

Herramienta software de ayuda al diseño de sistemas robotizados de filmación

Carlos Reca-Luque¹, José Luis Torres-Moreno¹, José Domingo Álvarez-Hervás², Antonio Giménez¹

¹Grupo de investigación ARM, Departamento de Ingeniería, Universidad de Almería, España. Email: jltmoreno@ual.es

²Grupo de investigación ARM, Departamento de Informática, Universidad de Almería, España. Email: jhervas@ual.es

Resumen

En los últimos años se han podido ver numerosas innovaciones tecnológicas asociadas al manejo y operación de videocámaras. En sectores como el cinematográfico, la publicidad o incluso eventos deportivos y culturales, ha aumentado la utilización de sistemas robotizados y cámaras grúas que permiten, entre otras ventajas, garantizar la repetitividad de maniobras, así como incrementar su precisión e incluso el alcance y la velocidad de operación. No obstante, hasta donde han llegado los autores, la oferta comercial de este tipo de sistemas es aún limitada. Por este motivo, mediante este trabajo se pretende potenciar el diseño y desarrollo de nuevos sistemas robotizados a partir de la implementación de una aplicación informática elaborada en Matlab que permite la obtención de la cinemática y dinámica inversa del robot en cuestión, de manera que se disponga información sobre las velocidades alcanzadas y los pares necesarios para la realización de las maniobras deseadas

Palabras clave: Brazo robótico; diseño; simulación; cinematografía; software;

Abstract

In the last few years we have seen many technological innovations associated with the use and operation of video cameras. In fields such as cinematography, advertisement or even sport and cultural events, the use of robotic systems and camera cranes has increased due to the fact that they allow, among other advantages, to guarantee the repeatability of movements, as well as to increase their precision and even the range and speed of operation. However, as far as the authors' research has reached, the commercial offer of this type of systems is still limited. For this reason, the purpose of this work is to promote the design and development of new robotic systems through the implementation of a computer application developed in Matlab that allows to obtain the kinematics and inverse dynamics of the robot in consideration, therefore providing information on the speeds reached and the torques required to perform the desired movements.

Keywords: Robotic arm; design; simulation; cinematography; software.

1. Introducción

Durante las últimas décadas ha existido una expansión en el empleo de los robots en diversos ámbitos de la actividad humana. Si bien esta tecnología encuentra su mayor acogida en el sector industrial, uno de los campos en el cual existe una menor penetración de estos es en la cinematografía. Con el ánimo de impulsar el empleo de estos sistemas en dicho ámbito, durante los últimos años se están desarrollando diversos estudios que buscan la robotización de las labores de filmación mediante el empleo de tres tipos de tecnologías principalmente: a) los robots cámara

controlados por cable, b) los vehículos aéreos no tripulados (UAV) y, c) los brazos robotizados.

Los robots cámara controlados por cable suspendido son unos dispositivos que permiten la sujeción y el control de la posición espacial de la cámara a partir de la conexión generalmente de cuatro cables a la base la cual sostiene el robot. Trabajos como el realizado por Wang et al. en 2013, buscan el diseño e implementación de un dispositivo de control capaz de posicionar y orientar de manera adecuada a la cámara para que pueda realizar las labores de grabación a la cual está destinada. Por otro lado, si se desea realizar el rodaje aéreo con mayor maniobrabilidad y simplicidad

estructural, el empleo de vehículos aéreos no tripulados es una opción sobre la cual se están desarrollando numerosas investigaciones. Por un lado, se está desarrollando metodologías capaces de planificar trayectorias óptimas con un equipo de drones que realicen cinematografía autónoma, como es el caso del estudio realizado por Alcantara et al. en el año 2021 [1]. Por otro lado, se está estudiando los requisitos de hardware y software para la detección automática de objetivos que permitan realizar un seguimiento de este [2], e incluso se están desarrollando modelos que permiten la determinación analítica de la distancia focal máxima permitida por la cámara para garantizar que el seguimiento visual del objetivo sea el adecuado [3].

Finalmente, para la realización de tomas desde tierra la opción empleada generalmente es el empleo de brazos robotizados. Estos presentan una destreza similar a la de un humano, pero con la posibilidad de realizarlas a velocidades tanto superiores como inferiores, así como de repetir los movimientos realizados para poder elaborar diferentes tomas de un mismo plano. Dentro de este campo la mayoría de las investigaciones se centran en tres objetivos fundamentales, por un lado, en el diseño de nuevos diseños de robots [4], [5]. Por otro lado, el desarrollo de diversas metodologías para la mejora en la resolución del problema cinemático inverso, donde adquiere especial relevancia el empleo de algoritmos genéticos [6]–[8]. Y, finalmente, el cálculo de la posición y las trayectorias asociadas a una secuencia de rodaje [9]–[11].

Como se puede advertir, las diferentes investigaciones realizadas respecto a este último tipo de sistemas robotizados están enfocadas a nuevos desarrollos enfocados en mejoras específicas en ciertos campos del proceso de diseño y cálculo de estos. Con el objetivo de ofrecer una aplicación integral que permita realizar un análisis completo del robot, en el presente trabajo se presenta el diseño de una interfaz desarrollada dentro del entorno de Matlab que permite la obtención del análisis cinemático y dinámico dentro del contexto de la cinematografía. Además, se presentará un ejemplo de aplicación asociado que permitirá validar la herramienta seleccionada frente a programas comerciales.

2. Materiales y métodos

2.1. Herramientas matemáticas

En este apartado se abordará de manera breve las diferentes herramientas matemáticas empleadas para el cálculo desarrollado por el programa para el brazo robótico a analizar. Para ello, se definirán tres subapartados relativos a la parte cinemática, la parte de interpolación de trayectorias y, finalmente, a la parte de cálculo dinámico.

2.1.1. Herramientas para el cálculo cinemático

Para poder realizar el cálculo cinemático, es fundamental establecer una relación matemática entre el sistema de referencia de la base de este con el sistema de referencia del extremo, de tal manera que esta relación se encuentre expresada en términos de las coordenadas articulares de este. De esta forma sería posible realizar el cálculo de la pose del extremo conocidas los valores articulares, lo que se denomina problema cinemático directo (PCD), o viceversa, denominado el problema cinemático inverso (PCI). Esta correspondencia se realiza mediante el empleo de lo que se conoce como la matriz de transformación homogénea, indicada en la ecuación 1. Esta formulación matemática permite, en robótica, establecer la relación respecto a la rotación (matriz $[R]_{3 \times 3}$) y la traslación (matriz $[T]_{3 \times 1}$). La construcción de esta matriz global de transformación se obtendrá a través del producto sucesivo de las matrices de transformación individuales definidas para cada eslabón.

$$\vec{p} = \begin{bmatrix} [R]_{3 \times 3} & [T]_{3 \times 1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} \quad (1)$$

En el caso del problema cinemático directo, bastaría con sustituir las variables articulares por el valor deseado para obtener numéricamente la orientación del extremo, indicada en la matriz de rotación, y la posición de este, indicada en la matriz de traslación.

Para la resolución del problema inverso existen dos estrategias de resolución: a) aplicando métodos numéricos iterativos o b) aplicando soluciones analíticas *ad hoc* para cada robot en concreto. En el caso del programa confeccionado, se ha optado por emplear métodos numéricos iterativos debido a que estos permiten trabajar sistemáticamente del mismo modo independientemente del tipo de robot. Para ello se establecen 12 ecuaciones, 9 de la matriz de rotación y 3 de la posición, a partir de la igualación de la matriz de transformación homogénea global en función de las coordenadas articulares calculada inicialmente, con la matriz homogénea numérica formada a través de la pose deseada en el extremo. Se debe tener en cuenta que aunque el número de ecuaciones de la matriz de rotación sean nueve, se puede expresar la misma información únicamente con tres variables, los ángulos de Euler, lo que nos indica que dichas ecuaciones se encuentran relacionadas entre sí y, por lo tanto, el número de ecuaciones efectivas para saber si el sistema presenta más incógnitas que ecuaciones sería seis. Por tanto, el método iterativo seleccionado buscará minimizar las diferencias existentes en esas 12 ecuaciones, llegando a una solución de compromiso en el caso de que el número de grados de libertad del brazo sea inferior a 6 o, a una solución particular fuertemente influenciada por el punto de partida si el número de grados de libertad es superior o igual a seis. Este sistema de ecuaciones será resuelto dentro del

programa a través del empleo de una de las dos funciones de Matlab implementadas: a) la función *fsolve* y, b) la función *lsqnonlin*. Ambas funciones pueden realizar el cálculo sin restricciones, sin embargo, sólo la segunda función permite la integración de las restricciones articulares propias del robot [12], [13].

2.1.2. Herramientas para la interpolación de las trayectorias

Para la realización de la interpolación de trayectorias mediante el programa elaborado se deberán de facilitar una serie de poses de la cámara, que describan la trayectoria de rodaje deseada. Para poder obtener una evolución suave del movimiento de la cámara que permita la transición de una pose definida a la siguiente, mediante la aplicación del PCI indicada en el subapartado anterior, se obtienen las coordenadas articulares asociadas a cada pose indicadas y, posteriormente, se calculan las ecuaciones de interpolación entre una coordenada articular y la siguiente para cada una de las articulaciones del robot. De tal manera que la función interpoladora empleada es una *spline* cúbica donde el criterio de velocidades se obtiene mediante el método de Craig [12]. La ecuación interpoladora se indica en la ecuación 2 junto con la definición de los coeficientes asociados (Ec. 3-6).

$$q(t) = a + b \cdot (t - t^i) + c \cdot (t - t^i)^2 + d \cdot (t - t^i)^3 \quad (2)$$

Donde los coeficientes son:

$$a = q^i \quad (3)$$

$$b = \dot{q}^i \quad (4)$$

$$c = \frac{3}{T^2} \cdot (q^{i+1} - q^i) - \frac{1}{T} \cdot (\dot{q}^{i+1} + 2\dot{q}^i) \quad (5)$$

$$d = -\frac{2}{T^2} \cdot (q^{i+1} - q^i) + \frac{1}{T^2} \cdot (\dot{q}^{i+1} + \dot{q}^i) \quad (6)$$

Además, a partir de la ecuación interpoladora se puede obtener las curvas de velocidades y aceleraciones asociadas a dicho movimiento mediante la derivada temporal primera y segunda respectivamente, como se puede apreciar en las ecuaciones 7 y 8.

$$qd(t) = b + 2c \cdot (t - t^i) + 3d \cdot (t - t^i)^2 \quad (7)$$

$$qdd(t) = 2c + 6d \cdot (t - t^i) \quad (8)$$

2.1.3. Herramientas para el cálculo dinámico

Para la realización del cálculo de la dinámica inversa, se ha confeccionado una función dentro del programa que permite el cálculo de los esfuerzos en las articulaciones del robot conocida la trayectoria interpolada del robot a partir del empleo del método de Newton-Euler. Esta formulación permite el cálculo dinámico mediante la aplicación de operaciones vectoriales ligadas al equilibrio de fuerzas y pares de cada eslabón del robot (véase las ecuaciones 9 y 10). A partir de este razonamiento, el método consta de una serie de pasos ordenados que permiten la

determinación de los esfuerzos del robot a partir del conocimiento de la masa, centro de masas e inercias de cada uno de los eslabones que componen a el robot. Estos pasos pueden consultarse en el libro "Fundamentos de la robótica" [12].

$$\sum f_i = m_i \cdot \dot{v}_i \quad (9)$$

$$\sum \tau_i = I_i \cdot \dot{\omega}_i + \omega_i \times (I_i \cdot \omega_i) \quad (10)$$

donde, f_i , son las fuerzas ejercidas en la barra i , τ_i , los pares ejercidos sobre la barra i en torno a su centro de masas, m_i , es la masa del eslabón i , I_i , es el tensor de inercia de la barra i entorno a su centro de masas, expresado en el sistema de referencia S_i ; \dot{v}_i , es la aceleración lineal del centro de masas de la articulación i , ω_i , es la velocidad angular de la articulación i y, $\dot{\omega}_i$, es la aceleración angular de la articulación i .

2.2. El brazo robotico estudiado

Con el objetivo de realizar una verificación de los códigos desarrollados a partir de los fundamentos expuestos, se definirá un brazo robótico cuya geometría debe estar fuertemente ligada a la aplicación que va a presentar, la cinematografía. A la hora de determinar la configuración de los ejes principales más adecuada, parece razonable pensar que el brazo robótico a emplear debe ser aquel que más necesidades sea capaz de satisfacer. Por lo tanto, este debería presentar una maniobrabilidad y velocidad así como un volumen de trabajo equivalente a un soporte humano, es decir, un operador que porte una cámara. Por otro lado, otra cualidad importante, aunque no fundamental, sería tener la capacidad de trasladarse a través de un eje, de tal forma que pudiera cubrir las necesidades equivalentes a las del travelling o Dolly. Finalmente, este brazo robótico debería presentar un espacio en planta reducido respecto a su volumen de trabajo, para permitir que el equipo de rodaje se pueda adaptar fácilmente a diferentes tipos de platós de cine. En virtud de lo comentado anteriormente, el empleo de una configuración de robot tipo TRR sería lo más adecuado.

En cuanto a las características geométricas generales de altura y capacidad de carga en el extremo, se puede acudir a las dos grandes compañías que ofertan este tipo de producto al mercado cinematográfico: Mark Roberts Motion Control (MRMC) y Motorized Precision. En ambos casos, se establecen tres segmentos de robots en función de dichas características geométricas de tal manera que se adapten a diferentes requerimientos de rodaje. Obviando la categoría de robots más pequeños, la geometría debería presentar una longitud de brazo de entre dos y tres metros así como una capacidad de carga de entorno 20 kg.

Por lo tanto, con el fin de trabajar con brazos robóticos contrastados en el mercado, se va a acudir al empleo de un brazo robótico comercial de la compañía ABB propio de la aplicación industrial para llevarlo al campo

de la cinematografía. El robot seleccionado teniendo en cuenta los requerimientos anteriormente definidos es el IRB 4600-20/2.50, cuya capacidad de carga es de 20kg y una altura de brazo de 2.50 metros. Por otro lado, con el objetivo de realizar un análisis lo más realista posible a nivel dinámico, parece conveniente la instalación de una cámara de grabación profesional empleada en la industria cinematográfica dentro del modelo del robot. Para ello, se ha modelado mediante el software CAD *SolidWorks* tanto la cámara digital *Sony Venice* como un soporte asociado. De tal manera que el conjunto robotizado empleado para la verificación del programa es el indicado en la figura 1.



Figura 1 Sistema robotizado de rodaje cinematográfico.

2.3. Definición de trayectoria

La trayectoria empleada en el caso del robot IRB 4600-20/2.50 se trata de la elaboración de un nuevo plano correspondiente a una de las secuencias de la película del año 1969 denominada *Patton*, cuyo rodaje se realizó en parte en la provincia de Almería, al igual que muchos otros grandes éxitos de la época. Para ello, partiendo de la secuencia original mostrada en la figura 2, se propone el rodaje de un plano que realice la grabación del lateral del vehículo militar norteamericano “Jeep Willys” para finalizar situando la cámara al lado de la cabeza del conductor, de tal forma que se permita la grabación de la profundidad de la carretera a través de la luna del propio vehículo (véase figura 3).



Figura 2 Secuencia real de la película *Patton*.

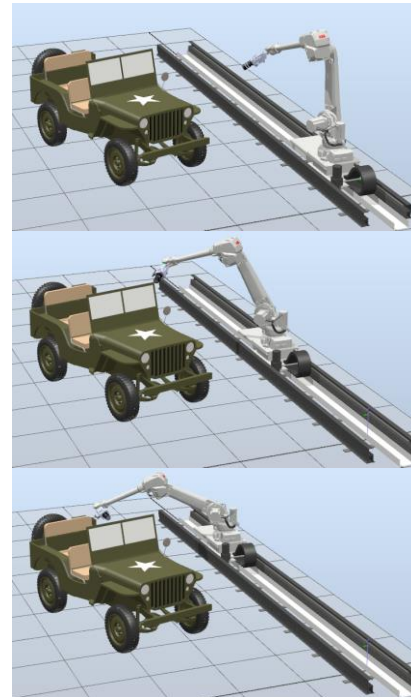


Figura 3. Simulación de la filmación con el brazo robótico en RobotStudio.

3. Resultados

3.1. Interfaz del programa desarrollado

A partir de los fundamentos teóricos expuestos en la sección anterior, se ha desarrollado un programa que se compone de cuatro pestañas sobre las cuales se pueden ir realizando diferentes etapas del proceso de cálculo. En la primera pestaña denominada *Tabla D-H* (véase la figura 4), se introduce la matriz asociada al brazo robótico definida mediante los parámetros de Denavit-Hartenberg del mismo. Una vez definida inicialmente, o cargada del archivo correspondiente con extensión *.txt* si se ha realizado un análisis anterior del mismo, se puede proceder a la obtención de un informe con las matrices de transformación homogéneas del mismo tanto individuales como la matriz global en variables simbólicas. Paralelamente, también es posible acudir a la siguiente pestaña titulada *PCD/PCI* para realizar el cálculo del problema cinemático deseado. En ella, introduciendo el valor de las coordenadas articulares deseadas, se puede obtener la posición en coordenadas cartesianas así como la orientación indicada en términos tanto en matriz de rotación como en cuaternios o ángulos de Euler, dando lugar al cálculo del problema cinemático directo. Para el caso opuesto, el problema cinemático inverso, se debe facilitar la pose del extremo del robot e , indicando una serie de ajustes relativos al método de resolución, se obtiene el valor de las coordenadas articulares asociadas a la pose de la cámara en cuestión. Para la indicación de dichos ajustes se desplegará una ventana en la cual se podrá indicar el método de resolución iterativo deseado (los cuales fueron abordados en el apartado 2.1.1) junto con

la selección de si se desea trabajar con restricciones si procede, los valores articulares iniciales del método y los valores articulares mínimos y máximos siempre y cuando el método lo contemple. Estos ajustes se podrán guardar para ser cargados en sucesivos análisis. Una vez realizado el análisis pertinente, es posible obtener una representación gráfica del robot en posición así como la elaboración de un informe específico del problema cinemático realizado. Esta pestaña se muestra en la figura 5.

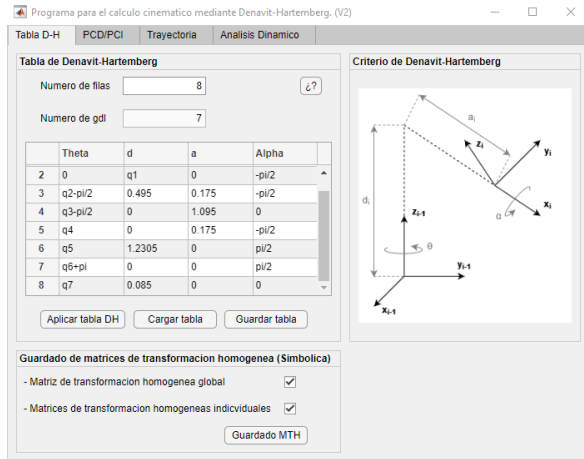


Figura 4 Interfaz inicial del programa desarrollado.

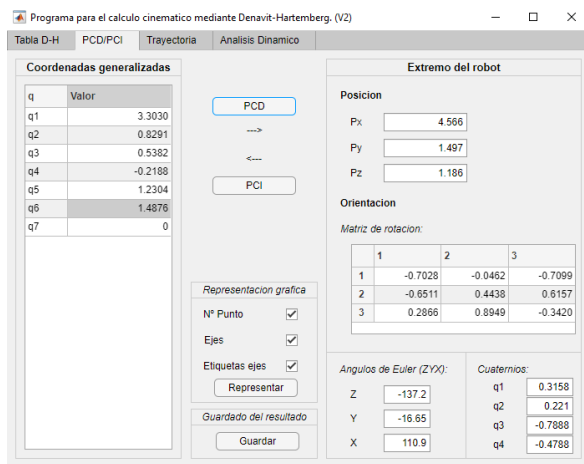


Figura 5 Interfaz para el cálculo cinemático.

La tercera pestaña (véase la figura 6) está orientada al cálculo de las trayectorias interpoladas a partir de la definición de una serie de poses de precisión, las cuales serán cargadas de un archivo Excel. Una vez indicado el número de poses intermedias y, configurado el método del PCI deseado al igual que en el caso anterior, se obtendrán las trayectorias interpoladas. Durante el proceso de cálculo es posible visualizar una representación animada del robot en cuestión siguiendo la trayectoria calculada así como gráficas relativas a la posición, velocidad y aceleración articulares del robot. Además, se ofrece la posibilidad de guardar un informe asociado que permite conocer todos los coeficientes de interpolación calculados, las

velocidades obtenidas mediante el método de Craig, los valores articulares interpolados y los valores de velocidad y aceleración interpolados para cada variable articular.

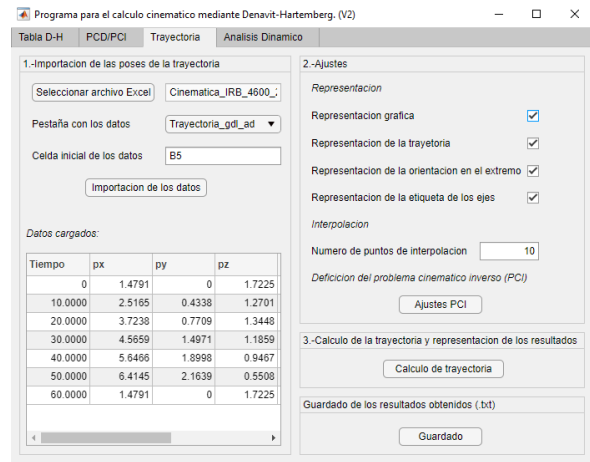


Figura 6 Interfaz para el cálculo de trayectorias interpoladas. Fuente: elaboración propia.

Finalmente, en la cuarta pestaña se permite la realización del cálculo dinámico inverso del robot en cuestión para la trayectoria calculada en la pestaña anterior. Para ello, se debe cargar los datos relativos a la masa, centro de masas y momentos de inercia para cada uno de los eslabones del brazo robótico de un archivo Excel. Posteriormente, una vez definido el valor de la fuerza gravitatoria correspondiente y el eje sobre el que actúa, se puede proceder al cálculo del mismo generando una gráfica que representa los esfuerzos de cada articulación respecto del tiempo. Adicionalmente, es posible guardar los datos obtenidos en un informe específico. La interfaz de esta pestaña se puede advertir en la figura 7.

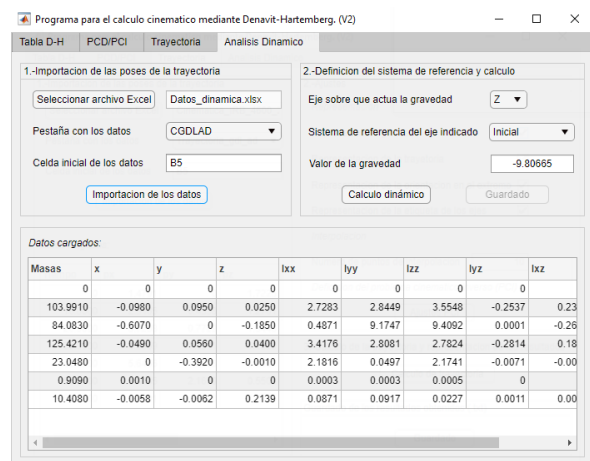


Figura 7 Interfaz para el cálculo dinámico.

Tabla 1 Ensayos cinemáticos realizados.

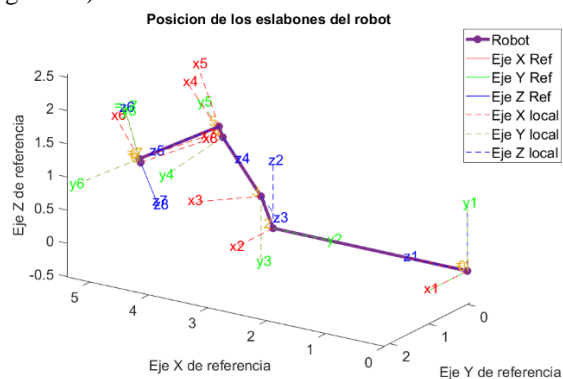
| Nº Prueba | Robot | Tipo de problema | Base | Solución válida |
|-----------|------------------|------------------|-------|-----------------|
| 1 | IRB 4600-20/2.50 | PCD | Fija | Sí |
| 2 | | | | Sí |
| 3 | | | | Sí |
| 4 | | | Móvil | Sí |
| 5 | | | | Sí |
| 6 | | PCI | Fija | Sí |
| 7 | | | | Sí |
| 8 | | | | Sí |
| 9 | | | Móvil | Sí |
| 10 | | | | Sí |

3.2. Resultados y verificación del programa desarrollado

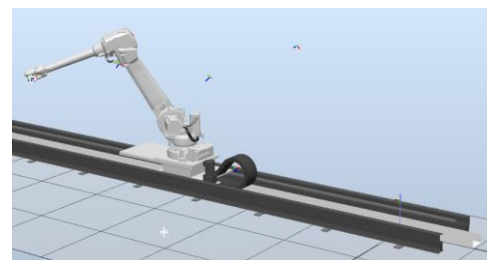
3.2.1. Verificación cinemática del programa

En este subapartado se validará el modelo de Denavit-Hartenberg del robot y la aplicación a nivel cinemático mediante la comparativa tanto numérica como visual de los resultados obtenidos con el programa desarrollado frente al software comercial del robot, *RobotStudio*. Para ello, se van a realizar un total de diez ensayos diferentes en el cual se van a estudiar, para el robot seleccionado, el problema cinemático directo e inverso considerando tanto la base móvil como la base fija, siendo esto último realizado mediante la aplicación de la restricción articular de la base a un movimiento nulo. Como se puede apreciar en la tabla 1, todos los ensayos realizados han obtenido soluciones válidas, permitiendo verificar la aplicación a nivel cinemático.

Un ejemplo visual de una de estas comparativas se puede observar en la figura 8. En ella se puede apreciar como la posición del robot calculada mediante el programa representada en la figura 8a, coincide con la representación realizada por RobotStudio para la misma pose del extremo del robot definida (véase figura 8b).



a) Posición obtenida con el programa desarrollado



b) Posición obtenida con RobotStudio

Figura 8 Ejemplo de representación visual de uno de los ensayos realizados.

3.2.2. Resultados cinemáticos de la trayectoria propuesta

Para el movimiento de rodaje propuesto en el apartado 2.3, se ha realizado el cálculo de la trayectoria interpolada mediante el programa elaborado, cuyo resultado se puede apreciar en la figura 9. Como se puede observar, el *software* ha sido capaz de obtener una transición suave de la cámara entre las diferentes poses definidas, indicadas con cuadros naranjas en la imagen. Si se compara dicha respuesta con la obtenida mediante RobotStudio, la cual se representa en la figura 10, se puede advertir que ambas trayectorias presentan un comportamiento muy similar, validando la respuesta del programa desarrollado. No obstante, en el caso de la obtenida con RobotStudio, esta es mucho más directa hacia los puntos de precisión haciendo que la imagen rodada sea ligeramente menos confortable visualmente.

A partir del cálculo de trayectorias realizado mediante el método expuesto en el apartado 2.1.2, se puede obtener la evolución de cada una de las coordenadas articulares interpoladas junto con las gráficas de velocidad y aceleración asociadas. Estas gráficas se pueden apreciar en las figuras 11, 12 y 13 respectivamente. Como se puede advertir, el empleo de polinomios cúbicos para la interpolación de trayectorias, tal y como se puede apreciar en la primera imagen, conlleva una evolución parabólica de la velocidad, lo que a su vez se traduce en un

comportamiento lineal de la aceleración que puede generar discontinuidades en esta variable. Esta fluctuación de la aceleración podría solventarse mediante el empleo de otro tipo de interpolador o mediante el uso de un polinomio de mayor orden, como es el caso del interpolador mediante splines quínticos.

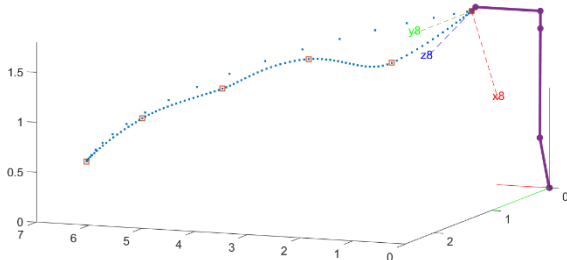


Figura 9 Trayectoria calculada por el programa para el plano de rodaje definido.

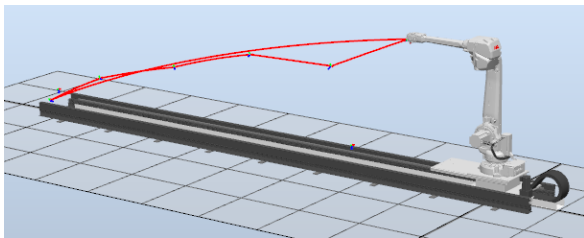


Figura 10 Trayectoria calculada por RobotStudio para el plano de rodaje definido.

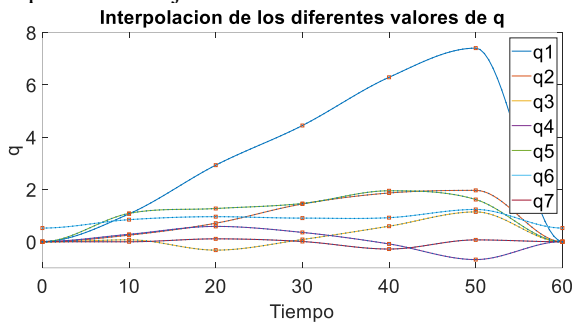


Figura 11 Evolución de las coordenadas articulares para la trayectoria interpolada.

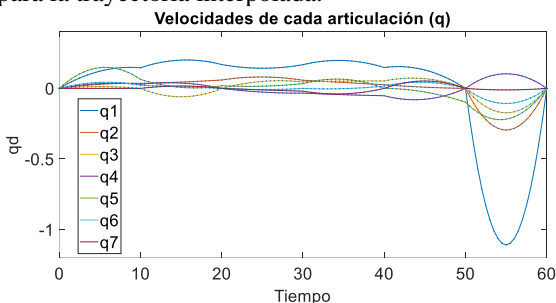


Figura 12 Velocidades asociadas a la trayectoria interpolada calculada.

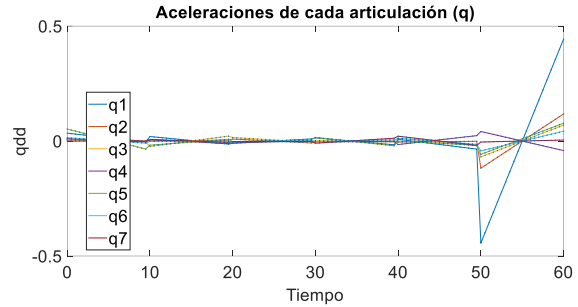


Figura 13 Aceleraciones asociadas a la trayectoria interpolada calculada.

3.2.3. Obtención y verificación de los resultados dinámicos

Finalmente, a fin de validar los resultados obtenidos mediante el empleo del programa para el cálculo dinámico, se ha decidido comparar los resultados de este con hasta otros cuatro métodos diferentes: a) Simulink empleando la librería de Simscape multibody (sm), b) Robotics Toolbox de Matlab (rt), c) el método de Euler programado *ad hoc* para este robot (bar) y, d) el método de Euler aplicado de manera pseudo-automática (Ali). El resultado de la comparativa se muestra en la figura 14. Como se puede apreciar, la respuesta general de todos los métodos para los diferentes valores de esfuerzos articulares es muy similar. En el caso de los valores correspondientes a la tercera, quinta, sexta y séptima articulación, se puede constatar que los resultados obtenidos son idénticos. Si por el contrario se analiza el resto de articulaciones rotacionales, articulaciones 2 y 4, se puede advertir como existen diferencias puntuales entre los resultados de Simulink y el resto de métodos de valor no superior a 10 kNm. Finalmente, en el caso de la articulación traslacional, la primera, se puede observar cómo al igual que en el caso anterior existen diferencias entre Simulink y el resto de métodos cuyo valor máximo se establece en 33 N. Cabe destacar que estas diferencias puntuales pueden ser debidas a la diferente manera de trabajar que presenta Simulink respecto al resto en lo relativo al cálculo del tensor de inercia a partir de los modelos geométricos. Sin embargo, teniendo en cuenta que las diferencias observadas son mínimas y se generan de forma puntual, se considera validada la función de cálculo dinámica desarrollada e integrada en el programa dentro de la interfaz mostrada en el apartado 3.1.

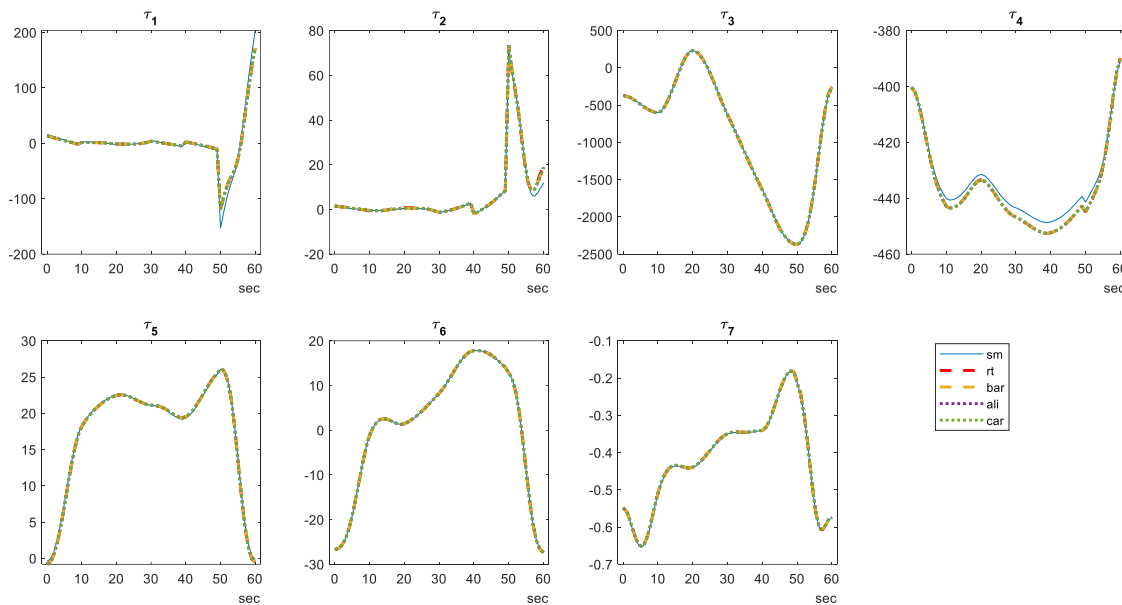


Figura 14 Comparativa dinámica para la trayectoria de rodaje indicada.

4. Conclusiones

En virtud de los resultados obtenidos en la realización de este proyecto se pueden extraer una serie de conclusiones.

En primer lugar, se destaca el desarrollo de una interfaz desarrollada en Matlab que permite de manera muy visual, confortable y rápida el cálculo de la cinemática, la trayectoria completa y la dinámica para cualquier robot definido siguiendo la trayectoria de rodaje deseada.

En segundo lugar, ha quedado demostrado la validez del programa frente a análisis cinemáticos y de interpolación de trayectorias mediante comparativa con el programa comercial de la compañía ABB *RobotStudio*.

En tercer lugar, tras la comparativa en la respuesta dinámica que presenta el brazo robotico, al llevar a cabo la trayectoria de rodaje indicada, entre las herramientas de Simulink, la Robotics Toolbox de Matlab y dos métodos basados en la formulación de Newton-Euler, se constata que la respuesta que ofrece el programa elaborado es correcta.

Como trabajo futuro se contempla la incorporación de un algoritmo de reconocimiento automático de distancia que permita ajustar automáticamente la distancia focal en función de las necesidades y los requerimientos del rodaje.

5. Agradecimientos

Este trabajo ha sido realizado parcialmente gracias al apoyo de la Universidad de Almería, de la Consejería de Transformación Económica, Industria,

Conocimiento y Universidades y del Fondo Europeo de Desarrollo Regional (FEDER) con los proyectos UAL2020-TEP-A1991, y PY20_00767.

6. Referencias

- [1] A. Alcántara, J. Capitan, R. Cunha, and A. Ollero, "Optimal trajectory planning for cinematography with multiple Unmanned Aerial Vehicles," *Rob. Auton. Syst.*, vol. 140, Jun. 2021, doi: 10.1016/j.robot.2021.103778.
- [2] O. Zachariadis, V. Mygdalis, I. Mademlis, N. Nikolaidis, and I. Pitas, "2D Visual tracking for sports UAV cinematography applications" in *2017 IEEE Global Conference on signal and information processing (GLOBALSIP 2017)*.
- [3] I. Karakostas, I. Mademlis, N. Nikolaidis, and I. Pitas, "UAV Cinematography constraints imposed by visual target tracking" in *2018 25th IEEE International Conference on Image Processing (ICIP)*.
- [4] L. Zhang and S. Yan, "Simulation and Control System of Camera Robot," in *Measuring Technology and Mechatronics Automation IV, pts 1 and 2*, vol. 128–129, no. 1–2.
- [5] D. Klimov and Y. Poduraev, "Design and Experimental Testing of a Robotic System for High Speed Recording," in *25th DAAAM International Symposium on Intelligent Manufacturing and Automation, 2014*.
- [6] J. He, "A Position-Level Global Optimization Inverse Kinematic Solution Algorithm for Dual Redundant Robots Based on Motion Characteristics," *Math. Probl. Eng.*, vol. 2018, 2018, doi: 10.1155/2018/9167837.
- [7] Z. Zhou, H. Guo, Y. Wang, Z. Zhu, J. Wu, and X. Liu, "Inverse kinematics solution for robotic manipulator based on extreme learning machine and sequential mutation genetic

- algorithm,” *Int. J. Adv. Robot. Syst.*, vol. 15, no. 4, 2018, doi: 10.1177/1729881418792992.
- [8] L. Zhang, J. He, and S. Wang, “Inverse Kinematic Solutions of Dual Redundant Camera Robot Based on Genetic Algorithm,” *Math. Probl. Eng.*, vol. 2017, 2017, doi: 10.1155/2017/7486178.
- [9] D. Nieuwenhuisen and M. H. Overmars, “Motion planning for camera movements,” in *2004 IEEE- International Conference on Robotics and Automation*.
- [10] Y.-H. Yu, C. Vo-Ky, S. Kodagoda, and Q. P. Ha, “FPGA-Based Relative Distance Estimation for Indoor Robot Control Using Monocular Digital Camera,” *J. Adv. Comput. Intell. Intell. Informatics*, vol. 14, no. 6, SI, pp. 714–721, Sep. 2010, doi: 10.20965/jaciii.2010.p0714.
- [11] K. Belghith, F. Kabanza, P. Bellefeuille, and L. Hartman, “Automated camera planning to film robot operations,” *Artif. Intell. Rev.*, vol. 37, no. 4, pp. 313–330, 2012, doi: 10.1007/s10462-011-9233-y.
- [12] A. Barrientos, L. F. Peñín, C. Balaguer, and R. Aracil, *Fundamentos de robótica*, 2 Edicion. Madrid: McGraw-Hill Interamericana de España S.L., 2007.
- [13] J. J. Craig, *Robótica*, Tercera ed. México D.F.: Pearson Educación, 2006.