



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
PROGRAMA DE MAESTRÍA Y DOCTORADO EN INGENIERÍA
INGENIERÍA ELÉCTRICA - INSTRUMENTACIÓN

DESARROLLO INSTRUMENTAL PARA LA SIMULACIÓN DE CIRUGÍA DE
MÍNIMA INVASIÓN EN RODILLA

TESIS
QUE PARA OPTAR POR EL GRADO DE:
MAESTRO EN INGENIERÍA

PRESENTA:
SOLIS OVIEDO CARLOS JAVIER

TUTORES PRINCIPALES
FERNANDO PÉREZ ESCAMIROSA, ICAT
TETYANA BAYDYK MYKOLAIVNA, ICAT

CD. DE MÉXICO, SEPTIEMBRE 2021



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

JURADO ASIGNADO:

Presidente: Dr. Arámbula Cosío Fernando, IIMAS
Secretario: Dr. Padilla Castañeda Miguel Ángel, ICAT
1 er. Vocal: Dra. Baydyk Mykolaivna Tetyana, ICAT
2 do. Vocal: Dr. Pérez Escamirosa Fernando, ICAT
3 er. Vocal: Dr. Bañuelos Saucedo Miguel Ángel, ICAT

Lugar o lugares donde se realizó la tesis: Cd. Universitaria, Centro de Ciencias Aplicadas y Tecnologías, Posgrado de Ingeniería Eléctrica, UNAM.

TUTORES DE TESIS:

Dr. Fernando Pérez Escamirosa

Dra. Baydyk Mykolaivna Tetyana

FIRMA

FIRMA

El presente trabajo de Tesis nivel Maestría fue desarrollado en el laboratorio de sistemas inteligentes del Instituto de Ciencias Aplicadas y Tecnología de la Universidad Nacional Autónoma de México, bajo la tutoría del Dr. Fernando Pérez Escamiroso y la Dra. Tetyana Baydyk Mykolaivna. Se contó con el apoyo de una beca para estudios de maestría del CONACYT. Además, este trabajo se realizó con el apoyo del programa UNAM-DGAPA-PAPIIT, con números de proyecto IT200820 e IT102320, y el programa UNAM-DGAPA-PAPIME, con número de proyecto PE217020.

Agradecimientos Institucionales

Universidad Nacional Autónoma de México, UNAM.

Facultad de Ingeniería, Programa de Maestría en Ingeniería, Ingeniería Eléctrica-Instrumentación.

Instituto de Ciencias Aplicadas y Tecnología, ICAT.

Laboratorio de Sistemas Inteligentes del Departamento de Tecnologías de la Información y Procesos Educativos.

Consejo Nacional de Ciencia y Tecnología, CONACYT.

Hospital Nacional de Rehabilitación, INR.

Programa UNAM-DGAPA-PAPIIT, con números de proyecto IT200820 e IT102320.

Programa UNAM-DGAPA-PAPIME, con número de proyecto PE217020.

Tutores:

Mi más profundo agradecimiento al Dr. Fernando Pérez Escamiroso, por la confianza, paciencia, y apoyo que me brindó durante la elaboración de este trabajo. En mi corta vida académica he conocido a excelentes investigadores, pero a pocos como el Dr. Fernando, que además tiene una gran vocación y compromiso para apoyar a sus estudiantes.

Mi sincero agradecimiento a la Dra. Tetyana Baydyk Mykolaivna, por su motivación, consejos, paciencia y todo el conocimiento que me compartió, durante la realización de este trabajo y durante mi formación académica. Admiro mucho su calidad como persona y su talento como investigadora.

Asesorías:

Agradezco al Dr. Francisco Javier Pérez Jiménez del Hospital Nacional de Rehabilitación por compartir su experiencia y su tiempo.

Agradezco a la Dra. Graciela Velasco Herrera por su apoyo en la realización de este trabajo.

Sinodales:

Agradezco al Dr. Miguel Ángel Bañuelos Saucedo, que con una gran vocación me compartió sus conocimientos en electrónica, además agradezco su tiempo y aportes a este trabajo.

Agradezco al Dr. Miguel Ángel Padilla Castañeda, por compartirme su conocimiento, por su tiempo y aportaciones a este trabajo.

Agradezco al Dr. Fernando Arámbula Cosío, por su tiempo y aportaciones a este trabajo.

Profesores:

Agradezco a los profesores que con tanta vocación me han formado como ingeniero, especialmente al M.I. Filiberto Gutiérrez Martínez, Dr. Enrique Jaime Chicurel Uziel, M.I. Antonio Zepeda Sánchez, M.A. Luis Yair Bautista, M.A. David Horton Cavazos, Dr. Sergio Teodoro Vite, M.I. Mayverena Jurado Pineda, Dr. Fernando Velázquez Villegas, M.I. Pedro Javier Gálvez Valadez, M.I. Ledezma Rubio Yahvé Abdul, Ing. Sergio Esteves Rebollo, M.A. Edmundo Guzmán Cortés.

Agradecimientos personales

Agradezco profundamente a mi familia por todo su apoyo incondicional, por enseñarme la importancia de ser una buena persona, y siempre tratar de ser mejor. A mi madre Maricela Oviedo Palomino, por todos sus buenos consejos que siempre me brinda, por todo su esfuerzo en convertirme en alguien con muchos valores y muchos sueños que poco a poco se hacen realidad. A mi padre Javier Solis Hernandez que siempre me ha apoyado en lograr lo que me propongo y que despertó en mí el gusto por la Ingeniería. A mi hermana Magui quien siempre me ha inspirado y me ha enseñado que el trabajo arduo y profesional es el camino para lograr lo que me propongo. A mi hermano Fer que siempre tiene algo que enseñarme, y me inspira con todos sus retos y logros. A Paulina Bastida que con tanto amor y cariño ha estado a mi lado apoyándome y animándome en todo momento, tengo toda la vida para agradecerlo.

Agradezco el apoyo incondicional de mi familia cercana, que siempre me anima e inspira a lograr lo que me propongo, especialmente a mi tío Efraín y su familia Oviedo Longinos, a mi tía Remedios y su familia Arvizu Oviedo, a mi tía Concepción y su familia Cordero Oviedo, a mi tío Francisco Oviedo, a mi tía María de Jesús y su familia Tovar Solis, a las familias Oviedo Urióstegui, Rico Oviedo, Oviedo Palomino, Quintana Solis, Solis Hernandez. A la memoria de mi abuela Victoria Hernandez quien siempre me brindó su cariño. A la memoria de mi tío Arturo Arvizu quien siempre me animó, desde pequeño, a esforzarme en todo lo que hago, y ahora vivirá en cada una de mis buenas acciones.

Agradezco a la familia Bastida Rojano, que siempre me brinda su apoyo incondicional, y que me hace sentir como a un integrante más de su familia. Al Dr. Alfonso Bastida Medina quien me brindó su confianza y espacio para mi formación profesional.

Agradezco a mis colegas Carmen Camargo, Rodrigo Olmos y Raúl Ayala por aconsejarme, apoyarme e inspirarme en mis retos profesionales, además de enseñarme, con su ejemplo, a ser un Ingeniero integro, responsable y profesional.

Agradezco a mis compañeros y amigos que me acompañaron durante mi formación académica, especialmente a Inri J., Fernanda A., Fernando H., Daniel V., Gerardo Y., Iván M., Ranfery S., Victoria C., Foly Y., Miriam A., Dani B., Estefanía G., Ximena Z., Liliana C., Emilio H., Raymundo P., Marcelino P., Mariana T., Jessica M., Javier F., Alejandra del A., Karla V., Isaias B., Yahir P., Ximena M., Desiré R., Daniela E., Karen R., Casandra N., Luis N., Armando B., Carla R., Karen A., Alejandra R., Noemi L., Zaira O., Tonatiuh N., Misrahim B., Ignacio R., Israel S., Jesús A., Fernanda E., David M., Erick A y Rodrigo D.

Agradezco a mis compañeros y amigos de la agrupación estudiantil Baja SAE UNAM de quienes aprendí, y compartí grandes experiencias, especialmente a Alfredo C., Raúl L., Abraham M., Viridiana A., Belén A., Hector M., Cabeto V., Francisco A., Angel V., Elena A., Guillermo J., Alejandro S., Karla A., Carlos J., Antonio V., y Linette T.

Agradezco a mis amigos del trabajo que siempre me apoyaron e inspiraron, especialmente a Elienai C., Nadia P., Diana M., Gabriela R., Diana G., Carolyn J., Ariana B., Karla O., Paola C., Javier S., Manuel M., Marissa G., Verónica A., Allison J., Montserrat S., Yazmin F., Anastacio M., Omar O., Luis S., Sandra Y., Joel H., Ubaldo R., Erika I., Pablo O., Estefania E., y Karen M.

“No creo que haya alguna emoción más intensa para un inventor que ver alguna de sus creaciones funcionando. Esa emoción hace que uno se olvide de comer, de dormir, de todo.”

Nicola Tesla

Contenido

RESUMEN.....	9
CAPÍTULO 1. INTRODUCCIÓN	10
1.1 Objetivo general.....	10
1.2 Objetivos específicos	10
1.3 Justificación del problema	11
1.4 Metodología	11
1.5 Hipótesis.....	12
1.6 Estructura de la Tesis	12
CAPÍTULO 2. ANTECEDENTES Y ESTADO DEL ARTE	14
2.1 Simuladores de artroscopía en rodilla.....	14
2.2 Conceptos y fundamentos teóricos.....	21
2.2.1 Cálculo de los ángulos de <i>Euler</i> a partir de los sensores inerciales	21
2.2.2 <i>Filtro complementario</i> para la estimación de los ángulos de <i>Euler</i>	24
2.2.3 Redes neuronales	27
2.2.4 Clasificador de umbrales aleatorios <i>RTC</i>	28
CAPÍTULO 3. INSTRUMENTACIÓN ELECTRÓNICA.....	30
3.1 Requerimientos para el arreglo instrumental.....	30
3.2 Componentes principales de la instrumentación electrónica.....	36
3.2.1 Sensor inercial <i>MPU9250</i>	36
3.2.2 Sensor de efecto <i>hall AS5600</i>	36
3.2.3 Sensor reflectivo <i>QRE1113</i>	37
3.2.4 Microcontrolador <i>PIC18F4550</i>	38
3.3 Funcionamiento del arreglo instrumental.....	39
3.3.1 Mediciones de los movimientos del modelo de rodilla	40

3.3.2 Mediciones de los movimientos del instrumental quirúrgico.....	43
3.3.3 Mediciones para identificar el instrumental quirúrgico.....	49
3.3.4 Mediciones en el artroscopio.....	51
3.3.5 Mediciones en las pinzas de artroscopía.....	53
3.4 Tarjeta electrónica del arreglo instrumental.....	54
3.5 Construcción del dispositivo instrumentado	56
3.6 Simulación de los movimientos medidos por el arreglo instrumental.....	59
3.6.1 Comunicación con la interfaz gráfica.....	59
3.6.2 Simulación del movimiento en Unity.....	60
CAPÍTULO 4. CALIBRACIÓN DE LOS SENSORES DEL ARREGLO INSTRUMENTAL..	63
4.1 Calibración de los sensores <i>MPU9250</i>	63
4.1.2 Estimación del rango y precisión del movimiento de la rodilla	64
4.2 Calibración de los sensores <i>AS5600</i>	66
4.2.1 Resolución, rango y precisión de los ángulos γ_1 , γ_2 , α_1 y α_2	67
4.2.2 Resolución, rango y precisión de los ángulos β_1 y β_2	70
4.2.3 Resolución, rango y precisión de los desplazamientos δ_1 y δ_2	71
4.2.4 Resolución, rango y precisión del ángulo del lente del artroscopio.	73
4.2.5 Resolución, rango y precisión del ángulo de apertura de las pinzas.	74
4.3 Calibración de los sensores <i>QRE1113</i>	75
4.4 Propuesta del método para determinar la exactitud del arreglo instrumental.....	77
CAPÍTULO 5 PROPUESTA DE RED NEURONAL PARA LA CLASIFICACIÓN DEL TIPO DE USUARIO.....	79
5.1 Características de la red neuronal para la clasificación del usuario del simulador... ..	79
5.1.1 Características generales de la red neuronal	79
5.1.2 Variables de entrada y salida de la red neuronal	80
5.2 Aplicación de la red neuronal <i>RTC</i>	83
5.3 Aplicación de la red neuronal perceptrón	84

CAPÍTULO 6. RESULTADOS	86
6.1 Características generales del arreglo instrumental	86
6.2 Resultados del filtro complementario.....	88
6.3 Precisión y rango de la posición del instrumental quirúrgico	92
6.4 Resultados de la aplicación de redes neuronales para la clasificación del tipo de usuario.....	99
CAPÍTULO 7 CONCLUSIONES	101
7.1 Conclusiones	101
7.2 Trabajo a futuro	102
REFERENCIAS	103
ANEXOS	110
Anexo A. Circuitos del arreglo instrumental	110
Anexo B. Programa del microcontrolador <i>PIC18F4550</i>	117
Anexo C. Scripts de Unity para la simulación virtual	123
Anexo D. Base de datos del simulador <i>EndoViS training system</i>	130
Anexo E. Programación de la red neuronal <i>RTC</i> en <i>Matlab</i>	131
Anexo F. Programación de la red neuronal perceptrón en <i>Matlab</i>	134

RESUMEN

Actualmente, existe la necesidad de desarrollar herramientas tecnológicas para que los residentes de ortopedia y trauma adquieran habilidades y destrezas quirúrgicas que les permitan realizar cirugías artroscópicas de manera segura y precisa. En el presente trabajo se muestra el diseño, construcción y pruebas de un arreglo instrumental para medir los movimientos de un modelo anatómico de rodilla, y del instrumental quirúrgico que se utiliza durante cirugías de mínima invasión en rodilla.

El arreglo instrumental propuesto se compone principalmente de dos sensores inerciales (*MPU9250* de *Sparkfun*, uno montado en el modelo de la rodilla para medir sus movimientos y el otro fijo en el artroscopio para medir su orientación), diez sensores de efecto hall (*AS5600* de *AMS*, para medir los movimientos del instrumental quirúrgico), dos sensores reflectivos (*QRE1113* de *ON Semiconductor*, para identificar el instrumental quirúrgico) y una tarjeta electrónica con un microcontrolador (*PIC18F4550* de *Microchip*).

Un *filtro complementario* digital se implementa en el microcontrolador de la tarjeta electrónica para combinar las señales y reducir el ruido de los acelerómetros, magnetómetros y giróscopos que están integrados en los sensores *MPU9250*. La tarjeta electrónica recibe las señales de todos los sensores, aplica los filtros digitales, estima los movimientos del modelo anatómico de rodilla y de los instrumentos quirúrgicos, por último, envía la información a un ambiente de realidad virtual en una computadora, donde se simulan dichos movimientos.

Se estudia el rango, resolución y precisión en la señal de cada variable que mide el arreglo instrumental. Los parámetros anteriores se obtienen a partir de las restricciones de movimiento del modelo anatómico de rodilla, de las hojas de datos de los fabricantes de los sensores, y del ruido de las señales. En las posiciones de las puntas del instrumental quirúrgico se alcanzó una precisión de ± 0.1 [mm], y un error observado en las señales de ± 0.32 [mm].

Una red neuronal artificial permite la clasificación de los usuarios, con base en su experiencia y rendimiento en cirugías artroscópicas o el uso de simuladores. Las variables de entrada de dicha red se calculan a partir de los datos de movimiento recopilados por un simulador. Las variables de salida de la red neuronal artificial representan la clasificación del grado de experiencia del usuario, entre experto, intermedio o principiante.

CAPÍTULO 1. INTRODUCCIÓN

La cirugía artroscópica es un tipo de cirugía mínimamente invasiva que se practica en las articulaciones como el hombro, tobillo, muñeca y rodilla. Una de sus dificultades es que el cirujano no tiene visión directa sobre el campo quirúrgico. En su lugar observa un monitor bidimensional conectado a un artroscopio, lo que lleva a la pérdida de profundidad. Otra dificultad es que el cirujano inserta el instrumental quirúrgico por pequeñas incisiones sobre la piel, lo que limita el rango de movimiento y su alcance [1, 2].

El ciclo de aprendizaje del cirujano inicia con una fase de estudio, seguido por un período de prácticas en quirófano, con operaciones reales, incorporándose gradualmente en tareas auxiliares, bajo la supervisión de un experto [3]. También, se realizan prácticas sobre cadáveres en salas de disección, limitadas por su capacidad, costo y tiempo [4].

Algunos lugares de enseñanza quirúrgica implementan simuladores para el entrenamiento de los cirujanos. Está demostrado en diversos estudios que los simuladores reducen la curva de aprendizaje en la técnica quirúrgica de artroscopía [5-9]. Además, ofrecen una opción segura, ética y eficiente en el entrenamiento de habilidades quirúrgicas [10-13]. Para la evaluación de estas habilidades, los simuladores emplean el análisis de movimiento del instrumental durante la ejecución de tareas y procedimientos quirúrgicos. Estos dispositivos utilizan diversas tecnologías para capturar el movimiento del instrumento, basados en articulaciones mecánicas, dispositivos electromagnéticos, ópticos o ultrasónicos. [14-17].

En este trabajo de tesis se presenta el desarrollo de un arreglo instrumental electrónico integrado a un modelo físico de rodilla, para medir los movimientos de dicho modelo anatómico y del instrumental quirúrgico que se utiliza durante cirugías artroscópicas. Estos movimientos se simulan en un ambiente de realidad virtual.

1.1 Objetivo general

El objetivo general del presente trabajo es desarrollar un arreglo instrumental con diferentes sensores para medir los movimientos de un modelo anatómico de rodilla y de los instrumentos quirúrgicos que se utilizan durante cirugías artroscópicas.

1.2 Objetivos específicos

1. Elaborar un estudio del estado del arte.
2. Identificar las variables del movimiento para proponer los sensores del arreglo instrumental.
3. Determinar los componentes y la electrónica del arreglo instrumental.
4. Programar un algoritmo para filtrar el ruido de las señales de los sensores y para estimar los movimientos del modelo anatómico de rodilla y de los instrumentos quirúrgicos.

5. Integrar los componentes del arreglo instrumental a un modelo anatómico de rodilla.
6. Construir el arreglo instrumental, para verificar su funcionamiento.
7. Establecer el rango, resolución y precisión de las variables de movimiento medidos por el arreglo instrumental en la articulación mecánica.
8. Simular en un ambiente de realidad virtual los movimientos del modelo de rodilla y del instrumental quirúrgico.
9. Proponer una red neuronal artificial para clasificar la experiencia quirúrgica de los usuarios, basada en sus habilidades quirúrgicas en cirugías artroscópicas o en el uso de simuladores.

1.3 Justificación del problema

Actualmente, existe una necesidad creciente en herramientas tecnológicas para el entrenamiento de destrezas técnicas en cirugías artroscópicas. Existe una falta de infraestructura en el país para importar, almacenar y dar mantenimiento a los simuladores artroscópicos [18]. Por lo tanto, es de interés el desarrollo de estos simuladores, que permitan a los residentes desarrollar habilidades y destrezas quirúrgicas de manera segura.

En el desarrollo actual de los simuladores de artroscopía se utiliza comúnmente instrumentación electrónica que no es destinada para este tipo de cirugías, en donde se adaptan sistemas diseñados para laparoscopia o sistemas de seguimiento comerciales. El desarrollo de instrumentación electrónica para medir los movimientos de los instrumentos quirúrgicos, durante cirugías artroscópicas, es escaso en América Latina.

Al desarrollar un dispositivo instrumental electrónico, específicamente para artroscopia en rodilla, se proporciona una opción mejor adaptada a las necesidades del aprendizaje de los cirujanos. Para desarrollar dicho arreglo, se requiere del estudio de sensores y microcontroladores, así como su integración en un modelo anatómico de rodilla. También, se requiere la implementación de algoritmos para estimar las posiciones y orientaciones del modelo anatómico de rodilla y de los instrumentos quirúrgicos.

Con la finalidad de obtener una herramienta de evaluación, se propone una red neuronal artificial, donde sus variables de entrada sea la información recopilada por la instrumentación electrónica desarrollada, y las variables de salida indiquen el nivel de experiencia quirúrgica del usuario.

1.4 Metodología

La metodología para el desarrollo del arreglo instrumental fue la siguiente:

1. Se estudió el estado del arte, y se plantearon los requerimientos del arreglo instrumental.
2. Se estudiaron y seleccionaron los elementos de la instrumentación electrónica, entre ellos, los sensores y microcontroladores.

3. Se diseñaron las piezas mecánicas que permitieron integrar a los sensores en un modelo anatómico de rodilla.
4. Se diseñaron los circuitos electrónicos con base en el tamaño de los componentes electrónicos y de los elementos mecánicos en los puertos de entrada del modelo de rodilla.
5. Se construyó el arreglo instrumental, donde las piezas de los mecanismos se manufacturaron con impresión *3D* y las *PCB* de los circuitos electrónicos se manufacturaron en una máquina *CNC*.
6. Se programó el microcontrolador para recibir las señales de los sensores, aplicar los filtros digitales y estimar las orientaciones de la rodilla y del instrumental quirúrgico para enviar esta información a un equipo de cómputo.
7. Se calibraron los sensores a partir de sus hojas de datos, y de los mecanismos que mantuvieron fijos a los instrumentos quirúrgicos, en posiciones conocidas.
8. Se hicieron pruebas para determinar el rango, resolución y precisión de las variables que se estimaron con el arreglo instrumental.
9. Se desarrollaron redes neuronales artificiales para clasificar a los usuarios de un simulador de cirugías de mínima invasión.
10. Se redactó este trabajo de investigación.

1.5 Hipótesis

El desarrollo de un arreglo instrumental, que está integrado en un modelo anatómico de rodilla, permitirá simular un ambiente en el que los cirujanos de ortopedia puedan desarrollar sus habilidades quirúrgicas para cirugías de artroscopía.

Al construir el arreglo instrumental se ofrecerán alternativas para el desarrollo de simuladores de cirugías artroscópicas. Así mismo, se contribuirá al área de investigación, al estudiar la aplicación de algunos sensores comerciales en la medición de movimientos para simuladores de artroscopía.

El uso de una red neuronal artificial permitirá clasificar a los cirujanos a partir de su experiencia en cirugías artroscópicas o el uso de simuladores.

1.6 Estructura de la Tesis

En el Capítulo 1 se describen las generalidades del trabajo.

En el Capítulo 2 se presenta la recopilación de la información del estado del arte, así como los fundamentos teóricos en el que se basó el desarrollo del arreglo instrumental.

En el Capítulo 3 se presenta el desarrollo de la propuesta, desde la selección de los sensores hasta su integración en el arreglo instrumental.

En el Capítulo 4 se presenta la calibración de los sensores a partir de sus hojas de datos. También se presenta la estimación del rango, resolución y precisión de las variables medidas con los sensores.

En el Capítulo 5 se presentan las propuestas de redes neuronales artificiales para la clasificación del cirujano, según su experiencia quirúrgica.

En el Capítulo 6 se encuentra la discusión de los resultados del arreglo instrumental.

En el Capítulo 7 se presentan las conclusiones de la investigación y el trabajo a futuro.

En los Anexos se presentan los circuitos del arreglo instrumental, el programa del microcontrolador, los *scripts* de la interfaz en *Unity*, la base de datos de la red neuronal y la programación de las redes neuronales propuestas.

CAPÍTULO 2. ANTECEDENTES Y ESTADO DEL ARTE

2.1 Simuladores de artroscopía en rodilla

Existen diferentes tipos de simuladores en artroscopia: simuladores cadavéricos, maniqués y simuladores con realidad virtual, Figura 1. Los simuladores cadavéricos son realistas, tienen una alta fidelidad, son costosos y requieren un tiempo de preparación previa. Los maniqués son portables, tienen un costo relativamente bajo y su realismo es limitado. Los simuladores de realidad virtual permiten llevar un historial de aprendizaje del cirujano, provee escenarios con un amplio rango de procedimientos quirúrgicos y tienen un alto costo inicial [19].

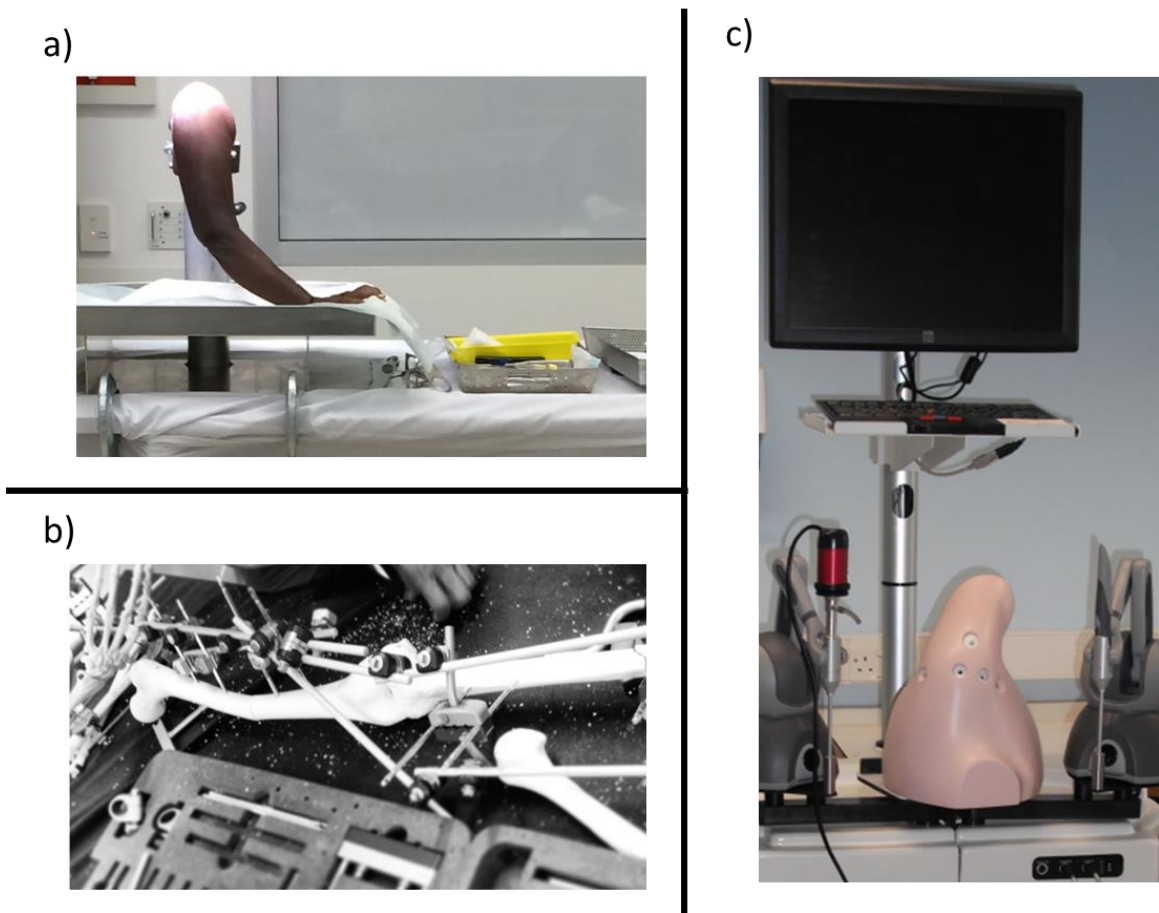


Figura 1. Simuladores en artroscopía. a) Simuladores cadavéricos. b) Simuladores con maniqués. c) Simuladores con realidad virtual. Imágenes obtenidas de [19].

Los simuladores permiten el desarrollo de habilidades quirúrgicas tales como coordinación ojo-mano, triangulación de instrumentos, interpretación de estructuras de tres a dos dimensiones.

El entrenamiento en cirugía artroscópica suele iniciar con la cirugía de rodilla, debido a que es la articulación más tratada con este tipo de operación [20]. Por ello, se han desarrollado simuladores específicamente para artroscopía en rodilla. En la Tabla 1 se muestran algunos desarrollos en la literatura de simuladores para cirugía artroscópica de rodilla.

El simulador *VE-KATS*, Figura 2, mide tres grados de libertad, dos rotaciones y un desplazamiento de cada instrumento quirúrgico. Dichos grados de libertad son medidos por sensores ópticos, además, se implementa una realimentación háptica con motores *DC*, para percibir los tejidos de la rodilla, con una fuerza máxima de 3.2 [N] [21].

Los sensores ópticos registran el movimiento del instrumental quirúrgico con una resolución de 0.036° en los ángulos de giro, y de $0.031 [mm]$ en el desplazamiento del instrumento quirúrgico. El rango máximo de los ángulos de giro es de 100° , y en el desplazamiento del instrumento es de $100 [mm]$.

Los movimientos del instrumental quirúrgico son mostrados en un ambiente virtual, donde se observan las colisiones de los instrumentos contra un modelo virtual del interior de la rodilla.

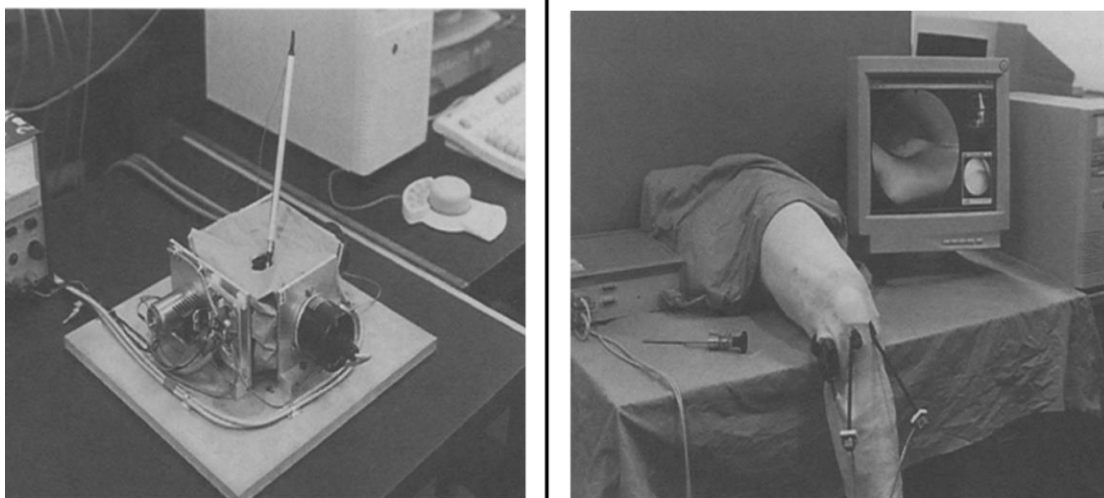


Figura 2. Simulador VE-KATS desarrollado en "University of Hull", en Reino Unido, en el año 1996. Figuras tomadas de [21].

Tabla 1. Desarrollos de simuladores en artroscopía de rodilla.

Simulador	Año	Arreglo instrumental utilizado	Tipo de gráficos
FAST [22]	2016	Sin arreglo instrumental	Maniquí de rodilla
insightArthroVR [23]	2015	<i>Omni –SensAble [24]</i>	Virtual 3D
ArthroCLEMI II [25]	2014	Sin arreglo instrumental	Maniquí de rodilla
ArthroSim de TolTech [26]	2013	Desarrollo propio	Virtual 3D
VirtaMed ArthroSTM [27]	2013	Desarrollo propio	Virtual 3D
Simbionix Arthro Mentor [28]	2012	<i>Touch X 3D Systems [29]</i>	Virtual 3D
PASSPORT [30]	2010	Desarrollo propio, <i>Knee phantom</i>	Virtual 3D
Simendo arthroscopy trainer [31]	2006	Desarrollo propio	Virtual 3D
OrthoForce [32].	2006	Desarrollo propio	Virtual 3D
SKATS [9]	2006	<i>miniBIRD de Ascension Technology Corporation [33]</i>	Maniquí de rodilla con cámara
Proyecto SARA [34]	2005	<i>Virtual Laparoscopic Interface de IMMERSION [35]</i>	Virtual 3D
Insight-Mist [36]	2004	<i>Virtual Laparoscopic Interface de IMMERSION [35]</i>	Virtual 3D
Fernández et al. [37]	2004	<i>Virtual Laparoscopic Interface de IMMERSION [35]</i>	Virtual 3D
Heng et al. [38]	2004	Desarrollo propio	Virtual 3D
WISHCATS [39]	2003	Desarrollo propio	Virtual 3D
VR-AKS [40]	1999	<i>Touch 3D Systems [24]</i>	Virtual 3D
VE-KAST [41]	1996	Desarrollo propio	Virtual 3D

El simulador publicado en [38], Figura 3, mide cuatro grados de libertad (*roll*, *pitch*, *yaw* y el desplazamiento del instrumento). Los sensores para medir los movimientos son ópticos y están acoplados a motores *DC*, que a su vez proveen de realimentación háptica.

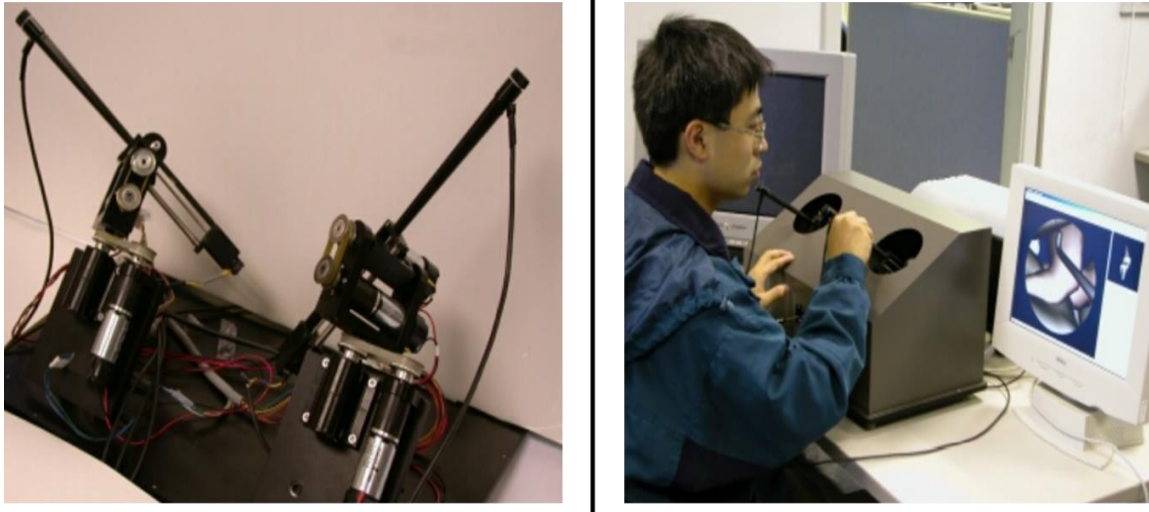


Figura 3. Simulador desarrollado en "The Chinese University of Hong Kong", en el año 2004. Figuras tomadas de [38].

El simulador *OrthoForce*, Figura 4, mide cuatro grados de libertad (*roll*, *pitch*, *yaw* y el desplazamiento del instrumento). Dichos grados de libertad son medidos por medio de un mecanismo cardán y por ejes estriados que mueven poleas y engranes acoplados a motores *DC* y sensores ópticos [42].

Los motores *DC* proporcionan una realimentación háptica. Los sensores ópticos registran el movimiento del instrumental quirúrgico, y dichos movimientos son mostrados en un ambiente virtual.

El rango de las variables *yaw* y *pitch* es de $\pm 63^\circ$, el rango del *roll* es continuo, y la traslación del instrumento es de 80 [mm]. La velocidad máxima del *yaw* y *pitch* es de $135^\circ/s$, del *roll* es de $180^\circ/s$ y de la traslación es de 100 [mm/s].

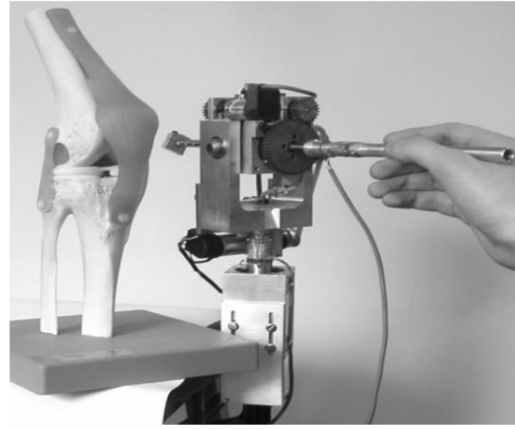
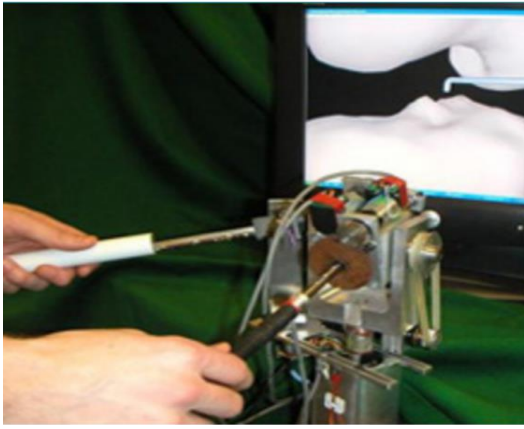


Figura 4. Simulador OrthoForce desarrollado en "Imperial College London", en Londres, en el año 2005. Figuras tomadas de [42].

El arreglo instrumental *Omni touch* [29] de 3D SYSTEMS y el *Phantom Omni* [43] de *Immersion* es utilizado en el desarrollo de simuladores para artroscopía en rodilla, Figura 5. Este sistema permite medir la orientación de su actuador, que simula un instrumento quirúrgico, con seis grados de libertad y realimentación háptica.

a)



b)



c)



d)

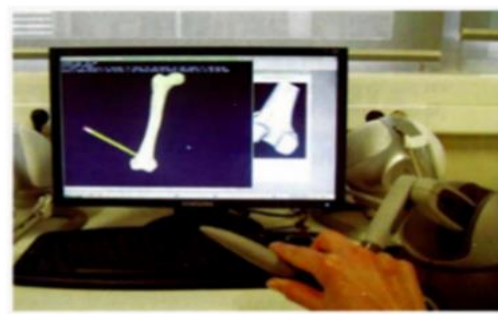


Figura 5. Simuladores de artroscopía en rodilla con el arreglo instrumental Phantom omni [24]. a) *insightArthroVR* [23]. b) Sistema publicado en [44]. c) Sistema publicado en [8]. d) Sistema publicado en [5].

Actualmente, el sistema *Omni touch* y el *Phantom Omni* tienen una resolución de hasta 0.055 [mm] y una fuerza háptica de hasta 3.3 [N].

El arreglo instrumental *Virtual Laparoscopy Interface* [35], de *Immersion*, se utiliza para la simulación de cirugías laparoscópicas, pero también se adapta para la simulación de cirugías artroscópicas de rodilla, Figura 6.

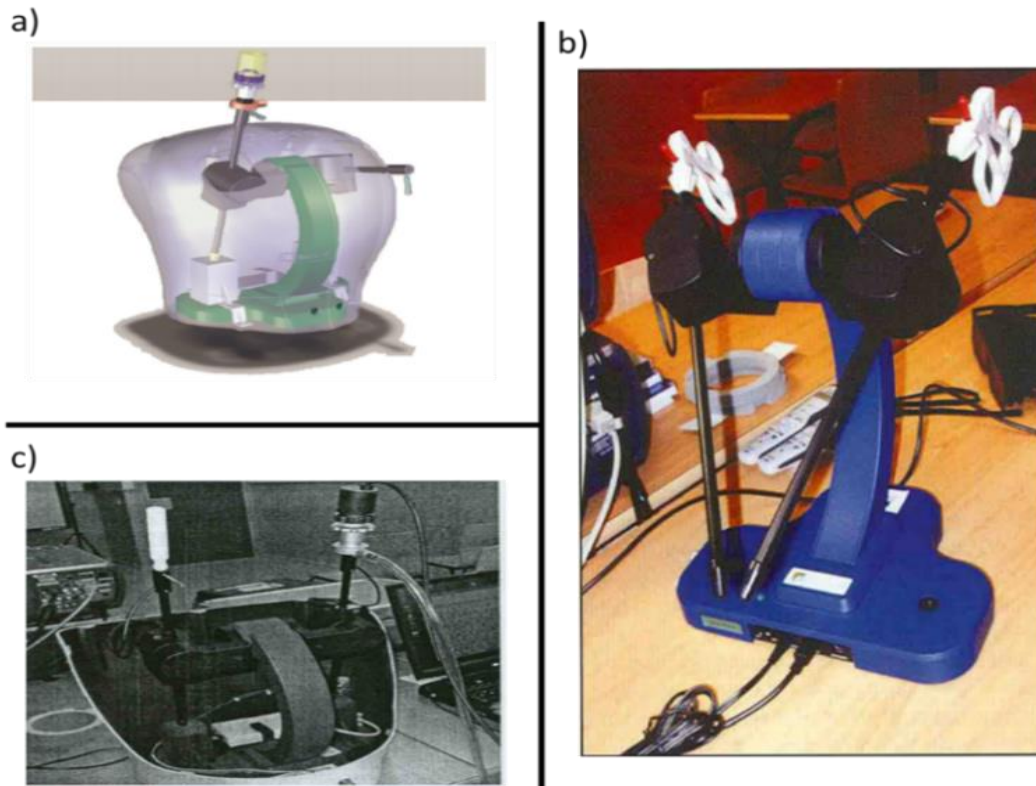


Figura 6. Simuladores artroscópicos que utilizan el arreglo instrumental *Virtual Laparoscopic Interface* [35]. a) Simulador publicado en [36]. b) Simulador publicado en [34]. c) Simulador publicado en [45].

Existen simuladores comerciales con realidad virtual para artroscopía en rodilla. Estos simuladores implementan tareas quirúrgicas que se realizan durante cirugías artroscópicas. También permiten llevar un registro del entrenamiento de cada usuario.

El simulador *ARTHRO MENTOR* [46] de *3D SYSTEMS SIMBIONIX*, Figura 7, permite cambiar de articulación entre rodilla, hombro y cadera. Su funcionamiento se basa en el sistema *Omni touch*, descrito anteriormente, por ello, cuenta con realimentación háptica. Este simulador incluye modelos anatómicos en fibra de vidrio, mientras que el simulador *ARTHRO MENTOR EXPRESS*, desarrollado por la misma empresa, incluye únicamente retroalimentación háptica.

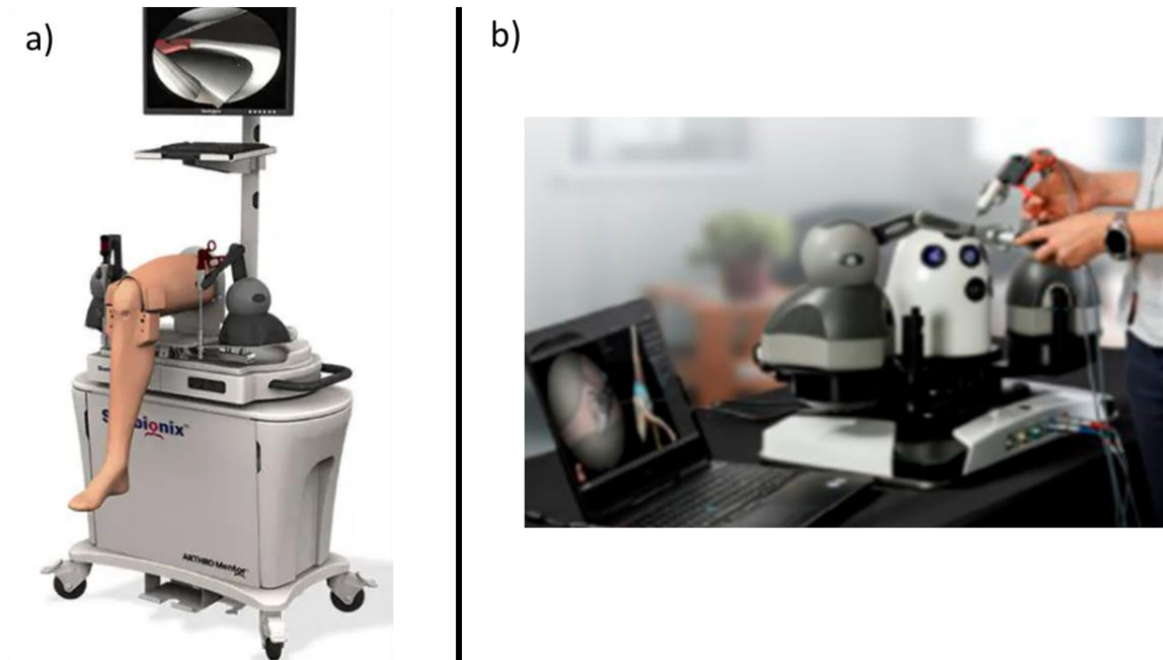


Figura 7. Simuladores de artroscopía desarrollados por 3D SYSTEMS SIMBIONIX [28].
 a) ARTHRO MENTOR. b) ARTHRO MENTOR EXPRESS.

El simulador *Simendo Arthroscopy* [31] de *SIMENDO*, Figura 8, permite simular tareas básicas para el entrenamiento de cirugías artroscópicas de rodilla y hombro. Este simulador no tiene realimentación háptica.



Figura 8. Simulador “Simendo Arthroscopy” de *SIMENDO* [31].

El simulador *VirtaMed ArthroS* [27] de *VirtaMed*, Figura 9, simula tareas quirúrgicas de cirugías en rodilla, hombro, cadera o tobillo. El simulador incluye cursos desarrollados en conjunto con la *Asociación de Artroscopía de América del Norte, AANA* por sus siglas en inglés. Este simulador cuenta con realimentación háptica y modelos anatómicos, mientras que el simulador *VirtaMed ArthroS FAST*, desarrollado por la misma empresa, incluye únicamente retroalimentación háptica.

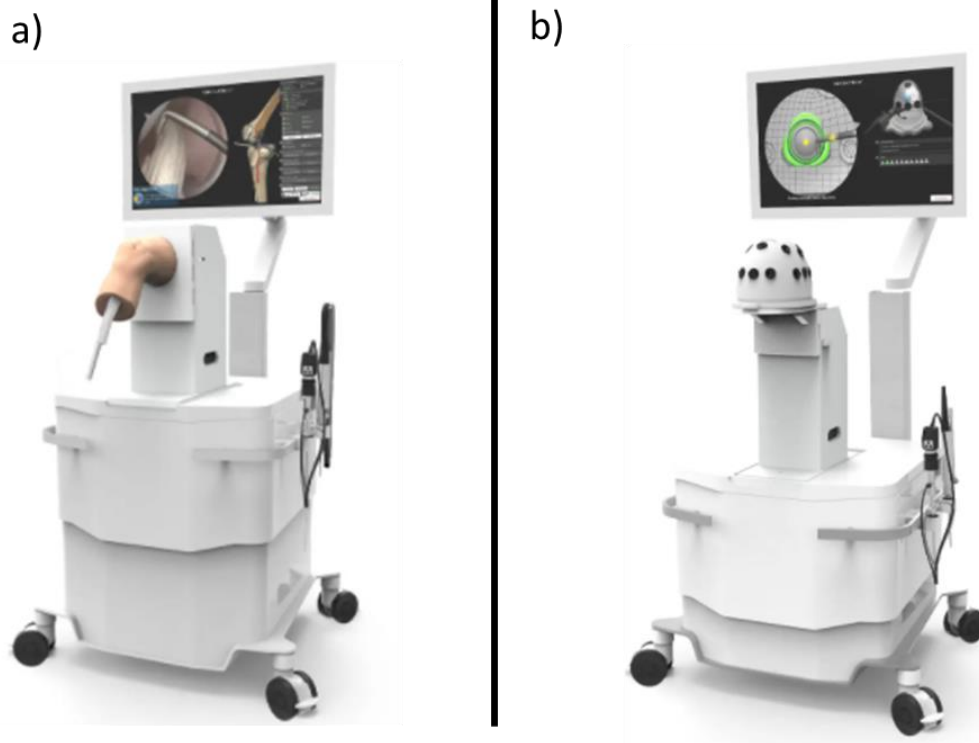


Figura 9. Simuladores de artroscopía desarrollados por *VirtaMed* [27]. a) *VirtaMed ArthroS*. b) *VirtaMed ArthroS FAST*.

2.2 Conceptos y fundamentos teóricos

2.2.1 Cálculo de los ángulos de *Euler* a partir de los sensores inerciales

Un sistema de referencia *A* se encuentra fijo y un sistema de referencia *B* se encuentra en un sensor inercial sujeto a la tibia de un modelo anatómico de rodilla [47]. Los acelerómetros y magnetómetros miden principalmente dos vectores conocidos, que son la aceleración de la gravedad *g* y el norte magnético *n*, Figura 10.

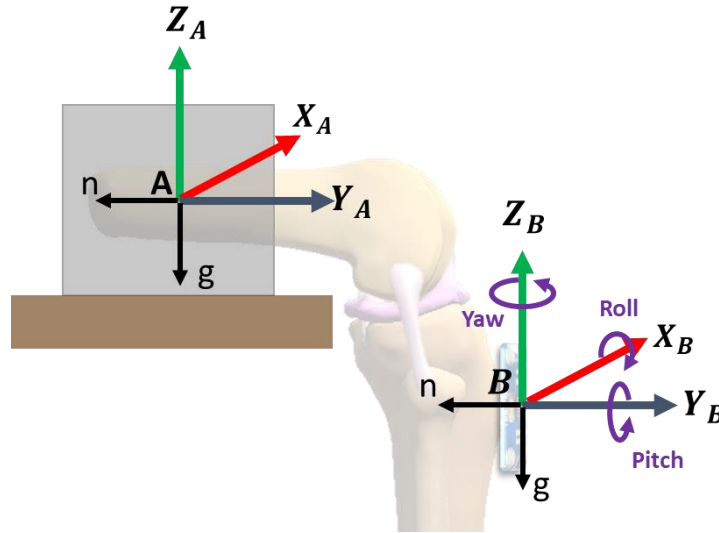


Figura 10. Sistemas de referencia en el modelo de rodilla, el sistema B está en el sensor inercial MPU9250 sujeto a la tibia y el sistema A está fijo en el fémur.

La transformación de coordenadas de un punto ${}^B P$ entre el sistema de referencia B y el sistema de referencia fijo A es:

$${}^A P = {}^A_B R {}^B P \quad (1.1)$$

Donde ${}^A_B R$ es la matriz de rotación del sistema de referencia B con respecto al sistema de referencia A, que se obtiene a partir de los ángulos de *Euler*, con giros sobre el eje Z, Y y X respectivamente.

$${}^A_B R = \begin{bmatrix} r_{1,1} & r_{1,2} & r_{1,3} \\ r_{2,1} & r_{2,2} & r_{2,3} \\ r_{3,1} & r_{3,2} & r_{3,3} \end{bmatrix} = \begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma \\ s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix} \quad (1.2)$$

Los giros γ , β y α son los ángulos roll, pitch y yaw, respectivamente, c representa al coseno y s al seno. Para estimar los ángulos de giro se utilizan los componentes de la matriz de rotación (1.2).

$$\gamma = \tan^{-1}\left(\frac{r_{3,2}}{r_{3,3}}\right) \quad \beta = \tan^{-1}\left(\frac{-r_{3,1}}{r_{3,2}/\text{sen}\gamma}\right) \quad \alpha = \tan^{-1}\left(\frac{r_{2,1}}{r_{1,1}}\right) \quad (1.3)$$

La matriz de rotación (1.2) se estima con las mediciones de los acelerómetros y magnetómetros del sensor *MPU9250* que está montado en el modelo anatómico de rodilla, Figura 10. Los acelerómetros miden principalmente la aceleración de la gravedad g del planeta. Los magnetómetros miden la magnitud del campo magnético terrestre, principalmente el vector del norte magnético n .

Si las coordenadas del punto ${}^B P$ de la ecuación (1.1) son los valores de los acelerómetros del *MPU9250*, entonces el punto ${}^A P$ representa al vector conocido de la gravedad g , que apunta hacia el eje Z negativo del sistema de referencia A . Por lo que la transformación inversa de la ecuación (1.1) es:

$${}^B P = {}^A R^{-1} \cdot {}^A P = \begin{bmatrix} Ac_x \\ Ac_y \\ Ac_z \end{bmatrix} = \begin{bmatrix} r_{1,1} & r_{2,1} & r_{3,1} \\ r_{1,2} & r_{2,2} & r_{3,2} \\ r_{1,3} & r_{2,3} & r_{3,3} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} \quad (1.4)$$

Los componentes $r_{3,1}$, $r_{3,2}$ y $r_{3,3}$ de la matriz (1.2) se estiman con las mediciones Ac_x , Ac_y y Ac_z de los acelerómetros, a partir de la ecuación (1.4):

$$\begin{bmatrix} r_{3,1} \\ r_{3,2} \\ r_{3,3} \end{bmatrix} = \frac{1}{-g} \begin{bmatrix} Ac_x \\ Ac_y \\ Ac_z \end{bmatrix} \quad (1.5)$$

Los magnetómetros del sensor *MPU9250* también son utilizados para estimar algunos componentes de la matriz de rotación (1.2), para ello se calcula la parte perpendicular del vector medido por los acelerómetros $Ac = \{Ac_x, Ac_y, Ac_z\}$ sobre el vector medido por los magnetómetros $m = \{m_x, m_y, m_z\}$. Primero se proyecta el vector Ac sobre el vector m :

$$\text{Proy } m_{Ac} = \frac{m \cdot Ac}{|Ac|^2} Ac \quad (1.6)$$

Con la ecuación (1.6) se obtiene un vector M perpendicular al vector Ac :

$$M = m - \text{Proy } m_{Ac} \quad (1.7)$$

Si las coordenadas del vector ${}^B P$ de la ecuación (1.1) son las coordenadas del vector M de la ecuación (1.7), entonces el punto ${}^A P$ representa a la parte del vector n que es perpendicular al vector g y que apunta hacia el eje Y negativo del sistema de referencia A . Por lo que la transformación inversa de la ecuación (1.1) es:

$${}^B P = {}^A R^{-1} \cdot {}^A P = \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} r_{1,1} & r_{2,1} & r_{3,1} \\ r_{1,2} & r_{2,2} & r_{3,2} \\ r_{1,3} & r_{2,3} & r_{3,3} \end{bmatrix} \begin{bmatrix} 0 \\ -n \\ 0 \end{bmatrix} \quad (1.8)$$

Los componentes $r_{2,1}$, $r_{2,2}$ y $r_{2,3}$ de la matriz (1.2) se estiman con los componentes del vector M , a partir de la ecuación (1.8):

$$\begin{bmatrix} r_{2,1} \\ r_{2,2} \\ r_{2,3} \end{bmatrix} = \frac{1}{-n} \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} \quad (1.9)$$

Los componentes $r_{1,1}$, $r_{1,2}$ y $r_{1,3}$ de la matriz (1.2) se estiman con el producto cruz de los componentes en (1.5) y (1.9), ya que la matriz de rotación (1.2) es ortonormal.

$$\begin{bmatrix} r_{1,1} \\ r_{1,2} \\ r_{1,3} \end{bmatrix} = \begin{bmatrix} r_{2,1} \\ r_{2,2} \\ r_{2,3} \end{bmatrix} \otimes \begin{bmatrix} r_{3,1} \\ r_{3,2} \\ r_{3,3} \end{bmatrix} \quad (1.10)$$

2.2.2 Filtro complementario para la estimación de los ángulos de Euler

El *filtro complementario* combina las mediciones de los acelerómetros, magnetómetros y giróscopos [48]. Se aplican *filtros pasa bajas* al cálculo de los ángulos a partir de los acelerómetros y magnetómetros. Se aplican *filtros pasa altas* a los ángulos calculados con los giróscopos. Los filtros descritos anteriormente tienen la misma frecuencia de corte, Figura 11.

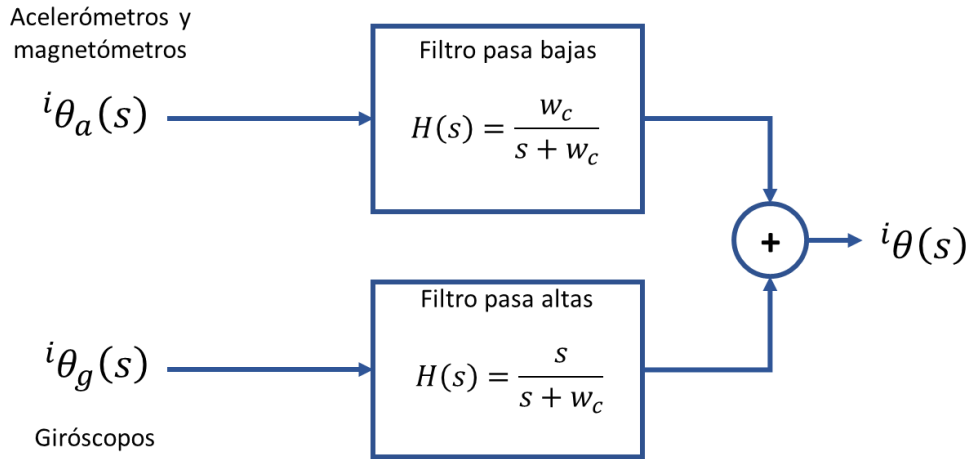


Figura 11. Aplicación del filtro complementario para combinar las señales de los acelerómetros, magnetómetros y giróscopos del sensor MPU9250, montado en el modelo anatómico de rodilla.

A continuación, se presentan los cálculos del *filtro complementario*, basados en [49] [50].

El *filtro complementario*, de la Figura 11, se expresa como:

$$i\theta_a(s) \left(\frac{w_c}{s+w_c} \right) + i\theta_g(s) \left(\frac{s}{s+w_c} \right) = i\theta(s) \quad (1.11)$$

donde:

i es el eje de giro X, Y o Z.

$i\theta(s)$ es el ángulo filtrado en el eje i , para tiempo continuo.

$i\theta_a(s)$ es el ángulo calculado con los acelerómetros y magnetómetros, en el eje i , para tiempo continuo.

$i\theta_g(s)$ es el ángulo calculado con los giróscopos, en el eje i , para tiempo continuo.

w_c es la frecuencia de corte en radianes.

Para programar el *filtro complementario* se requiere discretizar la ecuación (1.11), por lo que, primero se aplica un retenedor de orden cero:

$$i\theta_a(s) \left(\frac{w_c}{s+w_c} \right) \left(\frac{1-e^{-T_0s}}{s} \right) + i\theta_g(s) \left(\frac{s}{s+w_c} \right) \left(\frac{1-e^{-T_0s}}{s} \right) = i\theta(s) \left(\frac{1-e^{-T_0s}}{s} \right) \quad (1.12)$$

donde:

T_0 es el tiempo entre muestras.

Después, se aplica la transformada z a la ecuación (1.12):

$$(1 - z^{-1})^i \theta_a(z) \left(\frac{(1 - e^{-W_c T_o}) z^{-1}}{1 - z^{-1} (1 - e^{-W_c T_o} z^{-1})} \right) + {}^i \theta_g(z) \left(\frac{(1 - z^{-1})}{1 - e^{-W_c T_o} z^{-1}} \right) = (1 - z^{-1})^i \theta(z) \frac{1}{1 - z^{-1}} \quad (1.13)$$

donde:

${}^i \theta(z)$ es el ángulo filtrado en el eje i , para tiempo discreto.

${}^i \theta_a(z)$ es el ángulo obtenido con los acelerómetros y magnetómetros, en el eje i , para tiempo discreto.

${}^i \theta_g(z)$ es el ángulo obtenido con los giróscopos, en el eje i , para tiempo discreto.

Simplificando la ecuación (1.13) se obtiene:

$$e^{-W_c T_o} {}^i \theta(z) z^{-1} + (1 - e^{-W_c T_o}) {}^i \theta_a(z) z^{-1} + {}^i \theta_g(z) - {}^i \theta_g(z) z^{-1} = {}^i \theta(z) \quad (1.14)$$

Ahora, se aplica la anti transformada z a la ecuación 1.14:

$$e^{-W_c T_o} {}^i \theta(k - 1) + (1 - e^{-W_c T_o}) {}^i \theta_a(k - 1) + {}^i \theta_g(k) - {}^i \theta_g(k - 1) = {}^i \theta(k) \quad (1.15)$$

donde:

${}^i \theta(k)$ es el ángulo del eje i , en el tiempo discreto k .

${}^i \theta(k - 1)$ es el ángulo del eje i , en el tiempo discreto $(k - 1)$.

${}^i \theta_a(k)$ es el ángulo del eje i , calculado a partir de los acelerómetros y magnetómetros, en el tiempo discreto k .

${}^i \theta_a(k - 1)$ es el ángulo del eje i , calculado a partir de los acelerómetros y magnetómetros, en el tiempo discreto $(k - 1)$.

${}^i \theta_g(k)$ es el ángulo del eje i , calculado a partir de los giróscopos, en el tiempo discreto k .

${}^i \theta_g(k - 1)$ es el ángulo del eje i , calculado a partir de los giróscopos, en el tiempo discreto $(k - 1)$.

Sabiendo que ${}^i\theta_g(k) - {}^i\theta_g(k-1) = T_o {}^i\dot{\theta}_g(k)$, se expresa la ecuación (1.15) como:

$${}^i\theta(k) = e^{-W_c T_o} {}^i\theta(k-1) + (1 - e^{-W_c T_o}) {}^i\theta_a(k-1) + T_o {}^i\dot{\theta}_g(k) \quad (1.16)$$

Con la ecuación (1.16) se programa el *filtro complementario*.

2.2.3 Redes neuronales

Una red neuronal artificial está basada en una red neuronal biológica, donde las neuronas son la unidad de procesamiento fundamental, y se comunican con otras por medio de la sinapsis [51].

La neurona genérica artificial consta de un proceso en dos etapas. La primera es una combinación lineal independiente de las entradas con sus respectivos pesos, que es una contribución sináptica individual. En la segunda etapa se utiliza el resultado de la combinación lineal como entrada de una función no lineal para dar lugar a la salida final de la neurona, Figura 12.

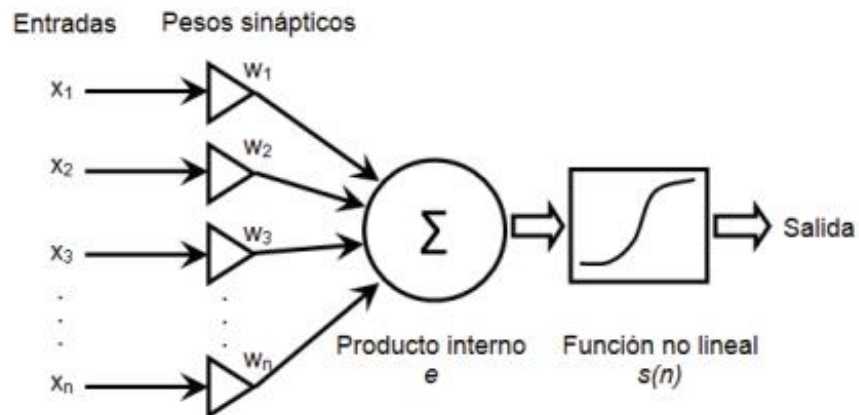


Figura 12. Modelo de una neurona genérica, tomada de [52].

La arquitectura de una red neuronal es la estructura del conexionado de dicha red. Estas conexiones suelen agruparse en unidades estructurales llamadas capas, Figura 13. Existen tres tipos de capas, de entrada, ocultas y de salida.

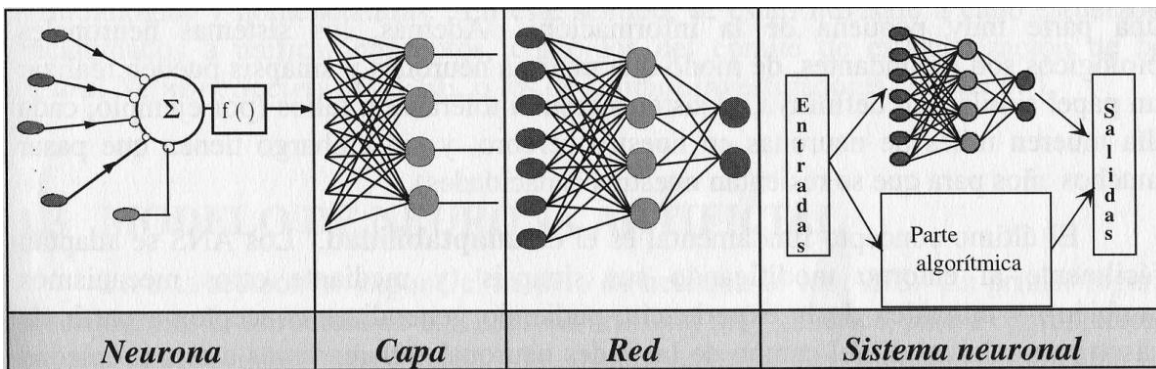


Figura 13. Modelo de una red neuronal, tomada de [53].

La capa de entrada o sensorial se compone de neuronas que reciben datos del entorno. La capa oculta proporciona los grados de libertad a la red, de esta manera es capaz de representar fehacientemente determinadas características del entorno que se está modelando. La capa de salida proporciona las respuestas de la red neuronal [53].

2.2.4 Clasificador de umbrales aleatorios RTC

Existen muchas estructuras de redes neuronales que proporcionan buenos resultados, pero tienen una baja velocidad de procesamiento o de entrenamiento. La estructura del clasificador de umbrales aleatorios (*Random Threshold Classifier, RTC*), Figura 14, posee un alto rendimiento en el procesamiento y entrenamiento [54].

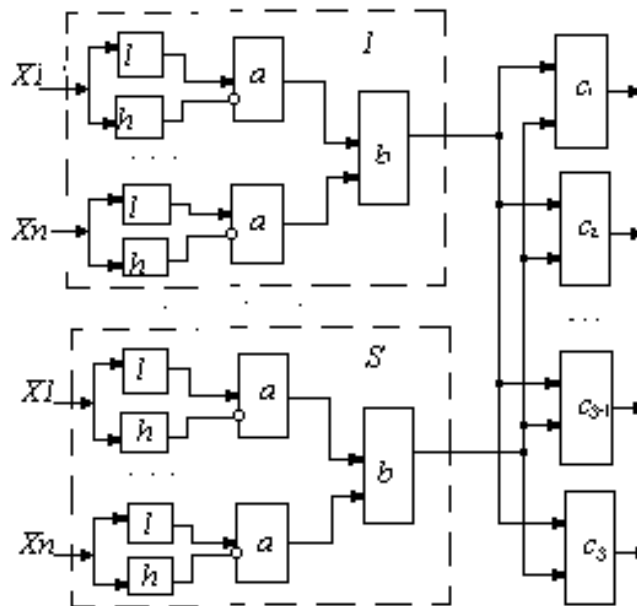


Figura 14. Estructura del clasificador de umbrales aleatorios RTC, tomada de [54].

La arquitectura del *RTC* consiste en una sola capa de conexiones de entrenamiento, por lo que se obtiene una alta velocidad durante dicho entrenamiento. Además, tiene otras capas no modificables de conexiones y neuronas binarias que hacen transformaciones no lineales de parámetros de entrada espaciales, a parámetros espaciales binarios, teniendo muy altas dimensiones.

El vector de entradas (X_1, \dots, X_n) son el juego de características del fenómeno estudiado, cada característica se conecta con dos neuronas l y h . La salida excitada de l y desactivada de h se conectan a una neurona a , de tal forma que representan a un umbral que activa a la neurona a únicamente cuando la característica X_n se encuentra dentro de dicho umbral. Todas las neuronas a de un grupo se conectan a una neurona b que se activa cuando todas las neuronas a son excitadas.

Las neuronas b representan la salida de todo el grupo neuronal, todas estas neuronas se conectan con todas las neuronas c por medio de conexiones entrenables, donde se involucra el peso w_i que se modifica por las neuronas b durante el entrenamiento de la red.

La representación geométrica de la primera etapa de la red neuronal se muestra en la Figura 15, considerando que se estudian dos características $(X_1$ y $X_2)$ del fenómeno. En el caso de más de dos características, en vez de rectángulos se tienen paralelepípedos multidimensionales.

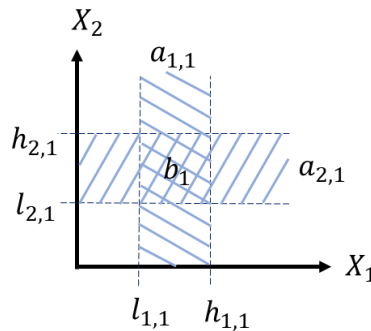


Figura 15. Identificación geométrica de la neurona b de la red neuronal *RTC*.

El clasificador *RTC* es aplicado en diversos problemas de reconocimiento de texturas naturales, escritura, palabras y señales, obteniendo buenos resultados [55, 56]. Con algunas modificaciones en su estructura, esta red neuronal artificial puede ser utilizada para la clasificación de los usuarios del simulador artroscópico presentado en este trabajo.

CAPÍTULO 3. INSTRUMENTACIÓN ELECTRÓNICA

3.1 Requerimientos para el arreglo instrumental

Una cirugía artroscópica de rodilla es un procedimiento médico donde se insertan dos instrumentales quirúrgicos a través de los puertos de entrada hechos en la articulación, Figura 16. La rodilla se posiciona conforme al procedimiento quirúrgico a realizar. En una simulación de cirugía artroscópica de rodilla se requiere la orientación y posición de la articulación y del instrumental quirúrgico insertado en los puertos de entrada.

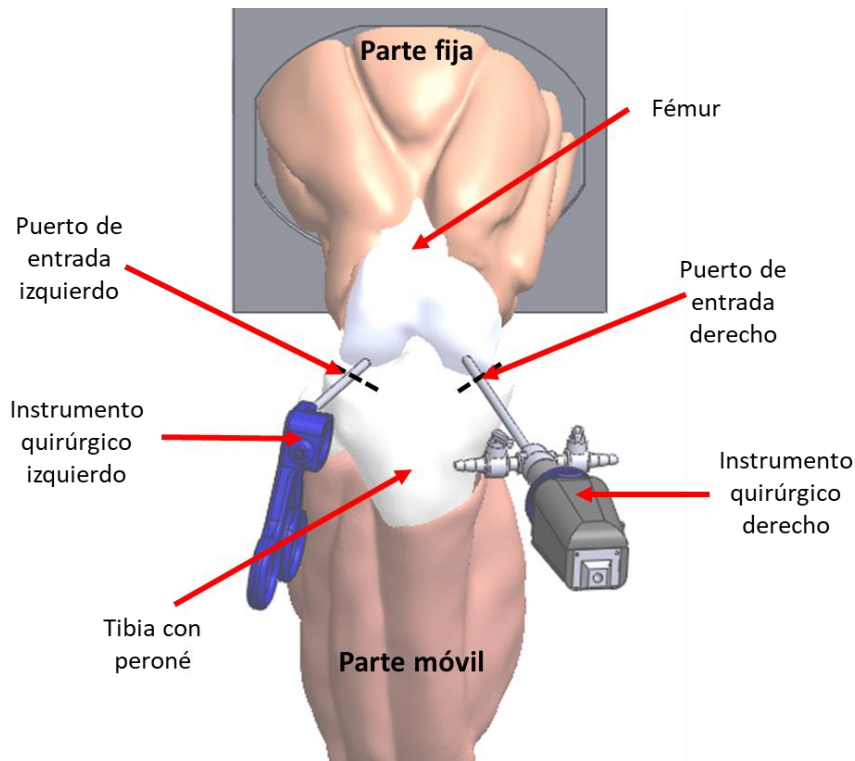


Figura 16. Instrumental quirúrgico insertado en la rodilla durante una cirugía artroscópica.

El arreglo instrumental requiere medir las siguientes variables:

1. El movimiento de la articulación de rodilla, con el fémur inmóvil y con cuatro grados de libertad en la tibia: tres giros y un desplazamiento.
2. El movimiento del instrumental quirúrgico insertado en el puerto izquierdo y derecho. Cada instrumental tiene cuatro grados de libertad, tres giros y un desplazamiento.
3. El instrumental quirúrgico insertado.
4. La orientación del lente del artroscopio.
5. La apertura de la pinza de corte.

En la Figura 17 se presenta un esquema con el resumen de las variables físicas que se requieren medir con el arreglo instrumental.

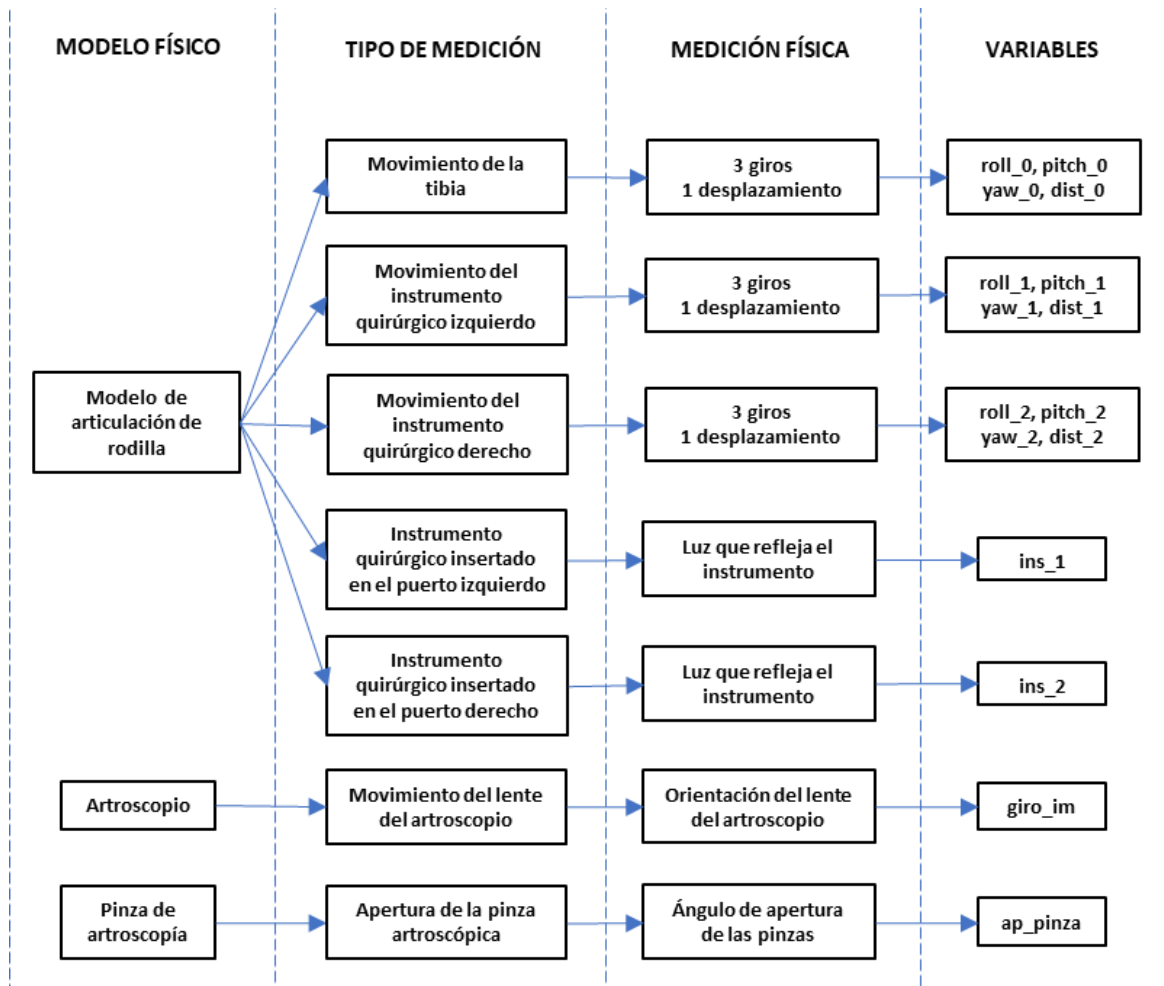


Figura 17. Esquema de las variables físicas que mide el arreglo instrumental.

Para seleccionar los sensores del arreglo instrumental se consideran principalmente tres etapas. La primera etapa consiste en definir los requerimientos de las mediciones, tales como, el rango, resolución y dimensiones, Figura 18. En la segunda etapa se realiza una búsqueda de los sensores que cumplen con los requerimientos de la etapa anterior. En la última etapa se elabora una tabla comparativa para seleccionar el sensor.

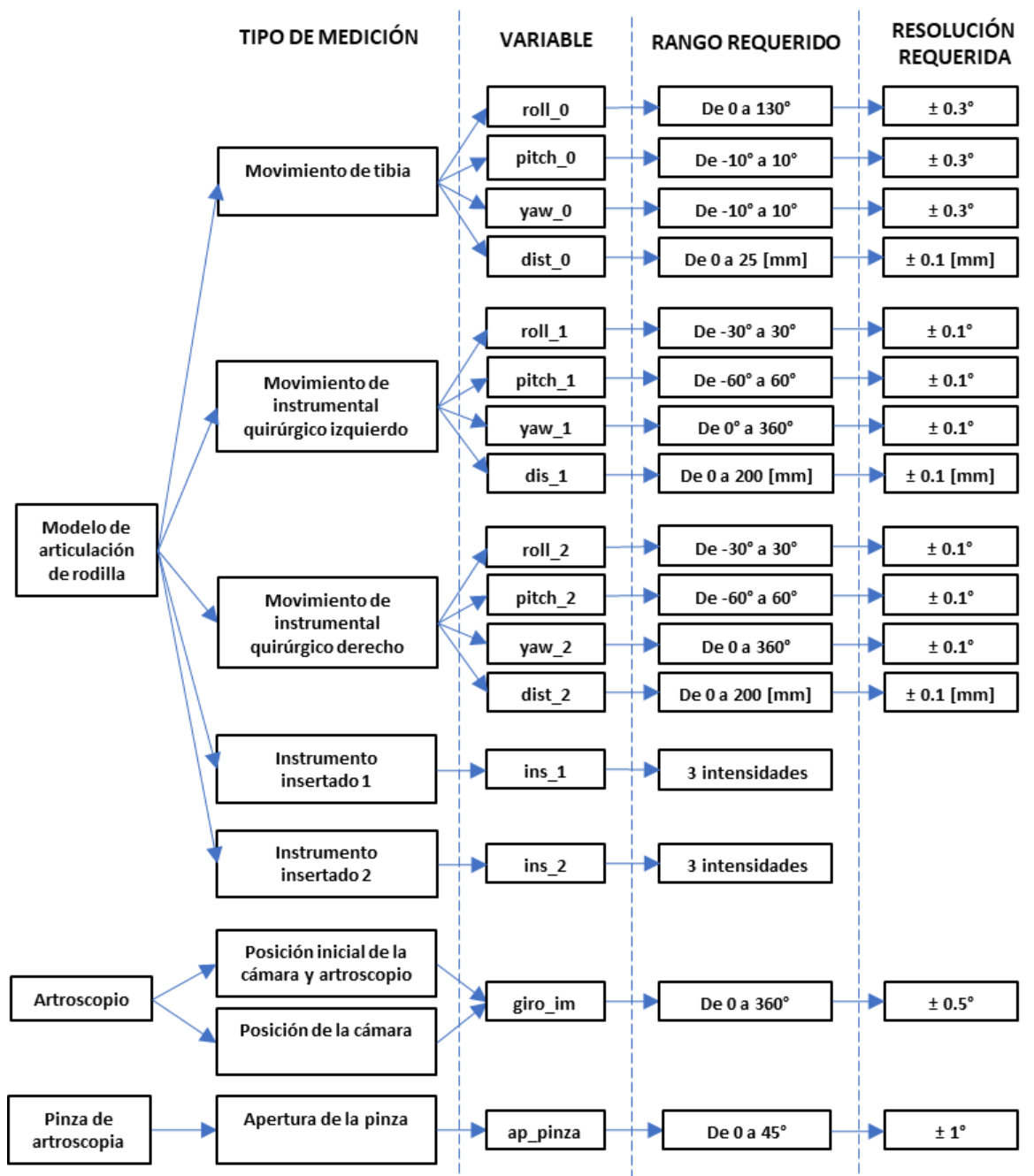


Figura 18. Requerimientos de los sensores para medir las variables físicas en el arreglo instrumental.

Una vez definidos los requerimientos de las mediciones, se proponen sensores que cumplen con dichos requerimientos y se realizan tablas comparativas, Tabla 2 y Tabla 3.

Para medir el movimiento del modelo de rodilla se utilizan sensores *IMU*, ya que son utilizados comúnmente en tareas de seguimiento de movimiento. En este trabajo se propone el uso de un sensor *IMU* para medir los movimientos de la articulación de rodilla.

En la Tabla 2 se muestra la información de los diferentes sensores *IMU*, que tienen nueve grados de libertad, que son accesibles en su costo y disponibilidad.

Tabla 2. Sensores *IMU*.

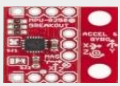




Nombre y Fabricante	Resolución	Rango	Comunicación y velocidad	Voltaje de operación [V]	Tamaño [mm]
<i>MPU9250</i> <i>Sparkfun</i>	Acel: 16 bits Gir: 16 bits Mag: 14 bits	Acel: ± 16 g Gir: ± 2000 dps Mag: ± 4800 μT	<i>I²c</i> : 400 KHz SPI: 1 Mhz	(1.9 - 3.6)	(26 x 16)
<i>MPU9150</i> <i>Sparkfun</i>	Acel: 16 bits Gir: 16 bits Mag: 13 bits	Acel: ± 16 g Gir: ± 2000 dps Mag: ± 1200 μT	<i>I²c</i> : 400 KHz	(2.37 - 3.46)	(16 x 28)
<i>BNO055</i> <i>Adafruit</i>	Acel: 14 bits Gir: 16 bits Mag: 15 bits	Acel: ± 16 g Gir: ± 2000 dps Mag: ± 2500 μT	<i>I²c</i> : 400 KHz UART: 115.2 Kbps	(1.7 – 3.6)	(27 x 20)
<i>LSM9DS1</i> <i>Parralax inc</i>	Acel: 16 bits Gir: 16 bits Mag: 16 bits	Acel: ± 16 g Gir: ± 2000 dps Mag: ± 1600 μT	<i>I²c</i> : 400 KHz SPI: 10 Mhz	(2.6 – 6.5)	(22x30)
<i>10 DOF MEMS</i> <i>DFRobot</i>	Acel: 16 bits Gir: 16 bits Mag: 12 bits	Acel: ± 16 g Gir: ± 2000 dps Mag: ± 800 μT	<i>I²c</i> : 400 KHz	(3 – 8)	(26x18)
<i>Minim U-9 v5</i> <i>Pololu</i>	Acel: 16 bits Gir: 16 bits Mag: 16 bits	Acel: ± 16 g Gir: ± 2000 dps Mag: ± 1600 μT	<i>I²c</i> : 400 KHz	(2.5 – 5.5)	(20x13)

El sensor *IMU* es mejor mientras las resoluciones y los rangos de sus sensores sean mayores, mientras tenga más medios de comunicación y con mayor velocidad, así mismo, mientras su tamaño sea menor. Para seleccionar al mejor sensor *IMU* se requieren más datos importantes, por ejemplo, el ruido y estabilidad de sus señales bajo determinadas condiciones de operación.

En la etapa actual de la investigación del presente trabajo se requiere validar el funcionamiento de los sensores *IMU* para medir los movimientos de una articulación de rodilla, por ello, únicamente se selecciona al sensor *IMU* a partir de los datos de la Tabla 2. El sensor *MPU9250* es elegido para medir los movimientos del modelo de rodilla, ya que sus magnetómetros tienen el rango más amplio, las resoluciones de sus señales son mayores que en los demás sensores *IMU*, cuenta con dos tipos de comunicación, es el segundo de menor tamaño y es el más accesible al momento de realizar este trabajo.

Para seleccionar los sensores que miden las variables de movimiento de la Figura 18, se consideran los siguientes datos: dimensiones, resolución y rango, Tabla 3. Los sensores de menor tamaño permiten mayor movimiento en el modelo de rodilla e instrumental quirúrgico. Una resolución cada vez mayor implica una detección de movimientos más pequeños. Un rango mayor en los sensores, permite un rango mayor en la medición de las variables de movimiento.

Tabla 3. Comparación de los sensores para medir movimiento.

Nombre	Tecnología	Resolución	Rango	Comunicación	Tamaño [mm]
 MPU9250	Efecto <i>hall</i> Inercial	Acel: 16 <i>bits</i> Gir: 16 <i>bits</i> Mag: 14 <i>bits</i>	Acel: $\pm 16\text{ g}$ Gir: $\pm 2000\text{ dps}$ Mag: $\pm 4800\ \mu\text{T}$	<i>I²c</i> , <i>SPI</i>	(26x16x8)
 E4T de US Digital	<i>Encoder</i> óptico	de 400 a 4,000 <i>ppr</i>	360°	2 pin	(\varnothing 22x15)
 AS5600 de AMS	<i>Encoder</i> magnético	12 <i>bits</i>	360°	<i>I²c</i> , análogo	(6x5x1.75)
 E6B2-C OMRON	<i>Encoder</i> rotativo capacitivo	720 <i>ppr</i>	360°	4 pin	(\varnothing 40x59)
 KY-040	<i>Encoder</i> rotativo	20 <i>ppr</i>	360°	3pin	(27x19x27)

En la Figura 19 se muestra la propuesta instrumental que tiene los siguientes componentes:

1. Un sensor *MPU9250-1* para medir los movimientos de la articulación de rodilla.
2. Un sensor *MPU9250-2* y un sensor *AS5600* para medir la orientación del artroscopio.
3. Cuatro sensores *AS5600* en cada uno de los dos puertos de entrada a la rodilla, para medir los cuatro grados de libertad de cada instrumental quirúrgico. Además, un sensor *QRE1113* en cada puerto de entrada, para identificar a los instrumentos quirúrgicos a partir de la luz que reflejan.
4. Un sensor *AS5600* para medir la apertura de las pinzas de artroscopia.
5. Un microcontrolador *PIC18F4550* que recibe las señales de los sensores, aplica los filtros digitales, estima las orientaciones y los desplazamientos, y se comunica con un ambiente de realidad virtual en un equipo de cómputo.

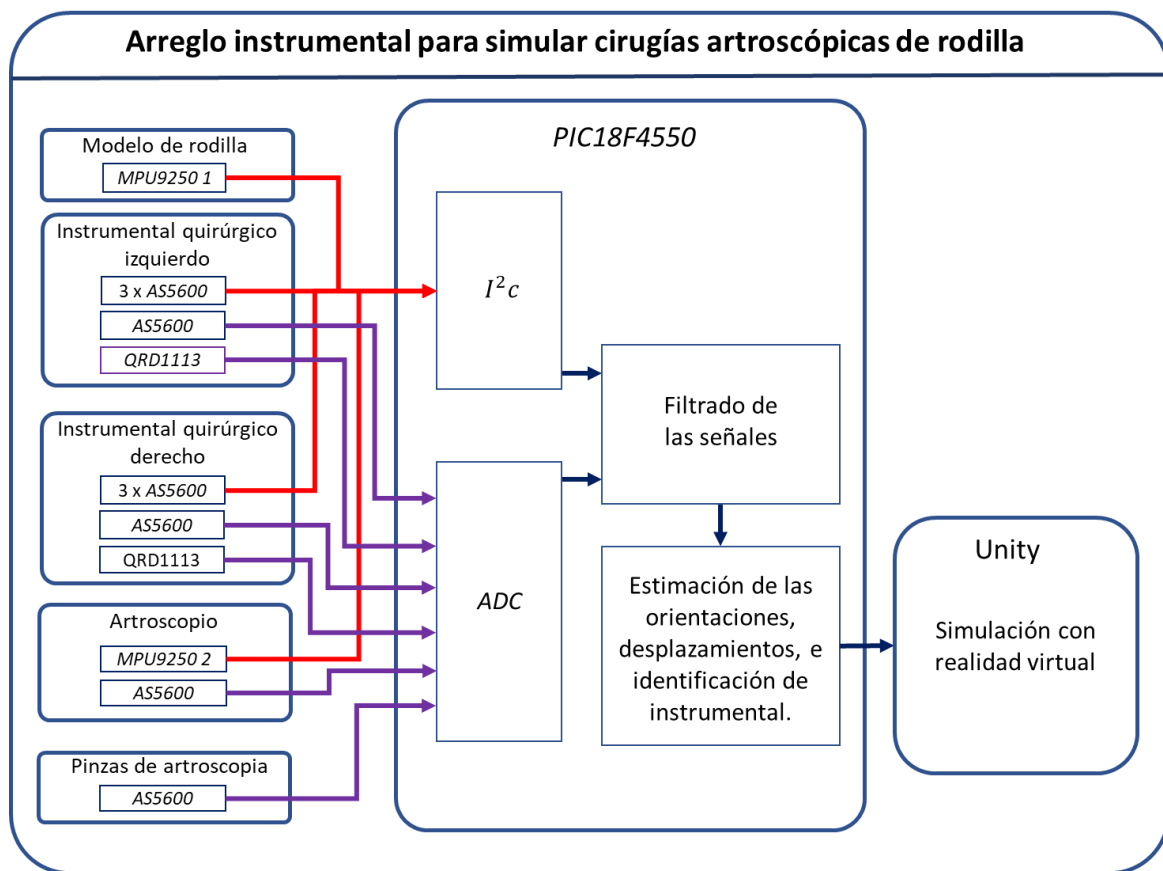


Figura 19. Integración del arreglo instrumental.

Los sensores se comunican con el microcontrolador a través del protocolo I^2C y los puertos *ADC*. El microcontrolador se configura para filtrar las señales de los sensores y calcular los movimientos de la articulación y del instrumental quirúrgico. Al final esta información se envía a través de un puerto *USB*, para manipular un ambiente virtual en *Unity*.

3.2 Componentes principales de la instrumentación electrónica

3.2.1 Sensor inercial MPU9250

Para medir el movimiento de la articulación de rodilla, se utiliza un sensor IMU MPU9250 fabricado por Sparkfun, Figura 20, que tiene las siguientes características:

- 9 grados de libertad (3 Acelerómetros, 3 Magnetómetros y 3 Giróscopos).
- Comunicación I^2C o SPI.
- Los acelerómetros tienen una escala programable de $\pm 2 [g]$, $\pm 4 [g]$, $\pm 8 [g]$ y $\pm 16 [g]$ con ADC de 16 [bits].
- Los giróscopos tienen una escala programable de $\pm 250 [^\circ/s]$, $\pm 500 [^\circ/s]$, $\pm 1,000 [^\circ/s]$ y $\pm 2,000 [^\circ/s]$ con un ADC integrado de 16 [bits].
- Los magnetómetros tienen una resolución de 14 [bits].

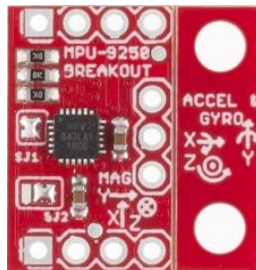


Figura 20. Sensor IMU MPU9250 de Sparkfun [57].

El sensor MPU9250 se utiliza comúnmente para estimar la orientación de objetos en tres ejes. La combinación de las señales de los acelerómetros y magnetómetros permite calcular la orientación del objeto con mayor exactitud cuando este se encuentra inmóvil, para ello, utilizan como referencia el vector de la aceleración de la gravedad y el vector del norte magnético del planeta. Los giróscopos también permiten calcular la orientación del objeto, con mejor exactitud cuando este se encuentra en movimiento.

Para estimar la orientación de un objeto, se combinan las señales de los acelerómetros, magnetómetros y giróscopos. Para dicha tarea se implementa un filtro digital, entre los más comunes están el *filtro Kalman* y el *filtro complementario* descrito con anterioridad.

3.2.2 Sensor de efecto hall AS5600

Para medir los movimientos del instrumental quirúrgico se utilizan diez sensores de efecto hall AS5600 fabricados por AMS, Figura 21. Dicho sensor tiene las siguientes características:

- Mide el giro de un imán radial, con un rango de 0 a 360°.
- Comunicación I^2C o PWM.
- Resolución de 12 [bits].
- El encapsulamiento es SOIC-8.

El sensor *AS5600* utiliza un imán radial (norte-sur), para detectar el campo magnético y su movimiento, el cual tiene un diámetro de 6 [mm] y una altura de 2.5 [mm]. Este imán debe incorporarse al elemento mecánico giratorio, lo que limita las dimensiones del mecanismo donde se integran estos sensores.

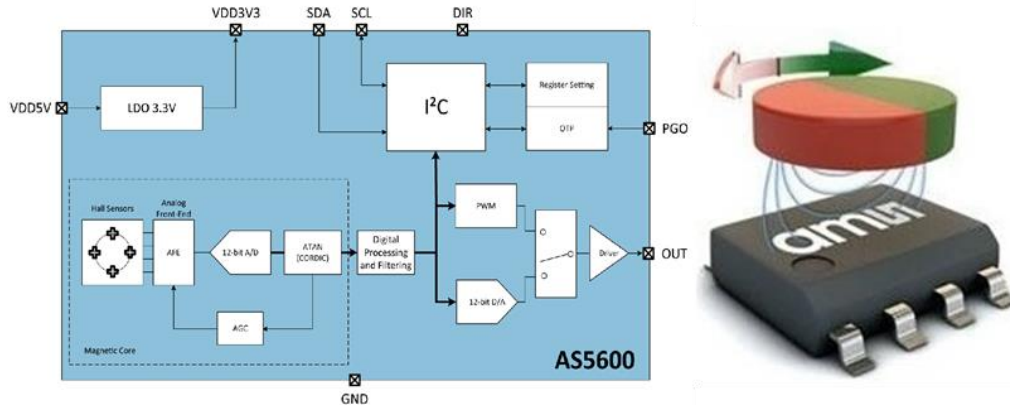


Figura 21. Sensor *AS5600* de AMS [58].

El sensor *AS5600* se comunica por un bus I^2c o por un puerto *PWM*. Cuando el sensor *AS5600* se comunica con el protocolo I^2c se obtiene una resolución de 12 [bits] o 0.087°. Cuando el sensor se conecta por el puerto *PWM* a un convertidor *ADC* se agrega ruido a la señal en el proceso de conversión, por lo tanto, disminuye su precisión, además la resolución del sensor se limita por los bits del convertidor *ADC*.

La dirección del sensor *AS5600* es 0x36 para la comunicación por el protocolo I^2c , por lo que se requiere multiplexar las señales *SDA* y *SCL* para comunicarse con más de un sensor *AS5600* por un solo bus I^2c del microcontrolador.

3.2.3 Sensor reflectivo *QRE1113*

El *QRE1113* es un sensor reflectivo fabricado por *ON Semiconductor* que mide la intensidad de luz reflejada en una superficie, a una distancia máxima de 5 [mm]. Este sensor se compone de un diodo de luz infrarroja que emite luz hacia una superficie y un fototransistor que recibe la luz reflejada, Figura 22.

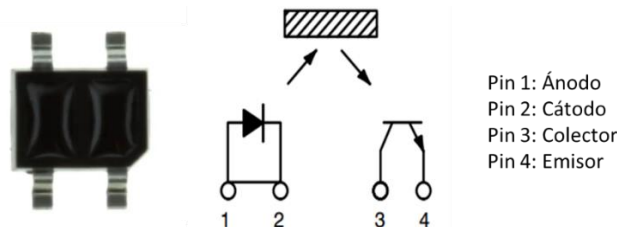


Figura 22. Sensor *QRE1113* [59].

El emisor se conecta a un convertidor *ADC*, por lo que su precisión depende de los *bits* de dicho convertidor. La señal que proporciona el *QRE1113* es ruidosa, por lo que se requiere de una etapa de filtrado.

3.2.4 Microcontrolador *PIC18F4550*

El microcontrolador seleccionado es un *PIC18F4550* fabricado por *Microchip*, Figura 23, tiene las siguientes características:

- Módulo *USB 2.0*.
- 35 pines *I/O* disponibles.
- 13 canales *ADC* de 10 [*bits*].
- *EEPROM* de datos de 256 [*Byte*].
- Oscilador externo de dos modos hasta 48 [*MHz*].
- Oscilador interno seleccionable entre 8 frecuencias desde 31 [*kHz*] hasta 8 [*MHz*].
- Comunicación *EUSART*, *SPP*, *SPI*, *I²c*.
- Memoria de programa flash de 32 [*kB*].
- Voltaje de operación 4.2 [*V*] a 5.5 [*V*].
- Encapsulado *DIP* de 40 pines.

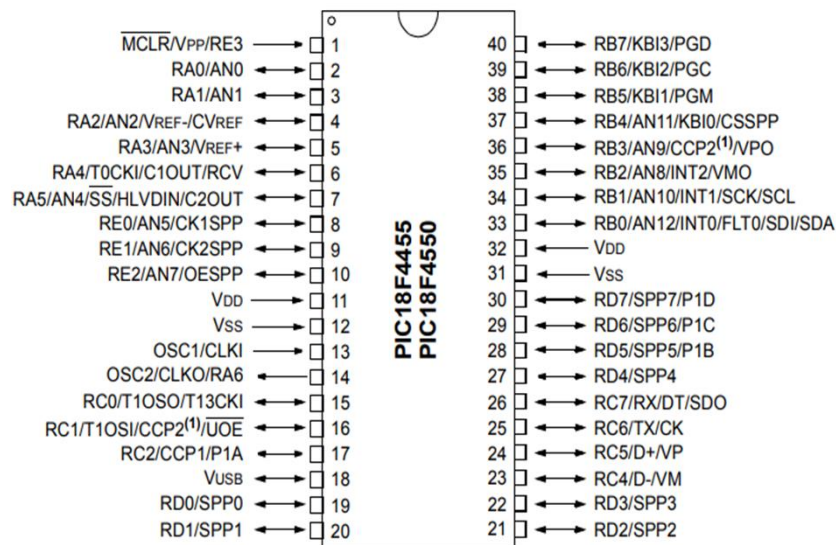


Figura 23. Microcontrolador *PIC18F4550* [60].

El *PIC18F4550* cumple con los requerimientos para el arreglo de sensores, ya que se utilizan 6 canales *ADC* de 10 [*bits*], comunicación *I²c*, y un módulo *USB* que simplifica la conexión del arreglo instrumental con la interfaz gráfica en una computadora.

3.3 Funcionamiento del arreglo instrumental

El arreglo instrumental tiene nueve módulos mecánicos, Figura 24. Los sensores y circuitos se integran a los módulos mecánicos como se muestra en la Figura 25. Los diagramas de los circuitos se encuentran en el Anexo A.

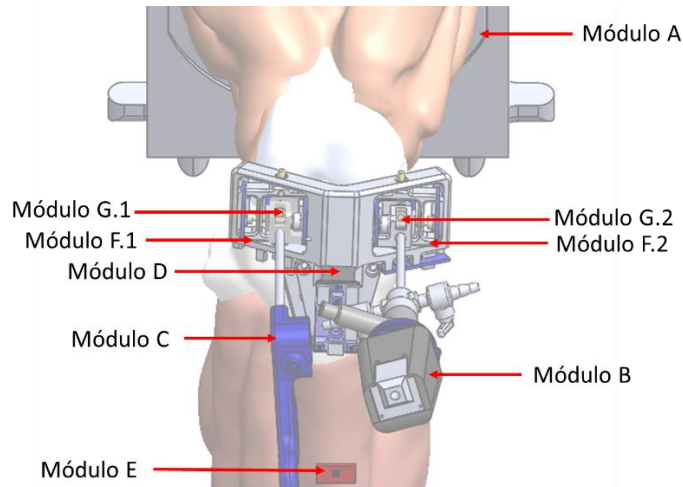


Figura 24. Módulos mecánicos del arreglo instrumental.

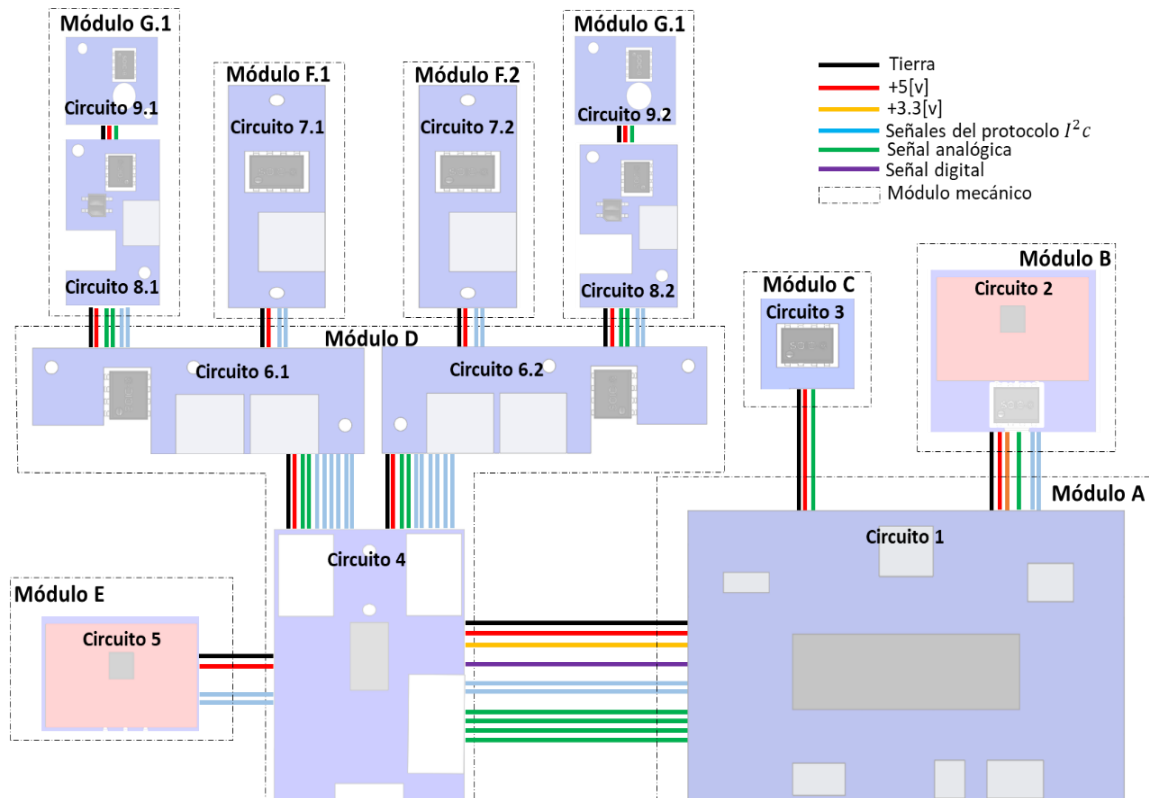


Figura 25. Integración de los sensores y los circuitos en el arreglo instrumental.

3.3.1 Mediciones de los movimientos del modelo de rodilla

El modelo de rodilla se compone de los fragmentos de un fémur, una tibia y un peroné, además de los ligamentos, meniscos, rótula y musculatura, Figura 26. El fémur se mantiene fijo y alineado al sistema coordenado A , mientras que la tibia con el peroné se mueve alineado al sistema coordenado B .

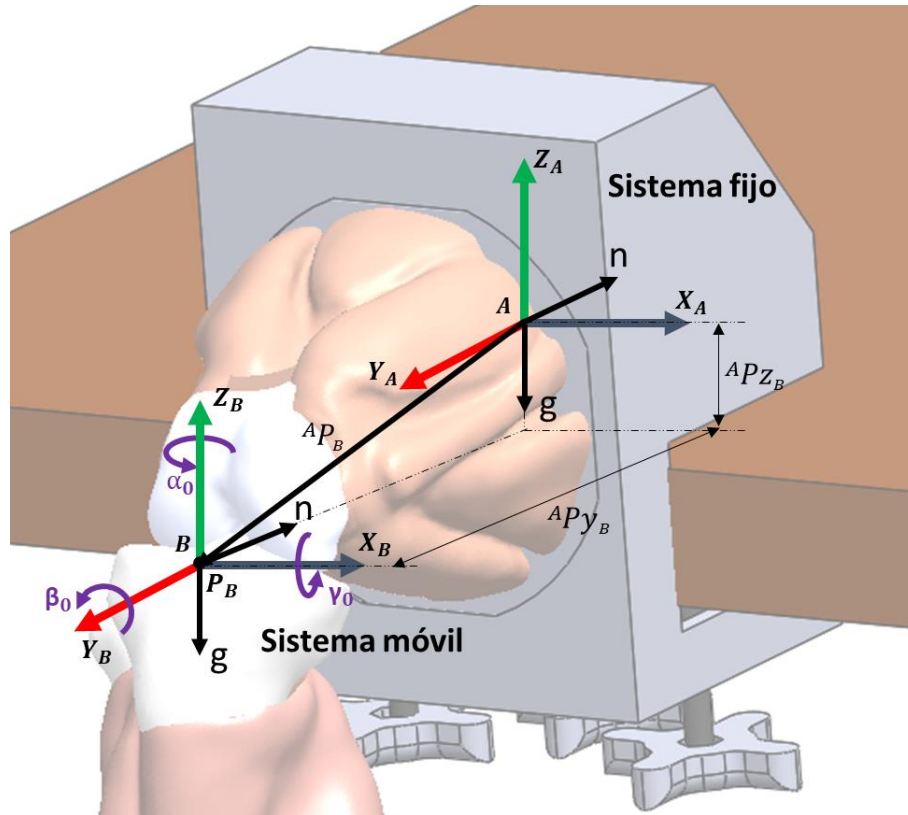


Figura 26. Sistemas de referencia para describir el movimiento de la tibia con el peroné.

La articulación de rodilla tiene 4 grados de libertad, tres giros y un desplazamiento. La orientación de la tibia se representa con los ángulos γ_0 , β_0 y α_0 , y su desplazamiento sobre el eje Z_B con δ_0 .

Un punto P_B se encuentra en el origen del sistema móvil B , sus coordenadas respecto al sistema A son:

$${}^A P_B = ({}^A P_{X_B}, 0, {}^A P_{Z_B} + \delta_0) \quad (3.1)$$

Dónde δ_0 representa el desplazamiento del fémur en el eje Z_B . Para calcular δ_0 , Figura 27, se multiplica el giro β_0 , con la constante a :

$$\delta_0 = \beta_0 * a \quad (3.2)$$

Donde a se determina experimentalmente, con un valor de $0.6 \left[\frac{mm}{grado} \right]$ Considerando que β_0 está en un rango aproximado de -10° a 13° .

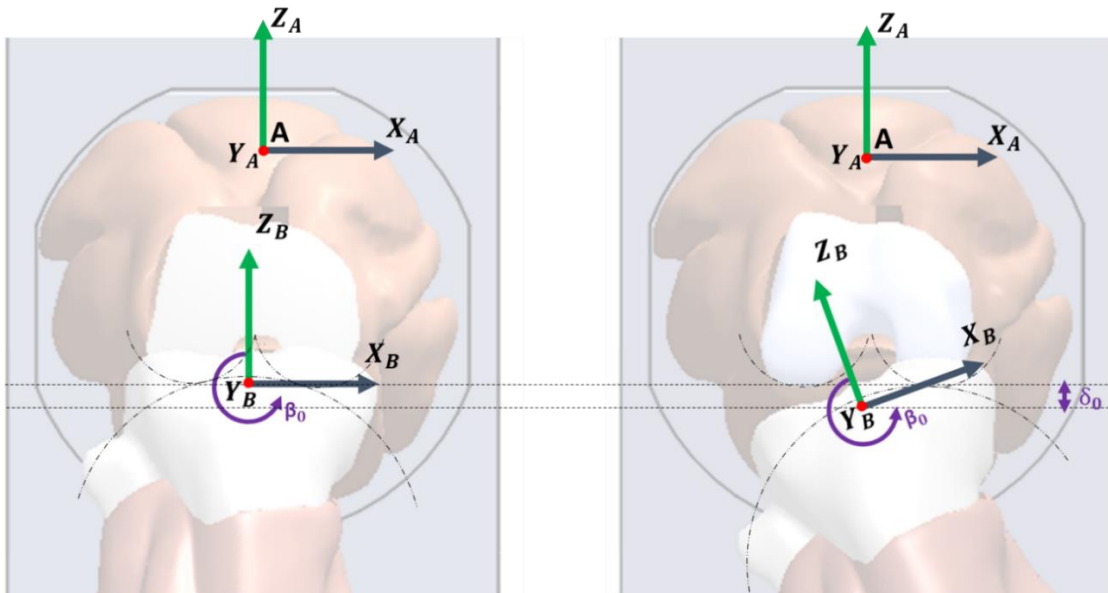


Figura 27. Cálculo de δ_0 a partir del movimiento de β_0 .

Los ángulos γ_0 , β_0 y α_0 se estiman a partir de la combinación de las señales de los sensores inerciales del MPU9250-1, ver Capítulo 2.2.1 y 2.2.2.

El circuito del sensor MPU9250-1, Figura 28, incluye el circuito integrado, capacitores de desacople y resistencias *pull-up*, conforme a las especificaciones en la hoja de datos [57]. El circuito donde se conecta el sensor MPU9250-1 requiere de una tensión de 3.3 [V] y se conecta a través de un *bus I²c* a un circuito integrado TCA9548 que funciona con una tensión de 5 [V]. Se utilizan transistores MOSFET para ajustar el nivel de tensión en el *bus I²c* de 3.3 [V] a 5 [V]. Para la miniaturización de los circuitos se utilizan conectores y cables planos del tipo FFC.

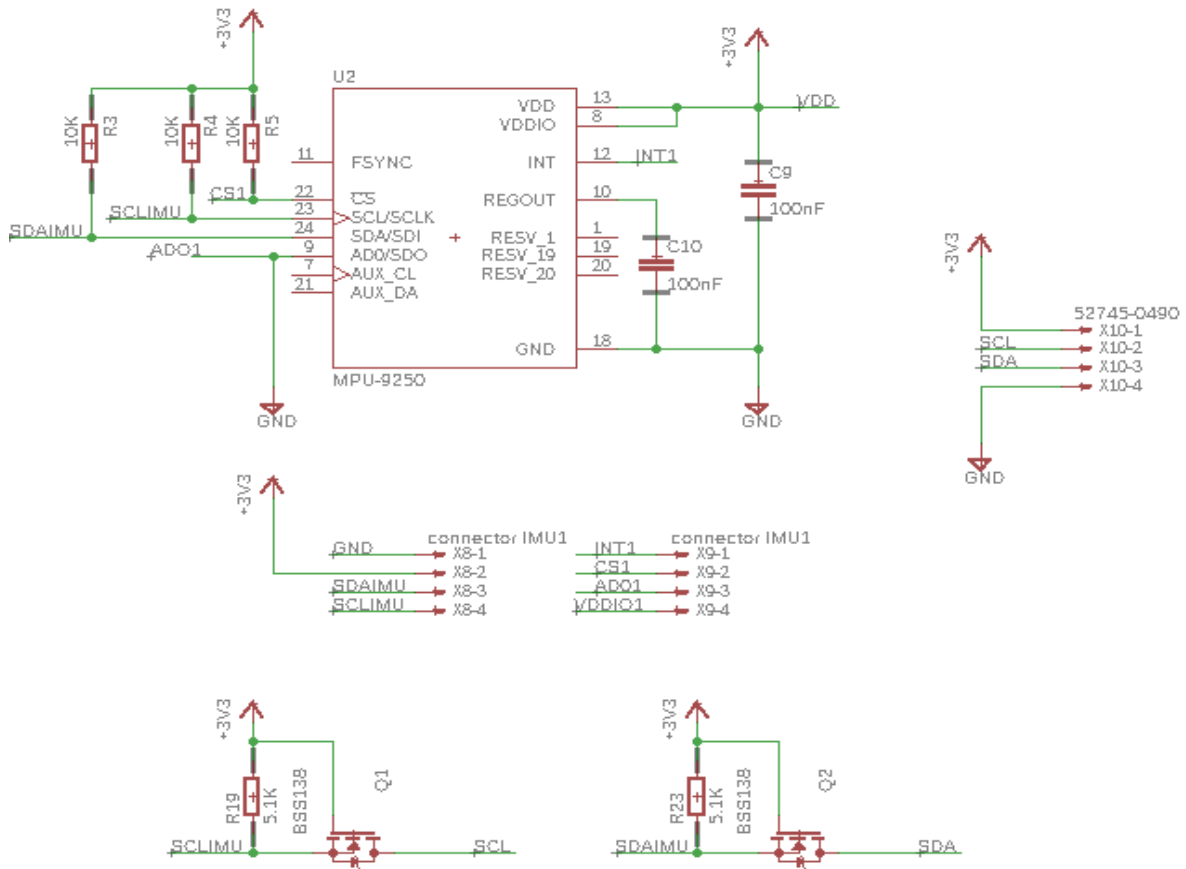


Figura 28. Circuito 5 que tiene el MPU9250-1 para medir el movimiento de la articulación.

Para calcular los grados de libertad γ_0 , β_0 , α_0 y δ_0 se programa el diagrama de flujo de la Figura 29 en el microcontrolador del arreglo instrumental.

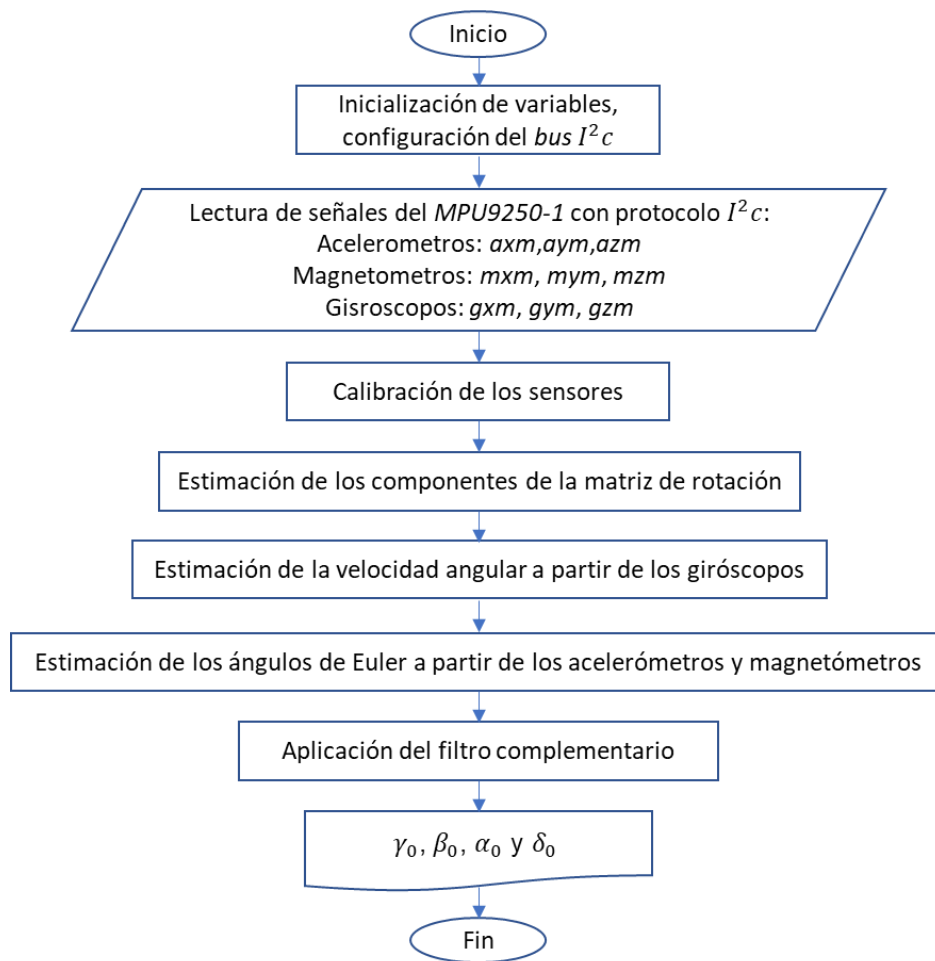


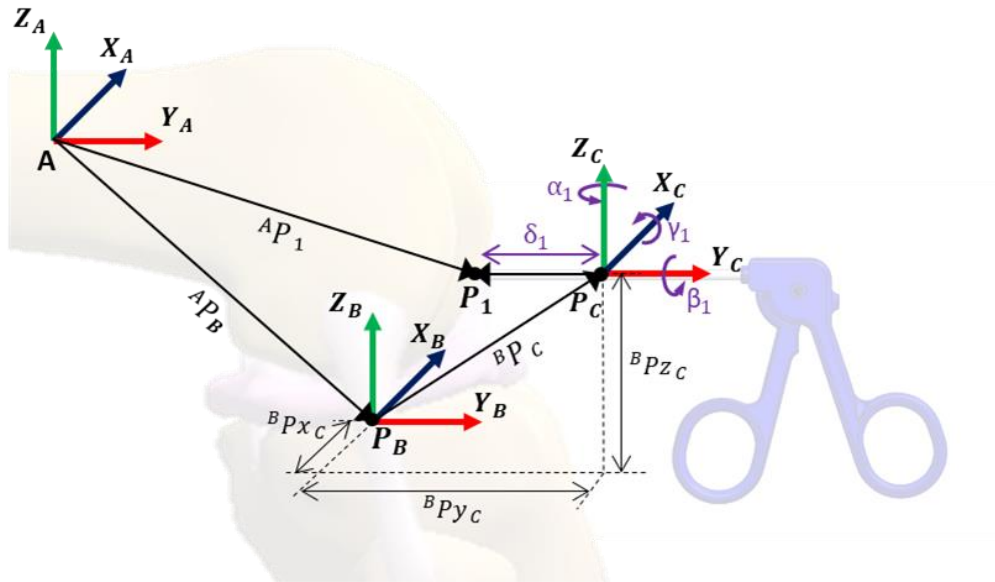
Figura 29. Diagrama de flujo para estimar los cuatro grados de libertad de la articulación de rodilla.

3.3.2 Mediciones de los movimientos del instrumental quirúrgico

Dos instrumentos quirúrgicos se introducen en la rodilla a través de los puertos de entrada. El instrumental quirúrgico insertado en el puerto izquierdo se alinea con el sistema de referencia C . El instrumental quirúrgico insertado en el puerto derecho se alinea con el sistema de referencia D , Figura 30.

El cirujano de ortopedia puede mover cada instrumento con cuatro grados de libertad, tres giros y un desplazamiento. En el puerto de entrada izquierdo se miden las variables $\gamma_1, \beta_1, \alpha_1$ y δ_1 . En el puerto de entrada derecho se miden las variables $\gamma_2, \beta_2, \alpha_2$ y δ_2 .

Puerto izquierdo



Puerto derecho

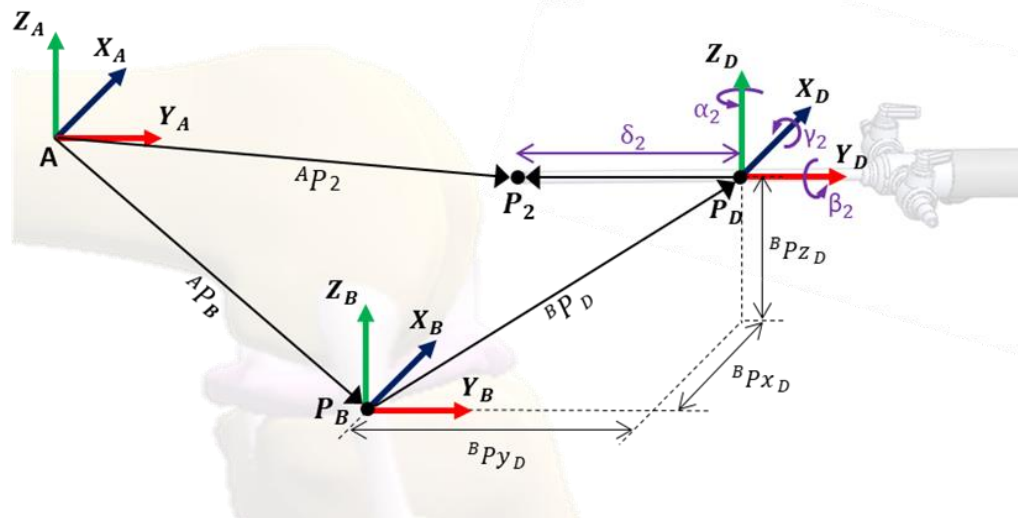


Figura 30. Estimación de la orientación y posición de los instrumentos quirúrgicos insertados en los puertos de entrada a la rodilla.

Los movimientos del instrumental quirúrgico y del modelo anatómico de rodilla se muestran en un ambiente de realidad virtual. Para ello se requieren las coordenadas de los puntos $^A P_1$ y $^A P_2$ ubicados en las puntas de los instrumentos quirúrgicos, insertados en el puerto de entrada izquierdo y derecho respectivamente.

$${}^A P_1 = {}^A P_B + {}^A R_B {}^B P_C + {}^A R_B {}^B R_C {}^C P_1 \quad (3.3)$$

$${}^A P_2 = {}^A P_B + {}^A R_B {}^B P_D + {}^A R_B {}^B R_D {}^D P_2 \quad (3.4)$$

dónde:

${}^A R_B$ es la matriz de rotación de la articulación de la rodilla, calculada a partir de los ángulos $(\gamma_0, \beta_0$ y $\alpha_0)$, ver Capítulo 2.2.1.

${}^B R_C$ es la matriz de rotación del instrumento insertado en el puerto de entrada izquierdo, calculada con los ángulos $(\gamma_1, \beta_1$ y $\alpha_1)$.

${}^B R_D$ es la matriz de rotación del instrumento insertado en el puerto de entrada derecho, calculada con los ángulos $(\gamma_2, \beta_2$ y $\alpha_2)$.

${}^A P_B$ es la posición del fémur, respecto al sistema A, determinado por las coordenadas.

$${}^A P_B = ({}^A P_{x_B}, 0, {}^A P_{z_B} + \delta_0)$$

${}^B P_C$ es el punto ubicado en el puerto de entrada izquierdo, que corresponde con el origen del sistema de referencia C, respecto al sistema de referencia B:

$${}^B P_C = ({}^B P_{x_C}, {}^B P_{y_C}, {}^B P_{z_C})$$

${}^B P_D$ es el punto ubicado en el puerto de entrada izquierdo, que corresponde con el origen del sistema de referencia C, respecto al sistema de referencia B:

$${}^B P_D = ({}^B P_{x_D}, {}^B P_{y_D}, {}^B P_{z_D})$$

${}^C P_1$ es el punto ubicado en la punta del instrumento insertado por el puerto de entrada izquierdo, con respecto al sistema de referencia C.

$${}^C P_1 = (\delta_1, 0, 0)$$

${}^D P_2$ es el punto ubicado en la punta del instrumento insertado por el puerto de entrada derecho, con respecto al sistema de referencia D.

$${}^D P_2 = (\delta_2, 0, 0)$$

Las matrices de rotación ${}^B R_C$ y ${}^B R_D$ se determinan a partir de los giros mostrados en la Figura 31. Primero un giro sobre el eje Z, después sobre el eje X y por último sobre el eje Y. Las matrices de rotación son:

$${}^B R_C = \begin{bmatrix} c\alpha_1 c\beta_1 - s\beta_1 s\gamma_1 s\alpha_1 & -c\gamma_1 s\alpha_1 & c\alpha_1 s\beta_1 + c\beta_1 s\gamma_1 s\alpha_1 \\ c\alpha_1 s\beta_1 s\gamma_1 + c\beta_1 s\alpha_1 & c\gamma_1 c\alpha_1 & -c\beta_1 c\alpha_1 s\gamma_1 + s\beta_1 s\alpha_1 \\ -c\gamma_1 s\beta_1 & s\gamma_1 & c\beta_1 c\gamma_1 \end{bmatrix} \quad (3.5)$$

$${}^B R_D = \begin{bmatrix} c\alpha_2 c\beta_2 - s\beta_2 s\gamma_2 s\alpha_2 & -c\gamma_2 s\alpha_2 & c\alpha_2 s\beta_2 + c\beta_2 s\gamma_2 s\alpha_2 \\ c\alpha_2 s\beta_2 s\gamma_2 + c\beta_2 s\alpha_2 & c\gamma_2 c\alpha_2 & -c\beta_2 c\alpha_2 s\gamma_2 + s\beta_2 s\alpha_2 \\ -c\gamma_2 s\beta_2 & s\gamma_2 & c\beta_2 c\gamma_2 \end{bmatrix} \quad (3.6)$$

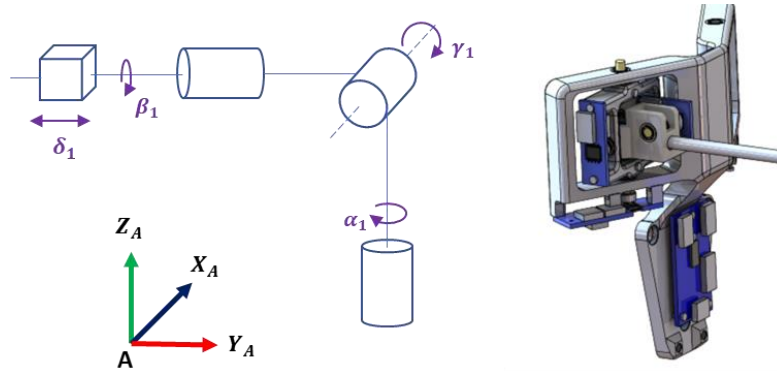


Figura 31. Sistema de giros del instrumental quirúrgico insertado en el modelo de rodilla.

Para medir los ángulos de rotación del instrumental insertado en cada puerto de entrada se utilizan cuatro sensores de efecto *hall* AS5600 en cada puerto de entrada. Dos sensores AS5600 miden el giro pitch (β) y la distancia (δ), estos sensores se colocan como se muestra en la Figura 32. El instrumento quirúrgico hace girar dos ruedas acopladas a un imán radial cada una.

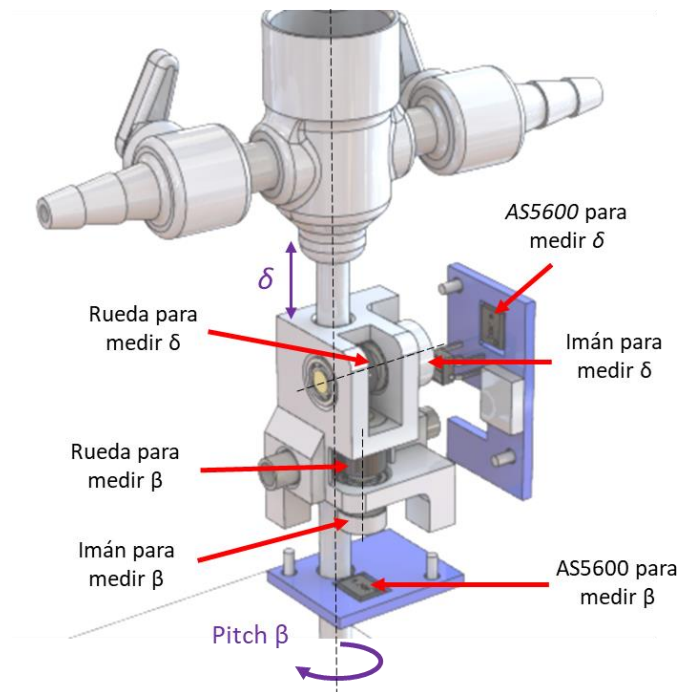


Figura 32. Sensores para medir el ángulo pitch β y la distancia δ de los instrumentos quirúrgicos.

El giro roll (γ) de los instrumentos quirúrgicos se mide con un sensor AS5600 colocado como se muestra en la Figura 33.

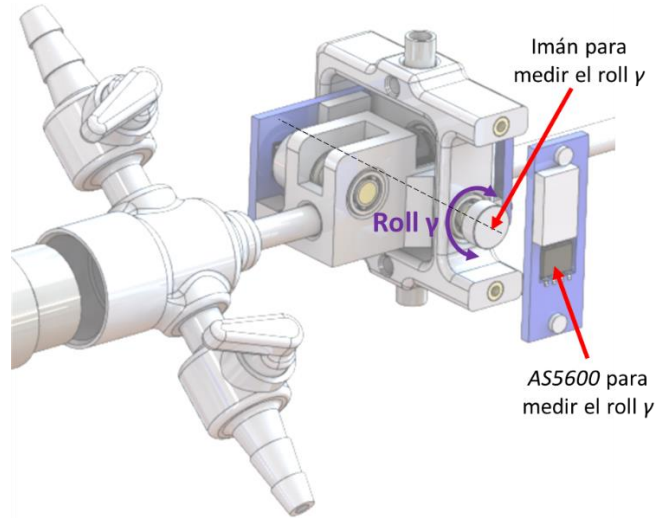


Figura 33. Sensores para medir el roll (γ) de los instrumentos quirúrgicos.

El giro yaw (α) de los instrumentos quirúrgicos se mide con un sensor AS5600 colocado como se muestra en la Figura 34.

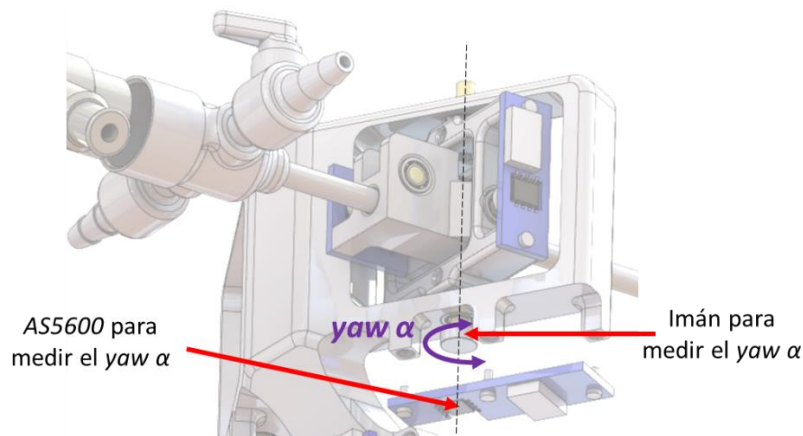


Figura 34. Sensores para medir el yaw (α) de los instrumentos quirúrgicos.

Los sensores AS5600 envían su información por medio de un bus I^2c o con el puerto ADC. Los sensores que miden los ángulos α_1 , α_2 , γ_1 , γ_2 y los desplazamientos δ_1 , δ_2 se

comunican por medio del protocolo I^2c . Los sensores que miden los ángulos β_1 y β_2 se conectan a convertidores ADC del microcontrolador.

El circuito 4 del arreglo instrumental multiplexa el bus I^2c , ya que los sensores AS5600 tienen la misma dirección $0x36$. Para ello, se utiliza el circuito integrado TCA9548A de Texas Instruments [61]. Las resistencias *pull-up* de cada bus I^2c se seleccionan a partir de la nota de aplicación [62]. Se utilizan tres conectores FPC de 10 pines y uno de 4 pines, Figura 35.

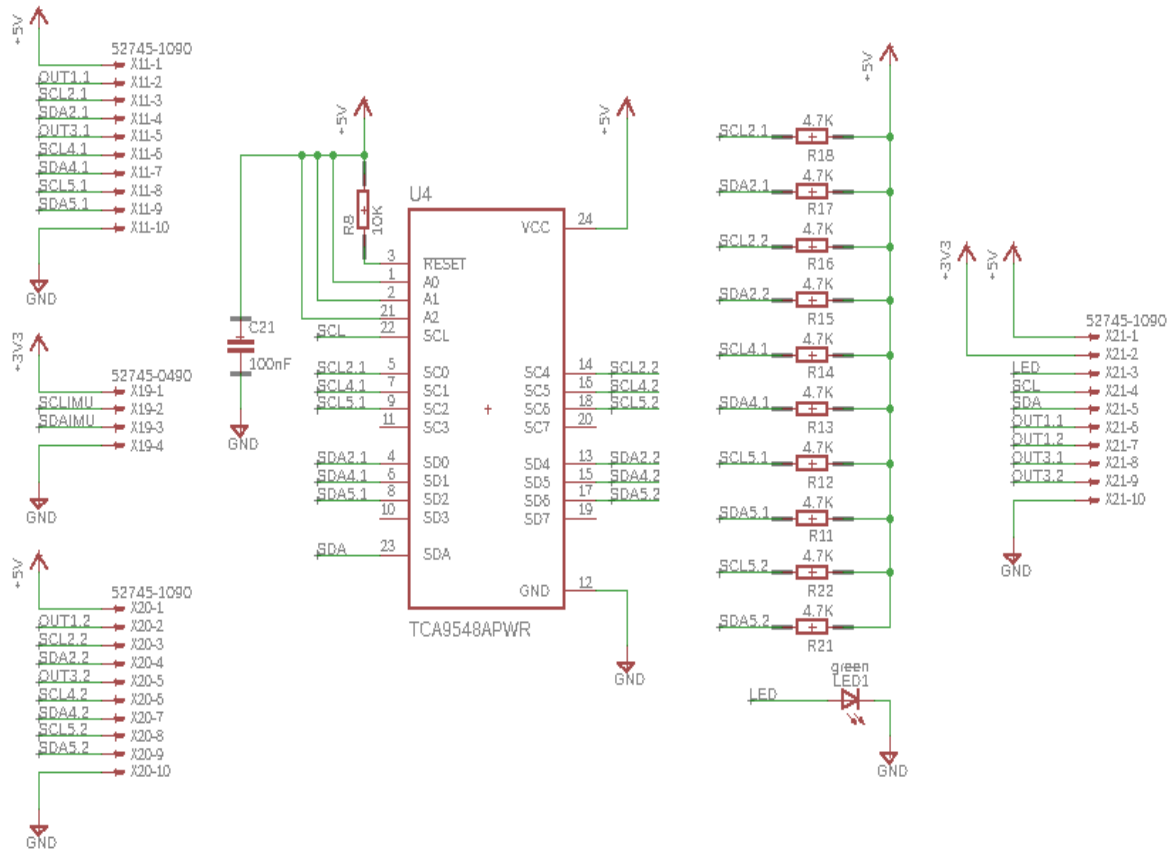


Figura 35. Circuito 4 para multiplexar las señales SDA y SCL del bus I^2c .

Los grados de libertad de los instrumentos quirúrgicos se calculan con base en el diagrama de flujo de la Figura 36.

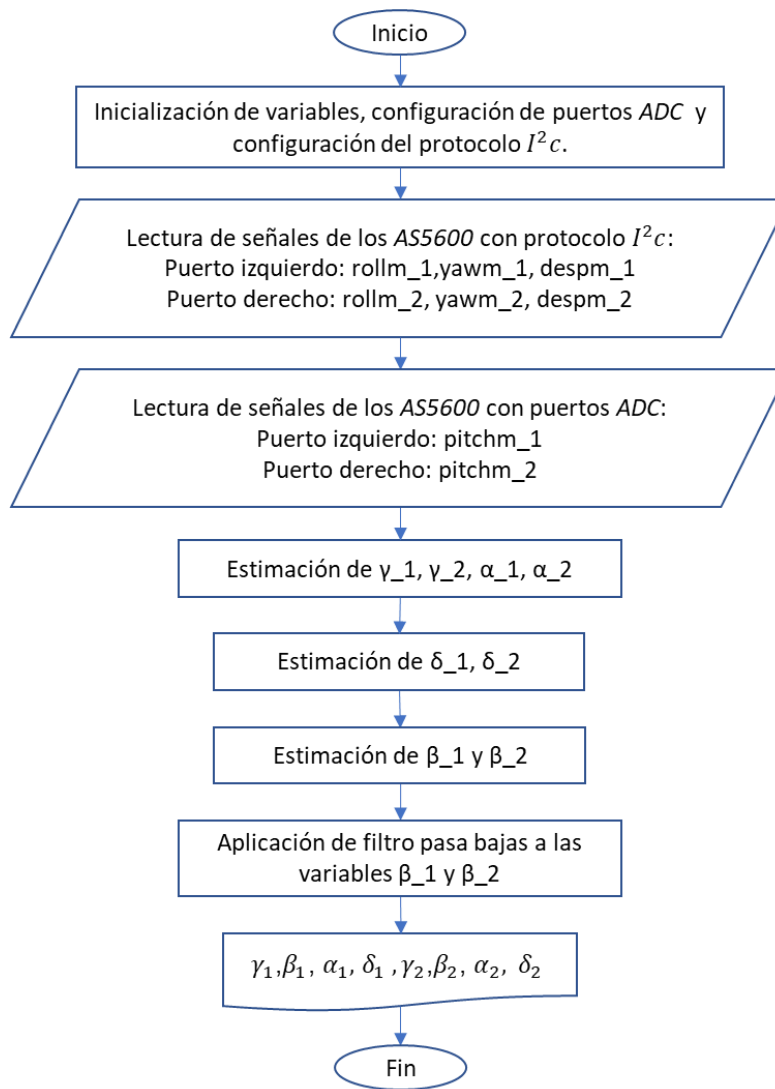


Figura 36. Diagrama de flujo para estimar los cuatro grados de libertad de cada instrumental insertado en la rodilla.

3.3.3 Mediciones para identificar el instrumental quirúrgico

El instrumental quirúrgico insertado en algún puerto de la rodilla se identifica con un sensor reflectivo QRE1113 que mide la cantidad de luz que refleja el instrumento quirúrgico, Figura 37. Para esta aplicación, cada uno de los tres instrumentos quirúrgicos simulados tienen un color diferente que refleja una cantidad de luz diferente.

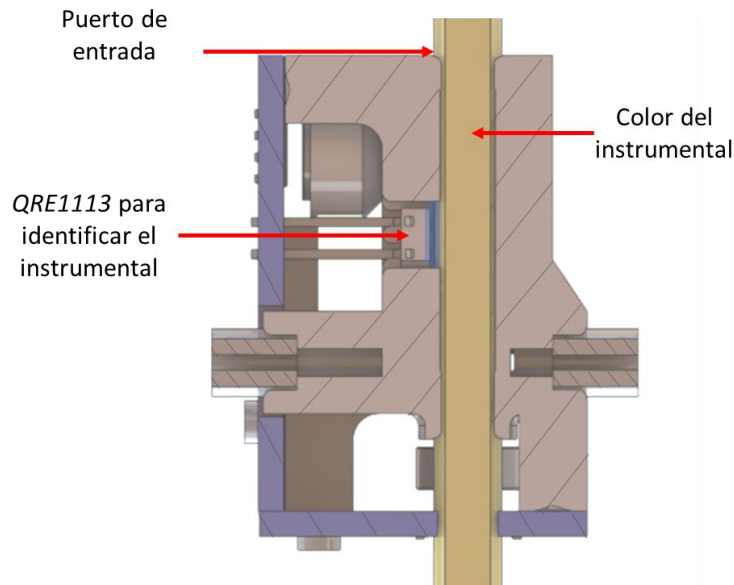


Figura 37. Sensor para identificar al instrumental quirúrgico.

El sensor *QRE1113* tiene un diodo emisor y un fototransistor. El fototransistor controla la cantidad de corriente que pasa entre su colector y emisor, que depende de la cantidad de luz que recibe. Entre el colector y emisor se mide la tensión con un puerto *ADC* del microcontrolador. En la Figura 38 se muestra el circuito del módulo 8.1 y 8.2, donde se conectan los sensores reflectivos y los sensores *AS5600* que miden el giro β_1 y β_2 .

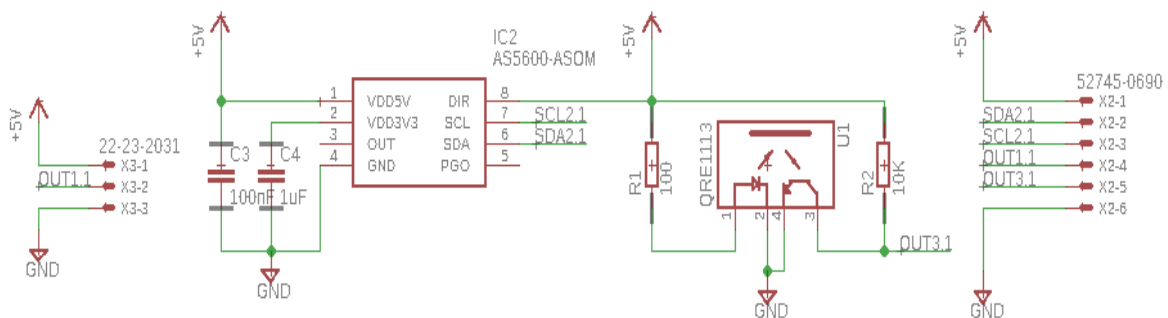


Figura 38. Circuito del módulo eléctrico 8.1 para identificar al instrumental quirúrgico.

Los puertos de entrada identifican cuatro estados diferentes: 1) cuando no se detecta un instrumental quirúrgico, 2) cuando se detecta el artroscopio, 3) cuando se detectan las pinzas de artroscopia y 4) cuando se detecta el palpador.

3.3.4 Mediciones en el artroscopio

El artroscopio tiene dos variables de giro. La primera es el giro del instrumento sobre su propio eje, que indica la posición del conector de luz por fibra óptica y del lente a 30°. La segunda es el giro del artroscopio gir_{art} que indica la orientación de la imagen, Figura 39.

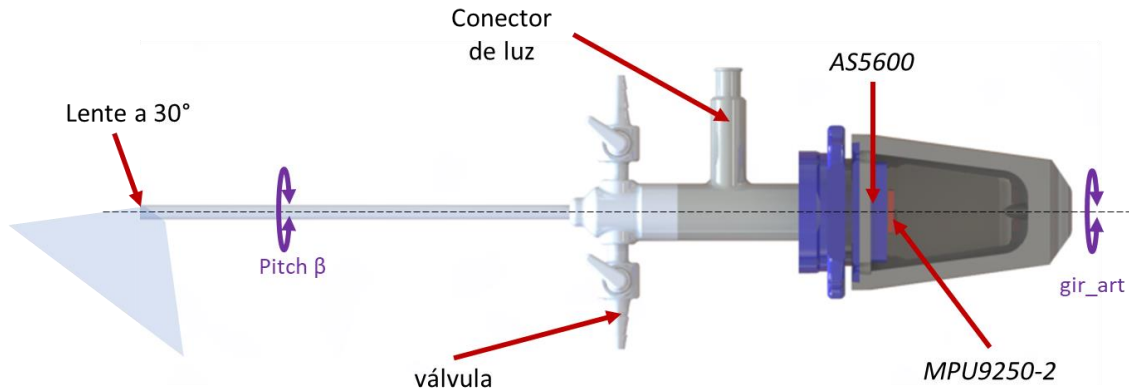


Figura 39. Sensores para medir la orientación del artroscopio y del lente.

Cuando se introduce el artroscopio en algún puerto de entrada a la rodilla se mide su orientación inicial y la del conector de luz, para ello, se combinan las mediciones del sensor MPU9250-2 con las del sensor AS5600 montado en el artroscopio.

Después de obtener las mediciones iniciales, la orientación del artroscopio se calcula con el sensor AS5600 montado en su estructura, y la orientación del conector de luz se estima con el sensor AS5600 montado en el puerto de entrada a la rodilla. Por lo que el sensor MPU9250-2 solamente se utiliza para calcular las orientaciones iniciales.

El procedimiento para estimar la orientación de la imagen del artroscopio, y la orientación del conector de luz, se presenta en el diagrama de la Figura 40.

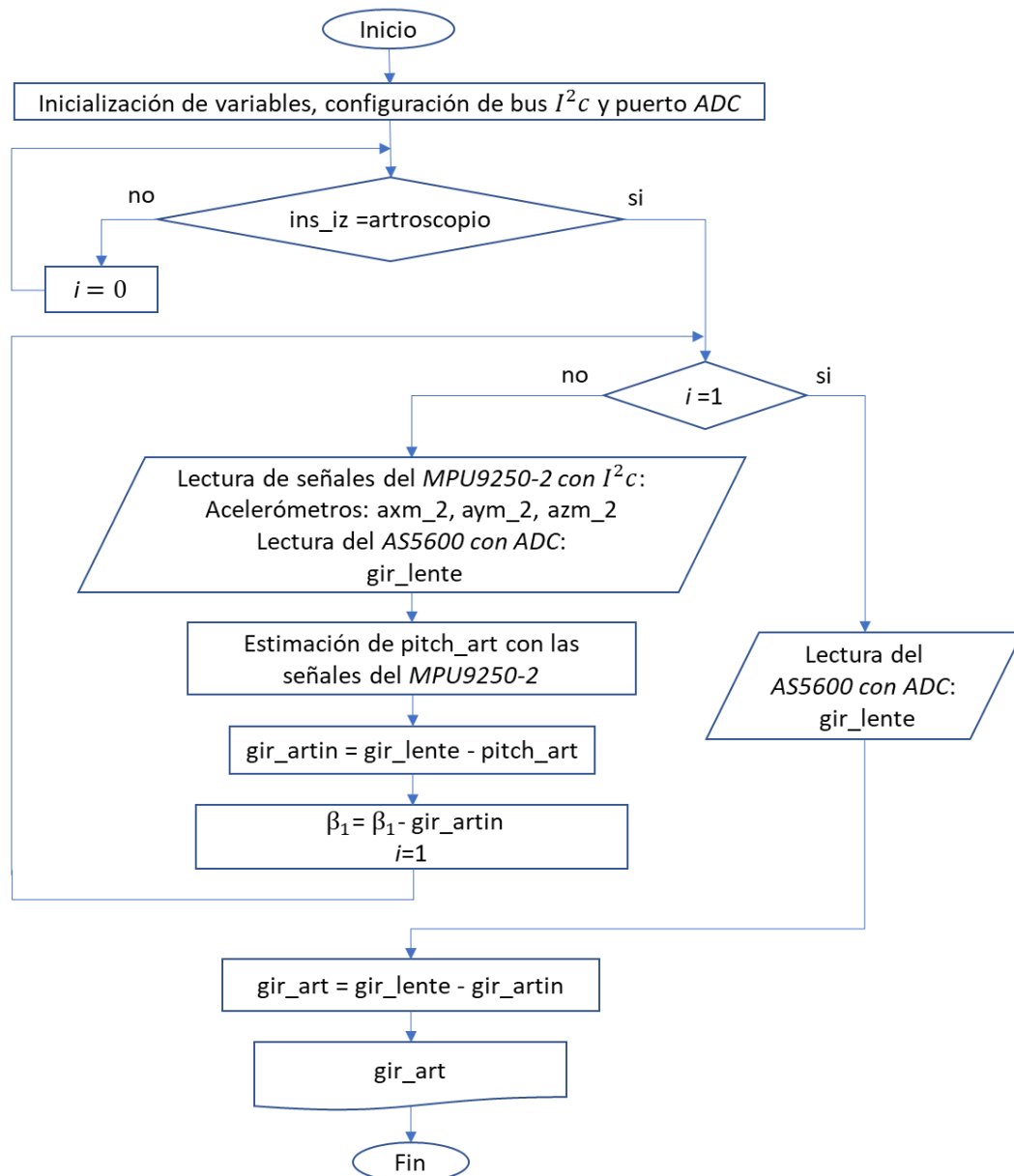


Figura 40. Diagrama de flujo para calcular la orientación del artroscopio y del conector de luz.

El circuito del artroscopio tiene un sensor *AS5600* y un sensor *MPU9250*, Figura 41, al igual que el circuito 5 de la Figura 28, se implementa un adaptador de nivel con transistores *MOSFET*.

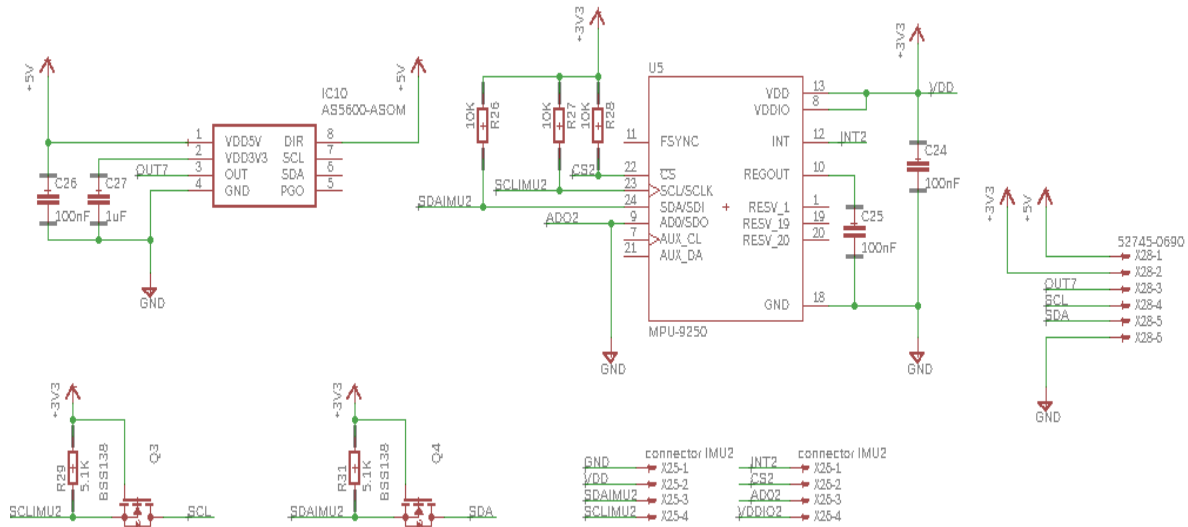


Figura 41. Circuito 2, del artroscopio.

3.3.5 Mediciones en las pinzas de artroscopía

La apertura de los mangos de las pinzas de artroscopía tiene un rango de 0° a 18° , que representa la apertura de las mordazas en un rango de 0° a 45° . Para medir el ángulo de los mangos, se utiliza un sensor AS5600, Figura 42. El circuito de las pinzas de artroscopía consiste en los elementos mínimos para el funcionamiento del puerto *PWM* del sensor AS5600, obtenidos de su hoja de datos [58], Figura 43.

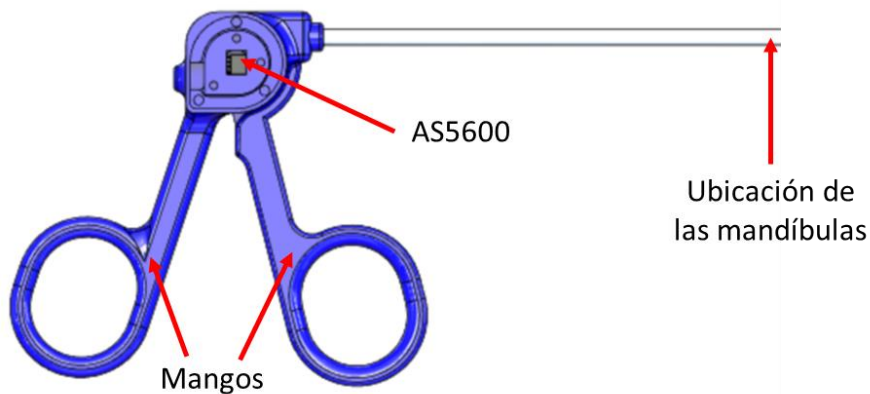


Figura 42. Sensor para medir la apertura de las pinzas.

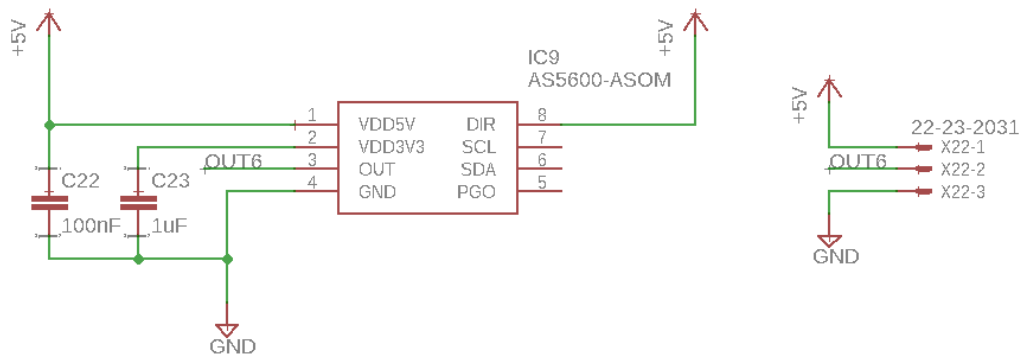


Figura 43. Circuito 3 de la apertura de las pinzas.

3.4 Tarjeta electrónica del arreglo instrumental

El microcontrolador de la tarjeta electrónica recibe las señales de todos los sensores, aplica el *filtro complementario* digital a las señales del MPU9250-1 y los filtros digitales a algunos sensores, detecta el instrumento insertado en los puertos de entrada, y al final envía toda la información a un programa para el entrenamiento quirúrgico en el motor gráfico de Unity, a través de un puerto *USB*, Figura 44.

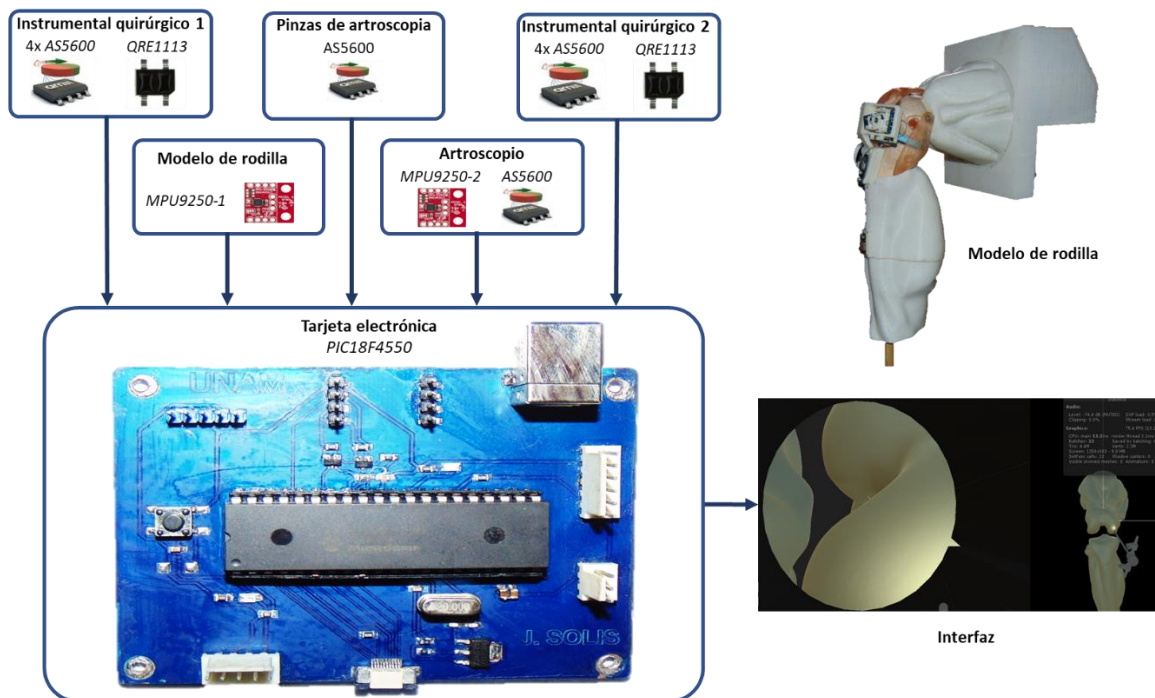


Figura 44. Diagrama de bloques del arreglo instrumental.

La tarjeta electrónica del arreglo instrumental o circuito 1, Figura 45, proporciona tensiones de 3.3 [V] y 5 [V] a los sensores, por lo que se utiliza un regulador de voltaje AMS117-3.3 [63]. La tarjeta electrónica también contiene al microcontrolador PIC18F4550 que incluye una interfaz USB.

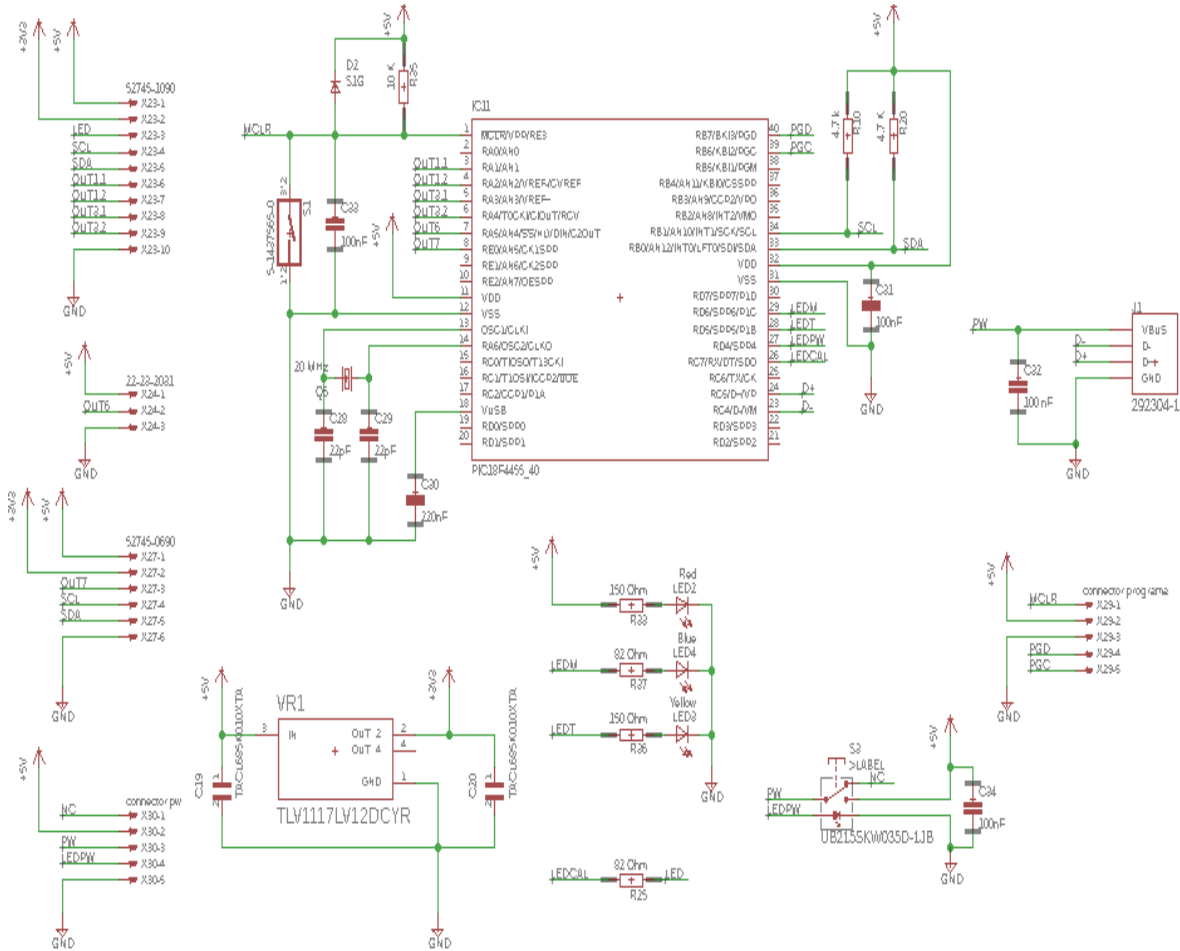


Figura 45. Circuito 1 del arreglo instrumental.

Las señales de los sensores AS5600 se leen con seis canales ADC y con el protocolo I^2c . Los canales ADC del microcontrolador PIC18F4550 son de 10 [bits], por lo que las señales de los sensores tienen un rango de entre 0 [V] y 5 [V], donde 5 [V] representa un giro de 360°. Los sensores QRE1113 también se conectan al ADC del microcontrolador, por lo tanto, su señal también tiene una tensión de entre 0 [V] a 5 [V], o de 0 a 1023.

El programa del microcontrolador está basado en el diagrama de flujo de la Figura 46. Dicho programa se encuentra en el Anexo B.

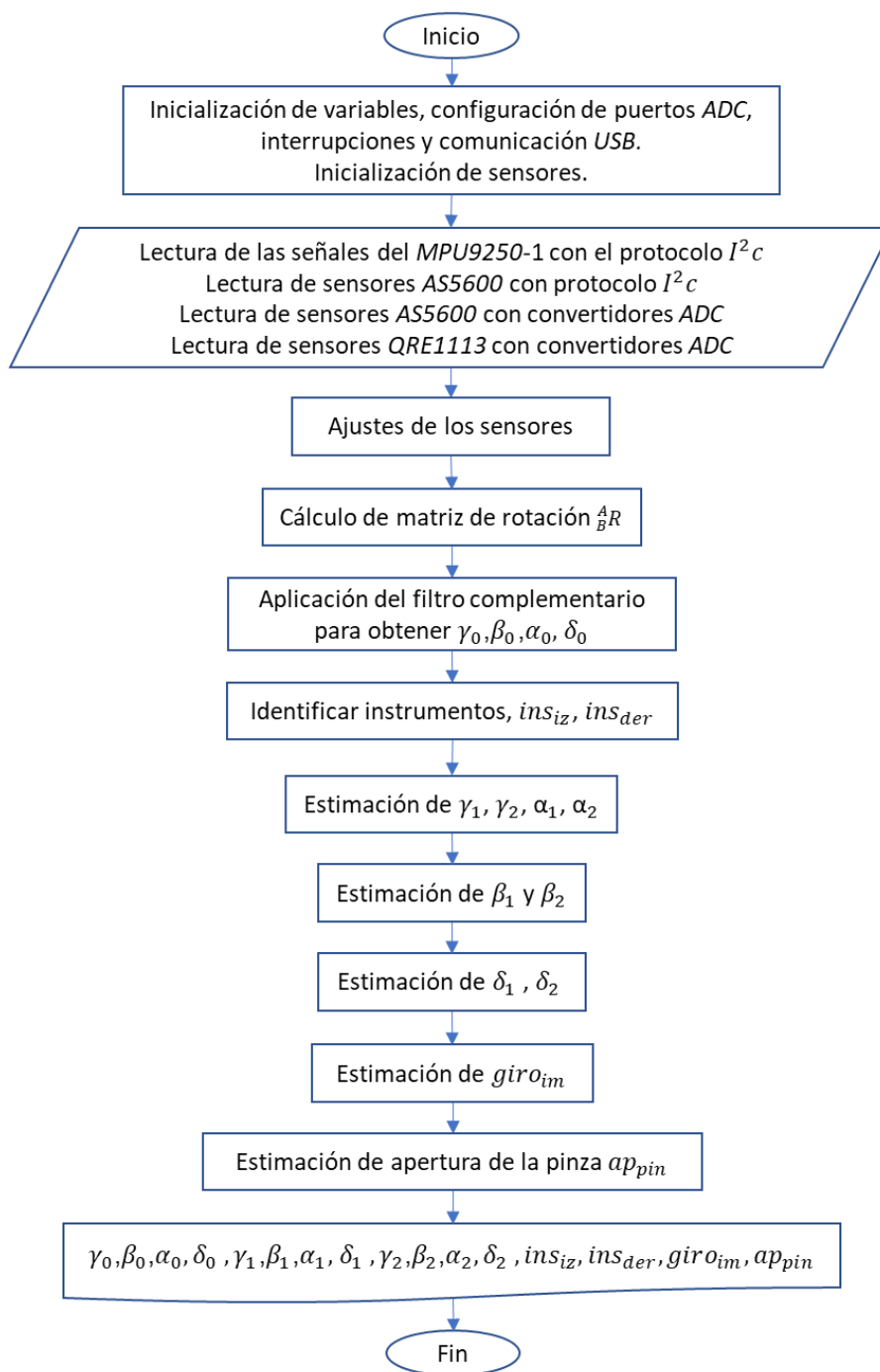


Figura 46. Diagrama de flujo del programa del microcontrolador.

3.5 Construcción del dispositivo instrumentado

Las piezas que integran al arreglo instrumental se manufacturan con una impresora 3D utilizando filamento PLA. El proceso de manufactura permite imprimir piezas obtenidas de un escáner 3D modelo Sense2 de la marca 3D Systems [64]. En la Figura 47 se muestra el simulador manufacturado y en la Figura 48 se muestra el instrumental quirúrgico.

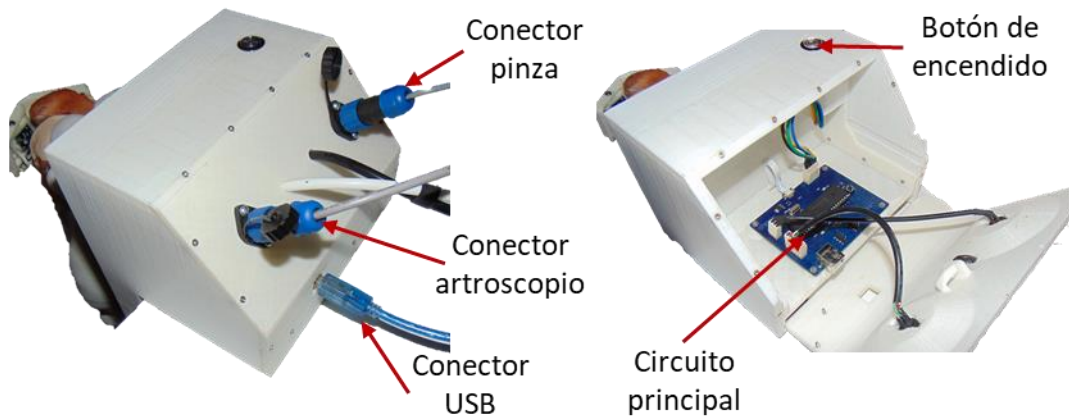
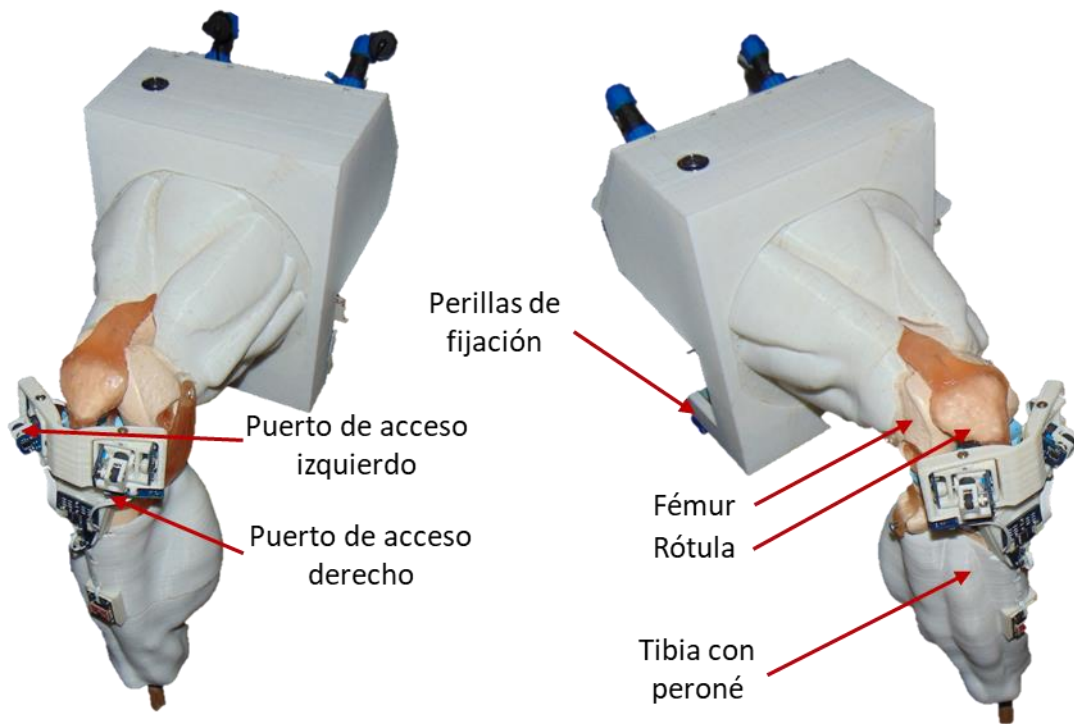


Figura 47. Simulador de artroscopía en rodilla, manufacturado con PLA en una impresora 3D.

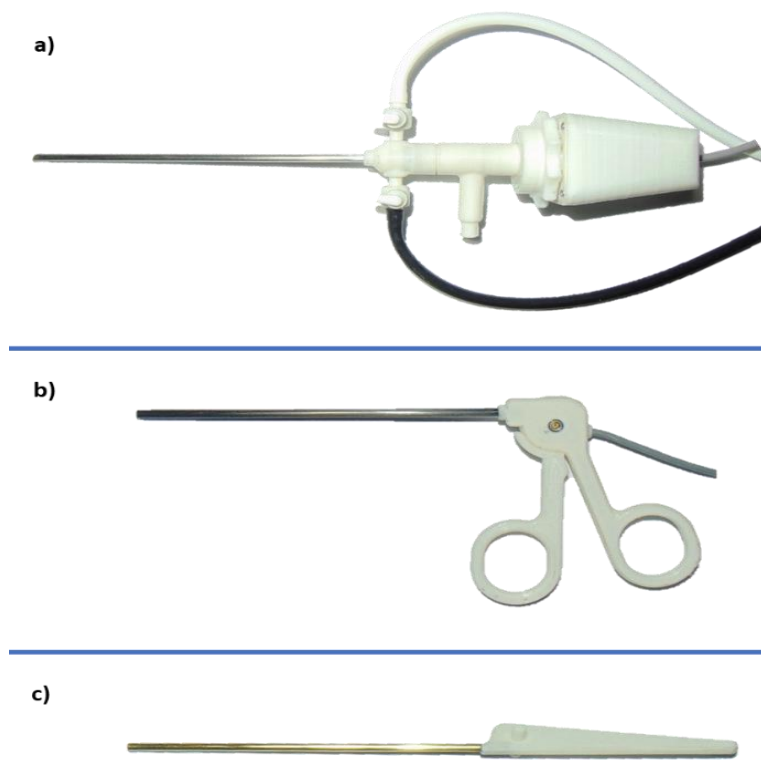


Figura 48. Instrumental quirúrgico impreso en 3D con filamento PLA, a) artroscopio, b) pinzas de artroscopía, c) palpador.

El proceso para la construcción de las piezas mecánicas del arreglo instrumental se muestra en la Figura 49. Se inicia con el modelo virtual, se genera el archivo de impresión 3D, se imprime la pieza con el material PLA y por último se limpia la pieza para agregar los elementos mecánicos como los rodamientos.

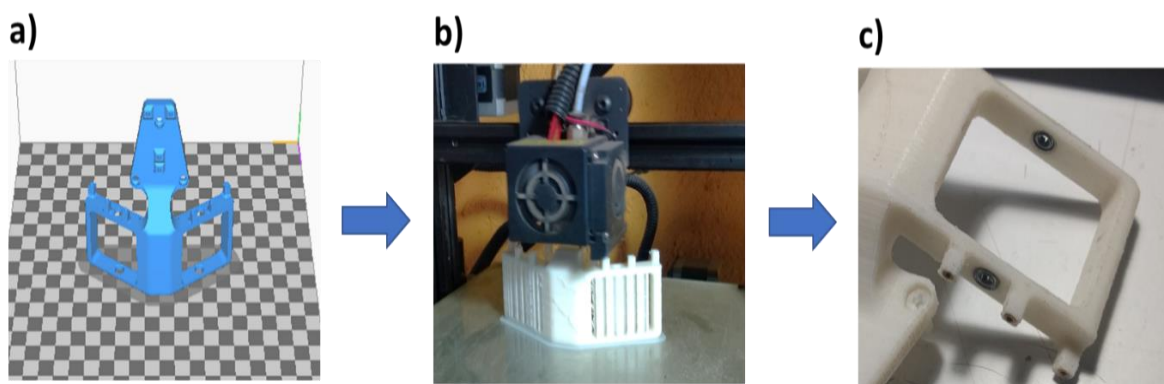


Figura 49. Proceso de manufactura de los elementos mecánicos del arreglo instrumental, a) diseño de la pieza, b) manufactura de la pieza, c) acabado final e incorporación de elementos mecánicos.

El proceso de manufactura de los circuitos inicia con el diseño del circuito y de los archivos para su manufactura, después, en la máquina *CNC* se trazan las pistas, barrenos y el corte del *PCB* en placas fenólicas de doble cara, posteriormente se agrega una capa protectora curable en luz *UV* y por último se sueldan los componentes electrónicos, Figura 50.

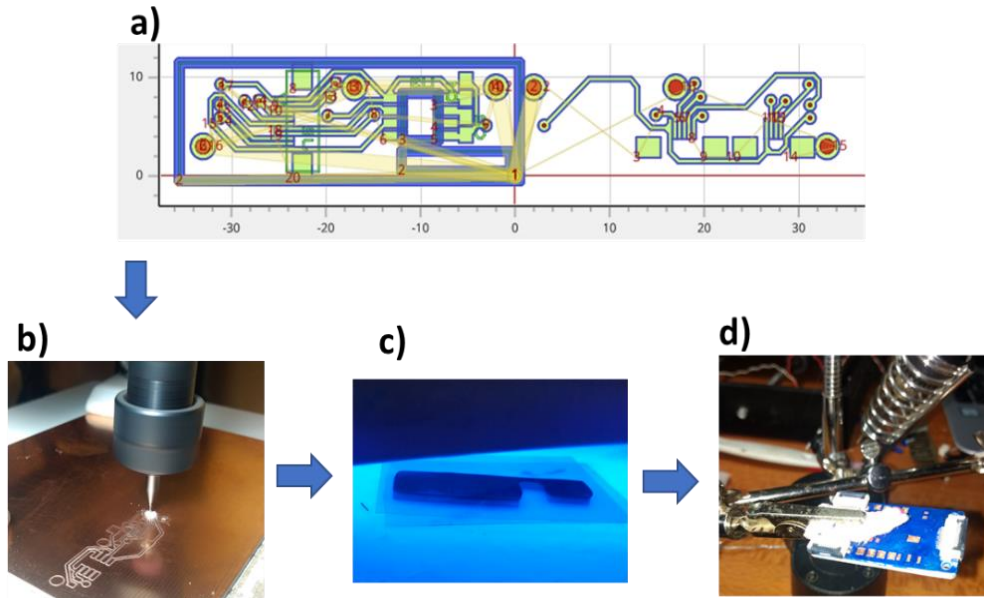


Figura 50. Proceso de manufactura de los circuitos del arreglo instrumental, a) diseño del circuito, b) Manufactura en la *CNC*, c) aplicación de protector curable *UV*, d) soldadura de componentes.

3.6 Simulación de los movimientos medidos por el arreglo instrumental

3.6.1 Comunicación con la interfaz gráfica

La información de los sensores es procesada en el microcontrolador y enviada a través del puerto *USB* a una interfaz gráfica en un motor de gráficos de realidad virtual en *Unity*. La información se envía en una matriz, Figura 51.

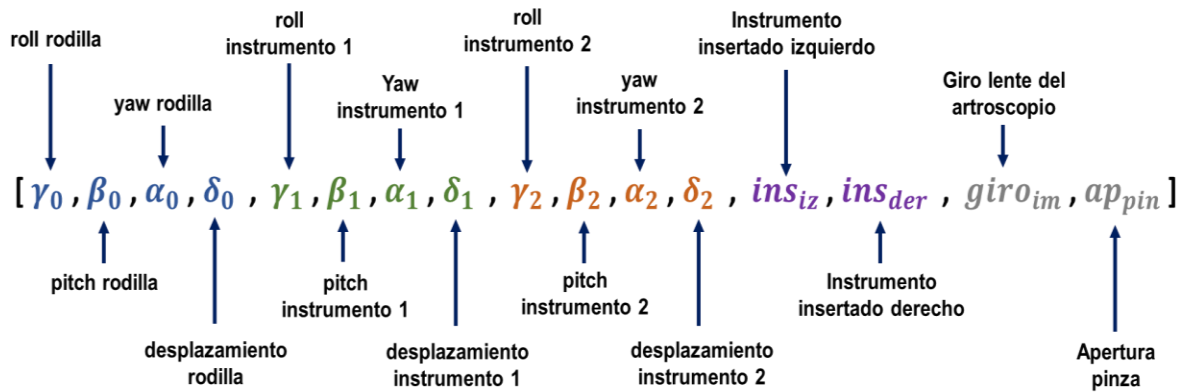


Figura 51. Información que envía el microcontrolador al equipo de cómputo, por un puerto USB.

Se envían alrededor de 25 matrices de datos por segundo. Se utiliza la interfaz *CDC USB* del *PIC18F4550*, que simula un puerto *COM* en el equipo de cómputo, de esta forma, el programa *Unity* recibe la información del arreglo instrumental.

3.6.2 Simulación del movimiento en Unity

La interfaz del simulador se desarrolla en el programa gráfico *Unity*, donde se muestra la imagen del artroscopio, con una marca que indica la orientación del conector de luz, y del lente a 30° . También se incluye la simulación de la posición actual de la rodilla, Figura 52.

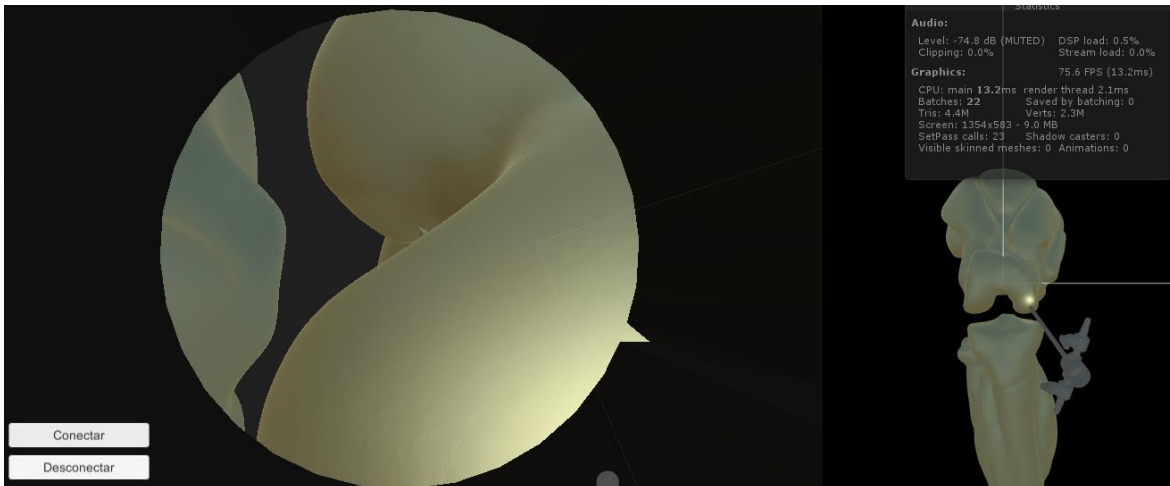


Figura 52. Interfaz gráfica en el programa gráfico Unity.

Se programan *scripts* para procesar la información recibida por el arreglo instrumental, y para mover los modelos virtuales en el ambiente virtual. Un *script* recibe la información del arreglo instrumental, hace el cálculo de las posiciones de las puntas de cada instrumento quirúrgico insertado en la rodilla, almacena la información en archivos de texto y la comparte con otros *scripts* que controlan el movimiento de los modelos virtuales, Figura 53. Los códigos de los *scripts* se encuentran en el Anexo C.

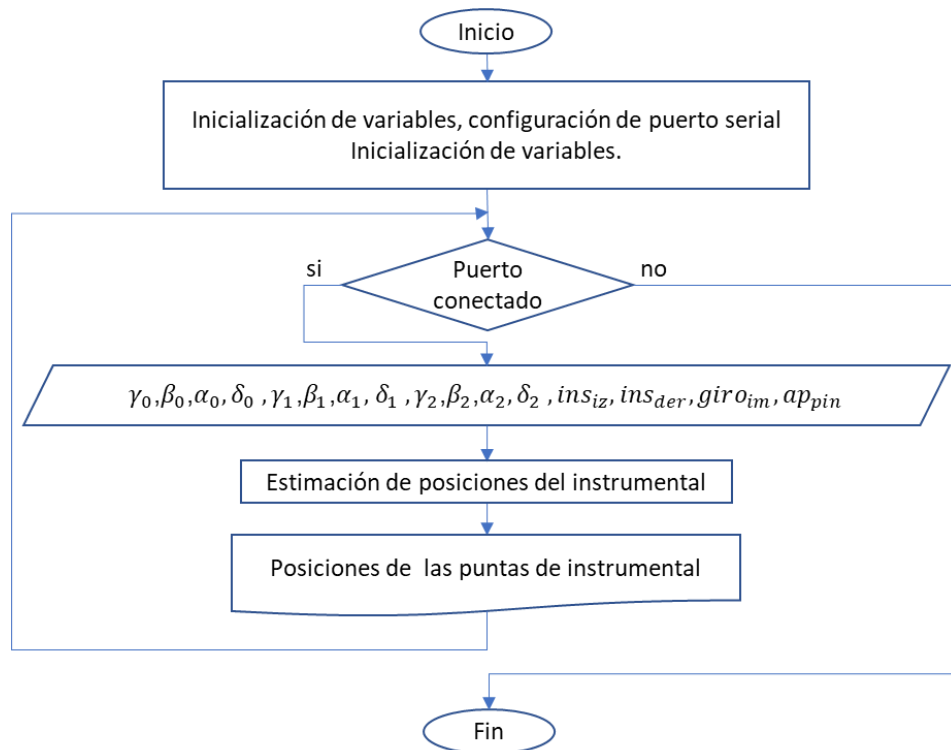


Figura 53. Diagrama de flujo del script principal en Unity.

Los *scripts* que controlan el movimiento de los modelos virtuales determinan si el instrumental quirúrgico está insertado en alguno de los puertos de entrada, Figura 54.

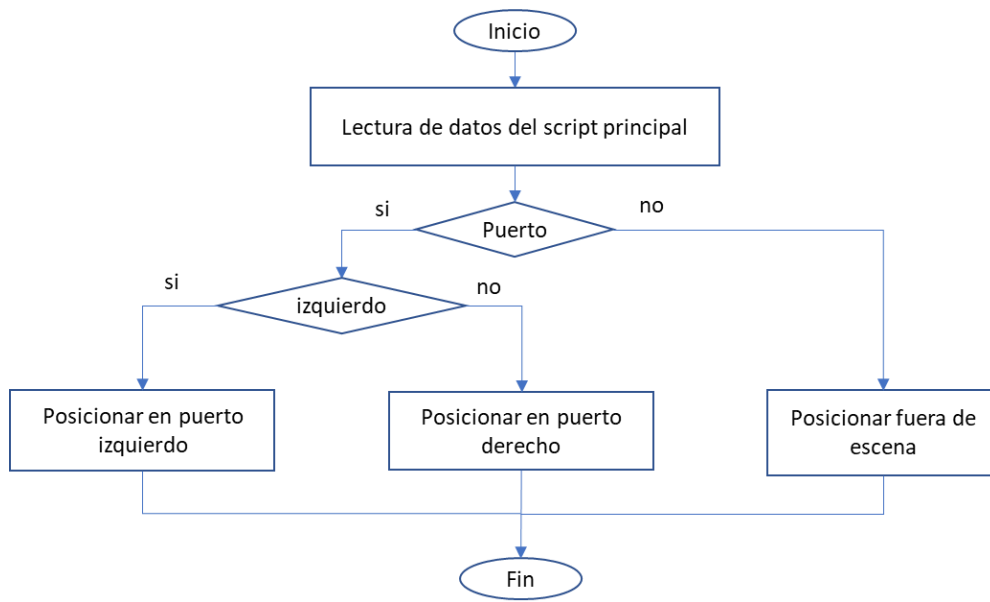


Figura 54. Diagrama de flujo para posicionar al instrumental quirúrgico.

Algunas actividades que se pueden realizar en el ambiente virtual son:

1. Triangulación (ubicar el instrumento insertado en el centro de la imagen del artroscopio).
2. Ubicación espacial, que consta de ubicar los instrumentos en las zonas indicadas.
3. Extracción de objetos de la articulación con las pinzas de artroscopia.

CAPÍTULO 4. CALIBRACIÓN DE LOS SENSORES DEL ARREGLO INSTRUMENTAL

4.1 Calibración de los sensores MPU9250

La calibración de los sensores se realiza con la información de las hojas de datos de sus fabricantes. En el caso de los acelerómetros y magnetómetros del MPU9250 se utiliza la siguiente ecuación:

$$a_i = \left(m_i - \frac{max_i - min_i}{2} - min_i \right) \frac{1}{rango} \quad (4.1)$$

donde:

i es el eje X, Y o Z.

a_i es la señal calibrada y normalizada, para el eje i .

m_i es la señal medida en el eje i .

max_i es el valor máximo de la señal en el eje i .

min_i es el valor mínimo de la señal en el eje i .

$rango$ es igual a la diferencia entre max_i y min_i , dividido entre dos.

Los valores mínimos y máximos de los acelerómetros y magnetómetros se determinan con el mecanismo de la Figura 55, que permite girar libremente al sensor MPU9250. Para la calibración de los giróscopos se utiliza la siguiente ecuación:

$$g_i = (m_i - offset_i) \frac{resolución/2}{rango} \quad (4.2)$$

donde:

i es el eje X, Y o Z.

g_i es la señal calibrada y normalizada para el eje i .

$offset_i$ es el offset determinado experimentalmente con el MPU9250 inmóvil, en el eje i .

$resolución$ es igual a 65536 (16 [bits]).

$rango$ es igual a 2000 [DPS].

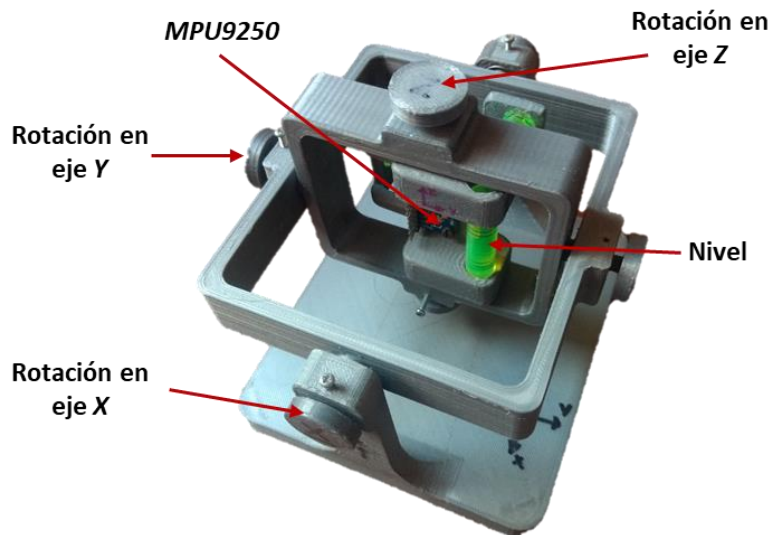


Figura 55. Mecanismo para la calibración del MPU9250.

4.1.2 Estimación del rango y precisión del movimiento de la rodilla

Como se mencionó anteriormente, el movimiento de la articulación de rodilla tiene cuatro grados de libertad: γ_0 , β_0 , α_0 y δ_0 , que anatómicamente están limitados por los ligamentos. El rango de las variables anteriores se determina experimentalmente, Figura 56.

Para determinar el rango de la rodilla se miden los límites de su movimiento. La variable γ_0 es el ángulo *roll* de la articulación y tiene un rango de -40° a 60° . La variable β_0 es el *pitch* de la articulación y tiene un rango de -10° a 13° . La variable α_0 es el *yaw* de la articulación y tiene un rango de -15° a 15° . La variable δ_0 , es el movimiento sobre el eje Z del sistema de referencia A y tiene un rango de 0 a 9 [mm].

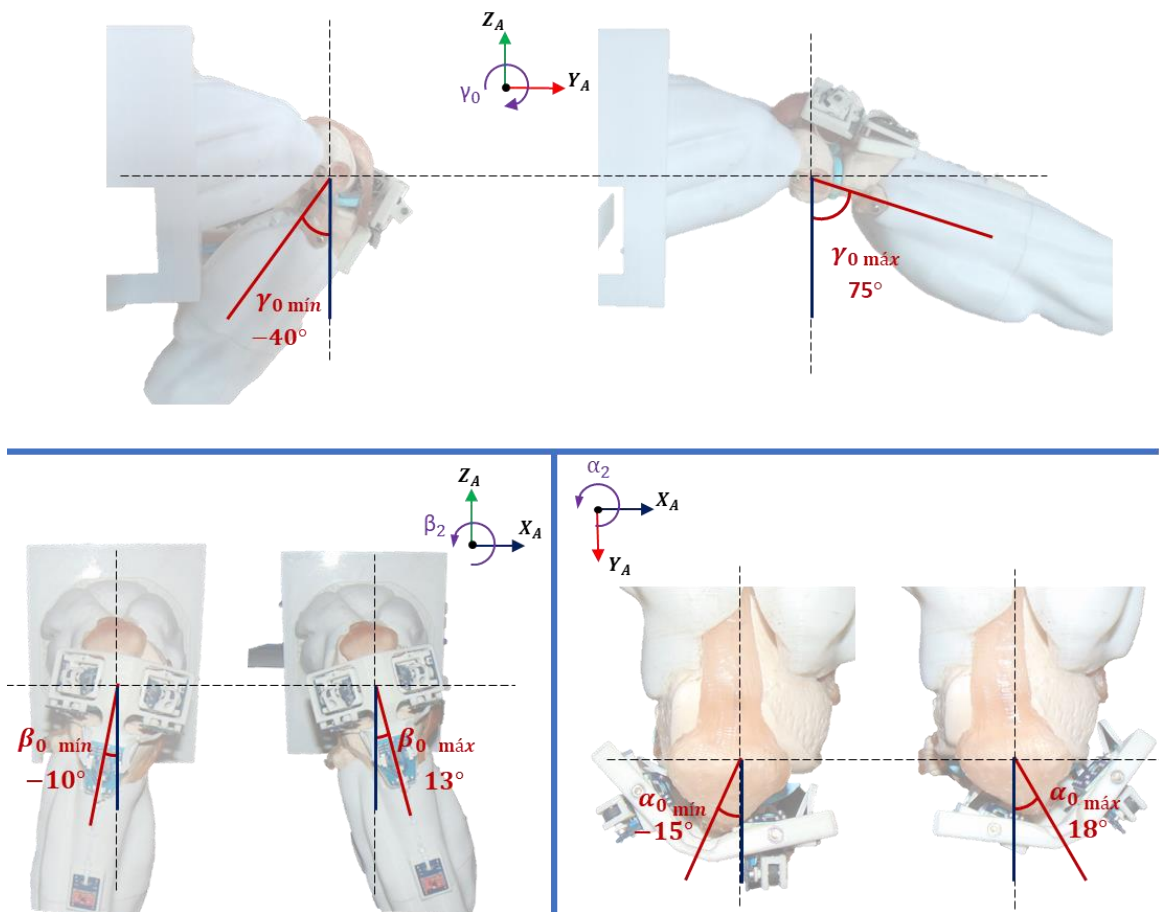


Figura 56. Rango de los movimientos del modelo de rodilla.

La precisión de un sensor es la dispersión que tiene un conjunto de mediciones obtenidas a partir de una misma magnitud, por lo que, se puede describir con medidas de dispersión, por ejemplo, la desviación estándar, varianza o coeficiente de variación [65]. Para obtener la precisión de los sensores primero se toman las medidas con el modelo de rodilla o los instrumentos quirúrgicos inmóviles, después se calcula la desviación estándar de los datos obtenidos.

La precisión del movimiento de la rodilla se determina a partir de las señales γ_0 , β_0 y α_0 , con el modelo de la rodilla inmóvil, Figura 57. La precisión de las variables, que se calculan con la desviación estándar son las siguientes: en el *roll* γ_0 es de $\pm 0.047^\circ$, en el *pitch* β_0 es de $\pm 0.036^\circ$, en el *yaw* α_0 es de $\pm 0.161^\circ$ y en la distancia δ_0 es de ± 0.021 [mm]. El error máximo observado en γ_0 es de $\pm 0.11^\circ$, en β_0 es de $\pm 0.1^\circ$ y en α_0 es de 0.35° y en δ_0 es de ± 0.06 [mm].

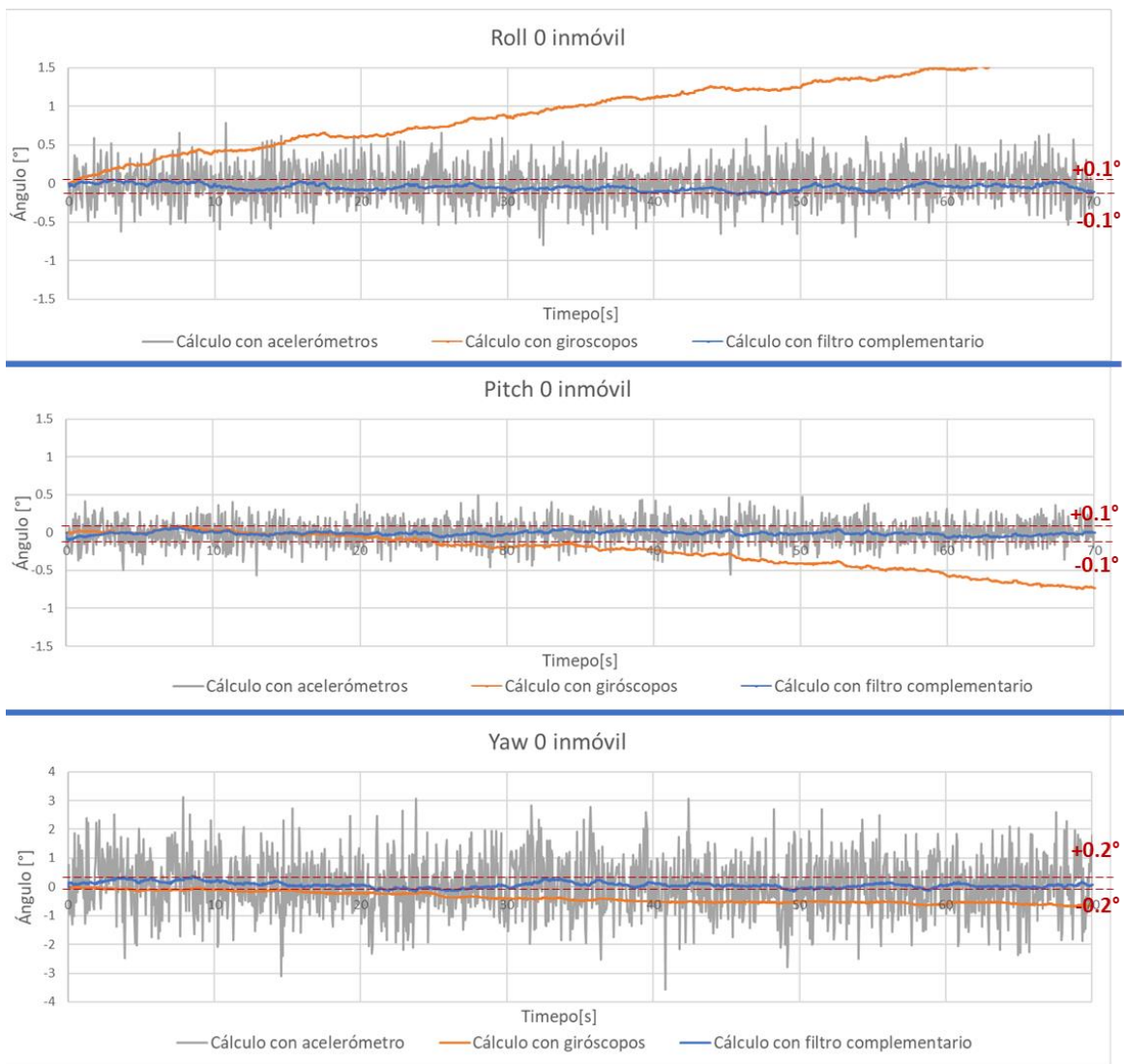


Figura 57. Ruido en las señales de los ángulos de giro del modelo de rodilla, cuando este se mantiene inmóvil.

4.2 Calibración de los sensores AS5600

La calibración de los sensores AS5600 se realiza con dos procedimientos. El primero se aplica cuando los sensores se comunican con el protocolo I^2c . El segundo se utiliza cuando los sensores se comunican con el puerto PWM que se conecta a un convertidor analógico digital ADC de 10 [bits].

Los sensores AS5600 tienen una resolución de 12 [bits], por lo que el valor de su señal tiene un rango de 0 a 4095, donde 4095 representa un ángulo de 360° . Cuando los sensores AS5600 son conectados a un ADC del microcontrolador, su resolución se limita a 10 [bits], es decir que la señal tiene un valor de 0 a 1023, donde 1023 representa un ángulo de 360° .

La señal del sensor AS5600 se calibra con la siguiente ecuación:

$$s_i = (v_i - offset_i) \frac{360^\circ}{rango} \quad (4.3)$$

donde:

i es el eje X, Y o Z.

s_i es la señal calibrada que representa el ángulo de giro de 0° a 360° , en el eje i .

$offset_i$ es el $offset$ medido experimentalmente para cada sensor, en el eje i .

$rango$ es la diferencia entre el valor medido máximo y mínimo. Con la comunicación I^2c el rango es aproximadamente de 4095 y con las señales analógicas el rango está alrededor de 1023.

4.2.1 Resolución, rango y precisión de los ángulos γ_1 , γ_2 , α_1 y α_2 .

Como se mencionó anteriormente, los ángulos γ_1 y α_1 son los giros del instrumental quirúrgico en el puerto de entrada izquierdo y los ángulos γ_2 y α_2 son los giros en el puerto de entrada derecho. Cada una de las variables anteriores se mide con un sensor AS5600 a través el protocolo I^2c . Por lo que la resolución de γ_1 , α_1 , γ_2 y α_2 es de 0.0879° .

El valor cero de las variables γ_1 , α_1 , γ_2 y α_2 se ajusta con un soporte que mantiene inmóvil al instrumental quirúrgico, Figura 58. El instrumental insertado en el puerto izquierdo permanece paralelo al eje Y_c del sistema de referencia C. El instrumental insertado en el puerto derecho permanece paralelo al eje Y_D del sistema de referencia D.

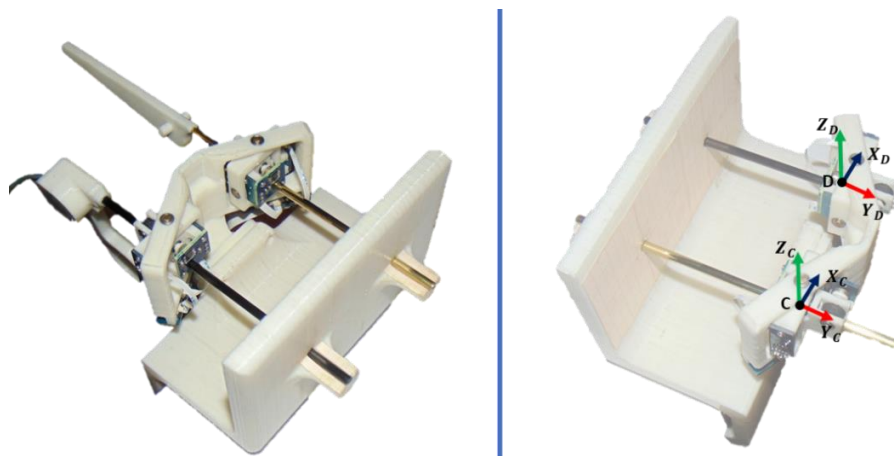


Figura 58. Soporte para determinar los ángulos cero de las variables γ_1 , γ_2 , α_1 y α_2 .

El rango de los ángulos γ_1 , γ_2 , α_1 y α_2 se determina con los valores máximos que alcanza el instrumental quirúrgico, Figura 59. Las variables γ_1 y γ_2 tienen un rango de -21° a 22° . La variable α_1 tiene un rango de -7° a 100° y la variable α_2 tiene un rango de -100° a 7° .

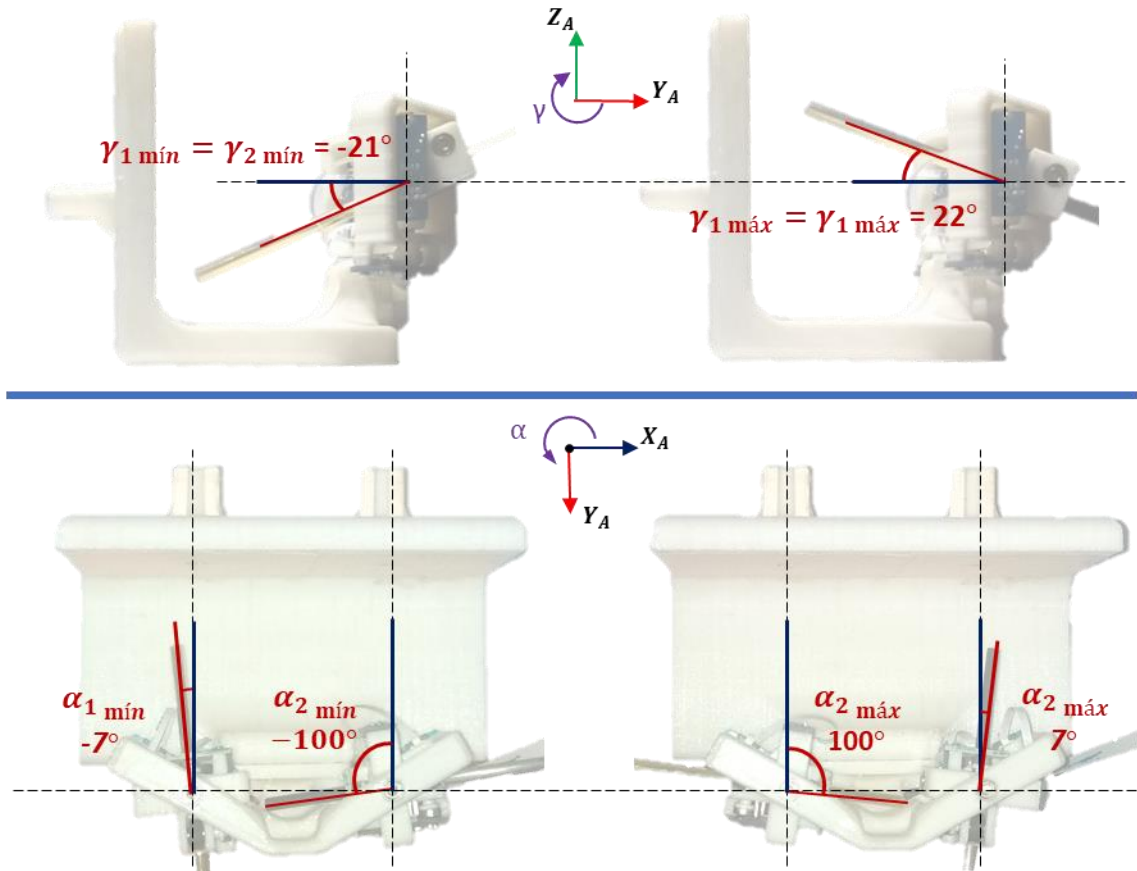


Figura 59. Rango de los movimientos del puerto de entrada izquierdo y derecho.

Las precisiones de las variables γ_1 , γ_2 , α_1 y α_2 se determinan con sus señales cuando el instrumental quirúrgico se mantiene inmóvil y posicionado en sus valores cero, Figura 60 y 61. La precisión de las variables, que se calculan con la desviación estándar, es de $\pm 0.042^\circ$ en γ_1 , $\pm 0.017^\circ$ en γ_2 , $\pm 0.043^\circ$ en α_1 y $\pm 0.039^\circ$ en α_2 . Mientras que el error máximo observado en las señales de $\pm 0.088^\circ$.

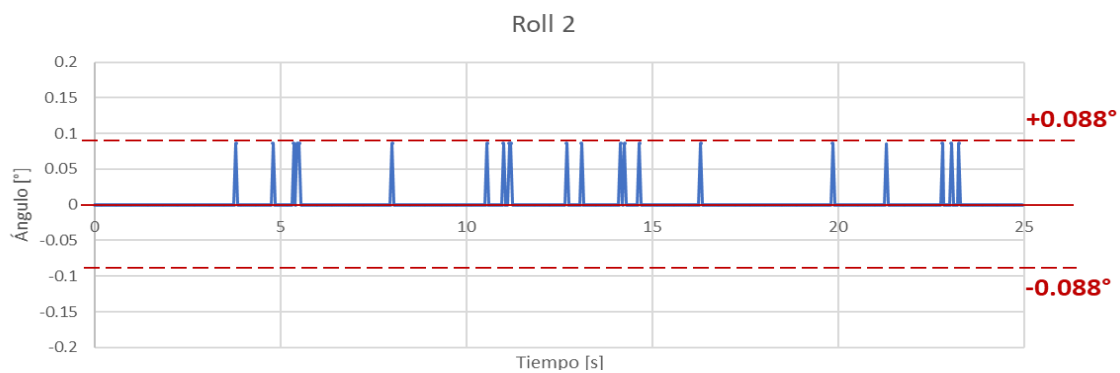
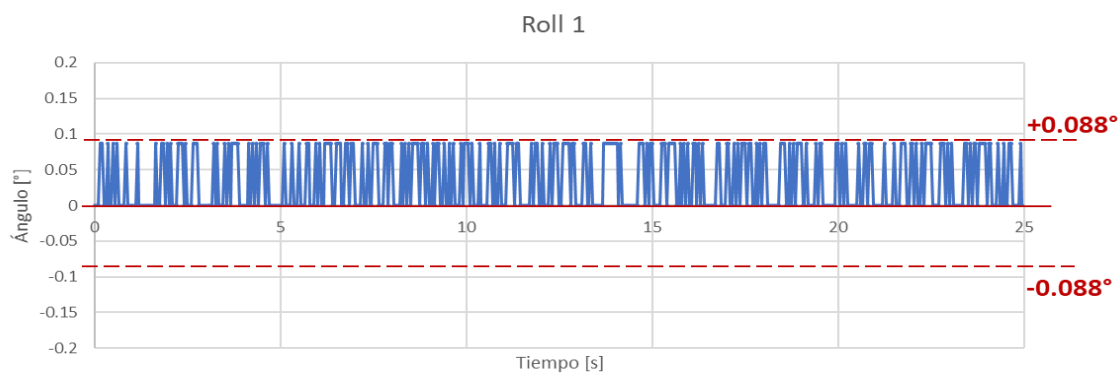


Figura 60. Señales de las variables γ_1 y γ_2 con el instrumental quirúrgico inmóvil.

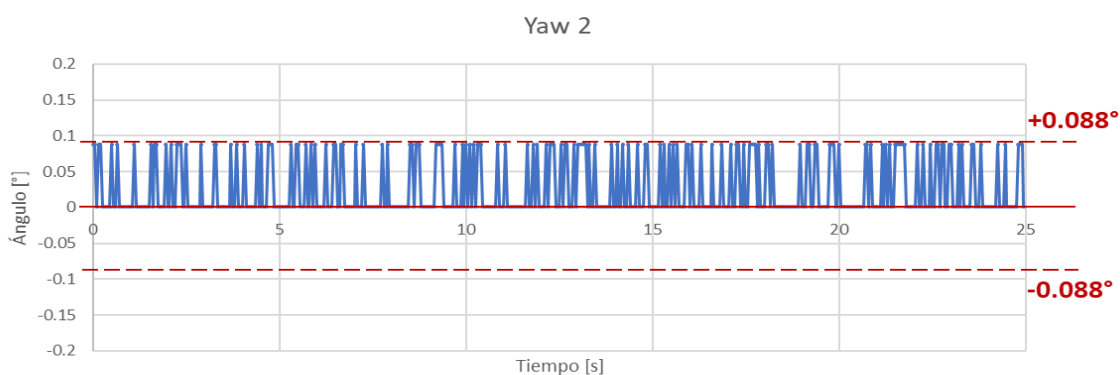
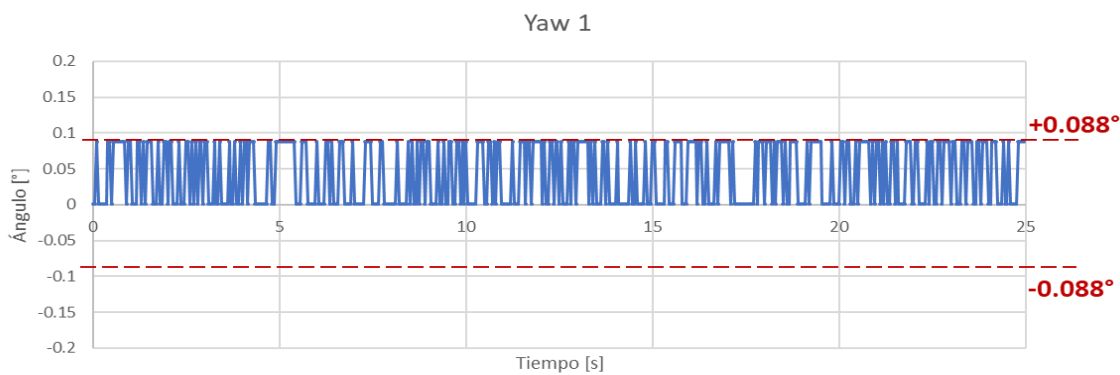


Figura 61. Señales de las variables α_1 y α_2 con el instrumental inmóvil.

4.2.2 Resolución, rango y precisión de los ángulos β_1 y β_2 .

Los ángulos β_1 y β_2 son los giros del instrumental quirúrgico sobre su propio eje, estos ángulos se calculan con los sensores AS5600 conectados a los puertos ADC del microcontrolador. Por lo tanto, las señales tienen un valor de 0 [V] a 5 [V], o de 0 a 1023.

Un sensor AS5600 mide el giro de un imán radial fijo en la rueda que mueve el instrumental quirúrgico, Figura 32. Una vuelta completa de la rueda representa 2.15 vueltas del instrumental quirúrgico, por lo que la resolución de las variables β_1 y β_2 es de 0.7565° .

Cada vez que se inserta un instrumento por alguno de los puertos de entrada, el ángulo β_1 y β_2 se ajusta a cero, con excepción del artroscopio, donde se indica la posición inicial del lente. En la Figura 62 se muestra el diagrama de flujo para estimar las variables β_1 y β_2 .

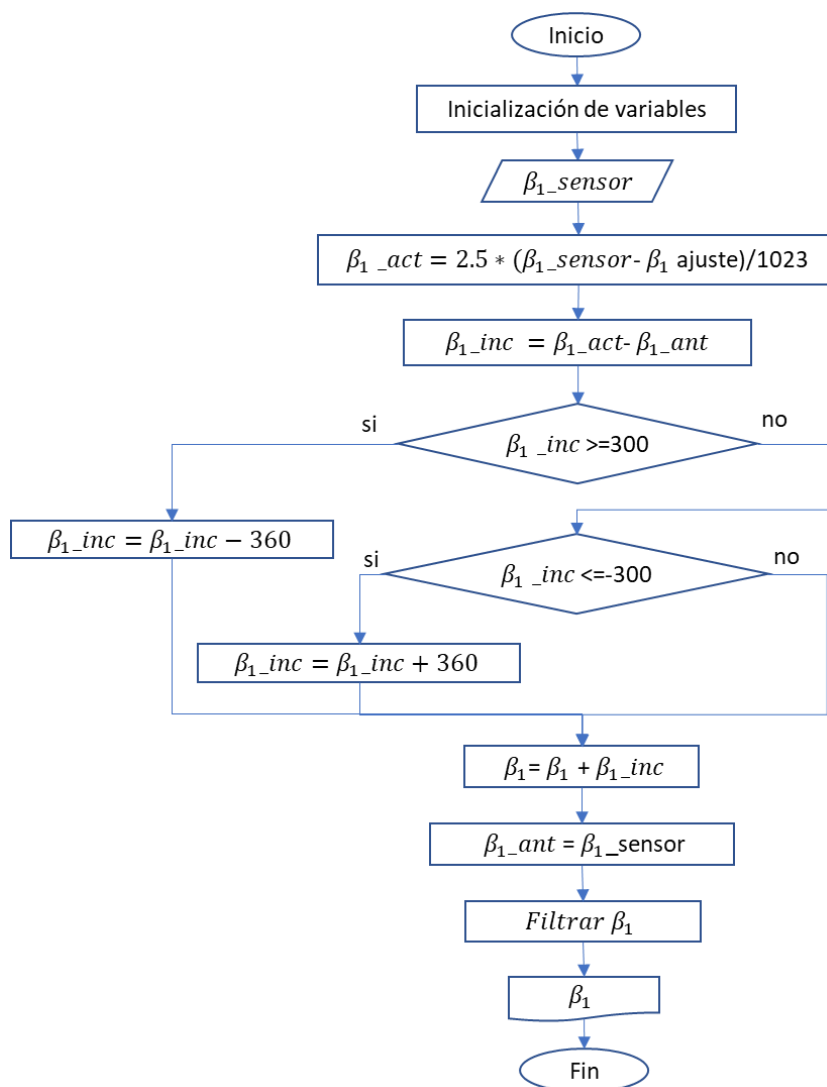


Figura 62. Diagrama de flujo para estimar el valor de las variables β_1 y β_2 .

El rango de las variables β_1 y β_2 no está limitado, ya que se pueden girar libremente los instrumentos insertados, de 0° a 360° .

La precisión de las variables β_1 y β_2 se determina a partir de sus señales, cuando el instrumental quirúrgico se mantiene inmóvil, Figura 63. La precisión de las variables, que se calculan con la desviación estándar, es de $\pm 0.093^\circ$ en β_1 y $\pm 0.07^\circ$ en β_2 . Mientras que el error máximo observado en las señales de $\pm 0.4^\circ$.

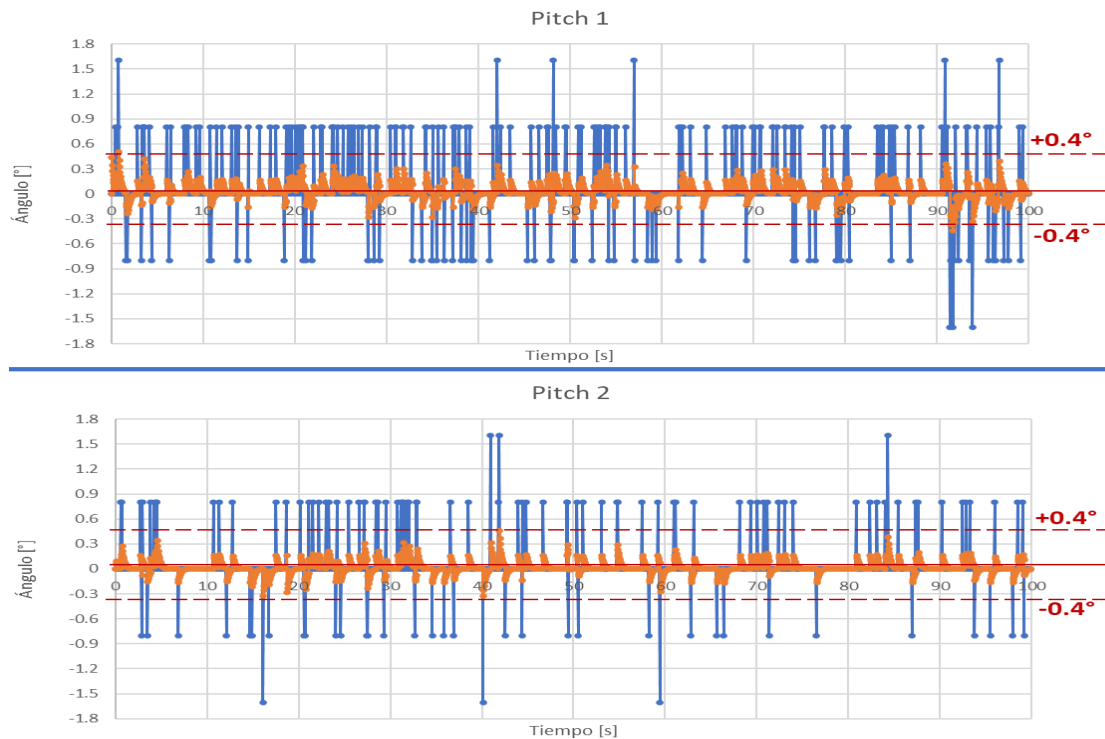


Figura 63. Señales de las variables β_1 y β_2 , con el instrumental quirúrgico inmóvil.

4.2.3 Resolución, rango y precisión de los desplazamientos δ_1 y δ_2 .

El desplazamiento δ_1 y δ_2 es el movimiento de los instrumentos al interior de la rodilla, se mide con sensores AS5600 comunicados con el protocolo I^2C , al hacer girar las ruedas mostradas en la Figura 32, por lo que una vuelta completa equivale al desplazamiento de 25.1328 [mm], por lo tanto, la resolución de δ_1 y δ_2 es de 0.006 [mm].

Cada vez que se inserta un instrumental quirúrgico en alguno de los puertos de entrada, las distancias δ_1 y δ_2 se ajustan a un valor cero, para calcular estas distancias se sigue el diagrama de flujo mostrado en la Figura 64.

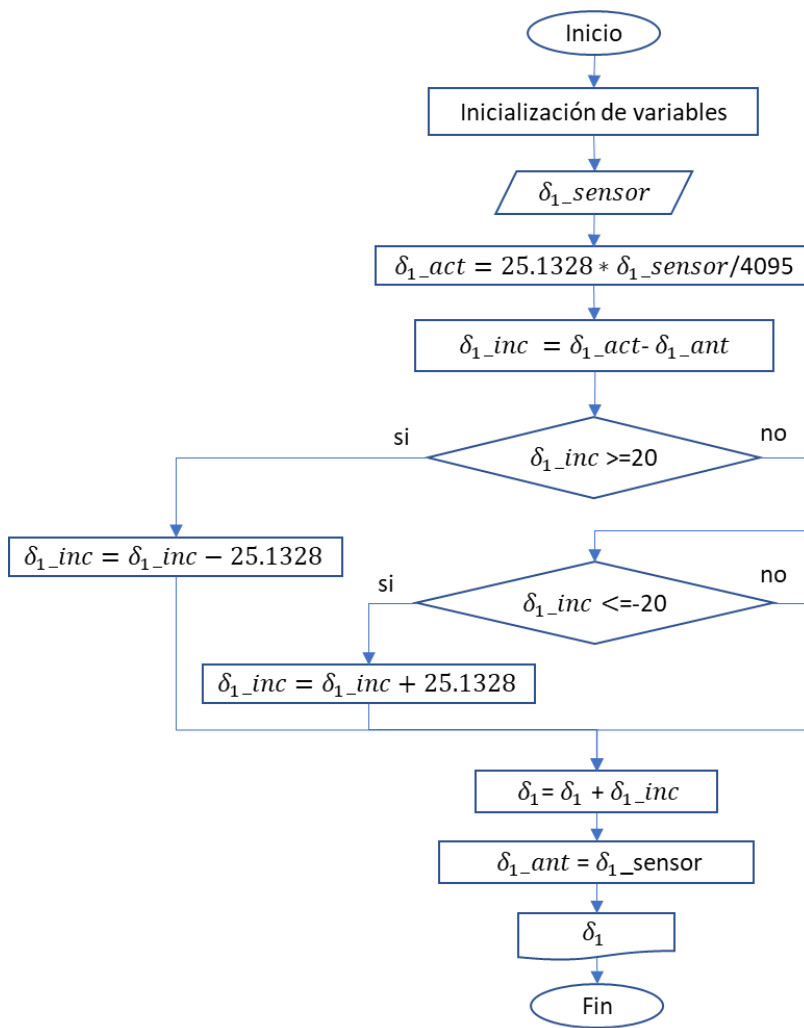


Figura 64. Diagrama de flujo para estimar el valor de las variables δ_1 y δ_2 .

El rango de las variables δ_1 y δ_2 está limitado por el tamaño del instrumental quirúrgico, por lo que el rango es de 0 [mm] a 120 [mm].

La precisión de las variables δ_1 y δ_2 se determina a partir de sus señales, cuando el instrumental quirúrgico se mantiene inmóvil, Figura 65. La precisión de las variables, que se calculan con la desviación estándar, es de $\pm 0.003^\circ$ en δ_1 y $\pm 0.002^\circ$ en δ_2 . Mientras que el error máximo observado en las señales es de ± 0.06 [mm].

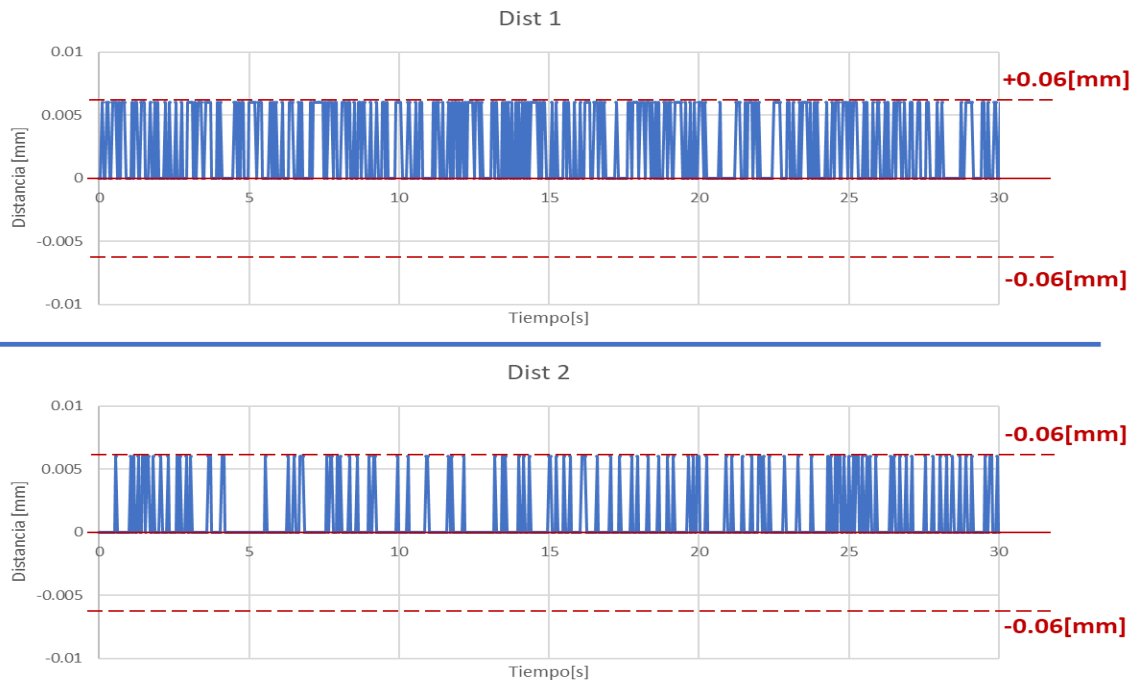


Figura 65. Señales de las variables δ_1 y δ_2 , con los instrumentos quirúrgicos inmóviles.

4.2.4 Resolución, rango y precisión del ángulo del lente del artroscopio.

La variable $giro_{art}$ representa el giro del lente del artroscopio, que se mide con el sensor *MPU9250-2* y un sensor *AS5600*. El sensor *MPU9250-2* permite estimar la orientación inicial del artroscopio, al insertarse por algún puerto de entrada de la rodilla, después dicha orientación se mide con un sensor *AS5600* conectado a un puerto *ADC* de 10 [bits]. Por lo tanto, la resolución de $giro_{art}$ es de 0.35° .

El lente del artroscopio puede girar libremente, sin límite, por lo que la variable $giro_{art}$ tiene un rango de 0° a 360° .

La precisión la variable $giro_{art}$ se determina a partir de su señal, cuando el artroscopio se mantiene inmóvil, Figura 66. La precisión de la variable $giro_{art}$, que se calcula con la desviación estándar, es de $\pm 0.056^\circ$. Mientras que el error máximo observado en la señal es de $\pm 0.25^\circ$.

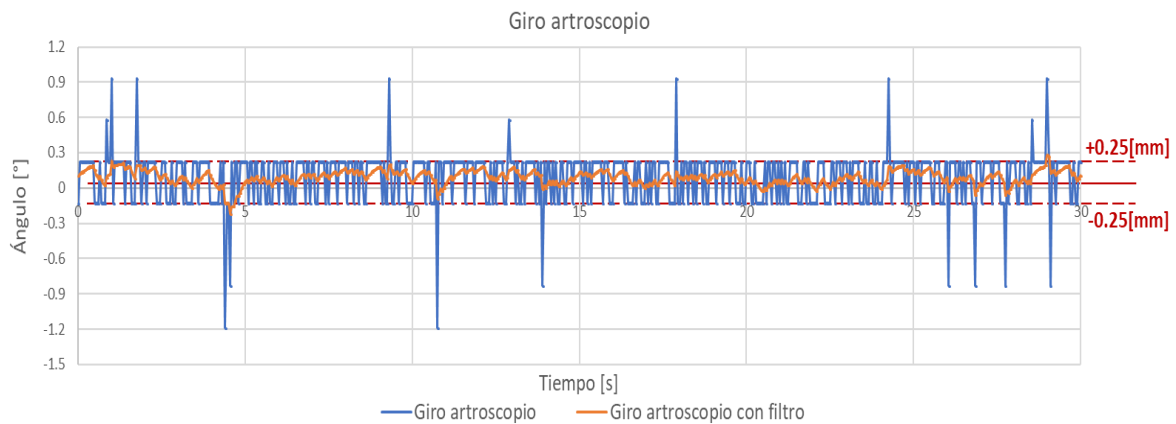


Figura 66. Señales de las variables $g_{ro_{art}}$, con el artroscopio inmóvil.

4.2.5 Resolución, rango y precisión del ángulo de apertura de las pinzas.

La variable ap_{pin} es la apertura de las pinzas que se estima con un sensor AS5600 conectado a un puerto ADC de 10 [bits]. El sensor es colocado para medir el ángulo de la articulación del cuerpo de las pinzas, que representa el rango de la apertura de las mordazas de 0° a 45° . De esta forma el valor del ADC tiene valores de 0 a 51.15, por lo tanto, la variable ap_{pin} tiene una resolución de 0.879° .

La precisión la variable ap_{pin} se determina a partir de su señal, las pinzas se mantienen inmóviles, Figura 67. La precisión de la variable ap_{pin} , que se calcula con la desviación estándar, es de $\pm 0.23^\circ$. Mientras que el error máximo observado en la señal es de $\pm 0.9^\circ$.

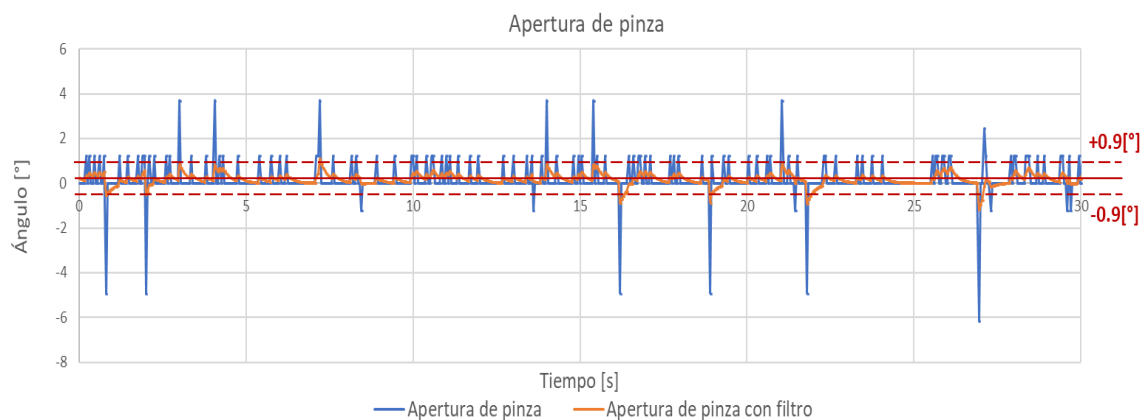


Figura 67. Señales de la variable ap_{pin} con las pinzas estáticas.

4.3 Calibración de los sensores QRE1113

Los sensores *QRE1113* se conectan a los puertos *ADC* de 10 [*bits*], por lo que sus señales tienen valores de 0 a 1023. Esta señal depende del color de la parte del instrumento quirúrgico que entra por los puertos de entrada a la rodilla, en el artroscopio es de color plateado, en las tijeras artroscópicas es de color dorado y en el palpador es de color negro. Los valores obtenidos en los *ADC* se muestran en la Tabla 4.

Tabla 4. Mediciones de los sensores *QRE1113*, para determinar las variables ins_{iz} e ins_{der} .

Instrumento	Color	Número del instrumento	Rango del sensor izq.	Rango del sensor der.
Vacío	N/A	3	400-650	550-630
Palpador	Dorado	0	0-77	0-72
Artroscopio	Plateado	1	77-100	72-110
Pinzas de artroscopia	Negro	2	650-800	630-800

Con los valores de la Tabla 4 se programa la detección de los instrumentos a partir de las mediciones de los sensores *QRE1113*, Figura 68.

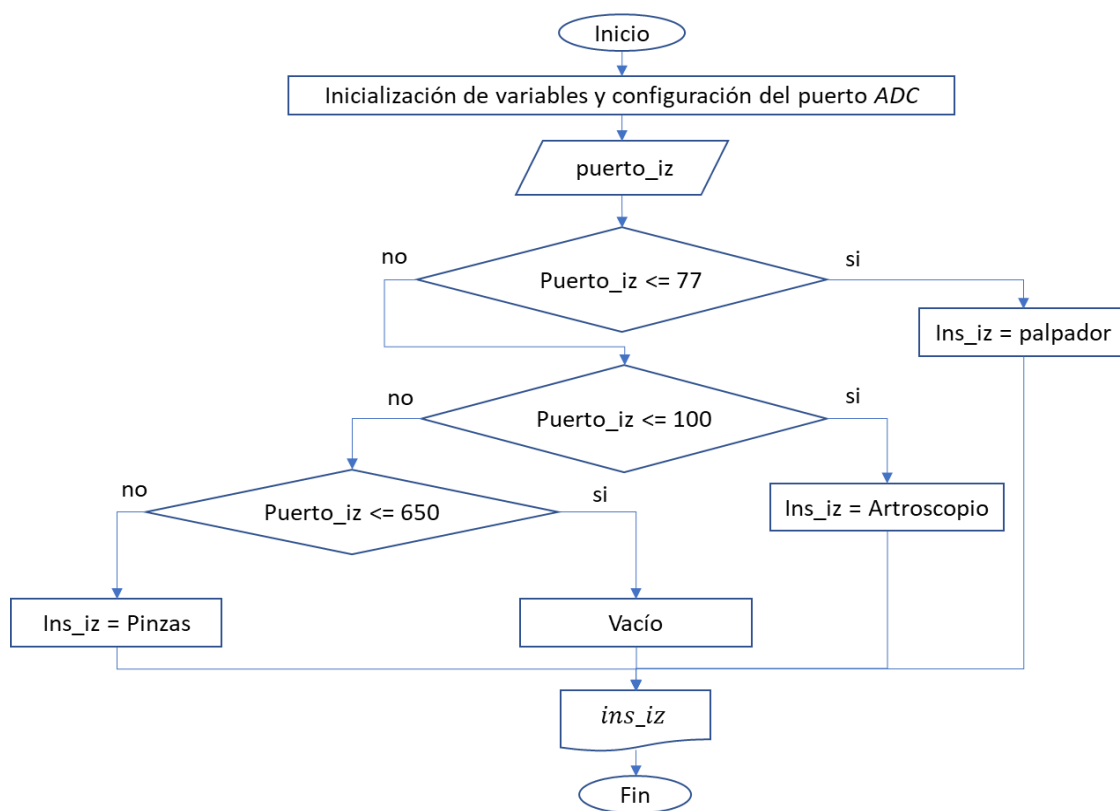


Figura 68. Diagrama de flujo para estimar el instrumental quirúrgico insertado en el puerto de entrada izquierdo, en el caso del puerto derecho cambian los valores en los condicionales “if”.

Las señales de las variables ins_{iz} e ins_{der} se muestran en la Figura 69.

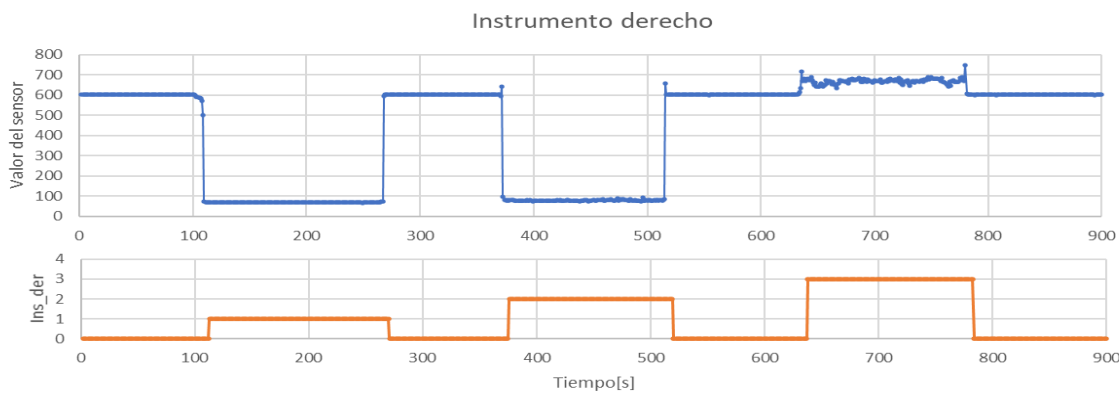
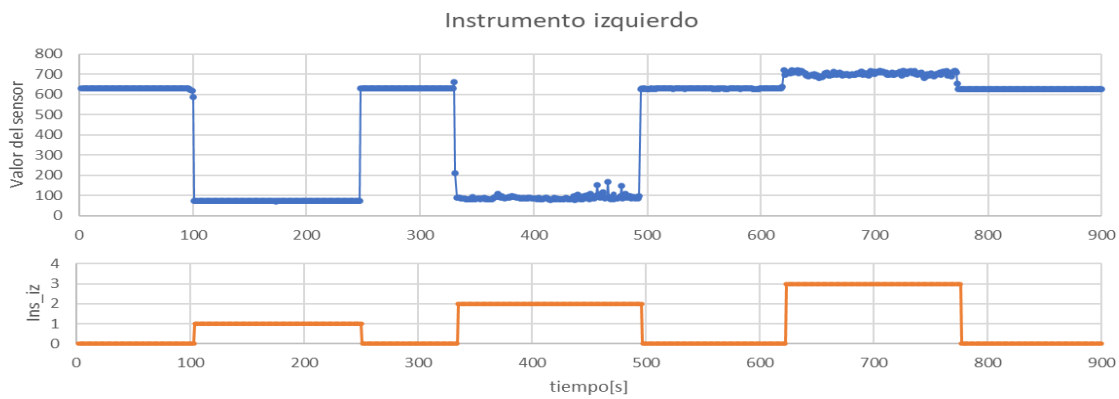


Figura 69. Señales de los sensores QRE1113 (en color azul), y señales de las variables ins_{iz} e ins_{der} , (en color naranja).

4.4 Propuesta del método para determinar la exactitud del arreglo instrumental

La exactitud del arreglo instrumental electrónico de sensores se determina a partir de la comparación de sus medidas con las de un equipo de medición con mayor precisión, resolución y exactitud conocida.

Una propuesta es utilizar una máquina de control numérico *CNC* de tres ejes, de tal forma que el actuador de dicha máquina esté fijo a las puntas del instrumental quirúrgico izquierdo y derecho. Por lo tanto, al mover el actuador de la máquina *CNC* se mueven las puntas del instrumental quirúrgico y se obtienen sus posiciones con el arreglo instrumental [66].

La metodología propuesta para determinar la exactitud del instrumento es la siguiente:

1. Determinar un equipo patrón con mejor precisión y resolución que el arreglo instrumental, además de una exactitud conocida. Por ejemplo, una máquina *CNC* de tres ejes.

2. Diseñar el banco de pruebas, que incluye la fijación de las puntas del instrumental quirúrgico en el actuador de la máquina *CNC*.
3. Realizar distintos movimientos con la máquina *CNC*, almacenando los datos tanto del arreglo instrumental como del actuador de la máquina *CNC*.
4. Comparar los movimientos del actuador de la máquina *CNC* con los movimientos de las puntas del instrumental quirúrgico del arreglo instrumental, para estudiar la diferencia entre ellas.
5. Analizar y calcular la exactitud del arreglo instrumental a partir del estudio en el paso anterior.
6. Reportar los resultados en un trabajo de investigación.

CAPÍTULO 5 PROPUESTA DE RED NEURONAL PARA LA CLASIFICACIÓN DEL TIPO DE USUARIO

5.1 Características de la red neuronal para la clasificación del usuario del simulador

5.1.1 Características generales de la red neuronal

Una tarea útil de las herramientas tecnológicas para el entrenamiento de cirujías de mínima invasión es la clasificación del tipo de usuario. Esta clasificación consiste en determinar el nivel de experiencia quirúrgica del cirujano, basada en sus movimientos con el instrumental quirúrgico, al realizar diversas tareas como triangulación de instrumentos, ubicación espacial y transferencia de objetos.

Una red neuronal puede aplicarse para clasificar al tipo de usuario, donde los datos de entrada son los datos proporcionados por el arreglo instrumental, y los datos de salida indiquen el nivel de experiencia quirúrgica del usuario.

La implementación de una red neuronal tiene las siguientes ventajas:

1. Procesar un gran volumen de datos de entrada y de salida.
2. La clasificación de los usuarios se automatiza, ya que la red neuronal clasifica al usuario inmediatamente después de terminar las actividades en el arreglo instrumental.
3. La red no está limitada en el número de clasificaciones, es decir podría considerarse un rango amplio para clasificar al usuario, por ejemplo: principiante-A, Principiante-B, Principiante-C, Intermedio-A, Intermedio-B, Intermedio-C, Experto A, Experto B.
4. Al estudiar el funcionamiento y eficiencia de la red se puede disminuir la cantidad de datos de entrada, lo que supone un funcionamiento más rápido a un menor costo computacional.

La implementación de la red neuronal propuesta en este trabajo está basada en dos procedimientos principales. El primero consiste en el entrenamiento de la red, a partir de una base de datos. El segundo consiste en el reconocimiento de la red, donde se clasifica a los usuarios que no participan en el entrenamiento, Figura 70.

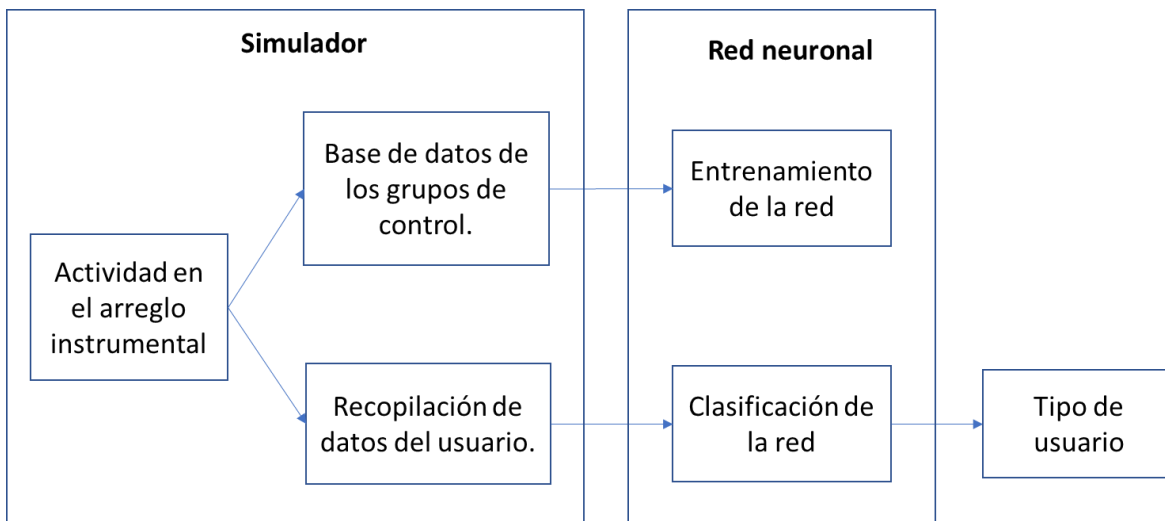


Figura 70. Proceso de la implementación de la red neuronal para clasificar al tipo de usuario.

5.1.2 Variables de entrada y salida de la red neuronal

Las mediciones del arreglo instrumental permiten calcular los Parámetros de Análisis de Movimiento (*Motion Analysis Parameters*, MAP por sus siglas en inglés) mostrados en la Tabla 5. Los MAP's se calculan con las posiciones de las puntas del instrumental quirúrgico, con su respectiva marca de tiempo y son utilizados para evaluar el desempeño de los usuarios que realizan tareas en los simuladores [67, 68].

Se propone una red neuronal en la que los datos de entrada son los MAP's y los datos de salida indican el nivel de experiencia del cirujano, entre experto, intermedio y principiante.

Se cuenta con una base de datos obtenida a partir del trabajo de investigación [63, 64]. Dicha base de datos utiliza el simulador *EndoViS training system*, con el que se realizan actividades en un simulador de mínima invasión. Se elige la tarea de traslado de objetos, que consiste en transferir donas de un lado al otro, Figura 71. Con un sistema óptico ortogonal se miden las posiciones de las puntas del instrumental quirúrgico izquierdo y derecho. La base de datos se conforma como se muestra en la Figura 72, y se puede consultar en el Anexo D.

Tabla 5 Parámetros de Análisis de Movimiento.

Métrica	Definición	Ecuación
Tiempo (T)	Tiempo total en realizar la tarea, en [s]	T
Longitud de la trayectoria (PL)	Trayectoria total seguida por las puntas del instrumental, en [m]	$\int_{t=0}^T \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2 + \left(\frac{dz}{dt}\right)^2} dt$
Percepción de la profundidad (DP)	Distancia total que viaja la punta del instrumento a lo largo del eje X, en [m]	$\int_{t=0}^T \sqrt{\left(\frac{dy}{dt}\right)^2 + \left(\frac{dz}{dt}\right)^2} dt$
Profundidad a lo largo del trocar (DTR)	Distancia total que viaja el instrumento respecto al trocar, en [m]	$\int_{t=0}^T \frac{(\sqrt{(x_{tr} - x_n)^2 + (y_{tr} - y_n)^2 + (z_{tr} - z_n)^2})}{dt} dt$
Suavidad del movimiento (MS)	Suavidad de los movimientos, en [mm/s^2]	$\sqrt{\frac{T^5}{2PL^2} \int_{t=0}^T \left(\left(\frac{d^3x}{dt^3}\right)^2 + \left(\frac{d^3y}{dt^3}\right)^2 + \left(\frac{d^3z}{dt^3}\right)^2 \right) dt}$
Velocidad promedio (V)	Tasa de cambio de la posición, en [mm/s]	$\frac{1}{T} \int_{t=0}^T \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2 + \left(\frac{dz}{dt}\right)^2} dt$
Aceleración promedio (A)	Tasa de cambio de la velocidad, en [mm/s^2]	$\frac{1}{T} \int_{t=0}^T \sqrt{\left(\frac{d^2x}{dt^2}\right)^2 + \left(\frac{d^2y}{dt^2}\right)^2 + \left(\frac{d^2z}{dt^2}\right)^2} dt$
Tiempo inactivo (IT)	Porcentaje del tiempo del instrumento inmóvil, en [%]	$\frac{\mathfrak{I}}{T} \cdot \mathfrak{I} = \frac{1}{T} \int_{t=0}^T \sqrt{\left(\frac{d^2x}{dt^2}\right)^2 + \left(\frac{d^2y}{dt^2}\right)^2 + \left(\frac{d^2z}{dt^2}\right)^2} dt$
Economía del área (EOA)	Relación entre el área máxima cubierta por el instrumental y la trayectoria total	$\frac{\sqrt{[Max(x) - Min(x)] \cdot [Max(y) - Min(y)]}}{PL}$
Economía del volumen (EOV)	Relación entre el volumen máximo cubierto por el instrumental y la trayectoria total	$\frac{\sqrt[3]{[Max(x) - Min(x)] \cdot [Max(y) - Min(y)] \cdot [Max(z) - Min(z)]}}{PL}$
Energía del área (EA)	Energía para mover el instrumental sobre el área cubierta, en [J/cm^2]	$\frac{\sum_{i=0}^T x_i ^2 + \sum_{i=0}^T y_i ^2}{[Max(x) - Min(x)] \cdot [Max(y) - Min(y)]}$
Energía del volumen (EV)	Energía para mover el instrumental sobre el volumen cubierto, en [J/cm^3]	$\frac{\sum_{i=0}^T x_i ^2 + \sum_{i=0}^T y_i ^2 + \sum_{i=0}^T z_i ^2}{[Max(x) - Min(x)] \cdot [Max(y) - Min(y)] \cdot [Max(z) - Min(z)]}$

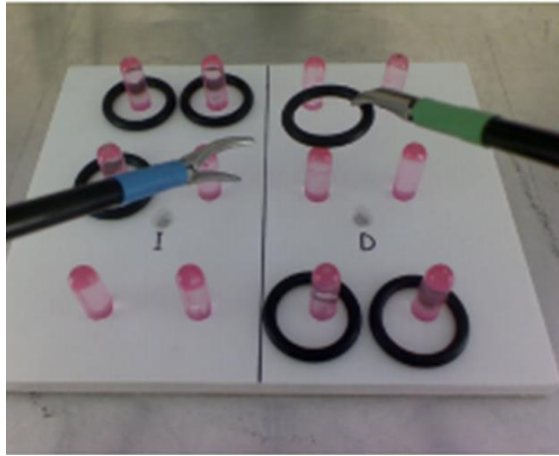


Figura 71. Transferencia de objetos, en el simulador Endo ViS training system, obtenido de [67].

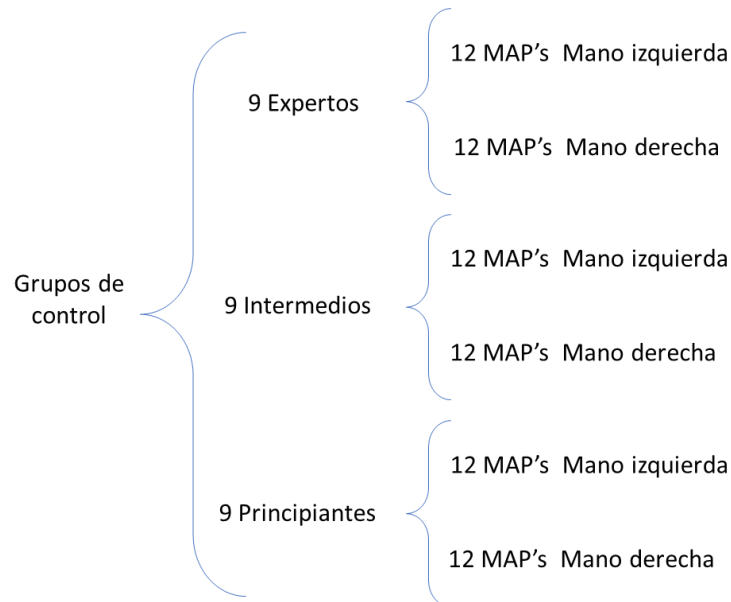


Figura 72. Estructura de la base de datos para el entrenamiento de la red neuronal.

La base de datos de la Figura 72 se generó con la participación de tres grupos de control:

1. El grupo principiante está conformado por 9 usuarios sin experiencia en simuladores, ni en cirugías de mínima invasión.
2. El grupo intermedio está conformado por 9 cirujanos con experiencia en simuladores o incluso en prácticas en cirugías laparoscópicas, con supervisión de un experto.
3. El grupo experto está conformado por 9 cirujanos con experiencia en más de 100 procedimientos quirúrgicos en cirugías laparoscópicas.

5.2 Aplicación de la red neuronal RTC

Existen muchos tipos de redes neuronales, algunas con mayor velocidad de clasificación, pero mayor tiempo de entrenamiento. Se propone el uso de la arquitectura *Random Thershold Clasifier* o *RTC* por sus siglas en inglés, presentado en [54]. Este es un clasificador que utiliza perceptrones multicapa con una sola capa de conexiones de entrenamiento, por lo que se obtiene una alta velocidad de entrenamiento, y alta velocidad de reconocimiento, Figura 73.

El clasificador *RTC* para el reconocimiento de usuarios del simulador, tiene las siguientes características:

1. El vector de entradas (X_1, \dots, X_n) se conforma de 24 MAP's, 12 para el instrumento de la mano izquierda y 12 para el instrumento de la mano derecha.
2. El vector de salidas ($C_1 C_2 C_3$) se conforma de tres clasificaciones, experto, intermedio y principiante.
3. La información pasa por los umbrales aleatorios, si dicha información está dentro de los umbrales, entonces activan a la neurona b_s del grupo.
4. La función de activación de las neuronas en la capa entrenable C es de tipo *softmax*.
5. El entrenamiento es de tipo supervisado con el 66.66% de la base de datos, es decir se utilizan 6 usuarios de cada grupo.
6. El reconocimiento se realiza con el 33.33% de los datos, es decir 3 usuarios de cada grupo. De esta manera se determina el correcto funcionamiento y una aproximación al error de la red.

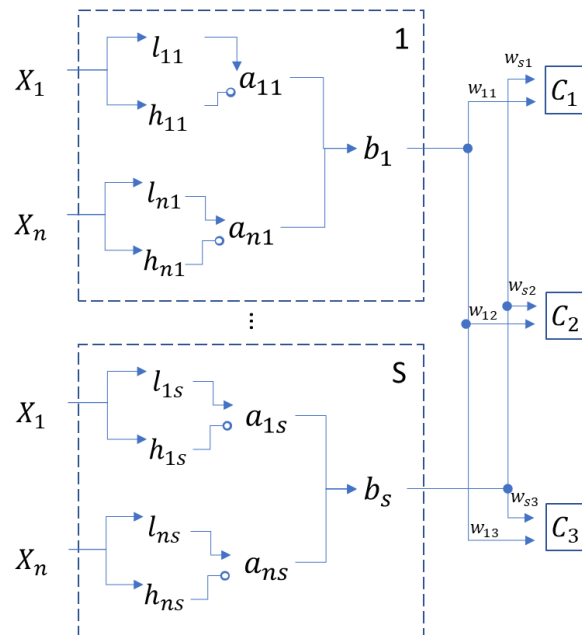


Figura 73. Propuesta del clasificador RTC para la clasificación del tipo de usuario del simulador.

En la figura 73 se observa:

X_1, \dots, X_n son los *MAP's* de la mano izquierda y la mano derecha.

1-S es el número de neuronas, 32 000.

b_s es la salida de cada grupo S.

$l_{n,s}$ y $h_{n,s}$ son los umbrales superiores e inferiores de la red, generados de forma aleatoria.

$a_{n,s}$ son las neuronas de la capa intermedia.

C_1 , C_2 y C_3 son las neuronas de la capa de salida, que indican si el usuario es experto, intermedio o principiante

w_i son los pesos asociados que se modifican durante el entrenamiento de la red.

En la Figura 73 se muestran los bloques 1-S que representan la cantidad de grupos de umbrales y de neuronas de salida b , se eligen 32,000. Las variables de entrada X_i, \dots, X_n son los valores de 24 variables *MAP's* del instrumental izquierdo y derecho. Las neuronas $l_{n,s}$ y $h_{n,s}$ representan el valor mínimo y máximo respectivamente del umbral. La neurona $a_{n,s}$ recibe la señal de excitación de la neurona $l_{n,s}$ y la señal deshabilitada de la neurona $h_{n,s}$. Las neuronas b_n se activan cuando todas las neuronas $a_{n,s}$ son excitadas, y representa la señal de salida de todo el bloque S. Las neuronas C se conectan con todas las neuronas b_s por conexiones entrenables, por lo que, se tienen pesos asociados w_i que se modifican durante el entrenamiento de la red.

La programación del clasificador *RTC* se desarrolla en tres bloques principales. En el primero se generan las capas no entrenables de la red, es decir las neuronas con los umbrales l y h , las neuronas a , y las neuronas b . En el segundo se realiza el entrenamiento de la red, utilizando un *perceptrón* simple que conecta a la capa de neuronas b con la capa de neuronas C . En el tercer bloque se hace un reconocimiento de los datos de los usuarios que no participaron en la fase de entrenamiento, de esta forma se determina el error de la red. El código del clasificador *RTC* se encuentra en el Anexo E.

5.3 Aplicación de la red neuronal *perceptrón*

Se programa una red neuronal *perceptrón* en el programa *Matlab*. Al igual que con el clasificador *RTC*, las entradas a la red neuronal son los *MAP's* de la base de datos descrita anteriormente, y las salidas representan el nivel de experiencia quirúrgica del usuario, entre principiante, intermedio y experto, Figura 74.

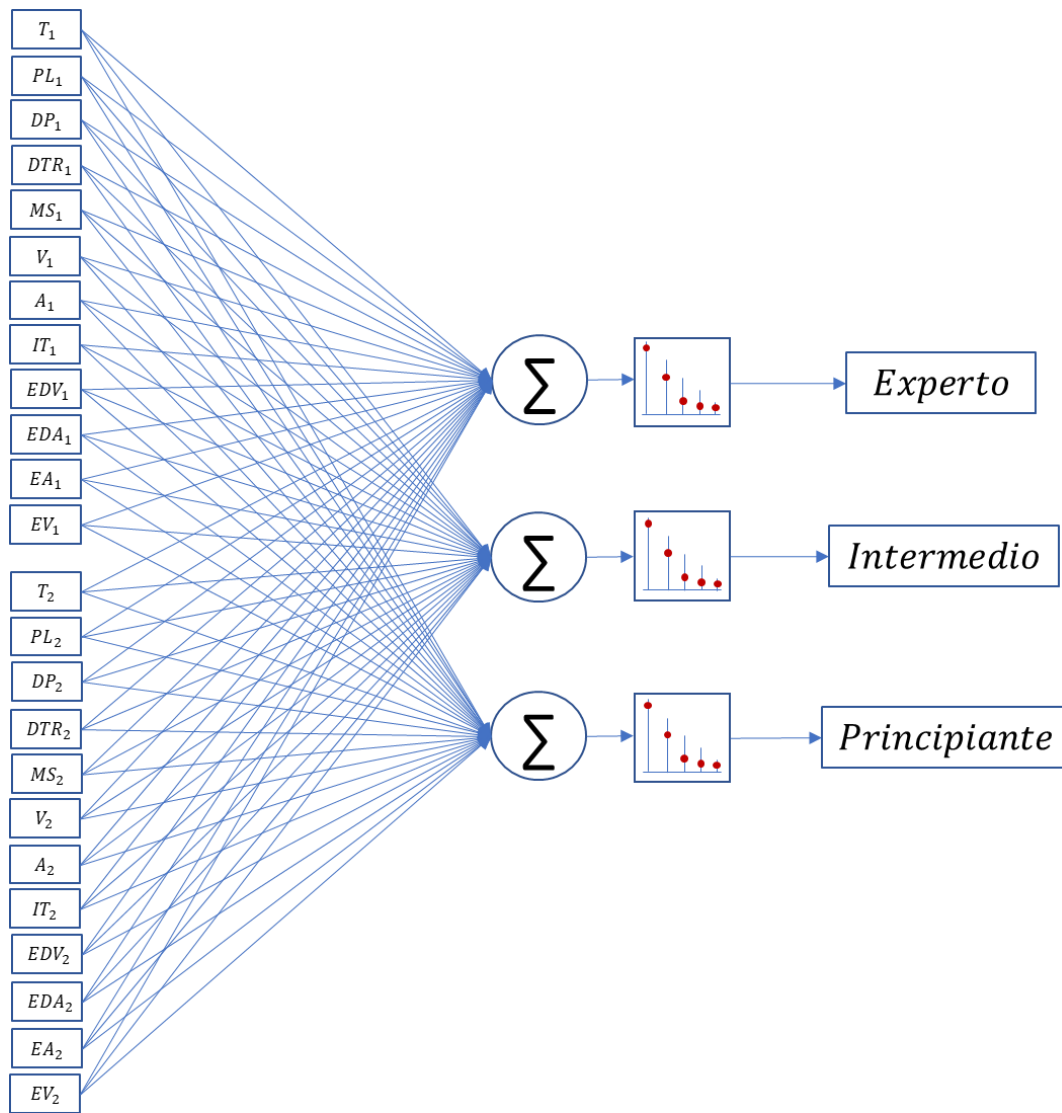


Figura 74. Red neuronal perceptrón simple.

El código de la red neuronal con *perceptrón* se puede consultar en el Anexo F. Esta red neuronal tiene las siguientes características:

1. El vector de entradas es de 24 MAP's, 12 para el instrumento de la mano izquierda y 12 para el instrumento de la mano derecha.
2. El vector de salidas se conforma de tres clasificaciones, experto, intermedio y principiante.
3. La función de activación de las tres neuronas de salida es *softmax*.
4. El entrenamiento es de tipo supervisado con el 66.66% de la base de datos, es decir se utilizan 6 usuarios de cada grupo.
5. El reconocimiento se realiza con el 33.33% de los datos, es decir 3 usuarios de cada grupo. De esta manera se determina el error aproximado de la red.

CAPÍTULO 6. RESULTADOS

6.1 Características generales del arreglo instrumental

El arreglo instrumental permitió medir los movimientos de los instrumentos quirúrgicos y del modelo de la rodilla, con estas mediciones se estiman 16 variables, 25 veces por segundo:

- 4 grados de libertad de la articulación: 3 giros y un desplazamiento.
- 4 grados de libertad del instrumento insertado en el puerto izquierdo: 3 giros y un desplazamiento.
- 4 grados de libertad del instrumento insertado en el puerto derecho: 3 giros y un desplazamiento.
- 2 instrumentos insertados en la rodilla: cada puerto detecta cuando el puerto está vacío, cuando se inserta el palpador, artroscopio o pinzas de artroscopia.
- 1 grado de libertad en el lente del artroscopio.
- 1 grado de libertad en la apertura de las pinzas.

Las variables anteriores fueron suficientes para simular los movimientos que se realizan durante una cirugía artroscópica. Para estimar estas variables se utilizan los siguientes sensores:

- 9 sensores inerciales: 3 acelerómetros y 3 giróscopos del *MPU9250-1*, 3 acelerómetros del *MPU9250-2*.
- 13 sensores de efecto hall: 10 *AS5600* y 3 magnetómetros del *MPU9250-1*.
- 2 sensores reflectivos: 2 sensores *QRE1113*.

El microcontrolador recibe las señales de los sensores a través del protocolo I^2c y por convertidores *ADC*, después calcula las variables mencionadas anteriormente, y las envía a la interfaz virtual. Las características de estas variables se muestran en la Tabla 6.

Tabla 6 Características de las variables del arreglo instrumental.

Variable	Descripción	Resolución	Rango	Precisión con la desviación estándar	Error máximo observado en la señal
γ_0	Giro roll sobre el eje X de la articulación de rodilla.	N/A	-40° a 75°	$\pm 0.047^\circ$	$\pm 0.1^\circ$
β_0	Giro pitch sobre el eje Y de la articulación de rodilla.	N/A	-10° a 13°	$\pm 0.036^\circ$	$\pm 0.1^\circ$
α_0	Giro yaw sobre en el eje Z de la articulación de rodilla.	N/A	-15° a 18°	$\pm 0.161^\circ$	$\pm 0.2^\circ$
δ_0	Distancia sobre el eje Z de la articulación de rodilla	N/A	0 a 10.8 [mm]	± 0.021 [mm]	± 0.06 [mm]
γ_1	Giro roll sobre el eje X del puerto de entrada izquierdo	0.0879°	-21° a 22°	$\pm 0.042^\circ$	$\pm 0.088^\circ$
β_1	Giro pitch sobre el eje Y del puerto de entrada izquierdo	0.7565°	0 a 360°	$\pm 0.093^\circ$	$\pm 0.4^\circ$
α_1	Giro yaw sobre el eje Z del puerto de entrada izquierdo	0.0879°	-7° a 100°	$\pm 0.043^\circ$	$\pm 0.088^\circ$
δ_1	Distancia sobre el eje Y del puerto de entrada izquierdo	0.006 [mm]	0 a 100 [mm]	± 0.003 [mm]	0.06 [mm]
γ_2	Giro roll sobre el eje X del puerto de entrada derecho	0.0879°	-21° a 22°	$\pm 0.017^\circ$	$\pm 0.088^\circ$
β_2	Giro pitch sobre el eje Y del puerto de entrada derecho	0.7565°	0 a 360°	$\pm 0.07^\circ$	$\pm 0.4^\circ$
α_2	Giro yaw sobre el eje Z del puerto de entrada derecho	0.0879°	-100° a 7°	$\pm 0.039^\circ$	$\pm 0.088^\circ$
δ_2	Distancia sobre el eje Y del puerto de entrada derecho	0.006 [mm]	0 a 100 [mm]	± 0.002 [mm]	0.06 [mm]
gir_{art}	Giro del lente del artroscopio	0.35°	0 a 360°	$\pm 0.056^\circ$	$\pm 0.25^\circ$
ap_{pin}	Apertura de las pinzas de artroscopia	1.137°	0 a 45°	$\pm 0.23^\circ$	$\pm 0.9^\circ$
ins_{izq}	Instrumento insertado en el puerto de entrada izquierdo	1	0 a 3	N/A	N/A
ins_{der}	Instrumento insertado en el puerto de entrada derecho	1	0 a 3	N/A	N/A

El rango, la resolución y la precisión de cada variable es adecuada para la simulación de movimientos de una cirugía artroscópica. Se observa que el rango de movimiento permite mover el modelo físico de la rodilla y los instrumentos para simular movimientos de cirugías artroscópicas. La resolución alcanzada permite observar los movimientos en la simulación que apenas se perciben visualmente en los modelos reales. La precisión de los sensores permite ver en la simulación los movimientos continuos que siguen el movimiento de los modelos reales de la rodilla e instrumentos.

Los filtros digitales implementados en el arreglo instrumental, Tabla 7, permiten mejorar la precisión de las variables medidas y disminuir el error máximo observado en sus señales. Las frecuencias de corte de los filtros se determinaron experimentalmente, a partir de la disminución de ruido y el retraso generado en las señales.

Tabla 7. Filtros digitales del arreglo instrumental.

Tipo de filtro digital	Descripción	Variables	Frecuencia de corte
Complementario	Para estimar los tres giros del modelo de rodilla.	γ_0, β_0 y α_0	0.2 [Hz]
Pasa bajas	Para disminuir el ruido del giro sobre el eje Z de la articulación de rodilla.	α_0	2.5 [Hz]
Pasa bajas	Para estimar los giros del instrumental quirúrgico sobre su propio eje.	β_1 y β_2	1 [Hz]
Pasa bajas	Para la estimación del giro del lente del artroscopio.	gir_{art}	1 [Hz]
Pasa bajas	Para la estimación de la apertura de las pinzas de artroscopia.	ap_{pin}	1 [Hz]

6.2 Resultados del *filtro complementario*

El *filtro complementario* elimina parte del ruido de las señales de los sensores inerciales en el MPU9250-1, además combina las mediciones de los giróscopos, acelerómetros y magnetómetros. El filtro disminuye principalmente dos tipos de ruido, el de baja frecuencia en los giróscopos, y el de alta frecuencia en los acelerómetros y magnetómetros.

Con la articulación inmóvil se varía la frecuencia de corte, Figura 75, para observar el ruido que se elimina.

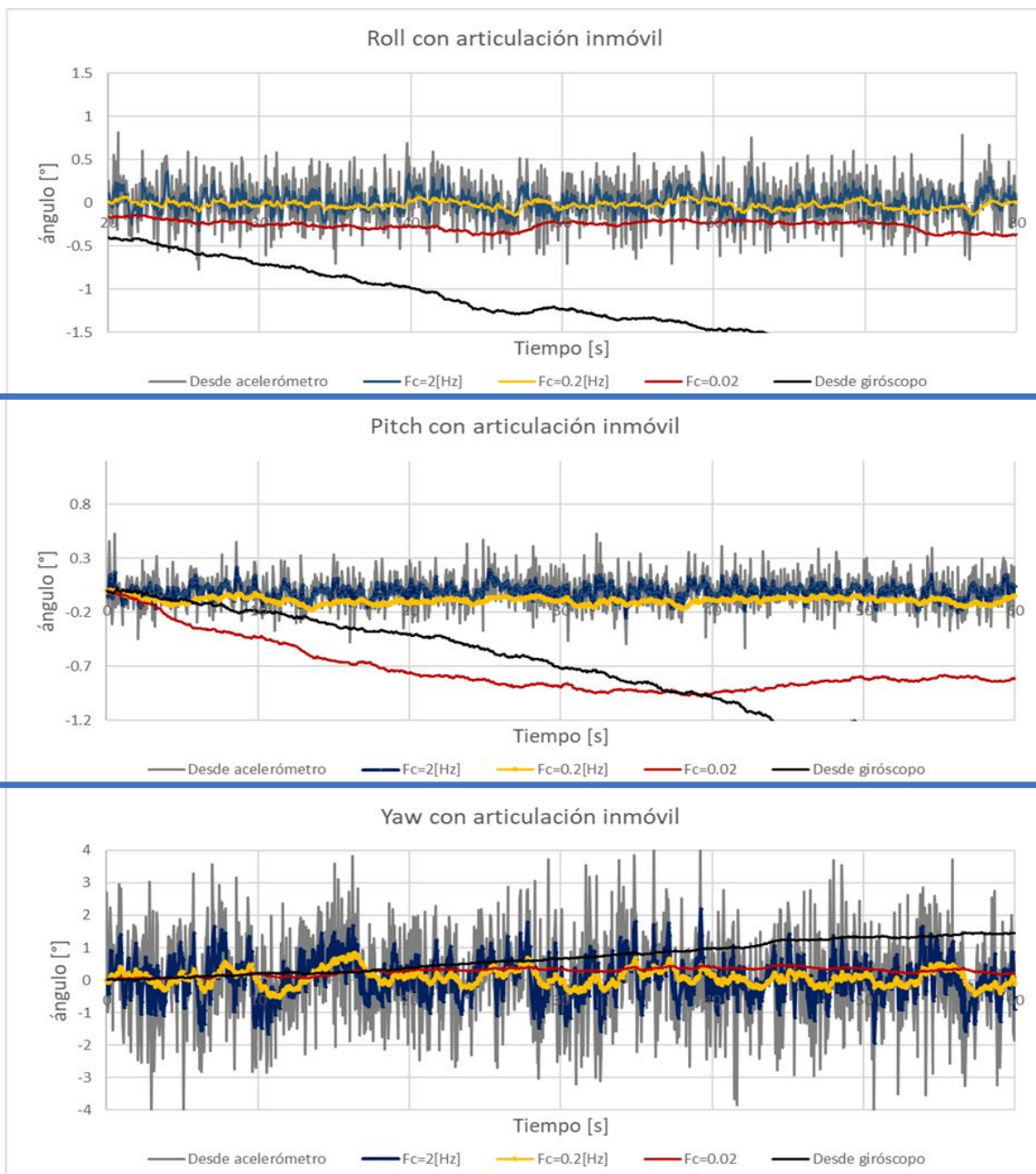


Figura 75. Señales de los giros de la articulación de rodilla, utilizando un filtro complementario con diferentes frecuencias de corte y con la articulación de rodilla inmóvil.

En la Figura 75 se observa que, mientras más grande sea la frecuencia de corte, el ruido eliminado disminuye y el resultado depende más de los acelerómetros y magnetómetros que de los giróscopos. Con frecuencias menores a 0.2 [Hz], el resultado depende más de los giróscopos y se presenta el problema de deriva, ya que no se elimina adecuadamente el ruido de bajas frecuencias de los giróscopos. Con frecuencias mayores a 0.2 [Hz] el nivel de ruido eliminado es notablemente menor, ya que no se elimina adecuadamente el ruido de los acelerómetros y magnetómetros.

El ángulo *yaw* presenta mayor ruido, debido a que su cálculo se basa principalmente en los magnetómetros. Para mejorar dicha señal se implementa un *filtro pasa bajas*, con una frecuencia de corte de 2.5 [Hz]. De esta manera la precisión de ese ángulo pasa de $\pm 0.9^\circ$ a $\pm 0.2^\circ$. Este filtro implica un retraso en la señal, sin embargo, el ángulo *yaw* de la articulación es el que menos está en movimiento durante las tareas artroscópicas, por lo que no afecta significativamente al simular sus movimientos en la interfaz.

El tiempo de estabilización inicial del filtro depende de la frecuencia de corte, Figura 76. Mientras la frecuencia de corte es mayor, el tiempo de estabilización inicial y la filtración de ruido es menor. Mientras la frecuencia de corte es menor, el tiempo de estabilización inicial y la filtración de ruido es mayor.

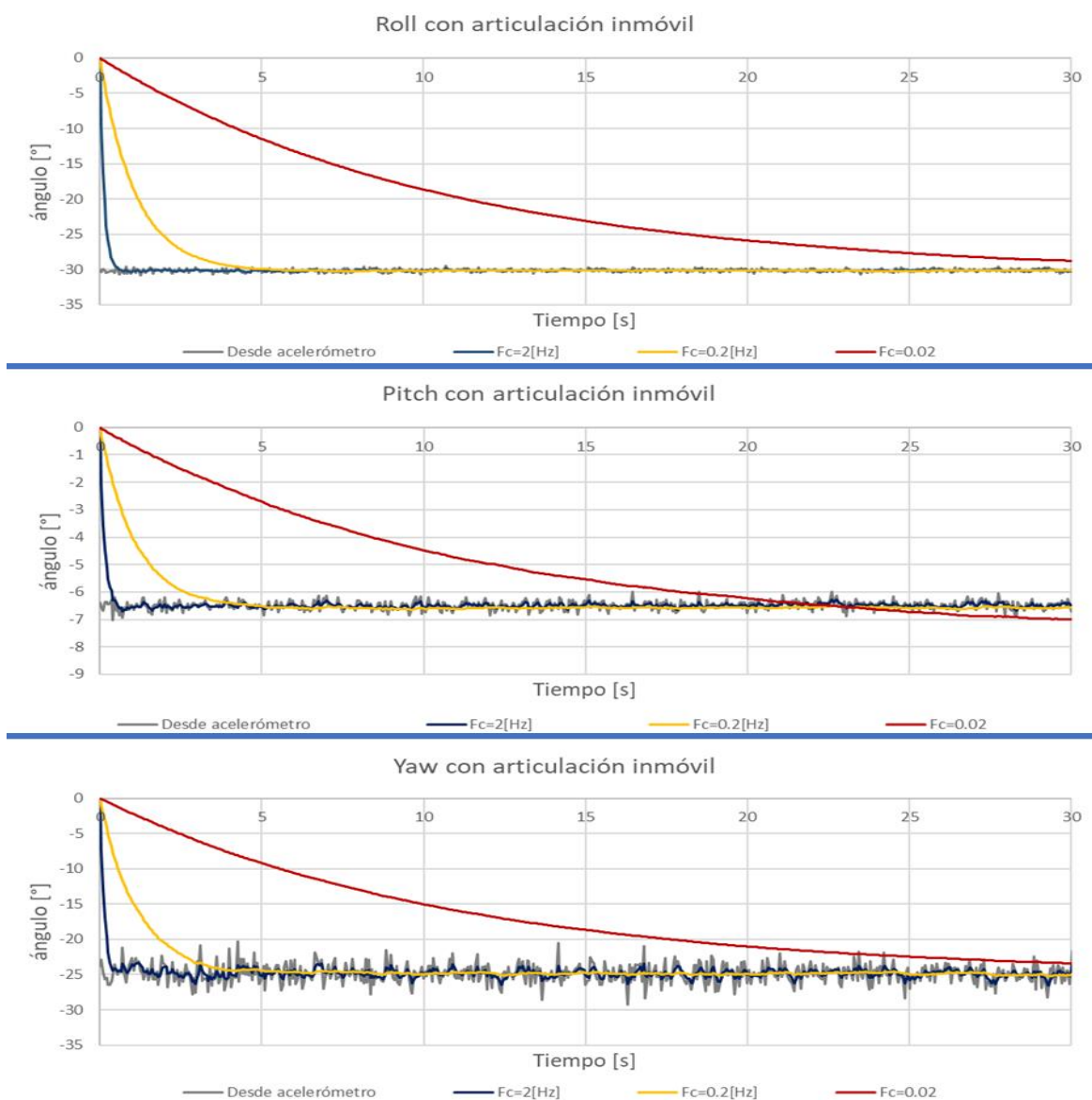


Figura 76. Tiempo de estabilización inicial del filtro complementario a diferentes frecuencias de corte.

En la Figura 76 se observa que: con una $f_c=0.02$ [Hz] el sistema tarda 35 [s] en estabilizarse inicialmente, con una $f_c=0.2$ [Hz] tarda alrededor de 4 [s], y con una $f_c=2$ [Hz] tarda alrededor de 1[s].

En la Figura 77 se observa el comportamiento del filtro conforme varía su frecuencia de corte, cuando el modelo de rodilla está en movimiento.

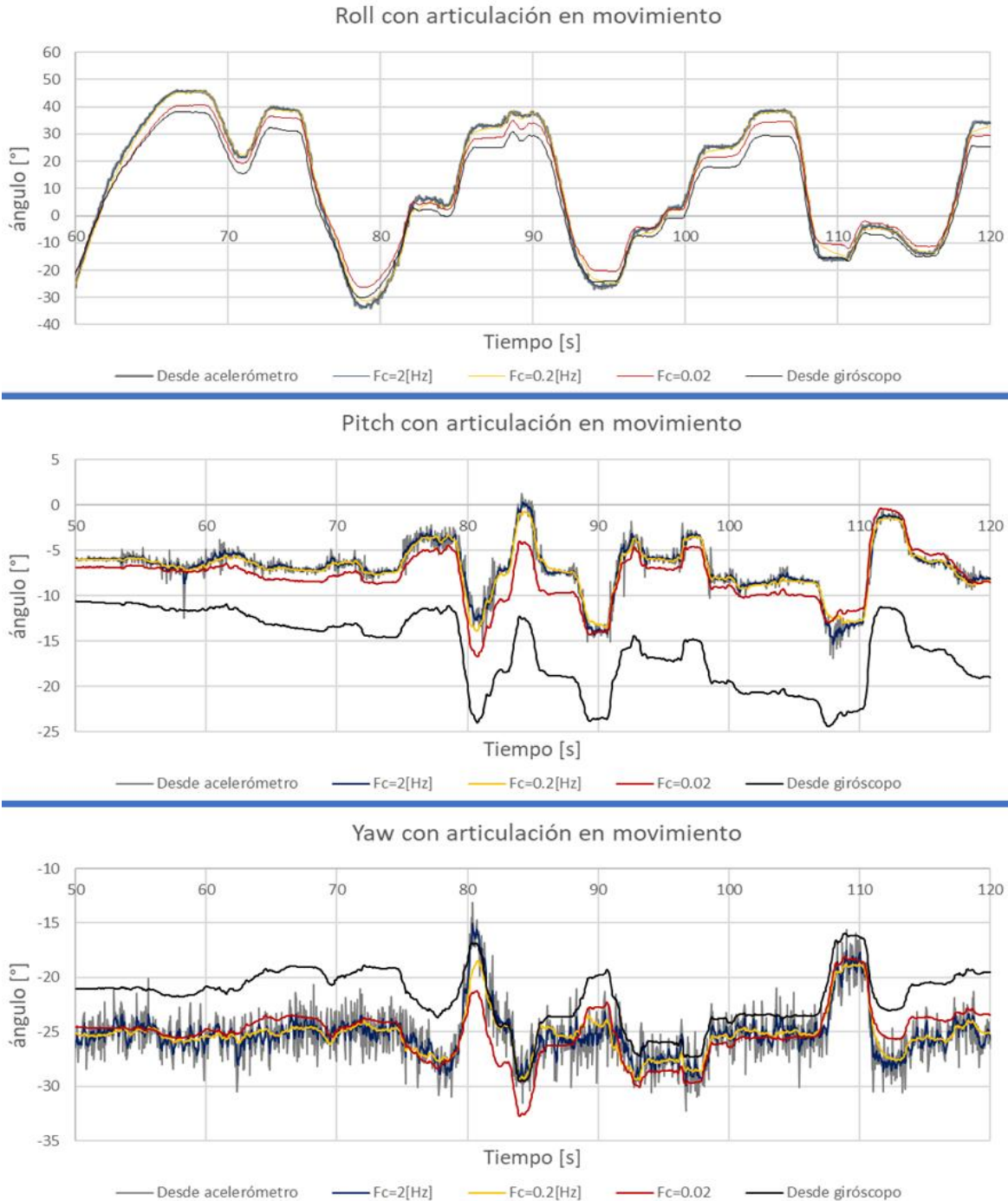


Figura 77. Señales de los giros de la articulación de rodilla con diferentes frecuencias de corte.

En la Figura 77 se observa que con una $fc=0.02$ [Hz] el filtro da mayor peso a los giróscopos y aparece el error de deriva. Con una $fc=2$ [Hz] el filtro da mayor peso a los acelerómetros y magnetómetros, por lo que no se elimina adecuadamente el ruido de altas frecuencias. Con una $fc=0.2$ [Hz] se obtiene mejor estabilidad, eliminando el ruido de baja frecuencia de los giróscopos y gran parte del ruido de alta frecuencia de los acelerómetros y magnetómetros.

Como se observa en la Figura 75, 76 y 77 la frecuencia adecuada para el *filtro complementario* es de 0.2 [Hz] ya que se combinan las señales de los acelerómetros y magnetómetros con las de los giróscopos, de tal forma que cuando el movimiento es lento o el modelo permanece inmóvil, se da prioridad a las mediciones de los acelerómetros y magnetómetros, mientras que, con movimientos rápidos se da prioridad a las mediciones de los giróscopos.

6.3 Precisión y rango de la posición del instrumental quirúrgico

Las posiciones de las puntas del instrumental quirúrgico se utilizan para analizar el desempeño del usuario, por ello, es importante calcular y almacenar dicha información. En el Capítulo 3.3.2 se presentan las ecuaciones para el cálculo de ${}^A P_1$ y ${}^A P_2$, que es la posición de la punta del instrumental izquierdo y derecho respectivamente.

La posición de la punta del instrumento en el puerto izquierdo es:

$${}^A P_1 = \{{}^A P_{x_1}, {}^A P_{y_1}, {}^A P_{z_1}\} \quad (6.1)$$

La posición de la punta del instrumento en el puerto derecho es:

$${}^A P_2 = \{{}^A P_{x_2}, {}^A P_{y_2}, {}^A P_{z_2}\} \quad (6.2)$$

La precisión de la posición de los instrumentos se determina a partir del análisis del ruido de los componentes de ${}^A P_1$ y ${}^A P_2$, cuando las puntas del instrumental quirúrgico están inmóviles. El ruido de estos componentes es mayor o menor dependiendo de la orientación de la articulación de rodilla, Figura 78. El ruido es mayor cuando la articulación está a 90°.

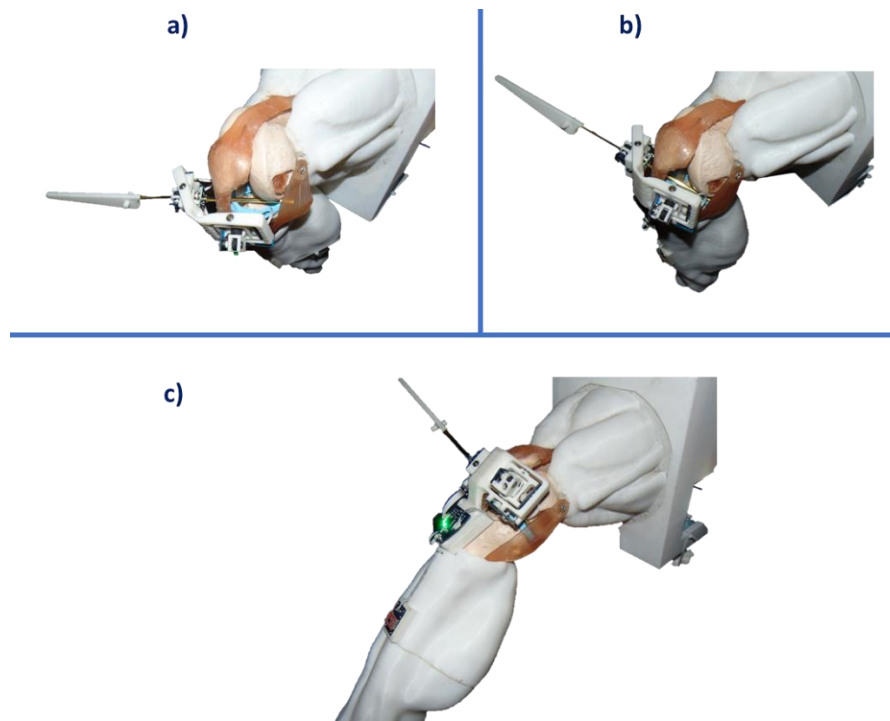


Figura 78. Posiciones de la articulación de rodilla para analizar el ruido de las señales de las puntas de los instrumentos, a) articulación flexionada, b) articulación a 90°, c) articulación estirada.

El ruido de las variables ${}^A P_1$ y ${}^A P_2$, Figura 79 y 80, es causado principalmente por el ruido de los sensores en el MPU9250-1, dicho ruido incrementa cuando el modelo de la rodilla se encuentra en la orientación *b* de la Figura 78. Las pruebas fueron realizadas con el simulador fijo a un escritorio de metal, a 300 [mm] del equipo de cómputo.

La capacidad del *filtro complementario* y de la precisión de todos los sensores permite llegar a una precisión de la posición del instrumental quirúrgico de ± 0.1 [mm], basada en el cálculo con la desviación estándar, y a un error máximo observado en la señal de ± 0.32 [mm].

El ruido de las variables ${}^A P_1$ y ${}^A P_2$ disminuye a medida que el simulador se aleja de aparatos electrónicos, de bases o mesas de metal, debido a que los magnetómetros en el MPU9250-1 son sensibles a estos objetos.

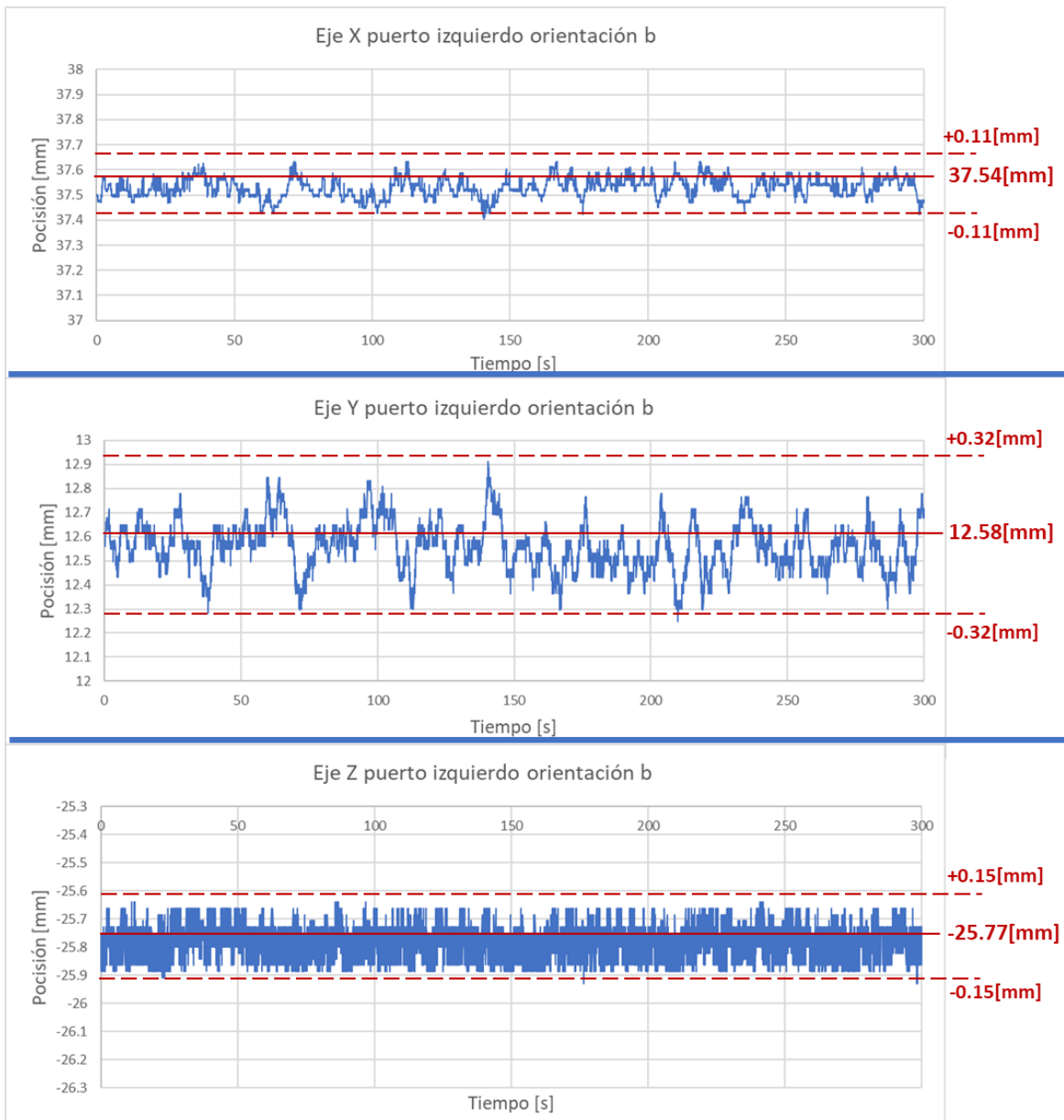


Figura 79. Señales de las variables que determinan la posición del instrumental quirúrgico izquierdo, con la articulación de rodilla inmóvil, en la orientación b.

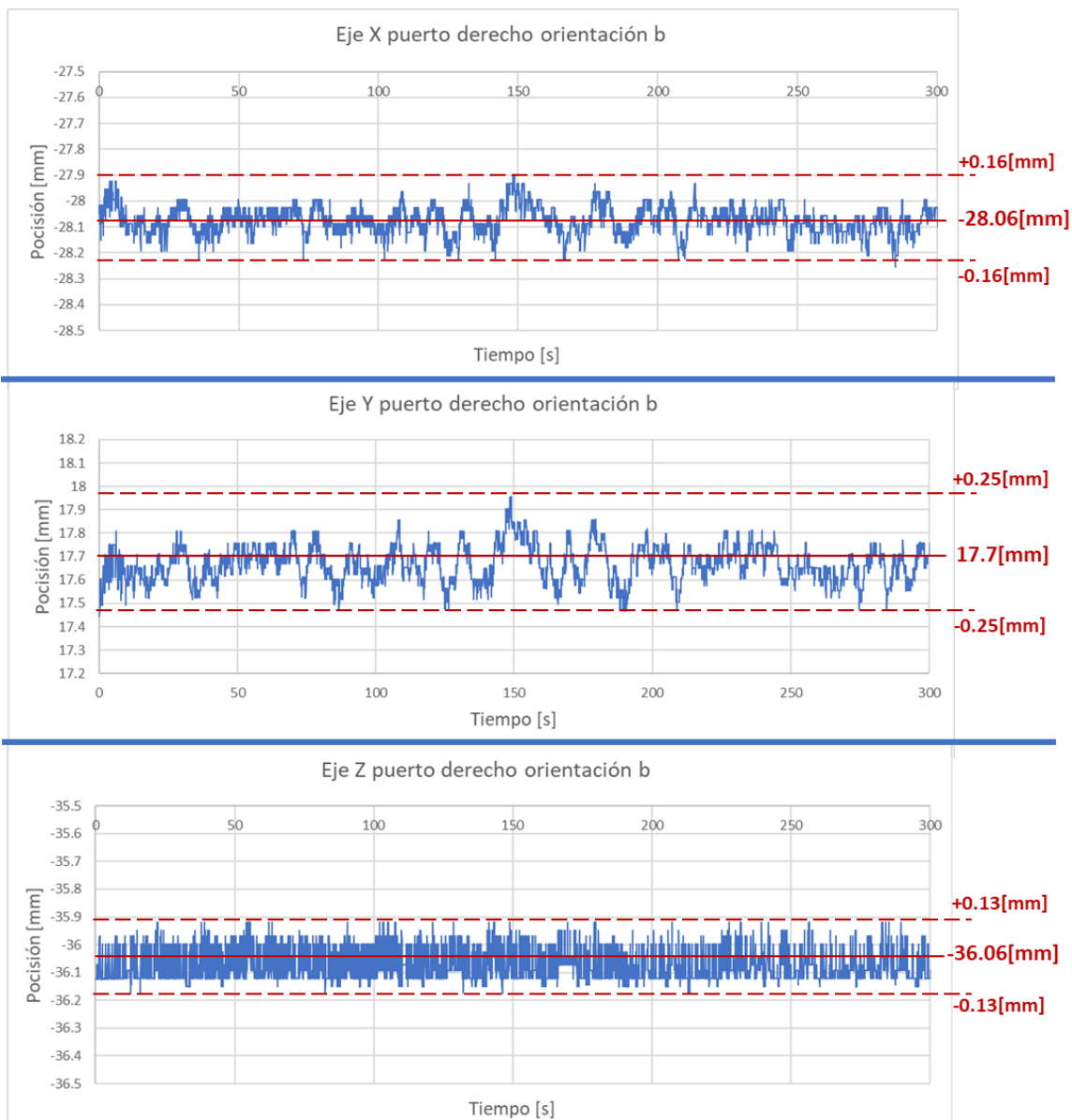


Figura 80. Señales de las variables que determinan la posición del instrumental quirúrgico derecho, con la articulación de rodilla inmóvil, en la orientación b.

Los errores que se observan en las mediciones de la posición de los instrumentos quirúrgicos son de tipo aleatorios. Es importante verificar que las mediciones del arreglo instrumental no generan errores de tipo sistemáticos. Por lo que se realizan pruebas donde se siguen trayectorias con las puntas del instrumental repetidamente.

En las Figuras 81 y 82 se muestran los movimientos del instrumental quirúrgico, cuando la articulación está en la orientación b de la Figura 78, ya que en esta posición existe mayor libertad de movimiento del instrumental. Los movimientos consisten en tocar con el palpador la superficie rotular del fémur y seguir su geometría.

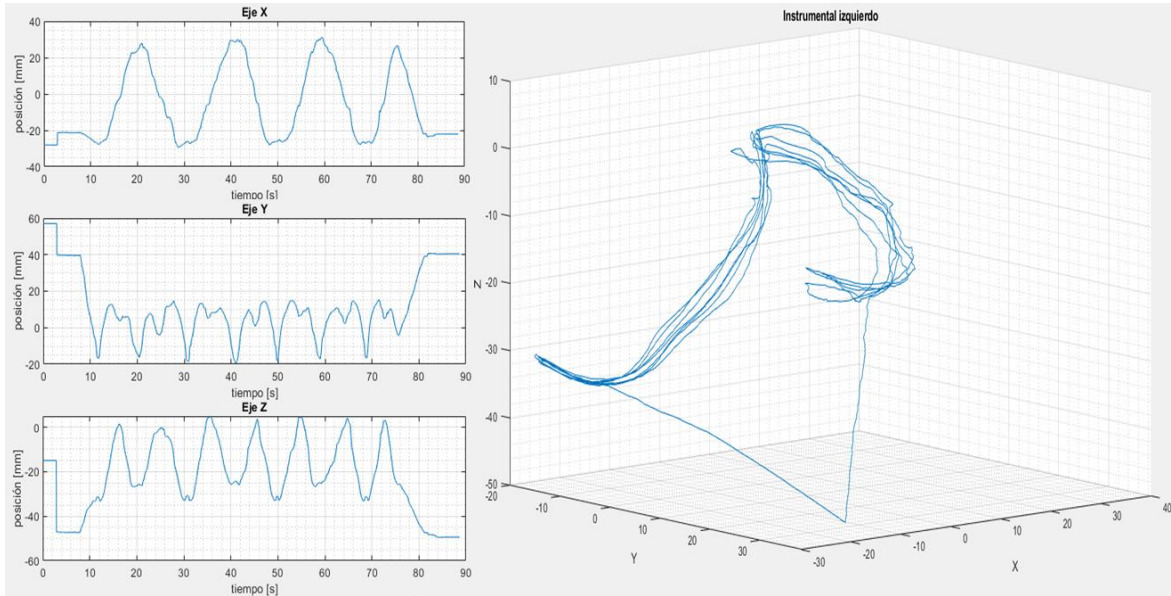


Figura 81. Movimientos del instrumental quirúrgico izquierdo.

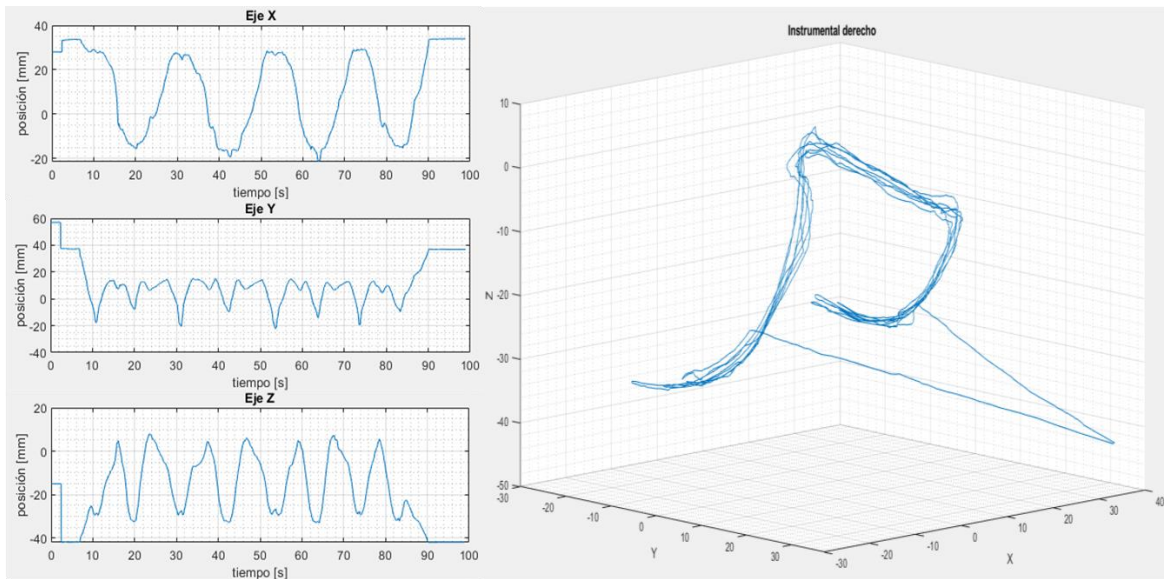


Figura 82. Movimientos del instrumental quirúrgico derecho.

Trayectorias circulares y rectas se siguieron con las puntas del instrumental quirúrgico. Las guías de las trayectorias se imprimieron con filamento *PLA* en una impresora 3D, por lo que su superficie es rugosa y con imperfecciones, Figura 83.

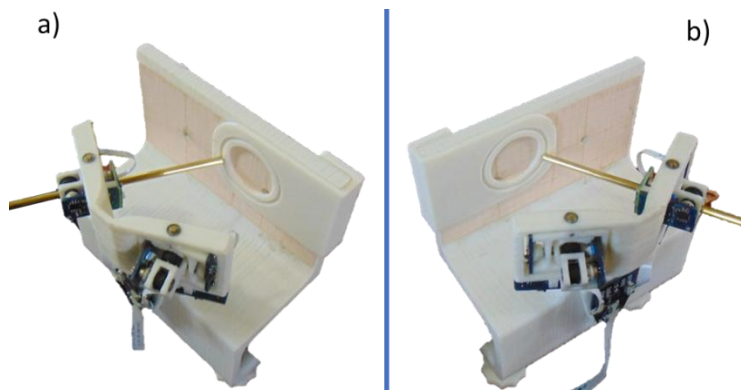


Figura 83. Soporte con geometrías para realizar pruebas de movimiento. a) Instrumental quirúrgico izquierdo con trayectoria circular. b) Instrumental quirúrgico derecho, con trayectoria circular.

Se siguió una trayectoria circular con ayuda de una guía impresa. Como se muestra en la Figura 83, la guía tiene una hendidura de 1 [mm] de espesor, un diámetro interno de 37 [mm] y un diámetro externo de 38 [mm]. La trayectoria registrada por el simulador se muestra en la Figura 84.

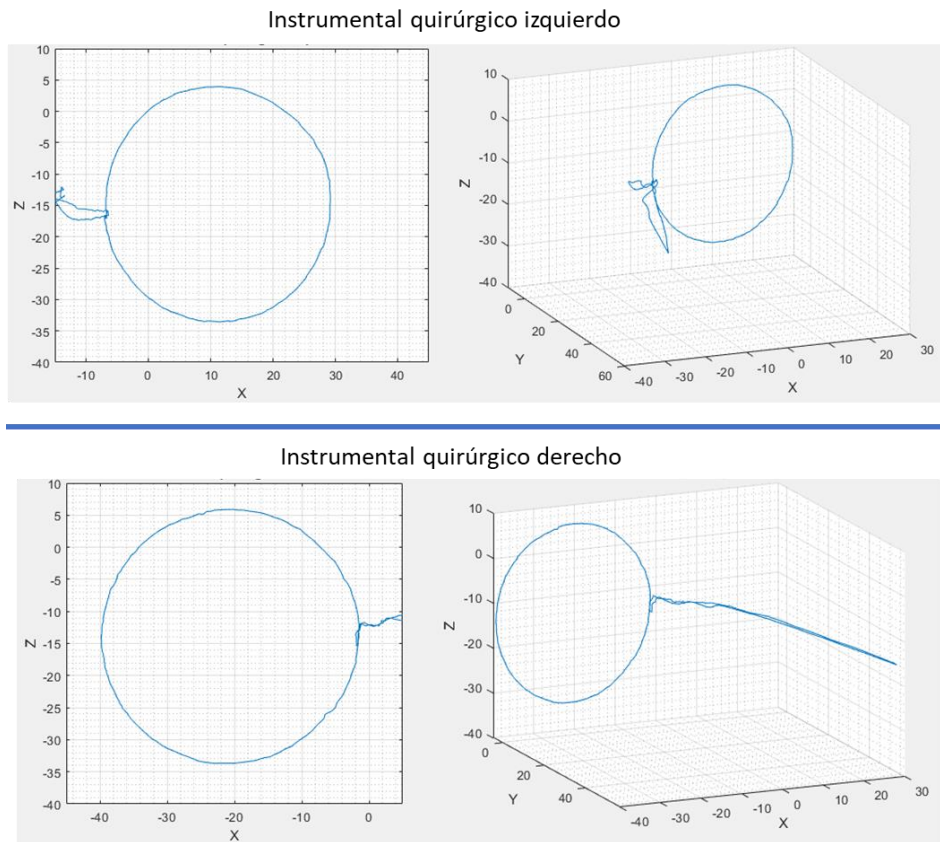


Figura 84. Resultados de las pruebas de movimiento del instrumental quirúrgico, con una trayectoria circular.

Como se observa en la Figura 84, el instrumental derecho se registra un radio de 37.6 [mm], mientras que con el instrumental quirúrgico izquierdo se registra un diámetro de 37.4 [mm].

La siguiente trayectoria que siguieron las puntas del instrumental quirúrgico fue una línea recta de 40 [mm], Figura 85. Las medidas se tomaron primero con el instrumento izquierdo y sucesivamente con el instrumento derecho, para comprobar que con ambos instrumentos se registrara la misma línea recta.

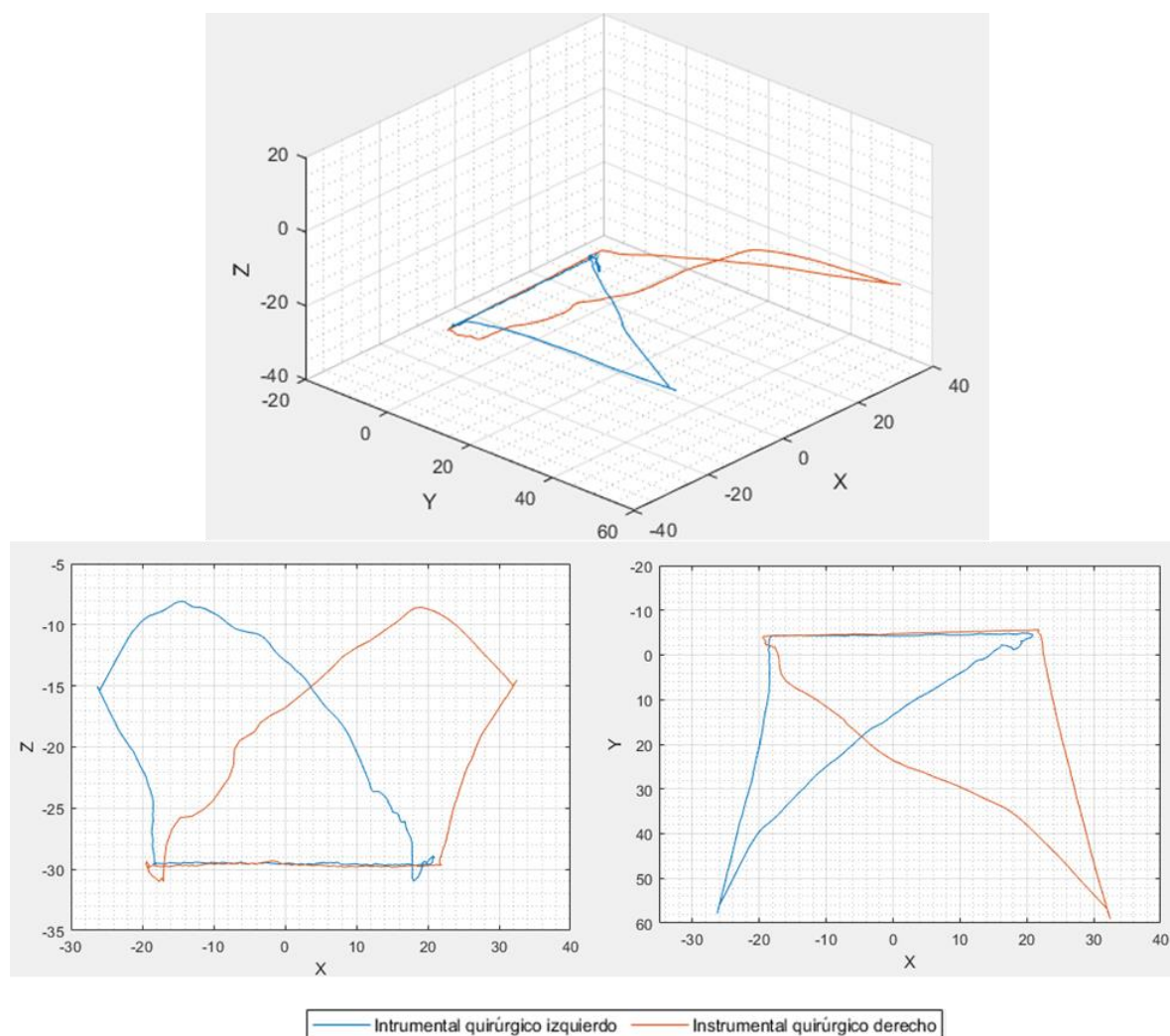


Figura 85. Resultados de las pruebas de movimiento del instrumental quirúrgico, con una trayectoria recta.

Con el instrumental quirúrgico izquierdo se obtiene una línea de 39.8 [mm], y con el instrumental derecho una línea de 40.3 [mm].

Las mediciones mostradas en las figuras 81, 82, 84 y 85 no muestran errores sistemáticos.

6.4 Resultados de la aplicación de redes neuronales para la clasificación del tipo de usuario

La red neuronal *RTC* y *perceptrón* logran clasificar adecuadamente a los usuarios de un simulador de mínima invasión.

Los resultados de la red *RTC* son los siguientes:

1. Se logra el entrenamiento de la red, que clasifica correctamente a los 6 usuarios de cada grupo.
2. En la etapa de reconocimiento, se logra la clasificación correcta de 2 usuarios del grupo principiante, 3 usuarios del grupo intermedio y 3 usuarios del grupo experto, es decir, se clasifica correctamente a 8 de los 9 usuarios utilizados para el reconocimiento.
3. Al reconocer 8 de 9 usuarios se puede estimar que el clasificador tiene una eficiencia aproximada de 88.8%.

Los resultados de la red *perceptrón* son los siguientes:

1. Se logra el entrenamiento de la red, que clasifica correctamente a los 6 usuarios de cada grupo.
2. En la etapa de reconocimiento, se logra la clasificación correcta de 3 usuarios del grupo principiante, 2 usuarios del grupo intermedio y 2 usuarios del grupo experto, es decir, se clasifica correctamente a 7 de los 9 usuarios utilizados para el reconocimiento.
3. Al reconocer 7 de 9 usuarios se puede estimar que el clasificador tiene una eficiencia aproximada de 77.7%.

Las redes neuronales *RTC* y *perceptrón* tienen una sola capa entrenable, la diferencia principal entre estas redes es que el clasificador *RTC* implementa capas no entrenables que hacen transformaciones no lineales de parámetros de entrada espaciales, a parámetros espaciales binarios, teniendo muy altas dimensiones.

La programación de la red neuronal *RTC* requiere más líneas de código de programación, ya que tiene más capas que la red *perceptrón*.

La red *RTC* resultó ser mejor en la etapa de reconocimiento que la red *perceptrón*, sin embargo, las dos redes pueden mejorarse. La primera al estudiar la relación entre las variables *MAP's* y las capas no entrenables. La segunda al aumentar el número de capas.

Para determinar adecuadamente la eficiencia de las redes *RTC* y *perceptrón*, se requiere de una base de datos con más usuarios.

A partir de los resultados de las redes neuronales *RTC* y *perceptrón*, se puede determinar que es posible aplicar este tipo de clasificadores para evaluar el tipo de usuario que utiliza simuladores de cirugías de mínima invasión, a partir de los *MAP*'s calculados con los movimientos del instrumental quirúrgico.

CAPÍTULO 7 CONCLUSIONES

7.1 Conclusiones

Se diseñó, construyó y probó un arreglo instrumental con diferentes sensores que miden los movimientos de un modelo físico de rodilla y del instrumental quirúrgico utilizado durante cirugías artroscópicas. Los movimientos se simularon en una interfaz de realidad virtual, que permitirá a los cirujanos de ortopedia desarrollar destrezas técnicas en cirugías de mínima invasión de rodilla.

El arreglo instrumental permitió medir cuatro grados de libertad para el movimiento del modelo de rodilla y cuatro grados para cada instrumental quirúrgico insertado en los puertos de entrada, estos movimientos fueron simulados en una interfaz de realidad virtual.

El arreglo instrumental se conformó principalmente de 10 sensores de efecto hall *AS5600*, 2 sensores inerciales *MPU9250*, 2 sensores reflectivos *QRE1113*, un *PIC18F4550*, que son componentes comerciales y de bajo costo.

Se implementó un *filtro complementario* con una frecuencia de corte de 0.2 [Hz] para reducir el ruido de las señales del sensor *MPU9250-1* con un tiempo de estabilización inicial aceptable. La frecuencia de corte se encontró experimentalmente para lograr reducir el ruido de alta frecuencia de los acelerómetros y magnetómetros, y el ruido de baja frecuencia de los giróscopos, el tiempo de estabilización inicial del filtro fue de 4 [s].

Los sensores fueron integrados al arreglo instrumental con piezas fabricadas en una impresora 3D con filamento *PLA*, lo que facilitó el diseño de las mismas. También, se utilizaron materiales no magnéticos como el latón, fibra de carbono y acero inoxidable para los componentes mecánicos, de esta manera se redujo el ruido en los sensores magnéticos.

Se establecieron los rangos de cada sensor, a partir de las restricciones del mecanismo donde se integraron. Se definieron las resoluciones de los sensores a partir de sus hojas de datos. Se estudió la precisión de cada variable, con la desviación estándar del ruido de sus señales, con la articulación y los instrumentales quirúrgicos inmóviles.

La precisión de las posiciones de las puntas del instrumental quirúrgico fue de ± 0.1 [mm], que se calculó con la desviación estándar de las señales del ruido, con los instrumentales inmóviles. Mientras que el error máximo observado en las señales fue de ± 0.32 [mm]. El factor principal que disminuyó la precisión y aumentó el error de las señales fue el ruido de los magnetómetros del *MPU9250-1*, este ruido se incrementa al acercarse el arreglo instrumental a aparatos eléctricos u objetos metálicos.

Se obtuvieron 25 matrices de datos por segundo, que permitieron una calidad aceptable al simular virtualmente los movimientos de la rodilla y del instrumental quirúrgico en una computadora.

Es posible implementar redes neuronales para la clasificación del tipo de usuario del arreglo instrumental a partir de sus habilidades quirúrgicas, donde las variables de entrada son las variables *MAP's* que se calculan con los datos del simulador y las variables de salida son la clasificación del tipo de usuario. Esto se comprobó a partir de dos clasificadores, el *RTC* clasificó correctamente el 88.8% de los datos y el clasificador *perceptrón* clasificó correctamente el 77.7%.

7.2 Trabajo a futuro

Para mejorar el simulador de artroscopía de rodilla se propone utilizar impresión *3D* con tecnología de estereolitografía en material de resina, de esta forma se miniaturizarían las piezas y se incrementaría la libertad de movimiento del instrumental quirúrgico. También se propone utilizar un recubrimiento flexible que simule la piel de la rodilla.

Para mejorar la precisión y disminuir el error máximo del movimiento de las puntas del instrumental quirúrgico se propone estudiar la aplicación de otros filtros, como el *filtro Kalman*.

Para mejorar el análisis de los movimientos que realizan los usuarios del simulador, se puede incrementar el número de mediciones por segundo, para ello se propone estudiar diferentes microcontroladores, u optar por alternativas como *FPGAs*.

Una parte importante para el usuario, además del realismo de la interfaz, es el seguimiento de su avance en el aprendizaje, por lo que se requiere implementar funciones en la interfaz gráfica como la creación de perfiles y monitoreo de avances de cada usuario.

Uno de los problemas que se observaron durante el aprendizaje de los estudiantes de ortopedia, en el quirófano, fue posicionar la articulación de la rodilla durante los procedimientos quirúrgicos. Se propone incrementar la articulación de la cadera para hacer los movimientos de la rodilla más realistas, esto requerirá medir al menos tres grados de libertad extras, donde podría utilizarse otro sensor *MPU9250*.

Para conocer la exactitud del instrumento, se podría implementar la metodología propuesta en el Capítulo 4.4.

Se requiere implementar una red neuronal, como el clasificador *RTC*, entrenada con una base de datos obtenida del simulador desarrollado en este trabajo. Por lo que se requiere elaborar dicha base de datos con grupos de control.

REFERENCIAS

- [1] G. J. Maria Tuijthof, M. v. Sterkenburg, I. Sierevelt y N. C. Dijk, «First validation of the PASSPORT training environment for arthroscopic skills,» *Knee Surgery Sports Traumatology Arthroscopy*, vol. 18, p. 2018, 2010.
- [2] J. M. Fernández Arroyo, C. J. Potti, A. C. Illiana, P. L. Pérez, d. B. A. Rodriguez Martinez y S. Bayona Beriso, «Entrenador-Simulador Avanzado para el aprendizaje y entrenamiento de la Cirugía Artroscópica insightArthroVR®,» *Trauma*, vol. 19, p. 54, 2008.
- [3] J. M. Fernández Arroyo, C. J. Potti, A. C. Illiana, P. L. Pérez, d. B. A. Rodriguez Martinez y S. Bayona Beriso, «Entrenador-Simulador Avanzado para el aprendizaje y entrenamiento de la,» *Trauma*, vol. 19, p. 54, 2008.
- [4] J. M. Fernández Arroyo, C. J. Potti, A. C. Illiana, P. L. Pérez, d. B. A. Rodriguez Martinez y S. Bayona Beriso, «Entrenador-Simulador Avanzado para el aprendizaje y entrenamiento de la Cirugía Artroscópica insightArthroVR®,» *Taruma*, vol. 19, p. 55, 2008.
- [5] J. A. Acosta Morón y E. A. Muñoz Vargas, «SIMULACIÓN VIRTUAL CON TECNOLOGÍA HÁPTICA: ENTRENAMIENTO PARA LA CIRUGÍA DE RODILLA,» *Repertorio de Medicina y Cirugía*, vol. 20, pp. 245-250, 2011.
- [6] N. Vaughan, D. Venketesh, T. Wainwright y R. Middleton, «A Review of Virtual Reality Based Training Simulators for Orthopaedic Surgery,» *Medical Engineering & Physics*, vol. 38, nº 2, pp. 59-71, 2015.
- [7] S. Fusentece, S. Rahm, K. Wieser, J. Spillmann, M. Harders y P. Koch, «Evaluation of a virtual-reality-based simulator using passive haptic feedback for knee arthroscopy,» *Knee Surg Sports Traumatol Arthrosc*, vol. 23, nº 4, pp. 1077-1085, 2014.
- [8] W. Dilworth Canno, D. G. Eckhof, W. E. Garrett, R. E. Hunter y H. J. Sweeney, «Report of a Group Developing a Virtual Reality Simulator for Arthroscopic Surgery of the Knee Joint,» *CLINICAL ORTHOPAEDICS AND RELATED RESEARCH*, vol. 442, pp. 21-29, 2006.
- [9] D. Cannon, D. Eckhoff, W. Garrett, R. Hunter y H. Sweeney, «Passive haptics in a Knee arthroscopy simulator: is it valid for core skills training?,» *CLINICAL ORTHOPAEDICS AND RELATED RESEARCH*, vol. 422, pp. 13-20, 2006.
- [10] B. Schout, A. Hendrikx, F. Scheele, B. Bemelmans y A. Scherpbier, «Validation and implementation of surgical simulators: a critical review of present, past, and future,» *Surg Endosc*, vol. 24, nº 3, pp. 536-546, 2010.

- [11] B. S. Kitson J., «A model for developing psychomotor skills in arthroscopic knot tying,» *Ann R Coll Surg Engl*, vol. 88, nº 5, pp. 501-502, 2006.
- [12] C. Andersen, T. Winding y M. Vesterby, «Development of simulated arthroscopic skills,» *Acta Orthop*, vol. 82, nº 1, pp. 90-95, 2011.
- [13] B. Schout, A. Hendriks, F. Scheele, B. Bemelmans y A. Scherpbier, «Validation and implementation of surgical simulators:a critical review of present, past, and future,» *Surg Endosc*, vol. 24, p. 536, 2010.
- [14] S. Yamaguchi, D. Yoshida, H. Kenmotsu, T. Yasunaga, K. Konishi, S. Leiri, H. Nakashima, K. Tanoue and M. Hashizume, "Objective assessment of laparoscopic suturing skills using a motion-tracking system," *Surg Endosc*, pp. 771-775, 2011.
- [15] J. Rosen, J. Brown, M. Barreca, L. Chang, B. Hannaford y M. Sinanan, «The Blue Dragon- a system for monitoring the kinematics and the dynamics of the endoscopic tools in minimally invasive surgery for objective laparoscopic skill assessment,» *Stud Health*, vol. 85, pp. 412-418, 2002.
- [16] H. Egi, M. Okajima, M. Yoshimitsu, S. Ikeda, Y. Miyata, H. Masugami, T. Kawahara, Y. Kurita, M. Kaneko y T. Asahara, «Objective assessment of endoscopic surgical skills by analyzing directiondependent dexterity using the Hiroshima University Endoscopic Surgical Assessment Device (HUESAD),» *Surg Today*, vol. 38, pp. 705-710, 2008.
- [17] M. Chmarra, N. Bakker, C. Grimbergen y J. Dankelman, «TrEndo, a device for tracking minimally invasive surgical instruments in training setups,» *Sens Actuators A Phys*, vol. 126, pp. 328-334, 2006.
- [18] C. Arroyo Berezowsky, «Desarrollo de plan de entrenamiento con simulación de artroscopía para residentes de ortopedia,» *Acta Ortopédica Mexicana*, vol. 32, pp. 297-302, 2018.
- [19] E. Stirling, T. Lewis y N. Ferran, «Surgical skills simulation in trauma and,» *Journal of Orthopaedic Surgery and Research*, vol. 9, pp. 1-9, 2014.
- [20] G. J. Maria Tuijthof, M. v. Sterkenburg, I. Sierevelt y N. C. Dijk, «First validation of the PASSPORT training environment for arthroscopic skills,» *Knee Surg Sports Traumatol Arthrosc*, vol. 18, p. 219, 2010.
- [21] J. Ward, D. Wills, K. Sherman y A. Mohsen, «The development of an arthroscopic surgical simulator with haptic feedback,» *Future Generation Computer Systems*, vol. 14, pp. 243-251, 1998.

- [22] S. Goyal, M. Abdel, I. Karam-allah y H. Galal, «Arthroscopic skills assessment and use of box model for training in arthroscopic surgery using Sawbones – “FAST” workstation,» *EDP Sciences*, vol. 2, p. 37, 2016.
- [23] J. M. Fernández Arroyo, C. J. Potti, A. C. Illiana, P. L. Pérez, d. B. A. Rodriguez Martinez y S. Bayona Beriso, «Entrenador-Simulador Avanzado para el aprendizaje y entrenamiento de la Cirugía Artroscópica insightArthroVR,» *Trauma Fund MAPFRE*, vol. 19, nº 1, pp. 50-58, 2008.
- [24] I. 3D SYSTEMS, «3D SYSTEMS,» Touch, 2020. [En línea]. Available: <https://es.3dsystems.com/haptics-devices/touch>. [Último acceso: September 2020].
- [25] G. O. Alonso Cuellar, L. N. Cogua Cogua, F. J. Camacho García y M. Cortés Barré, «Desarrollo de un simulador de bajo costo para la adquisición de destrezas básicas en cirugía artroscópica,» *Revista de la Asociación Argentina de Ortopedia Traumatología*, vol. 79, pp. 107,108, 2015.
- [26] T. o. L. T. Inc., «ArthroSim Arthroscopy Simulator,» TolTech, 2020. [En línea]. Available: <https://www.toltech.net/medical-simulators/products/arthrosim-arthroscopy-simulator>.
- [27] V. AG, «VirtaMed ArthroS™,» VirtaMed, 2020. [En línea]. Available: <https://www.virtamed.com/es/simuladores-capacitacion-medica/arthros/>.
- [28] S. U. Corporation, «ARTHRO MENTOR™,» 3D SYSTEMS, 2017. [En línea]. Available: <https://symbionix.com/simulators/arthro-mentor/>.
- [29] I. 3D Systems, «Touch X,» 3D Systems, 2020. [En línea]. Available: <https://es.3dsystems.com/haptics-devices/touch-x>.
- [30] G. J. Maria Tuijthof, M. v. Sterkenburg, I. Sierevelt y N. C. Dijk, «First validation of the PASSPORT training environment for arthroscopic skills,» *Knee Surgery Sports Traumatology Arthroscopy*, vol. 18, pp. 218-224, 2010.
- [31] Simendo, «Simendo arthroscopy,» SIMENDO, 2020. [En línea]. Available: <https://www.simendo.eu/products/simendo-arthroscopy/>.
- [32] A. Zivanovic , E. Dibble y B. Davies, «A high force haptic system for knee arthroscopy training,» *International*, vol. 3, nº 4, pp. 429-437, 2006.
- [33] N. D. Inc, «Ascension Technology Corporation,» Ascension Technology Corporation, 2020. [En línea]. Available: <https://www.ndigital.com/about/ascension-technology-corporation/>.
- [34] J. Fernández, G. Espadero , B. Berilo, M. Garcia y P. Toharia, «Proyecto SARA (Simulador Artroscópico Avanzado). Simuladores en la formación y docencia de la cirugía artroscópica de hombro mediante técnicas de realidad virtual,» *Span J Res*, vol. 8, pp. 13-19, 2005.

- [35] Immersion, «Immersion Medical Introduces New Laparoscopic Procedural Skill Module,» Immersion, 2020. [En línea]. Available: <https://ir.immersion.com/news-releases/news-release-details/immersion-medical-introduces-new-laparoscopic-procedural-skill>.
- [36] I. Alejandro, P. Cuervo, J. Fernández, P. Pérez, G. Espadero, B. Beriso y et al., «Entrenador-simulador avanzado para el entrenamiento de la artroscopia Insight-Mist.,» *Span J Res*, vol. 7, p. 217, 2004.
- [37] F. Fernandez, P. Pérez, G. Espadero, B. Beriso y T. Barros, «Aplicaciones de técnicas de realidad virtual en el aprendizaje artroscópico. Introducción, posibilidades y actualización,» *Cuadernos de artroscopía*, vol. 11, pp. 8-13, 2004.
- [38] H. Pheng Ann, C. Chun-Yiu, W. Tien-Tsin y X. Yangsheng, «A VirtualReality training system for knee arthroscopy surgery,» *IEEE*, vol. 8, nº 2, pp. 217-227, 2004.
- [39] L. Moody, J. Arthur, A. Zivanovic y A. Waterworth, «A parttask approach to haptic knee arthroscopy training,» *Stud Health Technol Inform*, vol. 94, pp. 216-8, 2003.
- [40] R. Poss, J. Mabrey, S. Gillogly, J. Kasser, H. Sweeney, B. Zarins, W. Garrett y W. Cannon, «Development of a virtual reality arthroscopic knee simulator,» *Stud Health Technol Inform*, vol. 70, pp. 192-194, 2000.
- [41] I. Logan, D. Wills, N. Avis, A. Mohsen y K. Sherman, «Virtual environment knee arthroscopy,» *Society for Computer Simulation, Simulation Series*, vol. 28, pp. 17-22, 1996.
- [42] A. ZIVANOVIC, E. DIBBLE y B. DAVIES, «A high force haptic system for knee arthroscopy training,» *World Scientific*, vol. 3, pp. 429-437, 2005.
- [43] IMMERSION, «Phantom Omni,» IMMERSION, 2020. [En línea]. Available: <http://www.immersion.fr/en/phantom-touch/>. [Último acceso: 2020].
- [44] O. Herrera, R. Espitia, G. Zeyel, P. Figueroa, J. Torres y A. García , «Desarrollo de un sistema de entrenamiento médico para artroscopia de rodilla,» *Revista Colombiana de Ortopedia y Traumatología*, vol. 22, p. 176, 2008.
- [45] J. Fernandez, P. Pérez, G. Espadero , B. Beriso y G. Triviño, «Aplicaciones de técnicas de realidad virtual en el aprendizaje artroscópico. Introducción, posibilidades y actualización,» *Cuadernos de Artroscopia*, vol. 11, pp. 8-13, 2004.
- [46] 3. SYSTEMS, «SIMBIONIX,» 3D SYSTEMS, 2017. [En línea]. Available: <https://symbionix.com/simulators/arthro-mentor/>.
- [47] J. Lee, E. Park y S. Robinovitch, «Estimation of Attitude and External Acceleration,» *IEEE Transactions on instrumentation and measurement*, vol. 61, nº 8, p. 2263, 2012.

- [48] H. Walter, «A Comparison of Complementary and Kalman Filtering,» *Aerospace and Electronic Systems*, vol. 3, p. 321, 1975.
- [49] D. Gaydou, J. Redolfi y A. Henze, «Filtro complementario para estimacion de actitud aplicado al controlador embebido de un cuatrirrotor,» de *Congreso Argentino de Sistemas Embebidos*, Buenos Aires, Argentina, 2011.
- [50] A. Baerveldt y R. Klang, «A Low-cost and Low-weight Attitude Estimation System for an Autonomous Helicopter,» de *Proceedings of IEEE International Conference on Intelligent Engineering Systems*, Budapest, Hungary, 1997.
- [51] R. Rojas, «Neural Networks a Systematic Introduction,» *Springer Verlag*, Berlín, 1996.
- [52] D. CALDERÓN, «Investigación de estructuras de ensamble en redes neuronales,» Tesis de Maestría UNAM, México, 2011.
- [53] P. Larrañaga, I. Inza y A. Moujahid, «Redes Neuronales,» Universidad del País Vasco–Euskal Herriko Unibertsitatea.
- [54] T. Baydyk y E. Kussul, *Redes Neuronales, visión computacional y micromecánica*, México: Itaca, 2009, p. 158.
- [55] E. Kussul, T. Baydyk y D. Wunsch, *Neural Networks and Micro Mechanics*, Springer Verlag, 2010, p. 210.
- [56] T. Baydyk, E. Kussul y D. Wunsch, *Intelligent Automation in Renewable Energy*, Springer, 2019, p. 300.
- [57] S. Electronics, «SparkFun IMU Breakout - MPU-9250,» 2020. [En línea]. Available: <https://www.sparkfun.com/products/retired/13762>.
- [58] ams, «AS5600 Position Sensor,» 2020. [En línea]. Available: <https://ams.com/as5600>.
- [59] O. Semiconductor, «QRE1113,» ON Semiconductor, 2021. [En línea]. Available: <https://www.onsemi.com/products/optoelectronics/infrared/reflective-sensor/qre1113>.
- [60] Microchip, «PIC18F4550,» Microchip, 2021. [En línea]. Available: <https://www.microchip.com/wwwproducts/en/PIC18F4550>.
- [61] T. Instruments, «TCA9548A,» Texas Instrument, 2021. [En línea]. Available: <https://www.ti.com/product/TCA9548A?keyMatch=TCA9548A&tisearch=search-everything>.
- [62] Texas Instruments, «I2C Bus Pullup Resistor Calculation,» Application report SLVA689 , 2015.

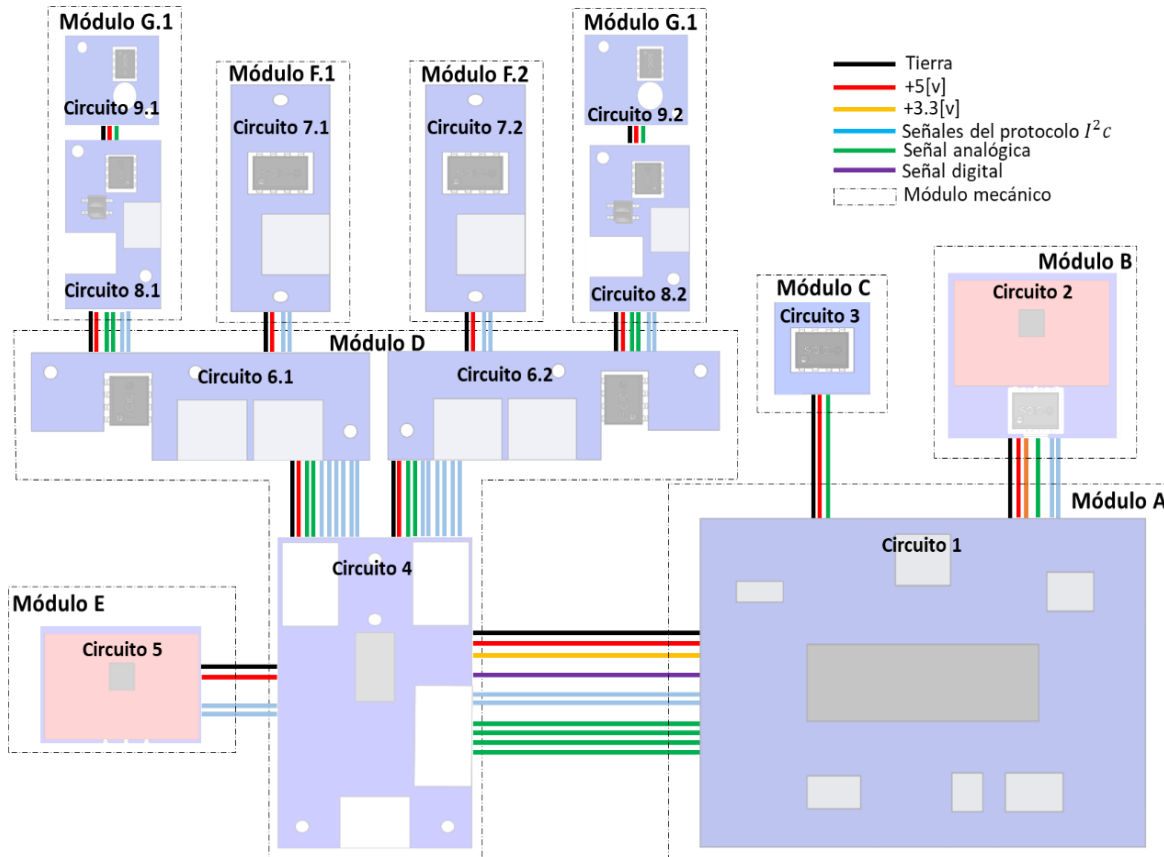
- [63] T. Instruments, «AMS1117 3.3,» Texas Instruments, 2021. [En línea]. Available: <https://www.ti.com/store/ti/en/p/product/?p=LM1117T-3.3/NOPB&keyMatch=AMS1117-3%203&tisearch=search-everything>.
- [64] 3. Systems, «Sense Scanner,» 3D Systems, 2020. [En línea]. Available: https://support.3dsystems.com/s/article/Sense-Scanner?language=en_US.
- [65] D. Skoog, J. Holler y S. Crouch, Principles of Instrumental analysis, 7 ed., Boston USA: CENGAGE Learning, 2018, p. 17.
- [66] F. Pérez, I. Oropeza, P. Sánchez, J. Tapia, J. Ruiz y A. Minor, «Orthogonal cameras system for tracking of laparoscopic instruments in training environments,» *Cir Cir*, vol. 6, pp. 548-555, 2018.
- [67] F. Pérez Escamirosa, R. Ordorica Flores, I. Oropesa, C. R. Zalles Vidal y A. Minor, «Face, content, and construct validity of the EndoViS training system for objective assessment of psychomotor skills of laparoscopic surgeons,» *Surgical Endoscopy*, 2014.
- [68] I. Oropesa, P. Sánchez Gonzáles, M. Chmarra, P. Lamata, R. Pérez Rodríguez, F. Willem Jansen, J. Dankelman y E. J. Gómez Aguilera, «Supervised classification of psychomotor competence in minimally invasive surgery based on instruments motion analysis,» *Surgical Endoscopy*, pp. 657-670, 2013.
- [69] H. M. Pedraza y M. L. Stetten, « Arthroscopy education,» *Orthopedics*, 1987.
- [70] O. Herrera, R. Espitia, G. Zeyel, P. Figueroa, J. Torres y A. García , «Desarrollo de un sistema de entrenamiento médico para artroscopia de rodilla,» *Revista Colombiana de Ortopedia y Traumatología*, vol. 22, p. 176, 2008.
- [71] A. Jarrillo Silva, O. A. Dominguez, Ramirez, V. Parra Vega y P. Ordaz Oliver, «PHANToM OMNI Haptic Device: Kinematic and Manipulability,» *IEEE*, vol. 55, pp. 193-198, 2009.
- [72] I. C. London, «Knee arthroscopy training system,» Imperial COLlege London, [En línea]. Available: <https://www.imperial.ac.uk/mechatronics-in-medicine/research/knee-arthroscopy/>.
- [73] U. Digital, «E4T MINIATURE OPTICAL KIT ENCODER,» 2020. [En línea]. Available: <https://www.usdigital.com/products/encoders/incremental/kit/E4T>.
- [74] O. Corporation, «E6B2-CWZ6C 200P/R 2M,» 2020. [En línea]. Available: <https://industrial.omron.es/es/products/E6B2-CWZ6C-200P-R-2M>.
- [75] G. W. y G. Bishop, «An Introduction to the Kalman Filter,» *Department of Computer Science, University of North Carolina at Chapel Hill*, pp. 27599-3175, 2006.

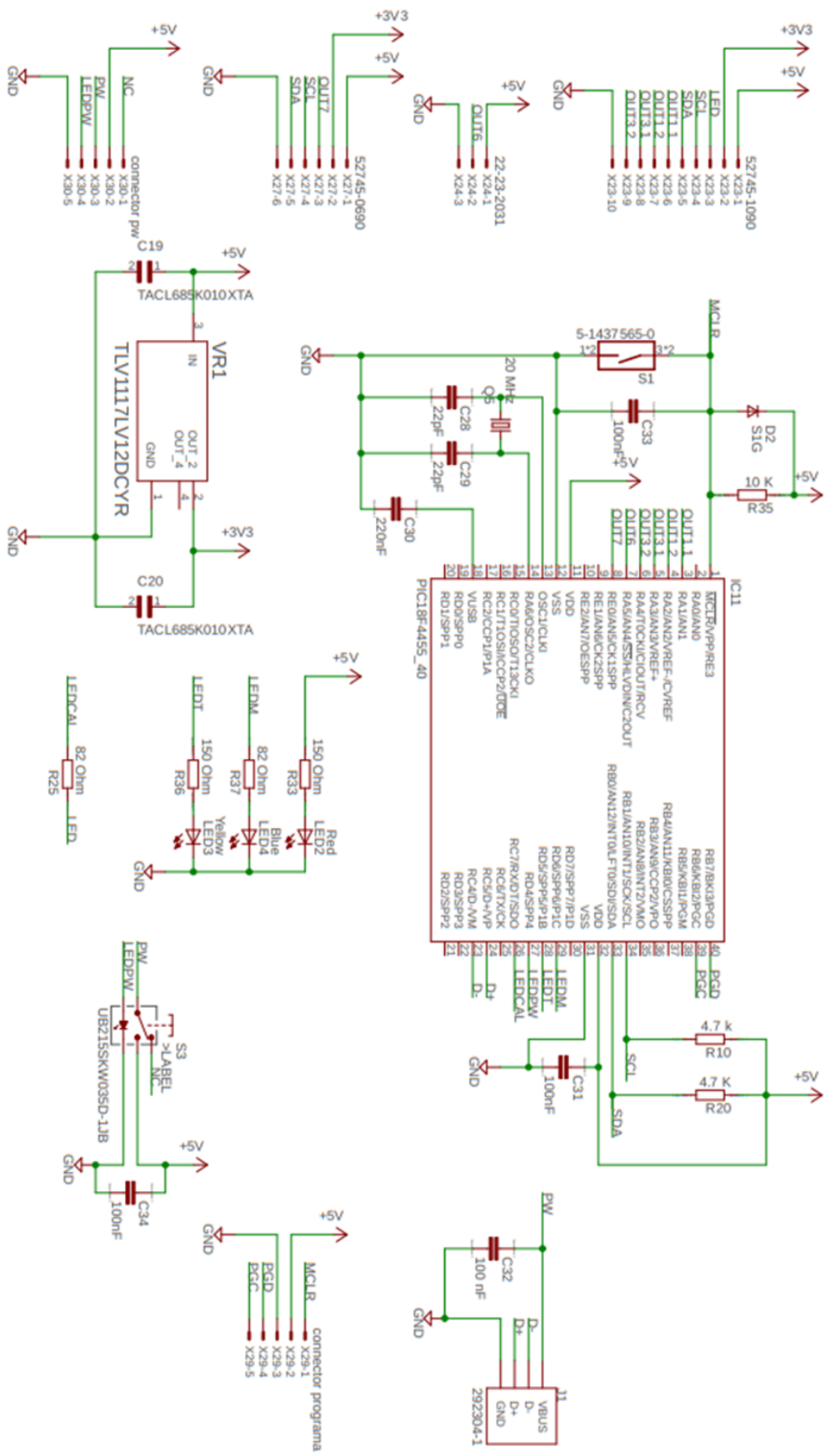
- [76] K. Sherman, J. Ward, D. Wills, V. Sherman y A. Mohsen, «Surgical trainee assessment using a VE knee arthroscopy training system (VE-KATS),» *Stud Health Technol Inform*, vol. 81, pp. 465-470, 2001.
- [77] G. Tuigthof, P. Heeman, C. Van Dijk y L. Blankevoort, «Physical simulation environment for arthroscopic joint irrigation,» *J Med Devices*, vol. 3, pp. 1-6, 2009.
- [78] H. Walter, «A Comparison of Complementary,» *IEEE TRANACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS*, Vols. %1 de %2AES-11, pp. 321-325, 1975.

ANEXOS

Anexo A. Circuitos del arreglo instrumental

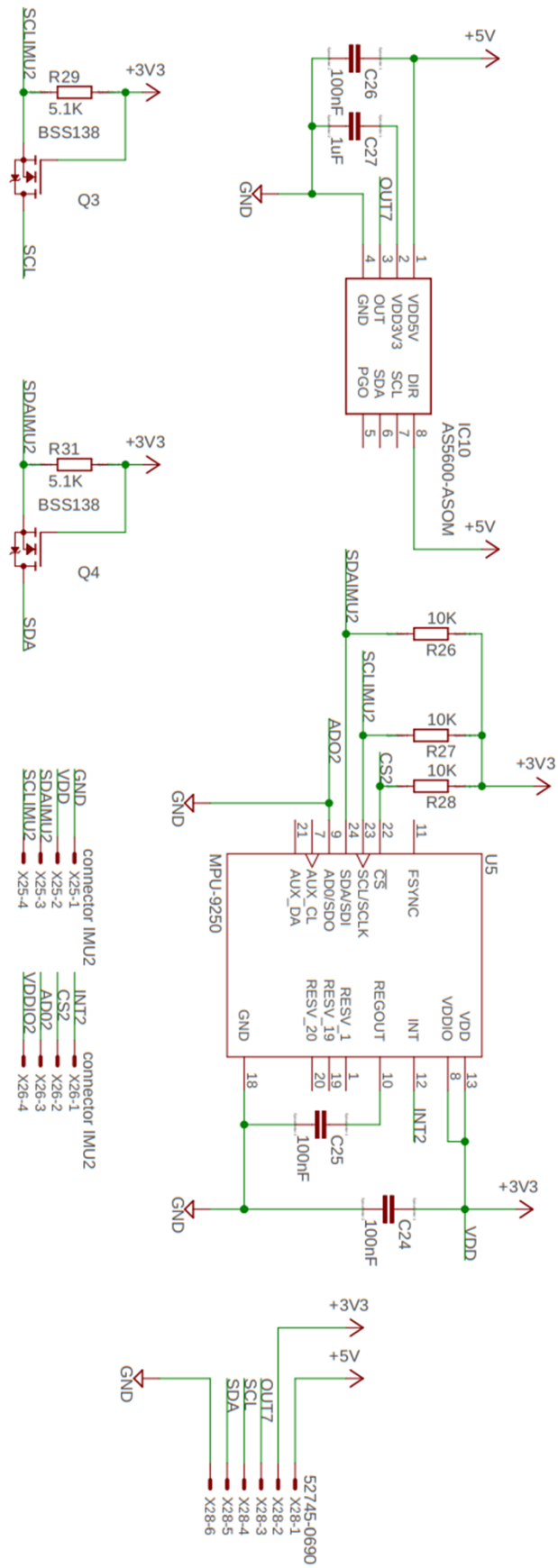
Los diferentes circuitos eléctricos se muestran en la siguiente figura:



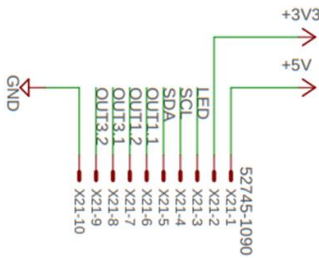
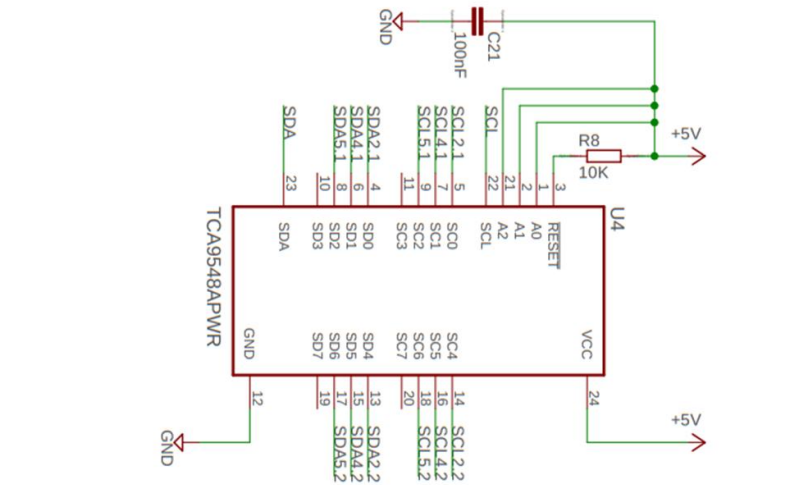
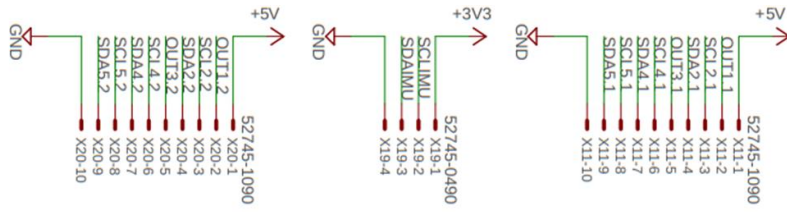
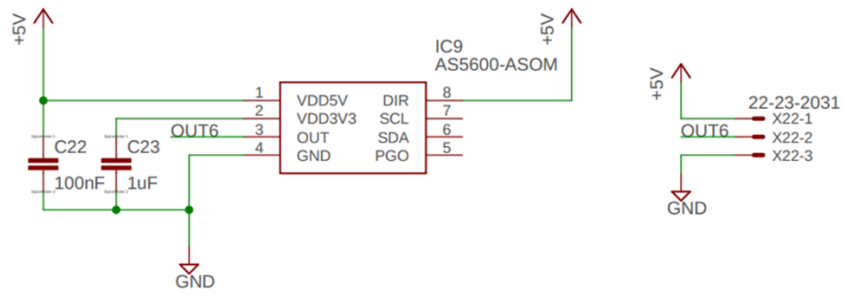


Circuito 1

Circuito 2

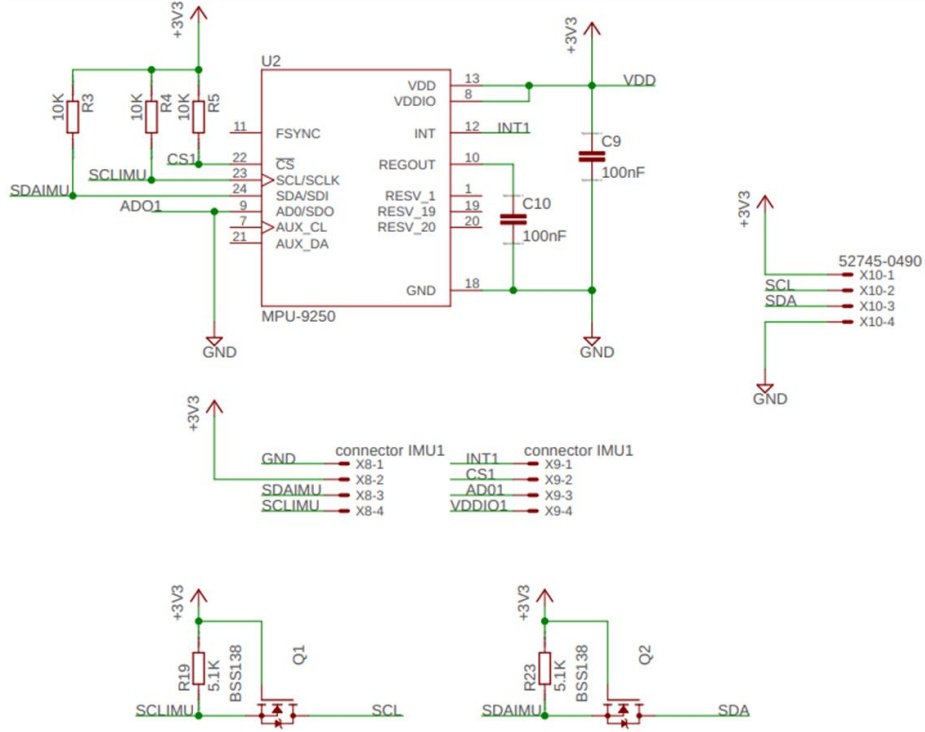


Circuito 3

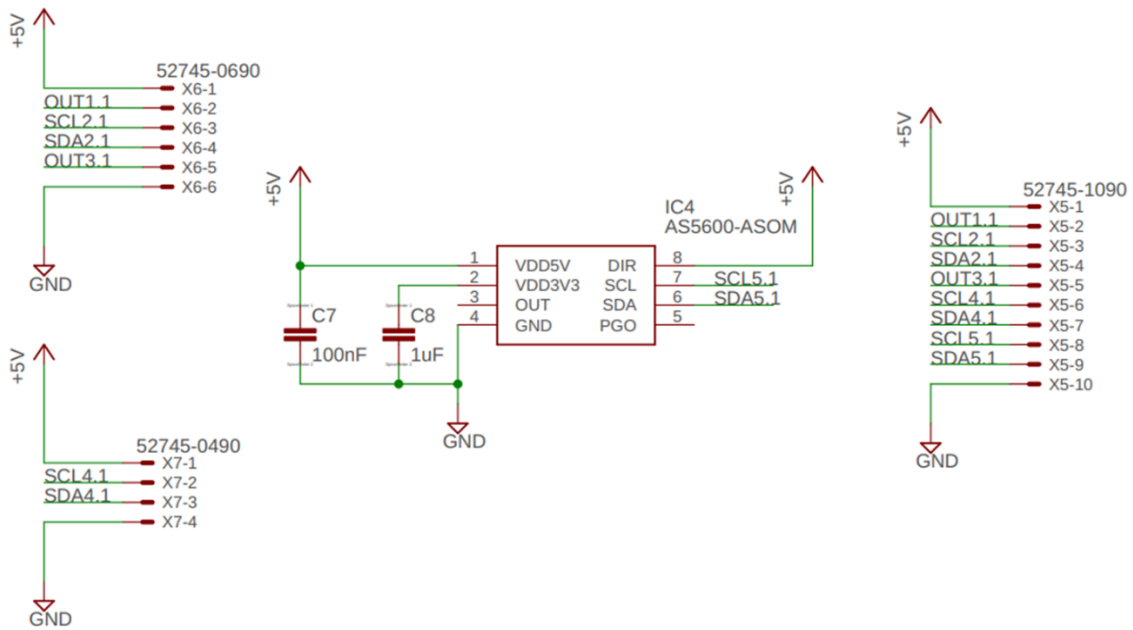


Circuito 4

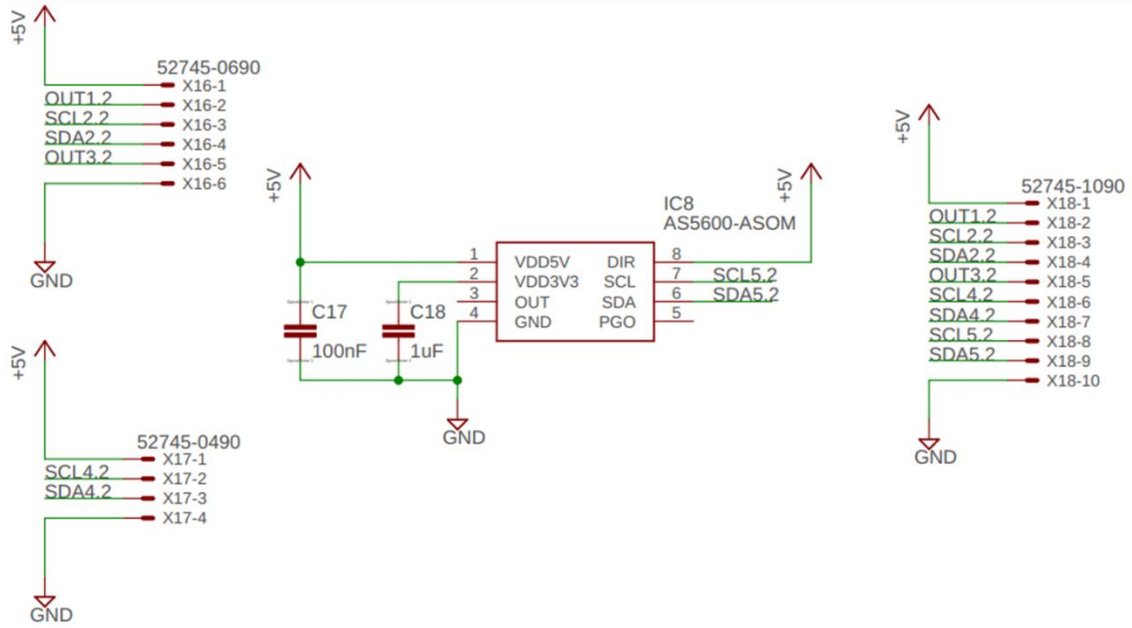
Circuito 5



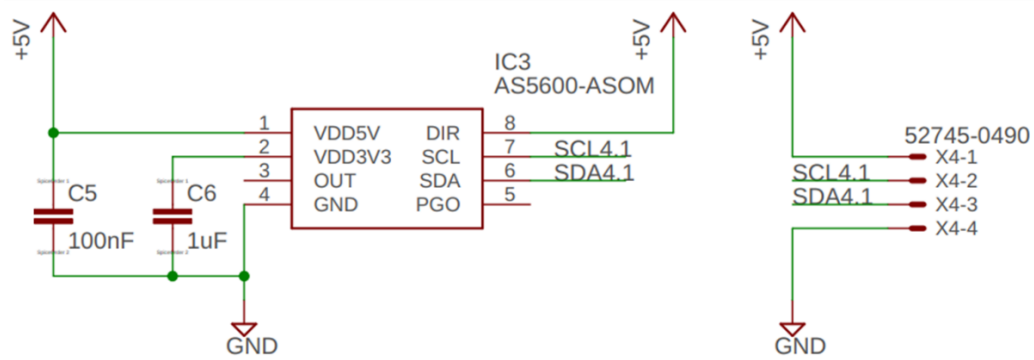
Circuito 6.1



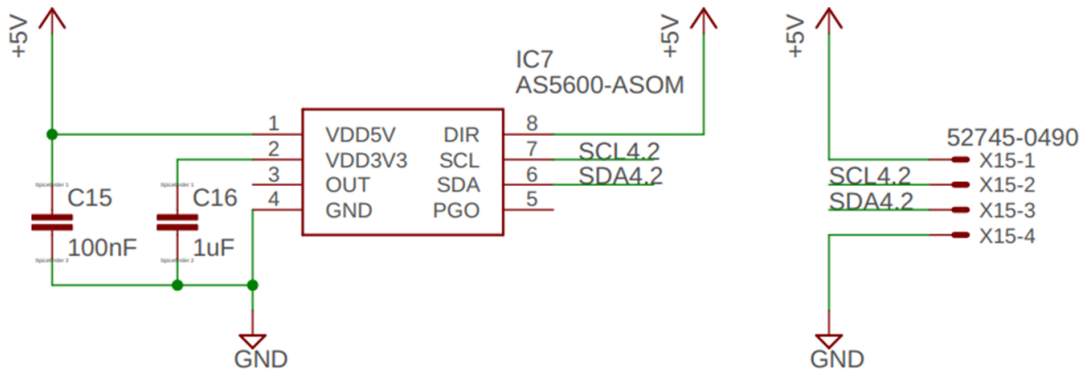
Circuito 6.2



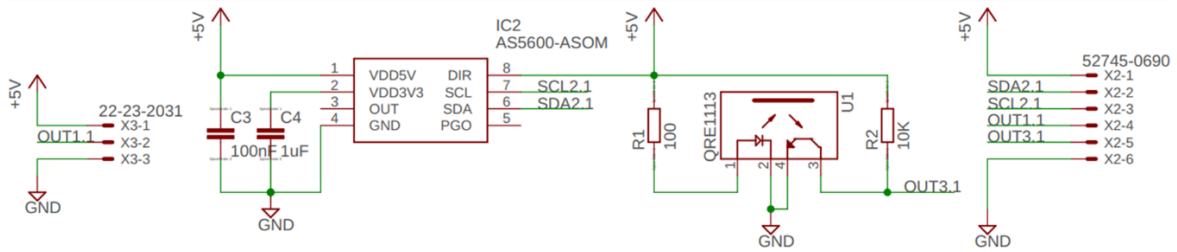
Circuito 7.1



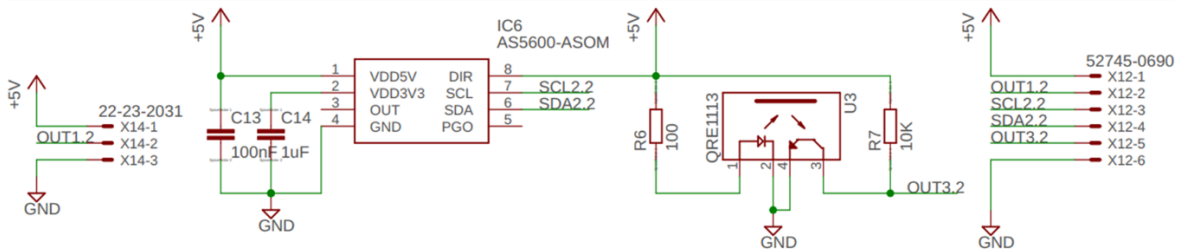
Circuito 7.2



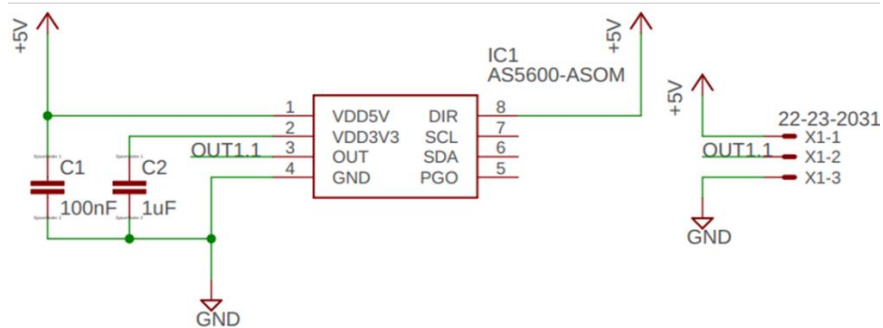
Circuito 8.1



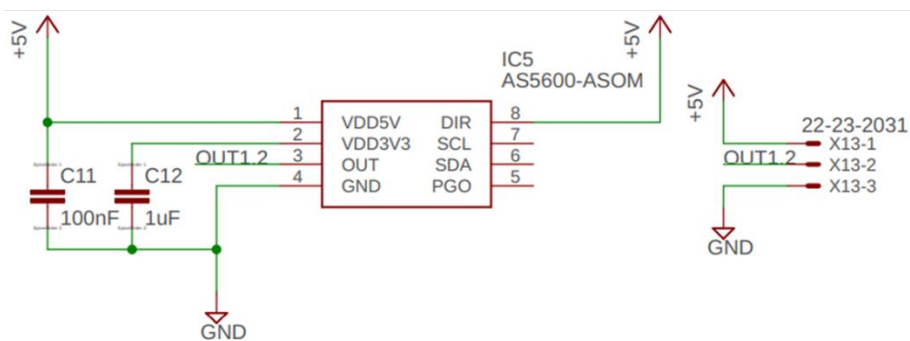
Circuito 8.2



Circuito 9.1



Circuito 9.2



Anexo B. Programa del microcontrolador PIC18F4550

```
#include <18F4550.h>
#fuses HSPLL, NOWDT, NOPROTECT, NODEBUG, USBDIV, PLL5, CPUDIV1, VREGEN
#DEVICE ADC=10;
#use delay(clock=4800000)
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#use I2C(master,sda=PIN_B0, scl=PIN_B1, force_sw,fast=450000)
#define USB_CDC_DATA_LOCAL_SIZE 128
#include <usb_cdc.h>
#include <string.h>
#define USB_CON_SENSE_PIN PIN_B2
//Inicialización de Variables
float yf_0;
float t_MPU, mxm, mym, mzm, axm, aym, azm, gxm, gym, gzm;//Datos del MPU
float To = 0;//Tiempo de muestra
int16 t = 0;//contador para interrupción
float cosang = 0;
//valores para calibración del mpu:
float axmin = -14428;
float aymin = -13617;
float azmin = -18198;
float axmax = 13889;
float aymax = 15882;
float azmax = 14442;
float mxmin = -468;
float mymin = -160;
float mzmin = -105;
float mxmax = 14;
float mymax = 400;
float mzmaz = 490;
float gxcer = 258;
float gyzer = 265;
float gzcer = -74;
float axfil = 0, ayfil = 0, azfil = 0;
float gxfil = 0, gyfil = 0, gzfil = 0;
float mxfil = 0, myfil = 0, mzfил = 0;
float A[] = { 0, 0, 0 };
float G[] = { 0, 0, 0 };
float M[] = { 0, 0, 0 };
float proyma[] = { 0, 0, 0 };
float R[3][3]={0,0,0},{0,0,0},{0,0,0}};
double roll_a, pitch_a, yaw_a;
float r_a = 0, p_a = 0, y_a = 0, r_g = 0, p_g = 0, y_g = 0;
float r_aant = 0, p_aant = 0, y_aant = 0;
float fc = 0.2; //Frecuencia de corte del filtro complementario en [Hz]
float z = 0; //Variable para la transformada z
float Wc = 2 * 3.1416 * fc;//Frecuencia de corte del filtro complementario en [rad]
//Variables para calibrar los sensores AS5600:
float sensor2_1 = 0, sensor2_2 = 0, d_1 = 0, d_2 = 0;//Dist al interior de la rodilla
float d1_inc = 0, d1_ant = 0, d2_inc = 0, d2_ant = 0;
float sensor1_1 = 0, sensor1_2 = 0; //Pitch 1 y 2
float p1_inc = 0, p1_ant = 0, p2_inc = 0, p2_ant = 0, p_1sf = 0, p_2sf = 0;
float calsensor4_1 = 269.914; //Roll 1
float calsensor5_1 = 281.6; //Yaw 1
float calsensor4_2 = 63.813; //Roll 2
float calsensor5_2 = 253.827; //Yaw 2
float ycero_0 = 0;//Calibración del yaw cero
float sensor6 = 0;
//Variables para MPU9250-2
float gxm2, gym2, gzm2;//giroscopos
float gxcer2 = -81;
float gyzer2 = 89.3;
float gzcer2 = -375;
float Ax, Ay, Az, Gx, Gy, Gz;
float gir_artin = 0;
float pitch_artA = 0;
float gir_lente = 0, gir_lentef = 0, gir_im = 0, ap_pin = 0;//Giros del artroscopio
```

```

int16 v = 0; //Variable para estabilizar la medición del MPU9250-2
//Variables para detectar los instrumentos insertados
int ins_iz = 0, ins_der = 0; //Instrumento insertado a la rodilla
float puerto_iz = 0, puerto_der = 0; //Valor del sensor QRE1113
float d_0; //Movimiento de la rodilla en el eje Z
int o = 0; //Variable para estabilizar el filtro complementario
int i = 0, j = 0, k = 0, l = 0, p = 0, q = 0, s = 0, u = 0;
//Variables para enviar la información
float r_0 = 0, p_0 = 0, y_0 = 0, r_1 = 0, p_1 = 0, y_1 = 0, r_2 = 0, p_2 = 0, y_2 = 0;
char msg[100];
//Interrupción para el temporizador:
#INT_TIMER1
void TIMER1_isr(void) //interrupción temporizador
{
    t = t + 1;
    set_timer1(53536); //Precarga
}
//Función para cambiar de canal del bus I2c:
void TCA9548A(unsigned int8 bus)
{
    i2c_start();
    i2c_write(0xEE); // Dirección 0x77 + bit escritura 0
    i2c_write(1 << bus);
    i2c_stop();
}
//Función para medir el sensor AS5600 seleccionado:
int16 MedirAS5600()
{
    int16 val = 0;
    long msb;
    long lsb;
    i2c_start();

    i2c_write(0x6C); // Dirección 0x36 + bit escritura 0
    i2c_write(0x0C); //Solicitar los bits más significativos
    i2c_stop();
    i2c_start();
    i2c_write(0x6D); // Dirección 0x36 + bit lectura
    msb = i2c_read(0); //bits más significativos
    i2c_stop();
    msb = msb << 8;
    i2c_start();
    i2c_write(0x6C); // Dirección 0x36 + bit escritura 0
    i2c_write(0x0D); //Solicitar los bits menos significativos
    i2c_stop();
    i2c_start();
    i2c_write(0x6D); // Dirección 0x36 + bit lectura
    lsb = i2c_read(0); // bits menos significativos
    i2c_stop();
    val = msb | lsb; //Medición del sensor
    return val;
}
//Función para aplicar un filtro paso bajas:
float Filtro_pasobajas(float valor_medido, float valor_anterior, float Wc)
{
    float valor_filtrado = 0;
    return valor_filtrado = (valor_anterior * (1 / (1 + (To * Wc)))) + (valor_medido * ((Wc * To) / (1 + (Wc * To))));
}
//Función para medir los acelerómetros y giróscopos del MPU9250:
void medir_MPU(int16 adress_imu, float* axm, float* aym, float* azm, float* gxm, float* gym, float* gzm)
{
    signed int16 ax = 0, ay = 0, az = 0, gx = 0, gy = 0, gz = 0;
    //Lectura de acelerometro y giroscopio
    i2c_start();
    i2c_write(adress_imu); //Dirección del MPU9250 + 0C bit de escritura
    i2c_write(0x3B); //Solicitar datos del acelerometro
    i2c_start();
    i2c_write(adress_imu + 1); //Dirección del MPU9250 + bit de lectura

```

```

az = -(((unsigned int16)i2c_read() << 8) | (unsigned int16)i2c_read()); //Aceleración, eje X
ax = (((unsigned int16)i2c_read() << 8) | (unsigned int16)i2c_read()); //Aceleración, eje Y
ay = -(((unsigned int16)i2c_read() << 8) | (unsigned int16)i2c_read()); //Aceleración, eje Z
T_MPU = (((unsigned int16)i2c_read() << 8) | (unsigned int16)i2c_read()); //Temperatura
gz = (((unsigned int16)i2c_read() << 8) | (unsigned int16)i2c_read()); //Vel. angular, eje X
gx = -(((unsigned int16)i2c_read() << 8) | (unsigned int16)i2c_read()); //Vel. angular, eje Y
gy = (((unsigned int16)i2c_read() << 8) | (unsigned int16)i2c_read(0)); //Vel. angular, eje Z
i2c_stop();
*axm = (float)ax;
*aym = (float)ay;
*azm = (float)az;
*gxm = (float)gx;
*gym = (float)gy;
*gzm = (float)gz;
}
//Función para medir los magnetómetros del MPU9250:
void medir_Mag(int16 address_mag, float* mxm, float* mym, float* mzm)
{
    signed int16 mx = 0, my = 0, mz = 0;
    i2c_start();
    i2c_write(address_mag);
    i2c_write(0x03);
    i2c_stop();
    i2c_start();
    i2c_write(address_mag + 1); //se reciben 7 bits
    mx = -(((unsigned int16)i2c_read() | (unsigned int16)i2c_read() << 8);
    mz = (((unsigned int16)i2c_read() | (unsigned int16)i2c_read() << 8);
    my = -(((unsigned int16)i2c_read() | (unsigned int16)i2c_read(1) << 8);
    i2c_stop();
    *mxm = (float)mx;
    *mym = (float)my;
    *mzm = (float)mz;
}
void main()
{
    output_high(PIN_A3); //Parpadeo de led para verificar funcionamiento
    setup_adc(ADC_CLOCK_DIV_64 | ADC_TAD_MUL_2); //Configuración de puertos ADC
    setup_adc_ports(AN0); //Activar puerto ADC
    setup_timer_1(T1_INTERNAL | T1_DIV_BY_1); //Configurar timer
    set_timer1(53536); //1us overflow
    enable_interrupts(INT_TIMER1);
    enable_interrupts(GLOBAL);
    usb_cdc_init(); //Inicializar comunicación USB
    usb_init();
    //Mediciones iniciales:
    set_ADC_channel(0); p1_ant = read_ADC(); //p_1
    set_ADC_channel(4); p2_ant = read_ADC(); //p_2
    TCA9548A(0);
    d1_ant = 25.1328 * (MedirAS5600() / 4095.0); //d_1
    TCA9548A(7);
    d2_ant = -25.1328 * (MedirAS5600() / 4095.0); //d_2
    TCA9548A(3);
    //Configurar MPU9250 como lectura
    i2c_start();
    i2c_write(0xD0); //Dirección del MPU9250 + bit de escritura
    i2c_write(0x6B); //Configuración de los datos del MPU9250
    i2c_write(0x0); //Configurar como lectura
    i2c_stop();
    delay_ms(100);
    i2c_start();
    i2c_write(0xD2); //Dirección del MPU9250 + bit de escritura
    i2c_write(0x6B); //Configuración de los datos del MPU9250
    i2c_write(0x0); //Configurar como lectura
    i2c_stop();
    delay_ms(100);
    i2c_start();
    i2c_write(0xD0); //Dirección del MPU9250 + bit de escritura
    i2c_write(0x37); //Configuración de los datos del MPU9250
    i2c_write(0x02); //Configurar como lectura
    i2c_stop();
}

```



```

delay_ms(100);
i2c_start();
i2c_write(0xD2); //Dirección del MPU9250 + bit de escritura
i2c_write(0x37); //Configuración de los datos del MPU9250
i2c_write(0x02); //configurar como lectura
i2c_stop();
delay_ms(100);
// Configurar acelerometro
i2c_start();
i2c_write(0xD0); //Dirección del MPU9250 + bit de escritura
i2c_write(0x28); //Configuración del acelerómetro
i2c_write(0x18); //Escala de +-16G
i2c_stop();
delay_ms(100);
i2c_start();
i2c_write(0xD2); //Dirección del MPU9250 + bit de escritura
i2c_write(0x28); //Configuración del acelerómetro
i2c_write(0x18); //escala de +-16G
i2c_stop();
delay_ms(100);
//Configurar giroscopio
i2c_start();
i2c_write(0xD0); //Dirección del MPU9250 + bit de escritura
i2c_write(0x27); //Configuración filtro del acelerómetro
i2c_write(0x10); //escala de +-1000DPS
i2c_stop();
delay_ms(100);
i2c_start();
i2c_write(0xD2); //Dirección del MPU9250 + bit de escritura
i2c_write(0x27); //Configuración filtro del acelerómetro
i2c_write(0x10); //escala de +-1000DPS
i2c_stop();
delay_ms(100);
//Configurar magnetómetro
i2c_start();
i2c_write(0x18); //Dirección del magnetómetro 0C
i2c_write(0x0A); //Configuración de lectura o escritura
i2c_write(0x16); //Control, 16bits, modo continuo
i2c_stop();
delay_ms(100);
//Bucle
while (true)
{
    To = t / 1000.0; //Tiempo muestral
    t = 0;
    usb_task(); //Verifica la comunicación USB
    //Lectura de sensores:
    TCA9548A(4); MedirAS5600();
    TCA9548A(0); sensor2_1 = 25.1328 * (MedirAS5600() / 4095.0); //d_1
    TCA9548A(7); sensor2_2 = -25.1328 * (MedirAS5600() / 4095.0); //d_2
    TCA9548A(1); r_1 = -((360 * (MedirAS5600() / 4096.0)) - calsensor4_1); //roll_1
    TCA9548A(2); y_1 = -((360 * (MedirAS5600() / 4096.0)) - calsensor5_1); //yaw_1
    TCA9548A(6); r_2 = (360 * (MedirAS5600() / 4096.0)) - calsensor4_2; //roll_2
    TCA9548A(5); y_2 = -((360 * (MedirAS5600() / 4096.0)) - calsensor5_2); //yaw_2
    TCA9548A(3);
    medir_MPU(0xD0, &axm, &aym, &azm, &gxm, &gym, &gzx);
    medir_Mag(0x18, &mxm, &mym, &mzm);
    axfil = axm;
    ayfil = aym;
    azfil = azm;
    mxfil = mxm;
    myfil = mym;
    mzfil = mzm;
    gxfil = gxm;
    gyfil = gym;
    gzfil = gzm;
    //Calibrando los acelerómetros
    A[0] = ((2 * axfil) - axmax - axmin) / (axmax - axmin);
    A[1] = ((2 * ayfil) - aymax - aymin) / (aymax - aymin);
    A[2] = ((2 * azfil) - azmax - azmin) / (azmax - azmin);
}

```

```

//Calibrando los giroscopos
G[0] = -(gxfil + gxcer) / 131.072;
G[1] = -(gyfil + gycer) / 131.072;
G[2] = -(gzfil + gzcerc) / 131.072;
//Calibrando los magnetometros
M[0] = (2 * mxfil - mxmax - mxmin) / (mxmax - mxmin);
M[1] = (2 * myfil - mymax - mymin) / (mymax - mymin);
M[2] = (2 * mzfil - mzmax - mzmin) / (mzmax - mzmin);
//Compensación del magnetometro
cosang = ((M[0] * A[0]) + (M[1] * A[1]) + (M[2] * A[2])) / ((sqrt((M[0] * M[0]) + (M[1] * M[1] + (M[2] * M[2]))) * sqrt((A[0] * A[0]) + (A[1] * A[1]) + (A[2] * A[2]))));
proyma[0] = (sqrt((M[0] * M[0]) + (M[1] * M[1]) + (M[2] * M[2])) * cosang * A[0]) / sqrt((A[0] * A[0]) + (A[1] * A[1]) + (A[2] * A[2]));
proyma[1] = (sqrt((M[0] * M[0]) + (M[1] * M[1]) + (M[2] * M[2])) * cosang * A[1]) / sqrt((A[0] * A[0]) + (A[1] * A[1]) + (A[2] * A[2]));
proyma[2] = (sqrt((M[0] * M[0]) + (M[1] * M[1]) + (M[2] * M[2])) * cosang * A[2]) / sqrt((A[0] * A[0]) + (A[1] * A[1]) + (A[2] * A[2]));
M[0] = M[0] - proyma[0];
M[1] = M[1] - proyma[1];
M[2] = M[2] - proyma[2];
//Cálculo de matriz de rotación:
R[2][0] = -A[0];
R[2][1] = -A[1];
R[2][2] = -A[2];
R[1][0] = -M[0];
R[1][1] = -M[1];
R[1][2] = -M[2];
R[0][0] = (R[1][1] * R[2][2]) - (R[1][2] * R[2][1]);
R[0][1] = (R[1][2] * R[2][0]) - (R[1][0] * R[2][2]);
R[0][2] = (R[1][0] * R[2][1]) - (R[1][1] * R[2][0]);
//Cálculo de ángulos de euler:
if (R[2][2] != 0)
{
    roll_a = atan2(R[2][1], R[2][2]);
    r_a = (180 * roll_a) / pi;
}
if (R[2][2] == 0)
{
    r_a = 0;
}
if (roll_a != 0)
{
    pitch_a = atan2(-R[2][0], ((R[2][1]) / sin(roll_a)));
    p_a = (180 * pitch_a) / pi;
}
if (roll_a == 0)
{
    p_a = 0;
}
if (R[0][0] != 0)
{
    yaw_a = atan((R[1][0]) / R[0][0]);
    y_a = (180 * yaw_a) / pi;
}
if (R[0][0] == 0)
{
    y_a = 0;
}
//Cálculo de ángulos con giróscopos
r_g = r_g + (G[0] * To);
p_g = p_g + (G[1] * To);
y_g = y_g + (G[2] * To);
//Implementación del filtro complementario:
r_0 = ((exp(-Wc * To)) * r_0) + ((1 - (exp(-Wc * To))) * r_aant) + (G[0] * To);
p_0 = ((exp(-Wc * To)) * p_0) + ((1 - (exp(-Wc * To))) * p_aant) + (G[1] * To);
y_0 = ((exp(-Wc * To)) * y_0) + ((1 - (exp(-Wc * To))) * y_aant) + (G[2] * To);
r_aant = r_a;
p_aant = p_a;
y_aant = y_a;

```

```

if (p_0 < 0) { d_0 = p_0 * (-0.5); }
if (p_0 >= 0) { d_0 = p_0 * 0.5; }
//Detección de instrumento insertado
set_ADC_channel(1); puerto_iz = read_ADC(); //lectura de QRE1113 izquierdo
set_ADC_channel(5); puerto_der = read_ADC(); //lectura de QRE1113 derecho
//Puerto izquierdo:
if (puerto_iz < 85)
{
    i = i + 1; j = 0; k = 0; l = 0;
    if (i > 3) { ins_iz = 1; }
}
if (puerto_iz > 85 && puerto_iz < 300)
{
    i = 0; j = j + 1; k = 0; l = 0;
    if (j > 3) { ins_iz = 2; }
}
if (puerto_iz > 760 && puerto_iz < 830)
{
    i = 0; j = 0; k = k + 1; l = 0;
    if (k > 3) { ins_iz = 3; }
}
if (puerto_iz > 740 && puerto_iz < 760 && ins_der != 2)
{ i = 0; j = 0; k = 0; l = l + 1; p_1 = 0; o = 0; if (l > 50) { l = l - 40; d_1 = 0; } if (l
> 3) { ins_iz = 0; } }
ins_iz = 0;
//Puerto izquierdo:
if (puerto_der < 75) { p = p + 1; q = 0; s = 0; u = 0; if (p > 3) { ins_der = 1; } }
if (puerto_der >= 75 && puerto_der < 300)
{
    p = 0; q = q + 1; s = 0; u = 0;
    if (q > 3) { ins_der = 2; }
}
if (puerto_der > 730 && puerto_der < 780)
{
    p = 0; q = 0; s = s + 1; u = 0;
    if (s > 3) { o = 0; p_2 = 0; ins_der = 3; }
}
if (puerto_der > 700 && puerto_der < 730 && ins_iz != 2)
{ p = 0; q = 0; s = 0; u = u + 1; p_2 = 0; o = 0; if (u > 50) { u = u - 40; d_2 = 0; } if (u
> 3) { ins_der = 0; } }
//Estimación de pitch 1 y 2
set_ADC_channel(0); sensor1_1 = 360 * ((read_ADC() - 28) / 963.0); // Pitch 1
p1_inc = (sensor1_1 - p1_ant) * 2.15;
if (p1_inc > 300 || p1_inc < -300) { p1_inc = 0; }
p_1sf = p_1sf + p1_inc;
p1_ant = sensor1_1;
p_1 = Filtro_pasobajas(p_1sf, p_1, 5);
set_ADC_channel(4); sensor1_2 = 360 * ((read_ADC() - 28) / 963.0); //pitch 2
p2_inc = (sensor1_2 - p2_ant) * 2.15;
if (p2_inc > 300 || p2_inc < -300) { p2_inc = 0; }
p_2sf = p_2sf + p2_inc;
p2_ant = sensor1_2;
p_2 = Filtro_pasobajas(p_2sf, p_2, 5);
//Estimación de desplazamientos 1 y 2
d1_inc = sensor2_1 - d1_ant; //Desplazamiento 1
if (d1_inc >= 20) { d1_inc = d1_inc - 25.1328; }
if (d1_inc <= -20) { d1_inc = d1_inc + 25.1328; }
d_1 = d_1 + d1_inc;
d1_ant = sensor2_1;
if (d_1 > 0) { d_1 = 0; }
d2_inc = sensor2_2 - d2_ant; //Desplazamiento 2
if (d2_inc >= 13) { d2_inc = d2_inc - 25.1328; }
if (d2_inc <= -13) { d2_inc = d2_inc + 25.1328; }
d_2 = d_2 + d2_inc;
d2_ant = sensor2_2;
if (d_2 > 0) { d_2 = 0; }
//Estimación de giros del artroscopio:
set_ADC_channel(7); gir_lente = 25 + ((read_ADC() / 1021.0) * 360.0);
gir_lentef = Filtro_pasobajas(gir_lente, gir_lentef, 6.3); //giro im artr
if (ins_iz == 2 || ins_der == 2)

```

```

{
  if (o == 0)
  {
    TCA9548A(0);
    //Calculando el ángulo inicial:
    medir MPU(0xD2, &axm, &aym, &azm, &gxm, &gym, &gzxm);
    Ax = -((2 * axm) - axmax - axmin) / (axmax - axmin);
    Ay = ((2 * aym) - aymax - aymin) / (aymax - aymin);
    Az = -((2 * azm) - azmax - azmin) / (azmax - azmin);
    Gx = -(gxm2 + gxcer2) / 131.072;
    Gy = (gym2 + gycer2) / 131.072;
    Gz = -(gzxm2 + gzcer2) / 131.072;
    pitch_artA = -(18 + (57.295 * (atan2(Ax, (sqrt((Ay * Ay) + (Az * Az)))))));
    gir_artin = gir_lente - pitch_artA;
    o = 1;
  }
  gir_im = -gir_artin + gir_lente;
}
v = v + 1;
if (v == 2100) { v = v - 500; }
//Estimación de apertura de la pinza:
set_ADC_channel(6); sensor6 = 3.5 * (((read_ADC()) / 1023.0) * 360.0) - 264.28);
ap_pin = Filtro_pasobajas(sensor6, ap_pin, 6.3); //apertura de la pinza
//Imprimiendo datos:
if (v <= 500)
{
  yf_0 = y_0;
  yf_0 = Filtro_pasobajas(y_0, yf_0, 10);
  output_toggle(PIN_A2);
  ycero_0 = yf_0;
}
if (v > 500)
{
  yf_0 = Filtro_pasobajas(y_0, yf_0, 10);
  if (usb_enumerated())
  {
    output_high(PIN_A2); //Parpadeo de led para verificar funcionamiento
    sprintf(msg, "%.1f,%.1f,%.1f,%.2f,%.2f,%.2f,%.2f,%.2f,%.2f,%.2f,%.2f,%.2f,%d,%d,%.1f,
%.1f\n", r_0, p_0, yf_0 - ycero_0, d_0, r_1, p_1, y_1, d_1, r_2, p_2, y_2, d_2,
ins_iz, ins_der, gir_im, ap_pin);
    printf(usb_cdc_putc, "%s", msg);
  }
}
output_toggle(PIN_D6); //Parpadeo de led para verificar funcionamiento
delay_ms(10);
}
}
}

```

Anexo C. Scripts de Unity para la simulación virtual

Script de recepción de datos:

```

using UnityEngine;
using System.Collections;
using System.Collections.Generic;
using System.Net;
using System.Net.Sockets;
using System.Linq;
using System;
using System.IO;
using System.Text;
using System.IO.Ports;
using UnityEngine.UI;
using TMPro;

```

```

public class Read : MonoBehaviour
{
    SerialPort serialPort = new SerialPort("COM10",115200);
    private void awake()
    {
        serialPort.Parity = Parity.None;
        serialPort.StopBits = StopBits.One;
        serialPort.DataBits = 8;
    }
    //Variables:
    String msg;
    public float r0, p0, y0;
    float y0sf, r0sf, p0sf;
    float dt;
    public float d0, r1 ,p1, y1, d1, r2, p2, y2, d2;
    public int ins_iz;
    public int ins_der;
    public float giro_im;
    public float ap_pin;
    public float x_1;
    public float y_1;
    public float z_1;
    public float roll_1;
    public float pitch_1;
    public float yaw_1;
    public float x_2;
    public float y_2;
    public float z_2;
    public float roll_2;
    public float pitch_2;
    public float yaw_2;
    public float P1_1;
    public float P1_2;
    public int i = 2;
    float[,] bRc = new float[,] { { 0, 0, 0 }, { 0, 0, 0 }, { 0, 0, 0 } };
    float[,] bRd = new float[,] { { 0, 0, 0 }, { 0, 0, 0 }, { 0, 0, 0 } };
    float[,] aRb = new float[,] { { 0, 0, 0 }, { 0, 0, 0 }, { 0, 0, 0 } };
    float[] aPb = new float[] { 0, 0, 0 };
    float[] cP1 = new float[] { 0, 0, 0 };
    float[] dP2 = new float[] { 0, 0, 0 };
    float[] bPc = new float[] { 0, 0, 0 };
    float[] bPd = new float[] { 0, 0, 0 };
    float[] aPd = new float[] { (float)31.5, (float)66.8, -15 };
    float[] aPc = new float[] { (float)-31.5, (float)66, (float)-15.4 };
    public float[] aP1 = new float[] { 0, 0, 0 };
    public float[] aP2 = new float[] { 0, 0, 0 };
    double g = 57.2958;//convertir grados a radianes
    //Conección con Arreglo instrumental:
    public void AttemptConnection()
    {
        try
        {
            serialPort.Open();
            serialPort.ReadTimeout = 10;
        }
        catch (System.Exception ex)
        {
        }
    }
    public void Disconnect()
    {
        serialPort.Close();
    }
    //Función para multiplicar dos matrices de 3x3:
    public static float[,] CrossM(float[,] a, float[,] b2)
    {
        float[,] multm = new float[,] { { 0, 0, 0 }, { 0, 0, 0 }, { 0, 0, 0 } };
        multm[0, 0] = (a[0, 0] * b2[0, 0]) + (a[0, 1] * b2[1, 0]) + (a[0, 2] * b2[2, 0]);
        multm[0, 1] = (a[0, 0] * b2[0, 1]) + (a[0, 1] * b2[1, 1]) + (a[0, 2] * b2[2, 1]);
        multm[0, 2] = (a[0, 0] * b2[0, 2]) + (a[0, 1] * b2[1, 2]) + (a[0, 2] * b2[2, 2]);
    }
}

```

```

    multm[1, 0] = (a[1, 0] * b2[0, 0]) + (a[1, 1] * b2[1, 0]) + (a[1, 2] * b2[2, 0]);
    multm[1, 1] = (a[1, 0] * b2[0, 1]) + (a[1, 1] * b2[1, 1]) + (a[1, 2] * b2[2, 1]);
    multm[1, 2] = (a[1, 0] * b2[0, 2]) + (a[1, 1] * b2[1, 2]) + (a[1, 2] * b2[2, 2]);
    multm[2, 0] = (a[2, 0] * b2[0, 0]) + (a[2, 1] * b2[1, 0]) + (a[2, 2] * b2[2, 0]);
    multm[2, 1] = (a[2, 0] * b2[0, 1]) + (a[2, 1] * b2[1, 1]) + (a[2, 2] * b2[2, 1]);
    multm[2, 2] = (a[2, 0] * b2[0, 2]) + (a[2, 1] * b2[1, 2]) + (a[2, 2] * b2[2, 2]);
    return multm;
}
//Función para multiplicar 1 matriz de 3x3 por un vector de 3
public static float[] CrossV(float[,] m, float[] v)
{
    float[] multv = new float[] { 0, 0, 0 };
    multv[0] = (m[0, 0] * v[0]) + (m[0, 1] * v[1]) + (m[0, 2] * v[2]);
    multv[1] = (m[1, 0] * v[0]) + (m[1, 1] * v[1]) + (m[1, 2] * v[2]);
    multv[2] = (m[2, 0] * v[0]) + (m[2, 1] * v[1]) + (m[2, 2] * v[2]);
    return multv;
}
//Función para sumar dos vectores:
public static float[] SumV(float[] a, float[] b)
{
    float[] sumv = new float[] { 0, 0, 0 };
    sumv[0] = (a[0] + b[0]);
    sumv[1] = (a[1] + b[1]);
    sumv[2] = (a[2] + b[2]);
    return sumv;
}
//Función para aplicar un filtro paso bajas:
float Filtro_pasobajas(float valor_medido, float valor_anterior, float Wc)
{
    float valor_filtrado = 0;
    return valor_filtrado = (valor_anterior * (1 / (1 + (dt * Wc)))) + (valor_medido * ((Wc * dt) / (1 + (Wc * dt))));
}

void Start()
{
}
void Update()
{
    try
    {
        if (serialPort.IsOpen)
        {
            print(r0 + "," + p0 + "," + y0 + "," + d0 + "," + r1 + "," + p1 + "," + y1 + "," +
d1 + "," + r2 + "," + p2 + "," + y2 + "," + d2 + "," + ins_iz + "," + ins_der + "," + giro_im + "," +
+ ap_pin);
            msg = serialPort.ReadTo("\n");
            String[] datos = msg.Split(',');
            r0 = float.Parse(datos[0]);
            p0 = float.Parse(datos[1]);
            y0 = float.Parse(datos[2]);
            d0 = float.Parse(datos[3]);
            r1 = float.Parse(datos[4]);
            p1 = float.Parse(datos[5]);
            y1 = float.Parse(datos[6]);
            d1 = float.Parse(datos[7]);
            r2 = float.Parse(datos[8]);
            p2 = float.Parse(datos[9]);
            y2 = float.Parse(datos[10]);
            d2 = float.Parse(datos[11]);
            ins_iz = int.Parse(datos[12]);
            ins_der = int.Parse(datos[13]);
            giro_im = float.Parse(datos[14]);
            ap_pin = float.Parse(datos[15]);
        }
        cP1[1] = d1;
        dP2[1] = d2;
    }
}
//Matriz de rotación de la rodilla:
aRb[0,0]=(float)(Math.Cos(y0/g)*Math.Cos(p0/g));
aRb[0,1]=(float)((Math.Cos(y0/g)*Math.Sin(p0/g)*Math.Sin(r0/g))-(Math.Sin(y0/g)*Math.Cos(r0/g)));

```

```

aRb[0,2]=(float)((Math.Cos(y0/g)*Math.Sin(p0/g)*Math.Cos(r0/g))+Math.Sin(y0/g)*Math.Sin(r0/g));
aRb[1,0]=(float)(Math.Sin(y0/g)*Math.Cos(p0/g));
aRb[1,1]=(float)((Math.Sin(y0/g)*Math.Sin(p0/g)*Math.Sin(r0/g))+(Math.Cos(y0/g)*Math.Cos(r0/g)));
aRb[1,2]=(float)((Math.Sin(y0/g)*Math.Sin(p0/g)*Math.Cos(r0/g)-(Math.Cos(y0/g)*Math.Sin(r0/g)));
aRb[2,0]=(float)(-Math.Sin(p0/g));
aRb[2,1]=(float)(Math.Cos(p0/g)*Math.Sin(r0/g));
aRb[2,2]=(float)(Math.Cos(p0/g)*Math.Cos(r0/g));
//Matriz de rotación del puerto de entrada izquierdo:
bRc[0,0]=(float)((Math.Cos(p1/g)*Math.Cos(y1/g)-(Math.Sin(p1/g)*Math.Sin(r1/g)*Math.Sin(y1/g)));
bRc[0,1]=(float)(-Math.Cos(r1/g)*Math.Sin(y1/g));
bRc[0,2]=(float)((Math.Cos(y1/g)*Math.Sin(p1/g)+(Math.Cos(p1/g)*Math.Sin(r1/g)*Math.Sin(y1/g)));
bRc[1,0]=(float)((Math.Cos(y1/g)*Math.Sin(p1/g)*Math.Sin(r1)+(Math.Cos(p1/g)*Math.Sin(y1/g)));
bRc[1,1]=(float)(Math.Cos(r1/g)*Math.Cos(y1/g));
bRc[1,2]=(float)(-Math.Cos(p1/g)*Math.Cos(y1/g)*Math.Sin(r1/g)+(Math.Sin(p1/g)*Math.Sin(y1/g)));
bRc[2,0]=(float)((-Math.Cos(r1/g)*Math.Sin(p1/g)));
bRc[2,1]=(float)(Math.Sin(r1/g));
bRc[2,2]=(float)(Math.Cos(p1/g)*Math.Cos(r1/g));
//Matriz de rotación del puerto de entrada izquierdo:
bRd[0,0]=(float)((Math.Cos(p2/g)*Math.Cos(y2/g)-(Math.Sin(p2/g)*Math.Sin(r2/g)*Math.Sin(y2/g)));
bRd[0,1]=(float)(-Math.Cos(r2/g)*Math.Sin(y2/g));
bRd[0,2]=(float)((Math.Cos(y2/g)*Math.Sin(p2/g)+(Math.Cos(p2/g)*Math.Sin(r2/g)*Math.Sin(y2/g)));
bRd[1,0]=(float)((Math.Cos(y2/g)*Math.Sin(p2/g)*Math.Sin(r2)+(Math.Cos(p2/g)*Math.Sin(y2/g)));
bRd[1,1]=(float)(Math.Cos(r2/g)*Math.Cos(y2/g));
bRd[1,2]=(float)(-Math.Cos(p2/g)*Math.Cos(y2/g)*Math.Sin(r2/g)+(Math.Sin(p2/g)*Math.Sin(y2/g)));
bRd[2,0]=(float)((-Math.Cos(r2/g)*Math.Sin(p2/g)));
bRd[2,1]=(float)(Math.Sin(r2/g));
bRd[2,2]=(float)(Math.Cos(p2/g)*Math.Cos(r2/g));
//posición de las puntas de los instrumentos
aP1 = SumV(SumV(aPb,CrossV(aRb,aPc)), CrossV(aRb,CrossV(bRc,cP1)));
aP2 = SumV(SumV(aPb, CrossV(aRb, aPd)), CrossV(aRb,CrossV(bRd,dP2)));
x_1 = aP1[0];
y_1 = aP1[1];
z_1 = aP1[2];
roll_1 = r0 + r1;
pitch_1 = p0 + p1;
yaw_1 = y0 + y1;
x_2 = aP2[0];
y_2 = aP2[1];
z_2 = aP2[2];
roll_2 = r0 + r2;
pitch_2 = p0 + p2;
yaw_2 = y0 + y2;
}
catch (System.Exception ex)
{
    ex = new System.Exception();
}
}
}
}

```

Script de movimiento de rodilla:

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class IMU : MonoBehaviour
{
    public float r0;
    public float p0;
    public float y0;
    public float d0;
    public Read read;

    void Start()
    {
    }
}

```

```

void Update()
{
    r0 = read.r0;
    p0 = read.p0;
    y0 = read.y0;
    d0 = read.d0;
    transform.rotation = Quaternion.Euler(r0, y0, -p0);
    transform.position = new Vector3(0,-d0,0);
}
}

```

Script de movimiento de artroscopio:

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using System;

public class Camara : MonoBehaviour
{
    private double x_cam;
    private double y_cam;
    private double z_cam;
    private double x;
    private double y;
    private double z;
    private double roll_cam;
    private double pitch_cam;
    private double yaw_cam;
    public instrumento1 read;
    float[] aPp = new float[] { 2, 2, 0 };
    double g = 57.2958;//convertir grados a radianes

    void Start()
    {
    }

    void Update()
    {
        x_cam = read.x_art;
        y_cam = read.y_art;
        z_cam = read.z_art;
        roll_cam = read.roll_art;
        pitch_cam = read.pitch_art;
        yaw_cam = read.yaw_art;
        transform.position = new Vector3((float)(x_cam), (float)(z_cam), -(float)(y_cam));
        transform.rotation = Quaternion.Euler((float)(roll_cam + 30*Math.Sin(pitch_cam/g)), (float)(yaw_cam +
30*Math.Cos(pitch_cam/g)), 0);
    }
}

```

Script de movimiento de luz del artroscopio:

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using System;

public class luz : MonoBehaviour
{
    private double x_luz;
    private double y_luz;
    private double z_luz;
    private double roll_luz;
    private double pitch_luz;
}

```



```

private double yaw_luz;
public instrumento1 read;

void Start()
{
}

void Update()
{
    x_luz = read.x_art;
    y_luz = read.y_art;
    z_luz = read.z_art;

    roll_luz = read.roll_art;
    pitch_luz = read.pitch_art;
    yaw_luz = read.yaw_art;

    transform.position = new Vector3((float)x_luz, (float)z_luz, 5-(float)y_luz);
    transform.rotation = Quaternion.Euler((float)roll_luz, (float)(yaw_luz), -
(float)pitch_luz);
}
}

```

Script de movimiento del lente del artroscopio:

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using System;
public class lente : MonoBehaviour
{
    private double x_lente;
    private double y_lente;
    private double z_lente;
    private double roll_lente;
    private double pitch_lente;
    private double yaw_lente;
    public instrumento1 read;
    double g = 57.2958;//convertir grados a radianes

    void Start()
    {
    }

    void Update()
    {
        x_lente = read.x_art;
        y_lente = read.y_art;
        z_lente = read.z_art;

        roll_lente = read.roll_art;
        pitch_lente = read.pitch_art;
        yaw_lente = read.yaw_art;

        transform.position = new Vector3((float)x_lente, (float)z_lente, -(float)y_lente);
        transform.rotation = Quaternion.Euler((float)(roll_lente + 30*Math.Sin(pitch_lente/g)), (float)(yaw_lente +
30*Math.Cos(pitch_lente/g)), - ((float)pitch_lente));
    }
}

```

Script del movimiento del palpador:

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

```

```

public class palpador : MonoBehaviour
{
    public Read read;
    public double x_pal;
    public double y_pal;
    public double z_pal;
    public double roll_pal;
    public double pitch_pal;
    public double yaw_pal;
    double roll_1;
    double pitch_1;
    double yaw_1;
    double roll_2;
    double pitch_2;
    double yaw_2;
    int ins_iz;
    int ins_der;
    double[] aP1 = new double[] { 0, 0, 0 };
    double[] aP2 = new double[] { 0, 0, 0 };

    void Start() { }

    void Update()
    {
        aP1[0] = read.x_1;
        aP1[1] = read.y_1;
        aP1[2] = read.z_1;
        roll_1 = read.roll_1;
        pitch_1 = read.pitch_1;
        yaw_1 = read.yaw_1;
        aP2[0] = read.x_2;
        aP2[1] = read.y_2;
        aP2[2] = read.z_2;
        roll_2 = read.roll_2;
        pitch_2 = read.pitch_2;
        yaw_2 = read.yaw_2;
        ins_iz = read.ins_iz;
        ins_der = read.ins_der;

        if (ins_iz == 0 | ins_der == 0)
        {
            x_pal = 500;
            y_pal = -500;
            z_pal = 0;
            roll_pal = 0;
            pitch_pal = 0;
            yaw_pal = 0;
        }

        if (ins_iz == 1)
        {
            x_pal = aP1[0];
            y_pal = aP1[1];
            z_pal = aP1[2];
            roll_pal = roll_1;
            pitch_pal = pitch_1;
            yaw_pal = yaw_1;
        }

        if (ins_der == 1)
        {
            x_pal = aP2[0];
            y_pal = aP2[1];
            z_pal = aP2[2];
            roll_pal = roll_2;
            pitch_pal = pitch_2;
            yaw_pal = yaw_2;
        }
        transform.rotation = Quaternion.Euler((float)roll_pal, (float)yaw_pal, -(float)pitch_pal);
        transform.position = new Vector3((float)x_pal, (float)z_pal, -(float)y_pal);
    }
}

```

#	Tiempo (s)	Mano izquierda												Mano derecha											
		Longitud de la trayectoria (cm)	Percepción de la profundidad (cm)	Profundidad a lo largo del Trocar (mm)	Suavidad del movimiento (km/s3)	Velocidad promedio (mm/s)	Aceleración promedio (mm/s2)	Tiempo inactivo (%)	Economía del área (1000)	Economía del volumen (1000)	Energía del área	Energía del volumen (1/100)	Tiempo (s)	Longitud de la trayectoria (cm)	Percepción de la profundidad (cm)	Profundidad a lo largo del Trocar (mm)	Suavidad del movimiento (km/s3)	Velocidad promedio (cm/s)	Aceleración promedio (cm/s2)	Tiempo inactivo (%)	Economía del área (1000)	Economía del volumen (1000)	Energía del área (10)	Energía del volumen (1/10)	
Expertos																									
1	62.402	14.0529	10.1182	73.51	38.22366	15.38089	22.0203	17.46	48.51	39.3	26.66102	9.321013	62.402	9.904	7.5434	55.54	26.16708	98.649	138.7548	41.27	57.02	51.56	74.5863	47.08855	
2	60.198	14.4289	10.4301	70.194	39.11983	15.25275	22.89343	22.95	45.13	34.13	26.8597	11.91273	60.198	9.9106	7.1827	49.405	25.74636	93.8883	134.2484	36.07	55.84	44.61	72.9095	61.38174	
3	47.641	7.4337	5.5611	45.217	21.63391	12.13728	15.68759	12	64.51	52.21	48.73298	23.50762	47.641	7.4257	5.9703	38.813	22.31019	107.5381	142.5752	36	68.65	56.72	85.3193	68.67263	
4	53.755	8.0489	6.1236	48.903	23.73795	10.75174	14.13666	23.64	65.05	53.76	42.37758	18.3774	53.755	8.7386	6.8826	46.734	25.20166	105.1946	141.2837	32.73	60.68	48.58	83.0688	76.067	
5	56.927	14.505	11.218	67.233	45.08815	14.9542	22.41582	20.69	46.49	40.08	24.02831	7.057194	56.927	10.3968	7.9141	50.048	33.52017	102.4687	143.7495	39.66	52.17	47.66	64.6378	44.98822	
6	50.359	10.9689	8.5254	62.152	31.47488	14.29733	19.55533	13.21	49.17	40.55	39.45165	16.12368	50.359	8.4711	7.0601	52.001	25.92832	108.5249	147.5152	35.85	57.13	51.03	85.1055	61.67858	
7	56.151	14.0854	10.1585	71.495	40.15122	16.36973	24.02986	16.07	45.66	35.96	25.41765	10.10212	56.151	9.2009	6.9228	49.914	24.76142	103.3772	143.9159	32.14	57.31	44.34	74.4112	75.99038	
8	46.868	7.3381	5.6354	40.565	21.39848	11.19392	15.06096	19.15	62.13	52.08	51.90357	24.34959	46.868	8.8815	7.1948	47.229	25.41522	125.5676	166.4065	19.15	69.91	58.78	72.9144	38.27711	
9	61.971	14.6378	10.7922	67.717	42.80481	16.27023	23.75911	15.87	43.18	35.93	27.56841	9.692821	61.971	10.1748	7.3887	44.176	30.68092	106.9244	148.5484	31.75	54.47	47.11	63.8053	48.06418	
Intermedios																									
1	86.891	9.9225	7.8358	50.702	22.28412	7.63348	10.0522	37.5	52.23	39.28	62.89672	35.2654	86.891	10.1873	8.1018	49.166	22.42734	71.7394	96.7317	46.59	52.56	44.15	137.8038	95.13153	
2	85.255	11.1372	8.8178	55.036	27.23613	8.26336	10.8862	39.77	55.15	42.92	45.21076	19.35876	85.255	9.6415	7.7315	45.081	21.68749	66.0236	89.1029	55.68	50.42	38.43	167.7909	160.4077	
3	70.098	10.3377	8.1672	59.431	27.2979	11.14544	15.26233	45.59	51.89	44.91	49.17748	19.01927	70.098	22.101	17.454	118.961	42.23184	88.3222	124.9718	38.24	22.09	18.84	107.5444	81.80615	
4	69.443	11.629	8.6407	68.109	31.17596	9.79962	14.08686	37.14	48.36	40.68	47.52755	18.00803	69.443	9.5816	7.0471	49.34	24.45747	84.3431	109.1302	47.14	54.34	46.55	134.7223	87.89153	
5	61.037	11.384	8.747	58.639	30.85865	11.37458	14.7701	14.06	46.78	38.91	50.81127	21.00374	61.037	9.395	7.354	54.484	26.61582	94.9638	126.1011	37.51	57.36	45.09	106.5107	88.84198	
6	66.876	12.476	10.332	61.47	32.81992	9.79228	13.04964	32.35	43.02	36.09	51.05793	19.83508	66.876	12.3692	10.3292	74.255	30.46547	87.9391	119.4836	26.47	39.15	34.16	143.5894	95.49489	
7	85.091	8.665	6.721	48.452	22.00671	7.43748	10.30871	54.55	54.17	43.86	83.08898	42.38119	85.091	11.1017	8.6567	58.722	27.52344	80.6786	110.85	44.32	46.93	36.82	155.5763	125.7094	
8	93.908	18.6541	14.531	98.43	41.19991	9.16162	13.11592	38.54	29.15	24.21	61.39687	25.88316	93.908	13.006	10.0356	61.05	26.34481	72.7279	91.9175	48.96	37.34	32.35	166.8191	141.517	
9	76.579	16.7534	13.3827	90.946	42.43416	12.3064	17.60444	20	35.93	33.06	49.68278	13.36719	76.579	13.5323	10.7111	73.34	33.88721	88.4202	127.2535	33.75	44.23	34.04	151.8068	93.31484	
Principiantes																									
1	146.34	27.2926	20.9398	128.769	52.09773	11.13155	15.55705	30.46	28.5	23.84	38.95739	10.93264	146.34	20.8154	16.1353	102.714	38.46332	77.9899	107.2812	45.03	29.49	26.34	117.3364	71.41481	
2	114.642	18.7984	15.2468	100.417	37.98871	8.99138	12.19563	29.41	27.46	24.52	96.1324	32.92794	114.642	20.4426	15.4418	98.914	39.34787	99.2451	136.2475	23.53	35.38	27.18	128.4796	71.45111	
3	65.564	13.3875	11.3216	69.452	33.89799	11.96771	17.31011	14.93	39.21	34.59	44.08778	15.99743	65.564	10.5211	8.7278	62.795	27.18173	99.5069	136.4291	35.82	47.88	41.7	100.105	79.0769	
4	56.543	7.9298	6.0371	40.335	20.6403	9.8622	13.67906	36.84	55.66	46.01	53.27555	27.63282	56.543	8.7012	6.8153	49.499	22.76565	108.6566	144.8412	28.07	57.05	46.96	91.9493	81.48049	
5	221.92	30.3602	23.5492	158.234	49.92355	6.93125	9.50992	50.43	18.79	17.38	165.3131	45.80651	221.92	40.6944	29.0981	170.852	69.15506	68.5101	100.5792	46.98	19.61	14.81	204.387	99.18147	
6	176.48	21.425	17.002	106.464	38.59066	6.07367	8.16738	56.22	31.24	26.45	92.61094	27.60902	176.48	31.1951	23.8279	142.587	59.57837	71.9847	97.9663	46.49	22.15	18.64	275.2915	107.0596	
7	103.039	18.6162	14.3473	90.314	39.88244	10.68015	15.68326	31.37	39.65	30.75	29.54754	10.69982	103.039	12.7649	9.5023	55.984	27.87724	69.5414	95.1385	51.96	47.79	37.2	111.219	77.56158	
8	56.013	11.398	9.4546	58.786	30.30233	13.34952	18.6227	17.54	46.4	39.14	37.31372	15.68822	56.013	9.8418	8.2434	63.017	26.89138	115.5309	146.5658	19.3	45.24	42.78	104.6816	81.80349	
9	127.919	17.0046	13.3001	83.873	34.3818	7.80116	10.58559	41.35	38.62	30.53	53.71287	21.06857	127.919	13.5329	10.4574	74.57	28.35399	60.5094	77.412	58.65	43.29	35.95	153.795	102.5451	

Anexo E. Programación de la red neuronal RTC en Matlab.

```
%Entrenamiento del clasificador RTC
clear all;clc;
%Se carga la base de datos:
load expertos_entrenamiento.txt;
load intermedios_entrenamiento.txt;
load principiantes_entrenamiento.txt;
%Características de la red:
Nn=32000; %Número de neuronas
%generación de neuronas X,l,h,a,b
X=zeros(18,24);
l=zeros(24,Nn);
h=zeros(24,Nn);
Neuronas_a=zeros(18,24,Nn);
Neuronas_b=zeros(Nn,18);
ll=0;
lm=0;
%Generación de capas no entrenables de la red RTC
%Vectores de entrada X:
for i=1:6
    for j=1:24
        X(i,j)=principiantes_entrenamiento(i,j);
    end
end
for i=1:6
    for j=1:24
        k=i+6;
        X(k,j)=intermedios_entrenamiento(i,j);
    end
end
for i=1:6
    for j=1:24
        k=i+12;
        X(k,j)=expertos_entrenamiento(i,j);
    end
end
end

%Generación de umbrales aleatorios h y l
for i=1:24
    for j=1:Nn
        r1=-50+(280+50)*rand; r2=-50+(280+50)*rand;
        if r1>r2
            h(i,j)=r1; l(i,j)=r2;
        end
        if r1<r2
            h(i,j)=r2; l(i,j)=r1;
        end
    end
end
end

%Generación de neuronas a
for i=1:18
    for j=1:24
        for k=1:Nn
            if X(i,j)>l(j,k) && X(i,j)<h(j,k)
                Neuronas_a(i,j,k)=1;
                ll=ll+1;
            else
                Neuronas_a(i,j,k)=0;
            end
        end
    end
end
end
```

```

%Generación de neuronas b
for i=1:18
    for j=1:Nn
        for k=2:24
            act = Neuronas_a(i,k,j)*Neuronas_a(i,k-1,j);
        end
        if act==1
            Neuronas_b(j,i)=1;
            lm=lm+1;
        else
            Neuronas_b(j,i)=0;
        end
    end
end

%Generación de vectores de salida deseados
for j=1:6
    T_exp(:,j)=[1,0,0];
end
for j=1:6
    T_inter(:,j)=[0,1,0];
end
for j=1:6
    T_princ(:,j)=[0,0,1];
end
targets=[T_exp T_inter T_princ];

%Entrenamiento de la red RTC
red = perceptron(); %Se utiliza una red perceptrón simple de una capa
red.trainParam.epochs=(200); %Número de ciclos máximos de entrenamiento
red.trainParam.max_fail=10; %verifica mínimos globales posibles
red.trainParam.min_grad=1e-6; %Error máximo permitido
red.trainParam.mu=0.01; %factor de aprendizaje de crecientes
red.trainParam.mu_inc=10; %factor de aprendizaje crecientes
red.layers{1}.transferFcn='softmax'; %Función de activación
configure(red,Neuronas_b,targets); %Configurar red
[red,tr]=train(red,Neuronas_b,targets); %Configurar entrenamiento
ll %variable para calcular el número de neuronas a activadas
lm %variable para calcular el número de neuronas b activadas

%Reconocimiento de usuarios que no participaron en la fase de entrenamiento

clear all;clc;
load('entrenamiento.mat')%Se cargan los datos del entrenamiento
%Se carga la base de datos:
load expertos_reconocimiento.txt;
load intermedios_reconocimiento.txt;
load principiantes_reconocimiento.txt;

%Generación de neuronas X,a y b de los datos del reconocimiento
X_reconocimiento=zeros(9,24);
Neuronas_a_reconocimiento=zeros(9,24,Nn);
Neuronas_b_reconocimiento=zeros(Nn,9);

%Vectores de entrada X de reconocimiento:
for i=1:3
    for j=1:24
        X_reconocimiento(i,j)=principiantes_reconocimiento(i,j);
    end
end
for i=1:3
    for j=1:24
        k=i+3;

```

```

        X_reconocimiento(k,j)=intermedios_reconocimiento(i,j);
    end
end
for i=1:3
    for j=1:24
        k=i+6;
        X_reconocimiento(k,j)=expertos_reconocimiento(i,j);
    end
end

%Neuronas a, de datos para reconocimiento
for i=1:9
    for j=1:24
        for k=1:Nn
            if X_reconocimiento(i,j)>l(j,k) && X_reconocimiento(i,j)<h(j,k)
                Neuronas_a_reconocimiento(i,j,k)=1;
            else
                Neuronas_a_reconocimiento(i,j,k)=0;
            end
        end
    end
end

%Neuronas b, de datos para reconocimiento
for i=1:9
    for j=1:Nn
        for k=2:24
            act = Neuronas_a_reconocimiento(i,k,j)*Neuronas_a_reconocimiento(i,k-
1,j);
            end
            if act==1
                Neuronas_b_reconocimiento(j,i)=1;
            else
                Neuronas_b_reconocimiento(j,i)=0;
            end
        end
    end
end

%Reconocimiento de la red
for i=1:Nn
    rec_1(i) = Neuronas_b_reconocimiento(i,1)';
end
exp_1=red(rec_1')%Reconocimiento usuario experto 1

for i=1:Nn
    rec_2(i) = Neuronas_b_reconocimiento(i,2)';
end
exp_2=red(rec_2')%Reconocimiento usuario experto 2

for i=1:Nn
    rec_3(i) = Neuronas_b_reconocimiento(i,3)';
end
exp_3=red(rec_3') %Reconocimiento usuario experto 3

for i=1:Nn
    rec_4(i) = Neuronas_b_reconocimiento(i,4)';
end
inter_1=red(rec_4') %Reconocimiento usuario intermedio 1

for i=1:Nn
    rec_5(i) = Neuronas_b_reconocimiento(i,5)';
end
inter_2=red(rec_5') %Reconocimiento usuario intermedio 2

```

```

for i=1:Nn
    rec_6(i) = Neuronas_b_reconocimiento(i,6)';
end
inter_3=red(rec_6') %Reconocimiento usuario intermedio 3

for i=1:Nn
    rec_7(i) = Neuronas_b_reconocimiento(i,7)';
end
prin_1=red(rec_7') %Reconocimiento usuario principiante 1

for i=1:Nn
    rec_8(i) = Neuronas_b_reconocimiento(i,8)';
end
prin_2=red(rec_8') %Reconocimiento usuario principiante 1

for i=1:Nn
    rec_9(i) = Neuronas_b_reconocimiento(i,9)';
end
prin_3=red(rec_9') %Reconocimiento usuario principiante 1

```

Anexo F. Programación de la red neuronal *perceptrón* en *Matlab*.

```

%Entrenamiento de una Red neuronal Perceptrón simple

clear all;clc;
%Cargar base de datos
load expertos_entrenamiento.txt;
load intermedios_entrenamiento.txt;
load principiantes_entrenamiento.txt;

%Generar vectores de entrada
input=[expertos_entrenamiento' intermedios_entrenamiento'
principiantes_entrenamiento'];

%Generar vectores de salida
for j=1:6
    T_exp(:,j)=[1,0,0];
end
for j=1:6
    T_inter(:,j)=[0,1,0];
end
for j=1:6
    T_princ(:,j)=[0,0,1];
end
targets=[T_exp T_inter T_princ];

%Entrenamiento de la red
red = perceptron(); %Se utiliza una red perceptrón simple de una capa
red.trainParam.epochs=(200); %Número de ciclos máximos de entrenamiento
red.trainParam.max_fail=10; %verifica mínimos globales posibles
red.trainParam.min_grad=1e-6; %Error máximo permitido
red.trainParam.mu=0.1; %Factor de aprendizaje decrecientes
red.trainParam.mu_inc=10; %Factor de aprendizaje crecientes
red.layers{1}.transferFcn='softmax'; %Función de activación de las neuronas
configure(red,input,targets); %Configurar red
[red,tr]=train(red,input,targets); %Configurar entrenamiento

%Reconocimiento de usuarios que no participaron en la fase de entrenamiento
clc; clear all;

```

```

load('entrenamiento.mat')
%Se carga la base de datos:
load expertos_reconocimiento.txt;
load intermedios_reconocimiento.txt;
load principiantes_reconocimiento.txt;

%Reconocimiento de la red
for i=1:24
    rec_1(i) = expertos_reconocimiento(1,i)';
end
exp_1=red(rec_1') %Reconocimiento usuario experto 1

for i=1:24
    rec_2(i) = expertos_reconocimiento(2,i)';
end
exp_2=red(rec_2') %Reconocimiento usuario experto 2

for i=1:24
    rec_3(i) = expertos_reconocimiento(3,i)';
end
exp_3=red(rec_3') %Reconocimiento usuario experto 3

for i=1:24
    rec_4(i) = intermedios_reconocimiento(1,i)';
end
inter_1=red(rec_4') %Reconocimiento usuario intermedio 1

for i=1:24
    rec_5(i) = intermedios_reconocimiento(2,i)';
end
inter_2=red(rec_5') %Reconocimiento usuario intermedio 2

for i=1:24
    rec_6(i) = intermedios_reconocimiento(3,i)';
end
inter_3=red(rec_6') %Reconocimiento usuario intermedio 3

for i=1:24
    rec_7(i) = principiantes_reconocimiento(1,i)';
end
prin_1=red(rec_7') %Reconocimiento usuario principiante 1

for i=1:24
    rec_8(i) = principiantes_reconocimiento(2,i)';
end
prin_2=red(rec_8') %Reconocimiento usuario principiante 1

for i=1:24
    rec_9(i) = principiantes_reconocimiento(3,i)';
end
prin_3=red(rec_9') %Reconocimiento usuario principiante 1

```