



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
PROGRAMA DE MAESTRÍA Y DOCTORADO EN INGENIERÍA
INGENIERÍA ELÉCTRICA – PROCESAMIENTO DIGITAL DE SEÑALES

SISTEMA DE GUIADO POR VISIÓN COMPUTACIONAL APLICADO A UN SISTEMA
ROBÓTICO PARA LA MEJORA DE PROCEDIMIENTOS QUIRÚRGICOS DE
COLUMNA

TESIS

QUE PARA OPTAR POR EL GRADO DE:
MAESTRO EN INGENIERÍA

PRESENTA:
IAN MICHEL CRUZ SORIA

TUTOR PRINCIPAL:
DR. MIGUEL ÁNGEL PADILLA CASTAÑEDA
INSTITUTO DE CIENCIAS APLICADAS Y TECNOLOGÍA (ICAT)

MÉXICO, CD. MX. DICIEMBRE 2023



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

JURADO ASIGNADO:

Presidente: Dra. Medina Gómez Lucía

Secretario: Dr. Lomas Barrie Víctor Manuel

1 er. Vocal: Dr. Padilla Castañeda Miguel Ángel

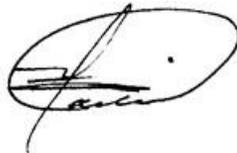
2 do. Vocal: Dr. Savage Carmona Jesús

3 er. Vocal: Dr. Felipe Miguel Camarillo Juárez

Ciudad Universitaria, Instituto de Ciencias Aplicadas y Tecnología (ICAT). Programa de Maestría y Doctorado en Ingeniería Eléctrica, UNAM.

TUTOR DE TESIS:

Dr. Padilla Castañeda Miguel Ángel



FIRMA

Agradecimientos

Deseo agradecer a los proyectos **“Esquema de neuronavegación quirúrgica por imágenes multiespectrales y simulación biomecánica para resección de tumores cerebrales guiada por computadora”** DGAPA-PAPIIT TA101422, **“Metodología de realidad aumentada multiespectral para la identificación y localización de tejido patológico en intervenciones quirúrgicas guiadas por computadora”** CONAHCyT 319585, SECTEI 219/2019, y SECTEI 087/2023 **“Laboratorio de Investigación y Desarrollo de Simuladores Clínicos y Cirugía Asistida por Computadora”** por financiar este trabajo.

Resumen

En este trabajo de tesis, se expone la investigación y desarrollo de un sistema de guiado por visión computacional aplicado a un robot para mejorar cualquier procedimiento de punción y procedimiento percutáneos, se aborda específicamente el ámbito ortopédico, focalizado en procedimientos de mínima invasión en la región lumbar de la columna vertebral, teniendo como tarea final la implantación precisa de tornillos pediculares.

Este enfoque se fundamenta en los logros obtenidos de investigaciones conjuntas anteriores entre el Dr. Miguel Ángel Padilla Castañeda por parte de UNAM-ICAT-UIDT (Universidad Nacional Autónoma de México, Instituto de Ciencias Aplicadas y Tecnología, Unidad de Investigación y Desarrollo Tecnológico) y el Ortopedista y médico cirujano Felipe Miguel Camarillo Juárez del HGM (Hospital General de México Eduardo Liceaga), donde los esfuerzos de ambos fueron clave para la consolidación de este trabajo.

En el capítulo 1 (Introducción) del presente trabajo de tesis se abordan los problemas presentados por el sistema esquelético, las intervenciones quirúrgicas enfocadas en la estabilización de la columna vertebral, las complicaciones de alto riesgo y la innovación tecnológica que permite mejorar la precisión quirúrgica.

En el capítulo 2 (Antecedentes) se destacan los desafíos que se han superado en la cirugía asistida por sistemas robóticos de última generación, enfatizando la seguridad proporcionada tanto al paciente como al cirujano en la minimización de la exposición a la radiación durante la colocación de tornillos pediculares, debido a la combinación de sistemas de navegación quirúrgica y sistemas robóticos.

El capítulo 3 (sistema de seguimiento) aborda la técnica de registro y seguimiento de la posición de las herramientas, el desarrollo de software y la creación del material necesario para su implementación dentro de un entorno gráfico para visualizar la reconstrucción 3D de las herramientas, detallando el algoritmo de alineación desarrollado para este trabajo.

El capítulo 4 (software de planeación) presenta un software para generar el plan preoperatorio mediante un entorno de simulación 3D con diferentes vistas del modelo virtual de la columna obtenido a partir de los estudios de imagen del paciente. El software sirve para visualizar estructuras anatómicas no visibles y establecer la postura ideal para cada tornillo en la columna del paciente.

El capítulo 5 (Integración del sistema robótico) describe las características del robot, el análisis cinemático de cada mecanismo planar, la interfaz de comunicación con el sistema físico, el algoritmo de estimación para alinear el efector final del robot con la trayectoria del tornillo y la integración en el entorno virtual.

En el capítulo 6 (Pruebas y resultados) se describen las pruebas y resultados de los diversos subsistemas vistos en los capítulos anteriores que conforman el sistema general propuesto, incluye la descripción de ciertas funcionalidades del mecanismo, la simulación de una trayectoria recta del mecanismo realizada en Wólfam Mathematica, además incluye una descripción de las pruebas y resultados que evalúan la precisión del sistema de seguimiento con resultados satisfactorios. El capítulo finaliza con la implementación del sistema general y el despliegue del prototipo funcional seguido de una evaluación

cualitativa de los resultados, comparando visualmente la escena real con la reconstruida en el entorno virtual desde perspectivas similares. Este análisis visual confirma la notable similitud entre ambas representaciones, subrayando la efectividad y fidelidad del sistema implementado.

En el capítulo 7 (Conclusiones) se menciona la contribución al campo de la cirugía ortopédica con la aplicación potencial del trabajo y la contribución al campo del procesamiento digital de señales mediante los algoritmos desarrollados para lograr la precisión requerida, destacando que, aunque se requieren más pruebas cuantitativas, la congruencia visual entre objetos en ambos entornos real y virtual sugiere que el sistema podría ser probado en modelos cadavéricos y posteriormente en un entorno quirúrgico real.

En el capítulo 8 (Trabajos a futuro), se mencionan algunas modificaciones en busca de mejorar el sistema para lograr alcanzar las capacidades de los robots quirúrgicos comerciales.

Abstract

In this thesis work, the research and development of a computer vision-guided system applied to a robot to enhance any puncture and percutaneous procedures are presented, it specifically addresses the orthopedic field, focused on minimally invasive procedures in the lumbar region of the spine, having as a final task being the precise implantation of pedicular screws.

This approach is based on the achievements obtained from previous joint research between Dr. Miguel Angel Padilla Castañeda from UNAM-ICAT-UIDT (Universidad Nacional Autónoma de México, Instituto de Ciencias Aplicadas y Tecnología, Unidad de Investigación y Desarrollo Tecnológico) and Surgeon Felipe Miguel Camarillo Juarez from the HGM (Hospital General de México Eduardo Liceaga), where the efforts of both were key to the consolidation of this work.

In Chapter 1 (Introduction) of this thesis work, the problems presented by the skeletal system, the surgical interventions focused on spinal stabilization, the high-risk complications, and the technological innovation that allows improved surgical precision are addressed.

In Chapter 2 (background), the challenges overcome in assisted surgery robotic systems are accentuated, emphasizing the safety provided to both the patient and the surgeons in minimizing radiation exposure during the placement of pedicular screws, due to the combination of surgical navigation systems and robotic systems.

Chapter 3 (Tracking System) addresses the technique of tool position registration and tracking, software development, and the creation of the necessary material for its implementation within a graphical environment to visualize the 3D reconstruction of the tools, detailing the alignment algorithm developed for this work.

Chapter 4 (Planning Software) presents software to generate the preoperative plan through a 3D simulation environment with different views of the virtual model of the spine obtained from the patient's imaging studies. The software is useful for visualizing non-visible anatomical structures and establishing the ideal position for each screw in the patient's spine.

Chapter 5 (Integration of the Robotic System) describes the characteristics of the robot, the kinematic analysis of each planar mechanism, the communication interface with the physical system, the estimation algorithm to align the robot's end effector with the screw trajectory, and the integration into the virtual environment.

Chapter 6 (Testing and Results) describes the test and results of the various subsystems seen in previous chapters that constitute the general system, including the description of certain mechanism functionalities, the simulation of a straight mechanism trajectory performed in Wolfram Mathematica, as well it includes a description of the test and results which evaluates the accuracy of the tracking system with satisfactory outcomes. The chapter concludes with the implementation of the general system and the deployment of the functional prototype followed by a qualitative evaluation of the results, visually comparing the real scene with the reconstructed one in the virtual environment from similar perspectives.

This visual analysis confirms the remarkable similarity between both representations, emphasizing the effectiveness and fidelity of the implemented system.

In Chapter 7 (Conclusions), the contribution to the field of orthopedic surgery with the potential application of the work and the contribution to the field of digital signal processing by the algorithms developed to achieve the required precision are mentioned, highlighting that, although more quantitative test are required, the visual congruence between objects in both real and virtual environments suggest that the system could be tested in cadaveric models and subsequently in a real surgical environment.

In Chapter 8 (Future work), some modifications are mentioned in pursuit of improving the system to achieve the capabilities of commercial surgical robots.

ÍNDICE GENERAL

Resumen	1
Abstract.....	3
1.- Introducción.....	9
2.- Antecedentes.....	11
2.1 Sistemas robóticos	11
2.1.1 Da Vinci Surgical System	11
2.1.2 Mazor	12
2.1.3 ROSA.....	12
2.2 Propuesta del sistema.....	13
2.3 Objetivo.....	14
2.4 Justificación.....	14
2.5 Metas	14
2.6 Alcances	15
2.7 Hipótesis.....	15
2.8 Metodología.....	16
2.8.1 Navegador.....	16
2.8.2 Robot.....	17
2.8.3 Integración del sistema	17
3.- Sistema de seguimiento.....	18
3.1 Modelado de marcos de referencia.....	20
3.2 Sistema de registro con cámaras	22
3.3 Estimación de la rotación del marco de referencia (Algoritmo de Alineación)	25
3.3.1 Calculo del vector normal promedio	25
3.3.2 Asignación de versores	28
3.3.3 Cálculo de los ángulos de Euler.....	30
3.4 Estimación de la posición del marco de referencia (Algoritmo de Alineación)	31
4.- Software de planeación	34
5.- Integración del sistema robótico	36

5.1 Descripción del sistema robótico.....	37
5.2 Cinemática del mecanismo planar.....	39
5.3 Cinemática Inversa del mecanismo planar.....	41
5.4 Interfaz de comunicación.....	44
5.5 Algoritmo de estimación para el posicionamiento del robot sobre trayectoria proyectada por el tornillo.....	45
5.6 Integración con el entorno virtual.....	47
6.- Pruebas y resultados.....	52
6.1 Simulación del movimiento rectilíneo del mecanismo.....	52
6.2 Descripción del prototipo robótico.....	53
6.3 Cálculo de la precisión del sistema de seguimiento.....	54
6.4 Despliegue del sistema robótico guiado por visión computacional.....	56
7.- Conclusión.....	61
8.- Trabajos a futuro.....	62
Referencias.....	63
Anexos.....	66
8.1 Conceptos relevantes.....	66
8.1.1 Entorno virtual.....	66
8.1.2 Websocket.....	66
8.1.3 Cambio de base.....	66
8.1.4 Matrices de transformación homogénea (M.T.H.).....	66
8.2 Algoritmos de registro y alineación recurrentes en la literatura.....	68
8.2.1 Iterative Closest Point (ICP).....	68
8.2.2 Coherent Point Drift (CPD).....	68
8.2.3 Ajuste por Mínimos Cuadrados basado en descomposición de valores singulares (SVD).....	68

ÍNDICE DE FIGURAS

Figura 1 Cirugia de columna vertebral de enfoque abierto	9
Figura 2 Da Vinci Surgical System [8]	11
Figura 3 Mazor X Stealth Edition robotic system [8]	12
Figura 4 ROSA ONE Spine System [8]	13
Figura 5 Diagrama de funcionamiento del sistema de seguimiento	18
Figura 6 Diagrama de flujo de información del sistema de seguimiento	18
Figura 7 Diagrama de comunicación cliente – servidor del algoritmo de alineación	19
Figura 8 Diagrama del proceso de generación de marcos de referencia	20
Figura 9 Interfaz gráfica de usuario para generar archivos JSON	21
Figura 10 Marco de referencia y estructura del archivo JSON	21
Figura 11 Sistema de cámaras V120 Trio	22
Figura 12 Diagrama de funcionamiento del sistema de cámaras y vista real - virtual de un marco de referencia.....	23
Figura 13 Diagrama de distribuciones con sistema coordinado asignado por Motive.....	24
Figura 14 Vistas de la distribución de esferas referidas al sistema coordinado global	25
Figura 15 Gráficas de la comparación entre vectores de acuerdo con la esfera evaluada	26
Figura 16 Gráficas del vector normal al plano para cada esfera evaluada	27
Figura 17 Gráfica del vector normal promedio.....	27
Figura 18 Gráficas de la distribución de esferas y asignación del versor “x”	28
Figura 19 Gráficas de generación del versor “y”	29
Figura 20 Gráficas 3D del sistema de coordenadas intrínseco a la distribución asociado al centroide	29
Figura 21 Matriz de rotación con secuencia ZYX y nomenclatura de elementos	30
Figura 22 Diagrama de flujo del algoritmo para cálculo de ángulos de Euler	31
Figura 23 Gráficas 3D de la generación del vector [iOd] referido a la base M y D	32
Figura 24 Gráficas 3D de la estimación de la posición del marco de referencia	33
Figura 25 Vista de la interfaz gráfica de usuario del software de planeación	34
Figura 26 Fragmentos que muestran la rotación del tornillo en los ejes globales	35
Figura 27 Imagen del prototipo robótico de configuración híbrida	36
Figura 28 Esquema del robot de configuración planar	37
Figura 29 Imágenes y esquema del robot de configuración hibrida.....	38
Figura 30 Diagrama de bloques del sistema de robótico.....	38
Figura 31 Diagrama de la cadena cinemática del mecanismo planar.....	39
Figura 32 Esquema de vectores para definir la ecuación de lazo cerrado.....	42
Figura 33 Gráfica del polinomio seleccionado	43
Figura 34 Diagrama de comunicación cliente – servidor para posicionar al robot	44
Figura 35 Gráficas 3D de los vectores que definen al tornillo referidos a la base del robot.....	45

Figura 36 Gráficas 3D de los vectores que definen la estimación de la posición de la base móvil	46
Figura 37 Componentes del entorno virtual.....	47
Figura 38 Generación de la trayectoria estimada para implantar el tornillo seleccionado.....	48
Figura 39 Vista anteroposterior de trayectorias para implantar en la columna vertebral del entorno virtual	49
Figura 40 Vista axial de trayectorias para implantar tornillo en la columna vertebral dentro del entorno virtual	49
Figura 41 Vista lateral de trayectorias para implantar tornillo en la columna vertebral dentro del entorno virtual	49
Figura 42 Casos de posicionamiento dentro del área de trabajo de cada base móvil	50
Figura 43 Marcos de referencia asociados a columna, efector final del robot y herramienta de punción reales.....	51
Figura 44 Marcos de referencia asociados a columna, efector final del robot y herramienta de punción virtuales.....	51
Figura 45 Gráficas de los ángulos y simulación del movimiento rectilíneo del robot	52
Figura 46 Vistas del prototipo robótico ensamblado.....	53
Figura 47 Tablilla con matriz de puntos y herramienta de punción.	54
Figura 48 Tablilla de calibración dentro del entorno virtual y real para el proceso de calibración del error de posicionamiento del sistema.	55
Figura 49 Diagrama de conexión para el despliegue del sistema	56
Figura 50 Vista lateral del robot y axial a la columna de la escena real y virtual	57
Figura 51 Vista frontal del robot y lateral a la columna de la escena real y virtual.....	57
Figura 52 Vista superior del robot y anteroposterior a la columna de la escena real y virtual	58
Figura 53 Vista axial a la columna de la escena real y virtual	58
Figura 54 Vista lateral de la columna de la escena real y virtual	59
Figura 55 Acercamiento a vistas: a) Anteroposterior, b) Axial c) Lateral.....	60

1.- Introducción

El sistema óseo desempeña un papel fundamental en la vida diaria y funcionalidad humana al proporcionar soporte estructural, proteger órganos vitales, facilitar el movimiento y actuar como reservorio de minerales esenciales. Sin embargo, puede presentar diversos problemas como fracturas, osteoporosis y deformidades, impactando negativamente la movilidad y la calidad de vida por el dolor o limitaciones que causan estas afecciones.

Para abordar estos desafíos, se recurre a intervenciones quirúrgicas destinadas a corregir problemas musculoesqueléticos como las cirugías ortopédicas, las cuales requieren una planificación personalizada. En el caso de la estabilización de la columna vertebral, se buscan soluciones como la inserción de tornillos pediculares y barras para dar soporte posterior a la columna al distribuir las cargas a través de los componentes [1], [2]. Como se describe a continuación, en la parte superior de la Figura 1 se muestra a un paciente con deformidad en la columna, junto con un fragmento de su estudio de imagen y en la parte inferior se muestran dos cirugías de enfoque abierto de pacientes distintos donde se observan los tornillos y las barras que se implantaron a lo largo de la columna vertebral.

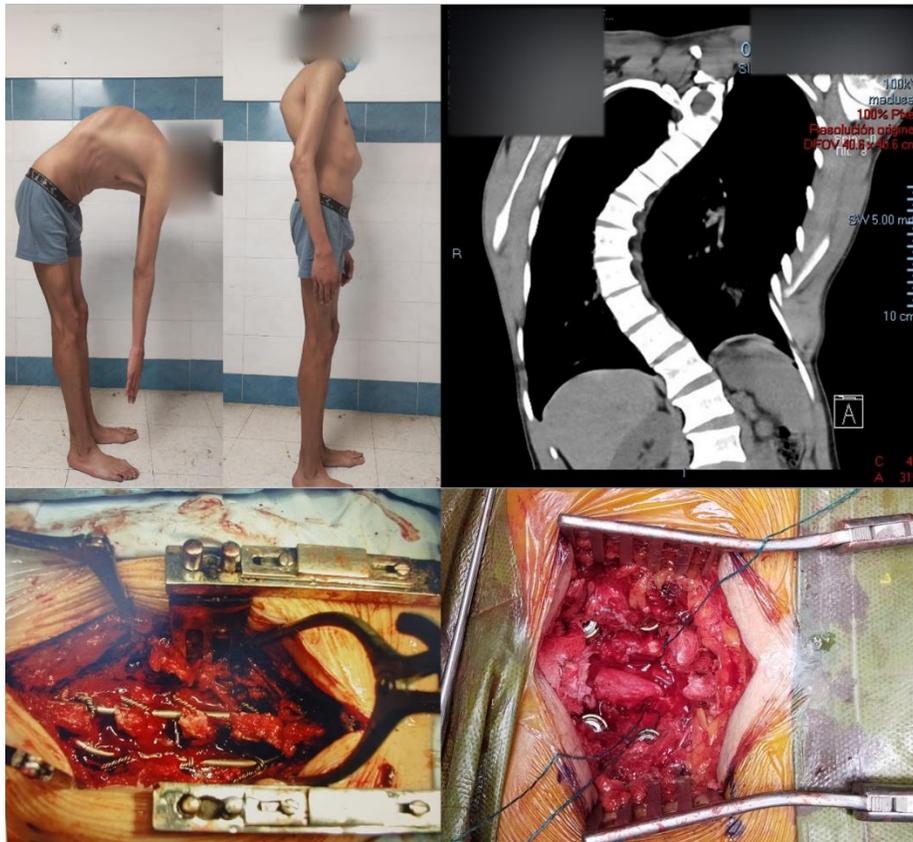


Figura 1 Cirugía de columna vertebral de enfoque abierto

Esta tarea demanda una definición precisa del ángulo de inclinación del tornillo para prevenir complicaciones derivadas de perforaciones incorrectas o violaciones a la pared medial del pedículo, las cuales podrían dañar los nervios de la medula espinal, el conducto raquídeo y raíces nerviosas. Ingresar al conducto raquídeo con el tornillo va a ocasionar una lesión neuronal irreversible, mientras que una posición muy lateral del tornillo provocaría una lesión vascular, llevando a consecuencias graves, incluso la muerte del paciente [1], [3].

La importancia de estas intervenciones se ve respaldada por la innovación tecnológica, que ha permitido el desarrollo de dispositivos médicos y técnicas quirúrgicas más precisas [1], [2], [3], [4]. El uso de estudios de imagen como tomografía computarizada (TC) o resonancias magnéticas (RM), junto con la posibilidad de reconstrucción en entornos virtuales y físicos mediante manufactura aditiva, facilita una mejor comprensión del caso clínico y una valoración más precisa antes de la operación, mejorando así los resultados.

La cirugía de columna desempeña un papel crucial en la movilidad y la estabilidad del cuerpo. Buscando aliviar diversas condiciones como hernias de disco, estenosis espinal y escoliosis, se utilizan enfoques de técnicas abiertas y de mínima invasión. La medicina actual ha avanzado significativamente, especialmente en el ámbito de las cirugías de mínima invasión (MIS). A pesar de los beneficios ya demostrados, como la reducción de pérdida de sangre, la disminución en la necesidad de aplicar narcóticos postoperatorios y una recuperación más rápida del paciente [5], [6], [7], [8]; se presentan desafíos en términos de visión y ejecución, debido a que el espacio que permite el acceso al órgano objetivo suele ser reducido y requiere de precisión milimétrica.

Estos desafíos en cirugías de columna han impulsado el desarrollo de sistemas robóticos que han ganado relevancia por colocar de implantes con precisión milimétrica, estas innovaciones mejoran la seguridad de los procedimientos y la calidad de vida de los pacientes ya que adoptan el enfoque y atienden las necesidades de las cirugías de mínima invasión.

Las plataformas robóticas DaVinci, Mazor y ROSA, son reconocidas por ofrecer gran estabilidad y control al cirujano, demostrado precisión en la colocación de tornillos pediculares [8], [9], [10]. Aunque se requiere más investigación para validar su superioridad, incluida la evaluación del costo-beneficio frente a técnicas tradicionales, estas plataformas mejoran significativamente la eficacia del proceso [11].

En el contexto de las técnicas guiadas, el uso de fluoroscopia para obtener referencias visuales del órgano de interés implica mayor exposición a la radiación del equipo quirúrgico [6], [12].

En contraste, las cirugías asistidas por ordenador, con referencias visuales en 2D y 3D, no solo mejoran el flujo de trabajo, sino que también reducen la exposición a la radiación al minimizar el uso de técnicas que involucran fluoroscopia [6], [7], [11], [12].

La combinación de navegación y sistemas robóticos posibilita procedimientos quirúrgicos más seguros con enfoques mínimamente invasivos; ya que generan un mapa del paciente utilizando los estudios de imagen (resonancias magnéticas o estudios de tomografía computarizada) de los pacientes, lo que facilita la visualización de estructuras anatómicas no visibles [12], [13], [14], y complementa la destreza manual del cirujano.

2.- Antecedentes

A lo largo de la historia, han surgido varios robots quirúrgicos para asistir distintos procedimientos, incluyendo laparoscopia y cirugías craneales, a continuación, se destacan los sistemas robóticos orientados a procedimientos espinales que han evolucionado, combinando un sistema de navegación utilizando cámaras para el seguimiento de herramientas y del paciente dentro de la escena quirúrgica.

2.1 Sistemas robóticos

2.1.1 Da Vinci Surgical System

El sistema Da Vinci (Figura 2) es un sistema robótico quirúrgico que en 2006 se adaptó para asistir procedimientos espinales, una de las características más importantes de este sistema es la configuración maestro-esclavo, el cual le permite al cirujano realizar movimientos precisos con instrumentos miniatura desde una consola, mejorando su destreza y visión en comparación con técnicas convencionales [8], [10].



Figura 2 Da Vinci Surgical System [8]

2.1.2 Mazor

El sistema Mazor (Figura 3) cuenta con distintas versiones que introdujeron mejoras, en 2004 Mazor SpineAssist estableció el primer sistema de control compartido aprobado por la FDA, el cual requería ser montado directamente en el paciente mediante alambres de Kirschner y utilizando imágenes tomografía computarizada (CT) se registraba una vértebra a la vez; posteriormente en 2011 Mazor Renaissance permitió al cirujano aplanar la prominencia ósea antes de realizar la perforación evitando desviar el tornillo y mejorando el rango de precisión que oscila entre 85% a 100%; en 2016 Mazor X cambio el sistema de montaje y el sistema robótico; incorporo el análisis preoperatorio (para simular el impacto de la corrección en la alineación espinal), además de la guía (imagen 3D usando estudios de CT y de fluoroscopia) y verificación intraoperatoria (seguimiento mediante cámara 3D mejorando el flujo de trabajo). En 2018 Mazor X Stealth Edition recibió la aprobación por la FDA, elimina la necesidad del sistema de alambres Kirschner e integra la tecnología Medtronic Stealth Station para el seguimiento en tiempo real de la escena quirúrgica mediante el uso de cámaras, lo cual además permite ajustar la posición del robot [8], [9], [10].



Figura 3 Mazor X Stealth Edition robotic system [8]

2.1.3 ROSA

El sistema ROSA (Figura 4) cuenta con dos versiones que puede asistir procedimientos de columna, aunque originalmente fue diseñado para operaciones craneales, el sistema ROSA Spine fue aprobado por la FDA en 2016, el sistema se conforma por un brazo robótico serial de seis grados de libertad montado en un carro y una segunda base con una cámara para seguimiento óptico.

En 2019 el sistema fue actualizado a ROSA ONE Spine, el cual es capaz de asistir procedimientos espinales intracraneales y ortopédicas [8], [9], [10].



Figura 4 ROSA ONE Spine System [8]

2.2 Propuesta del sistema

En este contexto, se propone un sistema robótico similar a los anteriormente expuestos, diseñado para incrementar la eficiencia y reducir el riesgo en cirugías de punción e implantación de tornillos pediculares en la columna vertebral con enfoque de mínima invasión.

Abordando los desafíos de visualización y ejecución para el cirujano, se plantea un sistema de navegación integrado por una cámara de visión estéreo capaz de detectar y estimar la postura de objetos relevantes en la escena quirúrgica, que, junto con un software, genere una reconstrucción virtual de los elementos registrados, desplegando una guía visual en el entorno virtual, cuya finalidad sea la de permitir al médico libertad de movimiento para localizar y monitorear continuamente el órgano objetivo durante la etapa intraoperatoria del procedimiento, basándose en un modelo 3D previamente reconstruido mediante estudios de imagenología del paciente y cargado al software en la etapa preoperatoria [8], [11], [14].

Aunque los métodos de imagenología son cruciales, la destreza manual del cirujano sigue siendo esencial para el éxito del procedimiento quirúrgico. Por lo tanto, se propone el uso de un sistema robótico híbrido con 4 grados de libertad capaz de brindar una guía física al cirujano, requiriendo montar la herramienta de punción en el efector final del robot, logrando posicionarla sobre la trayectoria requerida para la implantación de tornillos pediculares en la columna vertebral del paciente, con el fin de reducir el tiempo de ejecución y garantizar la precisión en la colocación del tornillo.

2.3 Objetivo

El objetivo de este trabajo consiste en desarrollar un prototipo funcional de un sistema robótico que integre navegación quirúrgica y un algoritmo de alineación preciso. Este sistema tiene como propósito guiar procedimientos ortopédicos, específicamente cirugías de mínima invasión destinadas a la inserción de tornillos pediculares. Se enfatiza la necesidad de evitar que los tornillos queden dentro de la pared medial del pedículo, con el fin de prevenir lesiones neurológicas durante los procedimientos de columna, especialmente en el tratamiento de la escoliosis. Además, se busca optimizar la relación costo-beneficio del sistema, asegurando su viabilidad como adquisición clínica y garantizando su eficacia en el ámbito médico.

2.4 Justificación

Desde el punto de vista social, a pesar de que los sistemas antes descritos han demostrado su eficacia en la práctica clínica real, en México, el acceso a esta tecnología es muy limitado a pocas instituciones de salud. En general, en el sector público solo algunas instituciones de alta especialidad cuentan con esta tecnología, por lo que resulta motivante el desarrollo de tecnología nacional como alternativa. Por otro lado, esto motiva la investigación y desarrollo de tecnología quirúrgica en nuestro país.

Desde el punto de vista de desarrollo de ingeniería, los siguientes puntos justifican el presente trabajo de tesis:

- Se explora la mejora de un algoritmo de orientación absoluta no iterativo para alinear un objeto dentro de cualquier entorno virtual.
- Se requiere desarrollar un algoritmo capaz de estimar la postura que debe adoptar la guía física proporcionada por el robot para alinear la herramienta sobre la trayectoria del tornillo pedicular que será implantado.
- Se busca la integración del sistema de navegación y el sistema robótico para atender las necesidades de las cirugías de mínima invasión asistidas por robot, correspondientes a la planeación, visualización y ejecución.

2.5 Metas

- Desarrollar el software de planeación, lo cual involucra:
 - Desarrollar un entorno virtual en cuya escena, el médico especialista pueda elaborar un plan preoperatorio ideal y específico, correspondiente al procedimiento que se realizará.
 - La escena debe contar con los modelos virtuales de las herramientas reales, por lo que es posible reutilizar de los modelos 3D con los que cuenta el laboratorio de BioInstrumentación del Instituto de Ciencias Aplicadas y Tecnología (ICAT).

- Desarrollar el subsistema correspondiente al navegador, lo que implica:
 - Dominar las técnicas de registro basado en marcadores usando una cámara de profundidad para seguir cada herramienta y elemento relevante dentro de la escena quirúrgica [14].
 - Desarrollar un algoritmo de alineación capaz de estimar los parámetros de localización (posición y orientación) de cada herramienta y visualizar su correspondiente reconstrucción tridimensional dentro del entorno virtual.
- Desarrollar un sistema robótico dotado de gran rigidez, capaz de orientar y posicionar milimétricamente el efector final del robot [15], [16], en el cual, será montada una herramienta quirúrgica para llevar a cabo la punción.
 - Generar un sistema de control básico de lazo cerrado que permita mantener la posición angular deseada en cada articulación actuada del mecanismo.
- Integrar los sistemas desarrollados para complementar la destreza manual del médico cirujano con el robot y brindar una guía visual de las herramientas, trayectorias definidas y estructuras anatómicas no visibles en la escena quirúrgica mediante el navegador.

2.6 Alcances

Es necesario desarrollar cada sistema individualmente y evaluar su precisión; culminando con un entorno controlado de laboratorio, donde el sistema integrado debe ser capaz de posicionar una herramienta de punción sobre una trayectoria específica respecto de una reproducción artificial del órgano objetivo.

2.7 Hipótesis

Es posible desarrollar un algoritmo con un enfoque distinto a los convencionales que permita al sistema de seguimiento estimar los parámetros de localización (posición y orientación) de las herramientas y objetos relevantes para la escena quirúrgica.

Un sistema de seguimiento aplicado a un robot es capaz de mejorar los tiempos de ejecución y disminuir la necesidad de equipo quirúrgico que emita radiación, brindando beneficios tanto para el paciente como para el médico en cirugías de mínima invasión.

El desarrollo de un robot de configuración híbrida con cinemática desacoplada puede tener un buen desempeño y mejorar la relación costo-beneficio, posibilitando la adopción a mediano plazo de enfoques quirúrgicos asistidos por robots.

2.8 Metodología

La integración del sistema robótico comprende el desarrollo de dos subsistemas.

2.8.1 Navegador

En Unity se desarrolló un entorno virtual que recrea la escena quirúrgica, con los elementos 3D de cada herramienta y se agregó una interfaz que facilita la libertad de movimiento para visualizar desde diversas perspectivas el modelo 3D de la sección lumbar de una columna vertebral reconstruida a partir de imágenes médicas [17].

Los modelos 3D empleados en este trabajo de tesis son producto de colaboraciones anteriores entre el laboratorio de BioInstrumentación del Instituto de Ciencias Aplicadas y Tecnología y el Hospital General de México “Dr. Eduardo Liceaga”.

A partir de modelos 3D diseñados en software de CAD, se fabricaron placas de acrílico que cuentan con una distribución de perforaciones basadas en conjuntos de imágenes generadas por software. Cada imagen tiene cuatro posiciones únicas de círculos, que simbolizan los marcadores esféricos de los marcos de referencia exclusivos para cada herramienta relevante. Estos marcadores se integran a las herramientas u objetos relevantes tanto en la en la escena quirúrgica real como en la virtual.

Se utilizó un sistema de cámaras infrarrojas con visión estéreo que emite luz infrarroja de baja intensidad a la escena quirúrgica, la cual es devuelta por los marcadores esféricos, permitiendo a la cámara detectar y estimar la posición de las herramientas u objetos respecto del origen de la cámara, permitiendo registrarlos en el entorno virtual.

Al observar el comportamiento y necesidades del sistema de cámaras, se desarrolló un algoritmo de orientación absoluta capaz de alinear el modelo 3D de las herramientas en el entorno virtual.

El funcionamiento del algoritmo se basa en encontrar una base ortonormal intrínseca a la distribución única de n esferas para cada marco de referencia [18], posteriormente se calculan los ángulos de Euler a partir de la matriz de rotación de la base ortonormal, permitiendo orientar un objeto 3D con fidelidad usando los parámetros de localización en cualquier entorno virtual considerando la convención que utiliza el entorno gráfico (dextrógiro o levógiro).

Las virtudes del algoritmo son:

- El método desarrollado está basado en la distribución de esferas del marco de referencia, y puede manejar más de tres posiciones de marcadores.
- El método no es iterativo, no pretende calcular la transformación que será aplicada a todos los puntos de un grupo y minimizar el error.
- El método busca conocer la base ortonormal intrínseca a la distribución de esferas que conforman el marco de referencia teniendo en cuenta tres o más posiciones de esferas.

2.8.2 Robot

Se diseñó y construyó un robot manipulador de configuración híbrida (paralelo-paralelo) de 4GDL [15], [16] como complemento a la destreza manual del cirujano.

El mecanismo se desacopló cinemáticamente agrupando paralelamente dos robots planares, donde cada uno cuenta con 2 grados de libertad (GDL), lo que simplificó el análisis cinemático del robot, necesario para conocer las ecuaciones de las articulaciones actuadas que rigen el movimiento del efector final del robot [15], [19], [20].

En la base móvil de ambos robots planares, se instaló un mecanismo de giro *Gimbal*. Dado que la posición de la base móvil de cada robot puede ajustarse de manera independiente, la localización del efector final del robot queda determinada por dos puntos ubicados en el cruce con los ejes de rotación del *Gimbal* ubicados al centro de la base móvil.

Al robot se acoplaron motores DC de la marca Maxon, la cual es reconocida por su excelencia en la fabricación de dispositivos de alta calidad, se conectaron tarjetas de control EPOS4 de la misma marca, garantizando la confiabilidad y robustez del sistema.

2.8.3 Integración del sistema

Al entorno virtual en Unity se importaron los modelos 3D, archivos con extensión .obj correspondientes a los objetos relevantes de la escena.

Se generó un sistema de planeación preoperatoria, donde el usuario puede recorrer una trayectoria a lo largo del modelo 3D de la columna mediante un control deslizante y colocar en la mejor disposición dos medidas de tornillos utilizando los botones de la interfaz.

Se desarrolló un algoritmo que estima la posición óptima que el robot debe adoptar para poder implantar un tornillo pedicular en la columna del paciente, la estimación se calcula utilizando la información de localización de cada elemento detectado por el sistema de seguimiento y presenta un resultado visual en un entorno virtual con los modelos 3D, donde se puede observar la intersección entre la trayectoria del tornillo y el plano de acción del robot.

De modo que al término de la integración se cuenta con una herramienta de precisión para la colocación de tornillos pediculares, asegurando que estos no penetren la pared medial del pedículo y así prevenir posibles lesiones neurológicas durante los procedimientos de columna, particularmente en el tratamiento de la escoliosis.

Previo a la integración del sistema completo, fue necesario desarrollar los sistemas individualmente, a continuación, se detalla cada sistema.

3.- Sistema de seguimiento

El sistema de seguimiento se encarga de:

1. Registrar la posición de los marcadores esféricos detectados por el sistema de cámaras.
2. Estimar los parámetros de localización (posición y orientación) de cada herramienta registrada.
3. Desplegar una guía visual que es el resultado de graficar las herramientas virtuales en un entorno 3D de acuerdo con los parámetros de localización obtenidos.

El diagrama de la Figura 5 refleja los dispositivos que intervienen en el sistema de seguimiento.

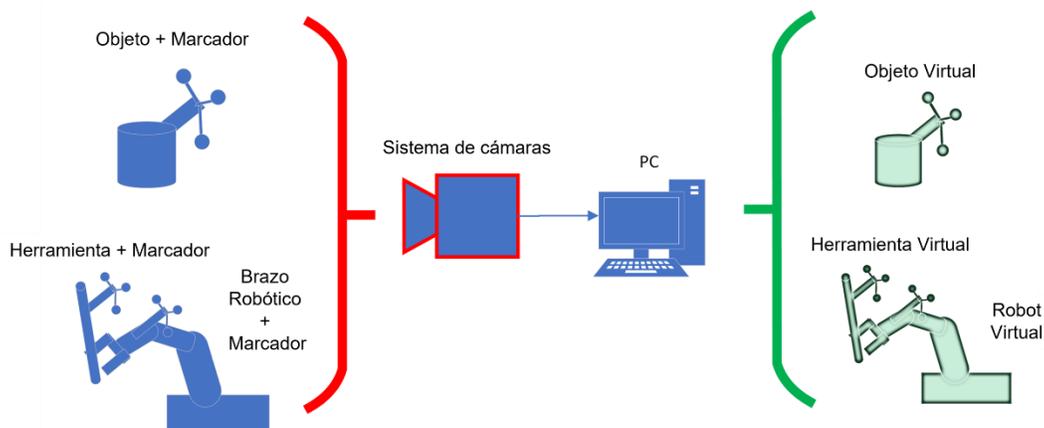


Figura 5 Diagrama de funcionamiento del sistema de seguimiento

El diagrama de la Figura 6 refleja el flujo de información que hay entre los distintos bloques que conforman al sistema de seguimiento.



Figura 6 Diagrama de flujo de información del sistema de seguimiento

El flujo de información comienza con la detección de objetos relevantes del entorno real que estén dentro del campo de visión del sistema de cámaras de profundidad.

Dado que cada objeto relevante en la escena tiene asociado un marco de referencia con marcadores esféricos, el sistema de cámaras registra y refiere el vector de posición correspondiente al marcador detectado con respecto al sistema de coordenadas global de la cámara [21].

Con el fin de desplegar la guía visual en el entorno de traficación 3D, se solicita al sistema de cámaras la posición de cada marcador registrado. Una vez recibida, esta información se procesa de inmediato mediante un algoritmo de alineación. Como resultado, se obtienen los parámetros de localización $(x, y, z, \phi, \theta, \psi)$ necesarios para la correcta representación y visualización de las herramientas en un instante dentro del entorno virtual.

Unity es usado únicamente como entorno de graficación 3D para desplegar la representación virtual de los objetos reales que tienen asociado un marco de referencia, mientras que el algoritmo de alineación que calcula los parámetros de localización (posición y orientación) se implementó usando un servidor en Python, de modo que el algoritmo puede ser implementado en cualquier entorno de visualización 3D.

El diagrama de la Figura 7 muestra la interacción entre cliente y servidor usando comunicación por WebSockets. Se utilizó este tipo de comunicación ya que estos establecen una conexión TCP persistente entre el cliente y el servidor, lo que garantiza la entrega de datos correctamente, además ofrecen una comunicación fiable y de tipo bidireccional.

La funcionalidad de cliente se programó en lenguaje C# y se ejecuta dentro del entorno de Unity, mientras que el servidor se programó usando Python y se ejecuta en una terminal, de modo que el servidor responde a las solicitudes que realiza Unity como cliente.

La frecuencia de operación del sistema de cámaras es de 120 Hz, mientras que la respuesta del servidor es de 200 Hz, por lo que, la comunicación entre ambos módulos funciona correctamente.

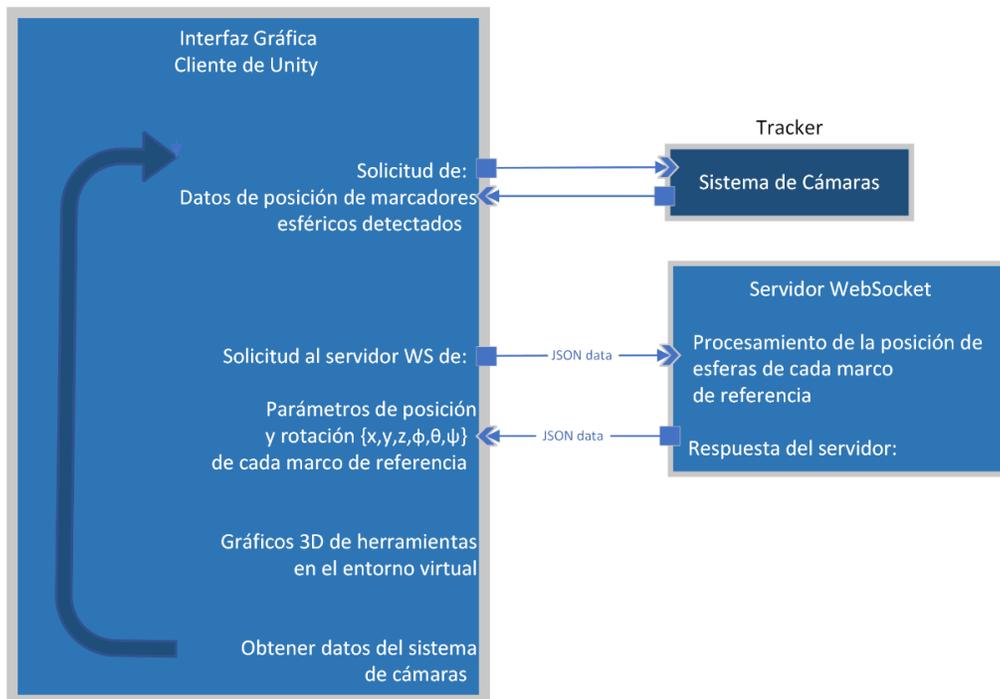


Figura 7 Diagrama de comunicación cliente – servidor del algoritmo de alineación

3.1 Modelado de marcos de referencia

Se desarrolló un software usando el entorno de desarrollo *Processing* para generar aleatoriamente círculos en 4 cuadrantes del plano X,Y de la imagen, la posición de los círculos simboliza la distribución de las esferas de alta reflectancias que compone al marco de referencia.

Posterior a la generación de 100 imágenes con distribuciones aleatorias, se seleccionaron manualmente 3 imágenes cuyas distribuciones fueran visualmente diferentes entre sí.

Sumando las imágenes de las distribuciones seleccionadas se generó una imagen con el total de casillas disponibles del marco de referencia, aplicando operaciones morfológicas [22] y redondeando las posiciones de las casillas a múltiplos de 10mm se generó un marco de referencia con medidas estandarizadas para que el usuario final tenga una mejor experiencia al definir los marcos necesarios.

La Figura 8 muestra un ejemplo de la pieza generada en el software CAD con las distribuciones empleadas para la generación de un marco de referencia.

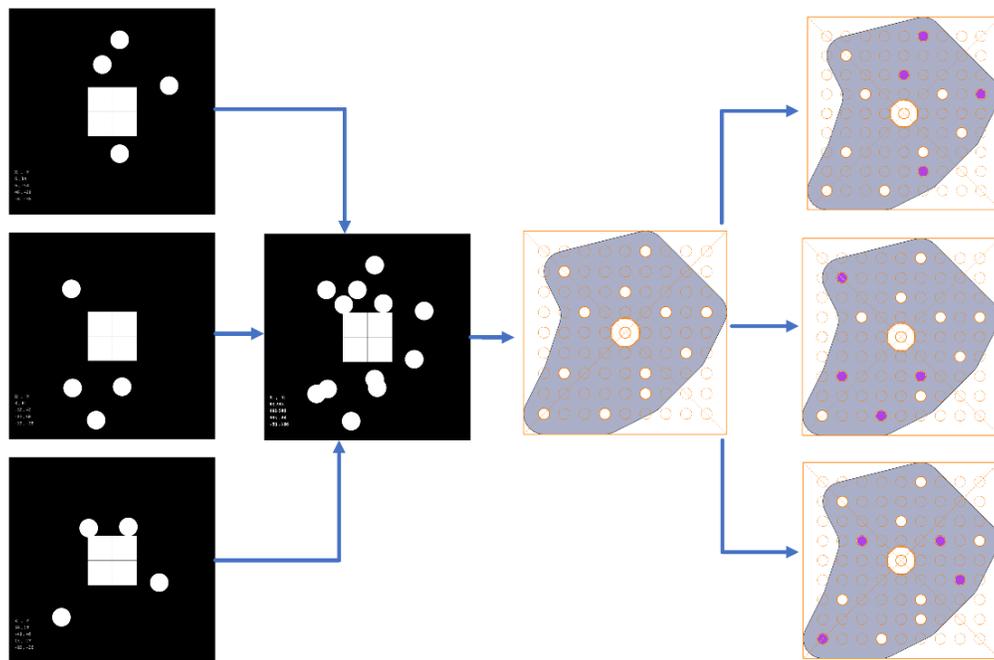


Figura 8 Diagrama del proceso de generación de marcos de referencia

Además de la herramienta de generación de marcos de referencia se desarrolló una interfaz gráfica de usuario para apoyar en la generación de archivos tipo *JSON* con las posiciones del marco de referencia molde (o modelo ideal) que requiere el servidor para ejecutar el algoritmo de alineación de las herramientas.

El software permite al usuario usar diferentes plantillas y establecer fácilmente una distribución a cada marco de referencia, únicamente requiere agregar un nombre y seleccionar las casillas que serán ocupadas por las esferas que conforman al marco de referencia ideal como se muestra en la Figura 9.

Es importante resaltar que, para todos los marcos de referencia, el vector origen $(0,0,0)$ del sistema coordenado de la herramienta se encuentra al centro de cada plantilla dentro del hexágono.

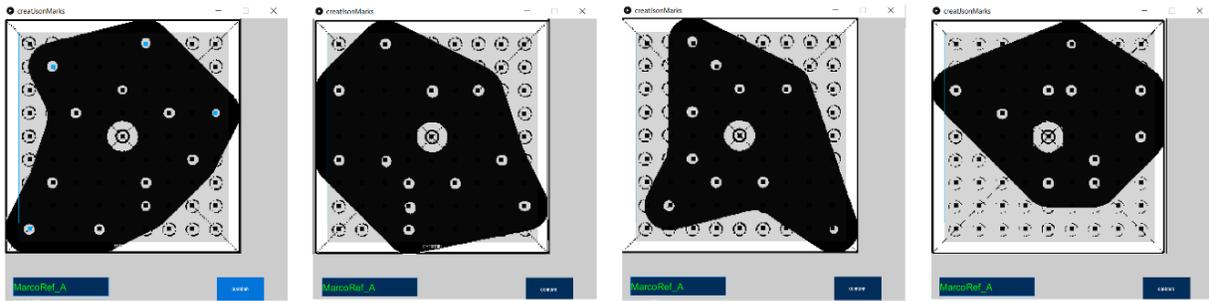


Figura 9 Interfaz gráfica de usuario para generar archivos JSON

La Figura 10 muestra un marco de referencia fabricado en acrílico compuesto por una distribución única de marcadores esféricos altamente reflectantes y el correspondiente archivo JSON conformado por el formato *Clave-Valor* que fue generado por el software de acuerdo con la disposición seleccionada mediante la interfaz; cada clave *Sphere* del archivo JSON hace referencia a las coordenadas del centro de cada esfera respecto al origen del marco de referencia. El formato JSON se usó debido a la diversidad de librerías existentes dada su popularidad.

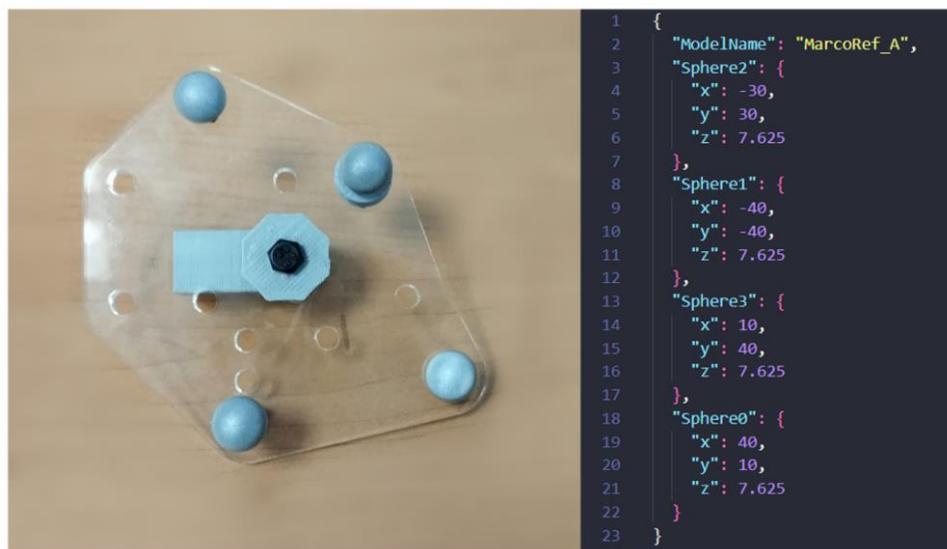


Figura 10 Marco de referencia y estructura del archivo JSON

3.2 Sistema de registro con cámaras

El sistema de seguimiento [21] utiliza el sistema estéreo de cámaras infrarrojas *V120 Trio* de la marca *Optitrack*, al cual nos referiremos de ahora en adelante como *tracker*, el sistema cuenta con:

1. Tres cámaras infrarrojas
2. Anillos de iluminación led de luz infrarroja
3. Interfaz de comunicación USB
4. Frecuencia de operación de 120 Hz
5. Interfaz gráfica *Motive* para el registro y visualización de los marcos de referencia

La Figura 11 muestra el sistema de cámaras.



Figura 11 Sistema de cámaras V120 Trio

El *tracker* posibilita rastrear la postura de las herramientas y objetos relevantes dentro de la escena mediante la detección de los marcadores esféricos, ya que, cada herramienta tiene asociado un marco de referencia compuesto por una distribución única de tres o más esferas completamente establecida.

Optitrack cuenta con una interfaz gráfica llamada *Motive* que permite registrar marcos de referencia y asignarles un nombre en la base de datos del sistema, este proceso es esencial para clasificar por nombre un grupo de esferas en tiempo de ejecución.

El proceso de rastreo inicia con la detección de las esferas, para ello, el *tracker* emite luz infrarroja hacia la escena, la luz es reflejada de regreso por los marcadores esféricos y captada por las cámaras del *tracker*, la posición de cada marcador detectado es referida al sistema de coordenadas del *tracker* como muestra el diagrama de la Figura 12.

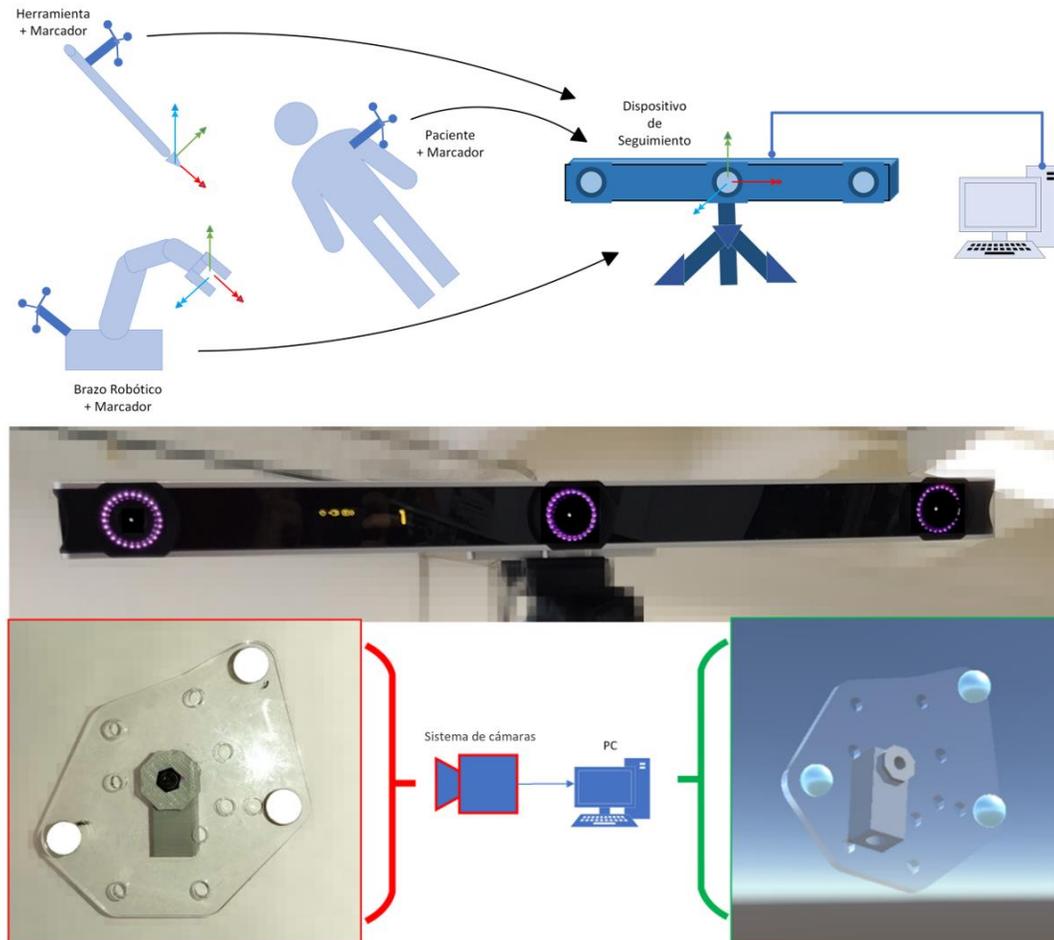


Figura 12 Diagrama de funcionamiento del sistema de cámaras y vista real - virtual de un marco de referencia

La clasificación de información se lleva a cabo mediante la comparación de las distribuciones detectadas y las distribuciones de los modelos registrados en la base de datos del sistema, de modo que, los conjuntos de marcadores detectados son agrupados y nombrados de acuerdo al modelo coincidente.

Es posible solicitar al *tracker* información sobre las herramientas registradas mediante una conexión IP al servidor local, la respuesta comprende los vectores de posición de cada esfera y el respectivo nombre del modelo al que es relacionado.

Durante el proceso de registro de una nueva herramienta en la base de datos del software *Motive*; se calcula el centroide del conjunto de marcadores y se le adjunta un sistema de coordenadas cuya orientación es idéntica a la del sistema de coordenadas del *tracker*, sin importar la orientación de la herramienta como se muestra en la Figura 13.

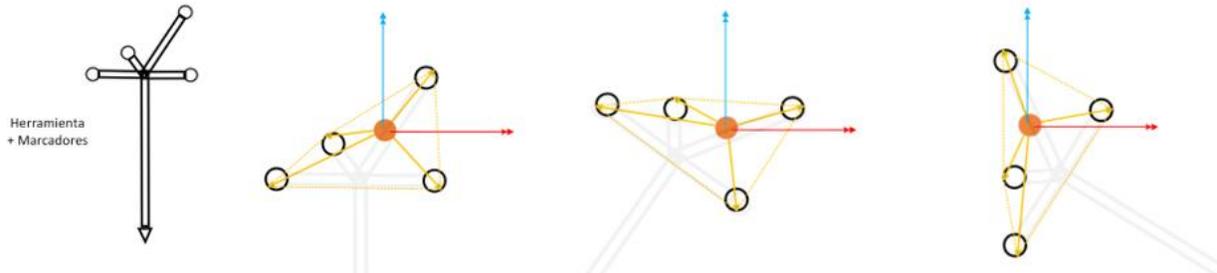


Figura 13 Diagrama de distribuciones con sistema coordinado asignado por Motive

Dado que es muy complicado lograr que el marco de referencia adopte una postura completamente paralela al *tracker*, se busca resolver el problema de alineación mediante un algoritmo capaz de encontrar un sistema de coordenadas intrínseco de la distribución de marcadores, para hacer corresponder la posición y orientación de cada herramienta con la estimación proporcionada por el sistema de cámaras.

En la sección de Anexos 8.2 se mencionan algunos de los algoritmos de registro y alineación más recurrentes en la literatura, incluyendo el algoritmo de orientación absoluta [18], [23] que fue la base para la implementación que se presenta a continuación, la cual omite el parámetro de escala del algoritmo original. Se adoptó este enfoque debido a que en [24] se reporta una similitud con el algoritmo de orientación absoluta [18], mostrando buen rendimiento al analizar conjuntos pequeños de puntos y superando a los algoritmos iterativos en grandes conjuntos de puntos.

3.3 Estimación de la rotación del marco de referencia (Algoritmo de Alineación)

El algoritmo que se propone está basado en el algoritmo de orientación absoluta [18]. Su objetivo es calcular los parámetros de posición y orientación definidos por un conjunto de puntos o distribución que tiene una cantidad de n esferas. Estas esferas son detectadas por el *tracker*, se procesan y comparan los parámetros obtenidos con los parámetros del respectivo conjunto de posiciones ideales de cada marco de referencia que fueron modelados previamente como se mostró en la sección 3.1 [21], [25]. Se destaca que los parámetros calculados se usan después para alinear el modelo 3D de la herramienta correspondiente en un entorno virtual creado en Unity.

El algoritmo es el siguiente:

1. Calcular el promedio de los vectores normales de la distribución.
2. Establecer los vectores unitarios que conforman la base ortogonal propia de la distribución.
3. Extraer los ángulos de Euler de la matriz de rotación.
4. Estimar el vector de posición de la herramienta.

3.3.1 Calculo del vector normal promedio

Para cada i -ésima esfera se calcula un vector normal al plano generado por dos vectores, posteriormente se promedian los vectores normales calculados para obtener un único vector propio de la distribución de acuerdo con el algoritmo.

Como ejemplo tenemos plasmadas en diferentes vistas una distribución única de cuatro esferas que muestra la Figura 14, resaltando en color rojo el vector unitario X, el verde el vector Y y en azul el vector Z del sistema de coordenadas al que están referidas las posiciones de cada esfera.

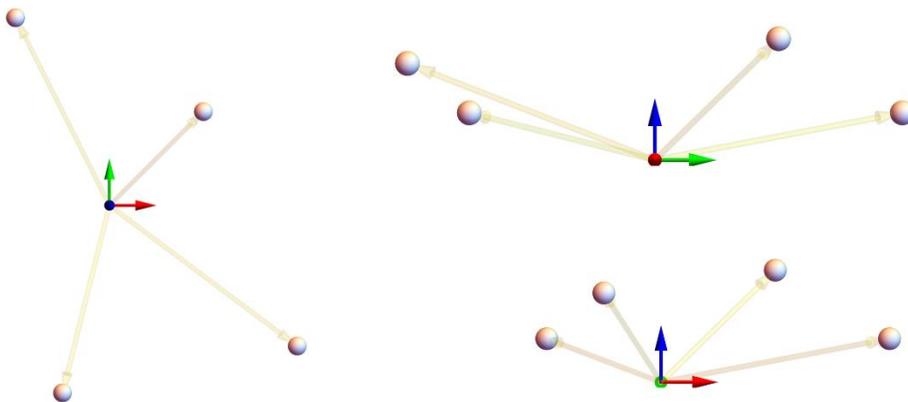


Figura 14 Vistas de la distribución de esferas referidas al sistema coordenado global

Inicialmente se calcula la magnitud de los vectores que parten de la i-esima esfera que está siendo evaluada y se extiende hacia la k-esima de las esferas restantes esferas, posteriormente se comparan las magnitudes, se seleccionan los vectores máximo y mínimo involucrando todas las n esferas que conforman la distribución, como se muestra en las ecuaciones Ec 1, Ec 2 y en la Figura 15 .

$$v_{i_min} = v_{i_min} > ||v_k - v_i||_{k=1}^n ; i \neq k \quad \dots (Ec 1)$$

$$v_{i_max} = v_{i_max} < ||v_k - v_i||_{k=1}^n ; i \neq k \quad \dots (Ec 2)$$

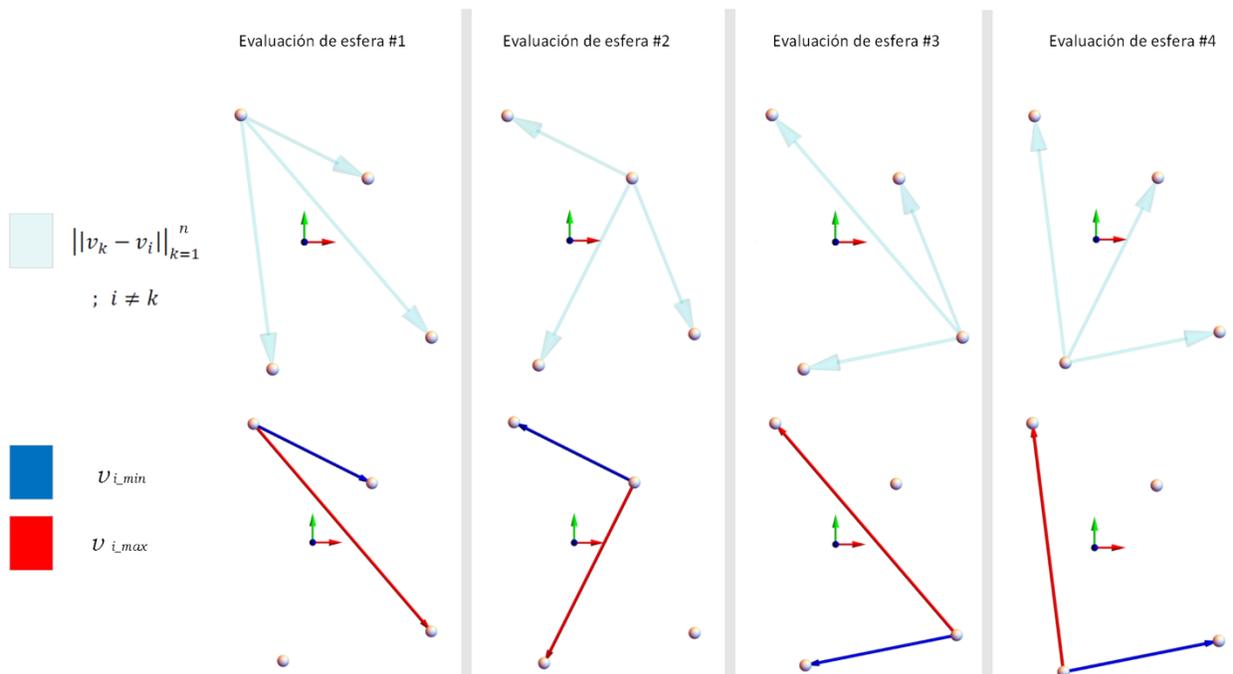


Figura 15 Gráficas de la comparación entre vectores de acuerdo con la esfera evaluada

Una vez definidos los vectores máximo y mínimo, para la i-esima esfera se calcula el producto cruz entre ambos vectores, obteniendo el vector **N** normal al plano definido por tres puntos (Ec 3), como se muestra en Figura 16.

$$N_i = v_{i_max} \times v_{i_min} \quad \dots (Ec 3)$$

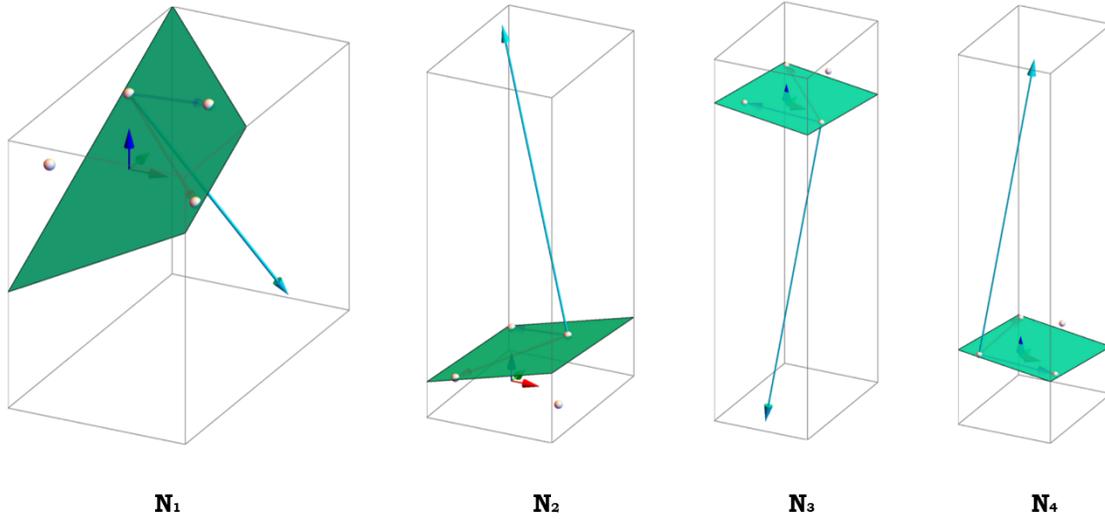


Figura 16 Gráficas del vector normal al plano para cada esfera evaluada

Usando la ecuación Ec 4 se calcula la media de los vectores normales N_i referidos al sistema inercial, se obtiene como resultado un único vector derivado de la información del total de esferas como se muestra en Figura 17.

$$N_{promedio} = \frac{1}{n} \sum_{i=1}^n N_i \quad \dots (Ec 4)$$

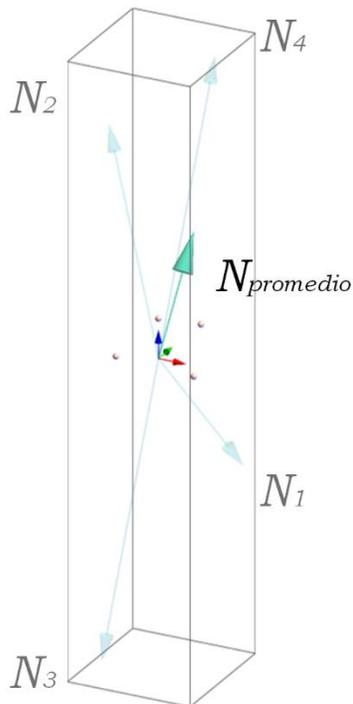


Figura 17 Gráfica del vector normal promedio

3.3.2 Asignación de versores

El cálculo de los versores (vectores unitarios) permite conocer la base ortogonal intrínseca a la distribución, mediante la cual es posible describir la orientación del marco de referencia; el sistema de coordenadas generado tiene su origen en el centroide de la distribución de esferas, el cual es calculado usando la ecuación Ec 5.

$$v_c = \frac{1}{n} \sum_{i=1}^n v_i \quad \dots (Ec 5)$$

Donde:

- v_c es el centroide de la distribución
- v_i es el vector de posición de la i -ésima esfera, referido al sistema global
- n es el número total de esferas

Arbitrariamente se considera al vector unitario X como el vector más cercano al centroide; la Figura 18 ejemplifica la asignación del versor X de color rojo.

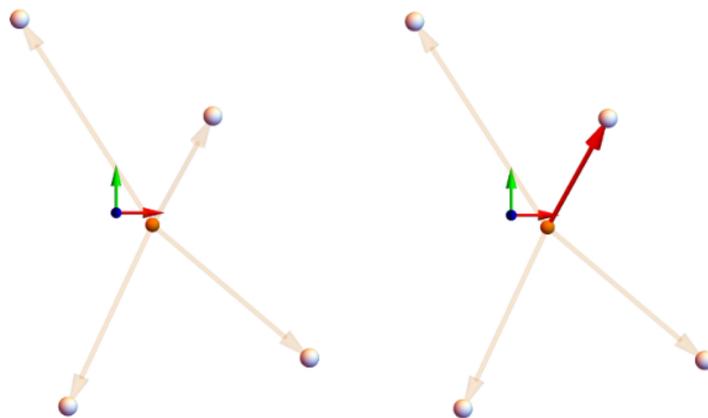


Figura 18 Gráficas de la distribución de esferas y asignación del versor "x"

Para conocer el vector unitario Y se calcula el producto cruz del versor X y el vector normal promedio $N_{promedio}$, de modo que el versor Y es normal al plano generado y ortogonal a ambos vectores como muestra la Figura 19 en color verde.

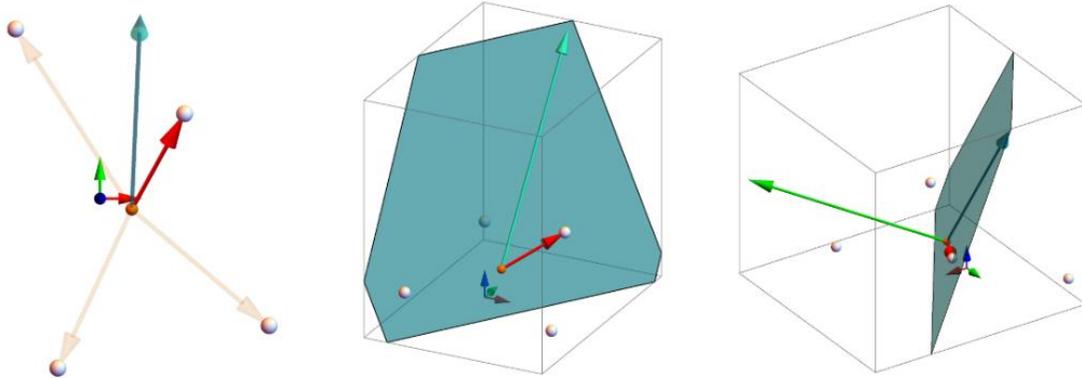


Figura 19 Gráficas de generación del versor "y"

El vector unitario Z se calcula aplicando el producto cruz del vector X y el Y, de modo que Z es normal al plano generado por ambos versores.

Una vez que se cuenta con los vectores unitarios normalizados con origen en el centroide es posible considerarla una base ortogonal propia de la distribución (de acuerdo al algoritmo), que define la orientación de la distribución respecto del marco de referencia como se muestra en la Figura 20 , donde, las posiciones de cada esfera se describen respecto a dicha base. Usando los versores (3x1) de la base ortogonal generada se construye su matriz de rotación, que describe la orientación del sistema generado respecto del sistema inercial al que originalmente estaban referidas las esferas.

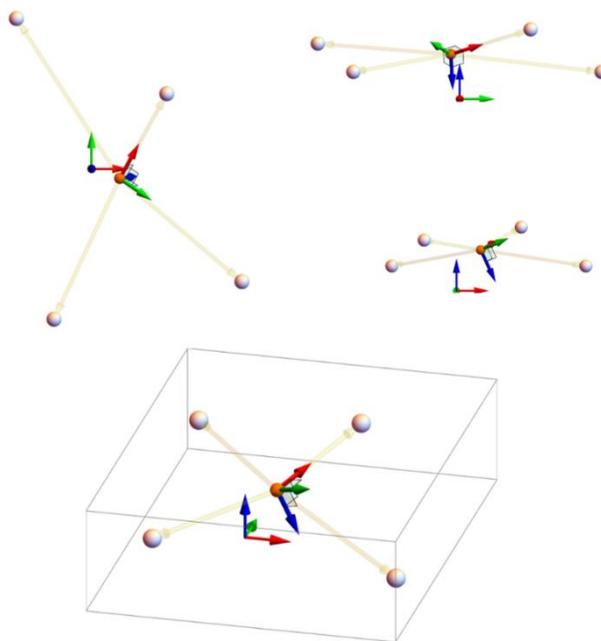


Figura 20 Gráficas 3D del sistema de coordenadas intrínseco a la distribución asociado al centroide

Debido a que cada entorno de graficación puede tener su propia composición de rotaciones para establecer la orientación de un objeto 3D, es necesario resolver el problema inverso mediante el cálculo de los ángulos de Euler, que corresponden (según la composición) a una secuencia de rotaciones sobre ciertos ejes, y genera a la matriz de rotación correspondiente a la orientación de un objeto.

3.3.3 Cálculo de los ángulos de Euler

El cálculo de los ángulos de Euler permite conocer los ángulos que producen una matriz de rotación de acuerdo con una secuencia de rotaciones establecida. Existen distintas convenciones de ángulos de Euler para obtener una matriz de rotación, en este trabajo se utiliza la composición de rotaciones Cardaniana ZYX [26].

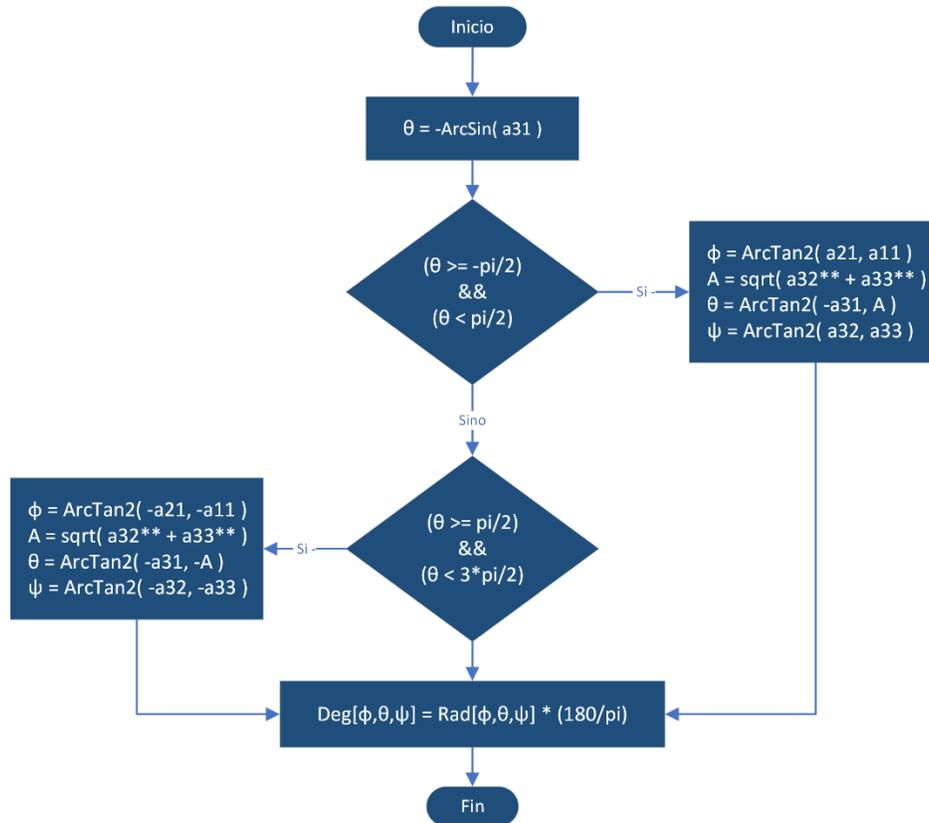
De modo que, se busca conocer el valor de los ángulos de Euler ϕ , ϑ , ψ que aplicados secuencialmente de acuerdo a la composición ZYX producen la matriz de rotación que se está analizando.

La Figura 21 muestra la matriz de rotación resultante de la aplicación secuencial de rotaciones con composición $Z(\psi)$, $Y(\vartheta)$, $X(\phi)$.

$$\begin{aligned}
 RotZ(\psi) \cdot RotY(\vartheta) \cdot RotX(\phi) &= \begin{pmatrix} \cos \theta \cos \psi & \cos \psi \sin \theta \sin \phi - \cos \phi \sin \psi & \cos \phi \cos \psi \sin \theta + \sin \phi \sin \psi \\ \cos \theta \sin \psi & \cos \phi \cos \psi + \sin \theta \sin \phi \sin \psi & -\cos \psi \sin \phi + \cos \phi \sin \theta \sin \psi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{pmatrix} \\
 &= \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}
 \end{aligned}$$

Figura 21 Matriz de rotación con secuencia ZYX y nomenclatura de elementos

Teniendo en cuenta las posiciones de la matriz y aplicando el algoritmo que se muestra en la Figura 22, es posible calcular los ángulos de Euler de la matriz de rotación.



$$f(\text{[Matriz de rotación]}) = \{ \phi, \theta, \psi \}$$

Figura 22 Diagrama de flujo del algoritmo para cálculo de ángulos de Euler

3.4 Estimación de la posición del marco de referencia (Algoritmo de Alineación)

El proceso de estimación de la posición consta de dos etapas:

1. Cálculo del vector $iOd_{[D]}$ (que parte del centroide y se extiende hasta el centro del marco de referencia), para cada distribución al inicializar el servidor.
2. Cálculos de la estimación de la posición de cada herramienta en tiempo de ejecución respecto del sistema de coordenadas T .

Al inicializar el servidor se leen los archivos *JSON* disponibles referentes a cada marco de referencia y se realiza lo siguiente.

- Suponiendo que M y D son bases ortogonales con origen en Om y Od respectivamente.
- Se calcula el vector de posición $Od_{[M]}$ correspondiente al centroide de la distribución referido al sistema de coordenadas M .
- Se calculan los versores del sistema de coordenadas D de la distribución con origen en el centroide $Od_{[M]}$.

- Se calcula $iOd_{[M]}$ que es el vector opuesto de $Od_{[M]}$ el cual está referido al sistema de coordenadas M , parte del origen Om (se considera como vector deslizante).
- Se aplica un cambio de base para conocer las coordenadas del vector $iOd_{[M]}$ referido al sistema D , de modo que se calcula el producto punto del vector $iOd_{[M]}$ y la matriz inversa de la matriz P (la cual está formada por los versores del sistema D , los cuales a su vez están definidos respecto de la base M), dando como resultado el vector nombrado $iOd_{[D]}$ (que bien podría ser el vector que hay de Od hacia Om respecto de la base D) como se muestra en la Figura 23.

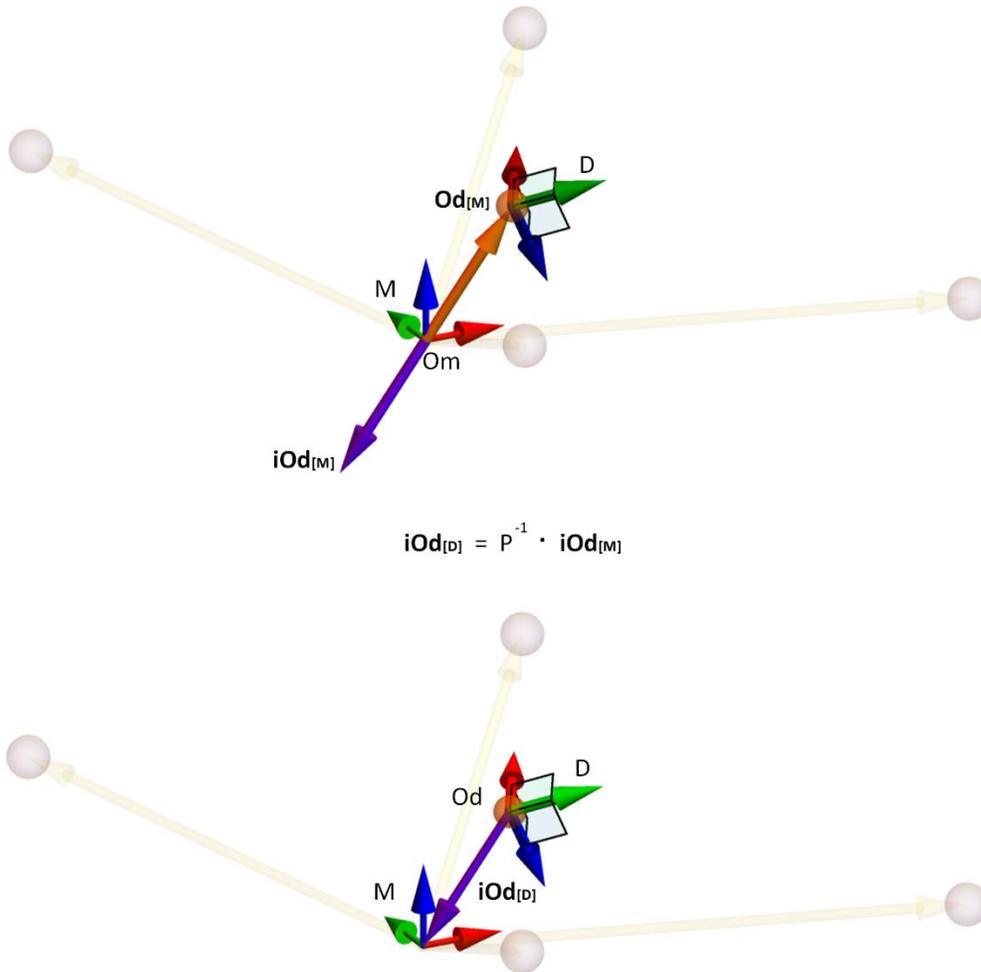


Figura 23 Gráficas 3D de la generación del vector $[iOd]$ referido a la base M y D

Durante el tiempo de ejecución del sistema de seguimiento, se estima el vector de posición de cada marco de referencia llevando a cabo los puntos anteriores. La diferencia es que no se usa la información ideal contenida en los archivos *JSON*, esta vez se usa la información que proporciona el *tracker*, de modo que se calcula lo siguiente:

- Se calcula el centroide de la distribución referido al sistema de coordenadas del *tracker* $O_{c[T]}$.
- Se calcula la base C intrínseca a la distribución correspondiente que proporciona el *tracker*.
- Se calcula el vector $v_{A[T]}$ mediante el producto punto del vector $iO_{d[D]}$ (calculado al inicializar el servidor) y la matriz de rotación Q (descrita por los versores de la base C respecto de la base T).
- Se calcula el vector $v_{Z[T]}$ que define la posición del centro del marco de referencia respecto del sistema T , para ello se suma el centroide de la distribución $O_{c[T]}$ y el vector $v_{A[T]}$ como se muestra en la Figura 24.

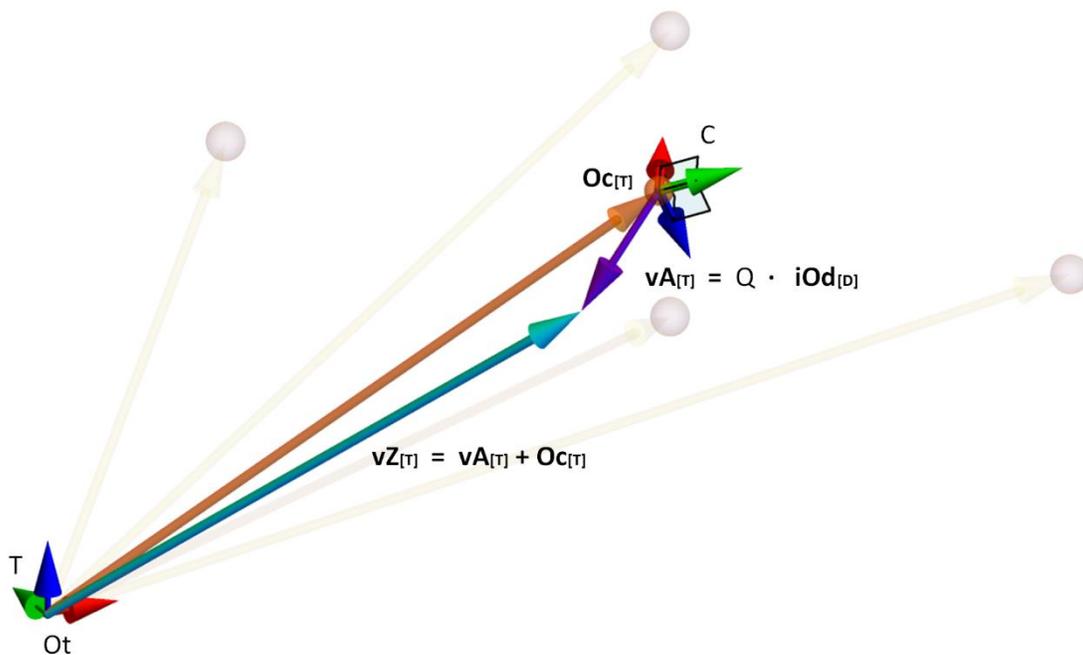


Figura 24 Gráficas 3D de la estimación de la posición del marco de referencia

Empleando el algoritmo de alineación es posible calcular los parámetros $\{\phi, \vartheta, \psi\}$ correspondientes a la orientación del marco de referencia y su posición $\{x, y, z\}$ respecto del sistema inercial global asociado al sistema de cámaras infrarrojas de visión estéreo, de modo que sin importar el entorno de graficación, es posible reconstruir la posición de cada herramienta u objeto relevante en la escena quirúrgica que tenga asociado un marco de referencia.

En el capítulo de resultados se detalla la implementación del algoritmo en un entorno controlado para conocer su precisión.

4.- Software de planeación

En Unity se desarrolló un software para generar un plan preoperatorio, la escena cuenta con diversas texturas y vistas auxiliares del modelo 3D de una columna vertebral generado a partir de imágenes médicas de Tomografía de un paciente real, el cual fue provisto por el equipo de trabajo del Laboratorio de BioInstrumentación del Instituto de Ciencias Aplicadas y Tecnología.

El objetivo del software es brindar una herramienta de apoyo al médico cirujano en la fase de planificación previa a la operación quirúrgica, facilitando al cirujano la visualización y el análisis que debe realizar basándose en los estudios de imagen del paciente.

Dentro del software, el medico experto debe establecer la postura ideal de los tornillos necesarios para lograr un procedimiento exitoso, este procedimiento es de gran relevancia porque, durante la fase intraoperatoria el robot colocará la herramienta de punción sobre la trayectoria del tornillo que será implantado.

La escena cuenta con diversas funcionalidades:

- Agregar dos tamaños de tornillos pediculares.
- Trasladar y rotar el tornillo para posicionarlo en la mejor disposición respecto del modelo 3D de la columna de acuerdo con las especificaciones del médico experto.
- Recorrer una trayectoria a lo largo de la columna definida por los puntos de control de una curva de Bézier Cubica.

La Figura 25 presenta la interfaz del software, destacando en color rojo los elementos interactivos de la interfaz que permiten el acceso a diversas funcionalidades. Al presionar los botones identificados como (B) se pueden añadir dos tipos de tornillo con longitudes específicas, y al manipular el control deslizante (A) la cámara se desplaza a lo largo del modelo 3D de la columna, facilitando la búsqueda de la ubicación óptima para la implantación del tornillo con una visualización de cefálico a caudal [1].

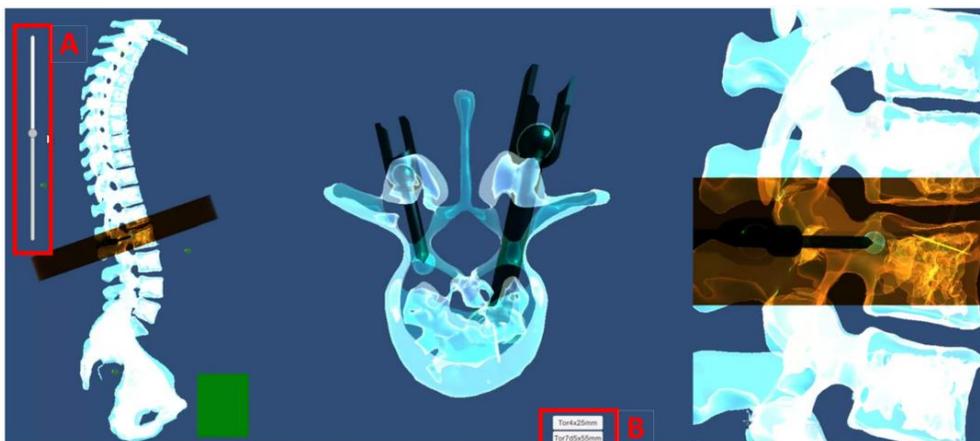


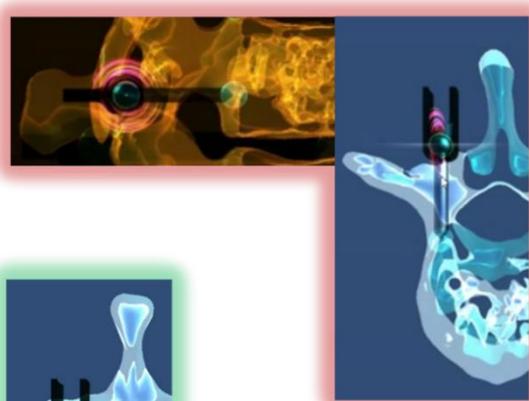
Figura 25 Vista de la interfaz gráfica de usuario del software de planeación

La localización de los tornillos respecto del origen de la columna se exporta en formato JSON para ser usado en el proceso intraoperatorio durante el despliegue del sistema general.

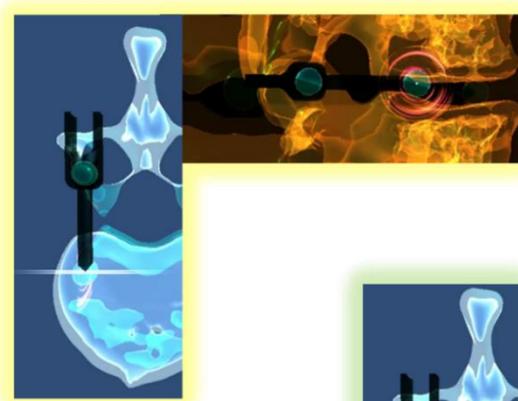
La Figura 26 muestra segmentos editados de las diferentes vistas del software y son separados en dos bloques sombreados, los cuales hacen referencia a rotaciones sobre un eje global específico (X o Y) y a un punto de rotación sobre la cabeza o la punta del tornillo.

En cada sección se observan marcas circulares que aparecen dentro de la interfaz y es posible visualizarlas en una de las diferentes vistas, estas marcas circulares señalizan la rotación del tornillo pedicular en los ejes globales de la escena X o Y, la rotación del tornillo puede efectuarse respecto de la cabeza o en la punta del tornillo.

Eje de rotación: X / Pivote: Cabeza de tornillo



Eje de rotación: X / Pivote: Punta de tornillo



Eje de rotación: Y / Pivote: Cabeza de tornillo



Eje de rotación: Y / Pivote: Punta de tornillo

Figura 26 Fragmentos que muestran la rotación del tornillo en los ejes globales

5.- Integración del sistema robótico

El sistema de seguimiento permite conocer la localización de los diversos objetos dentro del campo de visión de la cámara, entre los objetos relevantes está la columna del paciente y el sistema robótico cuyo efector final se alinea con la trayectoria (previamente establecida en la etapa de planeación) requerida para implantar un tornillo pedicular en la columna, de modo que es necesario:

- Conocer la posición de la columna y el robot respecto del *tracker*.
- Conocer la posición de un tornillo respecto de la columna.
- Estimar la posición que el robot debe adoptar para alinearlo con la trayectoria del tornillo que se pretende implantar en la columna.
- Enviar datos al controlador del robot para ajustar las bases móviles y posicionar el efector final en la trayectoria trazada dentro del entorno virtual.
- Resolver el problema de cinemática inversa en el servidor del robot y ajustar los motores hasta alcanzar las posiciones angulares calculadas, provocando alinear el efector final con la trayectoria definida por el tornillo pedicular.

La Figura 27 muestra el prototipo del sistema robótico montado sobre una base que le permite girar con el fin de adoptar una mejor postura para alcanzar una trayectoria establecida.

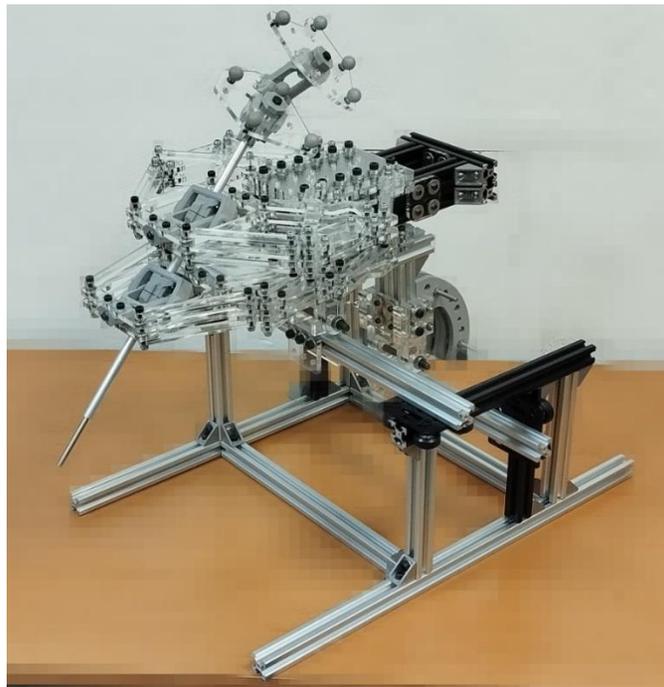


Figura 27 Imagen del prototipo robótico de configuración híbrida

5.1 Descripción del sistema robótico

El sistema robótico está conformado por un mecanismo de configuración híbrida de 4 grados de libertad (GDL).

El mecanismo consta de dos robots de configuración paralela tipo planar apilados uno sobre otro, cada uno cuenta con 2 GDL (como se muestra en la Figura 28), cada robot puede establecer la posición de su base móvil independientemente de la posición de la base móvil del otro robot.

Los brazos de cada robot planar están conformados por mecanismos de cuatro barras paralelas (paralelogramos), los cuales permiten a la base móvil mantener una orientación paralela respecto a la base fija sin importar su posición.

El diseño del mecanismo planar está pensado para tener un área de trabajo efectiva es de 5mm x 5mm.

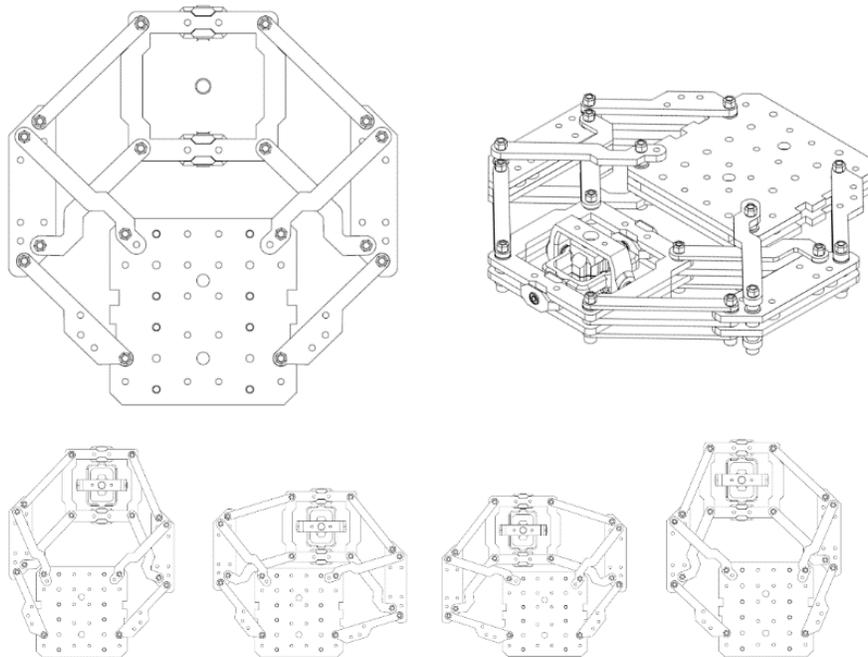


Figura 28 Esquema del robot de configuración planar

Para cada robot, la base móvil forma parte de un mecanismo de rotación análogo a un gimbal o junta universal, las piezas de forma cerrada están conectadas entre sí una seguida de otra, de modo que el eje de giro de la pieza interna es perpendicular al eje de giro de la pieza externa, todos los ejes de giro cruzan al centro de la base móvil.

Teniendo en cuenta que es posible definir una recta mediante dos puntos, se considera como *efector final del robot* a una guía cilíndrica que cruza el centro de los mecanismos de giro de ambas bases móviles, de modo que las posiciones de ambas bases móviles definen la posición y orientación del efector final como se muestra en los esquemáticos de la Figura 29.

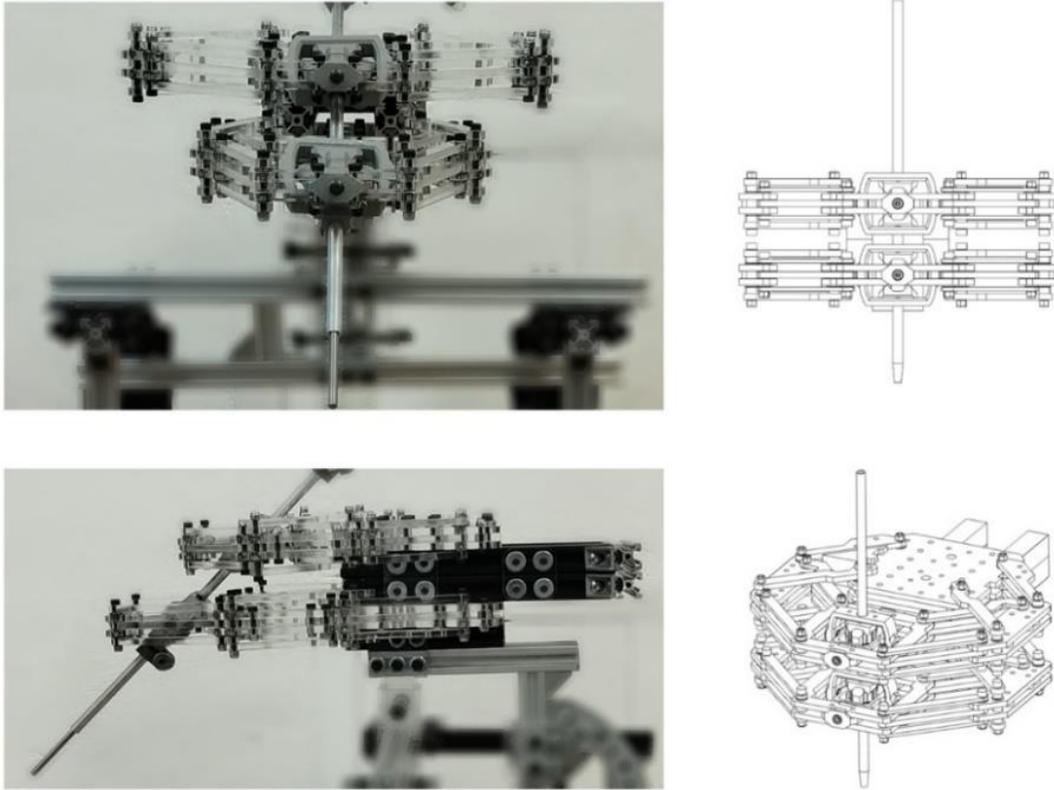


Figura 29 Imágenes y esquema del robot de configuración híbrida

Ambos robots planares apilados cuentan con dos motores y su correspondiente caja reductora de engranes de la marca *Maxon*, la flecha de cada motor está acoplada a las articulaciones actuadas del mecanismo, ubicados en la base fija del robot. Mediante un *encoder* colocado en la flecha del motor, los controladores EPOS4 de Maxon regulan la posición angular del motor, posibilitando alcanzar una posición angular específica mediante la información proporcionada por una computadora, por lo que es necesario conocer las ecuaciones de cinemática inversa de la articulación actuada mediante un análisis, la Figura 30 muestra el diagrama de conexión entre elementos.

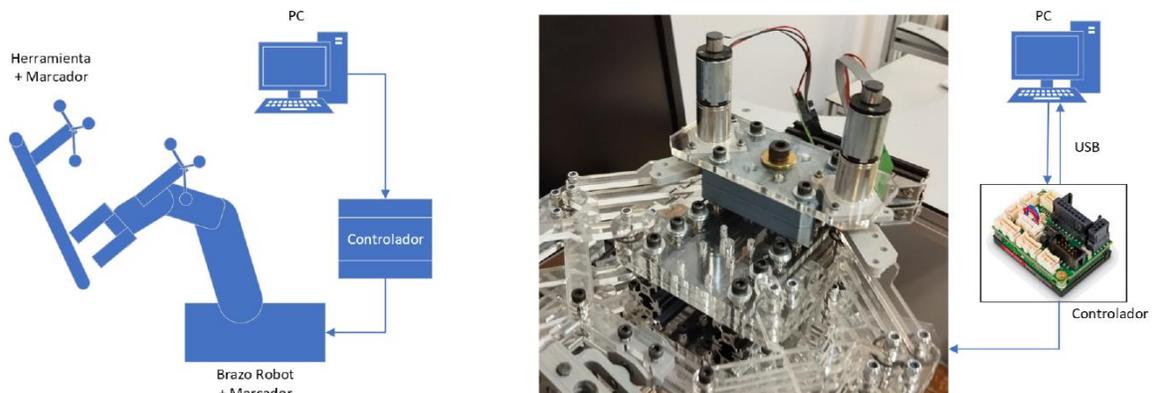


Figura 30 Diagrama de bloques del sistema de robótico

5.2 Cinemática del mecanismo planar

El análisis cinemático del robot permite conocer la posición que adoptará el efector final del robot según las posiciones angulares de las articulaciones actuadas del robot que le son especificadas como entrada.

Para llevar a cabo este análisis fue necesario el uso de matrices de transformación homogénea que consideran información de posición y orientación en una matriz 4x4.

Al definir el origen del sistema inercial con versores (i, j, k) ubicado en la base fija del mecanismo, es posible trazar la cadena cinemática de un brazo hacia el efector final, identificar eslabones y articulaciones, asignar variables de rotación o desplazamientos, mediante la aplicación de transformaciones sucesivas (acorde a la cadena cinemática) conocer la matriz de T.H. total en función de los valores articulares del mecanismo.

La Figura 31 muestra la construcción de la cadena cinemática de acuerdo a la identificación de variables de rotación y desplazamientos derivados de la morfología del mecanismo.

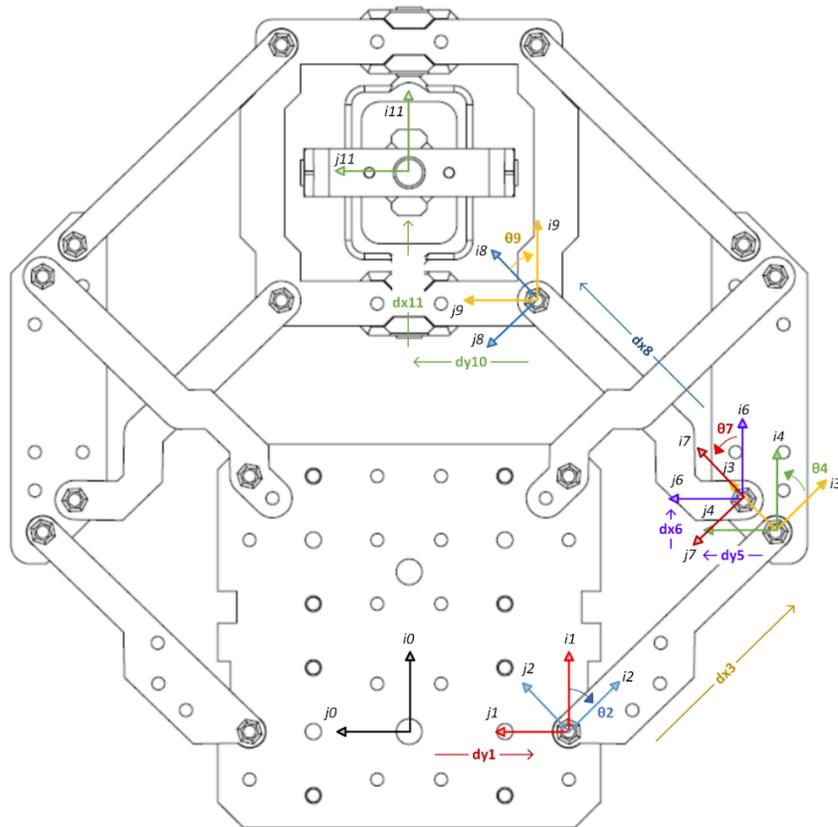


Figura 31 Diagrama de la cadena cinemática del mecanismo planar

La ecuación *Ec 6* es muestra el vector resultante de la posición que está en función de las variables articulares del robot manipulador.

$$f(\theta_2, \theta_4, \theta_7, \theta_9) = \{x, y, z\} \quad \dots (Ec 6)$$

La *Ec 7* es la aplicación sucesiva de transformaciones y la *Ec 8* representa la matriz de transformación homogénea resultante para un brazo del mecanismo. cuyos elementos correspondientes a la posición están en función de las variables articulares, además se tienen en cuenta las rotaciones opuestas generadas por los paralelogramos de cada brazo, así como las dimensiones específicas del mecanismo real para un brazo.

$$T_2(d_{y1}) \cdot T_6(\theta_2) \cdot T_1(d_{x3}) \cdot T_6(\theta_4) \cdot T_2(d_{y5}) \cdot T_1(d_{x6}) \cdot T_6(\theta_7) \cdot T_1(d_{x8}) \cdot T_6(\theta_9) \cdot T_2(d_{y10}) \cdot T_1(d_{x11}) = T_{total} \quad \dots (Ec 7)$$

$$T_{total} = \begin{pmatrix} 1 & 0 & 0 & 50 + 90 \cos \theta_2 + 90 \cos \theta_7 \\ 0 & 1 & 0 & 90 \sin \theta_2 + 90 \sin \theta_7 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \dots (Ec 8)$$

Donde:

θ_2 es la variable articular actuada que parte de la base fija.

θ_7 es la variable angular del eslabón que está conectado a la fase móvil.

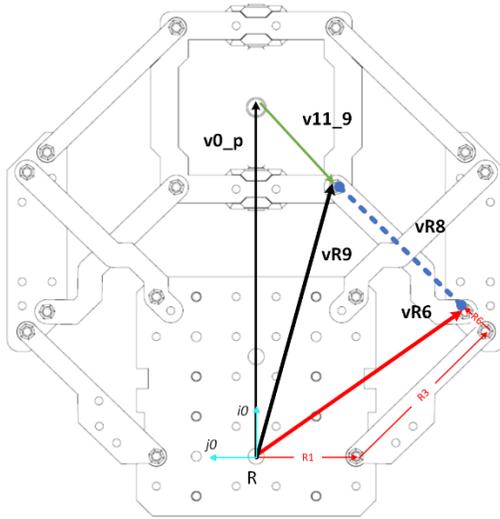
5.3 Cinemática Inversa del mecanismo planar

Resolver el problema de la cinemática inversa implica determinar la posición angular que deben adoptar las articulaciones actuadas del robot planar para que el centro de la base móvil adopte una posición deseada, de modo que el valor de posición angular está en función del vector de la posición deseada.

Durante el proceso de resolución es necesario considerar las restricciones y limitaciones de las articulaciones, la configuración de paralelogramo en los brazos garantiza que la base móvil se mantenga paralela a la base fija independientemente de su posición. Esto implica una reducción en las variables articulares, ya que las rotaciones (ϑ_2 , ϑ_4 y ϑ_7 , ϑ_9) que se producen en los extremos de los eslabones de cada paralelogramo ocurren en sentidos opuestos, manteniendo los ángulos internos congruentes (de modo que $\vartheta_4 = -\vartheta_2$ y $\vartheta_9 = -\vartheta_7$).

Para resolver el problema de la cinemática inversa de un brazo es necesario definir la ecuación de lazo cerrado, para ello se requiere, según la geometría de la Figura 31:

- Trazar de un vector $\mathbf{v0_p}$ que conecta el sistema inercial de la base fija con la posición deseada en el centro de la base móvil.
- Obtener la transformación T_{11_9} a partir del cálculo de la matriz inversa de la sucesión de transformaciones homogéneas T_{9_11} , que parten de un extremo final del paralelogramo del brazo y terminan en el centro de la base móvil.
- Calcular el vector $\mathbf{vr9}$ sumando del vector $\mathbf{v0_p}$ con vector de posición $\mathbf{v11_9}$ derivado de la transformación T_{11_9} .
- Conocer el vector de posición $\mathbf{vr6}$ derivado de la transformación T_{0_6} .
- Los vectores $\mathbf{vr6}$ y $\mathbf{vr9}$ inician en la base fija y terminan en los extremos de un eslabón del paralelogramo, se define la ecuación de lazo cerrado mediante la resta de ambos vectores, igualados al vector $\mathbf{vr8}$ que recorre al eslabón, excluyendo los ángulos ϑ_9 y ϑ_7 de la igualdad.
- Se calcula la magnitud de ambos lados de la ecuación, considerando que la magnitud del vector $\mathbf{vr8}$ corresponde a la distancia dx_8 entre los centros de las articulaciones del eslabón como se muestra en la Figura 32.



$$T9_11 = T6[\theta9].T2[dy10].T1[dx11]$$

$$T11_9 = T9_11^{-1}$$

$$T11_9 \xrightarrow{T.H. \gg \text{Vect}} v11_9_{(3x1)}$$

$$T0_6 = T2[dy1].T6[\theta2].T1[dx3].T6[\theta4].T2[dy5].T1[dx6]$$

$$T0_6 \xrightarrow{T.H. \gg \text{Vect}} vR6_{(3x1)}$$

$$vR9 = v0_p + v11_9$$

$$vR8 = vR9 - vR6$$

$$||vR8|| = ||vR9 - vR6|| = ||N||$$

$$dx8 = ||vR9 - vR6|| = ||N||$$

$$dx8^2 = N^T \cdot N$$

Figura 32 Esquema de vectores para definir la ecuación de lazo cerrado

Calculando el producto punto entre el vector \mathbf{N} con dimensiones 3×1 y su transpuesto \mathbf{N}^T de 1×3 , se obtiene como resultado una ecuación con ambos términos de dimensión 1×1 , de la cual se despeja y agrupan términos con la misma variable como se muestra en las ecuaciones Ec 9, Ec 10, Ec 11;

$$A \cos[\theta2] + B \sin[\theta2] + C = (dx8)^2 \quad \dots (Ec 9)$$

$$A \cos[\theta2] + B \sin[\theta2] + (C - (dx8)^2) = 0 \quad \dots (Ec 10)$$

$$A \cos[\theta2] + B \sin[\theta2] + D = 0 \quad \dots (Ec 11)$$

Donde:

$$A = 180 \cdot (50 - xp)$$

$$B = -180 \cdot yp$$

$$D = (50 - xp)^2 + yp^2$$

Para obtener la solución de forma cerrada de la cinemática inversa de forma algebraica, se emplea la ecuación trascendental para obtener la ecuación de la variable articular actuada $\vartheta2$, Ec 12 que depende del vector de posición deseado en función de los valores (xp, yp, zp) [27].

$$\theta_2 = \tan^{-1}\left(\frac{B}{A}\right) \pm \cos^{-1}\left(\frac{D}{\sqrt{A^2 + B^2}}\right) \quad \dots (Ec 12)$$

Una vez que se cuenta con la ecuación *Ec 12* de la articulación actuada ϑ_2 es posible determinar el movimiento de la base móvil desde el punto **Pi** hasta **Pf** siguiendo una trayectoria rectilínea. Para ello se emplea un perfil de velocidad v que suaviza el movimiento en los extremos de la trayectoria rectilínea **Pt**, siguiendo una función temporal jerk que, por seguridad, garantiza unas aceleraciones iniciales y finales cero, con el polinomio que muestra la Figura 33 y la ecuación *Ec 13* [28], [29], obteniendo como resultado la simulación del movimiento del mecanismo partiendo de un punto inicial a un punto final que se observa en el capítulo 6 donde se muestran los resultados obtenidos en este trabajo.

$$Pt = pi + \left(10\left(\frac{i}{t}\right)^3 - 15\left(\frac{i}{t}\right)^4 + 6\left(\frac{i}{t}\right)^5\right) \cdot (pf - pi) \quad \dots (Ec 13)$$

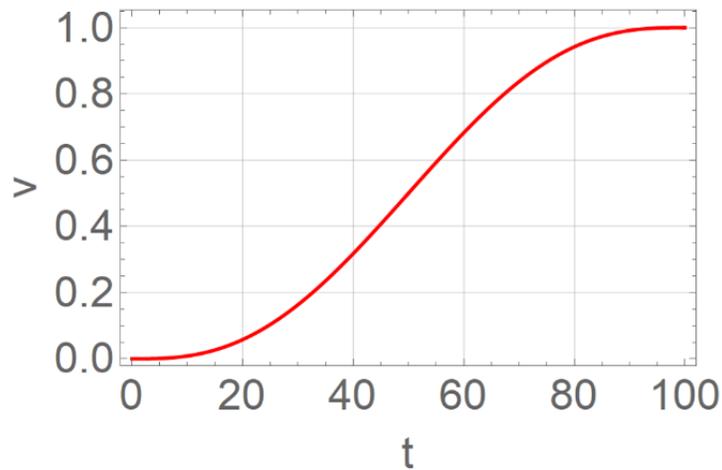


Figura 33 Gráfica del polinomio seleccionado

5.4 Interfaz de comunicación

Al considerar que las dimensiones del mecanismo son constantes debido a que están definidas por las longitudes y morfología de los eslabones, las ecuaciones cinemáticas son invariables, haciendo necesario que el usuario final no deba tener acceso a los parámetros que definen las ecuaciones de movimiento del robot, por lo que se creó un servidor encargado de:

- Recibir un vector de posición objetivo en formato *JSON* (a modo de petición del cliente)
- Resolver las ecuaciones de cinemática inversa considerando un perfil de velocidad, calculando una serie de posiciones angulares en grados que permiten seguir una trayectoria rectilínea.
- Transformar las posiciones angulares en el número de pasos que el *encoder* montado en el motor debe adoptar.
- Comunicar la información a los *drivers* Epos4 de cada motor, repositonando la base móvil del robot.
- Generar respuesta del servidor hacia el cliente con las posiciones angulares finales obtenidas de cada controlador del robot una vez finalizado el recorrido de la trayectoria.

Para acceder al robot, se utiliza un cliente que transmite la posición deseada en formato *JSON* a través de *websockets*, la interfaz de usuario se realiza mediante Unity.

El servidor espera valores alcanzables que se encuentren dentro del área de trabajo de la base móvil de cada robot planar y únicamente recibe la petición del cliente cuando la trayectoria del tornillo que se pretende implantar cruza el área de trabajo de ambos robots en el entorno virtual.

Este requerimiento hace necesario desarrollar un algoritmo para estimar el vector de posición que parte del origen de la base fija del robot y finaliza en la intersección entre la trayectoria del tornillo y el plano de trabajo donde puede desplazarse la base móvil del robot planar. La Figura 34 muestra la interacción entre el cliente de Unity y el servidor del robot.

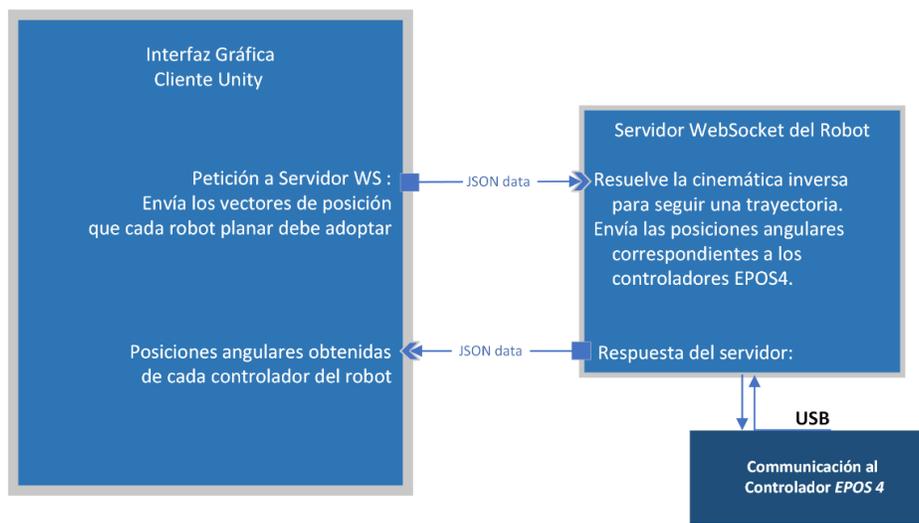


Figura 34 Diagrama de comunicación cliente – servidor para posicionar al robot

5.5 Algoritmo de estimación para el posicionamiento del robot sobre trayectoria proyectada por el tornillo

Dentro del entorno virtual, el algoritmo de posicionamiento utiliza la información de localización del robot y del tornillo; a modo de apoyo visual se genera una estimación que indica en donde debe colocarse el robot real respecto de la columna del paciente, tomando en cuenta el plano de operación de un robot planar.

Para cada robot planar se calcula lo siguiente:

- Conocer la base R que define la orientación del robot respecto del sistema global S y el vector de posición del robot $Or_{[S]}$ definido por el origen de la base R .
- Conocer los vectores $vp1_{[S]}$ y $vp2_{[S]}$ que define la localización del tornillo pedicular respecto de la base S , los dos puntos están ubicados en los extremos (punta y cabeza respectivamente) del tornillo, ambos cruzan el eje longitudinal.
- Calcular $iOr_{[R]}$ (vector opuesto a $Or_{[S]}$) referido al sistema de coordenadas R , mediante el producto punto entre el vector $iOr_{[S]}$ y la matriz de rotación W (descrita por los versores de la base R respecto de la base S).
- Referir los elementos que definen al tornillo respecto de la base R , para ello es necesario sumar $iOr_{[R]}$ a cada vector $vp1_{[S]}$ y $vp2_{[S]}$ referidos a la base R ; para referir dichos vectores a la base R es necesario calcular el producto punto entre vector $vp1_{[S]}$ y de la matriz de rotación W inversa y hacer lo mismo para el vector $vp2_{[S]}$, lo anterior se muestra en las gráficas de la Figura 35.

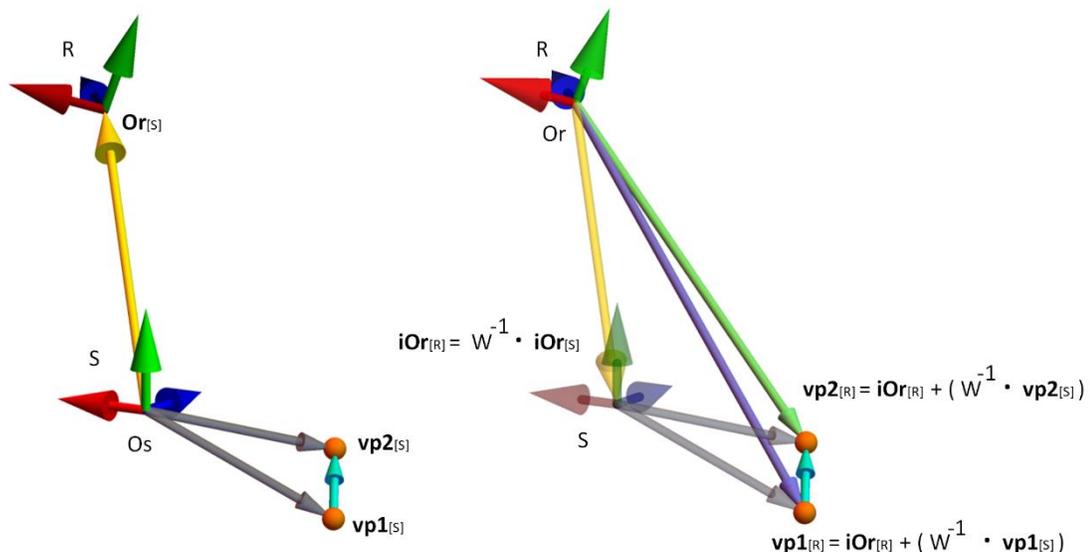


Figura 35 Gráficas 3D de los vectores que definen al tornillo referidos a la base del robot

Una vez que los vectores de posición de los extremos del tornillo son referidos a la base del sistema de coordenadas R, dado que el mecanismo es planar, es posible conocer el cruce de la trayectoria del tornillo con el plano de acción del robot, para ello es necesario hacer lo siguiente:

- Calcular el vector $\mathbf{vTray}_{[R]}$ que define la trayectoria del tornillo mediante la resta de vectores correspondientes a los extremos del tornillo.
- Conocer el factor de escala β permite proyectar el vector $\mathbf{vTray}_{[R]}$ desde su origen en $\mathbf{vp1}_{[R]}$ hasta el cruce con el plano ZX del sistema R, que es el plano de acción del robot planar; por lo que, β se calcula mediante la división entre el valor de la componente Y de dos vectores; el opuesto aditivo de la componente Y del vector $\mathbf{vp1}_{[R]}$ que indica la punta del tornillo dónde inicia la trayectoria se contempla como numerador, mientras que se considera a la componente Y del vector de trayectoria $\mathbf{vTray}_{[R]}$ como denominador.
- Al multiplicar el vector de trayectoria $\mathbf{vTray}_{[R]}$ por el escalar β , es posible conocer $\mathbf{vIntP}_{[R]}$.
- Al sumar $\mathbf{vp1}_{[R]}$ y $\mathbf{vIntP}_{[R]}$ se obtiene el vector $\mathbf{vPosEst}_{[R]}$ respecto del sistema coordenado del robot, representa el punto de intersección de la trayectoria del tornillo con el plano de acción del robot, ya que, el vector resultante debe tener componente Y nula con coordenadas $(x, 0, z)_{[R]}$.
- Una vez que se conoce el vector de estimación $\mathbf{vPosEst}_{[R]}$ respecto del sistema R, es necesario referirlo respecto del sistema de coordenadas inicial S, para ello se hace un cambio de base, mediante el cálculo del producto punto entre el vector $\mathbf{vPosEst}_{[R]}$ y la matriz W.
- Sumando el vector $\mathbf{vPosEst}_{[S]}$ con el vector de posición $\mathbf{Or}_{[S]}$ que describe el origen de la base del robot respecto del sistema S, se obtiene como resultado el punto dónde la base móvil debe estar colocada respecto del sistema de coordenadas S como de muestra en las Figura 36.

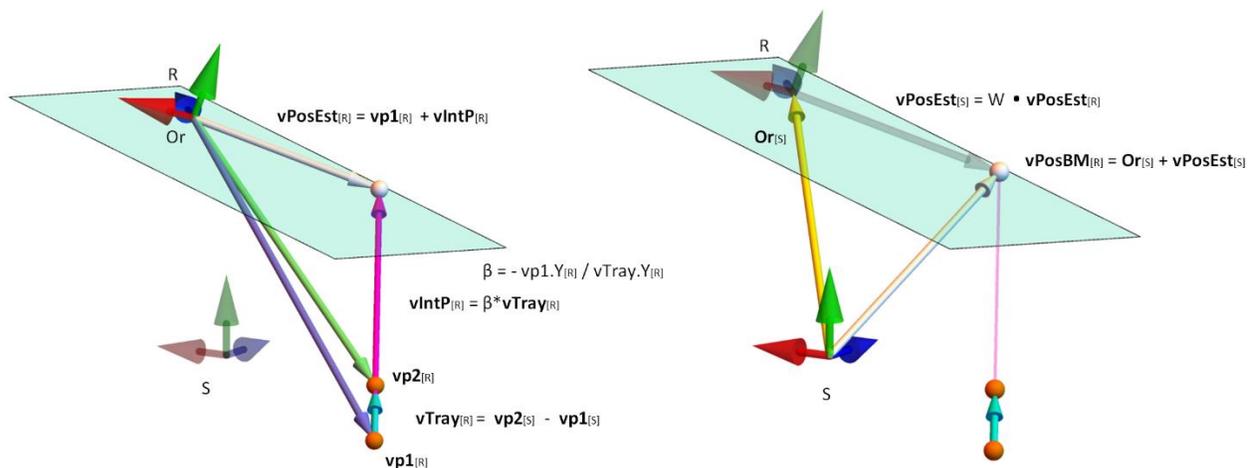


Figura 36 Gráficas 3D de los vectores que definen la estimación de la posición de la base móvil

5.6 Integración con el entorno virtual

Para realizar la integración en Unity, es necesario importar los archivos con extensión *.obj* de cada modelo 3D, los componentes importados son: robot, tornillos y columna vertebral lumbar; la cual fue generada con estudios de imágenes médicas de un paciente real, por el equipo de trabajo del laboratorio de BioInstrumentación del Instituto de Ciencias Aplicadas y Tecnología.

La Figura 37 muestra el entorno virtual que incluye el robot de configuración híbrida planar (1), la columna vertebral lumbar con los tornillos correspondientes que fueron colocados en el software de planeación preoperatoria (2) y un botón (3) que, al ser activado calcula dos vectores de posición que son el punto de cruce entre la trayectoria del tornillo que se pretende implantar y el plano de operación de cada robot planar independiente.

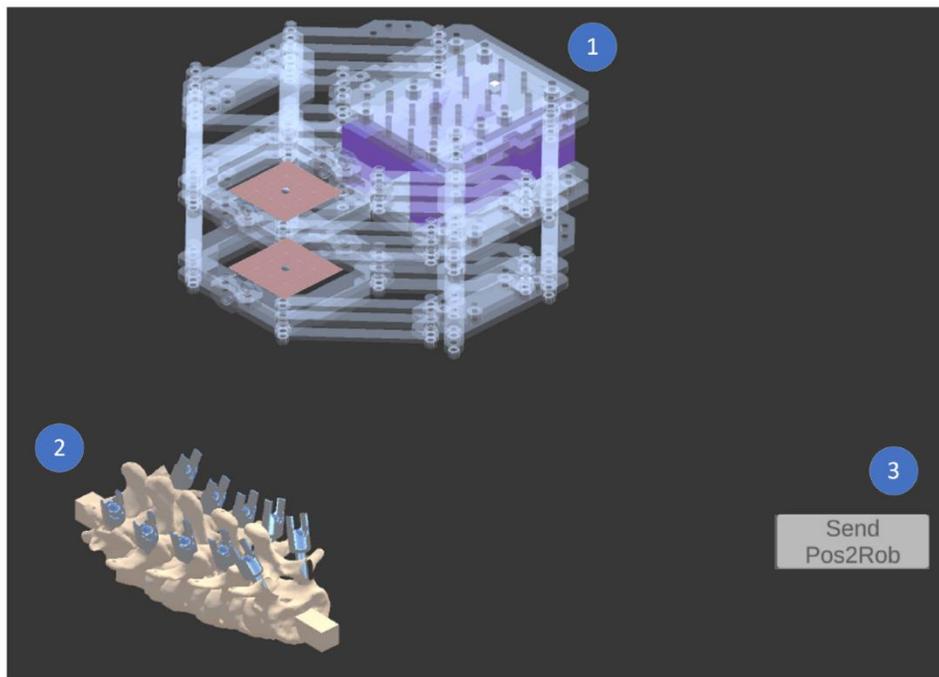


Figura 37 Componentes del entorno virtual

La Figura 38 muestra la proyección de la trayectoria estimada por el robot para la implantación del tornillo seleccionado, que está asociado a la columna vertebral en el entorno virtual. El trazo inicia desde la punta del tornillo y finaliza en la posición de cruce entre la trayectoria con el plano de acción de un robot planar, el cual es estimado mediante el algoritmo de posicionamiento de la base móvil sobre la trayectoria para cada nivel del robot híbrido.

Para las trayectorias azul y roja, los extremos de inicio y termino de se encuentran en la punta del tornillo seleccionado y en la esfera gris correspondiente, las esferas indican la estimación de la posición deseada para el centro de la base móvil de cada robot planar superior e inferior.

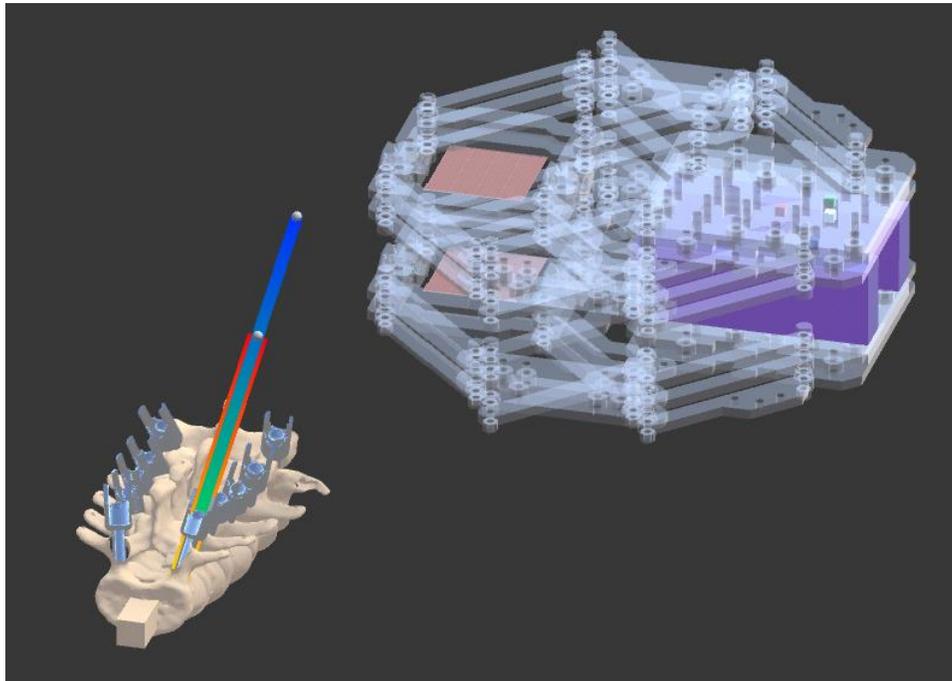


Figura 38 Generación de la trayectoria estimada para implantar el tornillo seleccionado

Las imágenes muestran la vista superior (Figura 39), frontal (Figura 40) y lateral (Figura 41) de la columna vertebral y el conjunto referencias visuales de la trayectoria requerida para implantar el tornillo seleccionado

- El bloque A representan el escenario total que muestra el escenario virtual con el robot y la columna lumbar con la posición deseada de los tornillos pediculares respecto de la columna
- El bloque B muestra la trayectoria estimada para la base móvil del robot planar inferior.
- El bloque C muestra la trayectoria estimada para la base móvil del robot planar superior.
- El bloque D muestra las trayectorias estimadas de ambos robots haciendo evidente que son colineales.

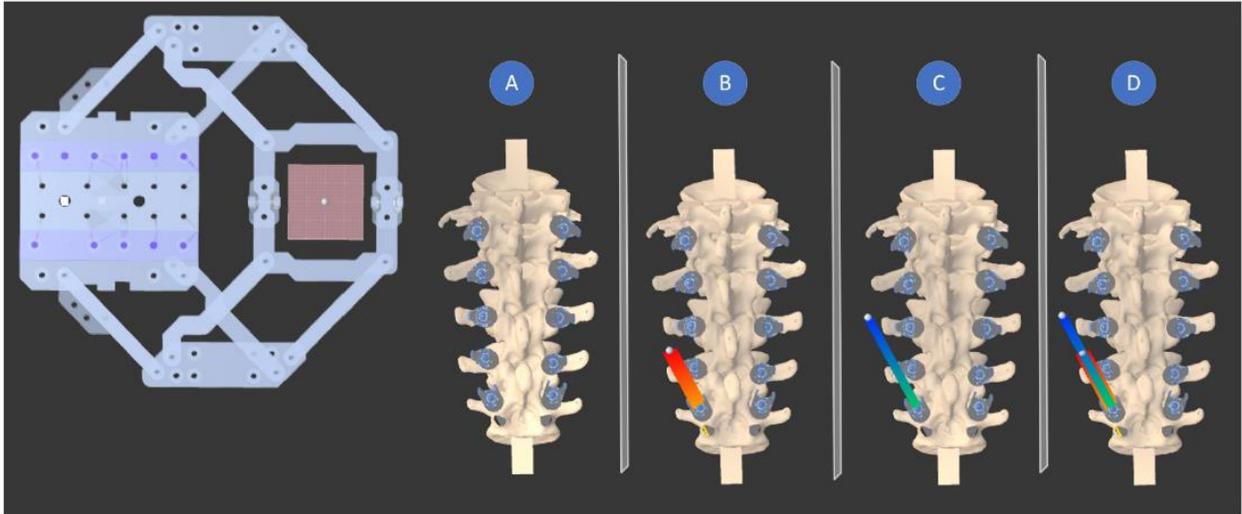


Figura 39 Vista anteroposterior de trayectorias para implantar en la columna vertebral del entorno virtual

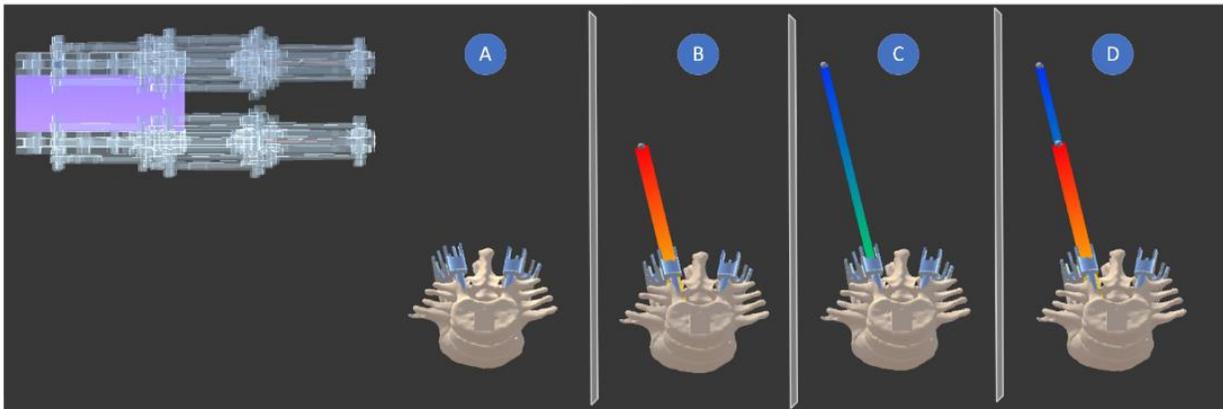


Figura 40 Vista axial de trayectorias para implantar tornillo en la columna vertebral dentro del entorno virtual

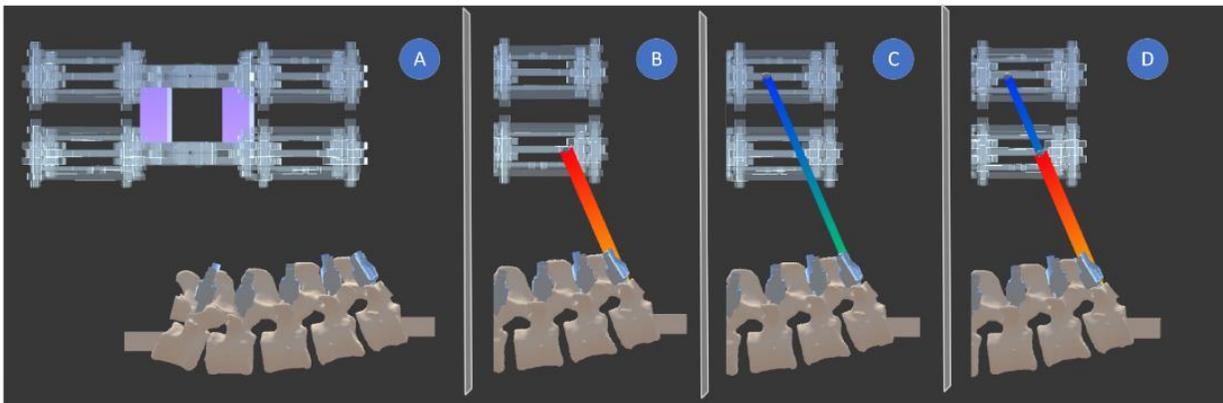


Figura 41 Vista lateral de trayectorias para implantar tornillo en la columna vertebral dentro del entorno virtual

Una vez que el robot se acerca a la columna dentro del entorno virtual, un indicador verde con forma de base móvil se posiciona en el centro de la estimación proyectada, es decir, cuando la trayectoria se encuentra dentro del área de trabajo del robot.

El botón dentro de la interfaz del usuario permite enviar una solicitud al servidor del robot únicamente cuando la trayectoria del tornillo seleccionado cruza el área de trabajo de ambos robots en el entorno virtual, es decir cuando los dos indicadores están en verde.

La Figura 42 ilustra diferentes casos:

1. Solo la base móvil del robot planar superior está en posición de alcanzar la posición estimada.
2. Únicamente la base móvil del robot inferior puede alcanzar la posición objetivo.
3. Ambas estimaciones están dentro del área de trabajo del robot, permitiendo que el botón envíe una solicitud con información válida en formato *JSON* de las posiciones deseadas al servidor, ya que, ambas bases móviles son capaces de adoptar la posición objetivo.

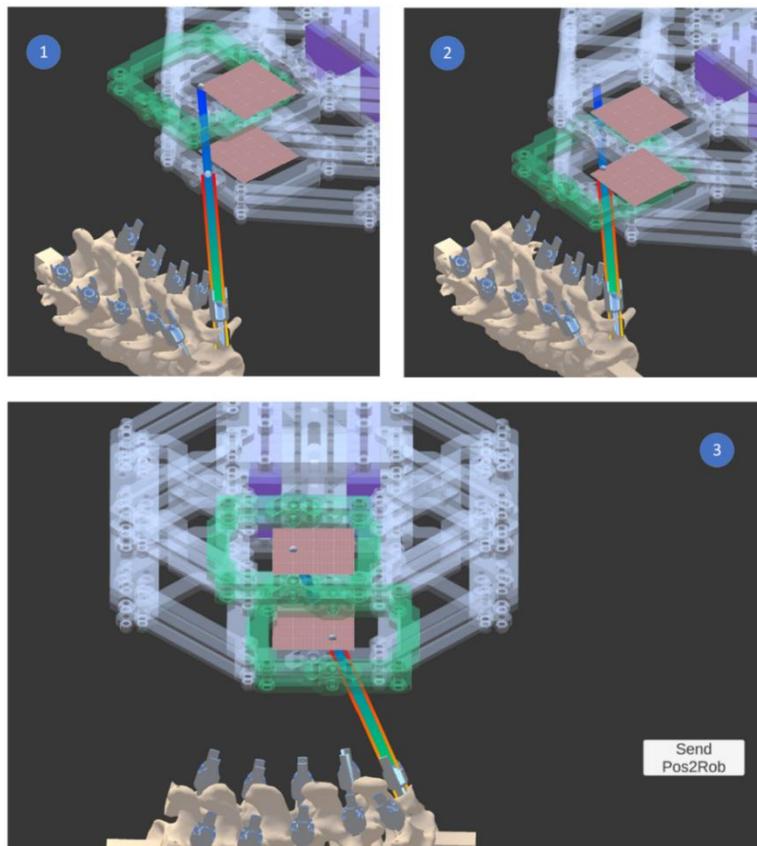


Figura 42 Casos de posicionamiento dentro del área de trabajo de cada base móvil

Una vez que se conocen los elementos considerados como estructuras rígidas dentro de la escena y se les asocia un marco de referencia con marcadores esféricos altamente reflejantes, es posible incluir el sistema de seguimiento, usando el algoritmo de alineación para complementar la funcionalidad del *tracker*, haciendo corresponder la postura del elemento virtual con su correspondiente real, eso incluye los siguientes elementos:

1. Modelo de columna vertebral lumbar fabricado utilizando manufactura aditiva material de PLA.
2. Herramienta guía tubular de aluminio con longitud de 30cm (Efector final del robot).
3. Herramienta de punción cilíndrica de acero de 5mm de diámetro y 40cm de longitud.

Las Figura 43 y Figura 44 muestran los elementos reales y virtuales con un marco de referencia asociado.



Figura 43 Marcos de referencia asociados a columna, efector final del robot y herramienta de punción reales

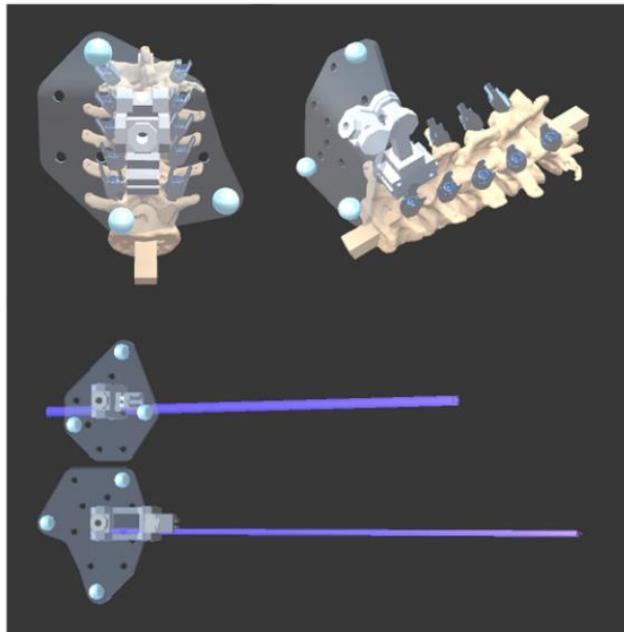


Figura 44 Marcos de referencia asociados a columna, efector final del robot y herramienta de punción virtuales

6.- Pruebas y resultados

En este capítulo se muestran los resultados de cada componente del sistema y los resultados generales de navegación guiada por el robot.

6.1 Simulación del movimiento rectilíneo del mecanismo

En el software Wolfram Mathematica de álgebra computacional, se simuló el movimiento rectilíneo de la base móvil considerando un perfil de velocidad que suaviza el movimiento de la base móvil en los extremos del recorrido que parte de la esquina inferior derecha del área de trabajo a esquina superior izquierda del área de trabajo efectiva establecida.

A modo de prueba, el recorrido desde un punto inicial pi hacia un punto final pf se dividió en 100 puntos, en cada uno de los puntos se debe calcular el valor ϑ de la posición angular que cada articulación actuada debe adoptar para posicionar el centro de la base móvil sobre el punto con coordenadas $\{x,y\}$.

La Figura 45 ilustra el resultado de la prueba, mostrando las gráficas correspondientes a la variable angular de cada brazo, así como un gráfico generado a partir de los fotogramas de la simulación, con intervalo de 10 posiciones entre cada fotograma. En este gráfico, se observa cada eslabón que compone al mecanismo sin considerar el paralelogramo que lo compone físicamente.

$$pi = (150, -25) \quad pf = (200, 25)$$

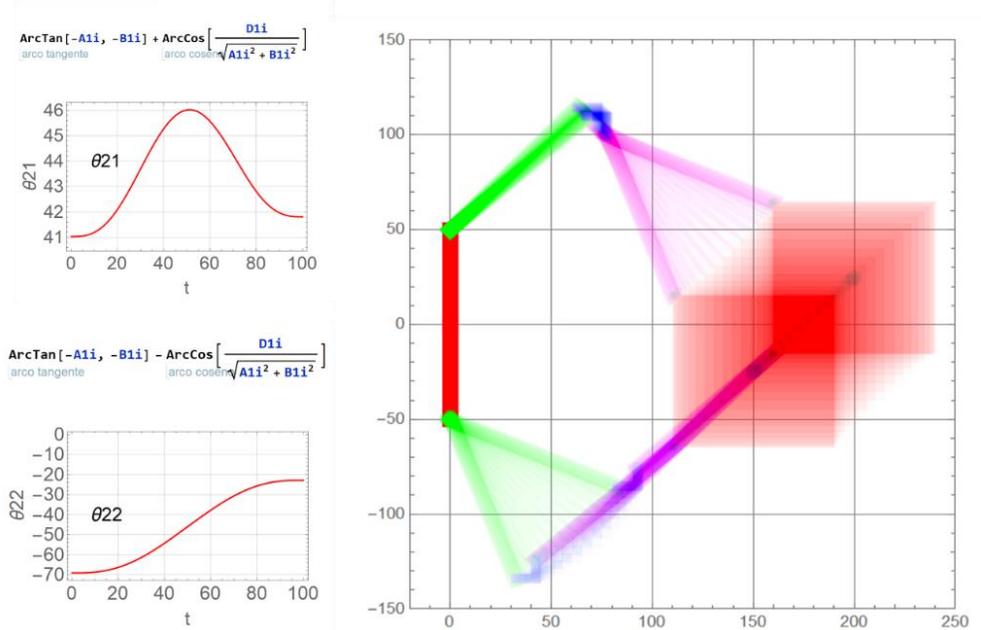


Figura 45 Gráficas de los ángulos y simulación del movimiento rectilíneo del robot

6.2 Descripción del prototipo robótico

El robot híbrido compuesto por el arreglo de dos robots planares apilados está montado sobre una base deslizante que le permite girar hasta 90 grados, ampliando el rango de movimiento y habilitando al efector final para que adopte una mejor postura respecto del paciente como se muestra en los incisos A y B de la Figura 46, se diseñó contemplando las necesidades de algunos abordajes de procedimientos quirúrgicos, ya que el abordaje de la columna vertebral puede llevarse a cabo colocando al paciente en posición prona (boca abajo), o en posición lateral, dependiendo del procedimiento [30].

Cabe destacar que el comportamiento del prototipo presenta resistencia ante perturbaciones físicas externas, ya que los controladores EPOS4 de cada motor Maxon logran mantener una posición angular específica, los incisos C y D de la Figura 46 presentan una vista superior e isométrica correspondiente del mecanismo.

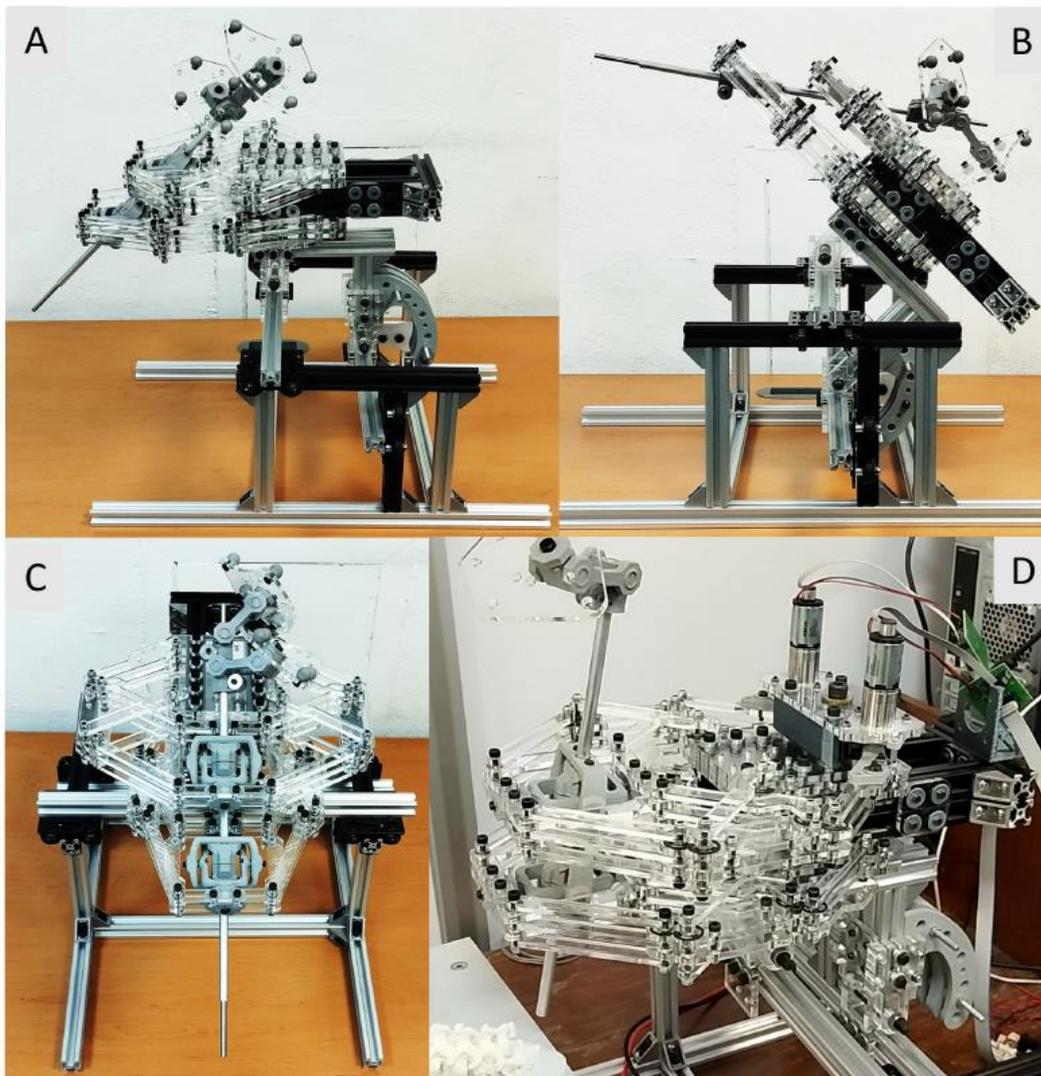


Figura 46 Vistas del prototipo robótico ensamblado

6.3 Cálculo de la precisión del sistema de seguimiento

Con el objetivo de conocer la precisión de la aplicación del algoritmo de alineación, se desarrolló un banco de pruebas con interfaz en Unity para calcular métricas estadísticas de su desempeño.

Se empleó una tablilla de dimensiones conocidas con hendiduras como casillas del banco de pruebas y una barra de acero de 40cm de largo como herramienta de punción, a ambos elementos se les acopló un marco de referencia con marcadores esféricos para ser registrados por el sistema de cámaras infrarrojas de visión estéreo (*Tracker*) vinculado a Unity, como se muestra en la Figura 47.

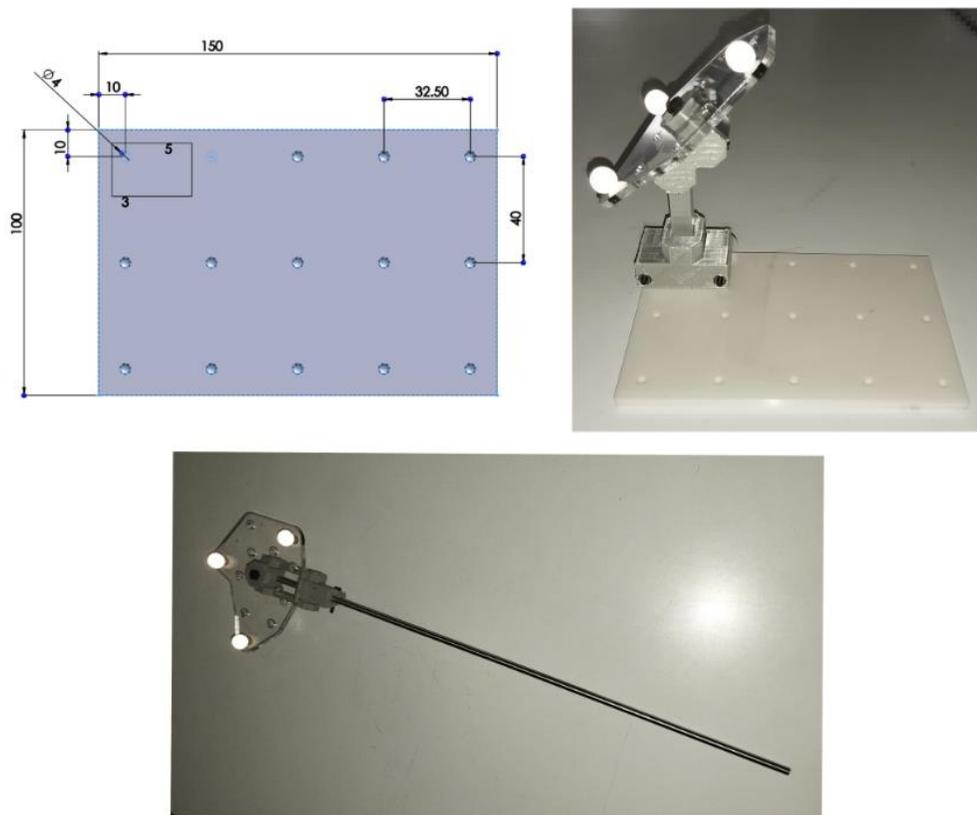


Figura 47 Tablilla con matriz de puntos y herramienta de punción.

Los modelos 3D de todos los objetos necesarios para replicar la escena real se añadieron a una escena de Unity.

El proceso de registro de datos requirió colocar la punta de la herramienta de punción en cada una de las hendiduras del banco de pruebas como se presenta en la Figura 48.

En el instante en el que la tecla G era presionada, un programa capturó en un archivo con formato *JSON* la posición de la punta de la herramienta y la posición de una hendidura del banco de pruebas más cercana a la punta de la herramienta de punción.

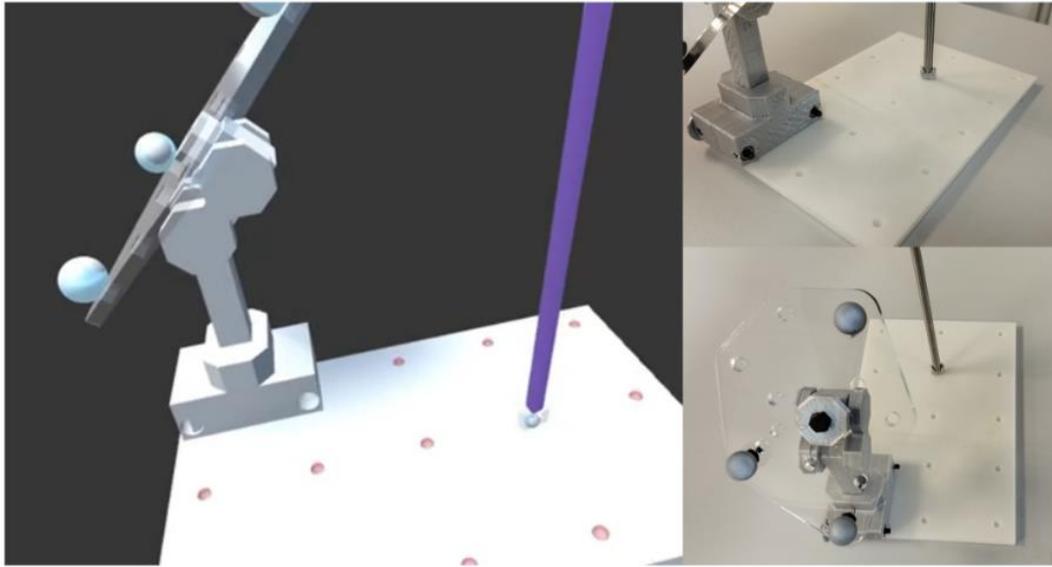


Figura 48 Tablilla de calibración dentro del entorno virtual y real para el proceso de calibración del error de posicionamiento del sistema.

Se llevaron a cabo dos repeticiones del experimento previamente mencionado, generando 33 y 39 muestras respectivamente, con aproximadamente 3 muestras por cada hendidura del banco de pruebas cuyas unidades son milímetros.

La tabla presenta los valores estadísticos asociados a: error absoluto medio (*MAE*), raíz del error cuadrático medio (*RMSE*), coeficiente de correlación (*r*) y las desviaciones estándar (σ) del error absoluto de cada componente (*x*, *y*, *z*).

Tabla 1 Valores estadísticos de los experimentos realizados

#Prueba	MAE [mm]	RMSE [mm]	r	σ_x [mm]	σ_y [mm]	σ_z [mm]
1	0.81354	1.01405	0.999999	0.394766	0.547696	0.439426
2	0.65921	0.940844	0.999999	0.342227	0.218399	0.947588

En el contexto del error absoluto medio y la raíz del error cuadrático medio se evidencia que el sistema de seguimiento tiene una precisión aceptable, con valores menores a 1 milímetro (mm) de error absoluto y alrededor de 1 mm para RMSE.

Estos indicadores sugieren una adecuada precisión por la baja dispersión de las posiciones de la punta de la herramienta de punción respecto a la posición de las hendiduras del banco de pruebas; las métricas señalan que hay un margen considerable para mejorar la precisión del sistema.

6.4 Despliegue del sistema robótico guiado por visión computacional

Con el fin de proveer una herramienta que haga frente a los desafíos en términos de visión y ejecución que se presentan dentro de una cirugía con enfoque de mínima invasión, es necesario desplegar los sistemas desarrollados y validar la fidelidad del sistema virtual con el real.

Para llevar a cabo el despliegue del sistema, inicialmente se conectó a la computadora el *tracker* y las placas de control de los motores acoplados al robot, posteriormente se inicializaron los servidores correspondientes al sistema robótico y al sistema de seguimiento, finalmente se conectó la interfaz de Unity como cliente como se muestra en la Figura 49 .

En Unity se presentan las herramientas y estructuras anatómicas no visibles para el cirujano, además, posibilita transmitir al servidor del robot las coordenadas que cada base móvil debe adoptar para posicionar de manera precisa la herramienta de punción sobre el paciente, complementando las destrezas manuales del cirujano.

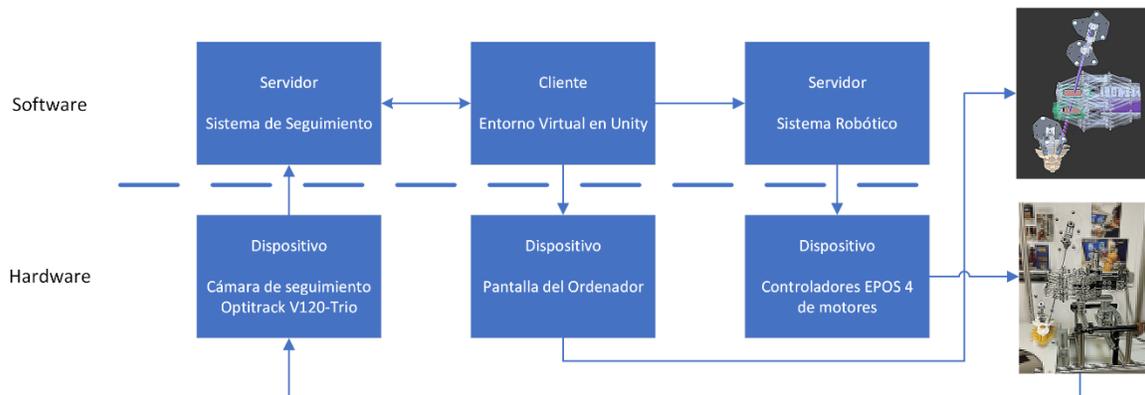


Figura 49 Diagrama de conexión para el despliegue del sistema

La prueba para validar la fidelidad del sistema virtual y real dentro de un entorno controlado de laboratorio requirió ubicar el robot cerca del modelo de la columna vertebral lumbar real, seguido de eso, la detección de objetos mediante el *tracker*, en conjunto con el algoritmo de alineación permitió recrear continuamente la postura de cada objeto dentro del entorno virtual.

Al seleccionar algún tornillo incrustado en la columna virtual, el software traza la trayectoria requerida para implantar dicho tornillo y calcula los puntos de cruce de la trayectoria con el plano de acción de cada robot planar.

Es necesario reubicar el robot físico porque, cuando se localizan los puntos de cruce dentro del área de trabajo del robot, la trayectoria es alcanzable; solo entonces se envían al servidor las coordenadas de los puntos de cruce estimados. El servidor procesa la petición y resuelve la cinemática inversa de cada variable angular actuada para cada punto del recorrido, reposicionando los motores y permitiendo al efector final adoptar una postura colineal a la trayectoria requerida por el tornillo seleccionado.

Del resultado de esta prueba se obtuvieron diferentes vistas donde se compara el entorno real con el virtual, observando que presentan gran similitud, ya que, visualmente ambos conservan la misma relación de posición entre componentes como se muestra en la serie de imágenes que se presentan a continuación Figura 50 Figura 51 Figura 52 Figura 53 Figura 54 .

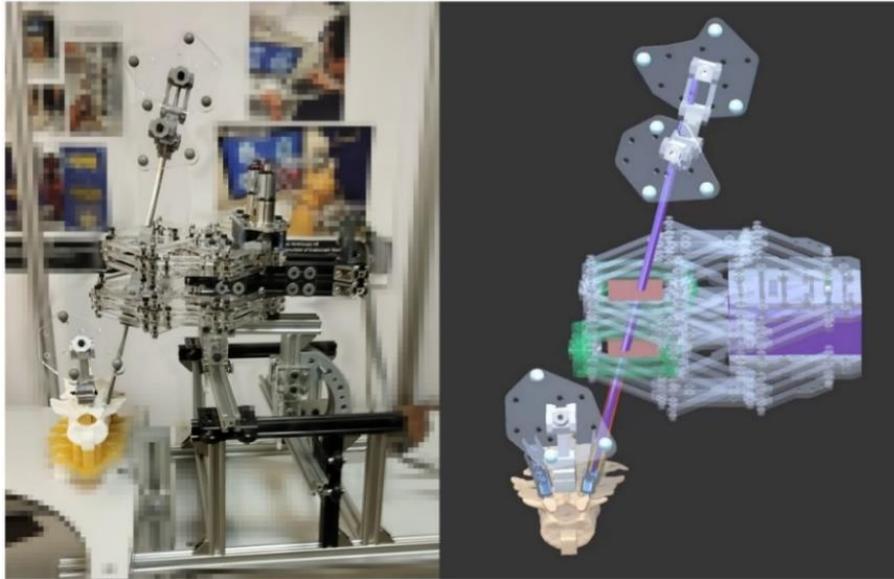


Figura 50 Vista lateral del robot y axial a la columna de la escena real y virtual

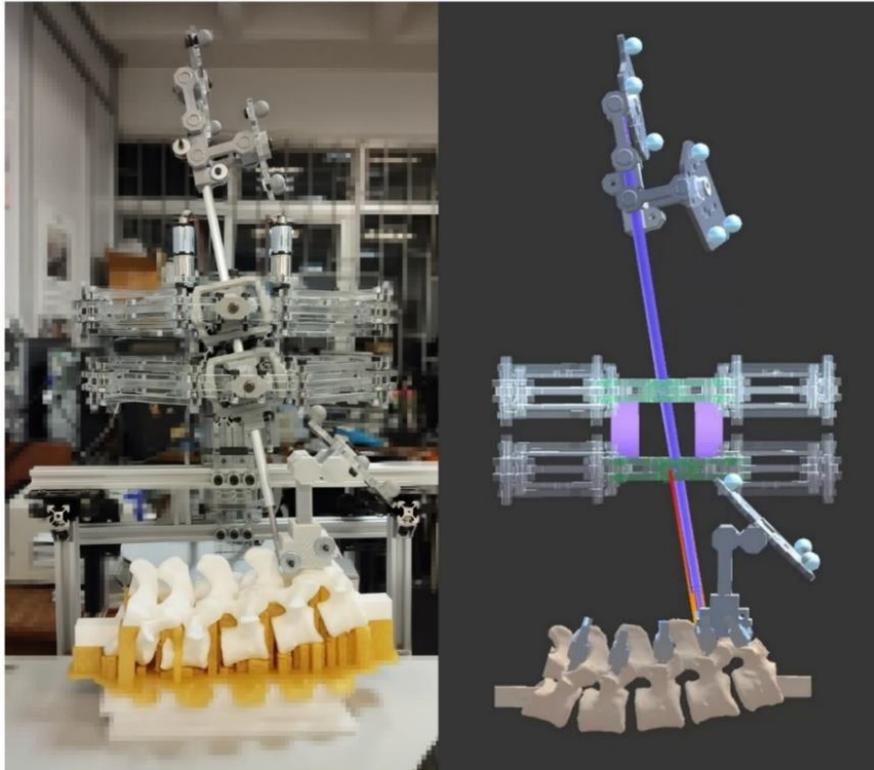


Figura 51 Vista frontal del robot y lateral a la columna de la escena real y virtual

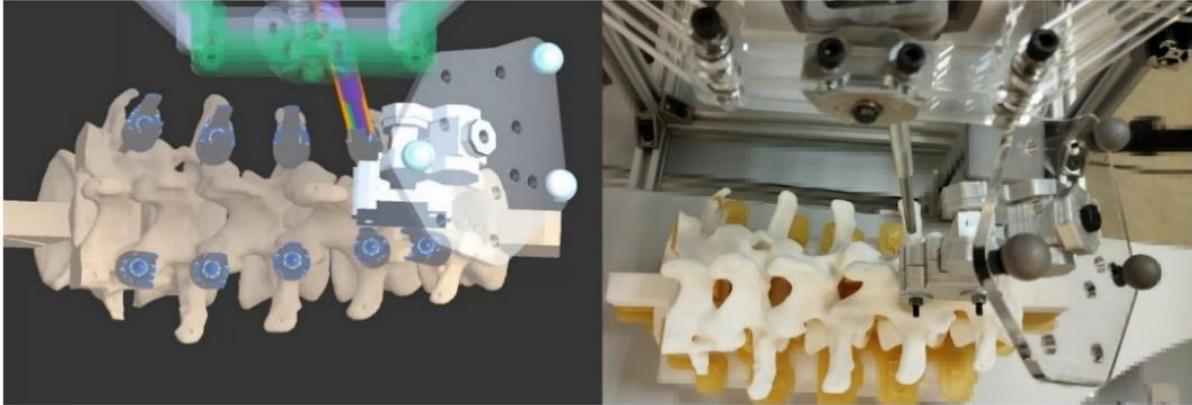


Figura 52 Vista superior del robot y anteroposterior a la columna de la escena real y virtual

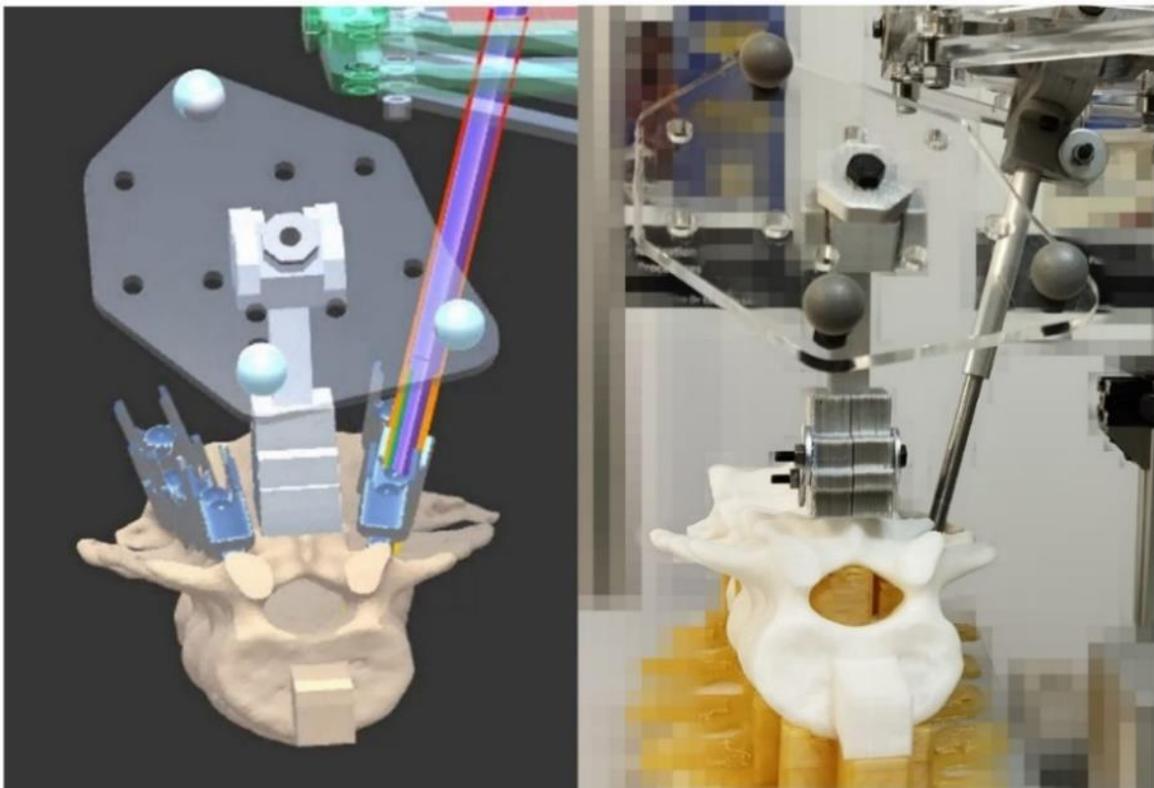


Figura 53 Vista axial a la columna de la escena real y virtual

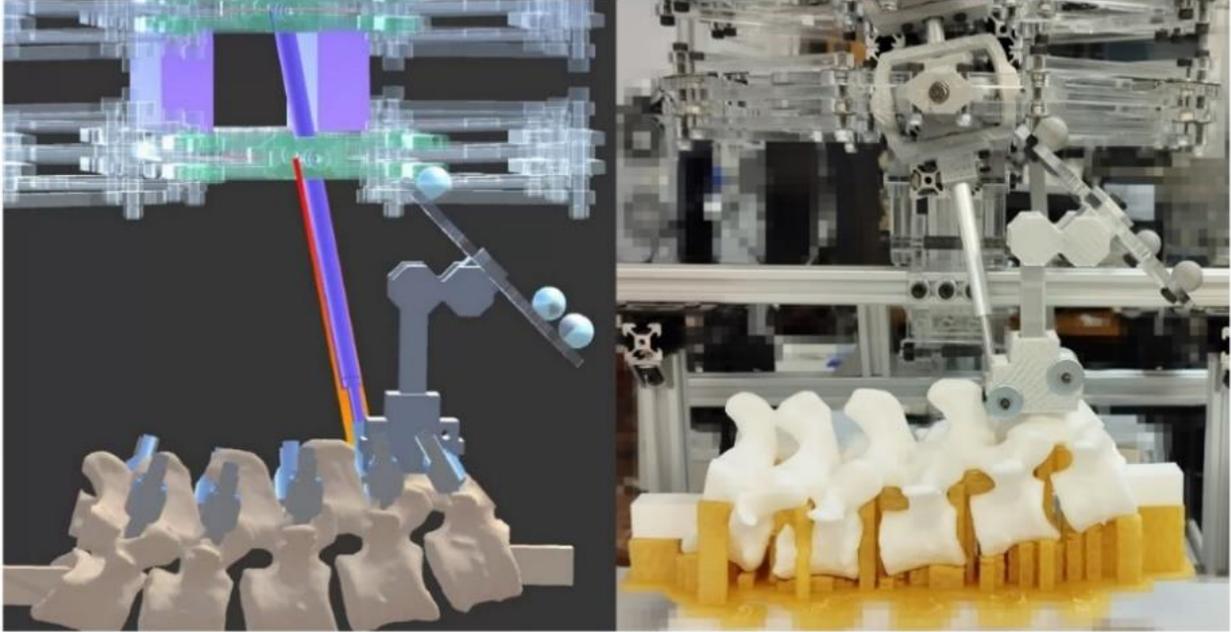


Figura 54 Vista lateral de la columna de la escena real y virtual

Teniendo en cuenta que las perspectivas de las capturas real y virtual no son exactamente las mismas, se hizo un acercamiento a las imágenes anteriores para compararlas visualmente, en cada una de las imágenes virtuales de la Figura 55 es posible observar que la herramienta de punción está alineada con la guía del efector final y ambos elementos se encuentran dentro del trazo que genera la trayectoria de inserción del tornillo seleccionado conservando la relación de las posiciones entre componentes del entorno real.

El entorno virtual presenta una reproducción de alta fidelidad, considerando que el diámetro externo de la herramienta guía es de 8mm y el diámetro del trazo rojo de la trayectoria es de 10mm, dado que la herramienta guía está dentro de la trayectoria trazada, el error visible está dentro del rango de error absoluto del sistema de seguimiento, el cual es de 0.81354mm.

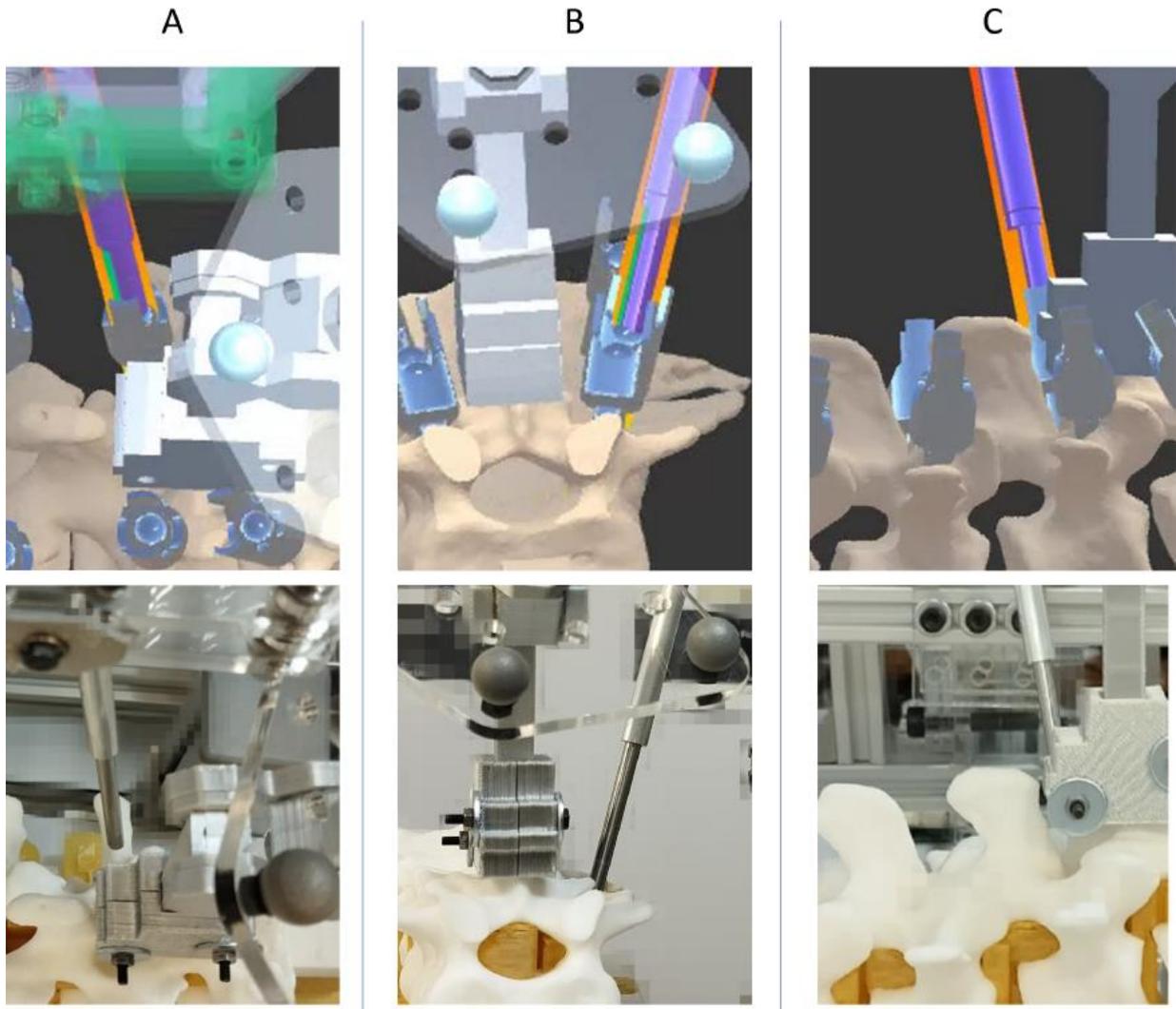


Figura 55 Acercamiento a vistas: a) Anteroposterior, b) Axial c) Lateral

7.- Conclusión

Los capítulos del trabajo proporcionaron una descripción de los requerimientos generales del sistema, desglosando los subsistemas y sus funciones específicas. Se hace énfasis en el desarrollo e implementación del sistema de seguimiento, las funcionalidades del software de visualización, así como en el diseño y construcción del prototipo robótico. Se finaliza con las pruebas y resultados obtenidos de cada sistema.

La simulación del movimiento rectilíneo permitió visualizar y analizar el comportamiento de la base móvil del robot. El cálculo y presentación de los valores angulares de las articulaciones actuadas respaldaron la viabilidad del diseño propuesto.

El prototipo robótico, con su composición híbrida de dos robots planares apilados y montados sobre una base deslizante, es capaz de girar hasta 90 grados, lo cual amplía el rango de movimiento y posibilita abordajes quirúrgicos específicos. Se subraya la resistencia del prototipo ante perturbaciones físicas externas, gracias a los controladores EPOS4 de los motores Maxon.

Los resultados estadísticos de los experimentos, presentados en la Tabla 1, evidencian una precisión aceptable del sistema, con valores de error absoluto medio (MAE) y raíz del error cuadrático medio (RMSE) inferiores a 1mm.

La implementación del entorno gráfico de Unity es una pieza clave en el despliegue del sistema robótico guiado por visión computacional. Este entorno posibilitó visualizar herramientas y estructuras anatómicas no visibles para el cirujano.

Los resultados obtenidos a modo de imágenes comparativas entre el entorno real y virtual muestran un comportamiento visualmente congruente, donde se conservan las proporciones y relación de posición entre cada elemento. Esto respalda la fidelidad del sistema, demostrando la precisión dentro del entorno controlado de pruebas.

Para consolidar la integración del prototipo robótico funcional, se requirió de precisión milimétrica en la reconstrucción tridimensional del entorno real mediante el algoritmo de alineación del sistema de seguimiento. Sin embargo, se reconoce la necesidad de mejorar la precisión del algoritmo de alineación y agregar nuevas funcionalidades al software, buscando equiparar su rendimiento con los robots comerciales, sin perder de vista que todas las modificaciones deben posibilitar que la adquisición del equipo sea viable para el sector salud del país.

Se destaca que la relación costo-beneficio del sistema mejoró en condiciones ideales en comparación con equipos profesionales comerciales. Esto se atribuye a los materiales empleados y la configuración implementada, la cual ofrece gran rigidez permitiendo que el prototipo sea económicamente viable. Sin embargo, es necesario considerar que un equipo profesional requiere utilizar materiales de grado quirúrgico, lo cual elevará el precio del sistema propuesto.

8.- Trabajos a futuro

Sistema de seguimiento:

- Refinar el algoritmo de alineación para que el sistema de seguimiento presente mejor precisión.
- Incorporar funcionalidad RGB de cámaras estéreo para implementar realidad aumentada.
- Agregar visores de realidad virtual para proporcionar una experiencia inmersiva al usuario.

Sistema robótico:

- Modificaciones a la configuración del robot para brindarle mayor rigidez al mecanismo.
- Emplear materiales de grado quirúrgico.
- Adaptar las dimensiones del robot al espacio e instrumental quirúrgico real.

La integración del sistema debe soportar los cambios realizados relativos al flujo de información de cada sistema y el flujo de trabajo de los médicos cirujanos.

Una evaluación exhaustiva cuantitativa mediante técnicas de imagen médica y valoraciones de médicos cirujanos ortopedistas, incluyendo pruebas controladas in-vitro y en modelos cadavéricos.

Referencias

- [1] F. M. C. Juárez, M. Cruz, A. Lazarini, B. H. Carbajal, J. R. Bonfil, and P. A. B. B, “Sistema de instrumentación vertebral HGM/Camarillo,” *Acta Ortop Mex*, vol. 14, no. 1, pp. 101–110, 2000.
- [2] S. Gertzbein and S. Robbins, “Accuracy of pedicular screw placement in vivo,” *Spine (Phila Pa 1976)*, vol. 15, no. 1, pp. 11–14, 1990, doi: 10.1097/00007632-199001000-00004.
- [3] M. Zdichavsky, M. Blauth, C. Knop, J. Lotz, C. Krettek, and L. Bastian, “Accuracy of pedicle screw placement in thoracic spine fractures - Part II: A retrospective analysis of 278 pedicle screws using computed tomographic scans,” *European Journal of Trauma*, vol. 30, no. 4, pp. 241–247, Aug. 2004, doi: 10.1007/S00068-004-1423-8/METRICS.
- [4] M. Zdichavsky *et al.*, “Accuracy of pedicle screw placement in thoracic spine fractures - Part I: Inter- and intraobserver reliability of the scoring system,” *European Journal of Trauma*, vol. 30, no. 4, pp. 234–240, Aug. 2004, doi: 10.1007/S00068-004-1422-9/METRICS.
- [5] W. R. Hendee, “To Err is Human: Building a Safer Health System,” *Journal of Vascular and Interventional Radiology*, vol. 12, no. 1, 2001, doi: 10.1016/s1051-0443(01)70072-3.
- [6] J. L. Goldberg, S. Kirnaz, J. A. Carnevale, L. McGrath, and R. Härtl, “History of Navigation Guided Spine Surgery,” *Technical Advances in Minimally Invasive Spine Surgery: Navigation, Robotics, Endoscopy, Augmented and Virtual Reality*, pp. 3–10, Jan. 2022, doi: 10.1007/978-981-19-0175-1_1.
- [7] B. Jaramaz, M. A. Hafez, and A. M. DiGioia, “Computer-assisted orthopaedic surgery,” *Proceedings of the IEEE*, vol. 94, no. 9, pp. 1689–1695, 2006, doi: 10.1109/JPROC.2006.880675.
- [8] G. Mao and N. Theodore, “Robotics in Spine Procedures,” *Robotics in Neurosurgery: Principles and Practice*, pp. 227–251, Jan. 2022, doi: 10.1007/978-3-031-08380-8_16.
- [9] K. G. Mensah-Brown, A. I. Yang, and J. W. Yoon, “Currently Available Robot Systems in Spinal Surgery,” *Technical Advances in Minimally Invasive Spine Surgery: Navigation, Robotics, Endoscopy, Augmented and Virtual Reality*, pp. 277–284, Jan. 2022, doi: 10.1007/978-981-19-0175-1_24.
- [10] Y. S. Lee and K. T. Kim, “Evidence of Navigation-Guided/ Robot-Assisted Spinal Surgery,” *Technical Advances in Minimally Invasive Spine Surgery: Navigation, Robotics, Endoscopy, Augmented and Virtual Reality*, pp. 285–293, Jan. 2022, doi: 10.1007/978-981-19-0175-1_25.
- [11] E. Schonfeld, M. N. Stienen, and A. Veeravagu, “Future Perspective of Robot-Assisted Minimally Invasive Spine Surgery,” *Technical Advances in Minimally Invasive Spine Surgery: Navigation, Robotics, Endoscopy, Augmented and Virtual Reality*, pp. 351–364, Jan. 2022, doi: 10.1007/978-981-19-0175-1_32.

- [12] A. G. Kulkarni, P. Rathi, and P. A. Rajamani, "MI-TLIF with 3D Navigation," *Technical Advances in Minimally Invasive Spine Surgery: Navigation, Robotics, Endoscopy, Augmented and Virtual Reality*, pp. 33–52, Jan. 2022, doi: 10.1007/978-981-19-0175-1_4.
- [13] K. Ishii, N. Isogai, and H. Funao, "Navigation-Guided Spinal Fusion: MIS Fusion and Reconstruction in Complex Spine Disease and Deformity," *Technical Advances in Minimally Invasive Spine Surgery: Navigation, Robotics, Endoscopy, Augmented and Virtual Reality*, pp. 65–71, Jan. 2022, doi: 10.1007/978-981-19-0175-1_6.
- [14] B. A. Freedman, A. Nassr, and B. L. Currier, "Stereotactic Navigation in Complex Spinal Surgery: Tips and Tricks," *Oper Tech Orthop*, vol. 27, no. 4, pp. 260–268, Dec. 2017, doi: 10.1053/J.OTO.2017.09.010.
- [15] L. Tsai, "Robot analysis: the mechanics of serial and parallel manipulators," *The Mechanics of Serial and Parallel Manipulators*, p. 520, 1999, Accessed: Dec. 06, 2023. [Online]. Available: <https://www.wiley.com/en-us/Robot+Analysis%3A+The+Mechanics+of+Serial+and+Parallel+Manipulators-p-9780471325932>
- [16] "Structural synthesis and architectures," *Parallel Robots*, pp. 19–94, Jun. 2006, doi: 10.1007/1-4020-4133-0_2.
- [17] S. Blackman and A. Tuliper, "Unity Basics," *Learn Unity for Windows 10 Game Development*, pp. 67–121, 2016, doi: 10.1007/978-1-4302-6757-7_2.
- [18] B. K. P. Hor, "Closed-form solution of absolute orientation using unit quaternions," *Journal of the Optical Society of America A*, vol. 4, p. 629, 1987.
- [19] P. Corke, "Dynamics and control," *Springer Tracts in Advanced Robotics*, vol. 118, pp. 251–281, 2017, doi: 10.1007/978-3-319-54413-7_9.
- [20] P. Corke, "Robot arm kinematics," *Springer Tracts in Advanced Robotics*, vol. 118, pp. 193–228, 2017, doi: 10.1007/978-3-319-54413-7_7.
- [21] A. Plopski, N. Ienaga, and M. Sugimoto, "Tracking Systems: Calibration, Hardware, and Peripherals," *Springer Handbooks*, pp. 211–238, 2023, doi: 10.1007/978-3-030-67822-7_9/COVER.
- [22] D. Sundararajan, "Morphological Image Processing," *Digital Image Processing*, pp. 217–256, 2017, doi: 10.1007/978-981-10-6113-4_8.
- [23] Q. H. Sheng and B. Zhang, "Absolute Orientation Based on Line Coordinates," *Photogrammetric Record*, vol. 32, no. 157, pp. 12–32, Mar. 2017, doi: 10.1111/PHOR.12178.
- [24] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-Squares Fitting of Two 3-D Point Sets," *IEEE Trans Pattern Anal Mach Intell*, vol. PAMI-9, no. 5, pp. 698–700, 1987, doi: 10.1109/TPAMI.1987.4767965.
- [25] D. Kragic and K. Daniilidis, "3-D Vision for Navigation and Grasping," *Springer Handbooks*, pp. 811–824, 2016, doi: 10.1007/978-3-319-32552-1_32/COVER.

- [26] P. Corke, "Representing position and orientation," *Springer Tracts in Advanced Robotics*, vol. 118, pp. 17–62, 2017, doi: 10.1007/978-3-319-54413-7_2.
- [27] K. J. Waldron and J. Schmiedeler, "Kinematics," *Springer Handbooks*, pp. 11–36, 2016, doi: 10.1007/978-3-319-32552-1_2.
- [28] S. Axler, "Polynomials," pp. 117–130, 2015, doi: 10.1007/978-3-319-11080-6_4.
- [29] P. Corke, "Time and motion," *Springer Tracts in Advanced Robotics*, vol. 118, pp. 63–92, 2017, doi: 10.1007/978-3-319-54413-7_3.
- [30] D. H. Kim *et al.*, "Surgical anatomy techniques to the spine," p. 717, 2017, Accessed: Jan. 11, 2024. [Online]. Available: [https://books.google.com/eg/books?id=JW1uAAAAQBAJ&pg=PA734&dq=cervical+spine+anatomy&hl=en&sa=X&ved=0ahUKEwj8_crduTWAhVMGsAKHRdhBfY4ChDoAQg8MAQ#v=onepage&q=cervical spine anatomy&f=false](https://books.google.com/eg/books?id=JW1uAAAAQBAJ&pg=PA734&dq=cervical+spine+anatomy&hl=en&sa=X&ved=0ahUKEwj8_crduTWAhVMGsAKHRdhBfY4ChDoAQg8MAQ#v=onepage&q=cervical%20spine%20anatomy&f=false)
- [31] R. Rocca, R. F. B. Santacruz, C. Sassanelli, P. Rosa, L. Fumagalli, and E. Negri, "Digital Twin and Extended Reality: Strategic Approach and Practical Implementation," *Springer Handbooks*, pp. 853–880, 2023, doi: 10.1007/978-3-030-67822-7_34.
- [32] S. Kelly and K. Kumar, "Networking Concepts," *Unity Networking Fundamentals*, pp. 1–29, 2022, doi: 10.1007/978-1-4842-7358-6_1.
- [33] D. Kortenkamp, R. Simmons, and D. Brugali, "Robotic Systems Architectures and Programming," *Springer Handbooks*, pp. 283–306, 2016, doi: 10.1007/978-3-319-32552-1_12.
- [34] M. A. Arteaga, A. Gutiérrez-Giles, and J. Pliego-Jiménez, "Position, Orientation and Velocity of Rigid Robot Manipulators," *Lecture Notes in Electrical Engineering*, vol. 798, pp. 15–69, 2022, doi: 10.1007/978-3-030-85980-0_2.
- [35] S. Axler, "Inner Product Spaces," pp. 163–202, 2015, doi: 10.1007/978-3-319-11080-6_6.
- [36] S. Axler, "Trace and Determinant," pp. 295–331, 2015, doi: 10.1007/978-3-319-11080-6_10.
- [37] K. Koniarski and A. Myśliński, "Feature Point Cloud Based Registration in Augmented Reality," *Lecture Notes in Networks and Systems*, vol. 364 LNNS, pp. 418–427, 2022, doi: 10.1007/978-3-030-92604-5_37.
- [38] S. Axler, "Operators on Inner Product Spaces," pp. 203–240, 2015, doi: 10.1007/978-3-319-11080-6_7.
- [39] P. Wang, P. Wang, Z. G. Qu, Y. H. Gao, and Z. K. Shen, "A refined coherent point drift (CPD) algorithm for point set registration," *Science China Information Sciences*, vol. 54, no. 12, pp. 2639–2646, Dec. 2011, doi: 10.1007/S11432-011-4465-7/METRICS.

Anexos

8.1 Conceptos relevantes

A continuación, se definen algunos de los conceptos más utilizados en el desarrollo del algoritmo de alineación, en el desarrollo del robot y en la integración de ambos sistemas.

8.1.1 Entorno virtual

Un entorno virtual es un espacio tridimensional digital hecho por computadora que permite recrear el comportamiento de un entorno físico real, generalmente se despliega por medio de algún dispositivo de visualización como puede ser un monitor de computadora [31].

8.1.2 Websocket

Es un tipo de socket TCP que establece un punto de conexión que facilita la comunicación bidireccional mediante el protocolo TCP (Transmission Control Protocol). Este tipo de socket mantiene una conexión abierta entre cliente y servidor, lo que permite que ambos envíen mensajes simultáneamente. La comunicación se establece a través de la combinación de dirección IP y puerto, lo que permite que cada puerto del servidor pueda ejecutar un servicio distinto [32], [33].

8.1.3 Cambio de base

El cambio de base permite representar un vector \mathbf{v} en un sistema de coordenadas B , distinto al sistema de coordenadas A que esta referenciado inicialmente [34], [35], [36].

Teniendo en cuenta la nomenclatura para un vector **vector***[base referida]*.

Para obtener un vector $\mathbf{v}_{[B]}$ se debe conocer lo siguiente (todo referido al sistema A): el vector $\mathbf{v}_{[A]}$ y la matriz W compuesta por los vectores unitarios de la base B ; calculando el producto punto entre $\mathbf{v}_{[A]}$ y la inversa de la matriz W , se obtiene $\mathbf{v}_{[B]}$ que son las coordenadas del vector ahora referido al sistema B .

8.1.4 Matrices de transformación homogénea (M.T.H.)

Las matrices de transformación homogénea (4x4) integran información con una matriz de rotación (3x3), un vector de posición (3x1), un factor de escala y un vector de perspectiva.

Las matrices permiten realizar transformaciones de rotación y traslación con únicamente operaciones matriciales ordenadamente, destacando que la multiplicación de matrices no es conmutativa [27].

Las matrices de Transformación Homogénea correspondientes a traslaciones en los ejes x,y,z se muestran en las ecuaciones *Ec 14*, *Ec 15*, *Ec 16*, mientras que las matrices correspondientes a rotaciones en los ejes x,y,z se muestran en las ecuaciones *Ec 17*, *Ec 18*, *Ec 19*.

Matrices de Traslación:

$$T1(x) = \begin{pmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \dots (Ec 14)$$

$$T2(y) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \dots (Ec 15)$$

$$T3(z) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \dots (Ec 16)$$

Matrices de Rotación:

$$T4(\theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \dots (Ec 17)$$

$$T5(\theta) = \begin{pmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \dots (Ec 18)$$

$$T6(\theta) = \begin{pmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \dots (Ec 19)$$

8.2 Algoritmos de registro y alineación recurrentes en la literatura

8.2.1 Iterative Closest Point (ICP)

ICP es un algoritmo iterativo que tiene variantes, pero se conserva el concepto de aplicar rotaciones a una nube de puntos para hacerla corresponder con el modelo original de la nube de puntos analizada, para ello se requiere de una estimación inicial que en cada interacción se aplica la transformación con el objetivo de minimizar el error de alineación, sin embargo, el algoritmo podría no converger dependiendo de la estimación inicial [21], [37].

Es posible utilizar descomposición de valores singulares (SVD) para conocer la transformación entre dos bases y corregir la diferencia en las estimaciones de posición y orientación de una serie de puntos [38].

8.2.2 Coherent Point Drift (CPD)

El algoritmo refinado busca obtener una transformación que permita alinear un conjunto de puntos respecto de otro conjunto, el algoritmo utiliza un peso w asociado a la distancia euclidiana que hay de un punto en un conjunto con su correspondiente punto transformado en otro conjunto, buscando minimizar la distancia se modifica automáticamente el peso w mediante algoritmos genéticos. El algoritmo CPD refinado presenta robustez y precisión en la alineación de datos que incluyen ruido o valores atípicos [39].

8.2.3 Ajuste por Mínimos Cuadrados basado en descomposición de valores singulares (SVD)

El algoritmo tiene como objetivo encontrar la transformación que alinea dos conjuntos de puntos minimizando el error cuadrático, asumiendo que un conjunto rotado tiene un punto correspondiente en otro conjunto utilizando la descomposición de valores singulares SVD. El artículo [24] hace una comparación entre el algoritmo propuesto y otros similares e iterativos, reportando que supera en velocidad a los algoritmos iterativos, sin embargo, es susceptible a datos ruidosos.