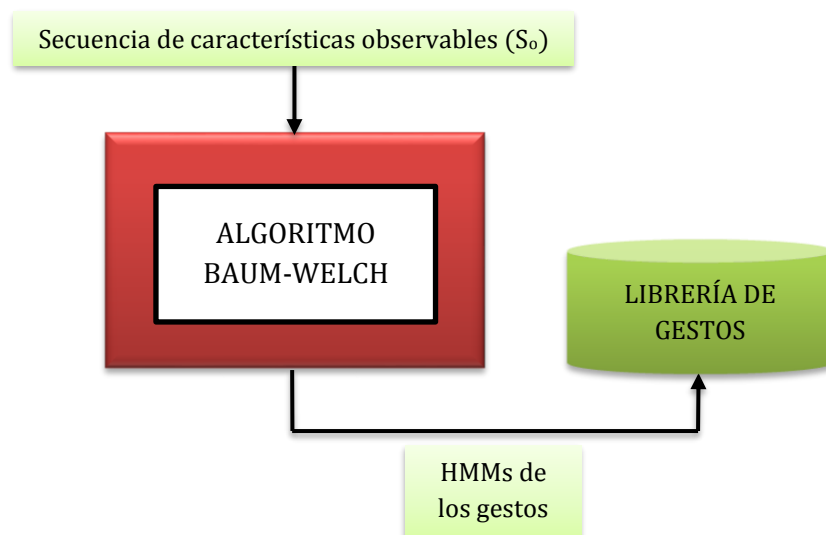


Figura 4.13. Entrenamiento de los HMMs mediante el algoritmo de Baum-Welch.

Cabe destacar que para que el modelado sea adecuado, es necesario disponer un número de grabaciones suficiente de cada gesto de las que obtener las secuencias utilizadas para el entrenamiento. Con esto se consiguen ajustar los valores de probabilidad asociados a las transiciones posibles entre los estados definidos en las redes de Markov.

Como resultado de aplicar este algoritmo, se obtienen las matrices A y B por cada gesto que se desea modelar, y estas se almacenan en la librería de gestos que se añadirá en el sistema. Dicha librería sirve como base en el proceso de reconocimiento en línea, ya que el modelado

probabilístico de los movimientos realizados en las grabaciones de entrenamiento permitirá evaluar si en una intervención dada se ha realizado un gesto.

Además, cabe destacar que este procedimiento debe realizarse para cada sujeto que utilice el sistema, de forma que se adapte con mayor precisión al individuo del que se pretenden reconocer los gestos.

## 4.6. Reconocimiento de gestos

En esta fase se detalla el proceso de detección de gestos tal y como se muestra en la figura 4.14:

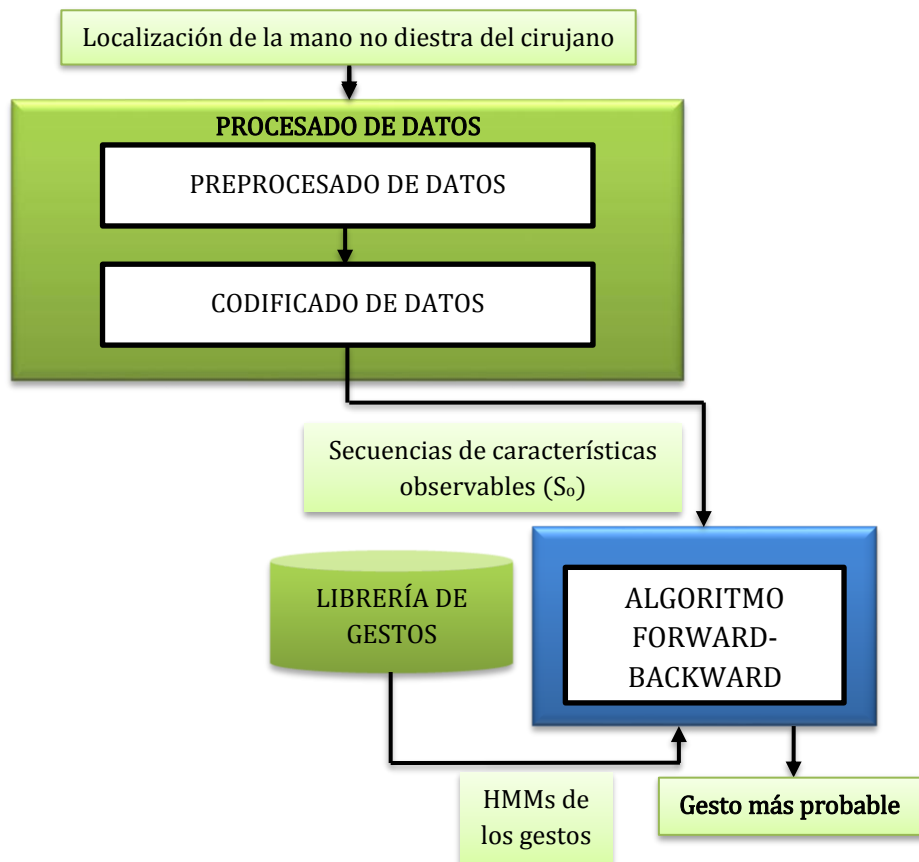


Figura 4.14. Esquema del sistema de reconocimiento de gestos quirúrgicos.

Como se puede observar en dicha figura, en dicho proceso participa el módulo de *Procesado de Datos* comentado en el apartado 4.4, así como un módulo reconocedor propiamente dicho, representado mediante el bloque *Algoritmo de Forward-Backward*. Ambos módulos interactúan teniendo como base la localización de la mano no diestra del cirujano, y con el objetivo final de permitir reconocer gestos en tiempo real. Esto es posible gracias a que se comparan *secuencias de características observables* con los patrones de movimiento de la mano almacenados en la *Librería de Gestos*, y se devuelve el gesto más probable.

Para ello, en primer lugar el módulo de *Procesado de Datos* recibe datos de la localización de la mano no diestra del cirujano. El procedimiento a seguir por dicho módulo es similar al comentado en el apartado 4.4, sin embargo, cabe destacar algunas diferencias:

- A priori, no se sabe cuál es el gesto que se está realizando, por lo que a partir de los datos de localización de la mano se deben codificar tantas secuencias como gestos incluya el sistema. Así, para el caso de un sistema genérico, se codificarían  $n$  secuencias  $o_i$  correspondientes a los  $n$  gestos entrenados ( $S_{o1}, S_{o2}, \dots, S_{on}$ ).
- La grabación del gesto a reconocer debe de ser comparable con las grabaciones utilizadas para el sistema de entrenamiento. Para ello, es necesario realizar la homogeneización de parámetros tales como la duración de la grabación o el número de muestras en el *Preprocesado de Datos*, para que la salida proporcionada por el sistema de reconocimiento posterior sea válida.
- Además, es necesario proporcionar al módulo de *Codificado de Datos* los valores de los centroides obtenidos en la etapa de entrenamiento de los gestos, así como la longitud mínima de las *secuencias de características observables*, para permitir una codificación basada en los mismos criterios que en las grabaciones del sistema de entrenamiento.

En la figura 4.15 se pueden observar de forma resumida los parámetros necesarios para realizar el ajuste de los datos necesario para el reconocimiento:

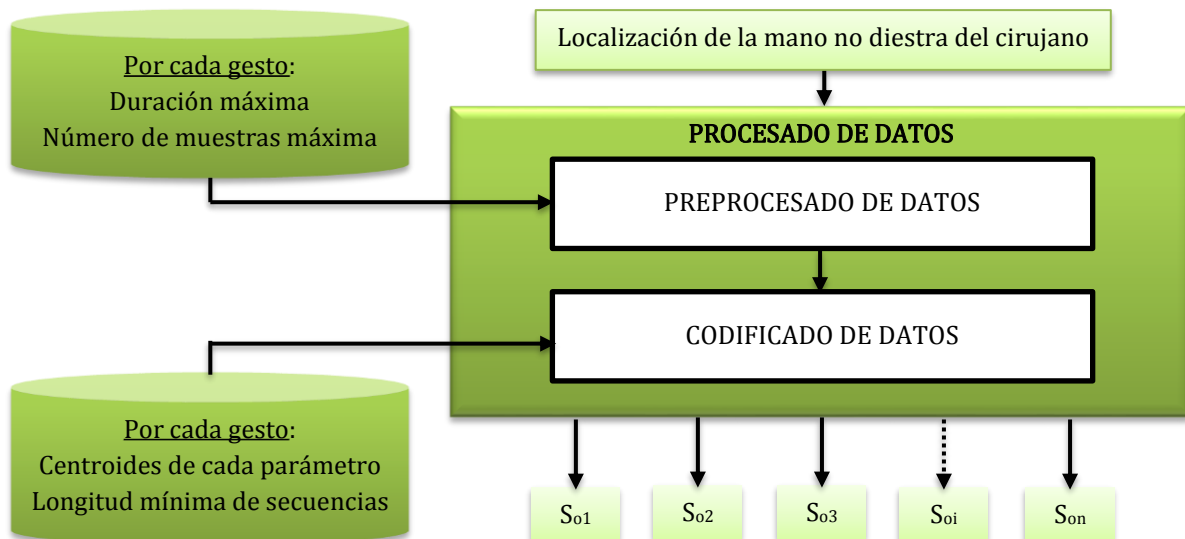


Figura 4.15. Obtención de secuencias de características observables en el proceso de reconocimiento.

Una vez codificadas y limitadas las secuencias a la longitud mínima dada por el sistema de entrenamiento, el *algoritmo de Forward-Backward* permite calcular la probabilidad ( $p_i$ ) de cada gesto, gracias a que cada secuencia obtenida se compara con los HMM que representan a los gestos entrenados. Estos valores de probabilidad vienen dados en escala logarítmica, debido a que son valores muy pequeños.

En este punto, el problema consiste en encontrar el modelo que mejor se adapte a la secuencia que le corresponde, y con esto, determinar qué gesto es más probable que se haya producido, o si no se ha detectado ninguno con suficiente confianza. Se puede observar un esquema más detallado del proceso de reconocimiento en la figura 4.16:

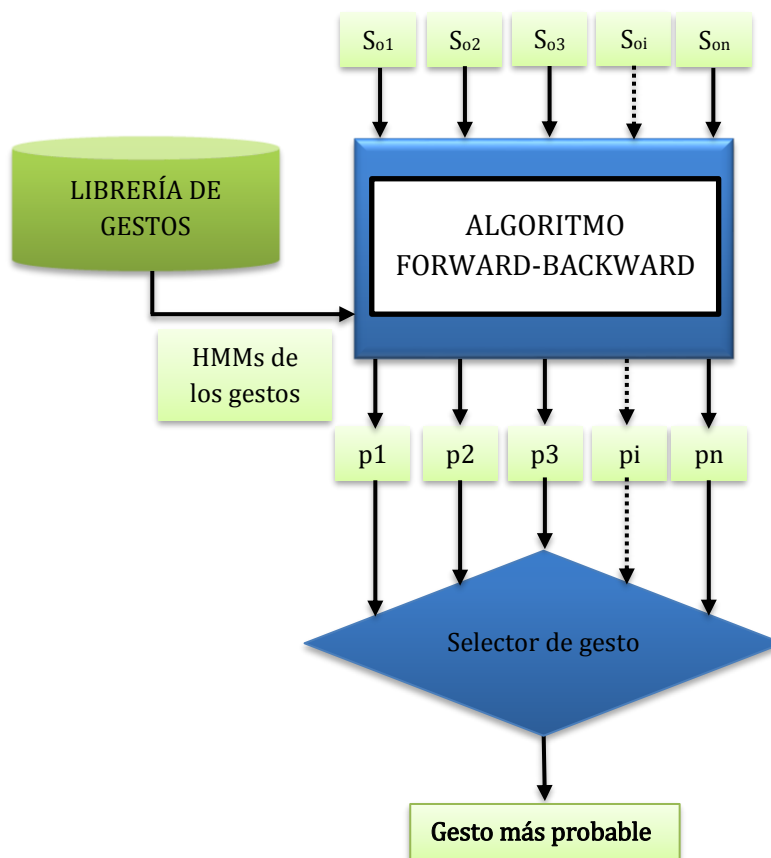


Figura 4.16. Esquema detallado del proceso de reconocimiento.

A partir de las salidas proporcionadas por el *algoritmo de Forward-Backward*, se puede extraer el modelo que posee un valor de probabilidad  $p_i$  más alto para las secuencias de observaciones obtenidas. Sin embargo, para que el reconocimiento realizado sea válido, se dispone de varios criterios más que permiten garantizar que en la detección no se han producido falsos positivos, es decir, que no se ha determinado que el gesto más probable es el primero por ejemplo cuando no se estaba realizando ningún gesto, o cuando se estaba realizando cualquier otro incluido en la librería de gestos.

Entre estos criterios, cabe la pena destacar los siguientes:

- Supongamos que la longitud de la secuencia de observaciones obtenida durante el proceso de reconocimiento para el gesto  $i$  ( $longS_{oi}$ ) es menor que la longitud mínima de las secuencias obtenidas durante el proceso de entrenamiento del gesto  $i$  ( $longMinSecEntrG_i$ ), tal y como se muestra en la expresión 4.9:

$$longS_{oi} < longMinSecEntrG_i \quad (4.9)$$

En dicho caso, puede considerarse que en la grabación a reconocer no se habrá producido el gesto  $i$ , ya que el entrenamiento incluye suficientes repeticiones de dicho gesto como para determinar un valor umbral mínimo de características observables que se han de producir para poder afirmar que dicho gesto se ha producido.

Además, ya que en estas secuencias se almacenan únicamente las variaciones entre características observables, una longitud menor que la mínima esperada implicará que los parámetros relevantes para ese gesto apenas han variado, lo que justifica la decisión tomada.

- En el caso de que se obtenga un valor de probabilidad infinito para uno de los gestos, este queda descartado. Como se ha comentado anteriormente, los valores de probabilidad vienen dados en escala logarítmica, y si se obtiene un valor de infinito se puede deducir que el logaritmo se está aplicando a un valor muy cercano a cero, es decir, con una probabilidad tan baja que la operación no está definida.

Desde el punto de vista matemático de los HMM, se puede entender que se obtengan estos valores como consecuencia de que en las matrices B (de probabilidad de que en un estado  $q_i$  aparezca una característica observable  $o_i$  determinada) las probabilidades asociadas a la secuencia de entrada que se le pasa al algoritmo sean muy bajas o incluso nulas, debido a que dichas características no son las que habitualmente aparecen en el gesto en cuestión, y por tanto no es posible avanzar adecuadamente a lo largo de la red de Markov.

- Si no ocurre ninguna de las dos situaciones comentadas previamente, entonces se sigue el criterio de comprobar si la probabilidad obtenida para el gesto  $i$  ( $p_i$ ) es igual o superior a la mitad de la probabilidad mínima de las grabaciones utilizadas en el sistema de entrenamiento para ese gesto ( $probMinSecEntrG_i$ ), tal como se puede ver en la expresión 4.10:

$$p_i \geq \frac{probMinSecEntrG_i}{2} \quad (4.10)$$

En dicho caso, la secuencia  $S_{oi}$  puede corresponder al gesto  $i$  que se está evaluando. Esto tiene el objetivo de descartar aquellas secuencias cuya probabilidad es muy pequeña, y se ha considerado que partir de las probabilidades de las grabaciones utilizadas en el entrenamiento del gesto  $i$  puede servir como base para resolver este problema.

Como resumen, se pueden ver en el diagrama de flujo que se muestra en la figura 4.17 los criterios para determinar si es posible que se haya producido un determinado gesto en una grabación:

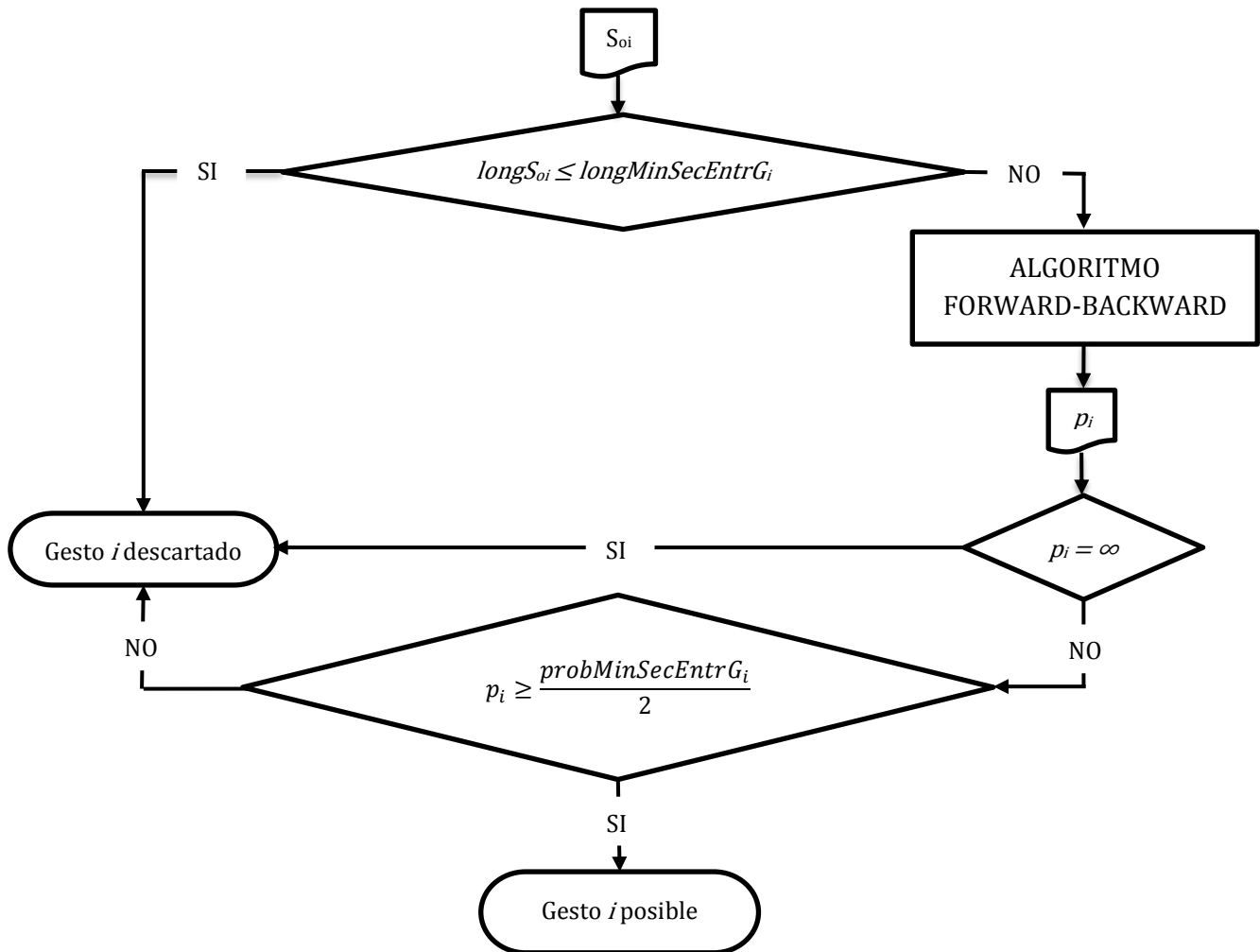


Figura 4.17. Diagrama de flujo con criterios de decisión en el proceso de reconocimiento.

Aplicando este procedimiento a todas las secuencias de características observables obtenidas, se puede llegar a cualquiera de las siguientes situaciones:

- Que todos los gestos queden descartados en dicha grabación, bien por los criterios de probabilidad o de longitud de las secuencias, con lo que se concluiría que no se ha identificado ningún gesto.

- Que se detecte más de un gesto como posible, en cuyo caso no se podrá afirmar con seguridad el reconocimiento en la grabación.
- Que se detecte un sólo gesto como posible, en cuya situación finaliza el proceso devolviendo el gesto reconocido.

El diagrama de la figura 4.18 que se observa a continuación nos permite esquematizar los casos que pueden darse en el proceso de reconocimiento.

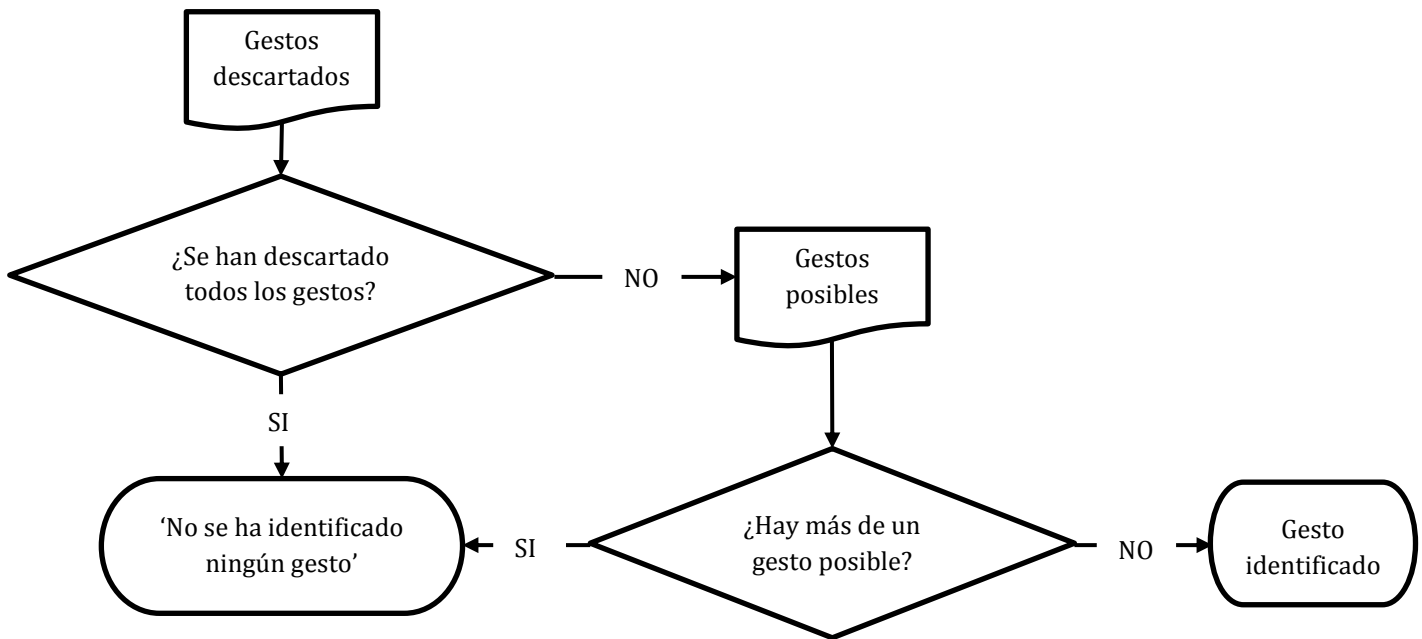


Figura 4.18. Diagrama de flujo con las situaciones posibles en el sistema de reconocimiento.

De esta forma, el sistema de reconocimiento finaliza con la emisión del gesto identificado, o bien con la emisión de un mensaje indicando que no se ha identificado ningún gesto si no se dispone de suficiente certeza para distinguirlo.

## 4.7. Conclusiones

En este capítulo se ha propuesto la metodología utilizada en el desarrollo de un sistema de reconocimiento de gestos quirúrgicos para cirugía laparoscópica asistida por la mano, con el objetivo de implementar una interfaz capaz de actuar en función del gesto realizado por el cirujano durante la maniobra quirúrgica. De esta forma, es posible dotar a un sistema robótico quirúrgico de la capacidad de colaborar con el cirujano en determinadas tareas, mediante comandos directos dictados por este.



Para comprender el sistema, se ha comentado el funcionamiento de los procesos y módulos utilizados desde la obtención de la localización de la mano hasta un módulo de reconocimiento, pasando por una fase de procesamiento de datos, en la cual se han tratado y sincronizado los datos obtenidos. De igual forma, se ha detallado la fase de entrenamiento, necesaria para entrenar los modelos que se han de utilizar para el reconocimiento posterior.

El modelado de los gestos es posible gracias a la creación de una librería de modelos estocásticos con los patrones de movimiento realizados durante una intervención. Para ello, se han utilizado los Modelos Ocultos de Markov, con los que se representan y analizan secuencias de características observables producidas durante la ejecución de los movimientos de la mano.

La topología inicial de los Modelos Ocultos de Markov es función del tipo de sistema que se desea modelar, los gestos que se desean reconocer, la flexibilidad permitida en la evolución entre los distintos estados y el número de características observables establecidas. Una vez establecida la topología de los modelos de los gestos, se lleva a cabo un proceso de entrenamiento para obtener las matrices A y B que caracterizan cada gesto incluido en el sistema.

Por otro lado, cabe destacar que los patrones que forman la librería de gestos deben formarse a partir de los datos extraídos del usuario que vaya a utilizar el sistema, y el modelado que mejor representa los movimientos realizados por la mano es aquel que ha sido entrenado con un gran número de repeticiones para cada uno de los gestos.

Finalmente, se ha comentado cómo el sistema de reconocimiento es capaz de identificar un gesto en una determinada grabación, gracias a la comparación entre secuencias de características observaciones obtenidas y los patrones almacenados en la librería de gestos. Así mismo, se ha detallado mediante diagramas de flujo algunos de los criterios utilizados en el módulo de reconocimiento para el disminuir la aparición de falsos positivos en la identificación.



# Capítulo 5: Implantación y experimentos

## 5.1. Introducción

En los capítulos anteriores se ha planteado, por un lado, la problemática a cubrir con el desarrollo de una interfaz que permita el reconocimiento de gestos, y por otro, los desarrollos teóricos necesarios para implantar cada uno de los módulos que constituyen dicha interfaz, con el objetivo de comprender el funcionamiento del sistema.

El presente capítulo describe en primer lugar las arquitecturas hardware y software propuestas para implantar el sistema con el objeto de incluirlo en una plataforma robótica dotada de una arquitectura basada en el entorno ROS, incluyendo una breve descripción de cada uno de los componentes utilizados. Posteriormente, se detalla el proceso de modelado de gestos quirúrgicos a través de ensayos realizados por diferentes usuarios. Así mismo, se incluyen diferentes experimentos realizados para comprobar cómo varía el reconocimiento en función del uso de diferentes topologías de los HMMs en la fase de entrenamiento.

Una vez validada la topología a utilizar, se realizan una serie de experimentos con el objetivo de demostrar que el sistema de reconocimiento de gestos funciona correctamente, y determinar si afecta lo familiarizados que estén los diferentes usuarios con el Leap Motion con el índice de acierto del sistema.

Por otro lado, hay que mencionar que los experimentos expuestos en este capítulo tienen como objetivo validar una metodología para la detección de gestos en cirugía laparoscópica asistida por la mano, y su integración en un entorno ROS. Pese a que dichos experimentos son realizados mediante el dispositivo Leap Motion (tal y como se detallará en el apartado 5.2), sus resultados son completamente extrapolables al entorno de una intervención donde se utilice un guante quirúrgico sensorizado, dispositivo con el que se tiene planteado utilizar el sistema de detección de gestos.

## 5.2. Implantación del sistema

Para comprender el modelado y reconocimiento de gestos planteado, se ha decidido explicar por un lado los componentes utilizados desde el punto de vista hardware, y posteriormente los componentes software.

### 5.2.1. Arquitectura hardware

En primer lugar se detalla el entorno hardware, el cual incluye un sistema de adquisición de datos de la localización de la mano del cirujano, que permite registrar los movimientos que

éste realiza durante la intervención, así como un dispositivo que permite el almacenamiento y tratamiento de los datos obtenidos. En la figura 5.1 que se muestra a continuación pueden observarse los dispositivos utilizados para recoger las grabaciones de los gestos utilizadas tanto para las fases de entrenamiento como para validar el sistema de reconocimiento:

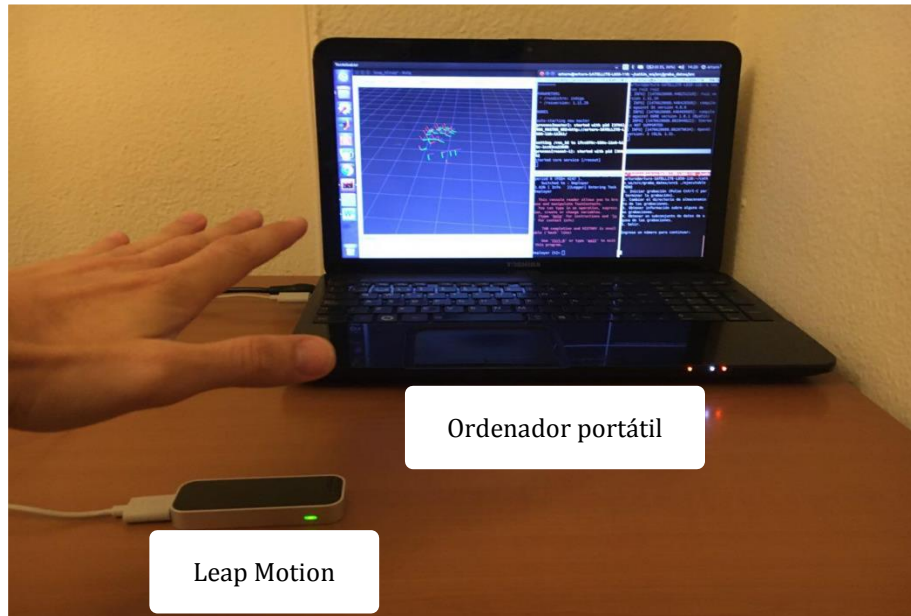


Figura 5.1. Plataforma hardware utilizada para el modelado y reconocimiento de gestos.

## Leap Motion

En el caso del sistema de adquisición de datos, se ha utilizado el componente Leap Motion (ver figura 5.2), un dispositivo usb periférico cuyo objetivo es sensar el movimiento de las manos en 3D, lo que puede permitir interactuar con computadores e incluso mejorar experiencias de realidad virtual conjuntamente con dispositivos tales como Oculus Rift (un casco de realidad virtual que permite la inmersión total en campos como videojuegos, la educación o el cine).



Figura 5.2. Dispositivo Leap Motion.

Dicho dispositivo está constituido internamente por dos cámaras, tres LEDs infrarrojos y un microcontrolador, tal y como se muestra en la figura 5.3:

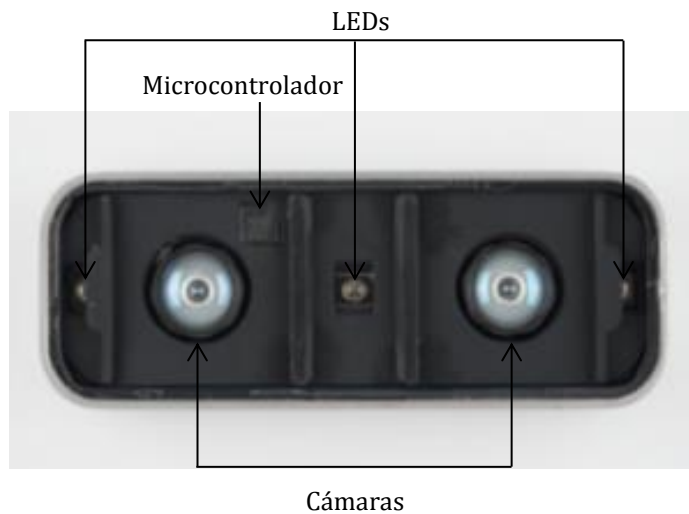


Figura 5.3. Disposición de componentes internos del dispositivo Leap Motion.

En primer lugar, las cámaras se encargan de capturar imágenes constantemente, lo que permite determinar la localización de las articulaciones y de algunos otros puntos de interés en las manos del usuario que usa el dispositivo. Para ello, cada una de las cámaras cuenta con un sensor monocromático sensible a la luz infrarroja, con una longitud de onda de 850 nm. Estos sensores pueden trabajar a una velocidad de hasta 200 fps dependiendo del rendimiento del equipo al que se conecte el dispositivo.

Por otro lado, los LEDs se encargan de iluminar la zona de cobertura del dispositivo en el espectro de luz infrarroja, a una longitud de onda de 850 nm, y varían su iluminación dependiendo de la luz del entorno, para asegurar una misma nitidez de imagen. Dichos LEDs se encuentran separados entre sí mediante barreras de plástico, para así garantizar que la iluminación sea uniforme en toda la zona de cobertura. Además, de esta forma se protege a los sensores ópticos de una posible saturación de luz, ya que con dicha disposición la luz infrarroja no les ilumina directamente.

Por último, el microcontrolador consiste en un circuito integrado utilizado como sistema básico de entrada y salida (BIOS). Contiene el programa que controla al dispositivo, y se encarga de recoger información de los sensores para luego enviarla al controlador (driver) instalado en el ordenador, así como de otras actividades básicas tales como regular la iluminación.

### **Ordenador portátil**

Como dispositivo de almacenamiento y tratamiento de datos se ha utilizado un ordenador portátil con sistema operativo Ubuntu 14.04 LTS, con ROS configurado y con los nodos

necesarios para la adquisición y visualización de la localización de las manos. Además, dispone de Matlab 2016a para permitir el tratamiento de los datos obtenidos, como se detallará en el apartado posterior.

### 5.2.2. Arquitectura software

A continuación se describe la arquitectura software utilizada para posibilitar la comunicación entre dispositivos a través de un entorno ROS, para iniciar y finalizar las grabaciones de datos, así como para permitir el ajuste y sincronización de los datos obtenidos.

#### Entorno ROS

La plataforma ROS consiste en un “meta-sistema operativo” de código abierto que proporciona un conjunto de librerías y herramientas para robots, lo que permite a desarrolladores software crear aplicaciones robóticas con mayor facilidad, así como integrar diferentes plataformas robóticas dentro de una red que permite la comunicación entre ellas. Entre sus principales características, destaca su diseño distribuido y modular, la abstracción del hardware, el control de dispositivos a bajo nivel o el envío de mensajes entre procesos.

Para comprender el funcionamiento básico de ROS, hay que destacar los principales elementos que lo componen:

- **Nodos:** Un nodo es un proceso ejecutable que usa ROS para comunicarse con otros nodos, cumpliendo una función específica. Son la estructura más básica de toda arquitectura basada en ROS, y son compilados, ejecutados y gestionados individualmente unos de otros mediante un nodo principal, denominado nodo roscore.
- **Mensajes:** Los mensajes son tipos de estructuras de datos que permiten la comunicación interna entre nodos de forma asíncrona. Pueden incluir tipos de datos primitivos, arrays de datos o estructuras.
- **Topics:** Son intermediarios en la comunicación entre nodos, ya que actúan como sistema de transporte a través del cual se intercambian mensajes. Los nodos pueden publicar mensajes en un topic o bien suscribirse a estos; de esta forma, si un nodo publica un determinado mensaje en un topic, todos los nodos que estén suscritos a éste lo leerán. Puede haber múltiples nodos publicadores y suscriptores para un mismo topic, y de igual forma, un mismo nodo puede publicar y/o suscribirse a múltiples topics.

- **Servicios:** Los servicios son transacciones síncronas entre nodos, basadas en la arquitectura cliente/servidor, en las que un nodo actúa como cliente realizando una petición a otro nodo, y este proporciona una respuesta actuando como servidor, bien en forma de mensaje con datos o actuando como controlador de algún componente robótico presente en la plataforma.

Por otro lado, cabe destacar que todo el software de ROS está organizado en **paquetes**, cuyo objetivo es ofrecer una funcionalidad determinada conteniendo nodos, conjuntos de datos, tipos de mensajes definidos por el usuario, archivos de configuración u otros elementos. Precisamente, para el desarrollo de este trabajo se han creado varios paquetes ROS:

- **Paquete 1: rtt\_dev\_leapmotion:** Este paquete dispone de un nodo encargado de publicar las lecturas proporcionadas por el dispositivo Leap Motion en diferentes topics dentro del entorno ROS. Además, permite observar la localización de los distintos sistemas de coordenadas proporcionados para ambas manos mediante un visualizador de ROS denominado *rviz*, tal y como se puede ver en la figura 5.4:

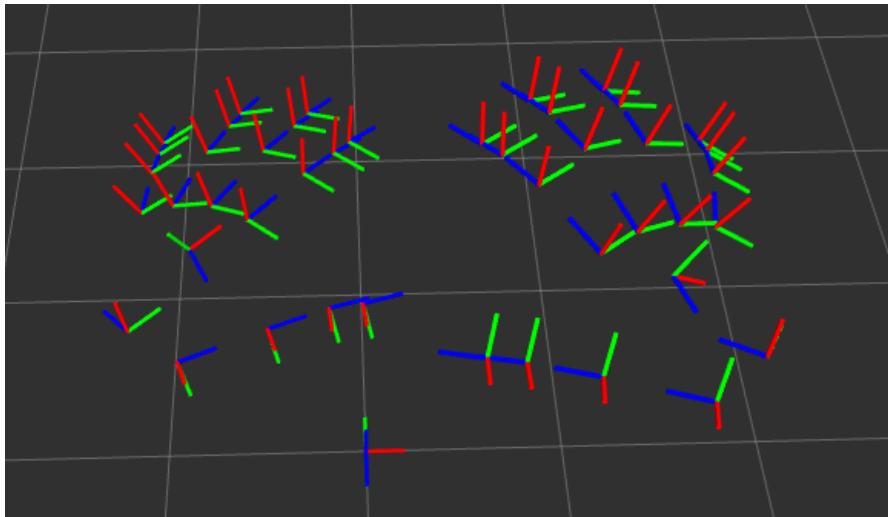


Figura 5.4. Visualización de sistemas de coordenadas de las manos mediante la herramienta *Rviz*.

- **Paquete 2: graba\_datos:** En este otro paquete se ha programado mediante código en C++ un menú con varias opciones a elegir, cada una con una tarea asignada. Entre dichas tareas destacan:
  - La posibilidad de realizar grabaciones de los datos proporcionados por Leap Motion durante la ejecución de un gesto.
  - Elegir cuáles de los topics publicados se desean almacenar.
  - Obtener información sobre alguna grabación ya almacenada.

- Establecer el directorio de almacenamiento.
- Obtener un subconjunto de datos de las grabaciones.

Para su funcionamiento, dicho código está basado en *rosvbag*, una herramienta de ROS para grabar, reproducir y modificar archivos de registro (bags) con los mensajes que recibe el nodo *roscore*.

Posteriormente, gracias a la toolbox de Matlab “Robotic Toolbox System”, se ha creado un topic denominado `/gestoIdentificado`, donde Matlab, que actúa como nodo publicador, puede enviar mensajes directamente sobre ROS con los gestos identificados tras pasar grabaciones por el sistema de reconocimiento. En la figura que se muestra a continuación se puede ver un ejemplo de cómo ocurriría esto:

```
>> rosinit
Initializing ROS master on http://arturo-SATELLITE-L850-118:11311/.
Initializing global node /matlab_global_node_06960 with NodeURI http://arturo-SATELLITE-L850-118:37881/
>> bag='Grabacion31.bag';
>> reconoceGesto(bag,maxT,maxMuestras,centroides,longitudMinimaSecuencias,AGestos,BGestos,probabilidadMinima);
Archivo seleccionado: Grabacion31.bag

Puede ser el gesto 1. Probabilidad -1.4194.
No parece ser el gesto 2. Probabilidad infinita.
No parece ser el gesto 3. Longitud de secuencia (1) menor de longitud mínima.
No parece ser el gesto 4. Longitud de secuencia (1) menor de longitud mínima.

Gesto identificado: 1
```

```
arturo@arturo-SATELLITE-L850-118: ~
arturo@arturo-SATELLITE-L850-118: ~ 80x24
arturo@arturo-SATELLITE-L850-118:~$ rostopic echo /gestoIdentificado
data: 1
---
```

Figura 5.5. Ejemplo de publicación de mensaje con el gesto identificado sobre ROS.

Como se puede observar, se adjuntan una captura de la ventana de comando de Matlab, y una ventana de terminal de Ubuntu. En Matlab, se muestra la inicialización de un *nodo maestro* o *roscore* mediante el comando “*rosinit*”, así como la llamada a la función “*reconoceGesto*” con los parámetros necesarios para determinar el gesto producido en el archivo “*Grabacion31.bag*”. Tras esto, se muestran los resultados obtenidos. Por otro lado, en la ventana de terminal se muestra la suscripción al topic de interés mediante el comando “*rostopic echo /gestoIdentificado*”, y cómo se recibe que se ha identificado el primer gesto.



De esta forma, tanto el dispositivo Leap Motion como el sistema de reconocimiento de gestos quedan integrados en la arquitectura de la plataforma robótica del grupo de investigación de Robótica Médica de la Universidad de Málaga, cerrando uno de los objetivos principales del trabajo.

## **MATLAB**

MATLAB es una herramienta de software matemático que ofrece un entorno de desarrollo integrado con un lenguaje de programación propio basado en matrices. Esta plataforma está optimizada para resolver problemas de ingeniería y científicos, permitiendo analizar y diseñar sistemas con grandes cantidades de datos, para lo cual dispone de gran cantidad de funcionalidades en campos como procesamiento de señales, robótica, comunicaciones, diseño de control, etc.

En el caso de este trabajo, ha servido como entorno donde procesar, recortar y sincronizar las grabaciones de datos almacenadas previamente en archivos bag, siguiendo el proceso comentado en el Capítulo 4. De igual forma, ha permitido crear las funciones necesarias para llevar a cabo el reconocimiento de gestos, y como se ha comentado previamente, crear los nodos necesarios para poder comunicar dicho reconocimiento al entorno ROS.

## **5.3. Descripción de los experimentos**

En este apartado se describe el procedimiento que se ha llevado a cabo para validar el sistema de reconocimiento de gestos quirúrgicos. En primer lugar, se exponen los objetivos que se desean alcanzar en el estudio. Tras esto, se describen los gestos registrados para evaluar el funcionamiento del reconocedor, así como los parámetros físicos utilizados para el codificado de secuencias de características observables. Posteriormente, se define la batería de experimentos utilizada, así como las tareas que se van a realizar y los datos registrados para evaluar el funcionamiento del sistema. Finalmente, se realiza un análisis de los resultados obtenidos, discutiendo acerca de la validez del trabajo desarrollado.

### **5.3.1. Objetivo**

El objetivo principal consiste en validar un sistema de reconocimiento de gestos aplicable a intervenciones realizadas mediante cirugía laparoscópica asistida por la mano.

Por otro lado, la realización de experimentos pretende establecer la topología de los Modelos Ocultos de Markov más adecuada para representar los gestos quirúrgicos. Para ello, se analizará cómo afecta dicha topología a la respuesta del sistema de reconocimiento y se seleccionará aquella que proporcione mejores resultados.

### 5.3.2. Gestos registrados

Para implantar el sistema de reconocimiento es necesario definir una serie de gestos quirúrgicos que permitan validar su correcto funcionamiento. Cabe destacar que en los gestos elegidos deben tenerse en cuenta parámetros físicos que sean diferenciadores entre un gesto y otro. Con esto, se evita que en el sistema de reconocimiento se puedan producir secuencias de etiquetas que puedan dar lugar a error al reconocer en una determinada grabación, debido a que dichas secuencias sean similares para un gesto y otro. En este caso, se han decidido modelar cuatro gestos, con los parámetros físicos y la funcionalidad inicial que se detalla a continuación:

#### Gesto 1

**Descripción:** Se introduce la mano izquierda de forma horizontal (con la palma mirando hacia abajo), y se produce un giro de unos 180 grados hacia la izquierda (alrededor del eje del brazo), de forma que por un instante la palma quede mirando hacia arriba. Posteriormente, la palma se gira hasta volver a la posición inicial del giro, quedando la palma hacia abajo, y se extrae la mano de forma horizontal. Dicho movimiento se puede observar esquemáticamente en la figura 5.6:



Figura 5.6. Esquema del movimiento producido durante el gesto 1.

#### **Parámetros físicos utilizados:**

P1. Ángulo entre los ejes z de sistema de referencia de la palma de la mano y sistema de referencia de Leap Motion.

P2. Ángulo entre el eje y del sistema de referencia de la punta del dedo índice y el eje x del sistema de referencia de Leap Motion.

P3. Ángulo entre el eje x de la muñeca y el eje y del sistema de referencia de Leap Motion.

**Funcionalidad prevista:** Comandar a un brazo robótico quirúrgico el desplazamiento de la óptica hacia la izquierda.

## Gesto 2

**Descripción:** Se introduce la mano izquierda de forma horizontal con la palma mirando hacia abajo, y se produce un giro de muñeca hacia la derecha de unos 30 grados, manteniendo en todo momento la palma mirando hacia abajo y en posición horizontal. Se vuelve a la posición inicial anterior al giro, y se vuelve a repetir el mismo movimiento, para identificar el gesto mediante dos "tics". Posteriormente, la palma vuelve a la posición inicial, y se extrae la mano. Dicho movimiento se puede observar esquemáticamente en la figura 5.7:

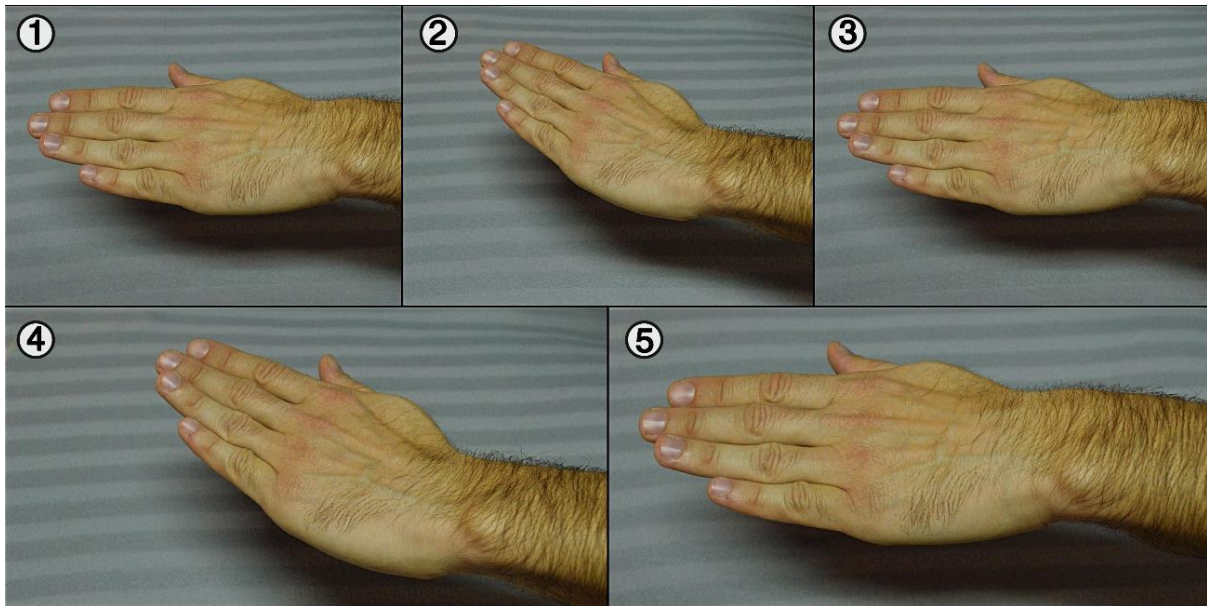


Figura 5.7. Esquema del movimiento producido durante el gesto 2.

### **Parámetros físicos utilizados:**

P1. Ángulo entre el eje x del sistema de referencia de la palma de la mano y el eje y del sistema de referencia de Leap Motion.

P2. Ángulo entre el eje z del sistema de referencia de la punta del dedo índice y el eje x del sistema de referencia de Leap Motion.

P3. Ángulo entre el eje z del sistema de referencia de la muñeca y el eje x del sistema de referencia de Leap Motion.

**Funcionalidad prevista:** Comandar a un brazo robótico quirúrgico el desplazamiento de la óptica hacia la derecha.

### **Gesto 3**

**Descripción:** Se introduce la mano izquierda de forma horizontal con la palma mirando hacia abajo y con los dedos índice y medio extendidos en forma de tijera abierta, mientras que los dedos meñique y anular se encuentran apoyados sobre la palma. Posteriormente, se cierran dedos índice y corazón simulando el cierre de una tijera. Se vuelven a abrir y a cerrar (para identificar el gesto con dos cierres). Posteriormente, se vuelven a abrir índice y corazón, y se extrae la mano. Dicho movimiento se puede observar esquemáticamente en la figura 5.8:

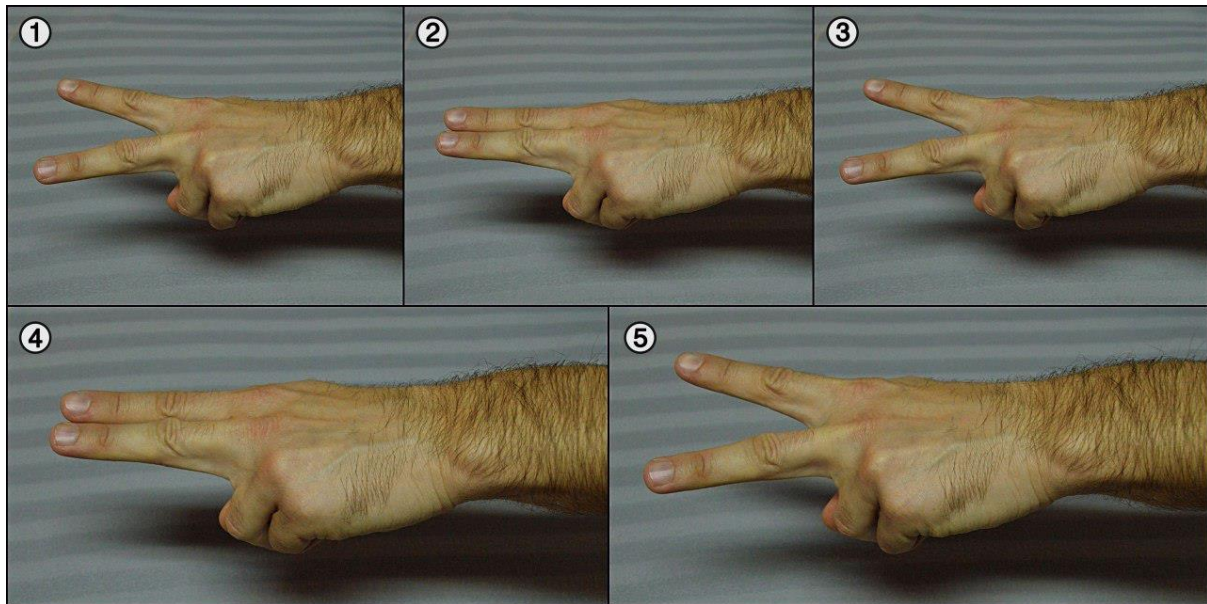


Figura 5.8. Esquema del movimiento producido durante el gesto 3.

#### **Parámetros físicos utilizados:**

P1. Distancia entre el sistema de referencia de la punta del dedo índice y el sistema de referencia de la punta del dedo corazón.

P2. Distancia entre el sistema de referencia medial del dedo índice y el sistema de referencia medial del dedo corazón.

P3. Ángulo entre el eje z del sistema de referencia de la punta del dedo índice y el eje z del sistema de referencia del dedo corazón.

**Funcionalidad prevista:** Comandar a un brazo robótico quirúrgico el inicio de la etapa de disección-coagulación.

## Gesto 4

**Descripción:** Se introduce la mano izquierda en forma de pinza abierta, con la palma en horizontal mirando hacia abajo. Posteriormente, se cierran los dedos índice, medio, anular y meñique sobre el pulgar y se abre hasta la posición inicial. Se vuelve a cerrar y a abrir la "pinza", y se extrae la mano en la misma postura que se introdujo al inicio del gesto. Dicho movimiento se puede observar esquemáticamente en la figura 5.9:

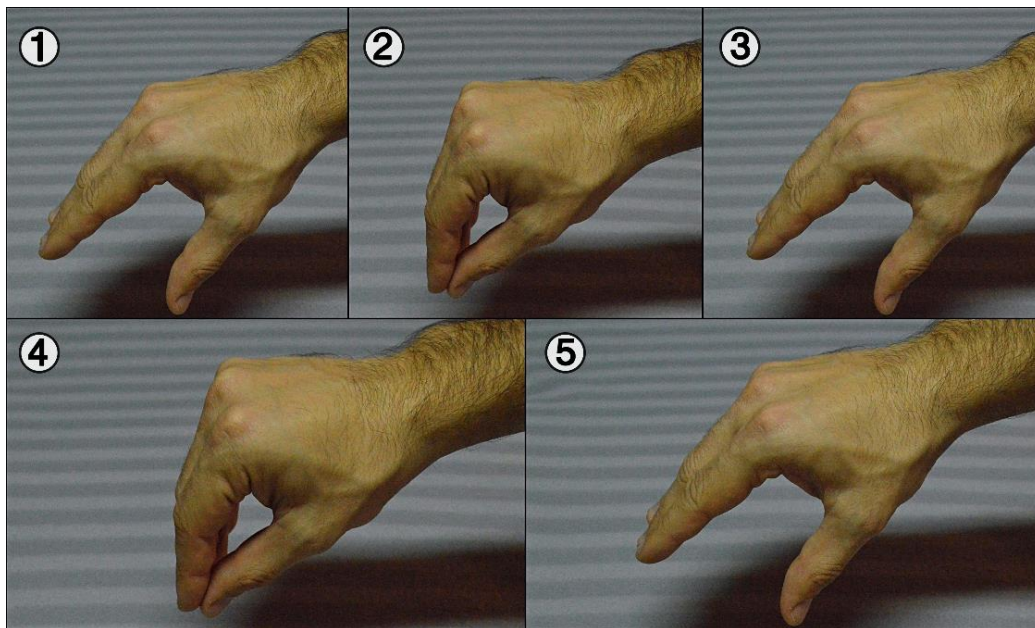


Figura 5.9. Esquema del movimiento producido durante el gesto 4.

### **Parámetros físicos utilizados:**

P1. Distancia entre el sistema de referencia de la punta del dedo índice y el sistema de referencia de la punta del dedo pulgar.

P2. Distancia entre el sistema de referencia de la punta del dedo corazón y el sistema de referencia de la punta del dedo pulgar.

P3. Distancia entre el sistema de referencia de la punta del dedo meñique y el sistema de referencia de la punta del dedo pulgar.

**Funcionalidad prevista:** Comandar a un brazo robótico quirúrgico la necesidad sostener ciertos órganos para facilitar la tarea del cirujano.

Por último, se puede ver la evolución media de los parámetros físicos elegidos para uno de los gestos en las siguientes figuras:

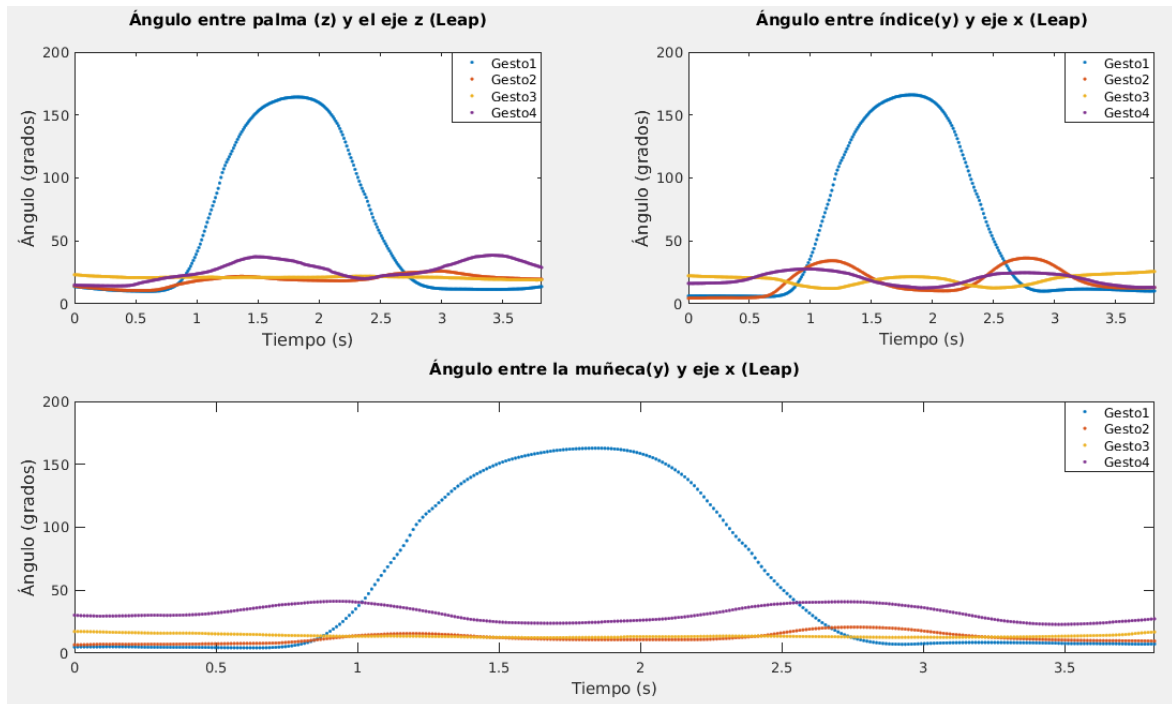


Figura 5.10. Evolución media de los parámetros físicos del gesto 1 para grabaciones de los 4 gestos elegidos.

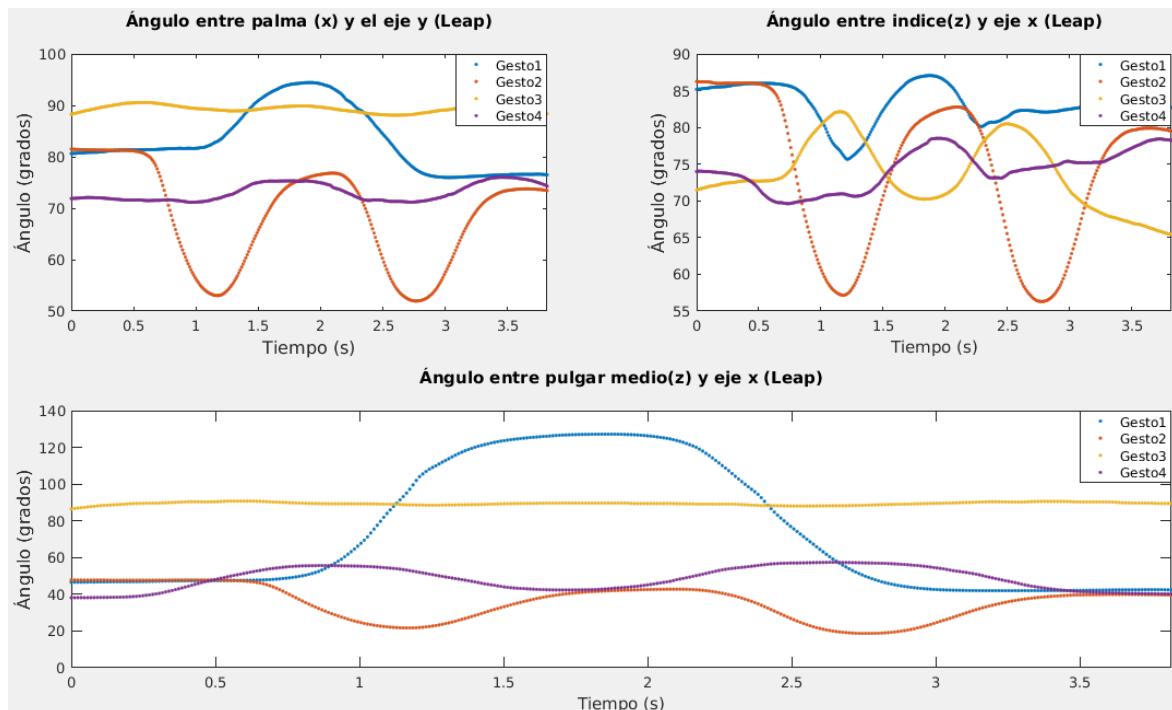


Figura 5.11. Evolución media de los parámetros físicos del gesto 2 para grabaciones de los 4 gestos elegidos.

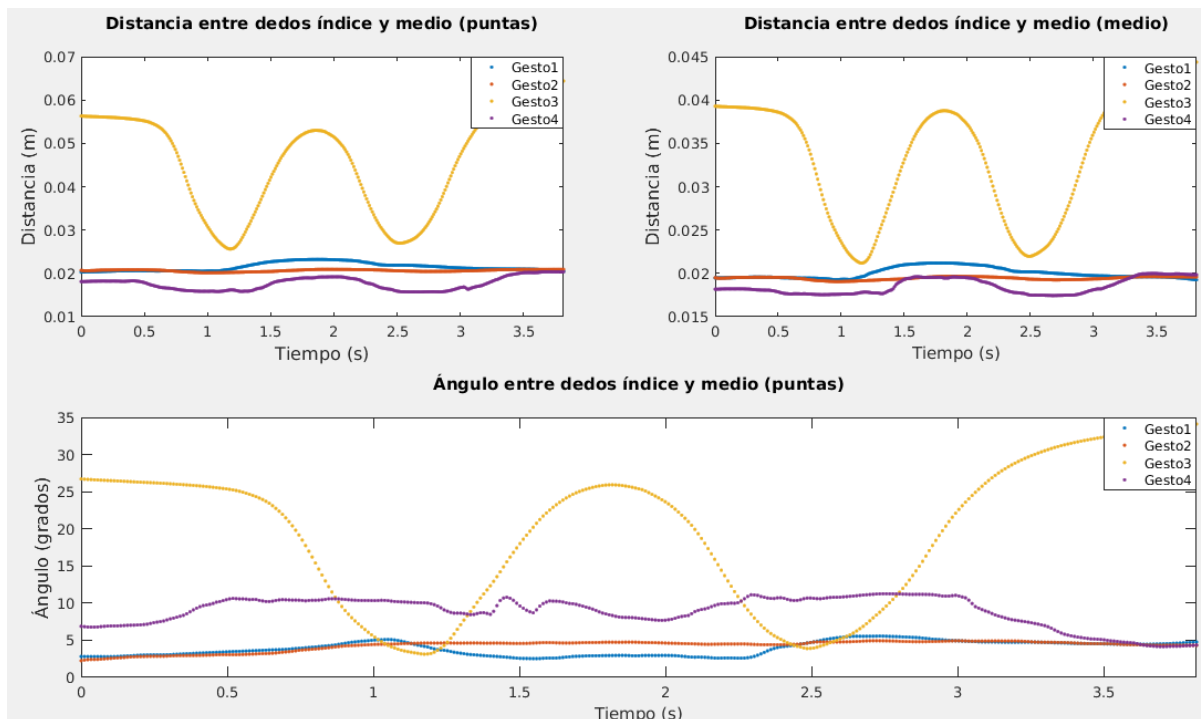


Figura 5.12. Evolución media de los parámetros físicos del gesto 3 para grabaciones de los 4 gestos elegidos.

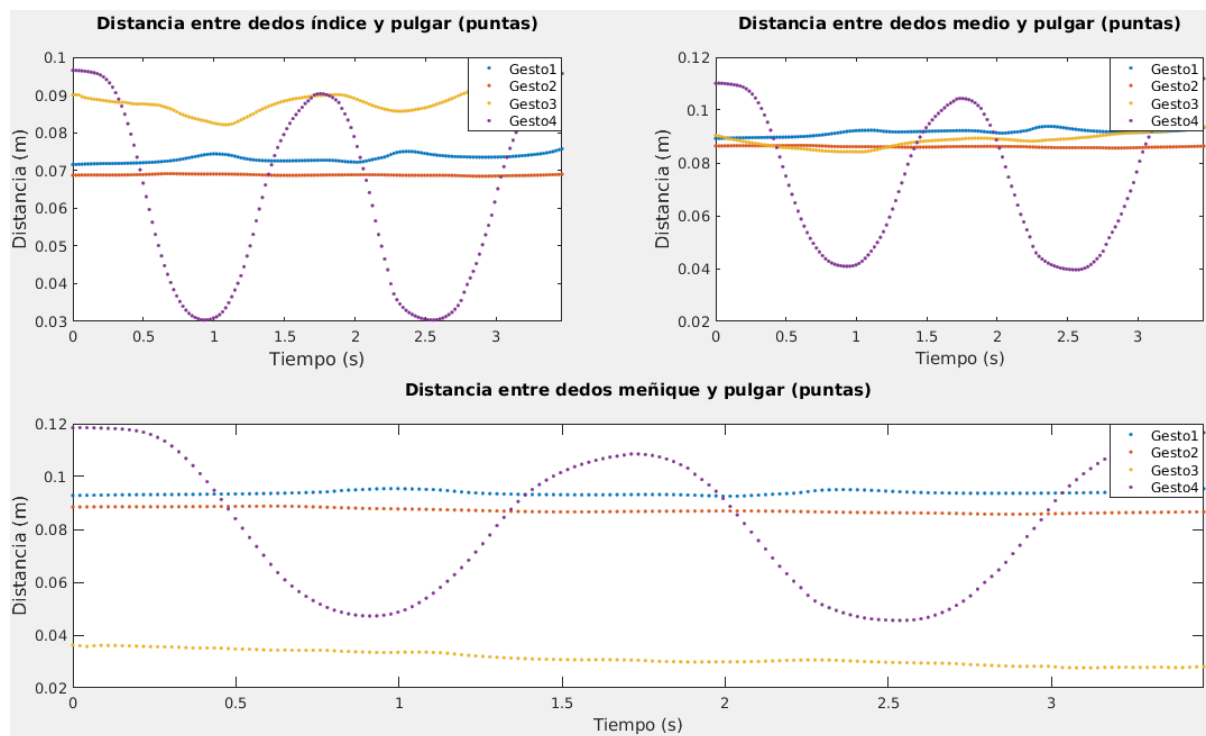


Figura 5.13. Evolución media de los parámetros físicos del gesto 4 para grabaciones de los 4 gestos elegidos.

Como se puede observar, la evolución de los parámetros físicos tiene una forma característica para el gesto en cuestión que se está analizando, y para el resto de gestos, dicha evolución difiere mucho, o incluso se mantiene constante en algunos casos, lo que demuestra que efectivamente los parámetros físicos elegidos permiten diferenciar a un gesto de otro. Gracias a esto, se simplifica mucho el trabajo del reconocedor, evitando que el sistema de reconocimiento produzca secuencias de características observables muy similares para diferentes gestos.

### **5.3.3. Batería de experimentos y resultados**

En este apartado se detalla la batería de experimentos utilizada para probar y validar cada uno de los módulos propuestos en el Capítulo 4 para el modelado y reconocimiento de gestos quirúrgicos.

Como ya se ha comentado en el apartado 5.1, hay que mencionar que en las intervenciones quirúrgicas se utilizaría un guante sensorizado para determinar la evolución de la mano durante los gestos, pero para la implementación del sistema desarrollado, dicho guante se ha sustituido por el dispositivo Leap Motion, que actúa como un sensor análogo.

Para observar el funcionamiento del sistema, se ha decidido experimentar con cuatro individuos diferentes. Cada uno de ellos ha realizado 50 grabaciones por cada gesto, y dentro de estas, 30 se han asignado para el sistema de entrenamiento y 20 para el sistema de reconocimiento propiamente dicho. De esta forma, las 30 primeras grabaciones realizadas permiten un ajuste personalizado de los Modelos Ocultos de Markov para cada usuario, mientras que las 20 grabaciones restantes pasarán por el sistema de reconocimiento y permitirán determinar el número de aciertos que proporciona el reconocedor.

Además, entre los individuos que han participado en el estudio, se encuentran varios niveles de destreza con dicho dispositivo, desde muy habituados a su uso hasta usuarios principiantes poco familiarizados con el entorno del Leap Motion, lo que nos permitirá determinar posteriormente si este factor es relevante en el porcentaje de aciertos.

Con respecto a las grabaciones utilizadas, es importante destacar que se requiere de unas condiciones de inicio y de fin de cada grabación, para detectar en qué intervalo de tiempo se está produciendo un gesto. En el caso del presente trabajo, se parte de que las condiciones de inicio y de fin vienen determinadas por la visibilidad de la mano izquierda.

A continuación se detallan los experimentos realizados especificando sus objetivos, las tareas realizadas para llevarlos a cabo, los datos registrados y los resultados obtenidos.



## Experimento 1

En primer lugar se llevará a cabo un experimento para determinar la topología inicial de las matrices de Markov que proporciona mejores resultados para los gestos incluidos en el sistema. Este se realizará con las grabaciones de un solo individuo, y posteriormente se utilizará la topología elegida con las grabaciones correspondientes a los otros individuos. Como se especificó en el apartado 4.5, hay diversas topologías iniciales para las matrices A de distribución de probabilidad de transición entre estados ( $q_i$ ). A continuación se comentan cada de una de las topologías estudiadas con sus resultados correspondientes.

**1. Ergódica.** En esta topología todos los estados se encuentran conectados entre sí, lo que posibilita que cada estado pueda alcanzarse desde cualquier otro tras un número finito de pasos. Esto permite transiciones tanto hacia adelante como hacia atrás. En la tabla 5.1 se pueden observar los resultados obtenidos:

Gesto	Grabaciones de Entrenamiento	Grabaciones de Reconocimiento	Número de Aciertos	% de Aciertos
1	30	20	20	100.00 %
2	30	20	20	100.00 %
3	30	20	14	70.00 %
4	30	20	17	85.00 %

Tabla 5.1. Resultados del reconocimiento con topología ergódica.

**2. Triangular superior.** Con esta forma, únicamente se permite avanzar en el diagrama de estados, sin posibilidad de retroceder a estados anteriores. En la tabla 5.2 se pueden observar los resultados obtenidos:

Gesto	Grabaciones de Entrenamiento	Grabaciones de Reconocimiento	Número de Aciertos	% de Aciertos
1	30	20	20	100.00 %
2	30	20	18	90.00 %
3	30	20	17	85.00 %
4	30	20	18	90.00 %

Tabla 5.2. Resultados del reconocimiento con topología triangular superior.

**3. De izquierda a derecha.** En esta topología, se permite únicamente las transiciones de un estado hacia sí mismo o hacia el siguiente más contiguo. En la tabla 5.3 se pueden observar los resultados obtenidos:

Gesto	Grabaciones de Entrenamiento	Grabaciones de Reconocimiento	Número de Aciertos	% de Aciertos
1	30	20	20	100.00 %
2	30	20	18	90.00 %
3	30	20	16	80.00 %
4	30	20	18	90.00 %

Tabla 5.3. Resultados del reconocimiento con topología de izquierda a derecha.

**4. Tridiagonal superior e inferior.** Con esta forma se permite la transición de un estado hacia sí mismo, así como las transiciones de retroceso o avance en el diagrama a los estados más contiguos. En la tabla 5.4 se pueden observar los resultados obtenidos:

Gesto	Grabaciones de Entrenamiento	Grabaciones de Reconocimiento	Número de Aciertos	% de Aciertos
1	30	20	20	100.00 %
2	30	20	18	90.00 %
3	30	20	16	80.00 %
4	30	20	18	90.00 %

Tabla 5.4. Resultados del reconocimiento con topología triangular superior e inferior.

Como se puede observar, no hay una diferencia notable al utilizar una u otra topología, por lo que se podría elegir cualquiera de ellas. En este caso, se ha considerado la estructura de la matriz A triangular superior en la que sólo se permite avanzar en el diagrama de estados como la configuración a utilizar para llevar a cabo el resto de experimentos, debido a que proporciona un porcentaje de aciertos algo superior al resto de topologías.

## Experimento 2

En este experimento se desea evaluar la validez del sistema de gestos desarrollado a lo largo de este trabajo. Para ello, se va a comprobar empíricamente los resultados que proporciona el reconocedor para cuatro individuos diferentes.

Por cada sujeto se han realizado un total de 200 grabaciones (50 repeticiones por cada gesto), de las cuales 30 se han asignado al sistema de entrenamiento, permitiendo el ajuste de las matrices que caracterizan a los Modelos Ocultos de Markov. Dicho entrenamiento ha permitido generar las librerías de gestos a partir de la topología triangular superior especificada en el apartado 4.5, tras comprobar que proporcionaba resultados ligeramente superiores al resto de topologías. Por otro lado, las 20 grabaciones restantes se han asignado al sistema de reconocimiento.

Con respecto a los parámetros físicos, se ha decidido tomar 3 por cada gesto (como se detalla en el apartado 5.3.2), y a su vez 3 centroides por cada parámetro, de forma que se permiten un total 27 características observables diferentes por gesto. Además, se han realizado ensayos experimentales de prueba-error para determinar el número de estados más apropiado para las redes de Markov, fijando finalmente 5 estados por cada gesto. Experimentalmente, se identificó que un número superior de estados complicaba el reconocimiento, ya que disminuían los valores de probabilidad de cada gesto, y que un menor número de estados proporcionaba menor número de aciertos en el reconocimiento, ya que dificultaba la correcta elección del gesto que se había producido.

A continuación se pueden observar los resultados obtenidos para los diferentes usuarios. Cabe destacar que los dos primeros se encontraban habituados al uso de Leap Motion, mientras que los dos últimos individuos apenas habían usado el dispositivo.

**Individuo 1.** En la tabla 5.5 se pueden observar los resultados obtenidos:

Gesto	Grabaciones de Entrenamiento	Grabaciones de Reconocimiento	Número de Aciertos	% de Aciertos
1	30	20	18	90.00 %
2	30	20	19	95.00 %
3	30	20	18	90.00 %
4	30	20	19	95.00 %

Tabla 5.5. Resultados del reconocimiento del individuo 1.

**Individuo 2.** En la tabla 5.6 se pueden observar los resultados obtenidos:

<b>Gesto</b>	<b>Grabaciones de Entrenamiento</b>	<b>Grabaciones de Reconocimiento</b>	<b>Número de Aciertos</b>	<b>% de Aciertos</b>
<b>1</b>	30	20	20	100.00 %
<b>2</b>	30	20	17	85.00 %
<b>3</b>	30	20	17	85.00 %
<b>4</b>	30	20	16	80.00 %

Tabla 5.6. Resultados del reconocimiento del individuo 2.

**Individuo 3.** En la tabla 5.7 se pueden observar los resultados obtenidos:

<b>Gesto</b>	<b>Grabaciones de Entrenamiento</b>	<b>Grabaciones de Reconocimiento</b>	<b>Número de Aciertos</b>	<b>% de Aciertos</b>
<b>1</b>	30	20	20	100.00 %
<b>2</b>	30	20	11	55.00 %
<b>3</b>	30	20	16	80.00 %
<b>4</b>	30	20	15	75.00 %

Tabla 5.7. Resultados del reconocimiento del individuo 3.

**Individuo 4.** En la tabla 5.8 se pueden observar los resultados obtenidos:

<b>Gesto</b>	<b>Grabaciones de Entrenamiento</b>	<b>Grabaciones de Reconocimiento</b>	<b>Número de Aciertos</b>	<b>% de Aciertos</b>
<b>1</b>	30	20	20	100.00 %
<b>2</b>	30	20	11	55.00 %
<b>3</b>	30	20	15	75.00 %
<b>4</b>	30	20	10	50.00 %

Tabla 5.8. Resultados del reconocimiento del individuo 4.

### **5.3.4. Discusión**

El objetivo de estos experimentos ha sido comprobar el funcionamiento de la interfaz basada en el reconocimiento de gestos durante las intervenciones de cirugía laparoscópica asistida por la mano. Tras analizar los resultados obtenidos, se puede llegar a la conclusión de que los algoritmos desarrollados son válidos. En resumen, se ha obtenido un 97.5% de acierto del gesto 1, un 72.5% de acierto del gesto 2, un 82.5% de acierto del gesto 3, y 75% de acierto para el gesto 4, con lo que se ha alcanzado un porcentaje de acierto global del 82%.

Como se puede observar, los resultados varían en función del individuo que realiza las grabaciones de los gestos. De esta forma, se ha podido ver cómo el porcentaje de aciertos disminuye de forma notable para los gestos 2 y 4 en aquellos usuarios que disponen de un menor nivel de destreza con Leap Motion. Esto demuestra que realmente afecta lo familiarizado que esté el sujeto con el dispositivo, y que puede ser un factor relevante si se desea mejorar el reconocimiento de gestos.

Por otro lado, cabe destacar que Leap Motion almacena los sistemas de referencia de la mano a partir de imágenes, como se comentó en el apartado 5.2.1, por lo que si en un momento dado unos dedos se ocultan con otros, la medición puede ser imprecisa. Así, un usuario que no esté informado o suficientemente familiarizado con este hecho puede realizar grabaciones que no se correspondan exactamente con el movimiento que realmente está realizando. Por tanto, se recomienda que antes de utilizar el sistema, el usuario haga pruebas en las que observe como se visualiza su mano al realizar diferentes movimientos.

## **5.4. Conclusiones**

En este capítulo se ha detallado en primer lugar el entorno propuesto para implantar el sistema, incluyendo tanto las arquitecturas hardware como software que se han utilizado, así como una breve descripción de cada uno de los componentes que las constituyen.

Por otro lado, se han verificado los desarrollos teóricos expresados en los capítulos anteriores. Para ello, se han realizado dos experimentos, el primero con el objetivo de determinar la topología inicial más adecuada a utilizar para los Modelos Ocultos de Markov, y el segundo con el fin de validar el funcionamiento del sistema de reconocimiento a través de ensayos experimentales realizados por diferentes usuarios, permitiendo comprobar que éste funciona de forma satisfactoria.



# Capítulo 6: Conclusiones y trabajos futuros

## 6.1. Conclusiones

A lo largo de la memoria se han ido exponiendo las tareas realizadas durante el desarrollo de este trabajo fin de grado, mostrando cómo se han alcanzado todos los objetivos expuestos en el anteproyecto.

En primer lugar se expuso una introducción a las técnicas de cirugía mínimamente invasiva, realizando especial hincapié en aquellas relacionadas con la cirugía laparoscópica asistida por la mano (HALS). Tras esto, se comentaron algunos sistemas de ayuda robotizada, especialmente interfaces persona-máquina que actuasen en función de órdenes del cirujano, destacando las aportaciones del presente trabajo en éste ámbito.

En segundo lugar, se realizó una descripción ideal del entorno quirúrgico utilizado para intervenciones HALS, a la par que se comentó la problemática existente ante la falta de interfaces quirúrgicas en estos procedimientos. Posteriormente, se propuso el desarrollo de una interfaz de comunicación natural entre cirujano y asistente robótico, detallando las hipótesis y simplificaciones llevadas a cabo para realizar el trabajo, así como la arquitectura utilizada para resolver el problema.

En tercer lugar, se comentaron paso a paso los algoritmos implementados para el desarrollo de la interfaz basada en el reconocimiento de gestos. Finalmente se ha dedicado un capítulo a dos objetivos fundamentales: por un lado, comentar el entorno hardware y software en el que se implantó el sistema y su integración en una arquitectura ROS, y por otro lado, a realizar una serie de experimentos para validar el correcto funcionamiento del sistema de reconocimiento desarrollado.

## 6.2. Futuras líneas de desarrollo

Pese a que el reconocedor desarrollado funciona con un porcentaje de acierto elevado para aquellos usuarios familiarizados con Leap Motion, se pueden encontrar en éste diversos problemas que pueden plantearse como líneas de desarrollo futuro para la mejora del sistema:

-En primer lugar, la necesidad de mejorar el sistema de adquisición de datos, ya que la mayor parte de dificultades que han ido surgiendo derivan de una detección imprecisa de la posición de la mano. Ésta problemática espera resolverse en el futuro (al menos parcialmente) mediante la sustitución de Leap Motion por un guante quirúrgico sensorizado.

-El segundo problema está relacionado con la elección de los parámetros físicos que caracterizarán a cada gesto. Durante la ejecución del presente trabajo, estos parámetros se han elegido manualmente por cada gesto, según la evolución cinemática esperada de la mano. Sin embargo, el desarrollo de un algoritmo que automatizase esta tarea podría simplificar bastante el proceso, además de que permitiría elegir parámetros que quizás fuesen aún más diferenciadores entre un gesto y otro. Asimismo, se podrían desarrollar algoritmos para automatizar el número de estados óptimo o para determinar la topología más idónea a utilizar para los Modelos Ocultos de Markov.

-En tercer lugar, el sistema se ha implementado comprobando su funcionamiento a partir del almacenamiento de una grabación, y no directamente en línea como se esperaba. Esto se debe principalmente a que el sistema se implantó en Matlab, y el tiempo de procesado era elevado como para reconocer una grabación online (el gesto se reconocería con un retardo importante). Para solucionar esto, se plantea extrapolar el sistema desarrollado a C++, lenguaje de programación en el que la ejecución del código desarrollado se produciría de forma mucho más rápida.

-Por último, se plantea la necesidad de proponer unas condiciones de inicio y fin de los gestos que, a diferencia de las usadas en el sistema actual (de visibilidad de la mano con el dispositivo Leap Motion), sean extrapolables al entorno quirúrgico en el que se pretende usar un guante quirúrgico sensorizado.



# A. Algoritmo K-means

## A.1. Introducción

Este anexo tiene por objeto presentar el algoritmo de agrupamiento que ha servido como base para clasificar en diferentes etiquetas las muestras de distancias y ángulos registradas entre sistemas de referencia de la mano.

## A.2. Descripción del algoritmo

El algoritmo K-means tiene como objetivo la partición de un conjunto de  $n$  observaciones en  $k$  grupos o clusters, de forma que cada observación se incluye dentro del grupo cuyo valor medio es más cercano. Este ha sido ampliamente utilizado en áreas relacionadas con la minería de datos, tales como visión por computador o astronomía.

El funcionamiento del algoritmo es el siguiente: dado un conjunto de observaciones  $(x_1, x_2, x_3, \dots, x_n)$ , dicho algoritmo construye una partición de las observaciones en  $k$  clusters  $S = \{S_1, S_2, \dots, S_k\}$ , con  $k \leq n$ , a fin de minimizar la suma de los cuadrados dentro de cada grupo, esto es, la suma de las funciones de distancia entre cada muestra de un conjunto y su centroide (el centro geométrico de dicho conjunto de datos). En otras palabras, su objetivo es encontrar:

$$\arg \min_{\mathbf{s}} \sum_{i=1}^k \sum_{x \in S_i} \|x - \mu_i\|^2 \quad (A.1)$$

Donde  $\mu_i$  es la media de las muestras incluidas en  $S_i$ .

## A.3. Implementación del algoritmo

Hay diversas implementaciones de este algoritmo. El estándar está basado en una técnica de refinamiento iterativo, en la cual se lleva a cabo el siguiente procedimiento:

1) Se sitúan  $K$  valores en el espacio, que representan los centroides iniciales de los grupos.

$$m_1^{(1)}, m_2^{(1)}, \dots, m_k^{(1)} \quad (A.2)$$

Tras esto, se repiten de forma iterativa los pasos 2 y 3 que se detallan a continuación:

2) Cada componente del vector de observaciones se asigna al grupo que tiene el centroide más cercano, basándose en la distancia euclídea:

$$S_i^{(t)} = \{x_p : \|x_p - m_i^{(t)}\| \leq \|x_p - m_j^{(t)}\| \forall 1 \leq j \leq k\} \quad (A.3)$$

De esta forma, cada observación  $x_p$  va exactamente dentro de un solo grupo  $S_i^{(t)}$ , aunque pudiese ir en dos de ellos por igualdad de distancias a sus correspondientes centroides.

3) Una vez clasificadas todas las muestras del vector, se vuelven a recalcular las posiciones de los centroides mediante la siguiente expresión:

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} * \sum_{x_j \in S_i^{(t)}} x_j \quad (A.4)$$

Esta iteración continúa hasta que el valor de los centroides se mantiene estable, es decir, hasta el momento en el que el algoritmo converge. Debido a que se trata de un algoritmo heurístico, no hay ninguna garantía de que se converja al óptimo global, y el resultado puede depender de los grupos iniciales. Para proceder a dicha inicialización, es posible encontrar varios métodos, entre los que destacan los siguientes:

**-Método de Forgy:** En este caso, se eligen aleatoriamente  $k$  observaciones del conjunto de datos y se utilizan como centroides iniciales. Éste tiende a dispersar los centroides iniciales.

**-Método de partición aleatoria:** Este asigna inicialmente un clúster a cada observación y después se procede a la etapa de actualización, de forma que el primer centroide de cada clúster viene dado por la media inicial de los valores que le son asignados aleatoriamente al comienzo del proceso. Éste método tiende a ubicar los centroides cerca del centro del conjunto de datos.

# B. Modelos Ocultos de Markov

## B.1. Introducción

Los Modelos Ocultos de Markov (Hidden Markov Models o HMM en inglés) son modelos con una estructura matemática muy rica, que ofrecen bases teóricas en un amplio rango de aplicaciones, y que pueden aplicarse a diversos problemas como pueden ser el reconocimiento del habla o el reconocimiento de maniobras quirúrgicas, gracias a la caracterización de procesos reales en términos de modelos de señales.

Para comprender su funcionamiento se ha decidido incluir este anexo, con el que se presentan los principales aspectos teóricos del modelado estadístico utilizado para desarrollar el sistema de reconocimiento de gestos de la mano del cirujano.

## B.2. Descripción del modelo

Un HMM es aquel modelo estocástico, es decir, no determinista, formado por un número finitos de  $N$  estados  $q_i$ , cada uno de los cuales representa una acción básica y posee unas características observables  $e_i$ .

Estos HMM vienen representados por una red en la que en cada intervalo de tiempo se avanza a un nuevo estado, emitiendo una señal de salida observable. La topología de la red viene definida por una matriz de distribución de probabilidad de salto entre estados, que establece la relación existente entre las diferentes acciones básicas. De forma genérica, el HMM de una maniobra viene caracterizado por la siguiente tupla de parámetros:

$$\lambda = (Q, O, A, B, \pi) \tag{B.1}$$

Donde:

- $Q = \{q_1, q_2, \dots, q_N\}$  es el conjunto de estados, donde  $N$  es el número de estados o acciones básicas durante un gesto o maniobra.
- $O = \{o_1, o_2, \dots, o_M\}$  es el conjunto de características observables cuando el sistema alcanza un estado, y donde  $M$  representa el número de salidas observables en cada estado.
- $A = \{a_{ij}\}$  es una matriz de probabilidad de transición entre estados, de forma  $a_{ij}$  es la probabilidad de pasar del estado  $i$  al  $j$ , teniendo que  $1 \leq i, j \leq N$ . De esta forma, para

todo par de estados  $(i, j)$  conectados entre sí directamente, se tiene que  $a_{ij} > 0$ . Por otro lado, por cada estado  $i$  debe de cumplirse la siguiente condición:

$$\sum_{j=1}^N a_{ij} = 1 \quad (B.2)$$

- $B = \{b_j(o_k)\}$  es una matriz de distribución de probabilidad de características observables en cada estado. De esta forma,  $b_j(o_k)$  indica la probabilidad de que aparezca la característica observable  $o_k$  cuando se está en el estado  $j$ , con  $1 \leq k \leq M$ .
- $\pi = \{\pi_i\}$ , donde  $\pi_i$  es la distribución de probabilidad inicial de estados, con  $1 \leq i \leq N$ .

A continuación se puede observar en la figura B.1 una red HMM constituida por 5 estados, en la que se muestra la relación existente entre dichos estados, así como su relación con algunas características observables:

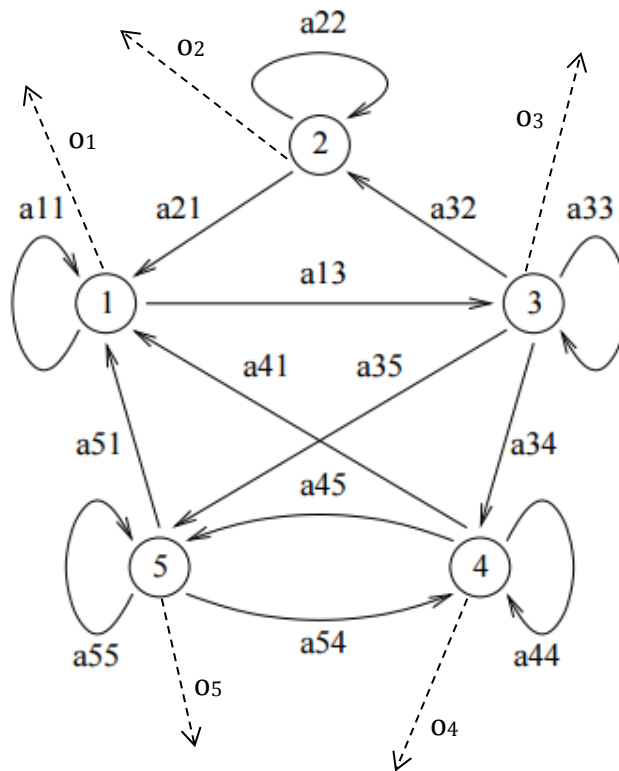


Figura B.1. Ejemplo de HMM de 5 estados.

En este caso solo aparece una observación por estado, y no todos los estados se encuentran conectados entre sí; así por ejemplo, del estado 2 sólo se puede saltar a sí mismo o al primer estado.

Una vez definida la topología de red y expuesto un ejemplo ilustrativo de la misma, se pueden destacar tres problemas básicos a resolver mediante los HMM:

- **El problema de la evaluación:** Dado el modelo  $\lambda = (A, B, \pi)$  y una secuencia de observaciones  $S$ , calcular  $P(S | \lambda)$ . Este problema se resuelve mediante el algoritmo de Forward-Backward.
- **El problema de la inferencia:** Dado el modelo  $\lambda = (A, B, \pi)$  y la secuencia de observaciones  $S$ , encontrar la secuencia de estados correspondientes  $Q$  que mejor explique las observaciones. Este problema se puede resolver mediante el algoritmo de Viterbi.
- **El problema del aprendizaje o entrenamiento:** Dada una secuencia de observaciones  $S$ , encontrar los parámetros del modelo  $\lambda = (A, B, \pi)$  que maximicen  $P(S | \lambda)$ . Esto se consigue mediante el algoritmo de Baum-Welch.

Para el desarrollo de este trabajo ha sido necesario resolver tanto el problema de aprendizaje (para el sistema de entrenamiento) como el problema de evaluación (para el reconocimiento de gestos de la mano), para lo cual se han utilizado funciones de Matlab ya definidas. Pese a ello, se incluyen a continuación los fundamentos teóricos y operaciones necesarias para desarrollar los algoritmos utilizados para su resolución.

### B.3. Algoritmo Forward-Backward

El problema de la evaluación consiste en calcular la probabilidad condicionada  $P(S_O | \lambda)$  de que se produzca la secuencia de observaciones  $S_O = (o_1, o_2, \dots, o_T)$ , dado el modelo  $\lambda = (A, B, \pi)$ .

En primer lugar, supongamos una secuencia de estados  $S_Q = \{q_1, q_2, \dots, q_T\}$ . La probabilidad de que se produzca esta secuencia es:

$$P(S_Q | \lambda) = \pi_1 * a_{12} * a_{23} \dots a_{T-1T} \quad (B.3)$$

Por otro lado, la probabilidad de obtener una secuencia de observaciones  $S_O = (o_1, o_2, \dots, o_T)$  cuando se da precisamente esta secuencia de estados  $S_Q$  es:

$$P(S_O | S_Q, \lambda) = \prod_{t=1}^T P(o_t | q_t, \lambda) \quad (B.4)$$

Donde, suponiendo que las observaciones son independientes unas de otras, se tiene que la probabilidad  $P(o_t | q_t, \lambda)$  se corresponde con el valor de  $b_t(o_t)$ , es decir, a la probabilidad de que en el estado  $t$  se produzca la característica observable  $t$ . De esta forma la expresión anterior puede reescribirse de la siguiente manera:

$$P(S_O | S_Q, \lambda) = \prod_{t=1}^T b_t(o_t) = b_1(o_1) * b_2(o_2) \dots b_T(o_T) \quad (B.5)$$

Por último para obtener la probabilidad de que se produzca una secuencia de características observables  $S_O$  dado un modelo  $\lambda$  es necesario calcular la probabilidad de  $S_O$  para cada una de las posibles secuencias de estados  $S_Q$ :

$$P(S_O | \lambda) = \sum^{S_Q} P(S_Q | \lambda) * P(S_O | S_Q, \lambda) \quad (B.6)$$

Que puede reescribirse como:

$$P(S_O | \lambda) = \sum^{S_Q} [\pi_1 * b_1(o_1) * a_{12} * b_2(o_2) * a_{23} \dots b_T(o_T) * a_{T-1T}] \quad (B.7)$$

Para realizar este cálculo, se emplean estrategias de programación dinámica, lo que permite reducir la complejidad del mismo, tales como el algoritmo de Forward-Backward, que es más eficiente que realizarlo mediante fuerza bruta. Con este algoritmo, se calculan dos variables intermedias:

- La variable forward  $\alpha_t(i)$ , que es la probabilidad de obtener la secuencia de observaciones parciales  $\{o_1, o_2, \dots, o_t\}$  en el instante de tiempo  $t$  y en el estado  $q_i$  para el modelo  $\lambda$ .
- La variable backward  $\beta_t(i)$ , que es la probabilidad de que se dé la secuencia parcial de observaciones desde el instante de tiempo  $t+1$  hasta el final, cuando el estado en el instante de tiempo  $t$  es  $i$ , y dado el modelo  $\lambda$ .

Estas variables se pueden calcular mediante las siguientes expresiones:

$$\alpha_t(i) = P(o_1 o_2 \dots o_t, q_t = q_i | \lambda) \quad (B.8)$$

$$\beta_t(i) = P(o_{t+1} o_{t+2} \dots o_T, q_t = q_i | \lambda) \quad (B.9)$$

### Cálculo de la variable forward

Para el cálculo de esta variable  $\alpha_t(i)$ , se requiere de los siguientes pasos: inicialización (B.10), recurrencia (B.11) y terminación (B.12).

$$\alpha_1(i) = \pi_i * b_i(o_1) \quad 1 \leq i \leq N \quad (B.10)$$

$$\alpha_{t+1}(j) = \alpha_t(i) * a_{ij} * b_j(o_{t+1}) \quad t = 1, 2, \dots, T-1 \quad 1 \leq j \leq N \quad (B.11)$$

$$P(S_O | \lambda) = \sum_{i=1}^N \alpha_T(i) \quad (B.12)$$

### Cálculo de la variable backward

Para el cálculo de esta variable  $\beta_t(i)$ , se requiere de los mismos pasos que en el apartado anterior: inicialización (B.13), recurrencia (B.14) y terminación (B.15).

$$\beta_T(i) = 1 \quad 1 \leq i \leq N \quad (B.13)$$

$$\beta_t(i) = \sum_{j=1}^N a_{ij} * \beta_{t+1}(j) * b_j(o_{t+1}) \quad t = T-1, T-2, \dots, 1 \quad 1 \leq i \leq N \quad (B.14)$$

$$P(S_O | \lambda) = \sum_{i=1}^N \beta_1(i) * \pi_i * b_i(o_1) \quad (B.15)$$

## B.4. Algoritmo de Baum-Welch

Este algoritmo se utiliza en la fase de entrenamiento off-line de los modelos ocultos de Markov, permitiendo encontrar el modelo  $\lambda$  que maximiza la probabilidad de una secuencia de observaciones  $S_O = \{o_1, o_2, \dots, o_T\}$ , o dicho de otra forma, determinando el modelo que mejor explica dicha secuencia.

Dicho modelo no se puede obtener analíticamente, por lo que es necesario un algoritmo iterativo como el de Baum-Welch, que permita estimar los parámetros que hacen máxima la probabilidad de una secuencia de características observables. Dicho algoritmo proporciona un modelo  $\lambda = (A, B, \pi)$  usando un procedimiento iterativo de ajuste a partir de parámetros  $(A, B, \pi)$  iniciales.

Para definir el proceso de reestimación de parámetros de HMM, es necesario conocer:

- El número esperado de transiciones desde el estado  $i$  en  $S_O$ .
- El número esperado de transiciones desde el estado  $i$  al estado  $j$ .

Para ello, se define una variable intermedia  $\xi_t(i, j)$ , que es la probabilidad de estar en el estado  $i$  en el instante  $t$  y en el estado  $j$  en el instante  $t+1$ , dado el modelo  $\lambda = (A, B, \pi)$  y la secuencia de observaciones  $S_O = \{o_1, o_2, \dots, o_T\}$ :

$$\xi_t(i, j) = P(q_t = q_i, q_{t+1} = q_j \mid \lambda, S_O) \quad (B.16)$$

Esta expresión (B.16) puede reescribirse de la siguiente manera:

$$\xi_t(i, j) = \frac{\alpha_t(i) * a_{ij} * b_j(o_{t+1}) * \beta_{t+1}(j)}{P(S_O \mid \lambda)} = \frac{\alpha_t(i) * a_{ij} * b_j(o_{t+1}) * \beta_{t+1}(j)}{\sum_{i=1}^N \sum_{j=1}^N \alpha_t(i) * a_{ij} * b_j(o_{t+1}) * \beta_{t+1}(j)} \quad (B.17)$$

Donde los valores  $\alpha_t(i)$  y  $\beta_t(i)$  se pueden calcular mediante el algoritmo de forward-backward.

Por otro lado, se define la probabilidad de estar en el estado  $i$  en el instante  $t$  como:

$$\gamma_t(i) = \sum_{j=1}^N \xi_t(i, j) \quad (B.18)$$

Sumando cada  $\gamma_t(i)$  en cada instante de tiempo  $t$ , se obtiene una cantidad que puede interpretarse como el número esperado de transiciones realizadas desde el estado  $q_i$ , tal y como se muestra en la ecuación B.19:

$$\sum_{i=1}^{T-1} \gamma_t(i) \quad (B.19)$$

De forma similar, la suma de  $\xi_t(i, j)$  en todo instante de tiempo se interpreta como el número esperado de transiciones desde el estado  $q_i$  hasta el  $q_j$ .

$$\sum_{i=1}^{T-1} \xi_t(i, j) \quad (B.20)$$

Con el uso de estas fórmulas, se obtiene un método de reestimación de los parámetros de los HMM, de forma que  $A$ ,  $B$  y  $\pi$  se pueden obtener a partir de las ecuaciones (B.21) (B.22) y (B.23) de forma iterativa:

$$\bar{\pi}_i = \gamma_1(i) \quad (B.21)$$



$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \quad (B.22)$$

$$\bar{b}_j = \frac{\sum_{t=1}^T \gamma_t(j)}{\sum_{t=1}^T \gamma_t(j)} \quad (B.23)$$

Este proceso de entrenamiento se continúa hasta que no exista mejora entre un modelo y el siguiente revisado. Además, hay que tener en cuenta que en cada iteración del proceso se deben de cumplir las restricciones estocásticas de los parámetros del HMM que aparecen a continuación:

$$\sum_{j=1}^N \bar{\pi}_i = 1 \quad (B.24)$$

$$\sum_{j=1}^N \bar{a}_{ij} = 1 \quad 1 \leq i \leq N \quad (B.25)$$

$$\sum_{j=1}^M \bar{b}_j(k) = 1 \quad 1 \leq j \leq N \quad (B.26)$$



# Glosario de términos

**Acción básica ( $q_i$ ):** Interacción entre sistemas de referencia de la mano no diestra del cirujano en un instante determinado de un gesto quirúrgico.

**Característica observable ( $o_i$ ):** Parámetro que caracteriza la interacción entre varios sistemas de referencia de la mano y que contiene información extraída de la localización relativa de dichos sistemas de referencia. Estas se obtienen a partir del codificado de varios parámetros físicos ( $P_i$ ) en un único valor discreto o etiqueta.

**Centroide ( $c_i$ ):** Centro geométrico de una agrupación de datos o clúster.

**Cirugía mínimamente invasiva (CMI):** Es toda aquella técnica o procedimiento quirúrgico que trata de minimizar las incisiones que se practican sobre el paciente, con objeto de mejorar el tiempo de recuperación y disminuir las complicaciones en el post-operatorio. A través de estas incisiones se insertan instrumentos especiales, denominados de caña larga, y una óptica, con el objetivo de ejecutar la intervención quirúrgica.

**Cirugía laparoscópica asistida por la mano (HALS):** La cirugía laparoscópica asistida por la mano (del inglés hand-assisted laparoscopic surgery) es una técnica quirúrgica mínimamente invasiva en la cual el cirujano inserta una mano dentro de la cavidad abdominal del paciente a través de una incisión especial que se le realiza al mismo, lo que le permite manipular órganos, mientras que con la otra mano utiliza un instrumental laparoscópico mínimamente invasivo.

**Fulcro:** Punto de inserción del instrumental laparoscópico practicado en el abdomen del paciente para acceder al interior del mismo.

**Gesto:** Patrón de movimiento de las manos que puede dividirse en acciones básicas y que se puede identificar como una secuencia determinada de estas.

**Herramienta laparoscópica:** Utensilio de gran longitud y pequeño diámetro que permite asistir al cirujano durante una intervención de CMI. Hay diversas herramientas según la acción deseada: pinzas, tijeras, porta-agujas, bisturí eléctrico, insufladores, etc.

**Parámetro físico ( $P_i$ ):** Variable que permite relacionar dos sistemas de referencia situados en la mano mediante una magnitud física, como puede ser la distancia o el ángulo entre dos dedos.

**Robot colaborativo co-worker:** Se dice de todo aquel asistente robótico que dispone de la capacidad de trabajar codo con codo con humanos, gracias a las prestaciones sensoriales y de actuación con las que ha sido dotado.

**Trocar:** Elemento que se coloca en el fulcro para facilitar la inserción de la herramienta laparoscópica, y que permite disminuir las lesiones debidas al rozamiento con la pared abdominal. Además, posee una válvula que impide el escape del dióxido de carbono que se introduce en la cavidad abdominal para aumentar el espacio de trabajo.

**Vector característico (VC):** Combinación de variables discretas que permiten, en un instante de muestreo de la grabación de un gesto, almacenar los códigos asociados a cada uno de los centroides que se corresponden con las muestras de los parámetros físicos.

# Referencias

- [1] Bauzano, E., B. Estebanez, P. del Saz-Orozco, I. García-Morales, y V.F. Muñoz. “Robot quirúrgico auto-guiado para cirugía mínimamente invasiva en solitario.” ROBOT 2011. Sevilla, 2011.
- [2] Niemeyer, G., N. Swarup, G. Guthart, G. Toth, R. Young, y W. Nowlin. “Camera referenced control in a minimally invasive surgical apparatus”. World Patente WO0060521. 2000.
- [3] Guthart, G.S., y J.K. Salisbury. “The Intuitive Telesurgery System: Overview and Application” IEEE International Conference on Robotics and Automation. 2000. 618-621.
- [4] Fernández-Lozano, J.M. Gómez-de-Gabriel, V.F. Muñoz, I. García-Morales, D. Melgar, C. Vara and A. García-Cerezo. “Human-Machine Interface Evaluation in a Computer Assisted Surgical System”. International Conference on Robotics & Automation, New Orleans, LA, Abril 2004.
- [5] Lidia Santos, José L. González, Javier P. Turiel, Juan C. Fraile, Eusebio de la Fuente. “Guante de datos sensorizado para uso en cirugía laparoscópica asistida por la mano”. Actas de las XXXVI Jornadas de Automática, Bilbao, 2 - 4 de septiembre de 2015.
- [6] E. J. Hazebroek and Color Study Group. “COLOR: a randomized clinical trial comparing laparoscopic and open resection for colon cancer”. Surgical Endoscopy 2002;16 (6):949-953.
- [7] I. Rivas-Blanco, B. Estebanez, M. Cuevas-Rodríguez, I. García-Morales, V.F. Muñoz. “Diseño de un asistente camarógrafo para técnicas de cirugía laparoscópica de puerto único”, XXXV Jornadas de Automática, Valencia, 2014
- [8] I. Rivas-Blanco, E. Bauzano, M. Cuevas Rodríguez, P. del Saz Orozco and V.F. Muñoz: “Force Position Control for a Miniature Camera Robotic System for Single Site Surgery”. 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). November 3-7, 2013. Tokyo, Japan.
- [9] Enrique Bauzano, B. Estebanez, I. García Morales, Víctor F. Muñoz, (2014). “Collaborative Human-Robot System for HALS Suture Procedures”.

- [10] Targarona, E.M., Garcia, E., Rodriguez, M., Cerdán, G., Balagué, C., Garriga, J., Trias, M., (2003) “Hand-assisted laparoscopic surgery”, *Archives of Surgery*, 138:2, 1 Febrero 2003, pp. 133-141.
- [11] Pietrabissa, A., Morelli, L., Peri, A., Pugliese, L., Zonta, S., Dionigi, P., Mosca, F., (2011) “Laparoscopic treatment of splenomegaly: A case for hand-assisted laparoscopic surgery”, *Archives of Surgery*, 146:7, Julio 2011, pp 818- 823.
- [12] Darzi, A., Jakimowicz, J., Cohen, R., Fleshman, J., Brinbaum, E., Davis, C., Medich, D., Himpens, J.M., Litwin, D.E.M., Kelly, J.J., Kercher, K.W., Marvik, R., (2000) “Hand assisted laparoscopic surgery vs standard laparoscopic surgery for colorectal disease: A prospective randomized trial”, *Surgical Endoscopy*, 14:10, 2000, pp. 896-901.
- [13] Ringley, C., Lee, Y.K., Iqbal, A., Bocharov, V., Sasson, A., McBride, C.L., Thompson, J.S., Vitamvas, M.L., Oleynikov, D., (2007) “Comparison of conventional laparoscopic and hand-assisted oncologic segmental colonic resection”, *Surgical Endoscopy and Other Interventional Techniques*, 21:12, December 2007, pp. 2137-2141.
- [14] J. D. Vogel, L. Lian, M. F. Kalady, L. F. deCampos-Lobato, P. C. Alves- Ferreira, and F. H. Remzi, “Hand-assisted laparoscopic right colectomy: How does it compare to conventional laparoscopy?” *J. Amer. Coll. Surg.* vol. 212, no. 3, pp. 367–372, Mar. 2011.
- [15] S. T. Orcutt, C. L. Marshall, C. J. Balentine, C. N. Robinson, D. A. Anaya, A. Artinyan, D. H. Berger, and D. Albo, “Hand-assisted laparoscopy leads to efficient colorectal cancer surgery,” *J. Surg. Res.*, vol. 177, no. 2, pp. e53–e58, Oct. 2012.
- [16] R. Pendlimari, S. D. Holubar, J. Pattan-Arun, D. W. Larson, E. J. Dozois, J. H. Pemberton, and R. R. Cima, “Hand-assisted laparoscopic colon and rectal cancer surgery: Feasibility, short-term, and oncological outcomes,” *Surgery*, vol. 148, no. 2, pp. 378-385, Aug. 2010.
- [17] Taylor, R.H., B.D. Mittelstadt, B.D. Williamson, B. Musits, B.L. Glassman, y W.L. Bargar. “An Image-Directed Robotic System for Precise Orthopaedic Surgery.” *IEEE Transactions on Robotics and Automation* 10, no 3 (1994): 261-275.
- [18] Siebert, W., y S. Mai. “One year clinical experience using the robot system CASPAR for tkr”. *Proceedings of CAOS USA 2001*. 2001. 141-142.
- [19] A.J. Madhani, G. Niemeyer, J.K. Salisbury. “The Black Falcon: a teleoperated surgical instrument for minimally invasive surgery”. In *Proceedings of 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 1998.

- [20] Hongo, K., T. Goto, T. Miyahara, J. Koyama, y Y. Tanaka. “Telecontrolled micromanipulator system (NeuroRobot) for minimally invasive neurosurgery.” *Medical Technologies in Neurosurgery* 98 (2006): pp. 63-66.
- [21] Lum, M.J.H., y otros. “Teleoperation of a Surgical Robot via Airborne Wireless Radio and Transatlantic Internet Links.” 2007: 1-10.
- [22] Jakopec, M., F. Rodriguez, S.J. Harris, P. Gomes, J. Cobb, y B.L. Davies. “The hands-on orthopaedic robot "Acrobot": Early clinical trials of total knee replacement surgery.” *IEEE Transactions on Robotics and Automation* 19, no 5 (2003): 902-911.
- [23] Taylor, R.H., Jensen, P., Whitcomb, L.L., Barnes, A., Kumar, R., Stoianovici, D., Gupta, P., Wang, Z., de Juan, E., Kavoussi, L.: “Steady-hand robotic system for microsurgical augmentation”. *The International Journal of Robotics Research*, 18(12), (1999) 1201-1210.
- [24] Kraft, B.M., C. Jäger, K. Kraft, B.J. Leibl, y R. Bittner. “The AESOP robot system in laparoscopic surgery.” *Surgical Endoscopy*, 2004: 1216-1223.
- [25] Polet, R., y J. Donnez. “Gynecologic laparoscopic surgery with a palm-controlled laparoscope holder.” *The Journal of the American Association of Gynecologic Laparoscopists*, 2004: 73-78.
- [26] Dowler, N., y S. Holland. “The evolutionary design of an endoscopic telemanipulator.” *IEEE Robotics and Automation Magazine* 3, no 4 (1996): 46-55.
- [27] A. Casals, J. Amat, D. Prats, y E. Laporte. (1995). “Vision guided robotic system for laparoscopic surgery. In *IFAC International Congress on Advanced Robotics*”. Barcelona. España.
- [28] Castrillón, W.A. «Implementación de una Interfaz Hombre-Máquina para el Control de un Brazo Robótico Mediante Posturas Labiales.» 2009.
- [29] J. Hsu and S. Payandeh, “Toward tool gesture and motion recognition on a novel minimally invasive surgery robotic system” in *Proc. IEEE Int. Conf. Robot. Autom.*, Orlando, FL, USA, 2006, pp. 631–636.
- [30] Muñoz, V.F., y otros. “Control movement scheme based on manipulability concept for a surgical robotic assistant” *IEEE International Conference on Robotics and Automation*. 2006. 245-250.

- [31] Fernández-Lozano, J., y otros. “Human Machine Interface Evaluation in a Computer Assisted Surgical System.” IEEE International Conference on Robotics and Automation. 2004. 231-236.
- [32] N. Padoy and G. D. Hager, “Human–machine collaborative surgery using learned models,” in IEEE Int. Conf. Robot. Autom., Shangai, China, 2011, pp. 5285–5292.
- [33] James, A., B. Vieira, A. Darzi, y G. Yang. “Eye-Gaze Driven Surgical Workflow Segmentation.” MICCAI 2007. 2007. 110-117.
- [34] Florent Lalys, Laurent Riffaud, David Bouget and Pierre Janin. “A framework for the recognition of high-level surgical tasks from video images for cataract surgeries.” IEEE Trans Biomed Eng. 2012 Apr; 59(4): 966–976.
- [35] Andru P. Twinanda, Sherif Shehata, Didier Mutter, Jacques Marescaux, Michel de Mathelin, Nicolas Padoy: “EndoNet: A Deep Architecture for Recognition Tasks on Laparoscopic Videos”. May 2016.
- [36] Lin, H., I. Shafran, T. Murphy, A. Okamura, D. Yuj, y G. Hager. “Towards automatic skill evaluation: Detection and segmentation of robot-assisted surgical motions.” Computer Aided Surgery, 2006: 220-230.
- [37] Stefanie Speidel, Tom Zentek, Gunther Sudra, Tobias Gehrig, Beat Peter Müller-Stich, Carsten Gutt, Rüdiger Dillmann. “Recognition of surgical skills using Hidden Markov Models” Medical Imaging 2009: Visualization, Image-Guided Procedures, and Modeling.
- [38] R. Muradore, D. Bresolin, L. Geretti, P. Fiorini, and T. Villa, “Robot surgery,” IEEE Robot. Autom. Mag., vol. 18, no. 3, pp. 24–32, Sep. 2011.
- [39] S. Y. Ko, W. J. Lee, and D. S. Kwon, “Intelligent interaction based on a surgery task model for a surgical assistant robot: Awareness of current surgical stages based on a surgical procedure model,” Int. J. Control, Autom. Syst., vol. 8, no. 4, pp. 782–792, Aug. 2010.
- [40] M. Kranzfelder, C. Staub, A. Fiolka, A. Schneider, S. Gillen, D. Wilhelm, H. Friess, A. Knoll, and H. Feussner, “Toward increased autonomy in the surgical OR: Needs, requests, and expectations,” Surg. Endosc., vol. 27, no. 5, pp. 1681–1688, May 2013.
- [41] H. Elhawary and A. Popovic, “Robust feature tracking in the beating heart for a robotic-guided endoscope,” Int. J. Med. Robot. Comput.-Assisted Surg., vol. 7, no. 4, pp. 459–468, Dec. 2011.



- [42] O. Weede, H. Mönnich, B. Müller, and H. Wörn, “An intelligent and autonomous endoscopic guidance system for minimally invasive surgery,” in IEEE Int. Conf. Robot. Autom., Shangai, China, 2010, pp. 5762–5768.
- [43] C. Staub, T. Osa, A. Knoll, and R. Bauernschmitt, “Automation of tissue piercing using circular needles and vision guidance for computer aided laparoscopic surgery,” in IEEE Int. Conf. Robot. Autom., Anchorage, AK, USA, 2010, pp. 4585–4590.
- [44] C. E. Reiley and G. D. Hager, “Task versus subtask surgical skill evaluation of robotic minimally invasive surgery,” *Med. Image Comput. Comput.-Assisted Intervention*, vol. 12, no. Pt. 1, pp. 435–442, 2009.
- [45] F. Nageotte, P. Zanne, C. Doignon, and M. Mathelin, “Stitching planning in laparoscopic surgery: Towards robot-assisted suturing,” *Int. J. Robot. Res.*, vol. 28, no. 10, pp. 1303–1321, Oct. 2009.
- [46] H. Fuhan and S. Payandeh, “Real-time knotting and unknotting,” in IEEE Int. Conf. Robot. Autom., Roma, Italy, 2007, pp. 2570–2575.
- [47] S. Patil and R. Alterovitz, “Toward automated tissue retraction in robot-assisted surgery,” in IEEE Int. Conf. Robot. Autom., Anchorage, AK, USA, 2010, pp. 2088–2094.