

# Control basado en modelo del robot UR10e

José Luis Torres-Moreno<sup>1</sup>, Manuel Beschi<sup>2</sup>, Antonio Visioli<sup>2</sup>, Antonio Giménez<sup>1</sup>

<sup>1</sup>Automática Robótica y Mecatrónica, Departamento de Ingeniería, Universidad de Almería, España. Email: agimfer@ual.es

<sup>2</sup>Department of Mechanical and Industrial Engineering, University of Brescia. Email: antonio.visioli@unibs.it

## Resumen

En este trabajo se presenta un esquema de control del robot UR10e basado en un modelo dinámico que permite utilizar tanto los parámetros proporcionados por el fabricante, basados en la convención propuesta por Denavit y Hartenberg, como los parámetros encontrados en los principales repositorios sobre robótica, basados en el estándar URDF. Debido a la falta de compatibilidad entre ambos formatos, se propone un procedimiento sistemático de conversión que permita comparar los datos, detectándose posibles desviaciones. Una vez corregidas estas desviaciones con los datos del fabricante, se desarrolla un esquema de control que implementa el modelo validado. Se realizan varias simulaciones tanto con la implementación manual del esquema en MATLAB como por medio de un simulador de propósito específico y se analizan los resultados. En vista de estos se concluye que el esquema propuesto, basado en la formulación de Euler-Lagrange, es correcto en términos de precisión y eficiencia computacional.

**Palabras clave:** robótica; dinámica; control; cinemática.

## Abstract

This work presents a control scheme for the UR10e robot based on a dynamic model that allows using both the parameters provided by the manufacturer, based on the convention proposed by Denavit and Hartenberg, and the parameters found in the main robotics repositories, based in the URDF standard. Due to the lack of compatibility between both formats, a systematic conversion procedure is proposed that allows data to be compared, detecting possible deviations. Once these deviations are corrected with the manufacturer's data, a control scheme is developed that implements the validated model. Several simulations are carried out both with the manual implementation of the scheme in MATLAB and by means of a specific purpose simulator and the results are analyzed. In view of these, it is concluded that the proposed scheme, based on the Euler-Lagrange formulation, is correct in terms of precision and computational efficiency.

**Keywords:** robotics; dynamics; control; kinematics.

## 1. Introducción

El control basado en modelo, dentro del campo de la robótica, es una técnica esencial que permite a los robots interactuar de manera eficiente y precisa con su entorno [1]. Los robots colaborativos en general, y los desarrollados por Universal Robot, en particular, han demostrado ser herramientas muy útiles en una variedad de aplicaciones, desde el ensamblaje hasta la pintura, el paletizado, el atornillado, el embalaje o incluso el pulido [2].

El robot UR10e, en concreto, se caracteriza por ser el primer robot dentro de la gama del que, debido a su considerable tamaño, el fabricante proporciona los tensores de inercia de sus eslabones [3]. Estos parámetros, que son esenciales para el control preciso del robot cuando se emplean controladores personalizados, no se proporcionan para modelos más pequeños como el UR3 y el UR5 por considerarse que no son tan relevantes como para los robots de tamaño superior [3], [4], [5]. Es por ello por lo que algunos trabajos se centran en la obtención precisa de estos parámetros [6], [7]. De hecho, en la literatura se pueden

encontrar numerosas técnicas de identificación de parámetros con el objetivo de obtener precisos modelos dinámicos de robot manipuladores [8]. A pesar de su importancia, existe una discrepancia notable entre los parámetros inerciales publicados por Universal Robot, que se basan en los parámetros de Denavit-Hartenberg (DH), y los que se encuentran en los formatos de Descripción Universal de Robótica (URDF) utilizados por varios simuladores.

Los simuladores de robótica, como Gazebo [9], Drake [10] y MuJoCo [11], son herramientas de código abierto ampliamente utilizadas en el diseño, control y simulación de robots. Estos simuladores utilizan el formato URDF o derivados de este, que es más complejo que la convención estándar de DH utilizada por Universal Robots. Esta discrepancia puede llevar a diferencias en los resultados de la simulación, lo que puede afectar la precisión del control del robot.

Para abordar este problema, este trabajo presenta un procedimiento sistemático para convertir los modelos basados en DH a URDF. Este procedimiento permite comparar directamente los modelos de robots basados en los parámetros proporcionados por el fabricante y los encontrados en los principales repositorios de robótica. A través de este análisis, se pone de manifiesto que los datos proporcionados por el fabricante y los encontrados en los repositorios no concuerdan.

Además, en este trabajo se presenta un esquema de control que implementa el modelo dinámico del robot UR10e, denominado control dinámico inverso [12]. Con esta arquitectura de control, se llevan a cabo varias simulaciones utilizando tanto la implementación manual del esquema en MATLAB como un simulador de propósito específico como Drake, basado en la formulación dinámica propuesta en [13]. Los resultados de estas simulaciones demuestran que el esquema propuesto es correcto en términos de precisión.

En resumen, este trabajo destaca la importancia de la precisión en la modelización de robots para el control eficaz, y propone una metodología para poder intercambiar los parámetros inerciales entre diferentes formatos. A través de este enfoque, se espera mejorar la precisión y la eficiencia del control de robots en una variedad de aplicaciones. El resto del artículo se estructura como sigue: En la siguiente sección se presenta la metodología llevada a cabo para poder elaborar el modelo dinámico del robot empleando los diferentes tipos de datos comentados. A continuación, se presenta el esquema de control propuesto, que es validado en la sección de resultados para dos tipos distintos de ensayos. Finalmente, se establecen las conclusiones y trabajos futuros.

## 2. Modelado

En primer lugar, desarrolla el modelo dinámico basado en los datos proporcionados por el fabricante, y que se resumen en la Tabla 1 para los tres primeros eslabones, y en la Tabla 2 para los eslabones de la muñeca. Estos datos están expresados de acuerdo con la convención estándar de Denavit y Hartenberg (DH).

Tabla 1 Parámetros característicos de los tres primeros eslabones del robot UR10e. Unidades en S.I.

	L1	L2	L3
a	0	-0,6127	-0.57155
$\alpha$	$\pi/2$	0	0
d	0.1807	0.0000	0.0000
$\theta$	0	$-\pi/2$	0
m	7.369	13.051	3.989
l <sub>cx</sub>	0.0210	0.380	0.240
l <sub>cy</sub>	0.000	0.000	0.000
l <sub>cz</sub>	0.027	0.158	0.068
I <sub>xx</sub>	0.0341	0.0281	0.0101
I <sub>xy</sub>	0.0000	0.0001	0.0000
I <sub>xz</sub>	-0.0043	-0.0156	0.0092
I <sub>yy</sub>	0.0353	0.7707	0.3093
I <sub>yz</sub>	0.0001	0.0000	0.0000
I <sub>zz</sub>	0.0216	0.7694	0.3065

Tabla 2 Parámetros característicos de los eslabones de la muñeca del robot UR10e. Unidades en S.I.

	L4	L5	L6
a	0.0000	0.0000	0.0000
$\alpha$	$\pi/2$	$-\pi/2$	0
d	0.17415	0.11985	0.11655
$\theta$	$-\pi/2$	0	0
m	2.100	1.980	0.615
l <sub>cx</sub>	0.000	0.000	0.000
l <sub>cy</sub>	0.007	0.007	0.000
l <sub>cz</sub>	0.018	0.018	-0.026
I <sub>xx</sub>	0.0030	0.0030	0.0004
I <sub>xy</sub>	0.0000	0.0000	0.0000
I <sub>xz</sub>	0.0000	0.0000	0.0000
I <sub>yy</sub>	0.0022	0.0022	0.0004
I <sub>yz</sub>	-0.0002	-0.0002	0.0000
I <sub>zz</sub>	0.0026	0.0026	0.0003

Los valores del ángulo  $\theta$  de las articulaciones 2 y 4 se corresponden con la posición *home*, equivalentes a una extensión vertical del brazo, tal y como se muestra en la Figura 1. Los sistemas de coordenadas relacionados con los parámetros publicados por el fabricante del robot se corresponden con los denotados por minúscula en la Figura 1. Así, las coordenadas del centro de gravedad de cada eslabón están referidas su correspondiente sistema de referencia. Por su parte, los tensores de inercia están referidos al centro de masas de cada eslabón, con ejes paralelos a estos sistemas de coordenadas.

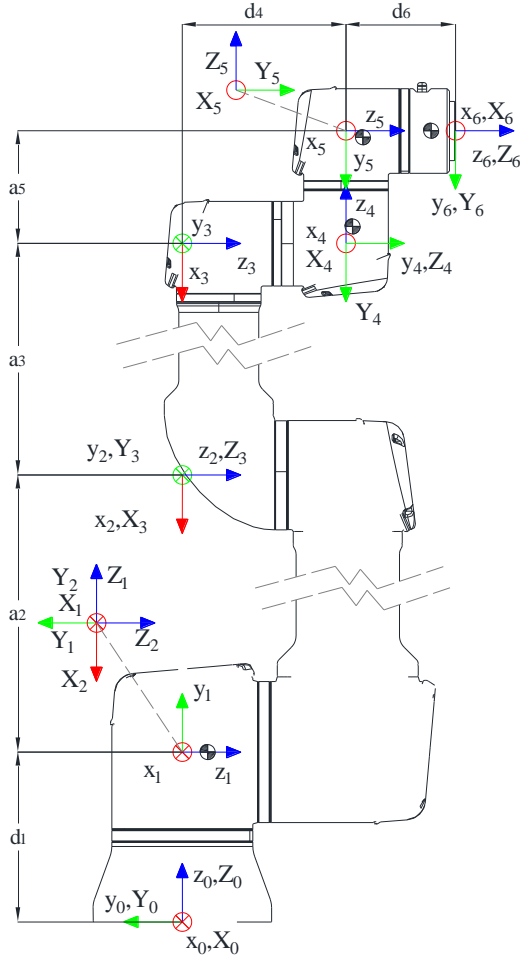


Figura 1. Sistemas de coordenadas: en minúscula ejes según convención DH; en mayúscula, ejes según formato URDF. Fuente: elaboración propia.

## 2.1. Modelo desarrollado

Los sistemas de coordenadas utilizados se corresponden con los mostrados en la Figura 1, estando los ejes de cada articulación alineados con el eje z. Debido al esquema de control empleado, y que será presentado en la siguiente sección, es necesario obtener una expresión del modelo dinámico en forma matricial:

$$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \boldsymbol{\tau}_g(\mathbf{q}), \quad (1)$$

donde  $\boldsymbol{\tau}$ , representa el vector de fuerzas generalizadas, es decir, los pares aplicados en las articulaciones,  $\mathbf{M}(\mathbf{q})$  es la matriz de masas dependiente de la configuración  $\mathbf{q}$ ,  $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$  es una matriz empleada para calcular las fuerzas de inercia y de Coriolis, y  $\boldsymbol{\tau}_g(\mathbf{q})$ , representa las fuerzas gravitatorias. La contribución de la energía cinética al Lagrangiano viene dada por:

$$\frac{d}{dt} \frac{\partial T}{\partial \dot{\mathbf{q}}} - \frac{\partial T}{\partial \mathbf{q}} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}. \quad (2)$$

De esta forma, se obtiene la matriz de masa a partir de la expresión de dicha energía teniendo en cuenta la

velocidad lineal del centro de gravedad de cada eslabón, así como las velocidades angulares a las que están sometidos, llegando a la expresión:

$$\mathbf{T} = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{M}(\mathbf{q}) \dot{\mathbf{q}}. \quad (3)$$

Una vez que se obtiene la matriz de masas, es posible calcular la matriz de fuerzas centrífugas y de Coriolis como:

$$[\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})]_i = \dot{\mathbf{q}}^T \bar{\mathbf{C}}_i \dot{\mathbf{q}} \quad (4)$$

Con:

$$\bar{\mathbf{C}}_{i,j,k}(\mathbf{q}) = \frac{1}{2} \left[ \frac{\partial \mathbf{M}_{ij}}{\partial \mathbf{q}_k} + \frac{\partial \mathbf{M}_{ik}}{\partial \mathbf{q}_j} - \frac{\partial \mathbf{M}_{jk}}{\partial \mathbf{q}_i} \right]$$

Finalmente, los pares asociados a la acción gravitatoria se obtienen a partir de la energía potencial, calculada a partir de la altura de los centros de masas de cada uno de los eslabones.

## 2.2. Verificación del modelo

Con el objetivo de verificar el modelo se desarrolla un entorno de simulación en el que se implementan las ecuaciones obtenidas en el siguiente subapartado y se lleva a cabo una simulación dinámica inversa para una trayectoria aleatoria definida por la siguiente expresión:

$$q_i(t) = A_i \cdot \sin(2\pi \cdot f_i \cdot t) + \theta_0$$

Donde  $A_i$ , representa la amplitud del recorrido articular,  $f_i$  su amplitud y  $\theta_0$  el ángulo necesario para partir de la posición home. Los valores correspondientes a cada una de las articulaciones se muestran en la Tabla 3.

Tabla 3 Parámetros que definen el movimiento articular empleado en el ensayo de dinámica inversa. Unidades en S.I.

i	$A_i$	$f_i$	$\theta_{0i}$
1	$120\pi/180$	0.1	0
2	$45\pi/180$	0.1	$-\pi/2$
3	$30\pi/180$	0.5	0
4	$120\pi/180$	0.1	$-\pi/2$
5	$45\pi/180$	0.1	0
6	$45\pi/180$	0.1	0

Además, se incluye en dicho entorno de simulación un modelo basado en los datos publicados en los principales repositorios de robótica, en formato URDF. En la Figura 2 se muestran los resultados obtenidos.

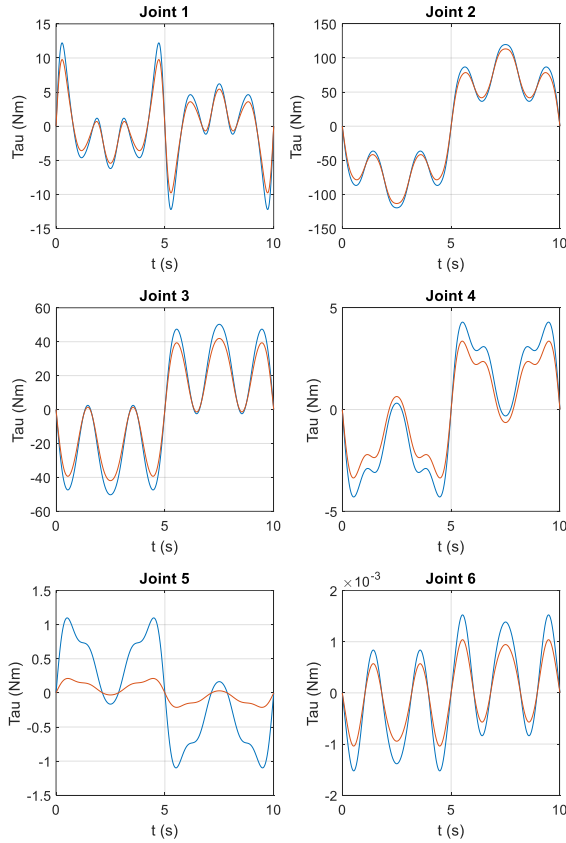


Figura 2. Pares articulares obtenidos en la simulación dinámica inversa para los dos modelos comparados. En azul, modelo basado en parámetros proporcionados por el fabricante. En rojo, modelo descargado del repositorio en URDF.

Como se puede observar, existen discrepancias entre los pares obtenidos mediante el modelo desarrollado a partir de los parámetros DH y el modelo publicado en el repositorio. Para encontrar las causas de esta diferencia es necesario analizar en profundidad cómo está confeccionado el modelo URDF. Tras una primera lectura, se identifica que las masas de los eslabones son ligeramente distintas. De hecho, se puede apreciar como en el modelo basado en parámetros DH, con valores de masa superiores, existen mayores requerimientos de par. Sin embargo, pese a corregir este valor de acuerdo con las especificaciones del fabricante, los resultados continúan siendo diferentes. Por tanto, es necesario comparar los valores correspondientes a la localización del centro de gravedad y el tensor de inercia. No obstante, esta tarea no es trivial ya que los sistemas de coordenadas empleados son diferentes, tal y como se muestra en la Figura 1.

Para poder comparar estos resultados de una manera eficiente y sistemática, se emplean una serie de transformaciones que se detallan a continuación.

Sea  $[I^{B/B_{cm}}]_{xyz}$  el tensor de inercia del eslabón  $B$  tomado sobre su centro de gravedad, denominado  $B_{cm}$  y expresado en el sistema correspondiente a la convención DH, denotado por  $xyz$ . Este tensor puede ser expresado respecto del sistema correspondiente a la convención URDF, denotado por  $XYZ$ :

$$[I^{B/B_{cm}}]_{XYZ} = {}^{XYZ}R^{xyz} [I^{B/B_{cm}}]_{xyz} ({}^{XYZ}R^{xyz})^T, \quad (5)$$

teniendo en cuenta  ${}^{XYZ}R^{xyz}$  que representa la matriz de rotación del sistema  $xyz$  respecto del sistema  $XYZ$ .

Por su parte, sea  $[{}^{xyz}p^{B_{cm}}]_{xyz}$  el vector de posición del centro de gravedad del eslabón  $B$ , respecto del sistema  $xyz$ , cuyas componentes se expresan también en  $xyz$ , podemos expresar este vector respecto de  $XYZ$ :

$$[{}^{XYZ}p^{B_{cm}}]_{XYZ} = {}^{XYZ}X^{xyz} [{}^{xyz}p^{B_{cm}}]_{xyz}, \quad (6)$$

teniendo en cuenta  ${}^{XYZ}X^{xyz}$  que representa la matriz de transformación homogénea del sistema  $xyz$  respecto del sistema  $XYZ$ .

Por tanto, para obtener el valor de los parámetros inerciales de cada eslabón a partir de los datos de la Tabla 1 basta con utilizar estas expresiones una vez que se disponga de las matrices  ${}^{XYZ}R^{xyz}$  y  ${}^{XYZ}X^{xyz}$  que afectan a cada eslabón. Estas transformaciones se describen en la Tabla 4.

Tabla 4 Parámetros que definen el movimiento articular empleado en el ensayo de dinámica inversa. Unidades en S.I.

i	Rotación	Traslación
1	$Rot_x(\pi/2)$	—
2	—	$[-a_2, 0, 0]^T$
3	—	$[-a_3, 0, 0]^T$
4	$Rot_x(\pi/2)$	—
5	$Rot_x(-\pi/2)$	—
6	—	—

Una vez referenciados los datos inerciales a los nuevos sistemas de coordenadas se identifica que estos valores efectivamente tampoco concuerdan. Por tanto, se decide actualizar el modelo URDF con los valores publicados por el fabricante. De esta forma, se podrán validar los resultados del controlador propuesto con respecto a los resultados obtenidos por un programa de propósito específico que implementa en formato URDF. Una vez que ambos modelos incorporan los mismos datos de partida se verifica que efectivamente los resultados obtenidos en las simulaciones de dinámica inversa coinciden, de lo que se concluye que el modelo tanto el dinámico desarrollado como el

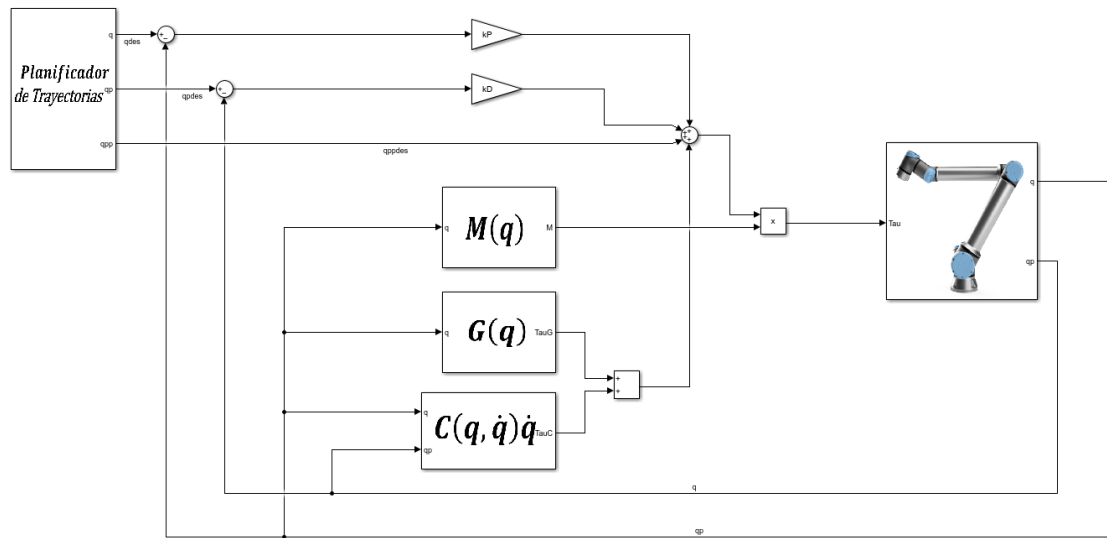


Figura 3. Esquema de control. Fuente: elaboración propia.

procedimiento de conversión de los dos métodos de parametrización cinemática son correctos. En la siguiente sección se explica cómo se integran estos modelos en el esquema de control propuesto.

### 3. Control

El esquema de control propuesto se basa en la utilización del modelo dinámico desarrollado en la sección anterior dentro de una estructura desacoplada, tal y como se muestra en la Figura 3. Su funcionamiento es el siguiente: a partir de los errores de posición y velocidad articulares se calcula una acción de control junto con la aceleración deseada por medio de un controlador PID. Debido a la inercia de los eslabones, esta aceleración conlleva unos esfuerzos, que se pueden calcular como el producto de dicha aceleración por la matriz de masas, resultando los pares actuantes necesarios para alcanzar dicha aceleración. En este punto, se realiza una prealimentación de los esfuerzos debidos a la acción gravitatoria y a las fuerzas centrífugas y de Coriolis. La acción de control es la resultante de sumar todos estos esfuerzos.

### 4. Resultados

El esquema de control desarrollado en la sección anterior se prueba mediante ensayos en simulación. Además, se compara con los resultados obtenidos por un software de propósito específico como Drake [10], y por un controlador basado en control PID articular. Para el caso del control basado en modelo, se utilizan unas constantes comunes para todas las articulaciones, con valores  $k_P=40$  y  $k_D=10$ . Por su parte, para el esquema PID articular se utilizan unos valores, también comunes para todas las articulaciones, de  $k_P=800$ ,  $k_I=1000$  y  $k_D=300$ . Además de los parámetros

inerciales mostrados en la Tabla 1, se ha considerado una fricción por rozamiento viscoso en las articulaciones de valor  $0.5 \text{ Nm/(rad/s)}$ . El tiempo de muestreo utilizado es de  $0.1$  milisegundos. Los ensayos consisten en una trayectoria punto a punto partiendo de la posición inicial  $q=[0, -1.57, 0, -1.57, 0, 0]^T$  y posición final  $q=[1.5, -1.2, 0.8, -1.2, -1.5, 1]^T$ , tal y como se muestra en la Figura 4.

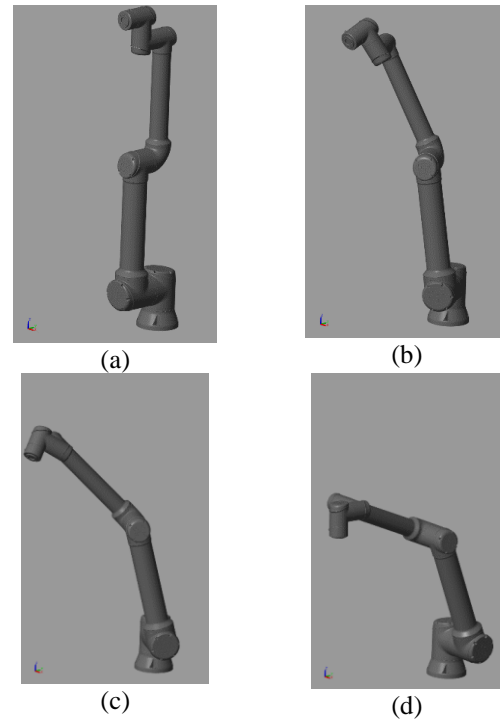


Figura 4. Trayectoria punto a punto: (a) Posición inicial del ensayo, (b,c) Posiciones intermedias, (d) Posición final.

Los resultados obtenidos se muestran en las Figuras 5 y 6. En primer lugar, se observa que los resultados obtenidos mediante el controlador propuesto coinciden exactamente con los obtenidos mediante Drake, lo que sirve como validación tanto del modelo dinámico como del lazo de control. Además, se observa una rápida convergencia hacia las posiciones demandadas. A continuación, si comparamos estos resultados con los obtenidos mediante el controlador basado en PID, se aprecia que cuando no se tiene en cuenta el modelo dinámico se obtiene una respuesta drásticamente peor, ya que se alcanza la referencia a 2 segundos más tarde. Además, se aprecia un comportamiento subamortiguado que puede ser inadmisibles para la mayoría de las aplicaciones. En todos los casos, los pares actuantes son inferiores a los pares máximos especificados por el fabricante. Respecto al tiempo computacional, la simulación, de 5 segundos de duración, se lleva a cabo en 730.50 segundos con un portátil con Windows 11 que cuenta con 16 gb de RAM y procesador 11th Gen Intel(R) Core(TM) i7-1165G7 a 2.80 GHz, cuando se utiliza la *Robotic Systems Toolbox* para el cálculo del modelo dinámico. Obviamente, resulta un tiempo inviable para una aplicación en tiempo real. Sin embargo, cuando se usa una implementación manual de la formulación de Euler-Lagrange, el tiempo de cómputo se reduce a 4.82 s. Por su parte, el uso de Drake, con el mismo computador, pero con sistema operativo Ubuntu 22.04, se obtienen unos tiempos de cómputo por debajo del tiempo real.

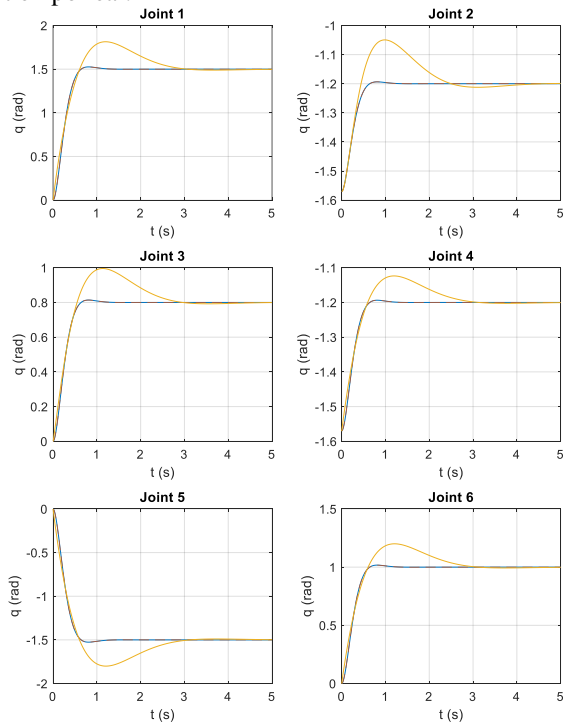


Figura 5. Evolución de la posición en la trayectoria punto a punto. En todas las gráficas, azul: controlador propuesto; rojo: Drake; amarillo: PID.

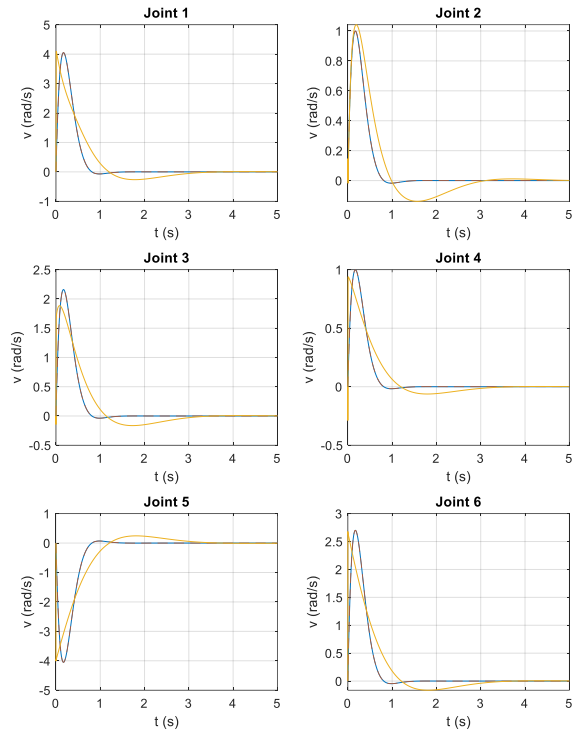


Figura 6. Evolución de la velocidad en la trayectoria punto a punto. En todas las gráficas, azul: controlador propuesto; rojo: Drake; amarillo: PID.

Además de la trayectoria punto a punto, el algoritmo de control propuesto fue puesto a prueba en una maniobra de *pick and place*. En la Figura 7 se muestra la configuración de una célula de trabajo colaborativo que incorpora un robot UR10e. En esta célula, el robot se encarga de detectar, por medio de una cámara de profundidad, la posición y orientación de las piezas situadas en la bandeja, para agarrarlas y colocarlas de forma ordenada en la zona del lateral derecho de la célula.

Con objeto de validar el controlador propuesto, se llevará a cabo una simulación correspondiente a las trayectorias que se deben llevar a cabo para un ciclo de trabajo correspondiente a la primera pieza. Los resultados correspondientes a la evolución de las coordenadas articulares junto con las consignas impuestas por el planificador de trayectorias se muestran en la Figura 8. Como se puede apreciar, el seguimiento es exacto. Por su parte, la evolución de los pares articulares se muestra en la Figura 9. Se puede observar que los valores de par se ubican dentro de los límites admisibles, siendo los valores más altos los de las articulaciones 2 y 3, como era de esperar. Se debe tener en cuenta que estos ensayos requieren de la adaptación del modelo dinámico presentado a la configuración boca abajo del robot, así como la incorporación de los parámetros inerciales de la pinza, para un correcto desempeño del controlador.





Figura 7. Ejemplo de aplicación del uso del robot UR10e para tareas de *pick and place* en una célula de trabajo colaborativo.

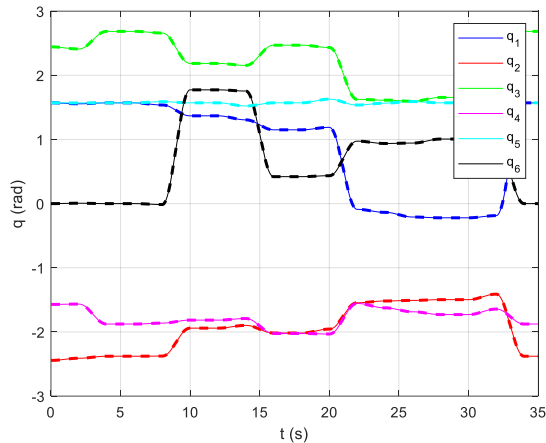


Figura 8. Evolución de las posiciones articulares en la trayectoria *pick and place*. En todas las curvas, el trazo continuo representa el valor medido y el trazo discontinuo la referencia.

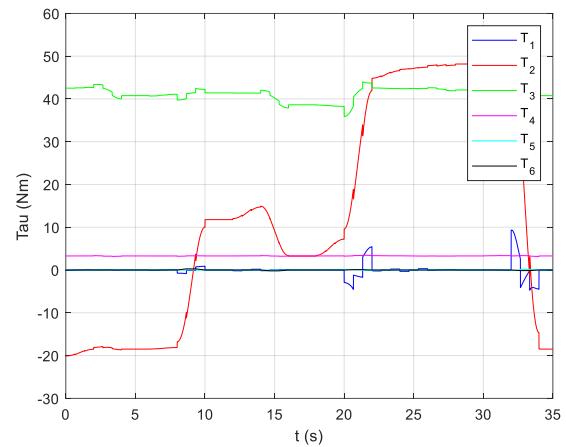


Figura 9. Evolución de los pares articulares en la trayectoria *pick and place*. En todas las curvas, el trazo continuo representa el valor medido y el trazo discontinuo la referencia.

## 5. Conclusiones

En este trabajo se ha desarrollado un controlador basado en modelo para el robot colaborativo UR10e a partir de los parámetros geométricos e inerciales proporcionados por el fabricante. Además, se ha propuesto una metodología que permite comparar estos datos con los datos empleados típicamente en el campo de la robótica, que suelen estar referidos a la convención URDF. Esta metodología fue validada mediante la realización de una simulación dinámica inversa, que puso de manifiesto las discrepancias entre ambos modelos. A continuación, una vez que se llevan a cabo las correcciones necesarias, se utilizaron estos modelos para validar el controlador propuesto. Para ello, se realizaron dos ensayos: uno de trayectoria punto a punto y otro de trayectoria *pick and place*. En vista de los resultados obtenidos, se puede concluir que tanto la metodología de modelado como el controlador propuesto son adecuados en términos de precisión y eficiencia computacional. Además, esta metodología puede ser extrapolada a cualquier otro robot manipulador. Como trabajos futuros se proponen la realización de un estudio comparativo de diferentes tipos de controladores que implementados en el robot real.

## Agradecimientos

El primer autor agradece la financiación recibida por parte del Ministerio de Universidades del Gobierno de España para estancias de movilidad de profesores e investigadores en centros extranjeros de enseñanza superior e investigación. Este trabajo también es resultado del Proyecto PID2021-122560OB-I00

financiado por MCIN/ AEI /10.13039/501100011033/ y por FEDER Una manera de hacer Europa.

.

[12] B. Siciliano, S. Lorenzo, L. Villani, y G. Orilo, *Robotics: Modelling, Planning and Control*. 2010.

[13] A. Jain, *Robot and Multibody Dynamics: Analysis and Algorithms*. 2012. doi: 10.1007/978-1-4419-7267-5.

## Referencias

- [1] P. Zhang Bowei and Liu, “Model-Based and Model-Free Robot Control: A Review”, en *RiTA 2020*, A. and L. P. and P. J. and M. H. and K. J. and K. J.-H. Chew Esyin and P. P. Abdul Majeed, Ed., Singapore: Springer Singapore, 2021, pp. 45-55.
- [2] Universal Robot, “Collaborative Robots Applications”. Fecha de consulta: 16/05/2024
- [3] Universal Robot, “DH parameters for calculations of kinematics and dynamics”. Fecha de consulta: 16/05/2024
- [4] C. Copot, C. Muresan, C. M. Ionescu, S. Vanlanduit, y R. De Keyser, “Calibration of UR10 robot controller through simple auto-tuning approach”, *Robotics*, vol. 7, n.o 3, jul. 2018, doi: 10.3390/robotics7030035.
- [5] T. Xu et al., “A novel collision detection method based on current residuals for robots without joint torque sensors: A case study on UR10 robot”, *Robot Comput Integr Manuf*, vol. 89, oct. 2024, doi: 10.1016/j.rcim.2024.102777.
- [6] T. Xu, J. Fan, Q. Fang, Y. Zhu, y J. Zhao, “Robot dynamic calibration on current level: modeling, identification and applications”, *Nonlinear Dyn*, vol. 109, n.o 4, pp. 2595-2613, sep. 2022, doi: 10.1007/s11071-022-07579-0.
- [7] A. Raviola y A. De-Martin, “Identification of a UR5 Collaborative Robot Dynamic Parameters”, en *Advances in Service and Industrial Robotics*, M. A. and S. J. Zeghloul Saïd and Laribi, Ed., Cham: Springer International Publishing, 2021, pp. 69-77.
- [8] Q. Leboutet, J. Roux, A. Janot, J. R. Guadarrama-Olvera, y G. Cheng, “Inertial Parameter Identification in Robotics: A Survey”, *Applied Sciences*, vol. 11, n.o 9, 2021, doi: 10.3390/app11094303.
- [9] N. Koenig y A. Howard, “Design and use paradigms for Gazebo, an open-source multi-robot simulator”, en 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566), 2004, pp. 2149-2154 vol.3. doi: 10.1109/IROS.2004.1389727.
- [10] R. Tedrake y the Drake Development Team, “Drake: Model-based design and verification for robotics”. 2019. [En línea]. Disponible en: <https://drake.mit.edu>
- [11] E. Todorov, T. Erez, y Y. Tassa, “MuJoCo: A physics engine for model-based control”, en 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2012, pp. 5026-5033. doi: 10.1109/IROS.2012.6386109.