

Desarrollo de un software para la simulación y control de un robot industrial

LAZARO-ARVIZU, Y*†, MORALES-CAPORAL, R, ORDOÑEZ-FLORES, R, QUINTERO-FLORES, P y LEAL-LÓPEZ, M

Recibido 5 de Julio, 2015; Aceptado 24 de Noviembre, 2015

Resumen

Hoy en día la programación de robots hace uso de herramientas de simulación que reproducen la dinámica del robot, tanto para capacitar al personal que lo utiliza previo a su operación, así como para eliminar movimientos erróneos antes de su implementación en el robot real. En algunas aplicaciones académicas, MATLAB® que es un software de desarrollo integrado (IDE) permite cálculos y visualizaciones (gráficas) de alta dimensión y excelente resolución, por lo que lo hace una herramienta muy poderosa para poder desarrollar aplicaciones orientadas a la robótica, como es el caso del Robotic Toolbox® con el cual se pueden simular los movimientos de un robot en un entorno gráfico. En este artículo se presenta un conjunto de aplicaciones utilizando esta herramienta con fines didácticos, para diseñar y controlar las trayectorias de un robot KUKA IRB-2600 previo a su implementación y que se visualiza en tres dimensiones. Las aplicaciones propuestas permiten a los alumnos realizar prácticas, para aprender el manejo de un robot y visualizar los movimientos sin necesidad de tener el robot real. Para comprobar el software propuesto, se utilizan comunicaciones OPC (OLE for Process Control) entre el servidor de ABB y el cliente de MATLAB® desarrollado para mover un robot real, en este caso un KUKA IRB-2600. El software desarrollado dispone de un entorno gráfico que facilita la interacción con el usuario.

Robótica, Simulación, Programación, Docencia.

Abstract

Today robot programming makes use of simulation tools that reproduce the dynamics of the robot, both to train personnel using prior to operation and to eliminate erroneous movements prior to implementation in real robot. In some academic applications, MATLAB software is Integrated Development Environment (IDE) enables calculations and visualizations (graphic) High dimension and excellent resolution, making it a very powerful tool to develop robotics applications-oriented, such as Robotic is the case with Toolbox® which can simulate the movements of a robot in a graphical environment. In this article a set of applications is presented using this tool for teaching purposes, to design and control the trajectories of a KUKA robot IRB-2600 prior to implementation and displayed in three dimensions. The proposed applications allow students to do internships to learn how to use a robot and visualize movements without having the real robot. To test the proposed software, communications OPC (OLE for Process Control) between the server and client ABB developed MATLAB to move a real robot, in this case a KUKA IRB-2600 are used. The developed software has a graphical interface that facilitates user interaction.

Robotics, Simulation, Programming, Teaching.

Citación: LAZARO-ARVIZU, Y, MORALES-CAPORAL, R, ORDOÑEZ-FLORES, R, QUINTERO-FLORES, P y LEAL-LÓPEZ, M. Desarrollo de un software para la simulación y control de un robot industrial. Revista de Tecnología e Innovación 2015, 2-5: 958-967

* Correspondencia al Autor (Correo Electrónico)

† Investigador contribuyendo como primer autor.

Introducción

Durante los estudios de algunas ingenierías, las asignaturas relacionadas con robótica tienen dos objetivos que no siempre son fáciles de combinar. Por un lado, se pretende proporcionar al alumno las bases teóricas de la robótica, entre ellas se explican los sistemas de coordenadas, las trayectorias, la cinemática y dinámica del robot y por otro el control del mismo. [1, 2, 3, 4].

Además, de ser posible se desea que el alumno aprenda a programar un robot en su entorno real, con sus particularidades en sus lenguajes. Una solución aquí propuesta para observar ambos objetivos es el uso de algunas aplicaciones de MATLAB®/ SIMULINK® y otras desarrolladas para robótica, donde de una forma práctica se puede aplicar y visualizar la teoría de robots simulados y reales.

MATLAB® tiene el “Robotic Toolbox®” donde se pueden desarrollar algoritmos teóricos para el modelado de robots, estudio de su cinemática, dinámica y diseño de trayectorias sobre un robot genérico [5, 6].

La aplicación “ARTE®” (“A Robotic Toolbox for Education”) [7] añade a la anterior aplicación el modelo en tres dimensiones (3D) de un gran número de robot industriales, con funciones para obtener sus cinemáticas inversas, de forma que los conceptos teóricos pueden ser aplicados a robots industriales.

En el Instituto Tecnológico de Apizaco (ITA) se dispone de dos robots industriales KUKA [8], uno de ellos es un antropomórfico de seis grados de libertad con una capacidad de 16 kg, ubicado dentro del tamaño de los de serie mediano, empleado principalmente en la industria de la automatización para tareas de soldadura, recubrimientos, manipulación de objetos, etc. Y en este caso para fines de investigación y docencia.

Las aplicaciones desarrolladas conseguirán que el alumno pueda comprender la teoría con la herramienta de trabajo que luego va a emplear. Se han realizado pantallas para la visualización del robot en 3D, una aplicación grafica basada en GUIDE® de MATLAB® [9], y simulaciones del robot en 3D basadas en SimMechanic® [10, 11, 12].

La aplicación propuesta está basada en el servidor OPC que dispone ABB [13] y el OPC cliente de MATLAB® [14].

Un servidor OPC es el método de conectividad de datos basado en los estándares más populares del mundo. Es utilizado para responder a uno de los mayores retos de la industria de la automatización: cómo comunicar dispositivos, controladores y/o aplicaciones sin caer en los problemas habituales de las conexiones basadas en protocolos propietarios.

Dicha aplicación permite, para un período de muestreo dado, mostrar y medir las variables de salida y permanentes del robot desde una aplicación OPC cliente.

Se consideran seis variables de salida en la estación robotizada, una para cada eje del robot. Estas variables serán leídas por el OPC servidor y modificadas desde MATLAB® por una aplicación OPC cliente. En la función de visualización en 3D del robot se ha añadido la opción de modificar de forma simultánea el valor de estas variables para definir la posición de los ejes. El procedimiento que el robot ejecuta en su propio lenguaje RAPID lee de forma ininterrumpida las seis variables de los ejes, y se mueve a la posición correspondiente. De esta forma se consigue que la trayectoria del robot sea controlada por la aplicación de MATLAB®, y que el robot y su modelo en 3D sigan la misma trayectoria de forma simultánea.

El alumno podrá aplicar esta herramienta tanto en el robot virtual representado con RobotStudio® [15] como el real, ya que el servidor OPC es válido para ambos casos. El alumno puede analizar y validar su programa de control en el robot virtual, y una vez verificado su funcionamiento ejecutarlo y verificarlo en el robot real. En cualquier caso, la aplicación es válida para poder ser aplicada con el robot en modo manual en condiciones de máxima seguridad, ya que si se suelta la tecla de hombre muerto, el robot se para automáticamente.

El resto del artículo tiene la siguiente estructura, en la sección 2 se analizan las aplicaciones en MATLAB® / SIMULINK® relacionadas con robótica, además de la cinemática del robot.

En la sección 3, se analiza las posibles comunicaciones entre un robot y una computadora. En la sección 4, se detallan las aplicaciones realizadas en MATLAB® / SIMULINK®, y en la sección 5 se citan las conclusiones del artículo.

Sistemas robotizados en MATLAB® / SIMULINK®

MATLAB® / SIMULINK® son dos entornos de cálculo numérico y visualización de datos de la compañía Matworks® [10, 16], con grandes posibilidades para el diseño de sistemas dinámicos y su simulación. Dentro de las aplicaciones comerciales de ambos entornos no existe uno específico para el diseño y simulación de sistemas robotizados, pero diversos autores han desarrollado aplicaciones de software libre usando MATLAB® / SIMULINK® para sistemas robotizados. [17, 18, 19] La aplicación de robótica más conocida para MATLAB® / SIMULINK® es "Robotic Toolbox®" realizada por Peter Corke [6] que desarrolla los contenidos del libro del propio autor [5]. Dicha aplicación dispone de software para robótica fija y móvil.

En la parte de robótica fija propone algoritmos para modelar un robot usando el método de Denavit-Hartenberg, con los que se puede obtener la cinemática directa del robot. La aplicación propone un método general de cinemática inversa para un robot con muñeca esférica, que puede ser aplicada a robots PUMA. Los modelos gráficos de los robots son muy simples, pero genéricos, ya que se representan únicamente los ejes y las uniones entre ellos. En la última versión se ha desarrollado un modelo 3D para el robot PUMA. La dinámica de los robots se representa mediante bloques convencionales de Simulink®.

La mayoría de los fabricantes de robots industriales han desarrollado modelos gráficos en 3D utilizando programas de diseño asistido por ordenador (CAD), y estos modelos en muchos casos son libres y se pueden obtener de internet. Del archivo del robot en los principales formatos de CAD se puede pasar a una estructura compuesta por un fichero "*.xml" con los datos del robot y ficheros "*.stl" con la información gráfica de cada modelo de brazo del robot. Existen aplicaciones de MATLAB® para leer la información de los ficheros "*.stl" y obtener graficas en 3D de los brazos del robot en MATLAB®.

La aplicación "ARTE®" ("A Robotic Toolbox for Education") [7] aprovecha estos ficheros para generar modelos de los principales brazos robóticos del mercado.

Esta aplicación dispone de la cinemática directa, basada en Denavit-Hartenberg, e inversa de un conjunto de robots. También dispone de datos dinámicos de alguno de estos robots. Con ello se puede estudiar los movimientos y trayectorias de un robot industrial viendo su evolución sobre modelos gráficos de 3D.

Además, esta aplicación dispone de algoritmos similares a las instrucciones de movimiento del lenguaje RAPID, para comprender la forma en la que se diseñan estas funciones y su aplicación a robot industriales.

La aplicación grafica ROKISIM (Robot Kinematic Simulation) [20] muestra un modelo grafico en 3D de muchos robots industriales y simula sus movimientos básicos.

La aplicación comercial SimMechanic® [11] para SIMULINK® dispone de una herramienta especial, SimMechanic® Link [12], que permite importar modelos de robot realizados con programas de CAD donde la información de su estructura esta en ficheros "*.xml" y la información sobre sus brazos en ficheros "*.stl". A partir de esta información, la aplicación desarrolla un modelo SIMULINK® (SimMechanic) con la estructura en serie articulación-brazo del robot y un modelo grafico en 3D con la figura del robot. Sobre el diagrama de SIMULINK® se pueden hacer cambios para simular los lazos de control de cada brazo del robot y de esta forma, simular la dinámica del mismo.

Se concluye indicando que existen muchas aplicaciones para modelar y simular un robot fijo usando MATLAB® / SIMULINK®. Algunas son genéricas para cualquier tipo de robot definido con el método de Denavit-Hartenberg y otras aprovechan los modelos de CAD de robot industriales para estudiar la cinemática y dinámica de robots concretos.

Cinemática del robot

La cinemática del robot estudia el movimiento del mismo con respecto a un sistema de referencia. Así la cinemática se interesa por la descripción analítica del movimiento espacial del robot como una función del tiempo.

Pero primordialmente por la orientación y posición del extremo final del robot con los valores que toman sus coordenadas articulares.

La cinemática estudia dos problemas particulares en el robot. El primero de ellos, se conoce como el problema cinemático directo, y consiste en determinar, los valores conocidos de las articulaciones, cual es la posición del extremo final del robot con respecto al sistema de coordenadas que se toman como referencia. [1]

El segundo es el denominado problema cinemático inverso, este resuelve la configuración que debe adoptar el robot para que su extremo alcance una posición y orientación conocida. [1]

Cinemática directa del robot IRB-2600

Un robot manipulador puede considerarse una cadena cinemática formada por objetos rígidos o eslabones unidos entre sí mediante articulaciones.

Se puede establecer un sistema de referencia fijo situado en la base del robot y describir la localización de cada uno de los eslabones con respecto a dicho sistema de referencia.

Las herramientas matemáticas que se han utilizado para describir la configuración del robot son las matrices de transformación homogénea y el algoritmo de Denavit-Hartenberg [1]. En la tabla I se muestran los parámetros del algoritmo de Denavit-Hartenberg correspondientes a la cinemática directa del robot IRB-2600.

i	θ_i	d_i	a_i	α_i
1	$\theta_1 - 90^\circ$	445	150	-90
2	θ_2	0	700	0
3	θ_3	0	0	90
4	θ_4	-795	0	-90
5	θ_5	0	0	90
6	θ_6	0	-85	180

Tabla 1 Valores de Denavit-Hartenberg correspondiente al robot IRB-2600.

Comunicación de la aplicación propuesta con el robot real

La comunicación entre la aplicación de MATLAB® / SIMULINK® y el robot se podrá haber realizado usando múltiples protocolos. Las ventajas e inconvenientes de dos de ellos se resumen a continuación:

Comunicación por OPC: ABB dispone de un servidor OPC que puede ser ejecutado desde una computadora. Esta aplicación detecta los robots conectados a la computadora, ya sean reales o virtuales, y extrae de ellos, en cada período de muestreo fijado, la información del sistema, sus variables de entradas y salidas, y las variables persistentes del programa Rapid del robot. Desde una aplicación OPC® cliente se puede leer y modificar dichas variables. No es preciso que la lectura y escritura estén sincronizadas, cuando se modifique una variable desde el exterior, dicha variable queda modificada en la estación del robot.

La comunicación entre la aplicación de Matlab y el robot de ABB ha sido realizada haciendo uso del servidor OPC proporcionado por ABB [13], y del cliente OPC proporcionado por la aplicación "OPC Toolbox®" de MATLAB® [14]. La comunicación se ha realizado usando variables de salida analógicas virtuales por una razón de seguridad.

La escritura en variables de salida en RAPID es posible en estado manual y automático, mientras que la escritura en variables persistente solo es posible en automático. Con fines educativos y por seguridad, el robot IRB-2600 se suele usar en modo manual, ya que en este modo se puede parar el robot ante cualquier eventualidad dejando de pulsar el botón de hombre muerto. Se han detenido seis variables de salida virtuales normalizadas entre [-100; 100] para transmitir la información de los seis ejes del robot.

La forma de comunicarse entre MATLAB® y el robot IRB-2600 es la siguiente:

MATLAB®: Cuando la aplicación de MATLAB® quiera cambiar la posición del robot debe modificar el valor de las variables del servidor OPC correspondientes a los ejes del robot.

Robot IRB-2600 (RAPID): El programa en Rapid consiste en un bucle infinito que lee las salidas analógicas correspondientes a los ejes y desplaza al robot a la posición definida por los valores de las variables.

El protocolo que se han empleado proporciona independencia frente al tiempo de lectura y escritura.

Si se envía una trayectoria definida por muchas posiciones de ejes con un período de muestreo muy bajo, puede que el programa Rapid no sea capaz de leer algún punto pero llegara al destino sin quedarse bloqueado. La sincronización entre MATLAB® y Rapid depende de la velocidad con que envía los datos a las variables de salida.

Descripción de las aplicaciones

En este trabajo se han desarrollado tres aplicaciones para modelar y simular el robot IRB-2600 y sus movimientos respectivamente con MATLAB® y después poder mover el robot real, o el virtual si no se cuenta con el robot real.

Las dos primeras aplicaciones hacen uso de las funciones de "Robotic Toolbox®" [6] y "ARTE®" [7]. La tercera aplicación, hace uso de las herramientas de SimMechanic® [11, 12].

Aplicación grafica de simulación

Primero se modela el robot IRB-2600, con su cinemática directa e inversa y se programan una función `plot irb2600()` para visualizar las posiciones del robot en 3D. La cinemática directa y las trayectorias del robot se han diseñado utilizando la aplicación de [6], mientras que la cinemática inversa ha sido obtenida de la aplicación de [7].

A partir de los ficheros "*.stl" del robot, se ha construido una función `plot irb2600()` con las siguientes objetivos (Ver anexo). La fig. 1 muestra el robot simulado.

Dibujar el robot en 3D con la posición de ejes deseada. Se puede dibujar una secuencia de posiciones si la entrada es una matriz de posiciones de ejes. Se puede definir un tiempo de parada entre posiciones que va a ser útil para sincronizar el movimiento de la gráfica y el del robot real.

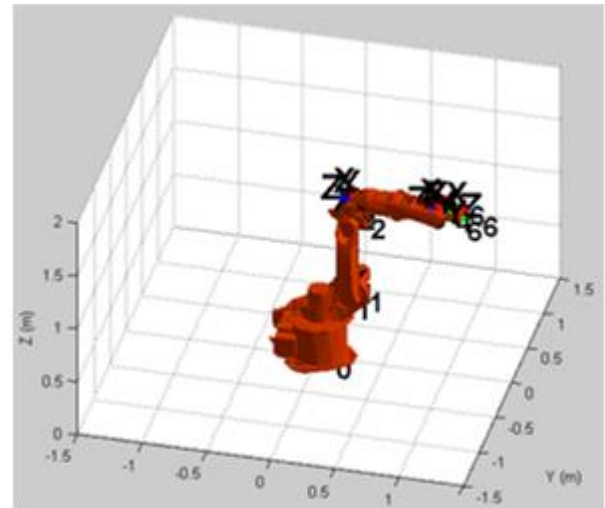


Figura 1 Robot simulado en 3D.

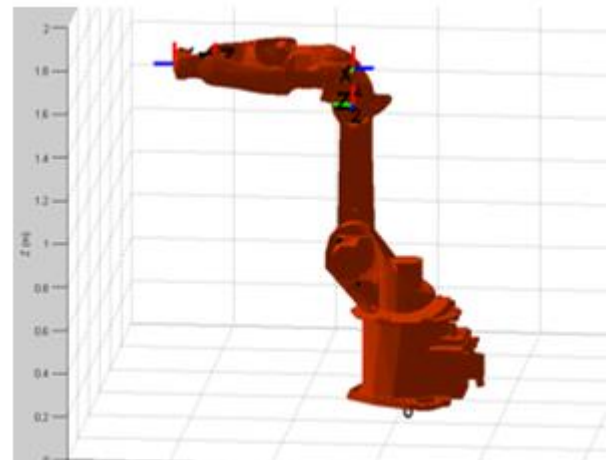


Figura 2 Posición inicial del robot simulado



Figura 3 Posición inicial del robot real.



Figura 5 Posición robot real.

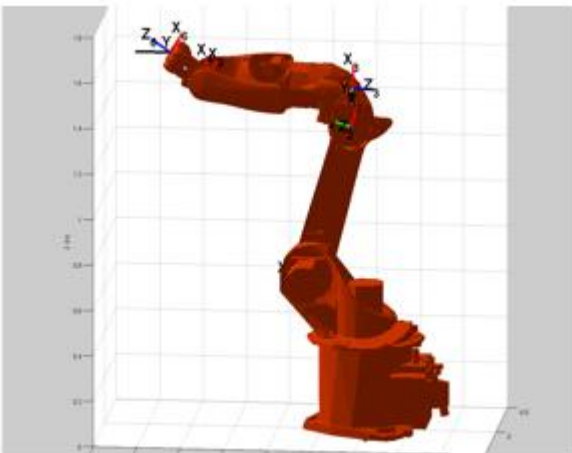


Figura 4 Posición final del robot simulado.

El usuario puede programar la posición en las variables de entrada del robot correspondiente, usando el servidor OPC de ABB.

Cuando el programa de Rapid se activa, el robot se moverá a las mismas posiciones que el robot simulado. El tiempo de espera entre posiciones es un parámetro de sintonización entre el robot y la simulación.

Las siguientes imágenes muestran las pruebas que se hicieron en la fig. 2 y 3 muestran la posición inicial del robot real y el robot simulado.

Las posiciones finales se muestran en la fig. 4 y 5.

Aplicación grafica (GUIDE®) para el movimiento del robot

A partir de la función `plot irb2600()` se ha realizado una aplicación grafica en GUIDE® de MATLAB® [9] que permite al usuario realizar de forma segura los movimientos de ejes por ángulos y posiciones cartesianas. Dichos movimientos se visualizan con la función `plot irb2600()` y podrán ser enviadas a las variables de los ejes del robot a través del servidor OPC. Con ello, el usuario podrá mover el robot con una aplicación grafica de MATLAB®, de forma similar a como se mueve con la aplicación FlexPendant de ABB [21].

Las principales características de esta aplicación son las siguientes:

- Movimiento a una posición en coordenadas articulares. Se puede grabar dicha posición para simular una trayectoria.
- Movimiento a una posición del punto de trabajo.
- Se puede grabar dicha posición para simular una trayectoria
- Movimiento del robot entre las posiciones grabadas anteriormente, ya sean posiciones articulares o del punto de trabajo.
- Movimientos de demostración del robot.
- Conexión con OPC para comunicar datos.

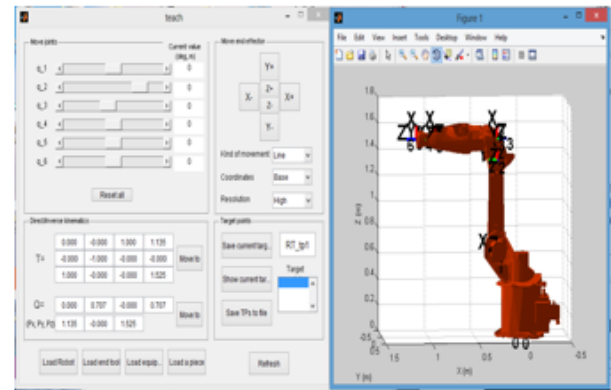


Figura 6 Pantalla inicial de la aplicación gráfica.

La fig. 6 muestra la pantalla inicial de la aplicación gráfica. Se aprecian los diferentes botones para mover el robot mediante coordenadas articulares, coordenadas de punto de trabajo, y resto de elementos mencionados.

Aplicaciones didácticas y de investigación

El objetivo de las aplicaciones realizadas en este artículo es servir de puente entre el estudio matemático y simulado de los sistemas robotizados y el práctico de movimiento real de un robot comercial. Se desea que el alumno pueda realizar algoritmos semejantes a los implementados en lenguaje RAPID y otros algoritmos nuevos que el alumno desarrolle.

Como ejemplos de ejercicios realizables con estas aplicaciones se pueden mencionar:

Modelar trayectorias interpolando puntos en el espacio de coordenadas articulares.

Modelar trayectorias interpolando puntos en el espacio de coordenadas espaciales del punto de trabajo.

Mover el punto de trabajo del robot para minimizar el error entre la posición de dicho punto y una referencia dada, usando un algoritmo de optimización local. Usar para ello incrementos en las coordenadas articulares.

Mover el punto de trabajo del robot en la dirección medida por un sensor de esfuerzos, que el robot IRB-2600 (que actualmente el robot no tiene).

Mover el punto de trabajo del robot hacia una posición móvil detectada por una cámara de visión (proyecto futuro) asociada a una aplicación de MATLAB®.

Programar las trayectorias punto a punto, adicionando al robot una pistola para soldadura y que este pueda realizar las trayectorias repetidamente (En proceso).

Conclusiones

Se han presentado las propuestas de aplicaciones usando el "Robotic Toolbox®" de MATLAB® / SIMULINK® y su comunicación con el software de ABB para controlar, simular, diseñar e implementar los movimiento de un robot industrial. Esto es posible gracias a que la herramienta de comunicación entre el robot y las aplicaciones de MATLAB® la cual se lleva a cabo por medio de un intercambio de información usando el servidor OPC de ABB y el cliente OPC de MATLAB®. Con esta herramienta los alumnos pueden realizar en un entorno visual la programación previa de los movimientos de cada eslabón del robot para una aplicación específica. Una vez verificado el diseño de los movimientos por simulación, es posible implementarlos de manera rápida en el robot real, en caso de contar con él. Eliminando movimientos riesgosos tanto para el usuario como para el equipo.

Referencias

Barrientos A., Peñin L.F., Balaguer C. y Aracil P. (2007) Fundamentos de Robótica (2 edición), MacGraw-Hill.

Ollero A. (2001) Robótica, manipuladores y robots móviles, Editorial Macombo.

Pérez Cisneros M.A., Cuevas Jiménez E.V., Zaldívar Navarro D. (2014) Fundamentos de Robótica y Mecatrónica con MATLAB y SIMULINK. Editorial RA-MA.

Spong M., Hutchinson S y Vidyasagar (2006) M., Robot, Modeling & Control, Wiley.

Corke, P. (2013) Robotics, Vision & Control, Springer.

Peter Corke (2014) Robotics Toolbox. [http://petercorke.com/Robotics Toolbox.html](http://petercorke.com/Robotics%20Toolbox.html).

Gil Aparicio A. (2014) ARTE (A Robotics Toolbox for Education) Universidad Miguel Hernández (Elche, España), [http://arvc.umh.es/arte/index en.html](http://arvc.umh.es/arte/index%20en.html)

www.kuka-robotics.com/Robotics.

MathWorks (2014). GUIDE Toolbox User's Guide.

MathWorks (2014). SIMULINK User's Guide.

MathWorks (2014). SimMechanic User's Guide.

MathWorks (2014). SimMechanic Link User's Guide (2014).

ABB robotic. Application manual: IRC5 OPC Server. Doc. ID: 3HAC023113-001.

MathWorks (2014). OPC Toolbox User's Guide.

ABB robotic. Manual del operador: RobotStudio. Doc. ID: 3HAC032104-005.

MathWorks (2014). MATLAB® User's Guide.

Hassine Belhadj, Saber Ben Hassen, Khaled Kaániche and Hassen Mekki, “KUKA Robot control based Kinect image analysis”, in IEEE International Conference on Individual, and Collective Behaviors in Robotics, pp. 21-26.

Rene González Rodríguez and Luis Hernández Santana, “Platform to develop real time visual servoing control in kinematics systems”, in revista Ingeniería Mecánica, vol. 15, no. 3, pp. 233-241, 2012. ISSN 1815-5944.

Velásquez Lobo, Ramirez Cortés and Rangel Magdaleno, “Modeling a Biped Robot on Matlab/SimMechanics”, in IEEE International Conference on Individual, and Collective Behaviors in Robotics, pp. 203-206.

Bonev I. (2013) ROKISIM (Robot Kinematic Simulation), Escuela de Superior de Tecnología (Quebec, Canadá)
<http://www.parallemic.org/RoKiSim.html>.

ABB robotic. Manual del operador: Introducción a RAPID (RobotWare 5). Doc. ID: 3HAC029364-005

ABB robotic. Manual del operador IRC5 con FlexPendant. Doc. ID: 3HAC16590-5.