

Estimulando el Aprendizaje en el Control de Robots Mediante la Interacción Hombre-Máquina

Raúl Santiago Pérez^a, José J. Quintana^a, Moises Diaz^b, Miguel A. Ferrer^c

^aDepartamento de Ingeniería Electrónica y Automática, Universidad de Las Palmas de Gran Canaria, Las Palmas, España. raul.santiago102@alu.ulpgc.es, josejuan.quintana@ulpgc.es

^b Universidad del Atlántico Medio, Las Palmas, España. moises.diaz@atlanticomedio.es

^cInstituto Instituto para el Desarrollo Tecnológico y la Innovación en Comunicaciones (iDeTIC) Universidad de Las Palmas de Gran Canaria, Las Palmas, España. miguelangel.ferrer@ulpgc.es

RESUMEN

Esta propuesta de innovación docente consiste en el desarrollo de una práctica de laboratorio para el control de robots industriales con la mano. El objetivo final es mejorar la atracción del alumno a esta disciplina aprendiendo a controlar robots mediante el movimiento de la mano. Específicamente, para reconocer los gestos de la mano se ha empleado el dispositivo electrónico Leap Motion como instrumento captador del movimiento y, como robot industrial, el robot ABB, modelo IRB 120. Para llevar a cabo este tipo de interacción humano-robot, este artículo desarrolla la comunicación entre Leap Motion con Matlab, Matlab con RobotStudio y RobotStudio con Robot ABB IRB-120. En consecuencia, el alumno logrará adquirir conocimientos en programación de robots a partir de una práctica de laboratorio que permite la interacción humano-robots.

Palabras clave: Interacción Humano-Robot, Leap Motion, Robot ABB, Programación de Robots, Gestos de la mano

1. INTRODUCCIÓN

La robótica es una de las disciplinas formativas más atractivas gracias a la posibilidad de automatizar, gestionar y optimizar procesos industriales. Al igual que muchos otros campos tecnológicos, la robótica evoluciona de manera progresiva, apareciendo nuevos retos industriales que, indudablemente, impacta en la formación académica. Enseñar y aprender a controlar robots industriales será, por tanto, necesario en un curso de robótica industrial que pretenda introducir al alumno en esta disciplina.

Los avances observados en este campo y el crecimiento de las industrias 4.0 en los últimos años han mostrado que cada vez es más frecuente encontrar ejemplos de interacciones humano-robots en la industria [4]. Esto pone de manifiesto la necesidad de la formación de estos contenidos en la academia. En consecuencia, se considera necesario el desarrollo de prácticas docentes que doten al alumno de competencias actualizadas en la realización de este tipo de conexiones e interacciones, cuya presencia en las instalaciones robóticas es cada vez más frecuente [5]. Además, usar tecnología actual en un curso de robótica conllevaría a una mejora en la estimulación y motivación durante el proceso de aprendizaje, así como a una mayor atracción del alumnado hacia estas disciplinas.

En este artículo se presenta el diseño de una práctica de laboratorio sobre la interacción humano-robots que pretende mejorar la estimulación del alumno durante su aprendizaje en control de robots. Así pues, con esta propuesta, el alumno desarrollará una serie de programas mediante los cuales tendrá que reconocer los movimientos ejecutados por las manos, para, posteriormente, realizar un tratamiento de los datos recogidos y generar una interacción humano-robot a tiempo real. La interacción entre humano y robot puede ser de diversos tipos: gestos faciales, movimientos de extremidades del cuerpo, movimiento de la cabeza, etc. En este caso se han elegido los gestos de la mano debido a que se considera la interacción más sencilla y útil para una práctica docente.

Para lograr la interacción humano-robot se ha de recopilar información sobre el dispositivo electrónico encargado de captar los movimientos ejecutados con las manos. En este caso el dispositivo empleado ha sido el Leap Motion. Estos dispositivos electrónicos disponen de un kit de desarrollo de software o SDK disponibles en su página oficial que deben ser instalados en un PC en el que se desee llevar a cabo esta propuesta de estudio. Una vez descargado e instalado el SDK del dispositivo electrónico, el siguiente elemento necesario será el software en el cual se llevará a cabo el desarrollo

del programa encargado de recibir la información procedente del Leap Motion. Matlab ha sido el software empleado en esta propuesta debido a que el alumno que cursa asignaturas de robótica generalmente ha adquirido competencias en programación en Matlab en otras materias tales como informática, la cual se imparte en los primeros años de grado. Además, Matlab se emplea como un programa de apoyo para otras asignaturas. En el caso del Grado en Ingeniería Electrónica Industrial y Automática pueden ser Control de Robots, Ingeniería de Control y Regulación Automática, entre otras. Para esta propuesta se ha considerado la gran versatilidad que ofrece Matlab para facilitar la comunicación entre los dispositivos empleados para el desarrollo de la interacción humano-robot. También, será necesario descargar un software de programación que tenga comunicación con el robot industrial que se desee emplear. Para el robot ABB modelo IRB 120, el software desarrollado por ABB para sus robots es el RobotStudio, el cual permite la programación en lenguaje RAPID. Una vez instalados todos los programas necesarios, a continuación, se comienza a buscar información sobre cada entorno de programación y los comandos a emplear para cumplir los objetivos deseados. Una parte importante de esta programación es conseguir la comunicación entre Leap Motion-Matlab, Matlab-RobotStudio y RobotStudio-RobotABB.

2. DISPOSITIVOS PARA INTERACCIÓN HUMANO-ROBOTS

Para la interacción humano-robots se ha usado el dispositivo Leap Motion y el robot ABB, modelo IRB 120. El primero consiste en un dispositivo óptico que percibe el movimiento de la mano a través de tres leds infrarrojos y dos cámaras. El robot ABB, modelo IRB-120, es un robot industrial de 6 grados libertad ampliamente usado en la industria y la academia. Además, la interacción requiere conocimientos de un lenguaje de programación para el control de robots. En este caso se programará tanto en el lenguaje de programación de Matlab (lenguaje M) como en el lenguaje de programación de RobotStudio (lenguaje RAPID).

2.1 Leap Motion

Leap Motion es un sensor que reconoce cada uno de los movimientos realizados con las manos y es capaz de transformar este movimiento en acciones precisas una vez se ha conectado con un ordenador. Este dispositivo dispone de 2 cámaras que generan una imagen con formato RAW y tres leds infrarrojos que analizan todos los movimientos de las manos en un radio de 1 metro. Además, es capaz de detectar los dedos, las manos y lápices recreando un entorno 3D [6].

La página oficial del producto [1] aporta un kit de desarrollo de software. Esto ofrece una API que permite la programación en varios lenguajes y plataformas de desarrollo. La Figura 1 muestra el logo de alguno de ellos.



Figura 1. Algunos de los lenguajes y plataformas de desarrollo de LeapMotion.

Desde la API se puede obtener todo tipo de información tridimensional referente a antebrazos, manos, herramientas, dedos, e incluso la anatomía de los dedos que se tratan como objetos. La Figura 2-a ilustra este dispositivo.

2.2 Robot ABB modelo IRB 120

El robot concreto usado para esta práctica de interacción humano-robot ha sido el robot ABB, modelo IRB120 [2]. Este robot forma parte de los robots industriales más modernos de ABB de 6 grados de libertad, con un peso de 25kg y una carga útil de 3kg (4kg en posición vertical de la muñeca), y diseñado tanto para fines docentes como fines industriales de fabricación que requieren una automatización flexible. Además, este robot es de utilidad para esta propuesta docente puesto que presenta unas grandes posibilidades de comunicación con sistemas externos.

En cuanto al sistema operativo del robot, está equipado con el controlador IRC5 Compact o IRC5 y el software de control de robots RobotWare. Este software admite todos los aspectos del sistema de robot, como el control del movimiento, el desarrollo y la ejecución de programas, la comunicación, etc. La Figura 2-b muestra el robot propuesto.

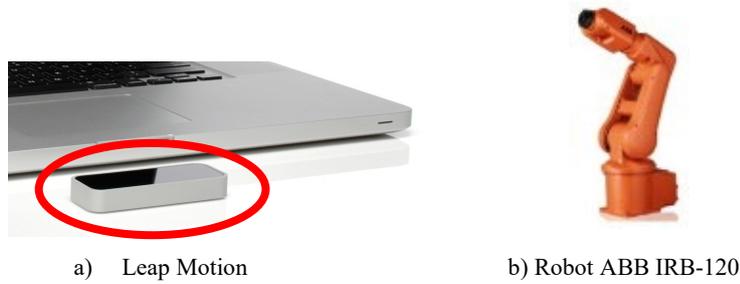


Figura 2. Dispositivos para la interacción humano-robot

2.3 Robot Studio

RobotStudio [3] es una aplicación de PC destinada al modelado, la programación fuera de línea y la simulación de células de robot. Para la correcta utilización de esta aplicación es recomendable tener conocimientos acerca de programación de robots, un manejo genérico de Windows y sobre programas de diseño asistido por computador (CAD) en tres dimensiones.

RobotStudio cuenta con un controlador IRC5 virtual que permite trabajar de manera local en el PC. Este controlador permite simular de manera virtual el movimiento del robot. Esto permite al alumno visualizar las instrucciones programadas en una estación virtual antes de realizar una ejecución final en el robot físico.

El lenguaje de programación de RobotStudio es RAPID, el cual es un lenguaje de programación textual de alto nivel desarrollado por la empresa ABB.

3. INTEGRACIÓN DE LEAP MOTION CON ROBOT ABB IRB 120

En el diagrama de la Figura 3 se puede observar la arquitectura desarrollada en la propuesta didáctica de innovación en robótica para la interacción humano-robot. Para el reconocimiento de gestos será necesario realizar la conexión entre Leap Motion y Matlab. Este entorno permite trabajar con las librerías que ofrece el fabricante de Leap Motion. De esta manera se obtendrán las coordenadas de la posición de la anatomía de una mano y las variaciones de movimientos en el tiempo. Posteriormente, se trabajará con estas coordenadas para definir las coordenadas relativas al área de trabajo del robot.



Figura 3. Diagrama de funcionamiento

Una vez definidos estos parámetros en Matlab, el siguiente paso a resolver es la apertura de un canal de comunicación entre Matlab y RobotStudio. En RobotStudio se recibirán los parámetros de Matlab y se realizará un programa que teniendo en cuenta estas variables, genere un desplazamiento en el robot dentro de las coordenadas determinadas por el programador. A continuación, se detallan los pasos para el desarrollo de la práctica propuesta.

3.1 Conexión Leap Motion- Matlab

Para el lograr el posicionamiento de las manos, el primer paso a seguir ha sido acceder a la página oficial de Leap Motion y descargar el kit de desarrollo de software del producto.

Cuando la descarga del paquete haya finalizado, se ejecutará el archivo .exe que permitirá el reconocimiento del dispositivo por parte del equipo, puesto que dispone de los drivers del controlador. Cuando se han instalado los drivers, ya se puede conectar el dispositivo correctamente y acceder a la interfaz del panel de control del Leap Motion, donde se puede acceder a la configuración de ciertos parámetros. Las opciones más relevantes que ofrece esta interfaz son “Recalibrar dispositivo”, que es un paso fundamental a seguir antes de comenzar a usar Leap Motion correctamente, y el “visualizador de diagnóstico”, mostrado en la Figura 4. Este se encarga de mostrar una ventana en la cual se pueden visualizar los datos que está recogiendo el dispositivo a tiempo real, esta ventana dispone de una gran variedad de opciones como son: el reconocimiento de gestos, las coordenadas cartesianas de los distintos puntos de la mano, velocidad de movimiento, información sobre imágenes por segundo (frames per second), latencia y acceso a las imágenes que recoge la cámara entre otras opciones.

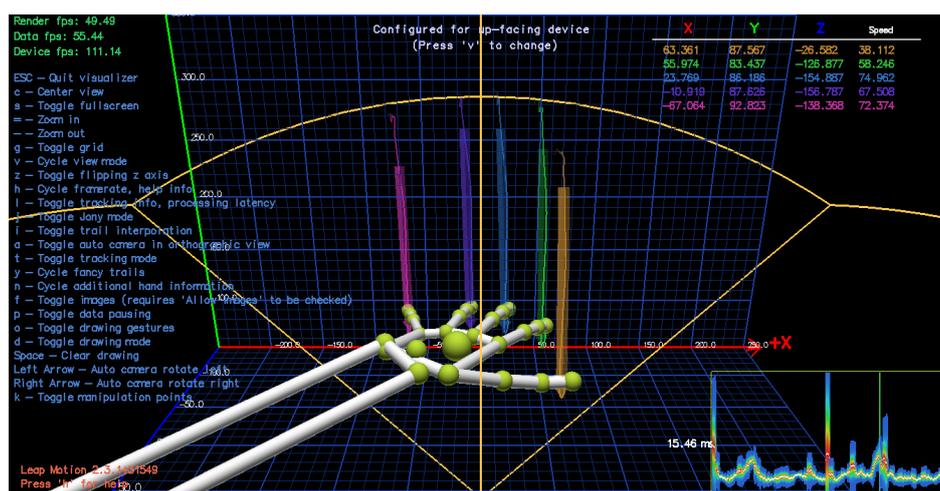


Figura 4. Ejemplo de vista del software de LeapMotion: visualizador de diagnóstico

El kit de desarrollo de software también incluye una serie de documentos y enlaces en los que se puede encontrar toda la información referida al Leap Motion en sus distintos lenguajes de programación, incluyendo también ejemplos de programación de cada uno de ellos. Una parte fundamental de este paquete son las librerías que ofrece para el desarrollo de aplicaciones y programas.

Después de comprobar que Leap Motion funciona correctamente, es necesario enlazar los datos recogidos desde el dispositivo con el programa Matlab, desde el cual se trabajará con estos datos.

Con el fin de lograr esta conexión, se creará un archivo tipo *mex* de Matlab que permitirá obtener los datos del dispositivo Leap Motion. Este tipo de archivos son subrutinas vinculadas dinámicamente que Matlab carga y ejecuta.

Es necesario instalar un compilador de C++ en Matlab y configurar el programa para crear archivos *mex*. En este caso se instalará el compilador “Microsoft Visual C++ 2017”. Para tener este compilador será necesaria la instalación del paquete “Desktop development with C++” del programa Visual Studio Community 2017. Para elegir el compilador en Matlab se hará con el código *mex -setup C++* en la ventana Command Windows [7].

Es necesario aclarar que el compilador seleccionado debe ser ‘Microsoft Visual C++ 2017’ porque el compilador ‘MinGW-w64 C/C++ Compiler’ no reconoce las librerías de Leap Motion, provocando problemas de compilación. Finalizada la instalación del compilador, se continuará con el desarrollo del programa que genere el archivo *mex* [8].

3.2 Comunicación entre Matlab y RobotStudio

Tras realizar la conexión entre Matlab y Leap Motion el siguiente paso dado en la práctica ha sido, el desarrollo de un código a partir del cual se genere una interpolación entre las coordenadas de posición de la mano y el área de trabajo del robot. Para determinar las coordenadas de la mano en el programa desarrollado se empleará el comando propio de las librerías de Leap Motion *f_hands(i).palm_position*, este comando devuelve la posición de las manos en un vector de coordenadas x, y, z. Estos valores de entrada son introducidos en ecuaciones de interpolación para obtener las coordenadas a las que tiene que acceder el robot industrial en el área de trabajo. Las coordenadas de acceso del robot ABB en el área de trabajo se han deducido según una serie de pruebas en las que se enviaba el robot a distintas coordenadas comprobando si se producía algún error.

La comunicación entre Matlab y RobotStudio [9] es necesaria para, una vez tratados los datos por el Matlab, estos sean enviados hasta RobotStudio que se encargará de establecer una conexión con el robot ABB y determinar los movimientos que debe ejecutar.

Para ello es necesario definir la dirección IP del cliente y del servidor, en este caso esa dirección IP viene determinada por el robot ABB, dato que se obtiene desde la pestaña “Archivo”, apartado en línea. Esta dirección IP se empleará en este caso debido a que se quiere realizar una comunicación con el robot ABB, en caso de querer realizar una comunicación con la estación virtual, la dirección IP será: 127.0.0.1.

Antes de ejecutar el código en Matlab se debe instalar la librería Instrument Control Toolbox, para el uso de los comandos tcpip, ByteOrder, fopen, etc. Una vez generado el código en Matlab se abrirá el programa RobotStudio donde se programará el otro canal de comunicación.

Antes de programar en RobotStudio, se debe crear una estación con las siguientes características. Elección del modelo IRB 120 en la biblioteca de ABB, y en opciones de sistemas seleccionar en el apartado “communication” la opción “616-1 PC Interface”. De esta manera se obtendrá la estación deseada en la cual se va a realizar la programación.

Cuando se ha generado la estación virtual, es el momento de comenzar con la programación del robot ABB. Para ello se deberá de acceder a la pestaña “RAPID” de RobotStudio y crear un nuevo módulo. El nuevo módulo se empleará para desarrollar el código de programación que posteriormente se le enviará al robot ABB real.

En cuanto al código de programación que se ha tenido que desarrollar, consta de dos partes. Una primera parte en la que se configuran los parámetros de comunicación entre RobotStudio y Matlab determinados en el apartado “Comunicación entre Matlab y RobotStudio”, y en la cual se definen las variables en las que se van a almacenar los datos trabajados en Matlab. Estos datos son enviados desde Matlab con el comando *fwrite*.

```
fwrite(tc,[Ejex_brazo Ejey_brazo Ejez_brazo extendedFingers], 'int32');
```

El comando se programará de esta manera en Matlab, de tal manera que se determina el canal por el cual se va a enviar, las variables a enviar y su tipo. Por otro lado, RobotStudio recibirá los datos transmitidos desde Matlab, el cual los divide y asigna los nombres de las variables que se quieren emplear en RobotStudio. Los datos se desempaquetan de 4 en 4 bytes puesto que desde Matlab se mandan enteros de 32 bits, lo que significa que son 4 bytes de espacio.

En la segunda parte del código se desarrollará la parte del programa que deberá de ejecutar el robot físicamente, de tal manera, que trabaja con las variables almacenadas previamente y determina las coordenadas a las que debe acceder el robot en cada momento, para realizar esta parte del programa, primero se ha hecho un estudio sobre las coordenadas accesibles por el robot ABB. Este estudio se ha realizado mediante el posicionamiento del robot en las coordenadas permitidas por su software, y a continuación ejecutando un desplazamiento entre los extremos para comprobar que se realizaban sin interrupciones por puntos inaccesibles.

En la Tabla 1 y Figura 5 se puede observar la relación entre los sistemas de referencia del dispositivo Leap Motion y el robot ABB IRB120. También se hará referencia al rango de captación de Leap Motion y a las coordenadas accesibles por el robot ABB IRB120 en su área de trabajo, ambas en milímetros.

Tabla 1. Relación entre sistemas de referencias y áreas de trabajo de Leap Motion y Robot ABB IRB120.

Sist. ref. Leap Motion	Sist. ref. robot ABB	Rango área captación Leap Motion (mínimo, máximo)	Rango área trabajo robot ABB (mínimo, máximo)
X	Y	-240, 240 (mm)	-250, 250 (mm)
Y	Z	50, 350 (mm)	-150, 150 (mm)
Z	X	-250, 250 (mm)	-110, 70 (mm)

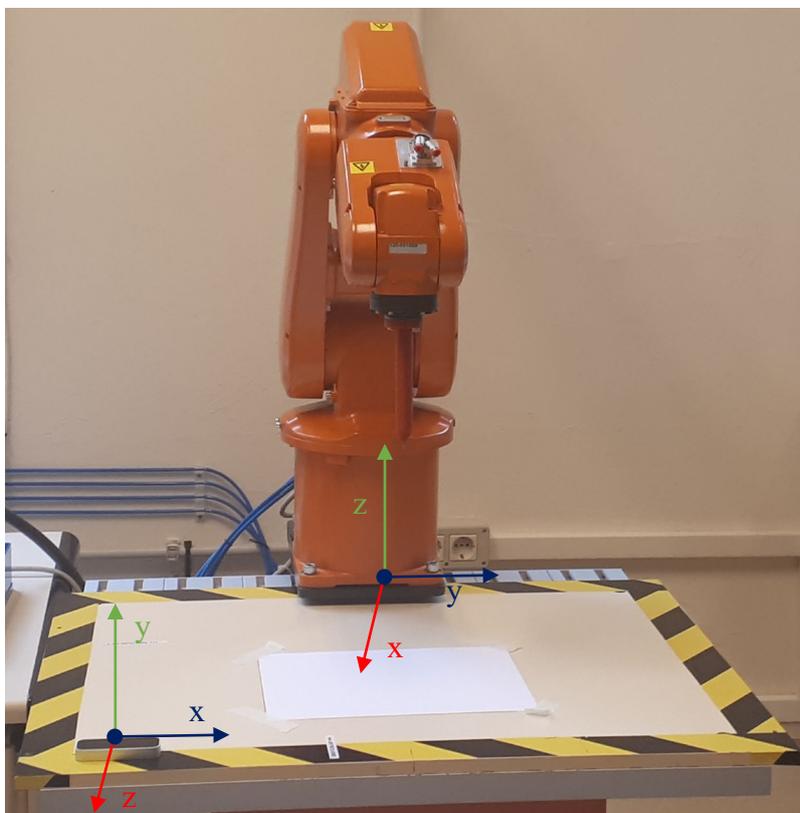


Figura 5. Relación entre sistemas de referencia

3.3 Relación entre RobotStudio y el Robot ABB

Por último, se debe crear una relación entre RobotStudio y el robot ABB para de esta manera comunicarle al robot el código de programa generado en la estación virtual. Para crear esta relación se deben seguir los siguientes pasos:

1º Acceder a la pestaña “Archivos”, apartado “En línea” y cuando se haya puesto en marcha correctamente el robot, se hará clic en “120-501089en”, este nombre hace referencia a nuestro controlador. Cabe destacar que aparecerá en estado disponible solo cuando este encendido, por lo que no se puede generar una relación previamente.

A continuación, en la ventana RAPID se añadirá el robot como otro controlador.

2º Con la estación virtual y el robot ABB como controladores, ya se puede crear una relación entre ambos.

3º Posteriormente RobotStudio preguntará sobre el nombre que quiera asignarle a la relación.

4º Cuando se ha creado la relación entre los dos controladores aparecerá un nuevo apartado en el interior de los controladores llamado “Relaciones”, en su interior se encontrará la relación creada con su nombre.

5º Al hacer clic en cualquiera de las dos relaciones se abrirá una ventana que permite seleccionar los datos que se quieren transmitir entre módulos. Esta dirección es bidimensional, ya que existe la posibilidad de trabajar en el módulo del robot y posteriormente guardarlo en la estación o viceversa. Por lo que una vez determinados los datos que se quieren enviar y la dirección se finaliza la operación haciendo clic en la opción “Transferir ahora...”. Es importante en el instante de enviar datos desde la estación hacia el robot que se le haya permitido el acceso de escritura.

Una vez desarrollados los programas y realizadas las comunicaciones correctamente es el momento de probar el funcionamiento del proyecto. Primero se ha de ejecutar el programa en RobotStudio desde la pestaña RAPID. Luego se iniciará el programa de Matlab. Iniciados los programas se comenzará a visualizar el comportamiento del robot en función a los datos tratados. La Figura 6 muestra el momento en el que se comprueba el comportamiento del robot en el laboratorio.

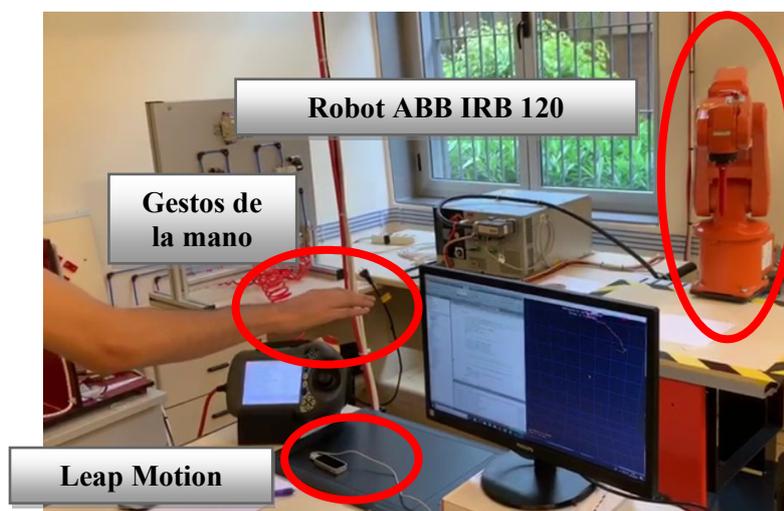


Figura 6. Ejemplo de funcionamiento de la interacción humano-robot en el laboratorio.

4. CONCLUSIONES Y LÍNEAS FUTURAS

En este artículo presentamos una práctica de laboratorio de innovación educativa. Su finalidad es que el alumno aprenda a interactuar con robots, estimulando así su aprendizaje en el campo del control de robots. La práctica pretende que el brazo robótico de ABB modelo IRB-120 reproduzca los movimientos de una mano, los cuales son percibidos mediante el dispositivo Leap Motion. Para ello, es requisito que el alumno estudie la API de Leap Motion y su conexión con el software Matlab. Además, es necesario adquirir conocimientos del lenguaje RAPID, ampliamente usado en programación de robots, poniendo como ejemplo la programación de la configuración de la comunicación entre RobotStudio y Matlab. Sobre RobotStudio el alumno obtendrá unos conceptos básicos para crear estaciones, programar en RAPID y crear relaciones entre la estación virtual y el robot real. El conjunto de competencias que se adquieren con esta práctica permite introducir al alumno en un campo tecnológico cada vez más usual en entorno laborales que precisen de la interacción humano - robot.

En cuanto a las líneas futuras, se trabaja en mejorar las velocidades de respuesta con la evolución de los equipos empleados. También, se pretende conseguir mayor fluidez de movimiento con la depuración óptima del programa generado. Además, se pueden crear nuevos gestos a reconocer, como la cantidad de dedos extendidos de la mano, o integrar gestos predefinidos por el fabricante de LeapMotion tales como dibujar círculos en el aire, señalar con un dedo, reconocer un dedo determinado, entre otros.

REFERENCIAS

- [1] “Web oficial del producto.”: <https://www.leapmotion.com/>.(Última visita 20 octubre 2019)
- [2] “Especificaciones del producto - IRB 2400,”: https://library.e.abb.com/public/140cb722d454491f9ddb4087155ae46a/3HAC042195_PS_IRB_2400-es.pdf
(Última visita 20 octubre 2019)
- [3] “Manual del operador - RobotStudio,”: http://www.infopl.net/files/descargas/abb_robotica/infopl_net_3hac032104-005_rev_d_es_.pdf
(20 octubre 2019)
- [4] Goodrich, M. A., & Schultz, A. C. (2008). Human–robot interaction: a survey. Foundations and Trends® in Human–Computer Interaction, 1(3), 203-275.
- [5] R. R. Murphy, T. Nomura, A. Billard and J. L. Burke, "Human–Robot Interaction," in IEEE Robotics & Automation Magazine, vol. 17, no. 2, pp. 85-89, June 2010.
- [6] “Especificaciones técnicas Leap Motion.” <http://blog.showleap.com/category/leap-motion>.
(Última visita 16 septiembre2019).
- [7] “Elija un compilador de C++.” https://es.mathworks.com/help/matlab/matlab_external/choose-c-or-c-compilers.html (Última visita 16 septiembre 2019).
- [8] “Conexión Leap Motion - Matlab.”: <https://github.com/jeffsp/matleap>. (Última visita 16 septiembre 2019).
- [9] Á. R. del Nozal, Aguilar, F. G. E., & Gutierrez, D. (2018). Plataforma robótica para aprendizaje activo multidisciplinar. In Tecnología, Aprendizaje y Enseñanza de la Electrónica: Actas del XIII Congreso de Tecnología, Aprendizaje y Enseñanza de la Electrónica, Tenerife, 20-22 de junio, 2018 (pp. 348-354). Universidad de La Laguna.