

Control de un robot Niryo desde Matlab y Simulink

Beneyto-Rodríguez, A.^{a,*}, Herreros-López, A.^a

^a *Departamento de Ingeniería de Sistemas y Automática, Universidad de Valladolid, C/ Dr Mergelina s/n, 47011 Valladolid, España*

To cite this article: Beneyto, A., Herreros, A. 2023. Control of a Niryo robot from Matlab and Simulink XLIV Jornadas de Automática, 523-527. <https://doi.org/10.17979/spudc.9788497498609.523>

Resumen

Este proyecto tiene como finalidad realizar la comunicación con el robot colaborativo Niryo Ned, a través de la herramienta de software matemático Matlab. Para conseguir este objetivo, se estudian y se describen las distintas aplicaciones que hacen posible la comunicación entre Matlab y el ROS (Robot Operating System) del robot.

Además de comentar todas las especificaciones técnicas del robot, se han creado distintas estaciones de trabajo con el robot viendo sus diferencias entre un modelo dinámico o uno cinemático.

Por último, se ha diseñado una aplicación en App Designer capaz de comunicarse con el robot Niryo Ned mediante la conexión de tipo TCP/IP. Este diseño tiene un fin educativo para que los alumnos demuestren sus conocimientos de robótica mediante el uso de este robot.

Palabras clave: Robot manipulador, Tecnología robótica, Planificación de trayectorias.

Control of a Niryo robot from Matlab and Simulink

Abstract

The project consists of communicating with the collaborative robot, Niryo Ned, using the mathematical software tool Matlab. To achieve this objective, the different applications that make communication between Matlab and the robot's ROS (Robot Operating System) possible are studied and described.

In addition to discussing all the technical specifications of the robot, different workstations have been created with the robot, showing the differences between a dynamic and a kinematic model.

Finally, an application has been designed in App Designer capable of communicating with the Niryo Ned robot via TCP/IP connection. This design has an educational purpose so that students can demonstrate their knowledge of robotics by using this robot.

Keywords: Robots manipulators, Trajectory planning, Robotics technology.

1. Introducción

Desde hace varios años, está surgiendo una nueva generación de robots, denominados robots colaborativos, capaces de trabajar de forma coordinada con los humanos. Llevan incorporados sensores de fuerza para garantizar la seguridad a su alrededor. Los robots colaborativos se caracterizan por ser ligeros, flexibles y fáciles de instalar, han

sido diseñados para interactuar con los humanos compartiendo el mismo espacio de trabajo. En cuanto al ámbito educativo, varias compañías han desarrollado diversos robots colaborativos con el fin de aprender robótica directamente con un robot similar a los industriales. De esta forma, permite al alumno crear sus propios programas y a la vez prototipar escenarios industriales donde se incorpore este robot.

Partiendo del robot colaborativo que ha integrado la escuela de Ingenierías Industriales de la Universidad de Valladolid, denominado Niryo Ned, se ha querido demostrar el uso didáctico de este robot, con el fin de utilizarse en diversas clases con los alumnos y probar su posible manipulación desde distintos programas a los habitualmente usados en el control de Niryo ned, como es Niryo Studio. La manipulación se realiza desde Matlab y para ello es necesario conocer todas las comunicaciones entre esta aplicación y el ROS del robot.

El desarrollo del proyecto tiene principalmente un fin didáctico en el área de la robótica y las distintas comunicaciones, además de los diversos ensayos realizados con este robot para su futuro uso en las aulas. Se ha incorporado una librería denominada SimRob, basada en Matlab y Simulink, para poder realizar las clases y se ha querido incluir el robot Niryo Ned en esta librería.

2. Desarrollo

Niryo Ned es un brazo robótico colaborativo de seis ejes diseñado para la educación y la investigación que está basado en Ubuntu 18.04 y en ROS Melodic. Este sistema operativo permite que el robot cuente con múltiples librerías para concebir muchos programas, desde los más sencillos a los más complejos, respondiendo a sus necesidades.

Debido a la complejidad de las comunicaciones entre el usuario y el ROS de Niryo Ned, se ha decidido crear una interfaz en Matlab para facilitar al usuario el control del robot. Para ello se establecerá una conexión entre Matlab y Python mediante varios archivos que permiten crear un enlace entre ambas aplicaciones. El diagrama de comunicaciones entre aplicaciones que se obtiene es el que aparece en la Figura 1.

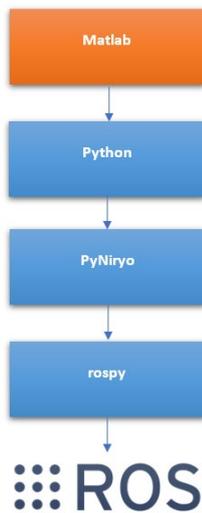


Figura 1: Comunicación entre Matlab y ROS

2.1. SimRob de Niryo NED

A la hora de crear las estaciones de trabajo en Simulink en las cuales aparece un robot, entre otros componentes, se debe considerar que los brazos y las articulaciones del robot quedan

definidos en un formato universal denominado URDF (Universal Robot Defined Format), que usa ficheros xml. Simulink (Simscape) es capaz de leer los ficheros con este formato y traducirlo a un icono de este entorno.

El archivo URDF que representa al robot, es distinto para un modelo cinemático que para uno dinámico, ya que cada uno tiene diferentes características.

Se han empleado distintas librerías de Matlab y Simulink para el desarrollo del proyecto: Robotic System Toolbox, Simscape Multibody, SimRob y el objeto Kin (creado por Alberto Herreros López). En la Figura 2 se muestra un esquema aclaratorio de estas librerías.

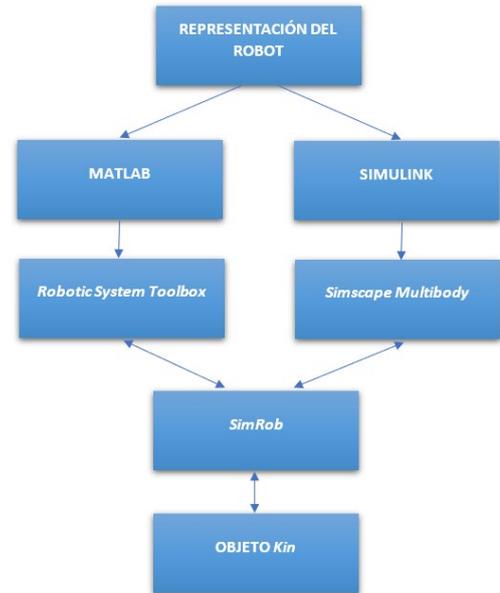


Figura 2: Conexión entre las librerías de Matlab y Simulink

2.1.1. Estación con un modelo cinemático

El icono del robot presenta dos señales de entrada, una de Simscape para asociarlo al sistema mecánico y otra de Simulink que introduce los ángulos de las articulaciones del robot. Además, tiene una variable de salida de Simscape para asociarlo a otros componentes mecánicos. La estación creada para el modelo cinemático aparece en la Figura 3.

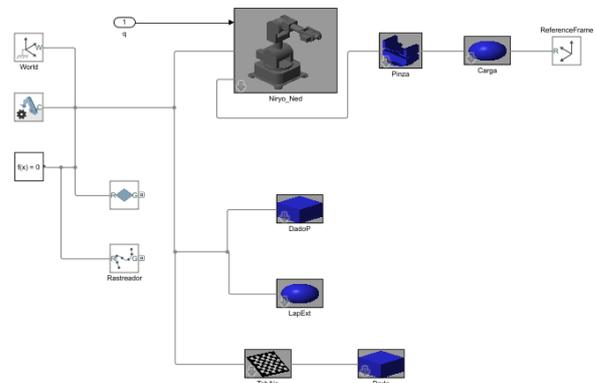


Figura 3: Estación para el modelo cinemático

En este modelo, las entradas en las articulaciones son los ángulos de las respectivas articulaciones, sin embargo, no existen salidas. Este esquema URDF se muestra en la Figura 4.

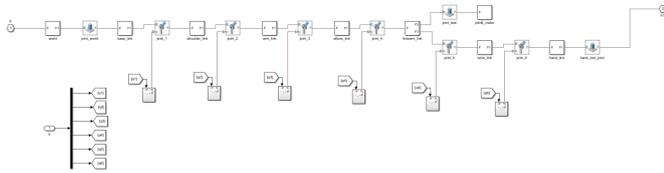


Figura 4: URDF del robot del modelo cinemático

2.1.2. Estación con un modelo dinámico con un lazo

Para el diseño de un modelo dinámico, como el de la Figura 5, es necesario realizar un sistema en lazo cerrado para conseguir mover al robot a la posición deseada, es decir, que la señal de salida siga a la de referencia. El lazo cerrado se compone de un controlador proporcional, en el cual se introduce el valor de $k_{control}$ que se desee, necesario para convertir la señal que proporciona los ángulos en una señal con los pares de las articulaciones, y un elemento de saturación para representar una limitación que tendrían los actuadores en la realidad, ya que limita los pares que pueden ejercer.

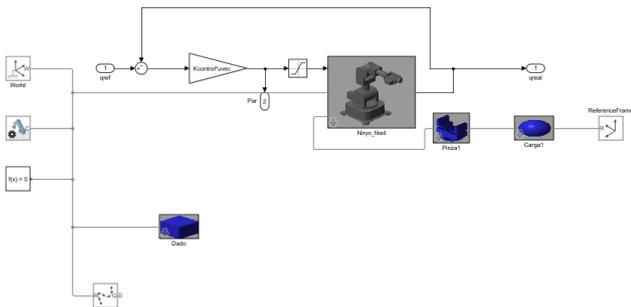


Figura 5: Estación para el modelo dinámico con un lazo

El icono del robot contiene dos señales de entrada, una de Simscape para asociarlo al sistema mecánico y otra de Simulink que introduce los ángulos de las articulaciones del robot, al igual que se ha explicado en el modelo cinemático. Sin embargo, a diferencia que el modelo anterior, tiene dos salidas, una de Simscape para asociarlo a otros componentes mecánicos y otra necesaria para realizar la realimentación de las posiciones angulares.

Las entradas en las articulaciones son los pares correspondientes a cada una, no los ángulos. En este caso, sí existen salidas en las articulaciones, las cuales miden el ángulo de cada una, este modelo se muestra en la Figura 6.

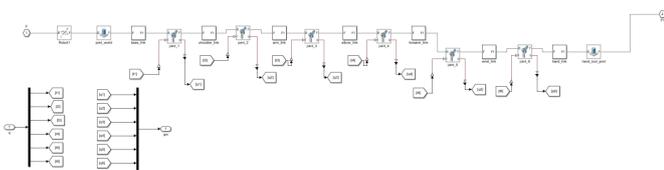


Figura 6: URDF del robot del modelo dinámico con un lazo

2.1.3. Estación con un modelo dinámico con doble lazo

El controlador del sistema en lazo cerrado determina las oscilaciones que tiene el robot en movimiento, por lo tanto, se ha introducido un nuevo lazo interno que realimenta la velocidad para reducir las oscilaciones del sistema, así aparece en la Figura 7.

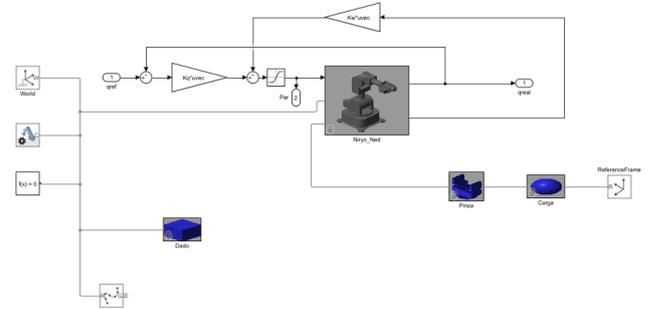


Figura 7: Estación para el modelo dinámico con dos lazos

Al haber incluido este lazo, se ha modificado el icono del robot añadiendo una salida necesaria para realizar la realimentación de la velocidad.

Como consecuencia, se ha incluido una nueva salida en las articulaciones con el objetivo de medir la velocidad de éstas, como se muestra en la Figura 8.

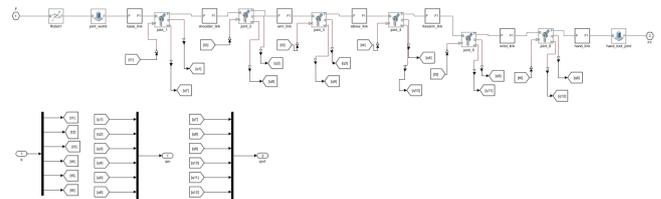


Figura 8: URDF del robot del modelo dinámico con dos lazos

Para esta estación se han realizado distintos ensayos donde se varía el valor de los controladores y de la saturación para una misma trayectoria del robot, cada coordenada articular varía de 0 a -30°, obteniendo los siguientes resultados:

- Ensayo 1: $K_q = 100$, $K_w = 0$, $sat = 10000$ Nm (Sin incluir lazo interno). Los resultados aparecen en la Figura 9.

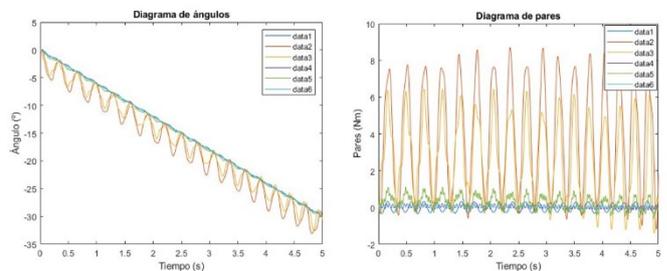


Figura 9: Diagramas obtenidos del primer ensayo

El robot no es capaz de seguir la trayectoria definida sin oscilaciones debido al bajo valor del controlador del lazo externo y la inexistencia del controlador en el lazo interno.

Se puede observar en el segundo diagrama como los actuadores que se encuentran en la posición 2 y 3 ejercen un mayor par para lograr llegar a la posición final.

- Ensayo 2: $K_q = 10000$, $K_w = 0$, $\text{sat} = 10000$ Nm. Los resultados se muestran en la Figura 10.

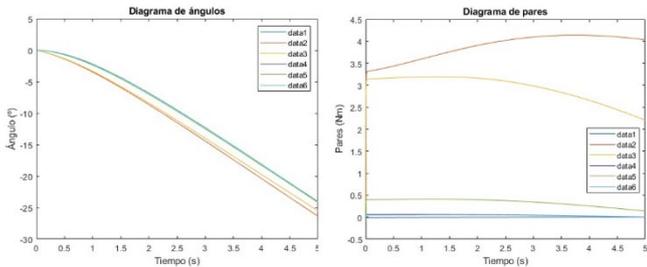


Figura 10: Diagramas obtenidos del segundo ensayo

Según se esperaba, el controlador del lazo interno ha conseguido reducir las oscilaciones del sistema, sin embargo, ha desencadenado una posición final distinta a la deseada. También se ha visto como en el diagrama de pares ya no aparecen oscilaciones, y además, ha reducido el par que debe ejercer cada actuador.

- Ensayo 3: $K_q = 1000$, $K_w = 10$, $\text{sat} = 4$ Nm. Los resultados aparecen en la Figura 11.

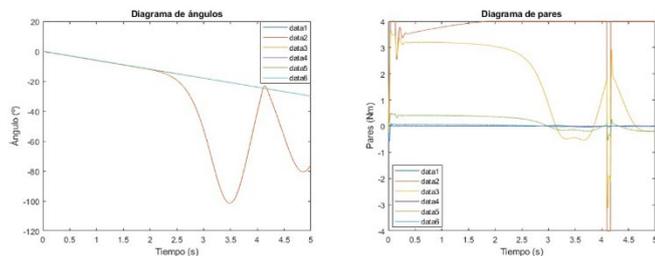


Figura 11: Diagramas obtenidos del tercer ensayo

Los actuadores del robot se han visto limitados a la hora de ejercer el par suficiente para realizar la trayectoria deseada. El sistema necesita ejercer un valor de par específico pero la saturación no lo permite, por lo que no se consigue la suficiente fuerza para elevar las articulaciones del robot y como consecuencia no realiza la trayectoria deseada.

2.2. Aplicación para Niryo Ned

La aplicación se encarga de establecer la comunicación entre el usuario y el robot, en este caso, Niryo Ned, aunque con pequeñas variaciones podría ser útil para cualquier otro robot. Se ha diseñado con un objetivo educativo y además, demostrar cómo se puede manipular este robot desde una aplicación distinta a las existentes hasta ahora, como es Niryo Studio.

La interfaz creada permite mandar instrucciones tanto a una estación de Matlab que contenga el robot que se quiere

controlar, SimRob de Niryo Ned, como al robot real o a la simulación del robot en Rviz, usando en estos dos últimos su IP. La Figura 12 muestra un esquema de la comunicación que permite realizar la aplicación creada.

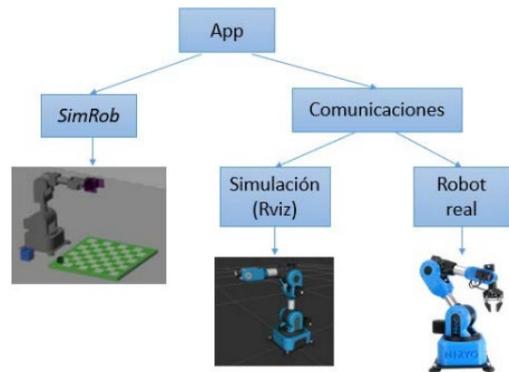


Figura 12: Diagrama de comunicaciones de la aplicación

En la Figura 13 se tiene una visión global de la pantalla al iniciarla y las distintas funciones que permiten el movimiento del robot.

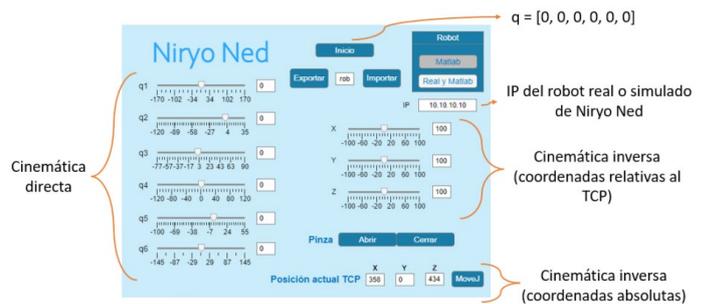


Figura 13: Captura de la aplicación al iniciarla

Se ha comprobado el correcto funcionamiento de las tres comunicaciones. El entorno de trabajo que aparece al establecer cada comunicación se muestra en las figuras 14, 15 y 16.

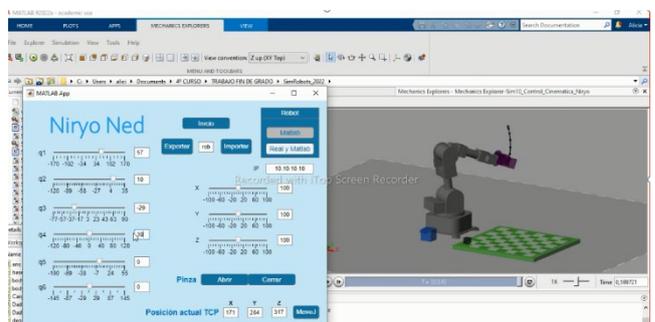


Figura 14: Resultado de la comunicación con SimRob (Matlab)

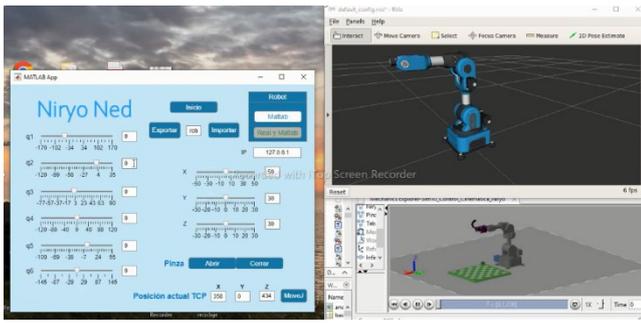


Figura 15: Resultado de la comunicación con la simulación de Niryo Ned (Rviz)



Figura 16: Resultado de la comunicación con Niryo Ned real

3. Conclusiones

Aunque la instalación de todas las aplicaciones para la comunicación de Matlab con Niryo Ned es un proceso laborioso, es de gran utilidad didáctica conocer el esquema de comunicaciones completo. De esta forma, el alumno no se rige solo a programar los movimientos del robot, sino que es capaz de entender la forma en la que se comunica.

Gracias al diseño de la interfaz en App Designer, el alumno podrá poner en práctica sus conocimientos teóricos de robótica, como el movimiento mediante cinemática directa e inversa. Además, un gran aporte de esta aplicación es que el alumno podrá comprobar cómo el robot real realiza sus trayectorias programadas. En la clase solo podrá conectarse una persona al robot, pero una vez se desconecte, cualquier otro compañero podrá controlar los movimientos del robot.

Respecto al modelo dinámico estudiado, se puede concluir que al incluir el elemento de saturación de los actuadores para que el modelo fuese más realista, conlleva una limitación a la hora de la ejecución de los movimientos.

Por último, entre todas las funciones que desempeña la pantalla creada, se ha incluido el movimiento relativo al TCP (Tool Center Point) para conseguir una mayor precisión a la hora de mover la herramienta de trabajo.

Referencias

- A. Barrientos, L. F. Peñín, C. Balaguer y R. Aracil, Fundamentos de robótica, Madrid: McGraw-Hill, 1997.
ISBN: 84-481-0815-9
- App Designer (versión 2022a). Software Mathworks.
- L. M. Crespo Martínez y F. A. Candelas Herías, Introducción a TCP/IP. Sistemas de transporte de datos, Alicante, 1998.
ISBN: 84-7908-435-9.
- Matlab (versión 2022a). Software Mathworks.
- Ned ROS Documentation (v4.0.1). Manual del operador Niryo SAS, 2022.
- Ned User Manual (v3.1.1). Manual del operador: Niryo SAS, 2021.
- Niryo Studio User Manual (v4.0.1). Manual del operador: Niryo SAS, 2022.
- Robotic System Toolbox (versión 2022a). Software Mathworks.
- «ROS (Robot Operating System)» [En línea]. Available: <http://wiki.ros.org/es>. [Último acceso: Junio 2022].
- S. Arévalo y A. Herreros, «Aplicación de Matlab y Simulink para estaciones robóticas» de XLII Jornadas de Automática: libro de actas., Castelló, 2021
DOI: 10.17979/spudc.9788497498043.625
- Simscape Multibody (versión 2022a). Software Mathworks.
- Simulink (versión 2022a). Software Mathworks.