CEA Comité Español de Automática

XLVI JORNADAS AUTOMÁTICA CARTAGENA 2025

UNIVERSIDADE DA CORUÑA

Jornadas de Automática

Simulación del manipulador PSM y herramientas del robot da Vinci usando "SimRobots"

Herreros López, Alberto[a,*]

[a]*GIR Instituto de las Tecnologías Avanzadas de la Producción de la Universidad de Valladolid*

To cite this article: Herreros López, Alberto. 2025. Simulation of PSM manipulator and tools of da Vinci robot using "SimRobots". Jornadas de Automática, 46. https://doi.org/10.17979/ja-cea.2025.46.12065

Resumen

Durante varios años se ha desarrollado el proyecto "SimRobots", colaboración entre las librerías "SinScape MultiBody" de Simulink y "Robotic Systems ToolBox" de Matlab. En este artículo se ha desarrollado herramientas para simular los manipuladores PSM (Patient Side Manipulations) y las herramientas *snake* y *caudier* del robot quirúrgico da Vinci con "SimRobots". Existen diferencias entre la estructura real del manipulador PSM y el que se puede modelar con el formato URDF (Unified Robot Description Format). El formato URDF no puede definir ejes paralelos como tiene el manipulador PSM. Este problema marca la forma de definir la cinemática directa y hallar la cinemática inversa. En cinemática directa, varios de los ejes del PSM tienen que tener el mismo valor. En cinemática inversa, esa misma restricción hace que no se puedan usar algoritmos convencionales. Ambos problemas han sido resueltos. Se ha diseñado una App para mover el TCP (Tool Center Point) del manipulador-herramienta de forma sencilla. Las herramientas diseñadas se usarán en clases de grado y máster del área de ingeniería biomédica.

Palabras clave: Tecnología robótica, Manipuladores robóticos, Ingeniería de control, Trayectorias óptimas

Simulation of the PSM manipulator and tools of the da Vinci robot using "SimRobots"

Abstract

For several years, the "SimRobots" project has been developed as a collaboration between the "SimScape MultiBody" libraries from Simulink and Matlab's "Robotic Systems Toolbox". This article presents tools for simulating the PSM (Patient Side Manipulations) manipulators and the snake and caudier tools of the da Vinci surgical robot using "SimRobots". There are differences between the actual structure of the PSM manipulator and the one that can be modeled using the URDF (Unified Robot Description Format). The URDF format cannot define parallel axes like those in the PSM manipulator. This limitation influences how direct kinematics is defined and how inverse kinematics is calculated. In direct kinematics, several of the PSM's joints must share the same value. In inverse kinematics, that same restriction prevents the use of conventional algorithms. Both problems have been resolved. An app has been designed to easily move the manipulator-tool's TCP (Tool Center Point). The developed tools will be used in undergraduate and master's level classes in the biomedical engineering field.

Keywords: Robotic Technology, Robotic Manipulators, Automatic Control, Optimal Trajectory.

1. Introducción

La robótica es un área de conocimiento en claro auge en los últimos años y su estudio es cada vez más frecuente en muchos grados de ingeniería. La educación práctica de la robótica es compleja, ya que la mayoría de las marcas de robot industriales tienen su propio software con fines industriales. Esto hace que el alumno no sepa relacionar los conceptos aprendidos en teoría con las simulaciones y movimientos reales de los robots industriales. Sorprende que alumnos que realizan

*albher@uva.es

las prácticas de robótica sin problemas no sepan conceptos básicos sobre cinemática directa e inversa o la representación de un punto en el espacio.

Existen muy buenas aplicaciones de software para robots, pero la mayoría de ellas son muy complejas desde un punto de vista informático. RobotStudio de ABB (ABB, 2025), ROS (Robot Operative System) (ROS, 2025) o incluso "Robotic Systems Toolbox" de Matlab (MathWorks, 2024b) son ejemplos de herramientas que no pueden darse en asignaturas con un número de créditos relativamente bajo.

Con las ideas de Corke (Corke, 2016) y de Arturo Gil (Gil et al., 2015) y (Gil, 2025) se ha desarrollado en los últimos años el proyecto "SimRobots" (Arévalo and Herreros, 2021), (Herreros, 2024) y (Herreros, 2025). Este proyecto trata de usar las librerías "SimScape Multibody" de Simulink (MathWorks, 2024c) y "Robotic Systems Toolbox" de Matlab (MathWorks, 2024b) para crear un objeto controlador de los movimientos de un robot.

Ambas librerías pueden importar ficheros de robots en formato URDF (Unified Robot Description Format) usados también en ROS (Robotic Operative System) (ROS, 2025). La librería de Simulink puede simular de forma muy precisa las trayectorias de un robot, pero está limitada en el calculo de las mismas. La librería de Matlab puede obtener las trayectorias en cinemática directa y cinemática inversa usando algoritmos clásicos (Barrientos et al., 2007).

En este artículo se ha desarrollado distintas herramientas para la simulación en cinemática directa e inversa de los manipuladores PSM y herramientas *snake* y *caudier* del robot quirúrgico da Vinci (Fontanelli et al., 2017). Los ficheros URDF del robot pueden obtenerse en la red en (DKRK, 2025).

La estructura del manipulador PSM (Patient Side Manipulations) no es la misma que la real, ya que el manipulador real tiene articulaciones paralelas que no se pueden diseñar en URDF (Batto et al., 2025). Por ello, las articulaciones del robot simulado tienen restricciones en el movimiento de 5 de sus ejes, que deben ser iguales en módulo para conseguir que el manipulador pivote alrededor del punto RCM (Remote Center Point) (Fontanelli et al., 2017) de la misma forma que el real.

Este hecho dificulta la obtención de la cinemática inversa, que ha sido obtenida con un algoritmo propio con buen resultado. Se ha diseñado una App con (MathWorks, 2024a) para mover de forma incremental la posición y rotación del TCP del manipulador-herramienta. Esta aplicación será usada en asignaturas de grado y máster relacionadas con la ingeniería biomédica, y el software será incorporado a la página de la red de "SimRobots" (Herreros, 2025).

2. Proyecto "SimRobots"

El proyecto "SimRobots" es una colaboración entre las librerías "SimScape MultiBody" de Simulink y "Robotic Systems Toolbox" de Matlab (Herreros, 2024) y (Herreros, 2025). Ambas librerías puede importar robots en formato URDF (Unified Robot Description Format). Con el robot en formato "SimScape MutiBody" se puede simular la cinemática directa del robot y visualizar los movimientos en un entorno gráfico "Mechanics Explorers".

Estos ficheros pueden ser importados al objeto *RigidBody* de "Robotic Systems Toolbox". Esta librería es usada para analizar trayectorias de cinemática directa e inversa para generar los movimientos del robot.

El proyecto "SimRobots" puede generar estaciones con elementos estáticos y con robots como la mostrada en la Figura 1.

El objeto controlador *Cin*(*RigidBody*, *MultiBody*, ...) puede simular un robot en su cinemática directa e inversa. Son principales argumentos son el objeto *RigidBody* de "Robotic Systems Toolbox" y el fichero Simulink con el modelo *MultiBody*. Se pueden realizar movimientos rectilíneos y circulares como se muestra en la Figura 2. Se puede simular la toma de piezas o el dejarla por la copia y pega de elementos estáticos de un icono a otro como se muestra en la Figura 3.

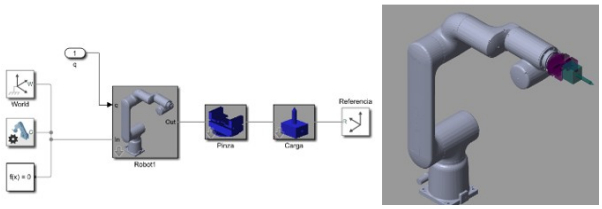

Figura 1: Icono robot con herramienta y carga.

Una estación puede disponer de varios robots con controladores diferentes, distintos objetos *Cin*(). Sus movimientos pueden ser secuenciales o simultáneos como se muestra en la Figura 4.

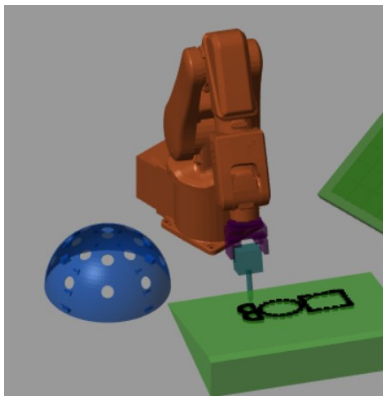

Figura 2: Trajectoria lineal y circular.

Los principales métodos del objeto *Cin*() son:

- *MoveAbsJ*(): Mueve el robot por ejes en cinemática directa.
- *MoveJ*(): Mueve el TCP del robot a un punto y orientación dados en cinemática inversa.
- *MoveL*(): Mueve el TCP del robot a un punto y orientación dados en línea recta.
- *MoveC*(): Mueve el TCP del robot a dos posiciones dadas en un arco de semicircunferencia.
- *Rec*(), *Rep*(): Graban y reproducen una serie de movimientos.

- *RepRob*(): Reproducen los movimientos guardados en varios robots de forma simultánea.

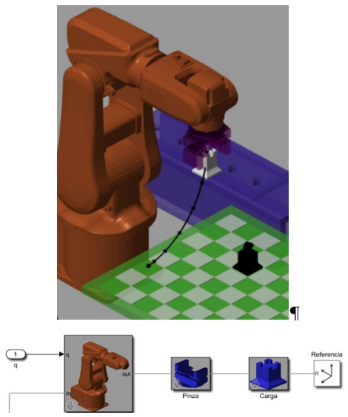

Figura 3: Pegar en *Carga* el icono que tenga la torre y mover.

Esta herramienta se pueden diseñar robot dinámicos realimentados. En ese caso las entradas de las articulaciones serán los pares de fuerza y la salida del robot, posición y velocidad, se realimentará para conseguir que el mismo siga una referencia, como se ve en la Figura 5. En este caso, se ha usado un controlador PD, con dos lazos de control, el de la posición y el de la velocidad.

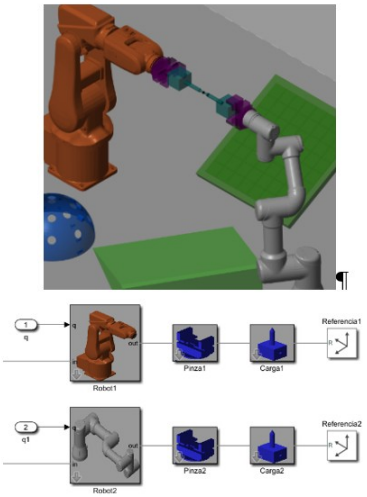

Figura 4: Estación Muli-Robot.

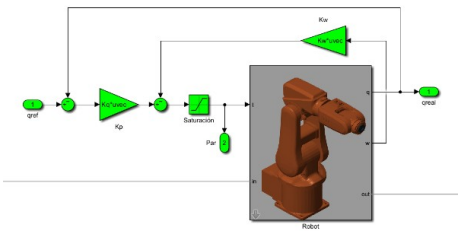

Figura 5: Robot dinámico realimentado con un PD.

3. Descripción del robot da Vinci con "SimRobots"

El robot da Vinci consta de dos partes fundamentales, la plataforma desde donde se controla el robot y el conjunto de brazos robóticos y manipuladores PSM (Patient Side Manipulations) que actúan sobre el paciente, como se describe en Fontanelli et al. (2017). La plataforma desde donde el médico controla el robot dispone de dos manipuladores MTM (Master Tool Manipulations). Un sistema multirobot posicionan los manipuladores PSM (Patient Side Manipulations), unos para operar y otro endoscópico para ver la operación como se ve en la Figura 6.

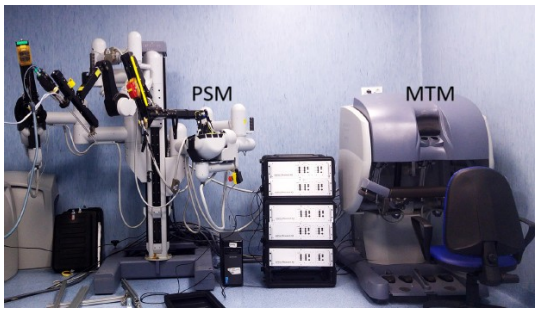

Figura 6: Robot da Vinci sus manipuladores MTM y PSMs (Fontanelli et al., 2017).

En este artículo se pretende usar el proyecto "SimRobots" para modelar los manipuladores PSM y sus herramientas tijera (*Snake*) y pinza (*Caudier*). Los ficheros URFD de estas modelos se hallan en la red en (DKRK, 2025).

3.1. *Manipulador PSM (Patient Side Manipulations)*

El principal objetivo del manipulador PSM es introducir y mover el brazo final y herramienta del manipulador en el cuerpo del paciente pivotando a partir de un punto fijo de entrada (Fontanelli et al., 2017). El esquema del manipulador PSM mostrado en la Figura 7 se observa que el punto RCM (Remote Center of Motion) permanece constante ante los movimientos de sus ejes. Este hecho se consigue usando articulaciones paralelas que fuerzan a que los movimientos de los ejes 2, 2′ y 2″ sean iguales en módulo.

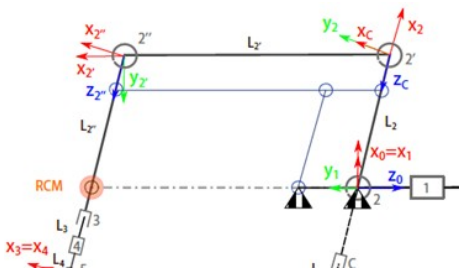

Figura 7: Estructura de los ejes del PSM (Fontanelli et al., 2017).

Sin embargo, el formato URDF no permite el modelado en lazos paralelos (Batto et al., 2025). Por ello, el modelado de este manipulador se realiza en estructura de árbol moviendo 5 de sus ejes con el mismo módulo. En la Figura 8 se pueden ver estos 5 ejes, $2-2'$, $3-3'$ y 4 con el mismo módulo para simular los ejes paralelos del manipulador real. El eje $2'$ (en rojo)

debería estar unido al 3 y el 3′ (en rojo) al 4 para ser una estructura paralela, pero están sueltos. Los ejes en rojo (sueltos) de la Figura 8 corresponden con ejes en rojo de la Figura 9. Por tanto, se tiene en conjunto un eje 1 balancín, las secuencia de los 5 ejes paralelos, un eje 7 prismático y un eje 8 de rotación final. Es decir, 8 ejes con 4 grados de libertad (DOFs). Esto va a causar problemas en el cálculo de la cinemática inversa.

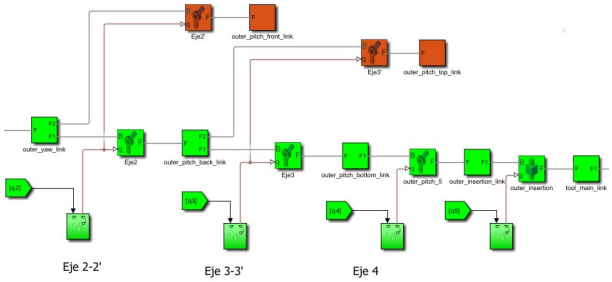

Figura 8: Manipulador PSM con 4 DOFs y 8 ejes.

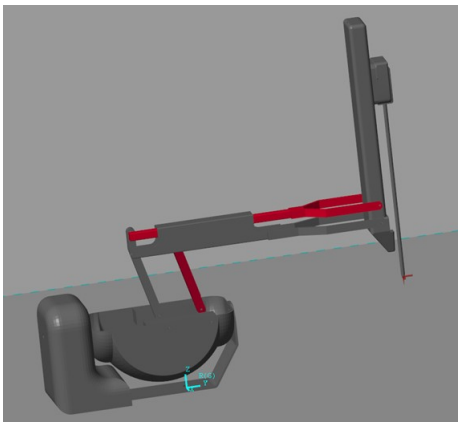

Figura 9: Manupulador PSM con 4 DOFs.

Este manipulador puede llevar dos herramientas, una tijera (*snake*) y una pinza (*caudier*).

3.2. *Herramienta snake*

La herramienta *snake* consta de 6 ejes rotacionales, pero los dos últimos deben ser simétricos, ya que es el movimiento de abrir y cerrar las tijeras. Los dos últimos ejes, en rojo en la Figura 10, no son útiles para mover el TCP a un punto dado, solo abre y cierran la herramienta. En la Figura 10 se muestra la estructura de los últimos ejes y en la Figura 11 su representación. La herramienta es muy pequeña comparada con el manipulador PSM, como se aprecia por el tamaño relativo del puntero TCP (en rojo) respecto del manipulador, Figura 9, y la herramienta 11.

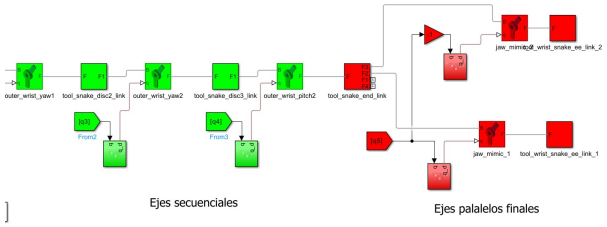

Figura 10: Estructuta de las herramienta *snake* y *caudier*.

El objetivo de los 4 ejes rotacionales primeros es la obtención de la orientación exacta del TCP final como se verá en el estudio de la cinemática inversa.

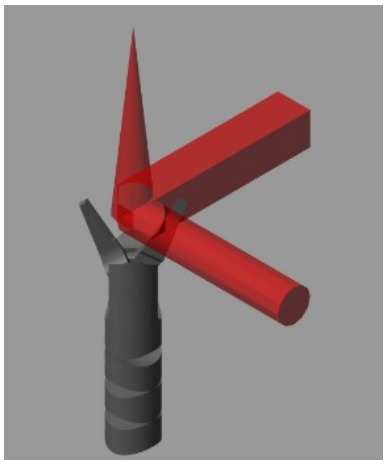

Figura 11: Herramienta *snake* con 5 DOFs.

3.3. *Herramienta caudier*

La herramienta *caudier* consta de 4 ejes rotacionales y 3 DOFs ya que los dos últimos son simétrico para abrir y cerrar la pinza. La Figura 10 muestra la estructura de los ejes y la Figura 12 su gráfica, con un tamaño similar a la *snake*.

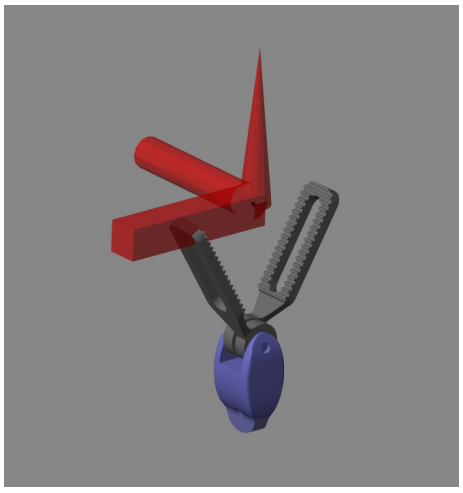

Figura 12: Herramienta *caudier* con 3 DOFs.

3.4. *Manipulador PSM con herramientas*

Se han creado dos manipuladores compuestos del manipulador PSM con las dos herramientas. En la Figura 13 se muestra el manipulador PSM y la herramienta *snake* con 9

DOFs en dos posiciones. El puntero (rojo) define el TCP del manipulador y el puntero (azul) el punto RCM. Se ha dibujado una figura estática que simula el torso de un paciente para comprobar que el punto RCM permanece constante.

La herramienta "Mechanics Explorers" de "SimScape MultiBody" tiene cámaras estáticas y dinámicas para poder grabar el movimiento del robot desde diferentes puntos de vista, como se muestra en la Figura 13.

El caso del manipulador PSM y la herramienta *caudier* es similar, pero con 7 DOFs.

Los dos manipuladores-herramienta se ha importado el objeto *RigidBody* correspondiente. De estos objetos se han eliminado los ejes $2' - 3'$ que no dan grados de libertad para que los cálculos sean más sencillos.

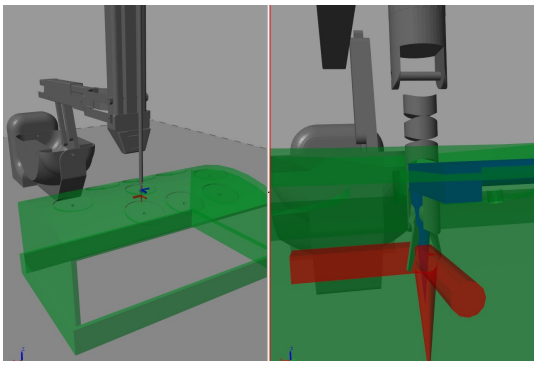

Figura 13: Manipulador PSM y *snake* con 9 DOFs.

4. Simulación cinemática directa de los manipuladores PSM y herramientas

Se ha creado una estación con dos manipuladores PSM y las respectivas herramientas, *snake* y *caudier* y el simulador de torso como se ve en las Figuras 14 y 15. Cada manipulador tiene su controlador independiente, pudiendo simular los movimientos de forma secuencial o simultánea. La simulación es compleja ya que el primer manipulador tiene 9 grados de libertad (DOFs) y el segundo 7 DOFs.

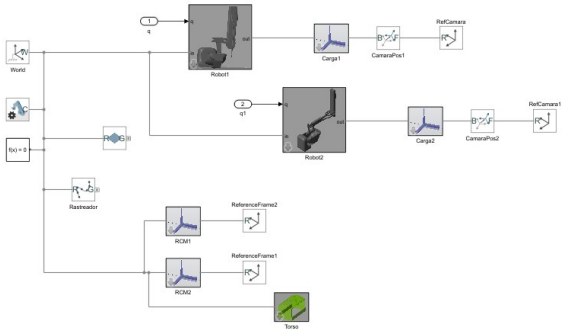

Figura 14: Dos manipuladores PSM, con herramientas *snake* y *caudier* respectivamente.

Los 3 primeros grados de libertad (5 ejes reales) del PSM son los destinados a la obtención de la posición del TCP mientras que el último eje del PSM y los ejes de la herramienta menos el último a la orientación del TCP. El último eje de la herramienta es el que abre o cierra la herramienta. Este hecho se va a tener en cuenta en el cálculo de la cinemática inversa.

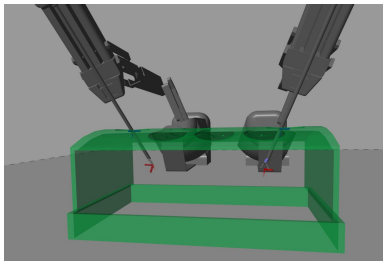

Figura 15: Dos manipuladores y herramientas simuladas.

5. Cinemática inversa de los manipuladores

La cinemática inversa de los dos manipuladores con sus herramientas es compleja, ya que 3 de sus eje (ejes $2 - 3 - 4$) deben tener el mismo módulo para que el manipulador pivote sobre el punto RCM. Notar que los ejes $2' - 3'$ ya han sido eliminados del objeto *RigidBody* como se comentó en la sección anterior. Normalmente, la cinemática inversa se obtiene de los objetos *RigidBody* importados desde los ficheros Simulink "Simscape MultiBody". Sin embargo, al tener la restricción de los ejes $2 - 3 - 4$ esta técnica no puede ser usada. Por ello, se ha optado por usar un método particular de optimización para obtener la cinemática inversa.

Los 5 primeros ejes del manipulador PSM, de los cuales 3 de ellos iguales, son los usados para optimizar la posición final del TCP del manipulador-herramienta. El resto de ejes menos el último son los usados para optimizar la rotación del TCP del manipulador-herramienta. El último eje solo se usa para abrir y cerrar la herramienta.

Unir los objetivos de posición y orientación en un único índice es también complejo, ya que las unidades de posición y orientación son diferentes y es difícil definir con precisión los pesos de cada factor.

Por ello se ha dividido la optimización en dos fases. La primera optimiza la posición del TCP y la segunda optimiza su orientación. Como la herramienta es muy pequeña, la optimización de la orientación modifica muy poco la posición del TCP. Sin embargo, es interesante repetir el proceso varias veces para que el error (cuadrático) sea mínimo como se verá en el ejemplo. Un problema en la optimización de la orientación es el intervalo de búsqueda, que se ha seleccionado entre $[-\pi, \pi]$.

En un ejemplo con pseudo-código es el siguiente. Se pretende llevar al TCP de los manipuladores-herramientas que parte de una posición de ejes *joint* a una posición cartesiana posición-rotación *pose*, que es la base del simulador de tórax, como se muestra en la Figura 16. Primero se optimiza los 5 primeros ejes respecto a la posición, y luego el resto menos el último para la rotación, y se da 3 vueltas.

```
for i= 1:N
  x0= joint(1:5);
 joint(1:5)= fmin(@(x)mPos(x,pose,joint),x0);
 x0= joint(6:end-1);
 joint(6:end-1)= fmin(@(x)mRot(x,pose,joint),x0);
end
```

Los errores cuadráticos entre la posición-rotación (*pose*) deseada y la obtenida queda reflejada en la Tabla 1 para ambos manipuladores-herramientas. Se parte de un error considerable en la fila $N = 0$, que en una vuelta queda reducido mucho y en tres el error es mínimo. El resultado puede verse en la Figura 16 donde en la parte derecha el TCP de ambos manipuladores-herramientas están solapados en la misma *pose*. Además, los punteros azules de la Figura 16 izquierda muestra que los puntos RCM de ambos manipuladores permanece fijos en las dos entradas al tórax simulado.

Tabla 1: Errores cuadráticos de los dos manipuladores

N	PSM-Snake	PSM-Caudier
0	2,23	4,9
1	0,0318	0,0378
2	0,0014	0,0013
3	$1{,}08e^{-5}$	$4{,}5e^{-5}$

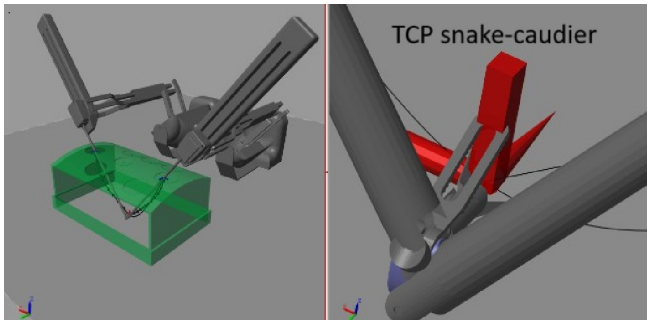

Figura 16: Los dos manipuladores-herramientas en la misma posición y orientación.

6. App de control

Se ha diseñado una App de control de cada uno de los manipuladores-herramienta. Con dicha App podemos incrementar la posición y orientación de cada uno de los manipuladores-herramienta de forma individual. La Figura 17 muestra dicha App, con la que se puede mover el robot de forma incremental respecto de un objeto de trabajo. Si el objeto de trabajo es *Tool* el manipulador-herramienta se mueve respecto del TCP del mismo.

Se puede incrementar las posiciones y las orientaciones de TCP. Se puede modificar el máximo incremento deseado tanto en posición como orientación. Además, se puede ver y modificar los parámetros absolutos de los ejes del manipulador-herramienta (tecla *MoveAbsJ*). Se puede importar y exportar el actual punto y orientación del TCP.

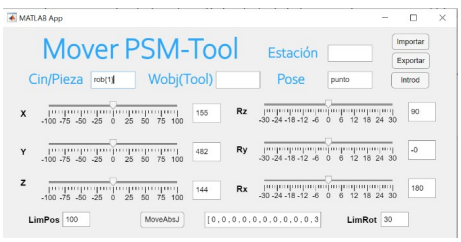

Figura 17: App de control de posición y orientación.

7. Conclusiones

Este artículo ha desarrollado herramientas para simular los manipuladores PSM y sus herramientas *snake* y *caudier* del robot da Vinci con el proyecto "SimRobots". Ha habido que resolver problemas en su cinemática directa e inversa, solucionados satisfactoriamente. Se ha diseñado una App para que el movimiento de los manipuladores sea más sencillo.

En el futuro se puede intentar diseñar los manipuladores MTM que en el robot real mueven a los manipuladores PSM y relacionar los movimientos de ambos. Otra posible aplicación es la simulación dinámica con realimentación del sistema como se muestra en la Figura 5. Para ello sería interesante construir un modelo paralelo a partir del actual modelo, hecho que es viable con "Simscape MultiBody".

El software realizado será incorporado a la página de la red (Herreros, 2025) como el resto de novedades del proyecto "SimRobots" de este curso.

Agradecimientos

Este trabajo ha sido realizado dentro del grupo GIR Grupo de Investigación de Tecnologías Avanzadas de la Producción de la Universidad de Valladolid.

Referencias

ABB, 2025. Manual de operador. ID de documento: 3HAC032104-005.

Arévalo, S., Herreros, A., 2021. Aplicaciones kkk de matlab y simulink para estaciones robóticas. XLII Jornadas de Automática: libro de actas, 625–631.
DOI: 10.17979/spudc.9788497498043.625

Barrientos, A., Peñin, L., Balaguer, C., Santoja, R., 2007. Fundamentos de Robótica (2º edición). MacGraw-Hill.

Batto, V., Matteis, L., Mansard, N., 2025. Extended urdf: Accounting for parallel mechanism in robot description. RAAD 2025 - 34th International Conference on Robotics in Alpe-AdriaDanube Region.

Corke, P., 2016. Corke, P. Robotics, Vision and Control. Fundamental Algorithms in MATLAB. Spriger.
DOI: 10.1007/978-3-319-54413-7

DKRK, 2025. John Hopkins University DVRK controller git repositories. [Online] Available: https://github.com/jhu-dvrk accessed: 2025-05-01.

Fontanelli, G., Ficuciello, F., Villani, L., Siciliano, B., 2017. Modelling and identification of the da vinci research kit robotic arms. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 2017, 1464–1469.
DOI: 10.1109/IROS.2017.8205948

Gil, A., 2025. Python toolbox focussed on robotic manipulators pyARTE (https://github.com/4rtur1t0/pyARTE) accessed: 2024-05-30.

Gil, A., Reinoso, O., Marin, J., Paya, L., Ruiz, J., 2015. Development and deployment of a new robotics toolbox for education. computer applications in engineering education. Computer Applications in Engineering Education. Ed. Wiley ISSN:1099-0542 1061-3773, 443–454.

Herreros, A., 2024. Proyecto "simrobots", colaboración entre "simscape multibody" y "robotic system toolbox". Jornadas de Automática.
DOI: https://doi.org/10.17979/ja-cea.2024.45.10770

Herreros, A., 2025. Proyecto "SimRobots", Colaboración entre "SimScape Multibody" y "Robotic System Toolbox". http://https://github.com/albher) accessed: 2025-05-1. Lugar de publicación.

MathWorks, 2024a. MATLAB® App Building.

MathWorks, 2024b. Robotics System Toolbox™ User Guide. Lugar de publicación.

MathWorks, 2024c. Simscape™ Multibody™ User's Guide. Lugar de publicación.

ROS, 2025. ROS (Robotic Operative System) Available: https://www.ros.org/ accessed: 2025-05-01.