

Planificación de trayectoria de un robot para neurocirugía e integración en set up experimental

Poveda-Perez, M.^{a,*}, Manrique-Cordoba, J.^a, Lidon-Calvo, S.^a, Martorell-Llobregat, C.^b, Sabater-Navarro, J.M.^a

^aUnidad de Investigación en Robótica Médica, Universidad Miguel Hernández, Avda de la Universidad, s/n, 03202, Elche, España.

^bUnidad de Neurocirugía, Hospital General de Elche, 03202, España.

Resumen

Este trabajo presenta un procedimiento para la transferencia de planificación quirúrgica desde un entorno médico tridimensional hacia la ejecución física de trayectorias en un robot colaborativo. Se muestra la aplicación de este procedimiento en tareas experimentales de fresado craneal. A partir de trayectorias definidas manualmente sobre modelos anatómicos segmentados en 3D Slicer, se desarrolla un módulo capaz de generar automáticamente poses robóticas completas mediante el cálculo de normales superficiales y parámetros de ejecución asociados. La solución se implementa como una extensión específica de 3D Slicer y se integra con una arquitectura basada en ROS para su ejecución en un robot UR3e. Además, se diseña una interfaz implementada en Python que unifica el control del manipulador, sensores de fuerza, sistema de tracking óptico y herramienta rotativa de fresado, permitiendo una validación experimental entre planificación digital y ejecución física. Los resultados demuestran la viabilidad del procedimiento con secuencia planificación quirúrgica–planificación de trayectorias–control y ejecución en un entorno experimental orientado a aplicaciones de neurocirugía asistida por robot.

Palabras clave: Planificación quirúrgica, robótica médica, generación de trayectorias, cirugía asistida por robot.

Trajectory planning for a neurosurgery robot and integration into experimental setup

Abstract

This paper presents a procedure for transferring surgical plans from a three-dimensional medical environment to the physical execution of trajectories by a collaborative robot. This procedure is applied to experimental cranial milling tasks. A module is developed that can automatically generate complete robotic poses by calculating surface normals and associated execution parameters based on trajectories defined manually on anatomical models segmented in 3D Slicer. This solution was implemented as a custom 3D Slicer extension and integrated with a ROS-based architecture to execute on a UR3e robot. In addition, a Python-based interface is designed to unify control of the manipulator, force sensors, optical tracking system, and rotary milling tool, enabling validation of the digital planning process against physical execution. The results demonstrate the feasibility of the procedure, which involves the sequence of surgical planning, trajectory planning, control, and execution, in an experimental environment oriented toward robot-assisted neurosurgery applications.

Keywords: Surgical planning, robot-assisted surgery, trajectory generation, robot-assisted surgery.

1. Introducción

La robótica quirúrgica ha experimentado un crecimiento significativo en las últimas décadas, principalmente mediante sistemas teleoperados que amplían la precisión y destreza del cirujano. Plataformas comerciales como da Vinci® o Versius® han demostrado mejoras clínicas relevantes; sin embargo, el control continúa siendo teleoperado y dependiendo en gran me-

didada de la intervención directa del operador humano (Attanasio et al., 2021), lo que limita la automatización de tareas repetitivas o geoméricamente bien definidas, especialmente en procedimientos sobre tejido óseo.

En neurocirugía, intervenciones como el fresado craneal requieren trayectorias precisas derivadas de la planificación preoperatoria basada en imagen médica. Las herramientas actua-

les de planificación tridimensional permiten definir regiones de intervención sobre modelos anatómicos segmentados, pero la información generada suele limitarse a posiciones espaciales, sin incluir orientaciones del instrumento ni parámetros necesarios para su ejecución directa por sistemas robóticos, dificultando así la transición desde la planificación clínica hacia movimientos robóticos reproducibles. La transferencia fiable entre planificación virtual y ejecución física constituye uno de los principales retos en cirugía asistida por robot. A diferencia de procedimientos sobre tejidos deformables, la geometría rígida del cráneo permite explotar estrategias geométricas basadas en la superficie anatómica para generar trayectorias robóticas reproducibles y potencialmente automatizables. No obstante, la correspondencia entre las coordenadas de planificación y el espacio quirúrgico sigue siendo crítica, y aunque el registro quirúrgico se ha estudiado mediante técnicas como Iterative Closest Point (ICP), marcado de rasgos anatómicos o fiduciales artificiales (Mountney et al., 2010), la planificación clínica continúa proporcionando principalmente información posicional, persistiendo la dificultad de generar trayectorias robóticas directamente ejecutables con orientaciones coherentes del efector final y parámetros temporales asociados al movimiento.

En este contexto, este trabajo propone un procedimiento semiautomático que extiende la planificación quirúrgica tradicional mediante la generación automática de poses robóticas completas a partir de trayectorias definidas en 3D Slicer. La orientación se calcula utilizando las normales de la superficie segmentada, permitiendo adaptar el efector final a la morfología craneal. La planificación resultante se integra en una arquitectura basada en Robot Operating System (ROS) y se valida experimentalmente mediante su ejecución en un robot colaborativo UR3e (Universal Robots, 2026) equipado con una herramienta rotativa de fresado, demostrando la viabilidad del procedimiento con secuencia planificación quirúrgica–planificación de trayectorias–control y ejecución.

El artículo se organiza de la siguiente manera. La Sección 2 describe la metodología propuesta, la Sección 3 presenta los resultados experimentales obtenidos, la Sección 4 discute las implicaciones del sistema y la Sección 5 expone las conclusiones y líneas futuras de investigación.

2. Materiales y métodos

2.1. 3D Slicer; módulo para el cálculo de las normales

El sistema ha sido desarrollado sobre la plataforma de software libre 3D Slicer, ampliamente empleada en cirugía guiada por imagen para segmentación anatómica, reconstrucción tridimensional y planificación quirúrgica personalizada (Fedorov et al., 2012). En este trabajo, Slicer se utiliza como entorno computacional para generar información robótica directamente a partir de modelos anatómicos segmentados.

Se ha implementado un módulo propio en Python, denominado *calculateNormalOrientationAndValidRangeOrientation*, cuyo funcionamiento general se ilustra en el diagrama de la Figura 1. El módulo transforma fiduciales definidos sobre la superficie anatómica en poses robóticas completas.

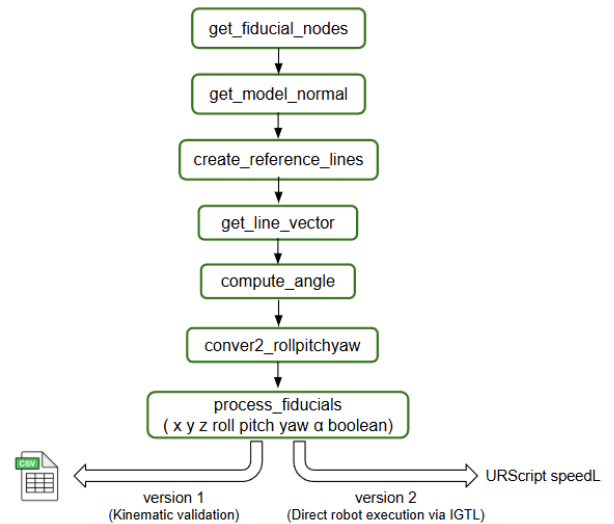


Figura 1: Diagrama del módulo calculateNormalAndValidRangeOrientation.

El módulo accede a los nodos generados por el cirujano mediante herramientas de marcado del entorno Medical Reality Markup Language (MRML), como (vtkMRMLMarkupsFiducialNode); ordena automáticamente los puntos siguiendo una nomenclatura estructurada y aplica, cuando existen, las transformaciones activas de la jerarquía MRML con el fin de expresar todas las coordenadas en el sistema de referencia global, garantizando la coherencia espacial entre el modelo anatómico y la trayectoria planificada. Esta relación se describe mediante la ecuación (1):

$$p_{\text{mundo}} = T \cdot p_{\text{local}} \quad (1)$$

donde p_{mundo} representa el punto en el sistema de referencia global, T es la matriz de transformación homogénea, y p_{local} corresponde al punto en el sistema de referencia local. Para cada fiducial, la normal superficial se calcula sobre la malla triangulada mediante funciones de la biblioteca Visualization Toolkit (VTK) que permiten localizar el vértice más cercano y evaluar la geometría local de la superficie. Cuando las normales no están previamente disponibles, estas se generan dinámicamente mediante el cálculo de normales de la malla (vtkPolyDataNormals), estimándose como el promedio de las caras adyacentes. El vector obtenido se normaliza posteriormente para definir la dirección unitaria de aproximación de la herramienta.

El sistema incorpora un parámetro clínico α que define la desviación angular máxima permitida respecto a la normal ideal. Esta restricción se representa geoméricamente como un cono de tolerancia alineado con la normal, proporcionando simultáneamente validación visual y limitación geométrica durante la planificación.

A partir de la normal unitaria se construye un sistema de referencia ortonormal local. El eje Z se alinea con la normal superficial; el eje X se obtiene proyectando un eje global sobre el plano perpendicular con el objetivo de evitar degeneraciones numéricas; y el eje Y se calcula mediante producto vectorial, formando una base ortonormal de orientación derecha (right-handed coordinate system). Con estos vectores se define la matriz de rotación $R=[x \ y \ z]$, que posteriormente se descompone siguiendo la convención Z–Y–X para obtener los ángulos

Roll–Pitch–Yaw compatibles con MATLAB, ROS y el controlador del robot UR3e, considerando explícitamente configuraciones singulares.

El procesamiento genera una trayectoria estructurada compuesta por posición cartesiana, orientación y parámetros de activación de la herramienta. En una etapa inicial, los datos se exportan en formato CSV para su integración con MATLAB, donde se realiza la validación cinemática del conjunto de poses generadas. Posteriormente, con el objetivo de aumentar el grado de automatización del flujo de trabajo, el módulo incorpora un mecanismo de salida directa hacia el entorno robótico mediante la generación automática de comandos `speedl`. Esta funcionalidad se implementa a través de una función de conversión desarrollada sobre la extensión OpenIGTLink (IGTL) (3D Slicer Community, 2018), lo que elimina la necesidad de archivos intermedios y permite una ejecución más directa y continua. De este modo, cada fiducial anatómico se transforma en una pose robótica completa, geoméricamente coherente y directamente utilizable tanto en la fase de planificación como en la ejecución experimental del procedimiento (UMH, 2025).

2.2. Interfaz sala de cirugía

La arquitectura del quirófano robótico experimental se implementa mediante una interfaz centralizada desarrollada en Python 3.8 y basada en Robot Operating System (ROS 1) como middleware de comunicación. Esta interfaz integra en un único entorno el control del robot colaborativo UR3e, la gestión de sensores de fuerza OnRobot, el sistema de seguimiento óptico VRPN/OptiTrack y los dispositivos de visualización, evitando la operación distribuida en múltiples consolas independientes. El diseño se fundamenta en una estructura modular compuesta por: (i) un módulo de acceso, implementado como el script `global_settings`, que emplea una clase singleton para inicializar y centralizar todas las variables globales, garantizando que todos los módulos accedan a un estado único y coherente; (ii) una interfaz gráfica principal (GUI, Main Interface) para la supervisión del entorno robótico y la monitorización de sensores y cámaras; y (iii) un módulo funcional (MRU.py) responsable de la coordinación de ROS, la ejecución de trayectorias, la lectura de topics y la gestión de procesos (Figura 2).

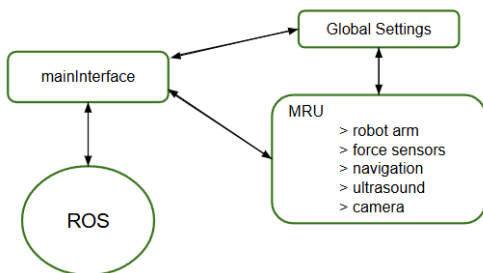


Figura 2: Diagrama de la interfaz.

La interfaz y el código desarrollado están disponibles públicamente en el repositorio GitHub enlace al repositorio, proporcionando acceso al entorno completo para su uso y replicación experimental (UMH, 2025). La Figura 3 muestra la GUI en ejecución, donde se visualizan estados articulares, fuerzas medidas y seguimiento óptico en tiempo real, permitiendo supervisión integral del quirófano experimental.

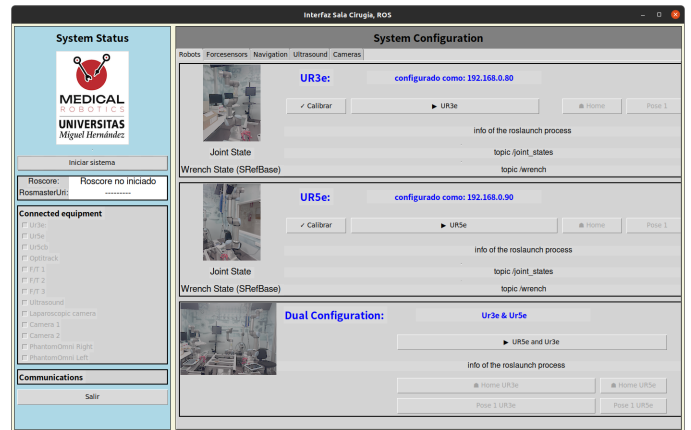


Figura 3: Interfaz sala de cirugía.

El núcleo funcional reside en MRU.py, donde se implementa una arquitectura multi hilo que permite la ejecución concurrente de robots y periféricos sin bloquear la interfaz gráfica. Los nodos ROS se gestionan mediante ROSLaunchParent o subprocesos controlados, permitiendo el arranque y parada segura de robots, sensores y sistemas de tracking. La lectura de topics se realiza en hilos independientes con actualización periódica, asegurando monitorización continua de estados articulares y fuerzas aplicadas. Se contemplan configuraciones mono-robot y dual-robot mediante un entorno de lanzamiento específico (workcell), facilitando la coordinación simultánea de múltiples manipuladores. La integración de sensores de fuerza, tracking óptico y cámaras proporciona un entorno experimental trazable, sincronizado y reproducible, con mecanismos de parada de emergencia y poses de inicio automatizadas que refuerzan la consistencia metodológica del sistema.

2.3. Efector final de fresado



Figura 4: Efector final UR3e con Dremel® y adaptador 3D.

Por simplicidad, como efector final se seleccionó la Dremel 4000, herramienta rotativa de alta velocidad (5.000–35.000 rpm, 175 W) adecuada para el mecanizado de materiales poliméricos empleados en la validación experimental. Debido a las limitaciones de carga del manipulador, el motor no se acopla directamente al flange, sino que se emplea un eje flexible que desacopla la masa y reduce las inercias, mejorando la estabilidad y precisión del sistema. El conjunto completo —incluyendo la herramienta, el eje flexible y el sistema de acoplamiento al

robot— se muestra en la Figura 4, donde puede observarse el montaje final del efector sobre el manipulador.

La restricción de curvatura del eje (~13 cm) condiciona el diseño de un adaptador específico desarrollado en Computer-Aided Design (CAD) y fabricado en ácido poliláctico (PLA) mediante impresión 3D, incorporando refuerzos estructurales y fijación compatible con el robot (Elvira Soler et al., 2022).

2.4. Descripción del flujo de trabajo

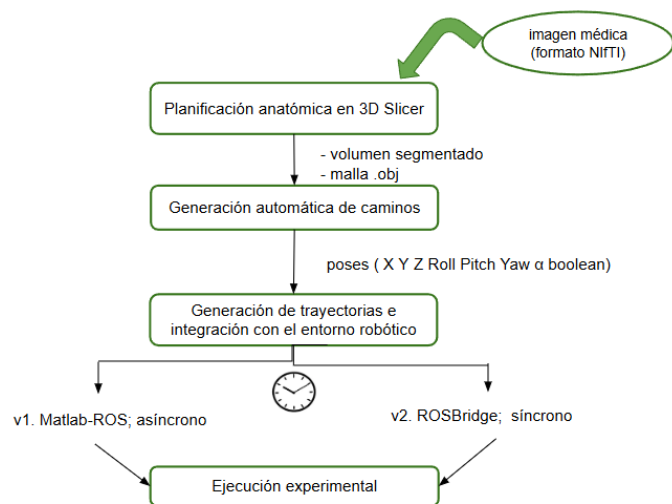


Figura 5: Flujo de trabajo

El flujo de trabajo propuesto, mostrado en la Figura 5, establece la transición completa desde la planificación quirúrgica basada en imagen médica hasta la ejecución física de trayectorias robóticas en un entorno experimental. El sistema se estructura en una cadena secuencial compuesta por planificación anatómica, generación de poses robóticas y ejecución mediante una arquitectura ROS, incorporando dos estrategias alternativas de integración entre el entorno médico y el sistema robótico.

Paso 1: Planificación anatómica en 3D Slicer

El proceso comienza con la carga de imágenes médicas de tomografía computarizada (CT) en 3D Slicer. La segmentación del cráneo se realiza mediante herramientas nativas del software basadas en umbralización de intensidad, aprovechando el alto contraste radiológico del tejido óseo. Posteriormente se aplica un refinamiento manual para obtener un modelo tridimensional específico del paciente.

La segmentación se transforma en una malla superficial que representa la geometría externa del hueso y constituye la base geométrica para la planificación. Sobre dicha superficie, el cirujano define de forma interactiva la trayectoria de fresado mediante la colocación de fiduciales proporcionados por el propio entorno de Slicer. Cada fiducial almacena su posición cartesiana en el sistema de referencia de la escena, permitiendo su posterior procesamiento automático en el módulo desarrollado. Durante el proceso se observó que la calidad de la malla influye directamente en la estabilidad del cálculo de normales y, por tanto, en la generación de orientaciones robóticas. Las mallas triangulares exportadas directamente en formato STL pueden

introducir inconsistencias geométricas, como caras invertidas o normales mal definidas, que afectan al procesamiento posterior.

Para evitar estos problemas, los modelos se exportaron en formatos que preservan mejor la estructura geométrica (OBJ/FBX) y se sometieron a un proceso de remallado controlado previo a su importación, optimizando la resolución y regularidad de los polígonos. Este procedimiento mejora la robustez del cálculo geométrico y permite además disponer de modelos preparados para fabricación mediante impresión 3D.

Paso 2: Generación automática de caminos

Una vez definida la trayectoria, se ejecuta el módulo desarrollado en este trabajo dentro de 3D Slicer (2.1), encargado de transformar los puntos anatómicos en poses robóticas completas. El procesamiento geométrico calcula, para cada punto, la orientación del efector final a partir de la geometría local de la superficie, generando una secuencia discreta de configuraciones espaciales independientes.

El resultado es un conjunto ordenado de poses definido por posición cartesiana, orientación espacial y parámetros de activación de herramienta, junto con las restricciones angulares establecidas por el usuario. Estas poses representan una descripción puramente geométrica de la trayectoria, sin parametrización temporal asociada.

En esta etapa, la variable tiempo no se asigna explícitamente. La trayectoria se define únicamente como una sucesión espacial de configuraciones, permitiendo desacoplar la planificación geométrica de la generación del movimiento. La parametrización temporal —velocidades, aceleraciones y continuidad cinemática— se introduce posteriormente en la fase de control robótico, donde las poses discretas se transforman en movimiento físicamente ejecutable.

Paso 3: Generación de trayectorias e integración con el entorno robótico

En esta etapa, las poses geométricas obtenidas durante la planificación se integran en el entorno robótico, permitiendo su posterior parametrización temporal y ejecución física. La integración fue desarrollada de forma progresiva a través de dos versiones sucesivas del sistema, que reflejan la evolución desde una arquitectura parcialmente manual hacia un flujo de trabajo completamente automatizado.

Primera versión (asíncrona):

Integración mediante MATLAB-ROS; en la primera versión del sistema, MATLAB actúa como capa intermedia entre la planificación y el entorno ROS. El archivo CSV generado se utiliza para validar la coherencia geométrica de las poses, adaptar convenciones de orientación y verificar la continuidad espacial antes de su ejecución. En esta etapa, cada fiducial deja de representar un punto anatómico aislado y pasa a constituir una pose robótica completa, definida por posición y orientación, preparada para su integración en el sistema de control.

La ejecución robótica se implementa mediante un esquema desarrollado en Simulink, encargado de procesar secuencialmente las poses almacenadas en el archivo CSV. El modelo lee cada fila del archivo en función de una señal temporal tipo step, generando de forma progresiva comandos de movimiento enviados al robot. Para cada instante discreto, el sistema cons-

truye y transmite instrucciones `speedl`, responsables de controlar la velocidad cartesiana del efector final.

Es en esta etapa donde se introduce explícitamente la variable tiempo. Mientras que el CSV contiene únicamente una descripción geométrica discreta, el comando `speedl` incorpora parámetros temporales y de velocidad que establecen la continuidad cinemática entre poses consecutivas. Como resultado, la secuencia de poses pasa a interpretarse como una trayectoria robótica ejecutable.

Segunda versión (síncrona):

Integración mediante ROSBridge; como evolución de la arquitectura anterior, se desarrolló una segunda versión orientada a eliminar etapas intermedias y aumentar el grado de automatización del flujo de trabajo. En la implementación inicial, la ejecución requería procesamiento externo mediante MATLAB, lo que introducía intervención manual y aumentaba la latencia del sistema.

La arquitectura propuesta sustituye este esquema por un canal de comunicación directa entre 3D Slicer y ROS mediante IGTL. El módulo desarrollado publica las poses calculadas directamente en los topics de ROS utilizando la herramienta `Converter_String`, que transforma la información geométrica en comandos de control robótico. Dichos comandos, implementados nuevamente mediante instrucciones `speedl`, son transmitidos a través de ROSBridge y ejecutados directamente por el controlador del robot. De este modo, la planificación quirúrgica y la ejecución robótica quedan integradas dentro de un flujo continuo, reduciendo la intervención manual y acercando el sistema a un escenario de integración clínica automatizada.

Paso 4: Ejecución experimental

Finalmente, la trayectoria es ejecutada por el robot colaborativo UR3e a través de la interfaz centralizada desarrollada en Python. Esta interfaz coordina la ejecución robótica junto con los sistemas de monitorización del entorno experimental, incluyendo sensores de fuerza, tracking óptico y supervisión visual. El efector final empleado consiste en una herramienta rotativa acoplada mediante eje flexible, permitiendo realizar operaciones de fresado manteniendo reducidas las cargas dinámicas sobre el manipulador.

3. Resultados

La validación del flujo de trabajo se realizó sobre un caso experimental de fresado craneal basado en imágenes de tomografía computarizada. La segmentación 3D permitió generar un modelo continuo del cráneo, posteriormente transformado en una malla superficial empleada para la planificación de la trayectoria (Figura 6). Sobre dicha superficie, el cirujano definió manualmente los puntos de la trayectoria de fresado, que fueron directamente procesados por el módulo automático de generación de poses (Figura 7(a)). Cada punto anatómico se tradujo en configuraciones robóticas completas, considerando posición cartesiana, orientación (Roll–Pitch–Yaw) y restricciones angulares (Figura 7(b)), garantizando que la herramienta mantuviera una dirección compatible con la superficie ósea y los límites cinemáticos del UR3e. La trayectoria final se exportó primero a

CSV para su validación intermedia y luego se integró directamente mediante ROSBridge, permitiendo un movimiento continuo sin reprocesamiento manual.

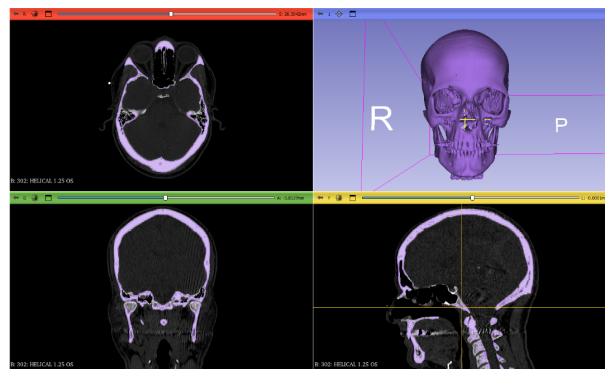
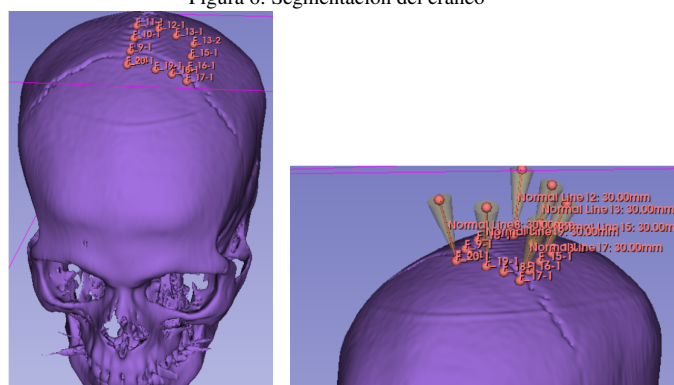


Figura 6: Segmentación del cráneo



(a) Camino sobre el modelo 3D (b) Trayectoria planificada sobre el modelo 3D

Figura 7: Resultados de `calculateNormalOrientationAndValidRange`

La ejecución robótica mostró una correcta interpretación de las poses y velocidades por parte del controlador, dando lugar a un movimiento continuo y estable del efector final. Para evitar la carga del motor en el efector final, se utilizó una Dremel con transmisión de eje flexible, de manera que el motor queda situado en una posición fija y estática, mientras que únicamente el eje flexible transmite el movimiento al efector final (Figura 8). La validación se verificó visualmente sobre el cráneo físico y mediante la representación gráfica de la trayectoria y orientación del efector final (Figura 9). En dicha gráfica, la línea discontinua negra corresponde a la trayectoria seguida por el efector final, mientras que la línea continua negra representa la trayectoria de la punta de la herramienta; las flechas azules indican la dirección del eje Z del efector final a lo largo del recorrido.

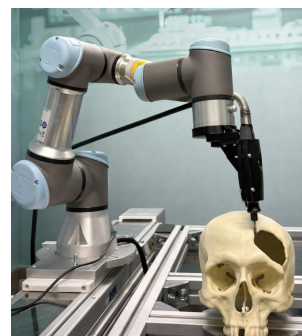


Figura 8: Validación experimental

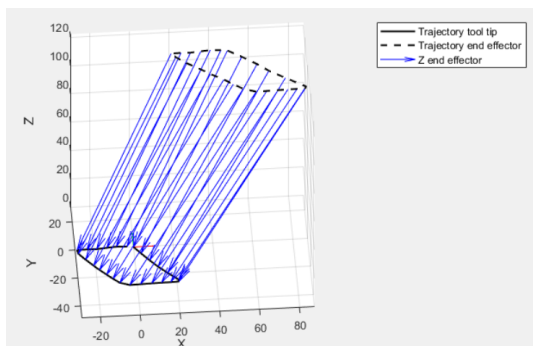


Figura 9: Gráfica trayectoria realizada por efector final y herramienta

Los resultados muestran que el flujo de trabajo propuesto asegura coherencia entre segmentación anatómica, generación automática de orientaciones y ejecución robótica. La calidad de la malla superficial se confirma como un factor crítico para la estabilidad geométrica, mientras que la integración mediante ROSBridge mejora significativamente la automatización y continuidad operativa. La correspondencia observada entre planificación virtual y ejecución física valida la viabilidad del enfoque para fresado craneal asistido por robot en un entorno experimental controlado.

4. Discusión

El procedimiento presentado aborda un reto clave en la transición de la planificación quirúrgica virtual a la ejecución robótica: generar trayectorias completas interpretables directamente por el robot. En muchas herramientas actuales, el cirujano define posiciones sobre modelos anatómicos, pero la orientación del instrumento se ajusta manualmente, limitando la reproducibilidad y la automatización de procedimientos geoméricamente definidos, como el fresado craneal. El método propuesto asocia automáticamente una orientación a cada punto mediante el cálculo de normales sobre la superficie craneal segmentada. La rigidez del cráneo permite criterios geométricos estables, generando poses robóticas coherentes tanto matemática como clínicamente. La validación por un neurocirujano confirmó que las orientaciones permiten un acceso perpendicular al hueso, esencial para un fresado controlado y seguro.

Otro aspecto relevante es la integración directa en 3D Slicer y su conexión con arquitecturas robóticas basadas en ROS, lo que acorta la brecha entre planificación médica y ejecución experimental. La coexistencia de una comunicación MATLAB-ROS y una alternativa directa vía ROSBridge evidencia la evolución hacia flujos de trabajo más automatizados y cercanos a un entorno clínico real. Entre las principales limitaciones del sistema destaca su dependencia de la calidad de la segmentación anatómica, ya que posibles irregularidades pueden afectar al cálculo de las normales y, en consecuencia, a la correcta orientación del efector final. Asimismo, la planificación actual se basa exclusivamente en criterios geométricos y no incorpora restricciones cinemáticas ni dinámicas del robot. Aunque la validación se llevó a cabo con un robot real, esta se limitó a la ejecución de los comandos generados durante la planificación, permitiendo verificar que el sistema reproduce las trayectorias de forma precisa y estable. No obstante, aún es necesaria su validación en escenarios quirúrgicos completos o mediante modelos anatómicos de alta fidelidad.

5. Conclusiones y trabajos futuros

Este trabajo ha presentado un sistema para generar y ejecutar trayectorias robóticas a partir de planificación quirúrgica en 3D Slicer, aplicado a fresado craneal experimental. El método transforma puntos definidos manualmente en poses robóticas completas mediante cálculo automático de normales superficiales, incorporando orientación y parámetros necesarios para la implementación física. La integración con ROS y una interfaz centralizada establece un flujo planificación-trayectoria-control robótico. La validación con un UR3e muestra la viabilidad de trasladar coherentemente la planificación digital al entorno físico. Las trayectorias evaluadas por un neurocirujano confirmaron coherencia geométrica y adecuación clínica, reforzando la aplicabilidad del enfoque como apoyo en cirugía asistida por robot.

Como líneas futuras, se propone la incorporación de restricciones cinemáticas y dinámicas del manipulador durante la generación de trayectorias para optimizar su ejecutabilidad, la integración de sensores de fuerza y sistemas de percepción visual para permitir adaptación intraoperatoria basada en la interacción con el entorno, la automatización completa de la comunicación entre el entorno de planificación y el sistema robótico mediante ROS para reducir la intervención manual y mejorar la continuidad del flujo de trabajo, y la evaluación del sistema en modelos anatómicos más realistas y en escenarios experimentales que simulen condiciones quirúrgicas complejas. Este trabajo representa un avance hacia sistemas de cirugía asistida por robot en los que la planificación quirúrgica puede entenderse como una descripción estructurada y directamente ejecutable por sistemas robóticos.

Agradecimientos

Esta investigación ha sido financiada por el Gobierno español —a través de la Agencia Estatal de Investigación (AEI)— mediante el proyecto PID2022-138206OB-C32, y por la Generalitat Valenciana mediante el proyecto CIPROM/2022/16.

Referencias

- 3D Slicer Community, 2018. Openigtlinkif module documentation. <https://www.slicer.org/wiki/Documentation/4.10/Modules/OpenIGTLinkIF>, accedido: 23 de febrero de 2026.
- Attanasio, A., Scaglioni, B., De Momi, E., Fiorini, P., Valdastri, P., 2021. Autonomy in surgical robotics. *Annual Review of Control, Robotics, and Autonomous Systems* 4(1), 651–679. DOI: 10.1146/annurev-control-062420-090543
- Elvira Soler, G., Manrique Córdoba, J., Vicente Samper, J., Sabater Navarro, J., 2022. Monitorización de fuerzas de mecanizado con un robot ur3e. *Jornadas de Automática: libro de actas*, 687–693. DOI: 10.17979/spudc.9788497498418.0687
- Fedorov, A., Beichel, R., Kalpathy-Cramer, J., Finet, J., Fillion-Robin, J.-C., Pujol, S., Bauer, C., et al., 2012. 3d slicer as an image computing platform for the quantitative imaging network. *Magnetic Resonance Imaging* 30(9), 1323–1341. DOI: 10.1016/j.mri.2012.05.001
- Mountney, P., Stoyanov, D., Yang, G.-Z., 2010. Three-dimensional tissue deformation recovery and tracking. *IEEE Signal Processing Magazine* 27(4), 14–24. DOI: 10.1109/MSP.2010.936728
- UMH, M. R., 2025. Interfaz-laboratorio. <https://github.com/medicalroboticsumh/Interfaz-Laboratorio.git>, accedido: 17 de febrero de 2026.
- Universal Robots, 2026. Ur3e collaborative robot. <https://www.universal-robots.com/products/ur3e/>, accedido: 27 de marzo de 2026.