

Entorno de trabajo ciber-físico para cirugía laparoscópica

Juan M. Herrera-López*, Álvaro Galán-Cuenca, Isabel García-Morales, Marcos Rollón, Irene Rivas-Blanco, Víctor F. Muñoz

Departamento de Ingeniería de Sistemas y Automática, Universidad de Málaga, C/ Severo Ochoa, nº 4, 29590, Málaga, España.

To cite this article: Herrera-López, J.M., Galán-Cuenca, A., García-Morales, I., Rollón-Rivas, M., Rivas-Blanco, I., Muñoz-Martínez, V.F. 2024. Cyber-physical framework for laparoscopic surgery. Revista Iberoamericana de Automática e Informática Industrial 21, 69-80. <https://doi.org/10.4995/riai.2023.18753>

Resumen

Este trabajo desarrolla la implantación de un entorno de trabajo ciber-físico para cirugía laparoscópica que permite utilizar un simulador quirúrgico con un brazo robótico, para así facilitar el aprendizaje y la investigación con este tipo de sistemas. Se propone y valida una configuración para cada uno de los elementos del quirófano que permite cumplir todas las restricciones funcionales. También se implanta un Modelo Cinemático Inverso para el brazo robótico redundante que devuelve la solución más adecuada que asegura el cumplimiento de estas restricciones. El entorno de trabajo se ha implementado haciendo uso de ROS y MATLAB, realizando una serie de pruebas a fin de validar el desarrollo de aplicaciones utilizando este framework.

Palabras clave: Cirugía laparoscópica, Simulador quirúrgico, Sistema ciber-físico, Robot manipulador redundante, Modelo Cinemático Inverso, Adquisición de vídeo.

Cyber-physical framework for laparoscopic surgery

Abstract

This work develops the implementation of a cyber-physical framework for laparoscopic surgery that allows for the use of a surgical simulator with a robotic arm, in order to facilitate learning and research with this type of systems. A configuration for every element in the operation room that allows all the functional restrictions to be fulfilled is proposed and validated. An inverse kinematic model for the redundant robotic arm is also implemented, which returns the most appropriate solution that ensures compliance with these restrictions. Finally, an interface is developed using ROS and MATLAB that allows future work to be carried out using this framework, presenting a simple application as an example of use.

Keywords: Laparoscopic surgery, Surgical simulator, Cyber-physical system, Redundant manipulator robot, Inverse Kinematics Model, Video acquisition.

1. Introducción

La cirugía laparoscópica se trata de un tipo de cirugía mínimamente invasiva en la que se abordan intervenciones en la zona abdominal. Este tipo de cirugía fue originada por la necesidad de desarrollar procedimientos que minimizaran las heridas causadas por las aperturas típicas de la cirugía convencional abierta. En esta, las intervenciones se realizan introduciendo instrumentos tubulares alargados especiales y una cámara endoscópica por pequeñas incisiones en el abdomen del paciente, previamente insuflado con dióxido de carbono para poder

crear un espacio de trabajo. Este tipo de cirugía presenta una serie de ventajas, como una reducción de la mortalidad, de la duración de las estancias hospitalarias, y de los costes de intervención (Tiwari et al., 2011). Sin embargo, también acarrea algunas desventajas como la pronunciada curva de aprendizaje, la pérdida de visión 3D, o la falta de ergonomía, que dan lugar a una mayor fatiga para el cirujano, reduciendo su rendimiento (Pérez-Duarte et al., 2012). Además, la falta de ergonomía y las posturas forzadas durante las operaciones de laparoscopia pueden acarrear problemas de salud para los cirujanos (Ae et al., 2009), por lo que se han desarrollado artículos cuyo objetivo es

*Autor para correspondencia: juanma.hl@uma.es

estudiar a fondo los aspectos cuantificables de la ergonomía en laparoscopia con un énfasis en las posiciones más arriesgadas como la rotación del cuello o la carga asimétrica de los hombros (Aitchison et al., 2016).

Para lidiar con la dificultad que supone el aprendizaje de cirugía laparoscópica, se han desarrollado dispositivos para el entrenamiento mediante simulación (Jaffray, 2005), que sirven tanto para mejorar la calidad de las intervenciones de los cirujanos que se entrenan con ellos como para elevar su satisfacción con su formación (Zendejas et al., 2013). Además, la inclusión de realimentación háptica en los simuladores también se ve correlada con la satisfacción de los cirujanos con su aprendizaje (Panait et al., 2009). Para abordar el resto de problemas que presenta la laparoscopia, ya hace tiempo que se desarrollan sistemas robóticos para asistencia en intervenciones mínimamente invasivas, siendo el más conocido de ellos el sistema Da Vinci, en el que el cirujano maneja una serie de instrumentos robotizados mediante teleoperación, consiguiendo aumentar la precisión en la intervención. Además de sistemas teleoperados, también se está dando el paso hacia la robótica colaborativa, en la que los robots permite reducir costes de personal mediante la sustitución del mismo (Beuss et al., 2021). Ya en 2014 había comenzado la investigación sobre cómo dotar de mayor autonomía a los sistemas robóticos de manejo de cámara en cirugía laparoscópica (Pandya et al., 2014; Capolei et al., 2017).

De la utilidad de usar robots en la asistencia a la cirugía mínimamente invasiva surge la necesidad de contar con entornos de trabajo o frameworks que permitan desarrollar la investigación en este tipo de sistemas. Ya existen investigaciones sobre este tipo de frameworks, como el estudio, por ejemplo, de un entorno de trabajo que puede asistir en intervenciones de cirugía transanal, para la extirpación de la vesícula biliar u operaciones de ruptura de hernia, centrándose en el análisis del espacio de trabajo y en la ergonomía (Hatzfeld et al., 2017). Este framework, junto con otros como (Geoghegan et al., 2019), que aborda el estudio de una plataforma de entrenamiento para cirugía transanal robotizada, emplean soluciones de robótica quirúrgica particularizada a intervenciones realizadas a través de orificios naturales.

En (Kundrat et al., 2021) se hace un estudio más particular, donde se analiza la implementación de una plataforma robótica para cirugía endovascular compatible con técnicas de imagen por resonancia magnética no ionizante evaluada mediante estudios de canulación. Otros desarrollan robots para intervenciones relacionadas con la próstata, como la plataforma ASTRO, que desarrolla un sistema de catéter robotizado para la cirugía de hiperplasia prostática benigna asistida por láser (Russo et al., 2015).

También existen otros trabajos más generalistas en cuanto al tipo de intervención que permiten, como el desarrollado por el proyecto SARAS. En (Setti et al., 2019) se describe esta plataforma para robótica médica, evaluada para la realización de prostatectomía radical, considerada el modelo de referencia en cirugía mínimamente invasiva robótica (Hu et al., 2009) (iData Research, 2018). La arquitectura de la plataforma SARAS, organizada mediante módulos, permite la captación de datos tanto de vídeo como cinemáticos del robot para su utilización en algoritmos de control, implementables mediante el módulo cognitivo de la plataforma. Además, esta plataforma utili-

za ROS (*Robot Operating System*) para implementar los distintos módulos que la componen (Leporini et al., 2020) (SARAS-Project, 2020).

Casi todos los frameworks mencionados en los párrafos previos se especializan en un tipo de procedimiento de cirugía mínimamente invasiva en concreto. Se centran en un escenario quirúrgico fijo, que por tanto no tiene la necesidad de presentar cambios ni en la cinemática del robot, ni en el espacio de trabajo en el que este opera. El trabajo desarrollado en SARAS es el que más flexibilidad aporta, ya que presenta un brazo robótico genérico que puede ser utilizado en más de un tipo de intervención diferente. Todos estos trabajos son implementados en set-ups físicos, y utilizan dummies de partes analógicas de forma que puedan evaluar al robot manipulador o al instrumento robotizado que implementen.

La propuesta de este trabajo se caracteriza por aportar flexibilidad y generalidad con un framework basado en la integración en un mismo sistema de un brazo manipulador, al que se le puede acoplar el instrumental necesario, y de un simulador de cirugía laparoscópica. La presencia de este simulador permite la utilización del framework para evaluar distintos algoritmos desarrollados en el grupo de investigación, sobre los diferentes tipos de intervenciones de cirugía mínimamente invasiva que el simulador permite ejecutar. Además, cuenta con una interfaz de programación lo suficientemente genérica para que cualquier miembro del grupo de investigación pueda desarrollar su trabajo de forma modular e independiente de la implementación del framework, permitiendo así avanzar en el campo de la robótica quirúrgica del paradigma de la teleoperación a la robótica colaborativa. La implementación del framework tiene en cuenta aspectos ergonómicos y particulariza la cinemática del robot utilizado, de forma que a este se le puedan especificar restricciones espaciales para que su volumen de trabajo interfiera lo menos posible con las intervenciones. Este espacio de trabajo también cuenta con capacidades de adquisición de imágenes y vídeo, que podrán ser utilizadas en la construcción de datasets útiles para entrenar algoritmos inteligentes en un futuro.

Este artículo se organiza de la siguiente forma. La sección 2 mostrada a continuación describe el sistema ciber-físico planteado, los problemas que su desarrollo plantea, y la arquitectura funcional diseñada para este framework, así como el modelo geométrico de la tarea a desarrollar por el robot. La sección 3 describe el Modelo Cinemático Inverso elegido para implementar en el brazo robótico redundante del framework, que es la parte más significativa del mismo y las restricciones bajo las cuales va a funcionar este modelo. La sección 4 describe cómo este modelo cinemático ha sido modificado para que cumpla las restricciones funcionales especificadas. La sección 5 describe los distintos subsistemas que componen la implementación framework, y la implementación de una aplicación sencilla que hace uso del framework desarrollado, y que pone de manifiesto las facilidades que presenta. Finalmente, la sección 6 aborda las conclusiones sobre el desarrollo de este trabajo y las líneas futuras propuestas.

2. Descripción del sistema

En esta sección se trata la descripción del sistema planteado en este artículo como entorno de trabajo ciber-físico para

responder a las motivaciones discutidas en la introducción. En la subsección 2.1 se trata una descripción del problema de la implementación de este framework, analizando las especificaciones funcionales objetivo de su desarrollo, los elementos que forman el entorno de trabajo y la arquitectura funcional empleada. En la subsección 2.2 se presentan los sistemas de referencia utilizados en el framework, y cómo se define la pose objetivo a alcanzar por el robot dentro de estos sistemas, mediante un modelo geométrico de dicha tarea.

2.1. Descripción del problema y características del espacio de trabajo

A la hora de desarrollar este trabajo, se han tenido como objetivo la consecución de una serie de requerimientos funcionales. Por un lado, el sistema de control del robot debe ser capaz de extraer información del simulador en forma de vídeo, y por otro lado, el robot debe poder manipular el instrumental del simulador para así poder realizar las tareas normalmente asignadas al asistente durante la intervención quirúrgica. El modelo cinemático del robot implementado en el framework debe ofrecer la posibilidad de introducir las restricciones necesarias para cumplir con los aspectos ergonómicos de la intervención, interfiriendo en esta lo menos posible, y esto solo será posible utilizando un brazo robótico con cinemática redundante, ofreciendo este muchas más posibilidades de configuración articular. Además se deben crear con MATLAB y ROS una serie de funciones de manejo del robot que implementen esta cinemática. Estas deben ser lo suficientemente genéricas y de alto nivel para que el desarrollador que las utilice no tenga por qué conocer los aspectos técnicos de su implementación.

Para cumplir con las especificaciones funcionales citadas en el párrafo anterior se propone la implementación de un sistema ciber-físico tal y como ilustra la Figura 1, que muestra el funcionamiento de los elementos del sistema. Estos elementos son los siguientes. Un cirujano controlará los mandos correspondientes a las herramientas del simulador de laparoscopia, que permiten, por ejemplo, abrir y cerrar pinzas o realizar cortes dentro de la cavidad del abdomen. El mando de la cámara o endoscopio no será controlado por el cirujano sino que, emulando lo que haría un asistente, será controlado mediante un robot manipulador. A pesar del reparto de tareas seleccionado, el robot podría manejar eventualmente también alguna de las herramientas laparoscópicas que actualmente maneja el cirujano, aunque eso supondría la creación de un efector final más complejo capaz de abrir y cerrar las pinzas de forma automática. Las imágenes simuladas que debería captar la cámara se muestran en la pantalla del simulador, de forma que el cirujano pueda realizar la intervención.

También de la Figura 1 se pueden inferir las ubicaciones de todos los elementos físicos dentro del quirófano. El cirujano queda directamente al frente del simulador de laparoscopia, mirando la pantalla que se encuentra sobre este, y manejando el instrumental laparoscópico. El robot queda a un lado del simulador, con su parte frontal apuntando a este, de forma que pueda alcanzar el mando de la cámara. Esta forma de disponer los elementos del quirófano presenta la mayor ergonomía para el cirujano, que se encuentra en la posición para la cuál fue diseñado el simulador, y de forma que el robot quede a un lado en la posición en la que un asistente se ubicaría en un quirófano.

Concretamente para cumplir con las especificaciones funcionales del sistema, se ha decidido utilizar un brazo manipulador WAM de la marca Barrett y un simulador quirúrgico Laparoscopy VR, de CAE Healthcare. El brazo robótico seleccionado tiene 7 grados de libertad, y por tanto cuenta con cinemática redundante, lo que le permite abordar una misma pose de destino para el efector final desde infinitas opciones para su configuración articular, aportando la flexibilidad necesaria para cumplir con los aspectos ergonómicos necesarios por el framework. Esta característica ha sido aprovechada implementando una solución analítica para el Modelo Cinemático Inverso (MCI) del robot que permite particularizar y elegir, de las infinitas posibilidades de configuración del manipulador para alcanzar la pose, la que permita respetar las restricciones introducidas por los aspectos ergonómicos del sistema, así como las restricciones físicas impuestas a este, tanto por las propias limitaciones de las articulaciones del robot como por las características del espacio de trabajo. El simulador de laparoscopia integrado en el sistema cuenta con una amplia selección de diferentes tipos de intervenciones, que aportan al framework la generalidad y flexibilidad suficientes para evaluar los distintos trabajos de investigación que se desarrollen en el laboratorio y que se implementen en este entorno de trabajo.

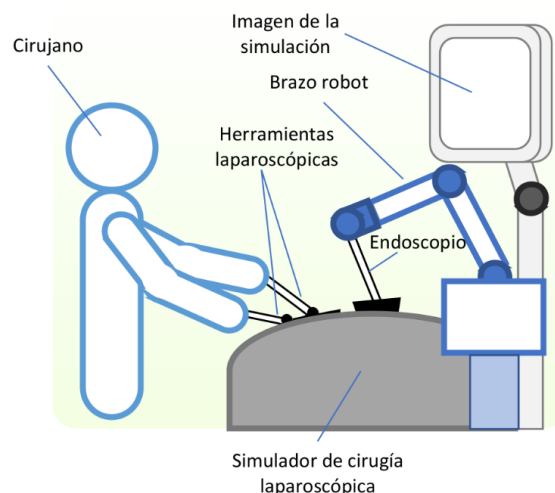


Figura 1: Disposición de los elementos del entorno de trabajo.

La arquitectura funcional del entorno de trabajo planteado se ha dividido en tres bloques funcionales principales, tal y como ilustra la Figura 2. El primero de ellos, denominado *Adquisición de vídeo*, es el encargado de adquirir la señal de vídeo del simulador. Para conseguir esta señal hay que acceder a la interfaz gráfica del simulador y lograr capturar la imagen, de forma que pueda ser procesada por un sistema externo. El siguiente bloque, denominado *Brazo manipulador*, es el encargado de actuar sobre el mando de la cámara del simulador mediante la utilización de un brazo manipulador robótico. Este bloque, a su vez se divide en dos partes. La primera de ellas es el propio sistema físico del robot, *Robot*, cuyo controlador implementa una interfaz de comunicación basada en ROS. La segunda parte de este bloque, *MCI*, es la formada por el script de MATLAB que implementa el Modelo Cinemático Inverso del robot, y que permite particularizar la solución del espacio articular del ro-

bot de la forma más conveniente para cumplir con las especificaciones funcionales. El último bloque del entorno de trabajo, *Integración en MATLAB* se trata de la integración mediante el uso de MATLAB de los otros dos subsistemas. A través de este bloque de integración se puede acceder a los datos de vídeo extraídos del simulador y también se pueda actuar sobre el simulador utilizando una serie de comandos enviados al robot. Será este bloque el que también actúe como interfaz y se encargue de permitir la comunicación con los posibles sistemas externos que pretendan hacer uso de este framework en un futuro y con el propio cirujano que también se encuentre actuando sobre el simulador.

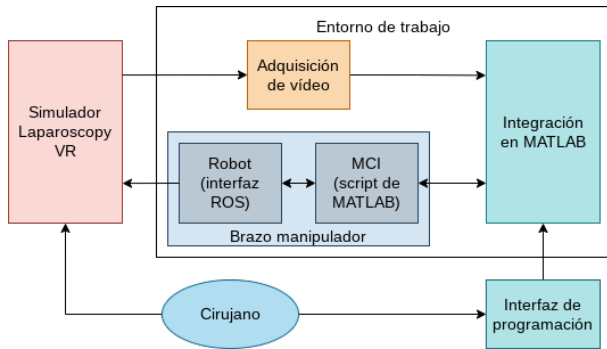


Figura 2: Diagrama de bloques del framework planteado.

2.2. Modelo geométrico de la tarea

La Figura 3 presenta un esquema de cómo se abordaría el control de la cámara mediante el robot, y establece los nombres de algunos puntos y de los sistemas de referencia utilizados en este artículo, que son:

- $\{robot\}$: sistema de referencia de la base del robot.
- $\{GC\}$: sistema de referencia de la circunferencia generadora de todas las posiciones posibles del punto C del codo.
- $\{cam\}$: sistema de referencia del mando de la cámara, que será la pose objetivo del robot.
- $\{fulcro\}$: sistema de referencia asociado al punto de inserción del laparoscopio en el simulador, que actúa como punto de fulcro del mando de la cámara.
- DW_{pos} : posición deseada para la muñeca del robot.

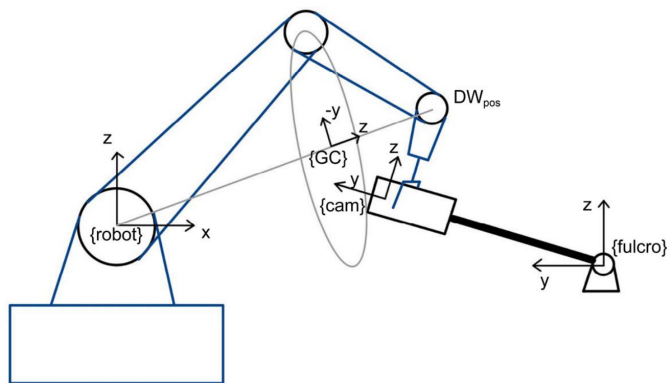


Figura 3: Esquema del modelo geométrico de la tarea.

Dado que la tarea que el robot debe realizar en el sistema ciber-físico planteado se trata del manejo del mando de la cámara, primero se debe modelar cinemáticamente. Analizando el mando de la cámara, se obtiene que esta posee 4 grados de libertad. Su movimiento se puede modelar de la siguiente manera. Primero, una rotación en el eje X del sistema $\{fulcro\}$, de la Figura 3, resultando en la elevación de la cámara; seguida por otra rotación en el eje Z de la cámara, resultando en el desplazamiento horizontal de la cámara; a lo cual le sigue otra rotación en el eje Y de la cámara, y un desplazamiento en el mismo eje Y , resultando en el desplazamiento del mando respecto del punto de pivote. De todos los movimientos descritos el único que no se va a considerar en el modelo usado en este trabajo es la rotación en el eje Y de la cámara, ya que no resultaría en un cambio del campo de visión, sino en una rotación del mismo, no añadiendo nueva información.

Si se consideran como parámetros los ángulos de rotación vertical y horizontal, α y β , así como el desplazamiento de la cámara respecto al punto de fulcro, ρ , la transformada que relaciona la pose final de la cámara respecto al pivote viene dada por la ecuación:

$${}_{fulcro}T_{cam} = rot_X(\alpha) \cdot rot_Z(\beta) \cdot despy(\rho) \quad (1)$$

Los ángulos asociados a estos movimientos se pueden ver en la Figura 4, que muestra la representación de los mismos sobre el sistema de referencia del punto de fulcro, y donde se puede ver cómo quedan situados los ejes del sistema de la cámara una vez realizados los movimientos de (1).

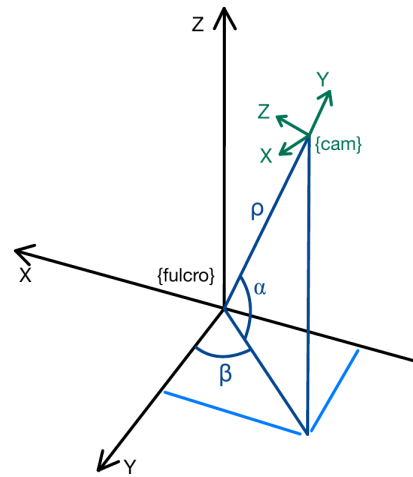


Figura 4: Esquema de relación del sistema de referencia $\{cam\}$ respecto a $\{fulcro\}$, mediante los parámetros α , β y ρ .

Experimentalmente se han encontrado los límites de valores mínimos y máximos para los parámetros α , β y ρ recogidos en la Tabla 1.

Tabla 1: Límites para los parámetros de la transformada de la cámara

Parámetro	Mínimo	Máximo
α	15°	80°
β	-30°	30°
ρ	14cm	22cm

El espacio de trabajo del mando de la cámara definido por el modelo geométrico de la tarea y por los límites para sus parámetros recogidos en la Tabla 1 debe ser contenido por el espacio de trabajo del manipulador, de forma que este pueda situar el mando de la cámara en cualquier posición posible. Cabe destacar que el estudio desarrollado en esta subsección está particularizado al manejo del mando de cámara del simulador. Para implementar el manejo de cualquiera de las otras herramientas laparoscópicas, el estudio seguiría la misma metodología de establecimiento de sistemas de referencia y parámetros definitorios, pero la transformada resultante podría ser diferente.

3. Modelo Cinemático Inverso del brazo manipulador

En esta sección se aborda el Modelo Cinemático Inverso utilizado para el brazo manipulador WAM de la marca Barrett. Este modelo cinemático debe permitir implementar las restricciones físicas descritas en la subsección 3.1, y su funcionamiento se describe en la subsección 3.2.

3.1. Restricciones para el brazo robótico

Para integrar el robot en el sistema ciber-físico se deben seguir una serie de restricciones funcionales y de seguridad:

- Inclusión del espacio de trabajo externo de la cámara dentro del espacio de trabajo del robot, como ilustra el diagrama de la Figura 1. Esto resultará en que el robot pueda alcanzar todas las poses posibles a las cuales debe tener acceso la cámara, asegurando que la tarea se puede resolver con el robot propuesto. El cumplimiento de este criterio resultará en la posición del robot respecto del punto de pivote de la cámara.
- Cumplimiento de los criterios de no oclusión y de no colisión. El de no oclusión consiste en que el elemento externo que se va a añadir al simulador, el brazo robótico, no debe estorbar al cirujano a la hora de adquirir información de la pantalla, por lo que el codo del robot no deberá tapar su línea de visión. El criterio de no colisión evitará que el robot choque con el cirujano o con el simulador.
- Cumplimiento del criterio de mínima interferencia. Según este criterio, el robot deberá interferir lo menos posible con el cirujano a la hora de controlar el resto de mandos del simulador, evitando en la medida de lo posible que su cuerpo entre en el espacio de trabajo del resto de herramientas laparoscópicas.

3.2. Funcionamiento del Modelo Cinemático Inverso

En (Singh and Claassens, 2010) se resuelve el Modelo Cinemático Inverso para el brazo robótico WAM de forma analítica, cuyo esquema se puede ver en la Figura 5, disponible en (Barrett-Technology, 2011), y que muestra los 7 grados de libertad del manipulador, donde los 4 primeros permiten posicionar la muñeca y los 3 últimos, orientarla. Este modelo de brazo robótico posee muñeca esférica, lo que permite resolver el Modelo Cinemático Inverso mediante desacoplo cinemático.

En (Singh and Claassens, 2010) se desarrolla que, conociendo la posición de la muñeca, y al contar con 4 grados de libertad

para posicionarla, hay infinitas configuraciones articulares que la pueden alcanzar. El conjunto de todas estas configuraciones hace que las posibles posiciones del codo formen una circunferencia contenida en un plano perpendicular al vector que une la base del robot, $\{robot\}$, con la muñeca, $\{DW_{pos}\}$, y definido por una distancia a la base y un radio concretos. A este plano se le asocia el sistema de referencia $\{GC\}$, y esta circunferencia se llama circunferencia generadora. También se considera que, al haber un offset entre la articulación 4 y las articulaciones adyacentes, 3 y 5, el codo no está definido por un solo punto C , sino también por los puntos LJ (Lower Joint) y UJ (Upper Joint). Por tanto, no habrá una sola circunferencia generadora, sino tres, una para cada uno de estos puntos. Para situar la circunferencia generadora de cada punto del codo en el espacio se define una misma matriz de rotación, R_{norm} .

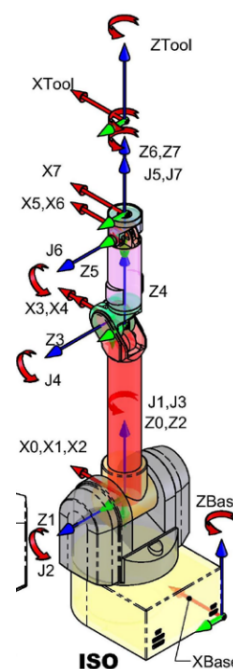


Figura 5: Representación esquemática del brazo robótico WAM de Barrett (Barrett-Technology, 2011).

Para elegir una posición en concreto para el codo dentro de todas las posiciones posibles comprendidas en las circunferencias generadoras se introduce el parámetro ϕ , que representa el ángulo en el que se situará el codo dentro del plano de dicha circunferencia. Este parámetro representa una cuarta restricción para el problema de la posición, que junto con las tres restricciones dadas por las componentes de DW_{pos} , permiten obtener los valores de las 4 primeras articulaciones del robot, pasando de un conjunto de infinitas posibles posiciones para el codo a una en concreto. De esta forma, dado el ángulo ϕ y dados los parámetros definitorios de los círculos generadores, se podrán calcular las posiciones de todos los puntos del codo, C , LJ y UJ .

Una vez se cuente con las posiciones de estos tres puntos, para calcular la configuración articular del robot que da la solución al problema de la cinemática inversa solamente hay que aplicar las fórmulas recogidas en (Singh and Claassens, 2010). En forma de pseudocódigo, la implementación de este algoritmo de cinemática inversa quedaría de la siguiente manera:

```

función  $[\theta_i] = mci\_wam(T_{objetivo})$ 
   $DW_{pos} = desacoplo\_cinematico()$ 
   $d = distancia(DW_{pos}, base)$ 
   $R_{norm} = rotacion\_CG(DW_{pos})$ 
   $\phi = seleccion\_phi(R_{norm}, DW_{pos})$ 
   $[C, UJ, LJ] = CG(\phi, d, R_{norm})$ 
  return  $[\theta_i] = mci(C, UJ, LJ)$ 
final de función

```

La función $[\theta_i] = mci_wam(T_{objetivo})$ que implementa el Modelo Cinemático Inverso toma como parámetro de entrada la pose objetivo, y devuelve el vector articular que da la solución a esta entrada. La primera función que este pseudoalgoritmo ejecuta $DW_{pos} = desacoplo_cinematico()$ y devuelve la posición de la muñeca esférica mediante el proceso estandarizado de desacoplo cinemático. Una vez se tiene la posición de la muñeca, se calcula la distancia de esta a la base mediante la función $d = distancia(DW_{pos}, base)$, y también se calcula la matriz de rotación R_{norm} mediante la función $rotacion_CG(DW_{pos})$. Una vez se tiene esta matriz, se utiliza la función $\phi = seleccion_phi(R_{norm}, DW_{pos})$ para seleccionar el parámetro ϕ más conveniente para la aplicación en concreto del entorno de trabajo analizado en este artículo. Teniendo todos estos parámetros, ya se pueden calcular los puntos definitorios del codo mediante la función $[C, UJ, LJ] = CG(\phi, d, R_{norm})$, y una vez se conocen sus posiciones, se puede calcular la solución al Modelo Cinemático Inverso, correspondiente con la función return $[\theta_i] = mci(C, UJ, LJ)$ del pseudoalgoritmo, que devuelve el parámetro de salida del modelo cinemático.

Para particularizar el Modelo Cinemático Inverso del robot WAM de (Singh and Claassens, 2010) al framework de este trabajo y a las restricciones impuestas por las especificaciones funcionales, las funciones del pseudoalgoritmo más significativas y en los que este artículo aporta una solución novedosa son las correspondientes con la definición de la matriz de rotación de los círculos generadores, $R_{norm} = rotacion_CG(DW_{pos})$, y con la selección del parámetro ϕ más adecuado para este caso en concreto, $\phi = seleccion_phi(R_{norm}, DW_{pos})$.

4. Particularización del Modelo Cinemático Inverso a las especificaciones funcionales del entorno de trabajo

En esta sección se analizan las particularizaciones realizadas el Modelo Cinemático Inverso del robot de forma que se le puedan introducir las restricciones físicas comentadas en la subsección 3.1. La subsección 4.1 trata sobre el proceso de definición de la matriz de rotación que sitúa los círculos generadores en el espacio, y la subsección 4.2 expone los cálculos y el proceso lógico de selección del ángulo ϕ que permite escoger la configuración articular más conveniente entre las infinitas posibilidades existentes.

4.1. Definición de R_{norm}

La definición de la matriz R_{norm} se corresponde con el paso del pseudocódigo del algoritmo que los movimientos utilizados para situar las circunferencias generadoras en el espacio, es decir, la función $rotacion_CG(DW_{pos})$. En (Singh and Claassens, 2010) se propone considerar primero que las circunferencias

generadoras se supongan en un plano paralelo al plano $X - Y$ de la base del robot, desplazado en el eje Z del mismo sistema base la distancia ya comentada anteriormente que los define. Para situar dichas circunferencias generadoras perpendicularmente al vector que une la base con la muñeca, se aplica una serie de rotaciones respecto al sistema de la base del robot, y que se definen como las rotaciones necesarias para mover el punto de la muñeca a una posición equivalente si estuviera a la misma distancia d pero situado en el eje Z del robot. En el caso de este trabajo se ha decidido particularizar este movimiento realizando una rotación sobre el eje X de α seguida por otra rotación en el eje Z de β :

$$R_{norm} = rot_X(\alpha_{norm}) \cdot rot_Z(\beta_{norm}), \quad (2)$$

donde

$$\alpha_{norm} = \arccos\left(\frac{DW_{pos,z}}{d}\right) \quad (3)$$

y

$$\beta_{norm} = \text{atan2}\left(\frac{DW_{pos,1}}{L_{norm}}, \frac{DW_{pos,1}}{L_{norm}}\right), \quad (4)$$

siendo L_{norm} el valor de la proyección de DW_{pos} (la posición deseada para la muñeca, obtenida mediante desacoplo cinemático) sobre el plano $X - Y$ del sistema de referencia del robot. El parámetro d se trata de la distancia desde el origen del sistema de referencia de la base del robot, $\{robot\}$, a la posición deseada para la muñeca, DW_{pos} . La Figura 6 muestra un diagrama con los ángulos α y β utilizados los movimientos necesarios para obtener R_{norm} respecto a los ejes del sistema de referencia de la base del robot, representando también de forma esquemática los primeros eslabones del robot, en azul, y la circunferencia generadora del punto C , en amarillo. También se muestran las ubicaciones de los puntos C , LJ y UJ del codo en el esquema de los eslabones del robot.

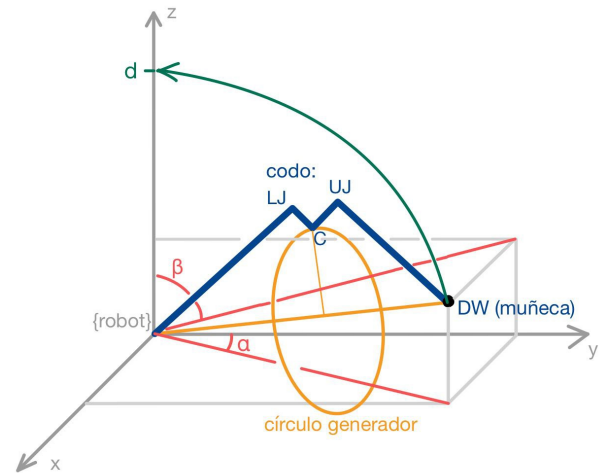


Figura 6: Diagrama de movimientos de la matriz de rotación R_{norm} .

La elección de este conjunto de movimientos en particular se ha realizado porque tiene la particularidad de que sea cual sea la pose final a la que se quiera llegar con el brazo robótico, el valor para el ángulo ϕ que dará lugar a la posición de codo más alta será el valor $\phi = -\pi/2 \text{ rad}$. Esto es realmente útil a la hora de seleccionar el valor de ϕ más adecuado para que se cumplan las restricciones funcionales especificadas en la subsección 3.1.

4.2. Selección de ϕ

La selección del ángulo ϕ se corresponde con el cuarto paso del pseudocódigo que define el algoritmo del Modelo Cinemático Inverso implementado, es decir, la función *seleccion_phi*(R_{norm} , DW_{pos}). En (Singh and Claassens, 2010) no se propone ningún criterio para seleccionar el parámetro ϕ , sino que este se trata como un parámetro de entrada junto con la pose objetivo T_{obj} . En el marco de trabajo desarrollado, T_{obj} viene dado por la pose objetivo marcada por la cámara, y esta es conocida, ya la subsección 2.2 desarrolla el modelo geométrico de dicha tarea. Sin embargo, ϕ , que situará el codo en el espacio, no es conocido a priori, sino que es el sistema desarrollado el que se encarga de encontrar el valor más adecuado para cumplir con las restricciones presentadas en la subsección 3.1.

Cabe destacar que la matriz de rotación R_{norm} elegida en este trabajo siempre situará las circunferencias generadoras de forma que el punto más alto del codo, el que maximiza la coordenada Z respecto de la base del robot, se alcance para $\phi = -\frac{\pi}{2} rad$, tal y como concluye la subsección 4.1. La Figura 7 muestra la circunferencia generadora del punto C, desde una perspectiva perpendicular al plano que la contiene, vista desde el punto en que se sitúa la muñeca del robot, junto con su sistema de referencia, y marcando con un asterisco rojo el punto más alto en Z.

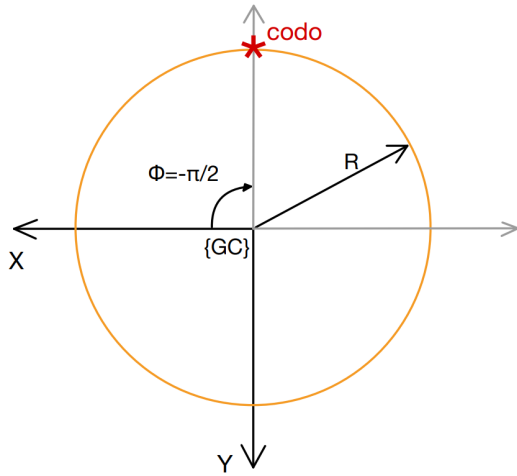


Figura 7: Situación de C_{rot} tras el giro.

En la Figura 8 también se marcan con asteriscos rojos los puntos más altos de las circunferencias generadoras de los tres puntos del codo. Esta última figura muestra las circunferencias en el espacio 3D para una pose objetivo de la cámara, marcada con un asterisco azul, y también muestra una representación esquemática de los primeros eslabones del robot, hasta llegar a la muñeca, marcada como DW_{pos} . El punto donde quedaría la base está marcado por un asterisco negro. Por la configuración del setup ya descrita, se sabe que el punto DW_{pos} quedará al frente del robot. Por lo tanto, para que el codo no se acerque demasiado al cirujano, se ha impuesto una primera restricción al ángulo $\phi_{max} = -\frac{\pi}{2} rad$. De igual manera, para evitar que el codo colisione con el cuerpo del simulador, se ha impuesto una segunda restricción al ángulo tal que $\phi_{min} = -\frac{9\pi}{5} rad$. De esta

forma se tiene un rango para ϕ inicial de:

$$\phi \in \left[-\frac{9\pi}{5}, -\frac{\pi}{2} \right] \quad (5)$$

Este rango asegura el cumplimiento del criterio de no colisión para cualquier pose T_{obj} deseada.

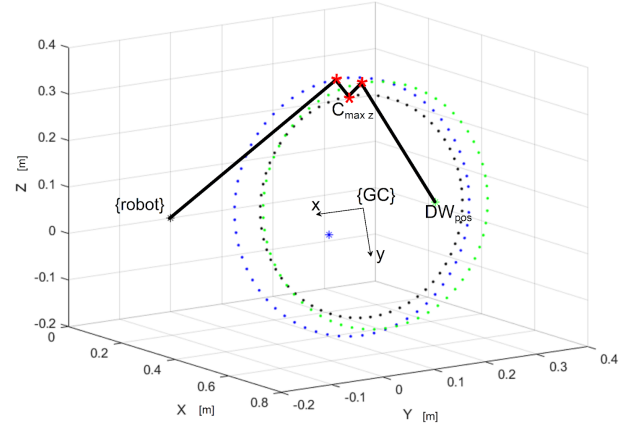


Figura 8: Situación de C_{rot} tras el giro, vista en perspectiva.

En (Singh and Claassens, 2010) también se analiza la limitación del ángulo ϕ de forma que se cumplan las limitaciones de ángulos máximos y mínimos para las articulaciones. En concreto esta limitación se discute para las articulaciones 2 y 6, especialmente problemáticas por su restringido rango de valores. Para θ_2 la limitación viene dada por la inecuación:

$$((C_{rot}(R_{LJ}, D_{LJ}, \phi) \cdot R_{norm})_z / d_3) \geq \cos(\theta_{2L}) = \cos(\theta_{2U}), \quad (6)$$

donde R_{LJ} y D_{LJ} son, respectivamente, el radio y la distancia de la circunferencia generadora del punto LJ , θ_{2L} y θ_{2U} son los ángulos límite de la articulación 2, y d_3 se trata del parámetro Denavit–Hartenberg de la cinemática directa del WAM. Desarrollando la expresión anterior se obtiene la inecuación:

$$R_{LJ} \cos(\phi) \cdot r_{13} + R_{LJ} \cos(\phi) \cdot r_{23} \geq \cos(\theta_{2L}) - D_{LJ} r_{33} \quad (7)$$

Para θ_6 la limitación viene dada por la inecuación:

$${}^{UJ}V_{WP} \odot TR_z \geq \cos(\theta_{6L}) = \cos(\theta_{6U}), \quad (8)$$

donde

$${}^{UJ}V_{WP} = \frac{C_{rot}(R_{UJ}, D_{UJ}, \phi) \cdot R_{norm} - DW_{pos}}{d_5}, \quad (9)$$

siendo R_{UJ} el radio y D_{UJ} la distancia de la circunferencia generadora del punto UJ , d_5 el parámetro Denavit–Hartenberg de la cinemática directa del WAM, TR_z la tercera columna de la matriz de rotación de la pose objetivo T_{obj} , y θ_{6L} y θ_{6U} los ángulos mínimo y máximo permitidos para la articulación 6. Desarrollando estas expresiones, la inecuación queda:

$$\begin{aligned} & R_{UJ} \cdot (TR_{z1}r_{11} + TR_{z2}r_{12} + TR_{z3}r_{13}) \cdot \cos(\phi) + \\ & + R_{UJ} \cdot (TR_{z1}r_{21} + TR_{z2}r_{22} + TR_{z3}r_{23}) \cdot \sin(\phi) \geq \\ & \geq d_5 \cos(\theta_{6L}) - TR_{z1} \cdot (D_{UJ}r_{31} - DW_{pos,1}) - \\ & - TR_{z2} \cdot (D_{UJ}r_{32} - DW_{pos,2}) - \\ & - TR_{z3} \cdot (D_{UJ}r_{33} - DW_{pos,3}) \end{aligned} \quad (10)$$

Además de los rangos del ángulo ϕ que permiten respetar los límites articulares del robot, existe la posibilidad de utilizar la posición z del punto LJ para asegurar un límite superior a la altura del codo que asegure el cumplimiento de la restricción de no oclusión, de forma que el codo no tape la línea de visión del cirujano. Para imponer esta restricción se parte de una modificación de (6), que puede interpretarse como una limitación inferior para la altura de LJ . Dicha modificación, que impone una limitación superior, queda de la forma:

$$R_{LJ} \cos(\phi) \cdot r_{13} + R_{LJ} \cos(\phi) \cdot r_{23} \leq H - D_{LJ} r_{33}, \quad (11)$$

donde H se trata de la altura máxima para el codo, referenciada al sistema de la base del manipulador, $\{robot\}$, tal que se cumple el criterio de no oclusión. Para seleccionar el valor de H hay que tener en cuenta, principalmente, los criterios de ergonomía propios de cada cirujano. Para empezar, hay que tener en cuenta que el simulador cuenta con una pantalla con regulación en altura, que cada cirujano puede situar en la posición que encuentre más ergonómica. Una vez la altura de la pantalla se encuentre fija, se puede seleccionar el parámetro H de altura máxima del codo del robot de forma que cumpla el criterio de no oclusión y permita al cirujano operar de una forma cómoda, y por lo tanto será diferente para cada individuo. Sin embargo, hay que tener en cuenta que se ha encontrado mediante un proceso de búsqueda numérico que el valor mínimo para H para el cual el robot es capaz de abordar todas las poses del espacio de trabajo del mando de la cámara se trata de $H_{min} = 23cm$ sobre el hombro del robot.

A la hora de resolver las inecuaciones planteadas, se conoce que tanto (7) como (10) y (11) son de la forma:

$$A \cdot \cos(\phi) - B \cdot \sin(\phi) \geq C, \quad (12)$$

y dada la equivalencia

$$A \cdot \cos(\phi) - B \cdot \sin(\phi) = R \cdot \cos(\phi + \alpha), \quad (13)$$

donde $R = \sqrt{A^2 + B^2}$ y $\alpha = \text{atan2}(B, A)$, siendo atan2 la función de MATLAB que devuelve ángulos en los cuatro cuadrantes, se puede resolver la inecuación. Si esta no tuviera solución, por no cumplir $\|C/R\| \leq 1$, o bien todos los ángulos posibles para ϕ cumplen con la restricción aplicada o bien ninguno lo cumple. En caso de que sí se cumpliera $\|C/R\| \leq 1$, se debe estudiar en qué rango de los delimitados por las dos soluciones se encuentra la solución de la inecuación planteada.

Los ángulos límite de los rangos que cumplen con la inecuación considerada vienen dados por:

$$\begin{aligned} \phi_{lim,1} &= \arccos(C/R) - \alpha, \\ \phi_{lim,2} &= -\arccos(C/R) - \alpha \end{aligned} \quad (14)$$

Una vez identificados los parámetros A , B y C para cada una de las inecuaciones resultantes de limitar ϕ según los rangos de θ_2 y θ_6 , y la altura H , y habiendo encontrado los rangos de ángulos para los cuales estas inecuaciones se cumplen, solo queda combinar todos los rangos encontrados para cada limitación, calculando la intersección entre estos. Los rangos utilizados en este artículo son (5) y los resultantes de (7), (10), y (11). El rango resultante de combinar los cuatro anteriores da como resultado el conjunto de posibles valores de ϕ para los cuales la

solución de la cinemática inversa cumple los límites de θ_2 y θ_6 , y con los criterios de no oclusión y no colisión.

$$\begin{aligned} \phi \in \left[-\frac{9\pi}{5}, -\frac{\pi}{2} \right] \cap [\phi_{min,2}, \phi_{max,2}] \cap \\ \cap [\phi_{min,6}, \phi_{max,6}] \cap [\phi_{min,H}, \phi_{max,H}] \end{aligned} \quad (15)$$

De todos los ángulos posibles para ϕ , el que resulta en una configuración más cómoda para manejar el resto de instrumentos del simulador es el más cercano a $\phi = -\pi/2rad$, por dar como resultado la posición del codo del brazo robótico más alta posible, y es el que se utilizará para resolver la cinemática inversa del robot.

Finalmente, la selección de ϕ se puede resumir mediante el siguiente algoritmo:

1. Definición del rango para ϕ de (5), de forma que se cumpla con el criterio de no colisión.
2. Resolución de (7).
3. Obtención del rango de ϕ según la solución anterior, obteniendo el rango que respeta los límites de la articulación 2.
4. Resolución de (10).
5. Obtención del rango de ϕ según la solución anterior, obteniendo el rango que respeta los límites de la articulación 6.
6. Resolución de (11).
7. Obtención del rango de ϕ según la solución anterior, obteniendo el rango que respeta la restricción de no oclusión.
8. Cálculo del rango de ϕ de intersección entre los de los puntos 1, 3, 5 y 7.
9. Selección del ϕ mayor del rango de 6, que es el más cercano a $\phi = -\pi/2$ y por tanto el que deja un mayor espacio para maniobrar el resto de herramientas al cirujano.

5. Implantación y Resultados

En la Figura 9 se encuentra un diagrama con la relación entre todos los elementos utilizados en la implantación del sistema ciber-físico. Como se puede observar el cirujano se relaciona directamente con el simulador y con el PC externo, que cuenta con capacidades de adquisición de vídeo del simulador y de manejo del robot, implementando la interfaz de programación en MATLAB que permite manejar el mando de la cámara. Las funciones que implementan esta interfaz permiten conocer la pose actual del robot, resolver el Modelo Cinemático Inverso del manipulador dada una pose objetivo, y enviar la solución encontrada por este modelo al sistema de control del robot. Este ordenador externo se comunica mediante ROS, a través de un switch, con el PC industrial de control del robot, que cuenta con los drivers ROS del brazo manipulador que permiten utilizar la librería de manejo del mismo, Libbarrett, de forma externa. Esta librería se comunica directamente con las articulaciones del robot mediante Bus CAN, y el robot se encarga de actuar sobre el simulador moviendo el mando de la cámara.

Las siguientes subsecciones describen la implementación desde el punto de vista del diagrama funcional de la Figura 2,

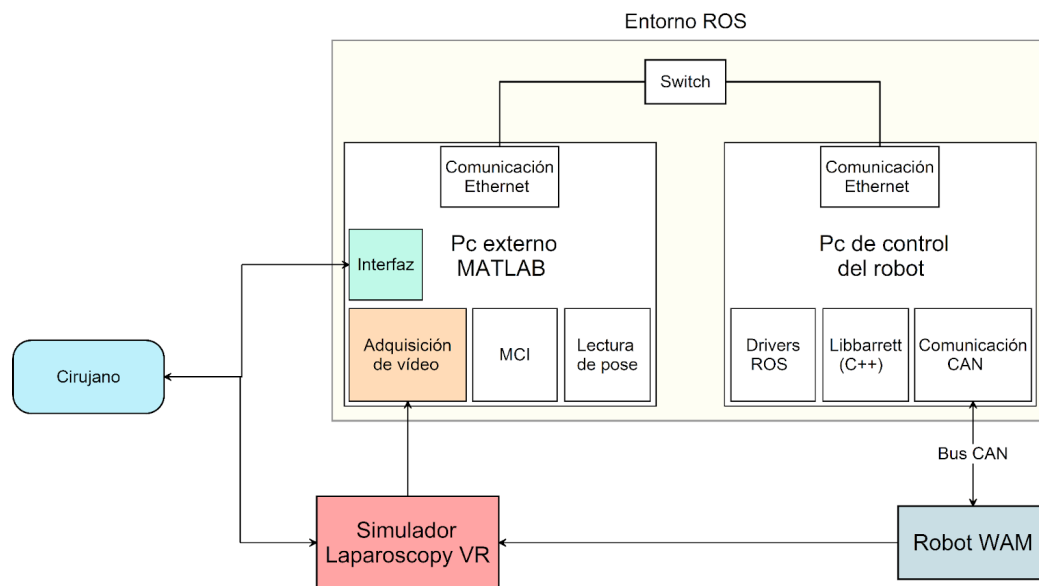


Figura 9: Esquema de interconexión de todos los componentes del framework.

que divide el framework en los bloques de manejo de la cámara con el brazo robótico, adquisición de vídeo, e integración de los dos bloques anteriores en MATLAB, así como un ejemplo de uso del framework mediante el desarrollo de una aplicación sencilla.

5.1. Manejo de cámara con brazo robótico

La solución propuesta en las secciones 3 y 4 para el problema del movimiento del mando de la cámara mediante la utilización del brazo robótico se ha implantado mediante la arquitectura ilustrada en la Figura 9, en la que se pueden distinguir distintos bloques que se corresponden con el subsistema “Brazo manipulador” de la Figura 2.

El primero de estos bloques se trata del código MATLAB que implementa la función que resuelve el problema de la cinemática inversa del robot, dada una pose objetivo, que se corresponde con el bloque “MCI” de la Figura 9. Este código se encarga de calcular, tal y como se ha desarrollado en la sección anterior, la solución más favorable para este problema, teniendo en cuenta las restricciones programadas, y eligiendo de forma autónoma la posición para el codo del robot que permita cumplirlas mediante la selección del parámetro ϕ . El script que lo implementa se ejecuta en un ordenador externo y devuelve la solución en el espacio articular. Además, permite visualizar mediante un modelo en 3D del robot cómo quedará este al aplicar la solución a la cinemática inversa. También se ha implementado la función *wam.pose*, que devuelve la pose actual del robot, y que se corresponde con el bloque “Lectura de pose” de la Figura 9. Una vez se tiene la solución al problema de la cinemática inversa, o cuando se quiere leer la pose del robot, este módulo se comunica mediante ROS (Robot Operating System) con los drivers del robot, a través del switch de comunicación.

El siguiente bloque se trata de los drivers ROS (Robot Operating System) del robot, correspondiente con el bloque “Drivers ROS” de la Figura 9. Estos drivers ROS son un paquete desarrollado por el fabricante del robot, Barrett, accesibles en (Barrett-Technology, 2021), que permiten utilizar distintos co-

mandos del robot mediante topics y servicios. Este bloque actúa como interfaz para acceder a esos comandos, implementados a más bajo nivel con la librería Libbarrett (Barrett-Technology, 2022) para C++ del fabricante, correspondiente con el bloque “Libbarrett (C++)” de la Figura 9, mediante la cual se puede comandar directamente al robot. Todo este software se encuentra instalado en un ordenador industrial con el sistema operativo Ubuntu 12.04 y el framework Xenomai, que dotan al ordenador con capacidades de control en tiempo real.

La interconexión entre estos dos bloques se realiza mediante ROS, tal y como se ha especificado anteriormente. Utilizando la ROS Toolbox de MATLAB, la aplicación que se encuentra en el primer bloque llama a un servicio provisto por los drivers ROS del robot llamado */wam/joint_move*, pasando como parámetro la solución al problema de la cinemática inversa. Al ejecutar este servicio, el ordenador industrial es el que se encarga de aplicar esta consigna para cada articulación, actuando sobre el robot. Los drivers ROS del robot también permiten llamar a otros servicios como, por ejemplo, */wam/go_home*, que al ejecutarse envía al robot a su posición de reposo. Los bloques comentados en los dos párrafos anteriores son los que implementan la capa de software para el movimiento del mando de la cámara del simulador.

Una vez los drivers ROS del robot reciben la consigna para las articulaciones, el comando se transmite mediante bus CAN a la placa de control del robot, el cual está representado como el bloque “Robot WAM” en el diagrama de la Figura 9, y este se encarga de mover el mando de la cámara del simulador de laparoscopia. Para mover este mando se ha diseñado y fabricado una pieza mediante impresión 3D que permite que el movimiento de la cámara se realice de forma solidaria al del efector final del robot. En el diseño de esta pieza se ha puesto especial interés en permitir la conexión y desconexión del mando de la cámara y el robot, para poder devolver al manipulador a su posición de reposo con la frecuencia requerida por este.

5.2. Adquisición de vídeo

La funcionalidad del framework de adquisición de imágenes del simulador, correspondiente con el subsistema “Adquisición de vídeo” de la Figura 2 se ha implementado utilizando un HDMI splitter de forma que de la salida de vídeo del simulador se obtuvieran dos puertos. Uno de ellos se ha conectado a la propia pantalla del simulador para que el cirujano pueda visualizar la simulación de la intervención, mientras que el otro puerto se conecta a una capturadora HDMI. Esta capturadora se conecta mediante USB al mismo PC externo donde se tiene la aplicación de MATLAB para controlar el robot. Ya en este ordenador, la señal de vídeo se trata de dos maneras. Por un lado se conecta directamente a MATLAB, tratándola como si procediera de una webcam convencional, permitiendo utilizar todo el potencial de esta herramienta para visión por computador, como la Image Processing Toolbox. Por otro lado, se hace uso de un nodo ROS, `usb_cam`, accesible en (Varas, 2019), que permite comunicar la señal de vídeo a otros nodos de ROS que pretendan hacer uso de esta.

5.3. Integración

Teniendo tanto la señal de vídeo como el programa para el control de la cámara mediante el robot en el mismo ordenador, ambas en MATLAB, la integración entre los dos subsistemas del framework se realizará precisamente con esta aplicación. Al ser MATLAB un lenguaje ampliamente utilizado en robótica, podrá ser utilizado como interfaz para el sistema ciber-físico, y permitirá desarrollar aplicaciones que cierren el bucle de control entre la adquisición de vídeo y el control de la cámara. Además, por su facilidad de uso, permitirá el prototipado y validación de nuevos algoritmos de forma rápida, pudiendo desacoplar los problemas presentados por dichos algoritmos de los problemas presentados por su implementación en lenguajes más complejos. Sin embargo, el uso de MATLAB supone, entre otras desventajas, la dificultad para implementar algoritmos en tiempo real, por su carácter de lenguaje interpretado y por el entorno de desarrollo asociado, y para desarrollar aplicaciones escalables. Es por eso por lo que los algoritmos desarrollados y ya validados eventualmente se deberán implementar en otros lenguajes más convenientes.

A continuación se presenta un listado de las funciones disponibles para trabajar con este framework e integrarlo en otros sistemas.

- Adquisición y procesamiento de vídeo
 - Image Processing Toolbox
 - Computer Vision Toolbox
- Control del robot de la cámara
 - `mci_wam(T)`: función para la resolución de la cinemática inversa de T para el WAM
 - `send_iksolution_to(T)`: función para el envío de la solución de la cinemática inversa a T del WAM mediante ROS
 - `T = pose_wam`: función para la lectura de la pose del robot
 - Robotics System Toolbox
 - ROS Toolbox

El código MATLAB que implementa las funciones del framework se encuentra en un repositorio en GitHub, accesible mediante el siguiente enlace: <https://github.com/juanmhl/TFG2>

5.4. Aplicaciones del entorno de trabajo

Para demostrar la facilidad de uso del framework como interfaz de desarrollo de aplicaciones de cirugía robótica colaborativa para el simulador de laparoscopia Laparoscopy VR, se ha desarrollado en MATLAB una interfaz de control del mando de la cámara mediante comandos de voz. Para desarrollar dicha aplicación se ha utilizado como base una red neuronal capaz de reconocer comandos de voz, proporcionada por Mathworks en (MathWorks, 2022), y se han integrado las funciones desarrolladas en este trabajo para interactuar con el framework ciber-físico. En trabajos previos como (García-Morales, 2006) se realizan estudios en profundidad sobre las dificultades y limitaciones asociadas al reconocimiento de voz en entornos quirúrgicos. Concretamente, para controlar los parámetros del modelo geométrico de la tarea, la aplicación reconoce las siguientes palabras:

- “Up”, “down”: Incremento o decremento de α .
- “Right”, “left”: Incremento o decremento de β .
- “Off”, “on”: Incremento o decremento de ρ .

Mediante el uso de estos comandos de voz, un cirujano manejando el instrumental quirúrgico del simulador podría manejar también el mando de la cámara sin necesidad de extraer ni soltar ninguno de los otros instrumentos, y sin la necesidad de un asistente específico para el movimiento del endoscopio. La Figura 10 muestra una fotografía de un operario utilizando el simulador y dando órdenes de movimiento a la cámara mediante comandos de voz.

El funcionamiento simplificado de la aplicación es el siguiente. La aplicación comienza inicializando parámetros y variables necesarias para el manejo del entorno de trabajo. Después, comienza el bucle de ejecución del programa. El primer paso consiste en la utilización de un modelo de inteligencia artificial basado en una red neuronal convolucional sencilla. A este modelo se le alimenta el micrófono del sistema, del cual extrae la señal de audio y calcula su espectrograma, que consiste en la representación del espectro de audio de una ventana de tiempo de la señal en forma de matriz. El modelo utilizado está entrenado para detectar en el espectrograma una serie de patrones correspondientes con palabras concretas en inglés. Toda la funcionalidad de reconocimiento de voz se encuentra encapsulada en una función. Si el modelo no identifica ninguna palabra en los 5 segundos durante los cuales se ejecuta, o reconoce una palabra que no corresponde a ningún comando conocido, se vuelve a ejecutar la función de detección de palabras durante otros 5 segundos. Si, en cambio, sí detecta un comando conocido, se aplica la acción asociada al mismo, variando los parámetros del modelo geométrico de la cámara. Con estos nuevos parámetros se calcula una nueva pose T para el robot, y se envía mediante el uso de la función `send_iksolution_to(T)`, que internamente hace uso del framework ROS y de la función para resolución del Modelo Cinemático Inverso `mci_wam`. Una vez se envía este nuevo comando al robot, se procede a volver a llamar a la función de reconocimiento de palabras durante otros 5 segundos.



Figura 10: Fotografía de un operario utilizando el simulador y dando órdenes de movimiento a la cámara mediante comandos de voz.

El pseudocódigo que define la aplicación es el siguiente:

```

inicializacion_framework()
while(true)
    palabra = reconocerPalabra(tiempo, microfono)
    if(palabra == comando)
        actualizar_parametros_cam( $\alpha, \beta, \rho$ )
         $T = \text{modelo\_cam}(\alpha, \beta, \rho)$ 
        send_iksolution_to(T)
    endWhile

```

La red neuronal ya entrenada y la función que la implementa para el reconocimiento de palabras se han obtenido de (MathWorks, 2022), y se encuentra encapsulada en la función *reconocerPalabra*. Gracias al desarrollo de este entorno de trabajo mediante MATLAB se ha podido utilizar un ejemplo ya desarrollado en el mismo, acortando enormemente los tiempos de desarrollo de la aplicación, en la que únicamente se ha tenido que combinar un ejemplo ya funcional con las funciones que actúan de interfaz con el framework desarrollado. Esto pone de manifiesto la utilidad del mismo para desarrollar aplicaciones rápidamente e integrar algoritmos en los que ya se está trabajando en el mismo grupo de investigación sin ninguna dificultad.

La aplicación desarrollada fue posteriormente utilizada para evaluar el cumplimiento de las restricciones desarrolladas en la subsección 3.1. Para ello se tomó un conjunto de 5 usuarios que realizaron en el simulador un ejercicio de recorte de patrones mientras recibían la asistencia del brazo robótico controlado por voz para el manejo del mando de la cámara. Una vez terminada la prueba, se les realizó un cuestionario consistente en tres preguntas:

1. Califique del 1 al 5 la facilidad de manejo del instrumental quirúrgico del simulador, considerando la interferencia física potencial del robot en el manejo del instrumental quirúrgico. El 1 significa que la presencia del robot impidió manejar el instrumental quirúrgico correctamente, siendo muy intrusivo con el espacio de trabajo, y el

5 significa que el instrumental se pudo manejar perfectamente.

2. Califique del 1 al 5 la visibilidad de la pantalla del simulador considerando la obstrucción potencial del robot en la visualización de la pantalla. El 1 significa que el robot ocluyó la pantalla impidiendo la visualización, y el 5 significa que el robot no influyó en la visualización de la pantalla.
3. Califique del 1 al 5 la confianza que le transmite la presencia física cercana de un robot manipulador durante el transcurso de la operación, teniendo en cuenta que se trata de un elemento extraño, siendo el 1 muy poca confianza y el 5 mucha confianza.

Las respuestas de cada uno de los participantes se recogen en la Tabla 2, donde se muestran el identificador de usuario, su nivel de familiaridad con sistemas de robótica quirúrgica, y la respuesta a cada una de las tres preguntas. En esta también se puede ver cómo todos los participantes en las pruebas han respondido con la puntuación más alta a las preguntas 1 y 2, lo cual demuestra que el framework desarrollado resuelve correctamente los problemas planteados por los criterios de no colisión, no oclusión, y de espacio de trabajo del cirujano. En cuanto a la pregunta 3, la media de las respuestas es de 4.2 sobre 5, indicando un alto índice de confianza en la presencia de un robot manipulador en un entorno quirúrgico muy próximo al operario. Esto se debe principalmente a la baja velocidad de movimiento del brazo manipulador, además de por las restricciones de posición del codo del impuestas en la sección 4.2, que impiden que este colisione con el operario. En general, también se aprecia una mayor puntuación en esta pregunta por parte de usuarios familiarizados con sistemas de robótica quirúrgica, indicando una mayor confianza en los elementos robóticos de la plataforma. Finalmente, se ha calculado el porcentaje de acierto del sistema de reconocimiento de voz, como la proporción de comandos captados correctamente respecto al total, que además incluye los reconocidos erróneamente y los no reconocidos. Para todos los usuarios participantes en el experimento, la media ha sido del 74 %.

Tabla 2: Resultados de la encuesta a los operarios tras la realización de pruebas con la aplicación desarrollada.

Usuario	Familiaridad	R1	R2	R3	Total
1	Alta	5	5	5	15
2	Alta	5	5	5	15
3	Alta	5	5	4	14
4	Baja	5	5	3	13
5	Baja	5	5	4	14
Media		5.0	5.0	4.2	14.2

6. Conclusiones

En este trabajo se ha descrito el desarrollo de un framework ciber-físico para cirugía laparoscópica en el que se trata la inclusión de restricciones de movimiento a un robot para cumplir con criterios ergonómicos del cirujano, teniendo en cuenta estas restricciones sobre las funciones de resolución del modelo cinemático inverso del robot cinemáticamente redundante empleado. Además, estas funciones desarrolladas en MATLAB

conforman un entorno de trabajo sobre el que se pueden diseñar en un futuro estrategias de automatización mediante robotización de instrumental quirúrgico. Este framework se ha implementado sobre un entorno realista basado en un simulador de laparoscopia, que ha permitido realizar experimentos que han validado el cumplimiento de las restricciones impuestas al framework, quedando así una herramienta para el desarrollo de algoritmos de robótica colaborativa y para el estudio de la ergonomía en este tipo de procedimientos en beneficio de la comunidad quirúrgica.

6.1. Líneas futuras

Se plantea la implantación de algoritmos desarrollados en el grupo de investigación mediante su implementación en MATLAB para su evaluación en el entorno de trabajo que este artículo recoge, así como estrategias basadas en el movimiento del endoscopio y en el uso de SLAM para obtener información 3D sobre la escena simulada. También se plantea la recolección de datos de vídeo mediante la interfaz de adquisición de imagen del framework con el ánimo de crear un dataset con el que entrenar algoritmos futuros, obteniendo una combinación de datos de vídeo y datos cinemáticos de las herramientas del simulador a través del uso de dispositivos de tipo *tracker*.

Agradecimientos

Este trabajo ha recibido financiación del proyecto "Sistema quirúrgico ciberfísico para abordaje laparoscópico", con identificador PY20-00738, al que se encuentra adscrito el grupo de investigación de Robótica Médica del Departamento de Ingeniería de Sistemas y Automática de la Universidad de Málaga.

Referencias

- Ae, G. P. Y. S., Ho, P., Albert, A. E., Ting, C. W., Jensen, A. E., Poon, T. C., Stephen, A. E., Cheng, W. K., Raymond, A. E., Tsang, C. C., 2009. Work-related musculoskeletal symptoms in surgeons. DOI: 10.1007/s10926-009-9176-1
- Aitchison, L. P., Cathy, Cui, K., Arnold, A., Nesbitt-Hawes, E., Abbott, J., 2016. The ergonomics of laparoscopic surgery: a quantitative study of the time and motion of laparoscopic surgeons in live surgical environments. DOI: 10.1007/s00464-016-4855-4
- Barrett-Technology, 2011. Barrett wam datasheet. Accessed: 2022-04-18. URL: https://web.barrett.com/files/WAMDataSheet_02.2011.pdf
- Barrett-Technology, 5 2021. barrett-ros-pkg repository. Accessed: 2022-02-23. URL: <https://git.barrett.com/software/barrett-ros-pkg>
- Barrett-Technology, 2022. libbarrett repository. Accessed: 2022-02-23. URL: <https://git.barrett.com/software/libbarrett>
- Beuss, F., Schmatz, F., Stepputat, M., Nokodian, F., Fluegge, W., Frerich, B., 2021. Cobots in maxillofacial surgery - challenges for workplace design and the human-machine-interface. Procedia CIRP 100, 488–493. DOI: 10.1016/J.PROCIR.2021.05.108
- Capolei, M. C., Wu, H., Andersen, N. A., Ravn, O., 2017. Positioning the Laparoscopic Camera with Industrial Robot Arm. DOI: 10.1109/ICCAR.2017.7942675
- García-Morales, I., 2006. Control operacional de un robot asistente para cirugía laparoscópica. Phd thesis, Universidad de Málaga.
- Geoghegan, R., Song, J., Singh, A., Le, T., Abiri, A., Mendelsohn, A. H., 7 2019. Development of a transoral robotic surgery training platform*. Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS, 5851–5854. DOI: 10.1109/EMBC.2019.8856971
- Hatzfeld, C., Neupert, C., Matich, S., Braun, M., Bilz, J., Johannink, J., Miller, J., Pott, P. P., Schlaak, H. F., Kupnik, M., Werthschützky, R., Kirschniak, A., 7 2017. A teleoperated platform for transanal single-port surgery: Ergonomics and workspace aspects. 2017 IEEE World Haptics Conference, WHC 2017, 1–6. DOI: 10.1109/WHC.2017.7989847
- Hu, J. C., Gu, X., Lipsitz, S. R., Barry, M. J., D'amico, A. V., Weinberg, A. C., Keating, N. L., 2009. Comparative effectiveness of minimally invasive vs open radical prostatectomy. JAMA 302, 1557–1564. URL: <https://jamanetwork.com/> DOI: 10.1001/jama.2009.1451
- iData Research, 2018. Robotic surgery statistics show movement towards more minimally invasive procedures - idata research. Accessed: 2022-09-28. URL: <https://idataresearch.com/robotic-surgery-statistics-show-movement-towards-more-minimally-invasive-procedures/>
- Jaffray, B., 2005. Minimally invasive surgery. Arch Dis Child 90, 537–542. URL: www.archdischild.com DOI: 10.1136/adc.2004.062760
- Kundrat, D., Dagnino, G., Kwok, T. M., Abdelaziz, M. E., Chi, W., Nguyen, A., Riga, C., Yang, G. Z., 10 2021. An mr-safe endovascular robotic platform: Design, control, and ex-vivo evaluation. IEEE Transactions on Biomedical Engineering 68, 3110–3121. DOI: 10.1109/TBME.2021.3065146
- Leporini, A., Oleari, E., Landolfo, C., Sanna, A., Larcher, A., Gandaglia, G., Fossati, N., Muttin, F., Capitanio, U., Montorsi, F., Salonia, A., Minelli, M., Member, G. S., Ferraguti, F., Secchi, C., Member, S., Farsoni, S., Sozzi, A., Bonfè, M., Sayols, N., Hernansanz, A., Casals, A., Hertle, S., Cuzzolin, F., Dennison, A., Melzer, A., Kronreif, G., Siracusano, S., Falezza, F., Setti, F., Muradore, R., Leporini, A. L. A., Bonfè, M., 2020. Technical and functional validation of a teleoperated multirobots platform for minimally invasive surgery. IEEE TRANSACTIONS ON MEDICAL ROBOTICS AND BIONICS 2. URL: <https://transenterix.com/> DOI: 10.1109/TMRB.2020.2990286
- MathWorks, 2022. Speech command recognition using deep learning. Accessed: 2022-11-01. URL: <https://www.mathworks.com/help/audio/ug/speech-command-recognition-using-deep-learning.html>
- Panait, L., Akkary, E., Bell, R. L., Roberts, K. E., Dudrick, S. J., Duffy, A. J., 10 2009. The role of haptic feedback in laparoscopic simulation training. Journal of Surgical Research 156, 312–316. DOI: 10.1016/J.JSS.2009.04.018
- Pandya, A., Reisner, L. A., King, B., Lucas, N., Composto, A., Klein, M., Ellis, R. D., 2014. A review of camera viewpoint automation in robotic and laparoscopic surgery. Robotics 3, 310–329. URL: www.mdpi.com/journal/robotics DOI: 10.3390/robotics3030310
- Pérez-Duarte, F. J., Sánchez-Margallo, F. M., Martín-Portugués, I. D.-G., ángel Sánchez-Hurtado, M., Lucas-Hernández, M., Gargallo, J. U., 5 2012. Ergonomía en cirugía laparoscópica y su importancia en la formación quirúrgica. Cirugia Espanola 90, 284–291. DOI: 10.1016/J.CIRESP.2011.04.021
- Russo, S., Dario, P., Menciassi, A., 2 2015. A novel robotic platform for laser-assisted transurethral surgery of the prostate. IEEE Transactions on Biomedical Engineering 62, 489–500. DOI: 10.1109/TBME.2014.2358711
- SARAS-Project, 4 2020. Multirobots-surgery platform conceptual ros* architecture - saras project. Accessed: 2022-09-28. URL: <https://saras-project.eu/?p=1442#more-1442>
- Setti, F., Oleari, E., Leporini, A., Trojaniello, D., Sanna, A., Capitanio, U., Montorsi, F., Salonia, A., Muradore, R., 5 2019. A multirobots teleoperated platform for artificial intelligence training data collection in minimally invasive surgery. 2019 International Symposium on Medical Robotics, ISMR 2019. DOI: 10.1109/ISMR.2019.8710209
- Singh, G. K., Claassens, J., 2010. An analytical solution for the inverse kinematics of a redundant 7dof manipulator with link offsets. The 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems. DOI: 10.1109/IRoS.2010.5649095
- Tiwari, M. M., Reynoso, J. F., High, R., Tsang, A. W., Oleynikov, D., 2011. Safety, efficacy, and cost-effectiveness of common laparoscopic procedures. DOI: 10.1007/s00464-010-1328-z
- Varas, N., 2019. usb_cam - ros wiki. Accessed: 2022-03-05. URL: http://wiki.ros.org/usb_cam
- Zendejas, B., Brydges, R., Hamstra, S. J., Cook, D. A., 4 2013. State of the evidence on simulation-based training for laparoscopic surgery: A systematic review. Annals of Surgery 257, 586–593. DOI: 10.1097/SLA.0B013E318288C40B