

DISEÑO E IMPLEMENTACIÓN DE UN ROBOT HÍBRIDO TELEOPERADO CON SIMULINK Y UN WEB SERVER EMBEBIDO EN RASPBERRY PI

*Design and implementation of a Hybrid Robot Teleoperated with Simulink
and Embedded Web Server on Raspberry Pi*

Rosa Canelo Cervantes, Miguel Vera Masías, Ricardo Palomares Orihuela¹

PRESENTACIÓN: OCTUBRE 2016

ACEPTACIÓN: NOVIEMBRE 2016

RESUMEN

La presente investigación pretende explicar el diseño e implementación de un robot híbrido, formado por una base móvil y un brazo de 5 GDL con herramienta tipo pinza, el cual es teleoperado bajo la plataforma Simulink del Matlab y un Web Server embebido en un controlador Raspberry Pi. El controlador Raspberry Pi manipula los actuadores del robot híbrido utilizando un navegador Web Chrome, dado que se tiene embebido un Web Server. El resultado de la investigación permitirá que los estudiantes de la Escuela Profesional de Ingeniería Mecatrónica de la Universidad Ricardo Palma cuenten con una plataforma interactiva para incrementar sus conocimientos e incentivar la investigación.

Palabras clave: robot híbrido, web server, embebido, Simulink, Raspberry Pi.

ABSTRACT

This research involves the design and implementation of a hybrid robot, consists of a mobile base and an arm of 5 GDL with similar to a crimp tool, which is teleoperated under the Simulink platform of Matlab and Web Server embedded in a Raspberry Pi controller. The Raspberry Pi controller manipulates the actuators of the robot hybrid using a Web browser Chrome, since it has embedded a Web Server. The result of the research will allow students of the Professional School of Mechatronics Engineering, University Ricardo Palma have an interactive platform to increase their knowledge and encourage research.

Keywords: Hybrid Robot, web server, embedded, Simulink, Raspberry Pi.

¹ Facultad de Ingeniería, Universidad Ricardo Palma ,Lima, Perú
rosacaneloc@gmail.com, miguelvera_31@hotmail.com, rpalomares@ieee.org

1. INTRODUCCIÓN

La robótica móvil empezó a tener su apogeo industrial en los años ochenta, cuando inició sus operaciones con tareas que requerían de un operador que reemplazara la mano de obra humana en labores de alto riesgo. Se introdujeron robots que realizaban múltiples tareas como el mapeado de áreas contaminadas, la limpieza de superficies, la detección de corrosión en superficies metálicas, el transporte de material nocivo, etc. Se requirió, así, que los robots, además de realizar eficientemente su función principal, fuesen lo suficientemente autónomos e inteligentes para moverse en su campo de operación de manera segura y eficiente esquivando obstáculos y haciendo recorridos cortos en sus tareas.

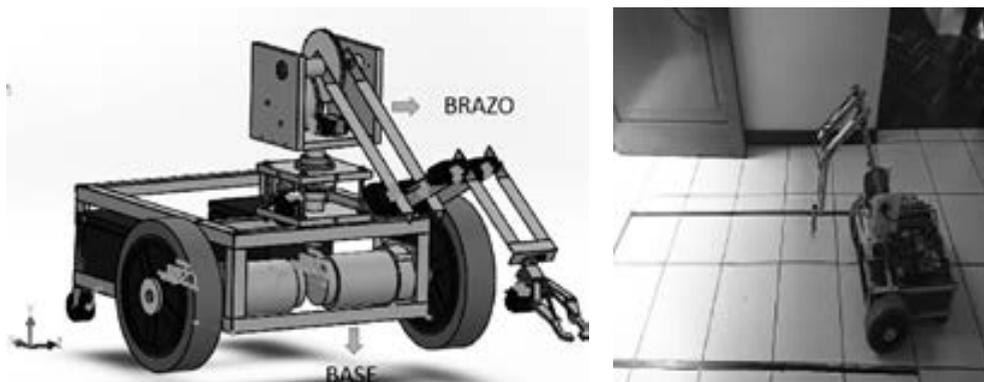
Posteriormente se diseñaron diferentes sistemas para obtener un robot autónomo y con la inteligencia para resolver problemas de desplazamiento en áreas estructuradas y no estructuradas. En la robótica móvil, se han desarrollado múltiples metodologías de diseño de sistemas de movilización eficaz y autónoma para diversas aplicaciones.

La presente investigación presenta el diseño e implementación de un robot híbrido, llamado *RaspiBot*, teleoperado para dar solución a la generación y el seguimiento de trayectorias en un entorno estructurado con obstáculos conocidos y predefinidos con la finalidad de generar la trayectoria más corta en el recorrido hacia su objetivo. Se expondrá la teoría básica de la estructura y los principios de movimiento de los robots móviles, así como el diseño e implementación del algoritmo seguido para la solución del problema. Finalmente, es necesario destacar que se realizó y documentó cada una de las pruebas experimentales del robot híbrido desarrollado.

2. DISEÑO E IMPLEMENTACIÓN DEL ROBOT HÍBRIDO

2.1. Sistema mecánico

El diseño mecánico que posee el robot híbrido le permite realizar, de forma eficaz, las tareas para las cuales fue programado. Por ende, es importante tener presente, al momento de realizar el ensamblaje de todas las piezas mecánicas que conforman al sistema total, la verificación de su correcta fijación y aseguramiento para así obtener un sistema mecánico robusto, liviano y estable. Estas piezas y mecanismos deben estar unidos con tal consistencia que permitan proteger al sistema ante cualquier ente externo que trate de afectarlo. El robot híbrido está formado por las siguientes partes:

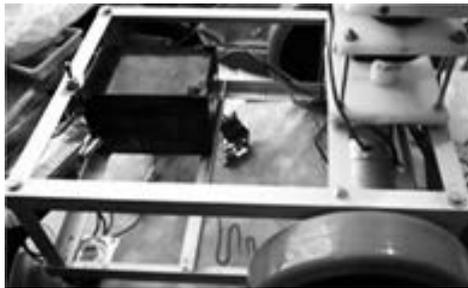


*Fig. 1. Divisiones del robot explorador RaspiBot.
Fuente: elaboración propia.*

- a) Base móvil: en esta parte se encuentran los actuadores principales, motores y baterías. Asimismo, sirve como alojamiento para las tarjetas electrónicas que controlan el sistema.
- b) Brazo: parte del robot que le permite sujetar, manipular y transportar objetos por medio de un sistema de transmisión de potencia y sus respectivos actuadores eléctricos, servomotores y motores DC.

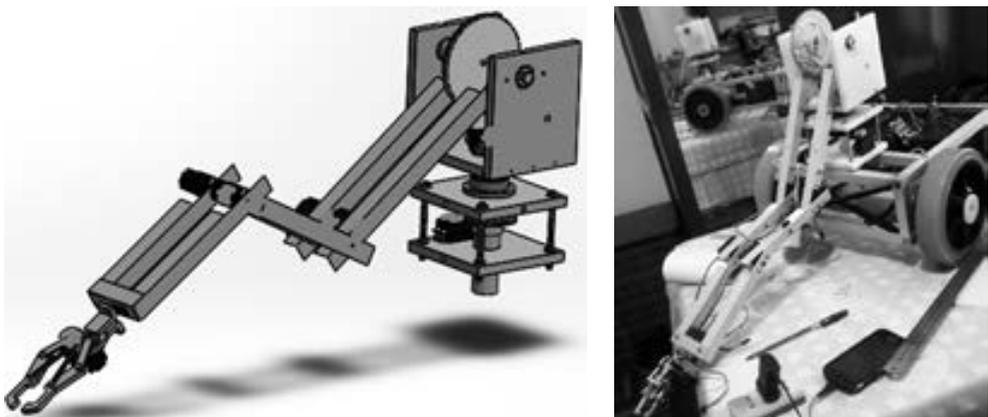
RaspiBot cuenta con un sistema mecánico formado por una base móvil y un brazo, tal como se muestra en la figura 1.

La estructura de la base del robot cuenta con el espacio suficiente para poder colocar y almacenar todos los elementos mecánicos (chumaceras, rodamientos, estructuras), eléctricos (motores, baterías, cables) y electrónicos (tarjetas de control del sistema) encargados de generar el desplazamiento y la operatividad. La figura 2 muestra la base terminada.



*Fig. 2. Base terminada con los motores instalados.
Fuente: elaboración propia.*

El brazo robótico fue fabricado de perfiles de aluminio tipo L de $\frac{3}{4}$ " x $\frac{3}{4}$ " con un espesor de 1.5 mm, en color natural, y de acuerdo con las dimensiones designadas para cada pieza. Asimismo, se cortaron planchas de polietileno de 110 x 110 mm de lado, de 10 mm de espesor, en color blanco, que servirán para sujetar los rodamientos encargados de realizar la rotación y giro de la cintura y el hombro del robot. Con las dimensiones determinadas, se diseñó el brazo con el programa Solidwork, el mismo que se muestra en la figura 3.

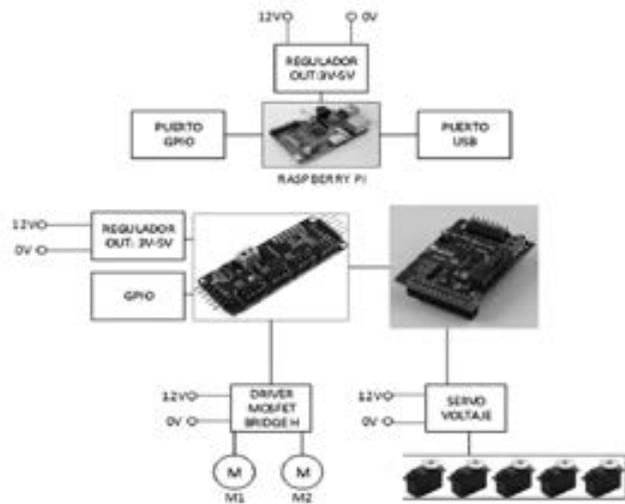


*Fig. 3. Diseño en Solidwork e implementación del brazo robótico de 5GDL.
Fuente: elaboración propia.*

2.2. Sistema electrónico

Esta etapa es una de las más importantes dentro del sistema robótico, puesto que aquí se desarrolla todo el sistema electrónico, encargado de brindar y distribuir la energía DC requerida por los dispositivos electrónicos para su operación. Por lo tanto, se debe tener en cuenta los diversos aspectos que pueden afectar el correcto funcionamiento de los circuitos electrónicos. Dentro de estos factores tenemos a los ruidos eléctricos y la cantidad de corriente máxima y mínima que debe suministrarse a los demás circuitos de control y actuadores. A continuación, se muestra un diagrama pictórico, en la figura 4, del sistema de tensiones requeridos por los circuitos y actuadores que constituyen al RaspiBot.

Fig. 4. Diagrama de distribución de tensiones en los actuadores del RaspiBot.
Fuente: elaboración propia.

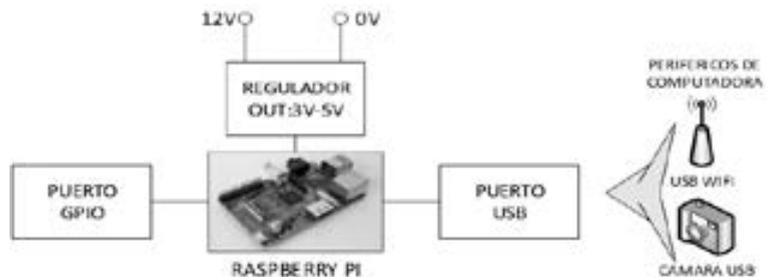


2.3. Implementación del Raspberry Pi

El Raspberry Pi, como se mencionó anteriormente, es el centro de control principal del robot, el cual se encuentra energizado por una fuente filtrada con la finalidad de evitar cualquier tipo de avería en el dispositivo. Además, se encarga de las pruebas por las que pasa el puerto GPIO para que sea capaz de llegar a los esclavos pertenecientes a los microcontroladores y sensores digitales si es el caso.

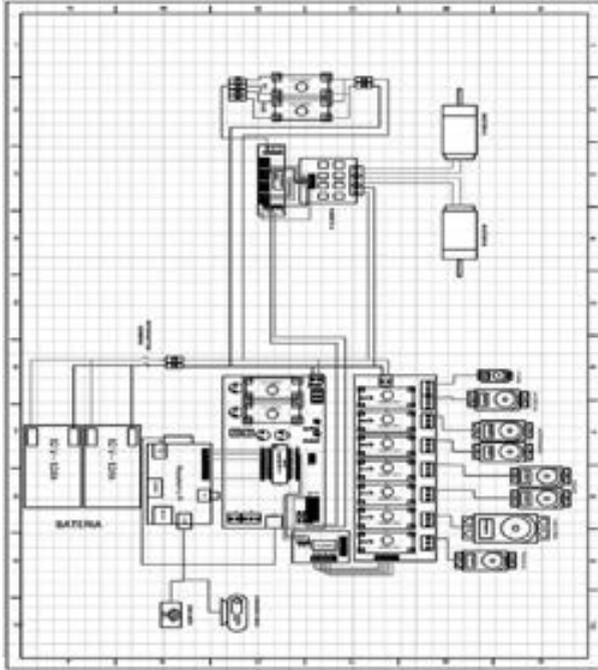
En el diagrama unifilar pictórico, se puede apreciar a los equipos o periféricos de una computadora en comunicación directa con el Raspberry Pi. En el caso de que se requiera incorporar a esta comunicación a otros dispositivos o periféricos, es necesario el uso de un HUB USB con alimentación externa. A continuación, se muestran las conexiones del Raspberry Pi a dispositivos o periféricos, tal como se muestra en la figura 5.

Fig. 5. Conexiones del Raspberry Pi.
Fuente: elaboración propia.



2.4. Sistema de suministro eléctrico

En esta etapa se detallan todos los elementos y dispositivos que conforman el sistema del robot híbrido encargado de proporcionar la energía eléctrica suficiente para que pueda desempeñarse correctamente. El cálculo total de la energía que el robot requiere se obtiene con base en el consumo eléctrico de los actuadores, sensores y tarjetas electrónicas que conformen el sistema en general. Asimismo, cabe mencionar que todos los dispositivos mencionados operan con corriente continua DC.



La mayoría de los circuitos y tarjetas electrónicas que se utilizaron funcionan a 5 VDC; los servomotores pertenecientes a la parte superior del brazo robótico operan a 5 VDC; mientras que los motoredutores de la base móvil lo hacen a 12 VDC. Con base en los voltajes de operación que se puede obtener y utilizando los reguladores de voltaje respectivos para los circuitos electrónicos y servomotores, se utilizaron las baterías

Fig. 6. Diagrama eléctrico general del RaspiBot.

Fuente: elaboración propia.

de 12 VDC. A continuación, en la figura 6, se detalla la ubicación y el recorrido del sistema de suministro eléctrico.

2.5. Interfaz de teleoperación con el Raspberry Pi

El controlador principal, el Raspberry Pi, está conformado por dos entradas USB, un puerto GPIO, una entrada para la tarjeta SD y una entrada de voltaje micro USB. Se instaló el sistema operativo Raspbian para poder programar mediante la herramienta Simulink de Matlab v2014. Asimismo, se cuenta con todas las librerías del puerto GPIO instaladas. Terminada la instalación del sistema operativo Raspbian de Linux, provisto por el software Matlab, se procede a encender el interruptor del robot y a conectar un cable de red *Cross Over* para la comunicación vía Ethernet. Para hacer posible la comunicación vía Ethernet, la computadora (o *workstation*) en donde está instalado el Matlab, debe estar en la misma red, ya que así se podrá obtener para el Raspberry Pi una asignación de IP por DHCP, es decir, de forma automática.

Se ha ingresado al escritorio remoto para realizar la configuración del USB WiFi que se ha instalado al Raspberry Pi, de manera que pueda ser conectado inalámbricamente a la red local y tener un

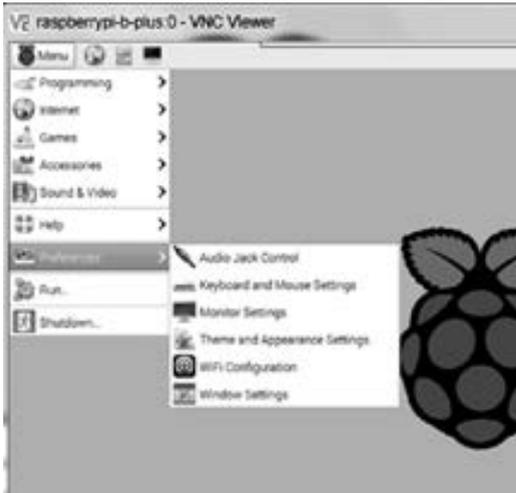


Fig. 7. Configuración del USB WiFi en el escritorio remoto Raspberry Pi.
Fuente: elaboración propia.



Fig. 8. Ingreso de la dirección IP del Raspberry Pi en el navegador web.
Fuente: elaboración propia.



Fig. 9. Ingresando dirección IP del Raspberry Pi a un navegador web.
Fuente: elaboración propia.

acceso directo a internet sin contar como intermediario a la *workstation* por cable de red, tal como se muestra en la figura 7.

2.6. Programación del servidor web

Existen numerosas aplicaciones que facilitan la instalación automática de servidores web Apache y aplicaciones adicionales como MySQL y PHP (entre otros), de forma conjunta, como XAMPP, JAMP o EasyPHP. Estas aplicaciones reciben el nombre de LAMP cuando se instalan en plataformas Linux; WAMP, en sistemas Windows; y MAMP, en sistemas Apple Macintosh. Para implementar el servidor web en el Raspberry Pi, se instaló la versión 2 de Apache y PHP versión 5.

Ahora, el Raspberry Pi es un servidor web local. Se puede ingresar a la página desde un navegador web cualquiera. Se procede a ingresar la dirección IP con la cual está conectado el Raspberry Pi a la red local (ver figura 8).

Se ingresa mediante un navegador web utilizando la dirección IP del Raspberry Pi y se procede a visualizar en pantalla lo enfocado por la cámara USB conectada al Raspberry Pi, tal como se muestra en la figura 9.

2.7. Página web para la teleoperación de Raspibot

Se describe la creación de la página web para la teleoperación del robot. La estructura de la página web está compuesta por los módulos control y servo (ver figura 10). Estos reflejan el resultado de sus procesos en el archivo raíz de la aplicación. Este último es el encargado de mostrar al usuario los controles mediante los cuales se interactúa con la aplicación, en otras palabras, la interfaz gráfica de usuario (GUI del inglés *Graphical User Interface*).



Fig. 10. Entorno de trabajo de la página web para la teleoperación del robot.
Fuente: elaboración propia.

2.8. Programación con Simulink

Simulink es un entorno de programación visual que funciona sobre el entorno de Matlab, la consola en la *workstation*. Ya que es un entorno de programación de alto nivel, podrá generar y compilar los códigos necesarios para que se ejecuten en el Raspberry Pi. Debido a que se usa la aplicación Linux remotamente desde la *workstation*, se debe definir los parámetros de la función "Raspberry". Esta función trabaja directamente con el software Simulink, muy diferente a la función "raspi", que trabaja directamente con Matlab.

Simulink posee definidos a los bloques de programación para el Raspberry Pi; además, posee diez objetos en su librería y programaciones de ejemplo por si se desea poner a prueba al Raspberry Pi. En la figura 11 se muestra la librería del Raspberry Pi en Simulink.



Fig. 11. Librería del Raspberry Pi en Simulink.
Fuente: elaboración propia.

En el entorno de Simulink, se usa los bloques V4L2 Video Capture para hacer un procesamiento de señal, y un SDL Video Display para visualizar en la *workstation* el procesamiento de imágenes con Simulink. En la figura 12, se muestra a los objetos para el reconocimiento de las imágenes.

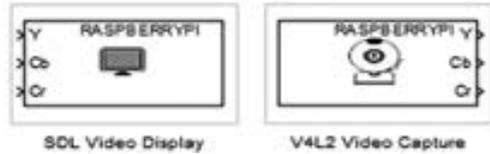


Fig. 12. Objetos para el reconocimiento de imágenes
Fuente: elaboración propia

Se elabora la primera programación en Simulink con dos bloques de la librería del Raspberry PI, uno que captura imágenes y un visualizador. Entre la conexión de estos dos habrá una función de Matlab que realice el procesamiento de imágenes (ver figura 13).

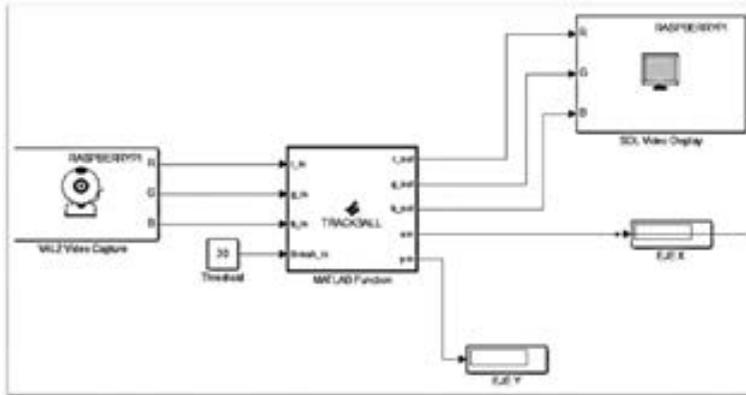


Fig. 13. Programación por bloques en Simulink para RaspiBot.
Fuente: elaboración propia.

Ahora se obtiene de forma automática el visualizador “SDL Video Display” en donde se pueden apreciar los primeros resultados de la programación realizada. Para esto, se presenta una esfera de color verde, que posee, en el centro dibujado por software, un cuadrado rojo que indica su ubicación en el plano XY (ver figura 14). Para tener un punto más exacto de estas coordenadas, se tiene como ayuda a los *display* que visualizan a cada uno de los datos.



Fig. 14. Visualizador de video reconociendo el color verde y dibujando un puntero rojo en el objeto.
Fuente: elaboración propia.

Al final se tiene la programación completa en Simulink para el reconocimiento de objetos de color verde a través de RaspiBot, tal como se muestra en la figura 15.

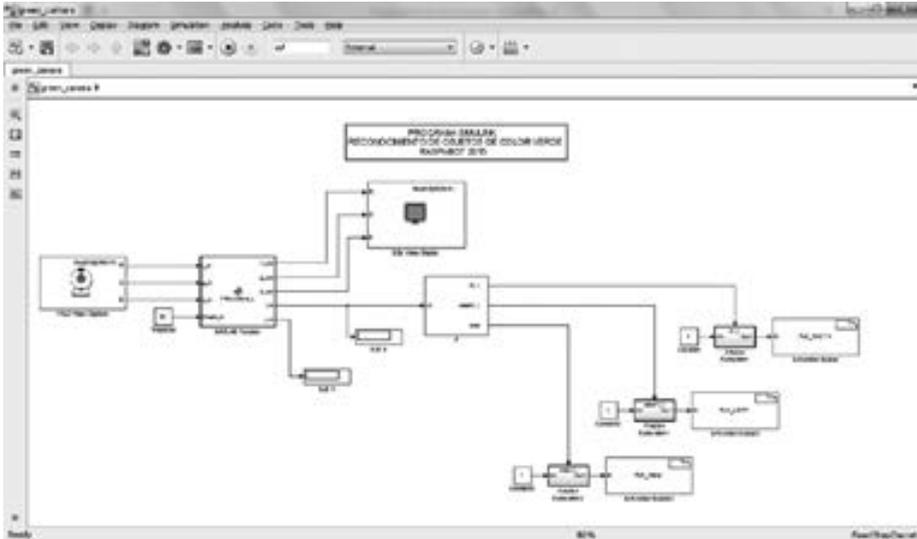


Fig. 15. Programación de RaspiBot con Simulink.
Fuente: elaboración propia.

Se obtiene los resultados mostrando las imágenes transmitidas por la cámara web a través del Web Server, lo cual se muestran en la figura 16.

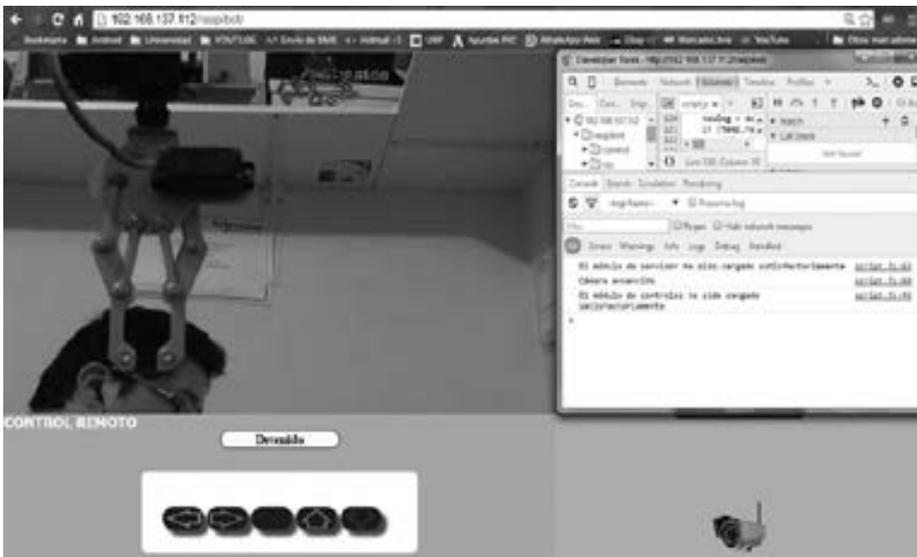


Fig. 16. Web Server con la cámara encendida.
Fuente: elaboración propia.

3. CONCLUSIONES

Ha sido posible el diseño e implementación de un robot híbrido teleoperado con Simulink y un Web server embebido en Raspberry Pi para el Laboratorio de Ingeniería Mecatrónica de la Universidad Ricardo Palma.

El diseño de la estructura mecánica permite un fácil ensamblaje, con la finalidad de que el estudiante pueda realizar modificaciones y mejoras maximizando así su capacidad intelectual en diseño mecánico.

El diseño del sistema eléctrico y electrónico ha permitido que el robot híbrido cuente con una autonomía de funcionamiento que le permita realizar de manera eficiente las actividades planteadas en la presente investigación.

El diseño del sistema de teleoperación permite integrar la programación maestra del Webserver con la aplicación del Simulink a través de comandos creados por PHP en Linux.

4. OBSERVACIONES Y RECOMENDACIONES

El diseño en dos sistemas de programación permitió determinar que el Simulink posee una latencia muy alta al trabajar con una Workstation, dado que, al tener una forma de comunicación bajo un protocolo TCP, los datos siempre van a tener que confirmarse para no llegar con problemas.

El web Server presenta latencia al ejecutar los comandos de teleoperación, y son menores que los tiempos de latencia del Simulink.

Es posible importar el entorno de programación a una placa embebida con mejores características, tal como el Raspberry PI b+ y el Raspberry PI 2, que son 6 veces más rápidos que el Raspberry PI utilizado en la presente investigación.

La comunicación vía Wifi permitió conectividad directa con un router de internet, dado que es recomendable mantener actualizado el sistema operativo del robot híbrido.

El robot híbrido RaspiBot podrá ser mejorado dado que sistema de programación es libre y su dispositivo de control tiene soporte técnico a nivel mundial de forma gratuita.

5. REFERENCIAS

- [1] Ángeles, J. (2007). *Fundamentals of Robotic Mechanical Systems. Theory, Methods and Algorithms*. Nueva York, USA: Springer-Verlag.
- [2] Aracil, R., Salterén, R., Sabaster, J., y Reinoso, O. (2006). Robots Paralelos: Máquinas con un pasado para una robótica del futuro. *Revista Iberoamericana de Automática e Informática Industrial*, 3, pp. 16-28.
- [3] Barrientos, A. (2007). *Fundamentos de robótica*. España: McGraw-Hill/ Interamericana de España.

- [4] Crespo Vargas, E. M. (2015). *Diseño e implementación de un dispositivo conversor con recepción Usb, Ethernet y RS- 232 a enviar mediante Ethernet, Wi-Fi y Usb con una tarjeta Raspberry Pi; para el monitoreo de los equipos del Laboratorio de circuitos electrónicos de la Universidad de las Fuerzas Armadas-ESPE Extensión Latacunga*. Sangolquí, Ecuador: Universidad de las Fuerzas Armadas.
- [5] González, J y Ollero, A. (1996). Estimación de la posición de un robot móvil. *Informática y Automática*, 29 (4), pp. 3-18.
- [6] Kadhim Al Ansari, S. H. (2011). *Metodología para el desarrollo de un manipulador móvil autónomo con características antropométricas*. Universidad Carlos III de Madrid.
- [7] Lam, S., y Parikh, P. (2005). *A Hybrid Strategy to Solve the Forward Kinematics Problem in Parallel Manipulators*. Nueva York, USA: IEEE Transactions on Robotics.
- [8] Lozano-Pérez, T. (1990). *Foreword: Mobile Robot and Robotics. Autonomous Robot Vehicles*. New York, USA: Springer-Verlag.
- [9] Muir, P. (1998). *Modeling and control of wheeled mobile robots. The Robotics Institute*. Pittsburgh: Carnegie Mellon University.
- [10] Siegwart, R. y Nourbakhsh, R. (2004). *Introduction to Autonomous Mobile Robots*. London: The MIT Press.