

# **Desarrollo de un brazo robot de tipo esférico**

## ***Development of a spherical configuration robot arm***

**Jesús Medina Cervantes**

Facultad de Ingeniería, Universidad Veracruzana, México

[jemedina@uv.mx](mailto:jemedina@uv.mx)

**Rubén Villafuerte Díaz**

Facultad de Ingeniería, Universidad Veracruzana, México

[rvillafuerte@uv.mx](mailto:rvillafuerte@uv.mx)

**Victorino Juárez Rivera**

Facultad de Ingeniería, Universidad Veracruzana, México

[vijuarez@uv.mx](mailto:vijuarez@uv.mx)

### **Resumen**

En este artículo se presenta el desarrollo de un brazo robot de configuración esférica, con la finalidad apoyar el aprendizaje de estudiantes de Ingeniería en temas relacionados con los sistemas mecatrónicos y la robótica. Con este prototipo los estudiantes podrán conocer los alcances y limitaciones de la cinemática de esta clase de brazo robot. El brazo robot que se presenta en este artículo cuenta con 4 grados de libertad, de los cuales dos son controlados mediante servomotores y los otros dos son controlados utilizando motores a pasos.

La estructura del brazo robot fue diseñada por completo empleando el software de diseño SolidWorks, de esta manera se contó con la forma y dimensiones de cada uno de los elementos de la estructura del brazo robot, lo cual facilitó su construcción.

Para controlar los movimientos de las articulaciones del brazo robot se empleó la tarjeta de desarrollo Arduino Mega 2560, la cual fue programada en lenguaje C mediante el entorno de desarrollo de Arduino.

El usuario puede controlar al brazo robot de manera inalámbrica utilizando una interfaz gráfica, la cual fue programada en Matlab. La comunicación inalámbrica entre la computadora y la tarjeta

Arduino Mega se realiza utilizando el protocolo de comunicación RS-232, mediante un módulo bluetooth.

Este prototipo permite a los estudiantes observar de manera real el funcionamiento de esta clase de brazos robóticos, así como la forma de programarlos y controlarlos.

**Palabras clave:** Robot de configuración esférica, motor a pasos, servomotor, Arduino, Matlab.

## **Abstract**

In this paper the development of a spherical robot arm is presented, with the goal to support engineering students in learning topics related to mechatronics and robotics. With this prototype, students will meet the kinematics capabilities and limitations of this class of robot arm. The robot arm presented in this article has 4 degree of freedom, which two of them are servomotor controlled and the other two are step motor controlled.

The robot arm structure was completely designed using SolidWorks design software, so that every component shape and dimensions are known to make easier its construction.

Arduino Mega 2560 development board was used to control the movements of all the robot arm articulations, which was programmed in C language by the Arduino software.

Users can control the robot arm wirelessly using a graphical interface which was programmed in Matlab software. Wireless communication between the PC and the Arduino Mega board is established by the RS-232 communication protocol, through a Bluetooth module.

This prototype lets students to watch and learn how this class of robot works and the way to program and control them.

**Keywords:** Spherical robot arm, step motor, servomotor, Arduino, Matlab.

**Fecha Recepción:** Diciembre 2017

**Fecha Aceptación:** Julio 2018

---

## Introducción

El robot de configuración esférica se caracteriza por tener dos articulaciones de rotación y una prismática (RRP), ver figura 1. Su área de trabajo es una esfera hueca y su efector final no alcanza todos los puntos espaciales dentro del volumen descrito, ya que esto depende de la longitud del elemento de la articulación prismática, también llamado brazo extensor. Esta clase de robots tiene aplicación en operaciones de manipulación y de fabricación; se utilizan para cargar maquinaria, en fundición de metal, prensa de troquel, forja y soldadura, por mencionar algunas. El prototipo que se presenta en este artículo tiene cuatro grados de libertad (RRPR), lo cual permite mantener fija la inclinación de la herramienta cuando esto se requiere.

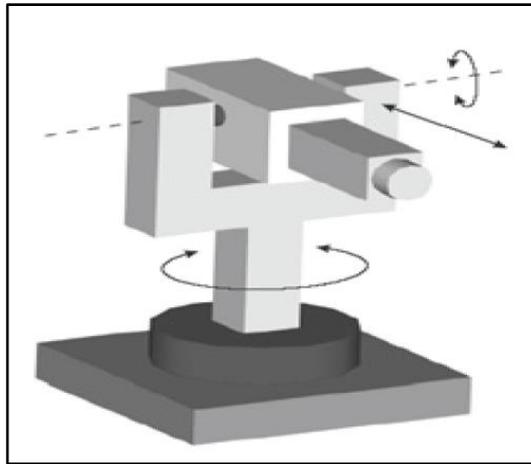


Figura 1. Robot de configuración esférica (Reyes, 2012).

## CINEMÁTICA DEL ROBOT

La cinemática directa de un robot se obtiene empleando el bien conocido algoritmo de Denavit-Hartenberg. Mediante transformaciones de rotación y traslación se pueden conocer las coordenadas cartesianas ( $X_o, Y_o, Z_o$ ) de la herramienta o efector final, referido a un sistema de coordenadas fijo. Existen numerosas fuentes que explican a detalle el algoritmo de Denavit-Hartenberg, entre ellas se encuentran (Spong & Vidyasagar, 1989), (Craig, 2006), (Appin Knowledge Solutions, 2007) y (Reyes, 2012).

Como ya se mencionó, el robot de tipo esférico que se desarrolla en este proyecto tiene cuatro grados de libertad (RRPR). En la figura 2 se presenta una vista lateral del robot, en la cual se muestran los parámetros empleados para su modelo cinemático. Cabe mencionar que los parámetros  $d_2$  y  $d_4$  no se muestran en la figura, pues corresponden a distancias en dirección del

eje x del servomotor para el brazo extensor y el servomotor de la muñeca del robot, respectivamente. En dicho esquema, el pequeño círculo en color amarillo representa la posición espacial de la herramienta con coordenadas cartesianas  $(X_0, Y_0, Z_0)$ , con respecto del origen de coordenadas.

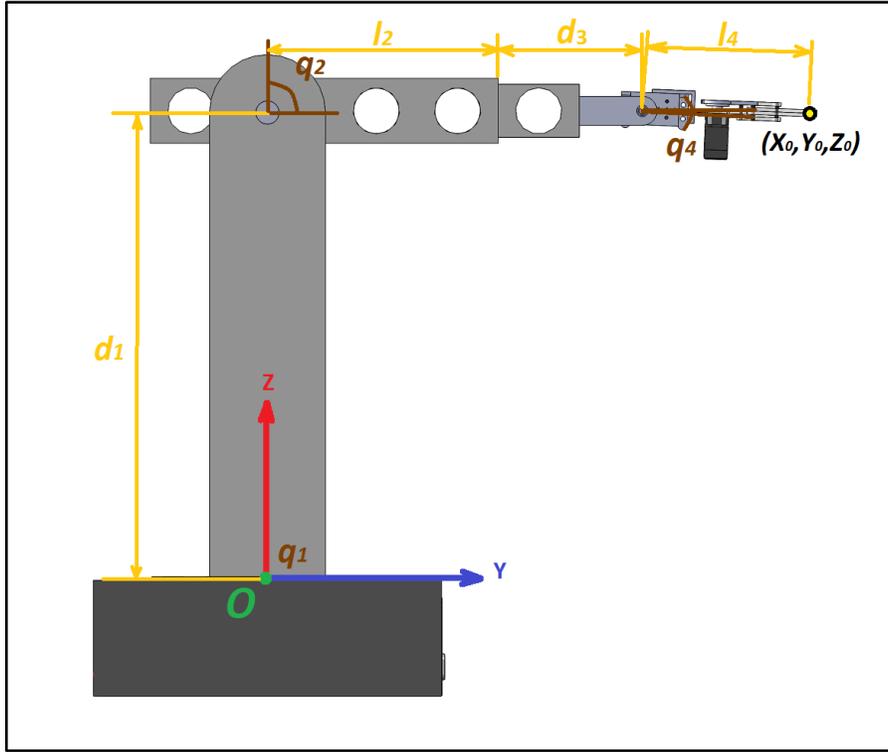


Figura 2. Parámetros del robot utilizados para obtener su modelo cinemático.

En la tabla 1 se presentan los parámetros a utilizar para obtener el modelo cinemático directo al aplicar el algoritmo de Denavit-Hartenberg.

Tabla 1. Parámetros de los eslabones del robot de tipo esférico.

Eslabón	$l_i$	$\alpha_i$	$d_i$	$\theta_i$
1	0	$\pi/2$	$d_1$	$q_1$
2	$l_2$	0	$d_2$	$q_2$
3	0	0	$d_3$	0
4	$l_4$	0	$d_4$	$q_4$

Con los parámetros de configuración del robot de tipo esférico se obtienen las matrices de transformación homogéneas para cada uno de los elementos.

Matriz de transformación de la articulación de la Base (R):

$${}^0T_1 = \begin{bmatrix} \cos(q_1) & 0 & \text{sen}(q_1) & 0 \\ \text{sen}(q_1) & 0 & -\cos(q_1) & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Matriz de transformación de la articulación del brazo extensor (R):

$${}^1T_2 = \begin{bmatrix} \cos(q_2) & -\text{sen}(q_2) & 0 & l_2 * \cos(q_2) \\ \text{sen}(q_2) & \cos(q_2) & 0 & l_2 * \text{sen}(q_2) \\ 0 & 0 & 1 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Matriz de transformación de la articulación prismática (P):

$${}^2T_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Matriz de transformación de la articulación de la muñeca (R):

$${}^3T_4 = \begin{bmatrix} \cos(q_4) & -\text{sen}(q_4) & 0 & l_4 * \cos(q_4) \\ \text{sen}(q_4) & \cos(q_4) & 0 & l_4 * \text{sen}(q_4) \\ 0 & 0 & 1 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Se multiplican las matrices para obtener la matriz de transformación homogénea  ${}^0T_4$ , en la cual los elementos  $(T_{14}, T_{24}, T_{34})$  corresponden a las coordenadas cartesianas  $(X_0, Y_0, Z_0)$  del extremo final del robot.

$$X_0 = T_{14} = (d_2 + d_3 + d_4) * \text{sen}(q_1) + [l_2 + l_4 * \cos(q_4)] * \cos(q_1) * \cos(q_2) - l_4 * \cos(q_1) * \text{sen}(q_2) * \text{sen}(q_4)$$

$$Y_0 = T_{24} = (-d_2 - d_3 - d_4) * \cos(q_1) + [l_2 + l_4 * \cos(q_4)] * \cos(q_2) * \text{sen}(q_1) - l_4 * \text{sen}(q_1) * \text{sen}(q_2) * \text{sen}(q_4)$$

$$Z_0 = T_{34} = d_1 + l_2 * \text{sen}(q_2) + l_4 * [\cos(q_2) * \text{sen}(q_4) + \cos(q_4) * \text{sen}(q_2)]$$

## **DISEÑO DEL ROBOT**

El robot fue diseñado por completo utilizando el software de diseño SolidWorks (Dassault Systèmes, 2016). Se diseñaron cada uno de los componentes del brazo robot de forma separada. Para ello, se empleó el modo “Pieza” de SolidWorks. Los modelos tridimensionales de los componentes se guardan en archivos con extensión \*.sldprt. En la figura 3 se presenta un ejemplo del entorno de trabajo de SolidWorks.

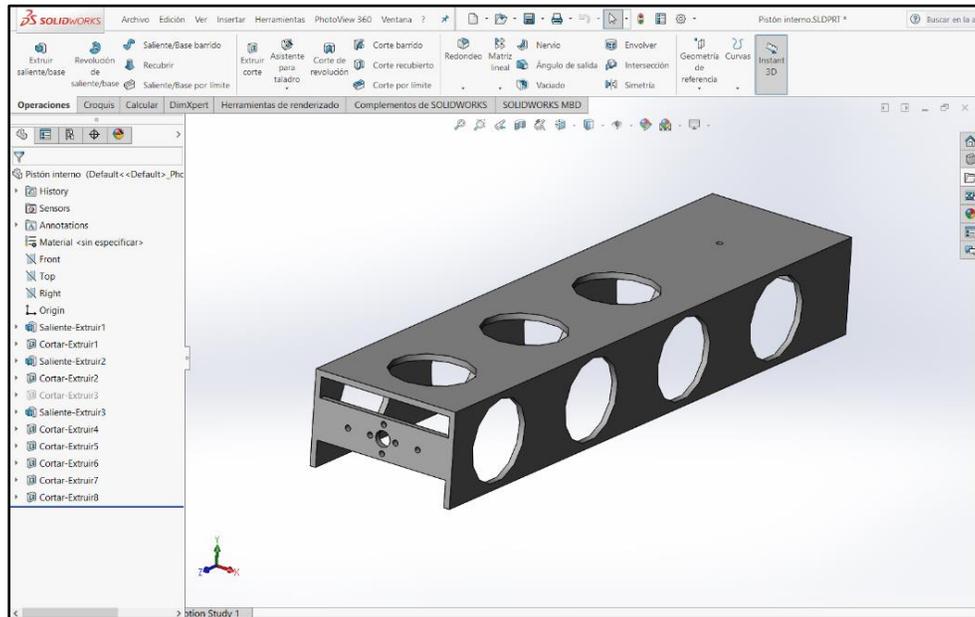


Figura 3. Creación de un componente en el modo “Pieza” de SolidWorks.

Una vez que se han diseñado todas las partes, se procede a ensamblarlas para formar la estructura completa del robot. Para realizar esta tarea se utiliza el modo “Ensamblaje” de SolidWorks. Es necesario establecer las restricciones de movimiento adecuadas, lo cual se realiza con la herramienta “Relación de posición”. El ensamblaje guarda en un archivo con extensión \*.sldasm.

En la figura 4 se presenta la imagen renderizada del brazo robot de tipo esférico diseñado.

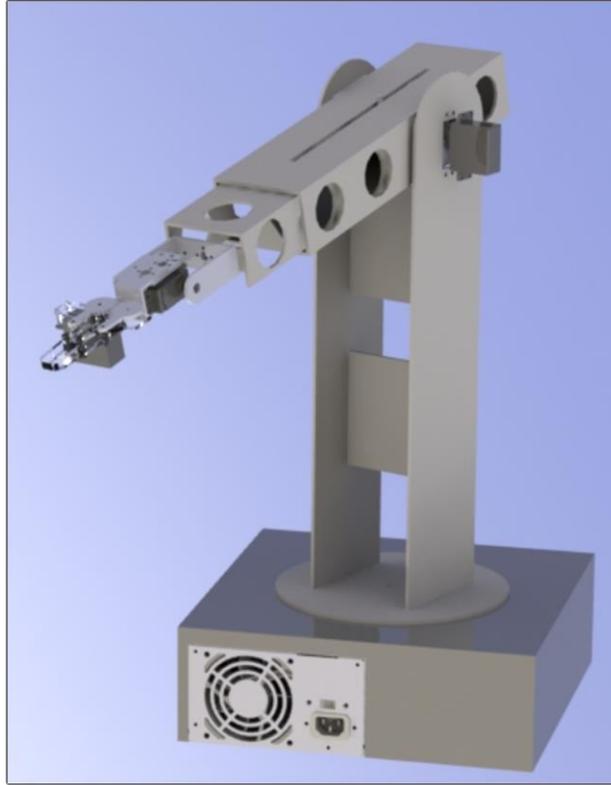


Figura 4. Modelo 3D del brazo robot de tipo esférico.

### **CONSTRUCCIÓN DEL ROBOT**

Para la construcción del robot de tipo esférico se empleó placa de Aluminio de 2mm de espesor para toda la estructura. Para la base del robot se empleó lámina de acero. Para elaborar los componentes metálicos se utilizaron máquinas herramientas como torno, taladro, máquina cortadora y máquina para soldar. Para unir las partes de Aluminio se empleó adhesivo especial. También se emplearon tuercas y tornillos para unir los servomotores, las partes de la pinza del robot, el ventilador y la fuente de alimentación.

En la figura 5 se presenta una imagen fotográfica del brazo robot completo.



Figura 5. Prototipo del robot de configuración esférica.

Enseguida se presenta, en la tabla 2, una lista con el resto de los componentes mecánicos y electrónicos que conforman al brazo robot.

Tabla 2. Elementos componentes del robot

Cantidad	Componente	Imagen
1	Servomotor CYS S8503	
2	Servomotor Tower Pro MG995	
1	Ventilador de 12V	

2	Motor a pasos Nema 17	
1	Driver TB6560	
1	Easy driver A3967	
1	Arduino Mega 2560	
1	Módulo Bluetooth HC-05	
1	Tornillo ACME y tuerca de bronce	
4	Rodamiento de bolas	
1	Cople flexible	
2	Base para motor a pasos	
1	Fuente de alimentación	

## CONTROL DEL ROBOT

Para controlar al brazo robot, se decidió emplear la tarjeta de desarrollo Arduino Mega 2560. La tarjeta se encarga de recibir y gestionar las órdenes del usuario a través de conexión inalámbrica por Bluetooth. A partir de ello, el Arduino genera los pulsos PWM para controlar los 3 servomotores y de generar los pulsos para las 2 tarjetas controladoras de los motores a pasos. En la figura 6 se muestra un diagrama de conexiones de todos los elementos de control.

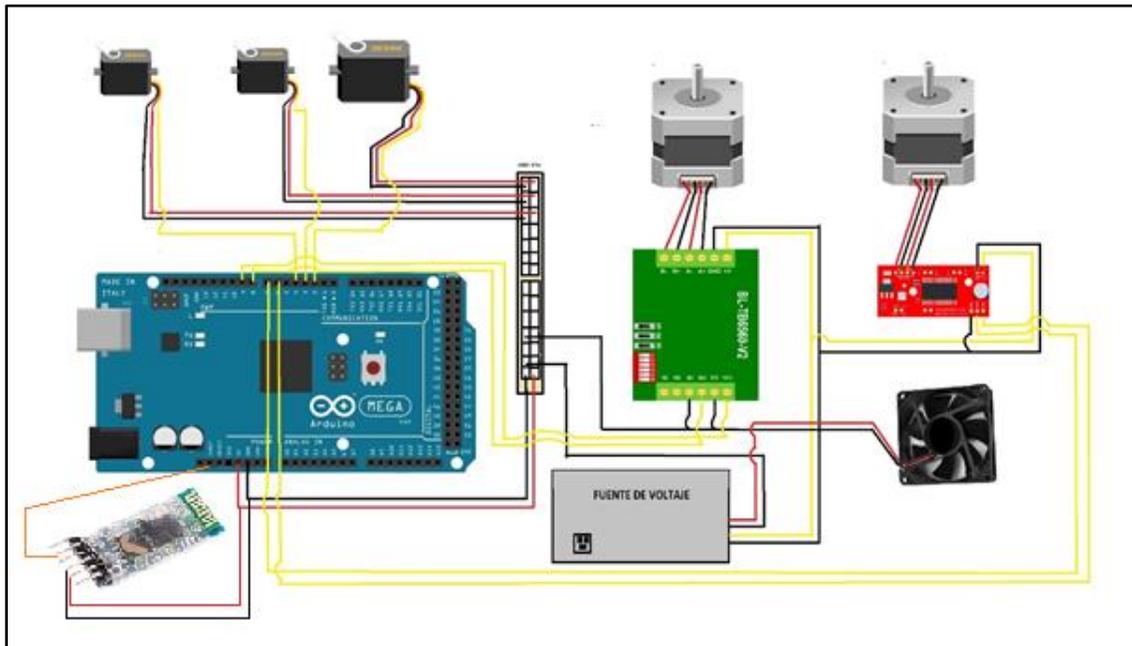


Figura 6. Esquema del circuito electrónico de control.

En la figura 7 se muestra un diagrama de flujo que explica como gestiona los datos la tarjeta Arduino Mega 2560 para realizar el control de todos los actuadores del robot.

Para que el usuario pueda controlar al robot de tipo esférico, se desarrolló una interfaz gráfica en Matlab. La interfaz cuenta con botones y barras de desplazamiento para mover todas las articulaciones del brazo robot. Además, la interfaz incluye la opción de crear programas de rutinas de movimiento del robot. Dichas rutinas se almacenan como archivos de texto, que después puede leer la interfaz nuevamente para que el robot se mueva de manera automática.

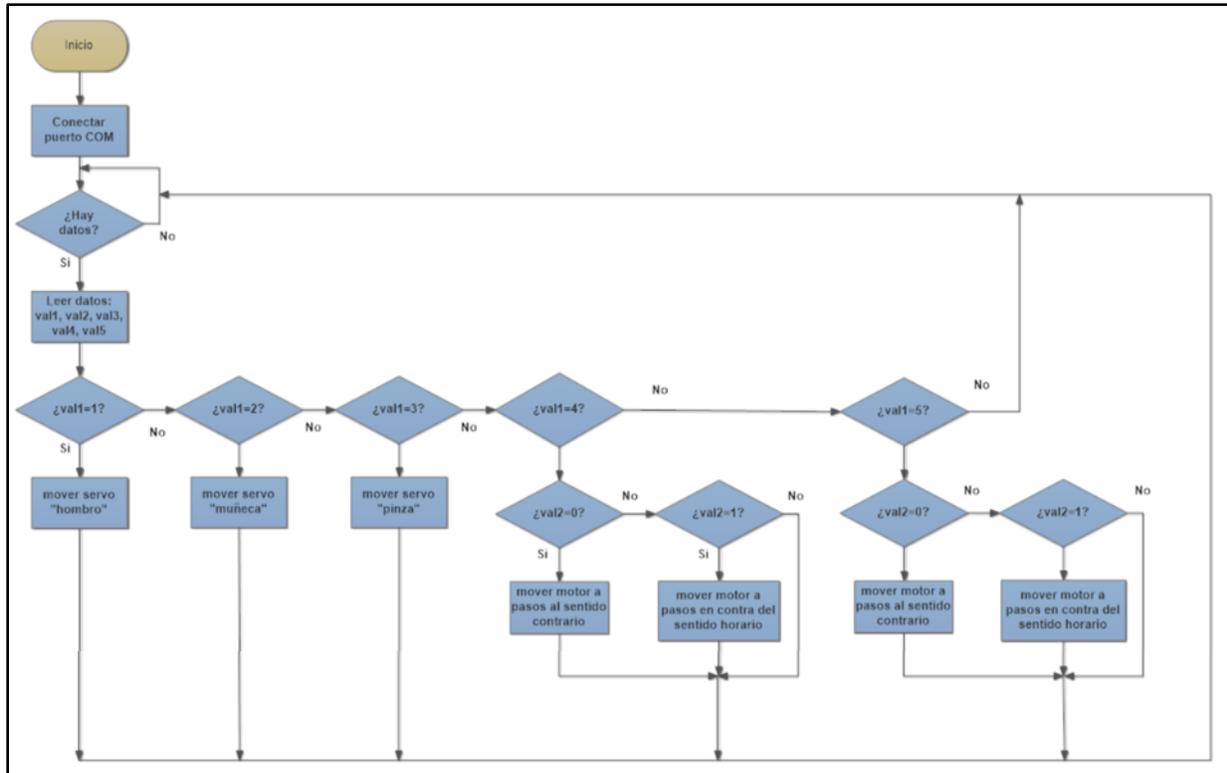


Figura 7. Diagrama de flujo de la gestión de datos y control de Arduino Mega 2560.

En la figura 8 se muestra una imagen de la interfaz gráfica desarrollada. Una vez encendido el brazo robot, el usuario debe iniciar la conexión Bluetooth entre la PC y el brazo robot. Ya establecida dicha conexión, el usuario puede inmediatamente controlar las articulaciones del brazo robot, elegir programar una rutina o ponerlo en modo automático.

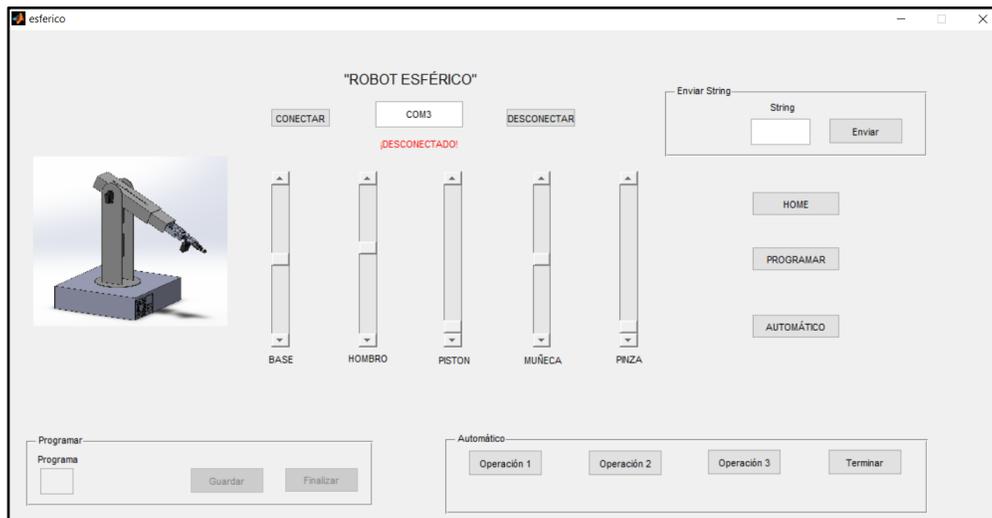


Figura 8. Interfaz gráfica de usuario para el control del robot.



## Referencias

Angeles, J. (2003). *Fundamentals of robotic mechanical systems, Theory, Methods and Algorithms*.

Montreal: 2003.

Appin Knowledge Solutions. (2007). *Robotics*. Hingham: Infinity Science Press LLC.

Bergren, C. (2003). *Anatomy of a robot*. New York: McGraw-Hill.

Bishop, R. (2002). *The mechatronics handbook*. New York: CRC press.

Bolton, W. (2002). *Mecatrónica, sistemas de control electrónico en ingeniería mecánica y eléctrica*. México: Alfaomega.

Craig, J. (2006). *Robótica, 3a. Edición*. México: Pearson Educación.

Dassault Systèmes. (22 de Enero de 2016). *SolidWorks*. Obtenido de <http://www.solidworks.com/>

Gu, E. (2013). *A journey from robot to digital human, mathematical principles and applications with MATLAB programming*. Berlin: Springer.

Kurfess, T. (2005). *Robotics and automation handbook*. New York: CRC Press.

Marchand, P., & Holland, T. (2003). *Graphics and GUIs with MATLAB*. New York: Chapman & Hall / CRC press.

Margolis, M. (2012). *Arduino Cookbook*. New York: O'Reilly.

Reyes, F. (2012). *MATLAB Aplicado a Robótica y Mecatrónica*. México: Alfaomega Grupo Editor, S.A. de C.V.

Schmidt, M. (2011). *Arduino, A quick-start guide*. Raleigh: The pragmatic bookshelf.

Selig, J. M. (1992). *Introductory robotics*. New York: Prentice Hall.

Siciliano, B., & Khatib, O. (2008). *Sringer handbook of robotics*. Berlin: Springer.

Spong, M., & Vidyasagar, M. (1989). *Robot dynamics and control*. New York: John Wiley & Sons, Inc.

## **INFORMACIÓN DE CV DE AUTORES**

El M.C. Jesús Medina Cervantes es profesor de tiempo completo adscrito a la Facultad de Ingeniería, Campus Ciudad Mendoza, Universidad Veracruzana. Es ingeniero mecánico egresado del Instituto Tecnológico de Orizaba en 1998 y en 2002 obtiene el grado de maestría en Ingeniería Mecánica en el Centro Nacional de Investigación y Desarrollo Tecnológico (Cenidet). Actualmente es coordinador del Programa de tutorías y miembro activo del Cuerpo Académico “Investigación en Ingeniería Aplicada” UV CA – 318.

El Dr. Rubén Villafuerte Díaz es profesor de tiempo completo adscrito a la Facultad de Ingeniería, Campus Ciudad Mendoza, Universidad Veracruzana. Es ingeniero industrial electricista, egresado del Instituto Tecnológico de Morelia en 1979. Realizó estudios de maestría y doctorado en el área de Ingeniería Eléctrica en el IPN, recibiendo los grados correspondientes en 1988 y 1993, respectivamente. Actualmente es coordinador de la carrera de Ingeniería Eléctrica y miembro activo del Cuerpo Académico “Investigación en Ingeniería Aplicada” UV CA – 318.

El Dr. Victorino Juárez Rivera es profesor de tiempo completo adscrito a la Facultad de Ingeniería, Campus Ciudad Mendoza, Universidad Veracruzana. Es ingeniero industrial egresado del Instituto Tecnológico de Orizaba en 2004, en 2007 obtiene el grado de maestría en ciencias en Ingeniería Mecánica en el Instituto Tecnológico de Veracruz y en 2012 obtiene el grado de Doctor en Educación en la Escuela libre de Ciencias Políticas y Administración Pública. Actualmente es coordinador de la carrera de Ingeniería Industrial, coordinador de movilidad estudiantil y miembro activo del Cuerpo Académico “Investigación en Ingeniería Aplicada” UV CA – 318.