

Plataforma didáctica para robótica articulada por medio de una interfaz gráfica

Didactic platform for articulated robotics by means of a graphical interface

Darvi Echeverría Sosa¹ , Alejo Mosso Vázquez² , Roberto Iván Dzul Duran¹ , Jimer Emir Loría Yah² , Gerardo Israel de Atocha Pech Caraveo² , Carlos Alberto Decena Chan²  y Ramiro José González Horta² 

¹Departamento de Electromecánica Tecnológico Nacional de México / Instituto Tecnológico Superior de Motul. Carretera Mérida Motul, Tablaje Catastral 383, Motul, Yucatán, México. C.P. 97430

²Departamento de Mecatrónica Tecnológico Nacional de México / Instituto Tecnológico Superior de Calkiní en el Estado de Campeche. Av. Ah Canul S/N por Carretera Federal, Calkiní, Campeche, México, CP 24900.

*Correo-e: darvi.echeverria@itsmotul.edu.mx

PALABRAS

CLAVE:

Brazo robótico,
interfaz gráfica,
control en tiempo
real, Arduino.

RESUMEN

Este artículo expone el desarrollo de una plataforma robótica didáctica. El propósito de la plataforma es realizar trabajos experimentales que permitan la ejecución y evaluación de tareas específicas relacionadas con la cinemática y control de movimiento de las articulaciones de un mecanismo robótico. Un elemento básico de la plataforma es el brazo robótico xArm-1s del fabricante *Hiwonder*, el cual posee una estructura articulada con seis servomotores y una tarjeta propia para su control mediante una tarjeta Arduino. Los estados de movimiento se monitorean por medio de una interfaz gráfica desarrollada en Matlab que permite el manejo de las variables utilizadas. La plataforma está dirigida a la enseñanza como una herramienta experimental que permita el análisis y comprensión de cada concepto de la cinemática de la robótica articulada.

KEYWORDS:

robotic arm,
graphical interface,
Arduino.

ABSTRACT

This article exposes the development of a didactic robotic platform. The purpose of the platform is to carry out experimental work that allows the execution and evaluation of specific tasks related to the kinematics and movement control of the joints of a robotic mechanism. A basic element of the platform is the robotic arm xArm-1s from the manufacturer *Hiwonder*, which has an articulated structure with six servomotors and its own card for its control through an Arduino card. The states of movement are monitored by means of a graphical interface developed in Matlab that allows the management of the variables used. The platform is aimed at teaching as an experimental tool that allows the analysis and understanding of each concept of the kinematics of articulated robotics.

• **Recibido:** 6 de agosto 2021 • **Aceptado:** 10 de enero 2022 • **Publicado en línea:** 30 de noviembre 2022

1 INTRODUCCIÓN

Las investigaciones más avanzadas en el área de la tecnología robótica son encabezadas por los países con mayor desarrollo y recursos, lo que representa un desafío para naciones como la nuestra que realiza trabajos con recursos reducidos. A fin de mejorar e incrementar los trabajos de enseñanza e investigación en el área de la mecatrónica, se propone el desarrollo de una plataforma robótica. La plataforma facilitará el trabajo de docentes, estudiantes e investigadores de las escuelas de nivel superior de nuestro país en las temáticas relacionadas con la robótica, que abarcan desde la cinemática, la dinámica, el control, el diseño de interfaces, etc.

Como se comenta en [1], La robótica es una excelente herramienta para enseñar ciencia y tecnología desde un enfoque con la metodología STEM (*Science, Technology, Engineering and Mathematics*), por lo que para enseñar la robótica es más evidente la utilización de procesos prácticos como el uso de una plataforma que cubra los aspectos más importantes para diseñar, construir o armar un mecanismo robótico.

En nuestro país la enseñanza y la robótica se relacionan principalmente en la robótica educativa que es la que se está implementando de manera rápida en muchas escuelas de nivel básico, medio básico, e incluso a nivel superior fomentando el modelo educativo STEM como se puede apreciar en [2]. En lo que respecta a la enseñanza de la robótica a nivel superior, ésta se enfoca en desarrollar proyectos donde los estudiantes construyen robots pequeños con aplicaciones específicas utilizando tarjetas de bajo costo y con entornos de desarrollo gratuito.

Existe una gran diversidad de opciones para adentrarse en la robótica, en todos los niveles educativos, desde el preescolar, primaria, secundaria, bachillerato y universidad. Para cada nivel se puede encontrar una multitud de cursos, dispositivos, lenguajes de programación, precios, formas y aplicaciones de dispositivos robóticos que se utilizan para iniciarse en el mundo de la robótica.

Una vez visualizada esta necesidad de utilizar la robótica en los aspectos educativos se desarrolla una plataforma robótica que aporte en el ámbito de la educación superior proporcionando a los estudiantes y

docentes un medio para realizar prácticas relacionadas con la cinemática y el movimiento de las articulaciones de un robot. La aportación de este trabajo es el desarrollo de una interfaz gráfica que se integra a la plataforma mecánica comercial y sirve para visualizar las variables cinemáticas que se miden en las articulaciones.

La plataforma está formada por dos partes la primera incluye el mecanismo xArm 1s disponible en el mercado que usa una tarjeta de control propia y posee varios métodos de comunicación. La segunda parte consiste en el desarrollo de una interface de comunicación entre el xArm 1s y una computadora, para lograrlo se desarrolló la comunicación por medio de la programación de una tarjeta Arduino Uno y para la interfaz gráfica se elaboró un programa en Matlab.

Los beneficios que esta plataforma aporta son: primero, se tiene una interfaz gráfica basada en Matlab para el monitoreo en tiempo real de los estados del mecanismo robótico, lo que permite el desarrollo y comprensión de la cinemática directa del mismo. En segundo plano se puede usar como herramienta práctica para la comprensión de la cinemática de las articulaciones de un robot. El estudiante podrá tener la experiencia de ensamblar toda la estructura electromecánica del mecanismo, los servomotores y la tarjeta de control; podrá, integrar la comunicación mediante una tarjeta Arduino Uno y enlazarla a la interfaz gráfica en Matlab. Al final el estudiante habrá concluido con todo un proceso de desarrollo mecatrónico. Un tercer aspecto de este trabajo consiste en que puede ser empleado para que los estudiantes profundicen en el estudio de movimiento del mecanismo, creando interfaces propias para programar movimientos de las articulaciones.

2 DESARROLLO

La plataforma está formada por tres partes, a saber: (1) El mecanismo robótico, (2) el hardware electrónico para el control de los movimientos y (3) el software para la interfaz gráfica de usuario (GUI – *Graphical User Interface*). Estos elementos permiten el correcto funcionamiento y operación del dispositivo. Ahora se

presenta el desarrollo de cada una de estas etapas en el orden que fueron ejecutadas.

2.1 EL MECANISMO ROBÓTICO.

Se emplea un dispositivo disponible en el mercado denominado xArm 1s del fabricante *Hiwonder*. Es un mecanismo de seis grados de libertad con una estructura en aluminio anodizado y una base que se puede fijar mediante ventosas a la superficie. Como se aprecia en la figura 1, posee seis servomotores de bus serie que permiten leer la posición de cada uno durante su funcionamiento. Incluye una tarjeta para su control y la posibilidad de realizar una interfaz propia si se desea.

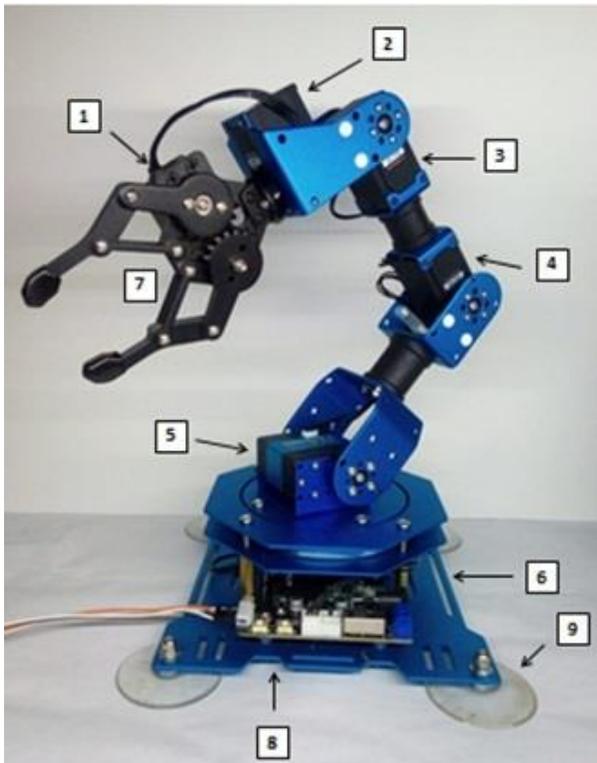


Figura 1. Mecanismo robótico xArm 1 del fabricante *Hiwonder*. Se puede observar las partes que lo componen: (1-6) Servomotores, (7) Elemento de sujeción o *Gripper*, (8) Tarjeta de control, (9) Ventosa de fijación.

Este mecanismo permite el movimiento de cada una de las articulaciones según las características del servomotor instalado, por ejemplo: el elemento de sujeción tiene el servo No. 1 y posee un rango de movimiento de 240°, los servomotores 2, 3 y 4 se pueden mover en un rango de 240°, y los servomotores 5 y 6

poseen un rango de 320°. En cuanto a las características de los servos se tiene que todos se alimentan de 6 a 8 volts de C.D., para los servos 1,2,3 y 4 el torque es de 15 Kg. cm y su velocidad de 0.23 sec/60°, para los servos 5 y 6 el torque es de 25 Kg.cm y la velocidad de 0.22sec/60°. Es importante mencionar que todos los servomotores son tipo Bus lo cual reduce en gran medida el número de cables para su conexión.

El xArm 1s incluye de fábrica un control manual inalámbrico, una APP para control mediante teléfono inteligente, un software para PC para su control, y la posibilidad de controlarlo mediante un mouse alámbrico.

2.2 EL HARDWARE ELECTRÓNICO PARA EL CONTROL DEL MOVIMIENTO.

El movimiento del mecanismo se realiza con los servomotores uno para cada articulación, el manejo de éstos es gobernado mediante una tarjeta de control que viene de fábrica y se puede apreciar en la figura 2. También permite comunicación al exterior mediante módulos de Bluetooth, puerto USB, y un módulo de control inalámbrico.

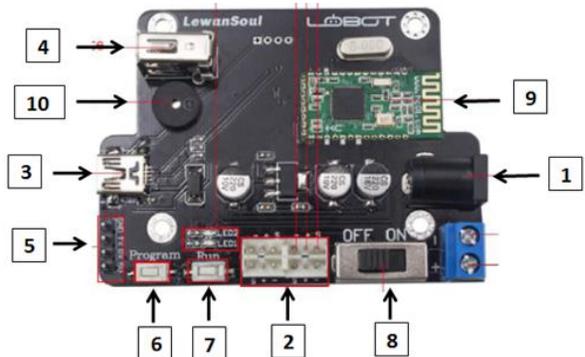


Figura 2. Tarjeta de control para el xArm 1s. Esta incluye: (1) Puerto de alimentación en 7.5 volts de C.D., (2) Puerto para la conexión de los servomotores, (3) Conector USB para el control de la tarjeta, (4) Módulo para control inalámbrico, (5) Interface de comunicación para desarrollos secundarios, (6) Botón para programación, (7) Botón de *Run*, (8) Interruptor de *On*, (9) Módulo de *Bluetooth*, (10) *Buzzer*.

Para realizar la comunicación hacia el exterior del mecanismo se enlaza la tarjeta de control con una tarjeta Arduino UNO, según [5] es una placa de hardware libre que incorpora un microcontrolador reprogramable y una

serie de pines que permiten conectar allí de forma muy sencilla una serie de sensores y actuadores, posee un software de desarrollo gratuito, que se puede descargar del sitio www.arduino.cc, la programación se realiza en lenguaje C, por lo que es muy utilizada para desarrollar proyectos de control entre estudiantes, en [6] se puede consultar el proceso para usar la tarjeta Arduino Uno y variados ejemplos de aplicación ya desarrollados. La tarjeta se puede ver en la figura 3.

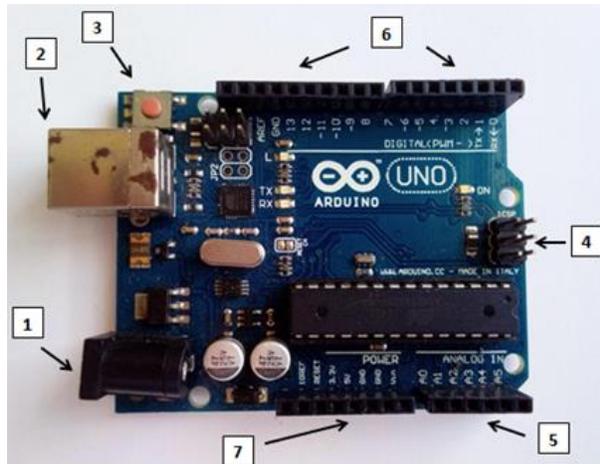


Figura 3. Tarjeta Arduino UNO. Esta desarrollada con el microcontrolador ATmega328, no es parte del xArm 1s. Incluye: (1) Alimentación a 5 volts de C.D., (2) Puerto USB para programación, (3) Botón de *reset*, (4) Puerto para comunicación ICSP, (5) Entradas analógicas, (6) Entradas y/o salidas digitales, (7) Alimentación para sensores.

La tarjeta Arduino UNO como se comenta en [7], es un sistema embebido dedicado a la automatización de procesos físicos e implementación de aplicaciones en ingeniería como electrónica, mecatrónica, robótica o informática. Al utilizar un entorno de desarrollo en lenguaje C permite la adquisición de datos, el procesamiento y control de variables, el envío de señales y comandos. También hace posible el intercambio de información usando una comunicación USB con la computadora y se puede enlazar con otros programas de cómputo como Matlab Y LabVIEW.

Arduino UNO incluye entradas y salidas digitales, entradas analógicas, puertos programables para comunicación por USB, puerto ICSP, puerto de programación por USB, software de programación libre, la principal característica de Arduino UNO es que permite vía USB una comunicación funcional, rápida a

bajo costo entre el mecanismo robótico y una PC o Laptop donde se encuentra la interface de usuario.

El control del mecanismo se realiza a través del intercambio de comunicación entre la tarjeta Arduino UNO y Matlab, sin embargo, se realizó una interface gráfica de usuario (GUI) para mejorar el aspecto visual y motivar el empleo de esta plataforma. En la figura 4 se aprecia la estructura que posee la plataforma para su funcionamiento.

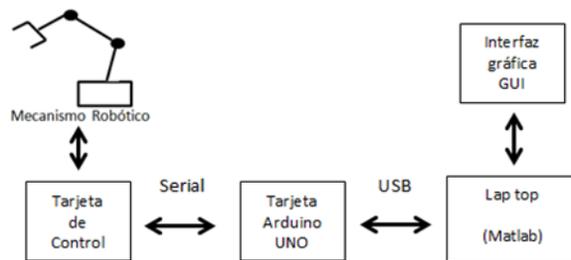


Figura 4. Estructura de la plataforma robótica. Aquí se observa la conexión para realizar la comunicación entre el mecanismo robótico, la tarjeta Arduino UNO y la Laptop con la interfaz gráfica (GUI).

Todas las tarjetas Arduino permiten la comunicación con otros dispositivos usando el puerto serial que se encuentra en los pines D0 (Rx) y D1 (Tx); como se menciona en [8], esto se logra con el protocolo UART (*Universal Asynchronous Receiver-Transmitter*), que funciona mediante cuatro cables conectados entre los dos dispositivos que se comunican, cada dispositivo posee un módulo de transmisión y recepción independiente que no se encuentran sincronizados por lo que la comunicación es asíncrona, cuando un dispositivo transmite datos los envía como una serie de pulsos; cada pulso representa un bit de información, ambos dispositivos deben poder leer los pulsos a la misma velocidad en baudios para que la lectura sea correcta.

La comunicación se ejecuta con el protocolo serie-UART, la tarjeta Arduino Uno se comunica primero con el xArm 1s y luego con la interfaz en Matlab, por lo que necesita dos puertos seriales, uno para conectarse a cada dispositivo. Cabe aclarar que la tarjeta Arduino UNO solamente trae un puerto serial de fábrica que es con el que se conecta a una computadora para grabar los programas con los que trabajará, se ha utilizado la librería *SoftwareSerial.h* que permite crear otro puerto serial en los pines 10 (rx) y 11 (tx). Para la comunicación con el mecanismo xArm 1s se conecta los cuatro cables

de la siguiente manera, los dos primeros son de comunicación el Tx del primer dispositivo se conecta con el Rx del segundo dispositivo y el Tx del segundo se conecta con el Rx del primer dispositivo, los otros dos cables son de alimentación por lo que el tercero se conecta a tierra (GND) en ambos dispositivos, y el último a Vin de la placa Arduino y a V en la placa de control del mecanismo. Esto se puede verificar en la figura 5.

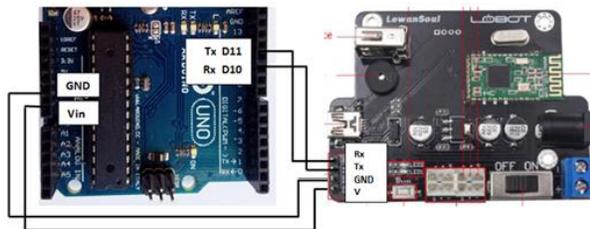


Figura 5. Conexión de la tarjeta de control con la tarjeta Arduino Uno. Se puede ver a detalle la conexión entre las dos, es importante realizarla correctamente para evitar errores en el funcionamiento.

2.3 EL SOFTWARE PARA LA INTERFAZ.

La tarjeta Arduino UNO permite la comunicación con una lap top por el puerto USB, mediante el envío de datos formados por cadenas de caracteres. Por lo que se hace necesario el empleo de otro software que reciba esos datos y posea la capacidad de presentarlos de manera gráfica para observar su evolución a través del tiempo.

Como se recomienda en [9] para realizar gráficas de datos en función del tiempo Matlab es una excelente solución ya que es una herramienta matemática muy potente para realizar los cálculos requeridos en la robótica, facilitando la interpretación y el análisis de los datos obtenidos.

Como ejemplo de las aplicaciones que se pueden lograr con Matlab se puede consultar [10], que muestra la simulación y modelado de un robot que camina, en [11] se tiene un robot basado en el microcontrolador AVR que puede sujetar y colocar piezas.

La interfaz gráfica se realizó considerando una primera etapa que pueda controlar el movimiento de los servomotores y recibir los datos que provienen de éstos para hacerlos visibles al usuario, como segunda etapa se propone agregar el despliegue gráfico y almacenamiento de los datos para un análisis posterior, como tercera etapa

anexar un módulo para el procesamiento matemático de los datos a fin de realizar un análisis cinemático completo.

Para implementar una GUI (*Graphical User Interface*) en Matlab [12], recomienda trabajar con GUIDE. Es una función disponible que genera dos archivos uno **.Fig** que contiene los elementos de la interfaz gráfica y otro **.m** que alberga las funciones y controles de la interfaz. En [13]. Se sugiere para lograr una interfaz funcional elaborar primero un programa en Matlab que contenga los objetos y las acciones que cada objeto realizará, proponer un aspecto de cómo debe ser la GUI, crear la GUI con GUIDE y conectar los elementos gráficos de la interfaz con el programa.

En la figura 6 se aprecia GUIDE con la interfaz ya terminada, se puede observar todos los elementos que conforman la interfaz y están relacionados con el programa de control.

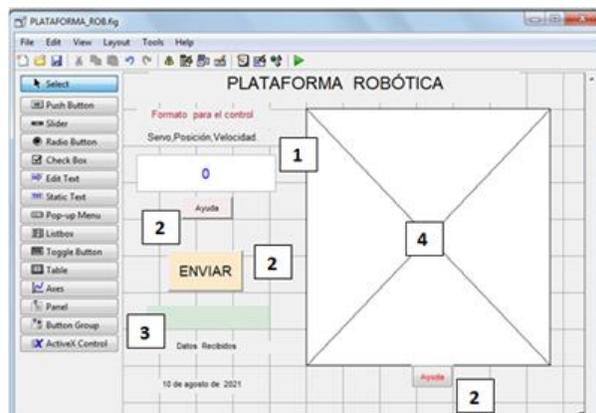


Figura 6. Pantalla de GUIDE. Muestra todos los elementos que integran la Interfaz gráfica, este es el archivo **.fig** que se ha creado para la plataforma robótica, incluye: (1) Cuadro de edición de texto, (2) *Push button*. (3) Cuadro de texto estático, (4) Elemento *axes*.

3 RESULTADOS

El funcionamiento de la interfaz gráfica permite de visualizar la relación entre las variables cinemáticas y la geometría del movimiento.

La figura 7 presenta la interfaz en Matlab ya funcionando. Contiene en la parte superior izquierda un cuadro etiquetado como: *Datos deseados de un servomotor*, este se utiliza para capturar las variables de

la cinemática que controlan un servo del mecanismo robótico xArm 1s.

Primeramente, se tecldea el número de servo que se desea mover, enseguida el número de líneas del encoder que se desplazará el servo, y por último el tiempo en milisegundos con que lo hará. Este proceso es importante ya que se deberán capturar los datos de la misma manera para cualquiera de los otros servos.

En la parte central de la interfaz se encuentra un *push button* denominado (ENVIAR) permite mandar los valores capturados hacia el mecanismo para moverlo físicamente.

Debajo del *push button* hay otro cuadro con fondo verde que permite visualizar la confirmación de los datos alcanzados por el mecanismo, por lo que sus valores serán idénticos a los tecleados en el cuadro de datos deseados.

A la derecha se encuentra una imagen del brazo robótico completo que, mediante cuadros numerados, indica la ubicación de cada uno de los seis servomotores. Con flechas se muestra los movimientos que ejecutan, también se puede ver con números en rojo y azul el rango de valores que se les puede enviar a los servos.

Por ejemplo, para el servo número 6 que presenta un número de líneas de 100 a 900, se podrá desplazar con un rango de 800 líneas del encoder y se podrá capturar cualquiera de los valores entre 100 y 900 en la interfaz.



Figura 7. Interfaz gráfica en Matlab para la plataforma robótica. Se puede apreciar el cuadro denominado: Datos deseados de un servo con números azules. Es para la captura de los valores de las variables cinemáticas que

mueven el mecanismo para este caso se tiene: (6,800,500). Que son los valores que representan el número de servomotor (6), el número de líneas del *encoder* que se debe mover (800), y el tiempo en milisegundos que tardará en realizarse el movimiento (500).

Para analizar el movimiento de las articulaciones se posicionó la plataforma en un sistema coordenado tridimensional con la intención que realice un movimiento siguiendo una trayectoria desde un punto inicial (1), hasta un punto final (4), pasando por los puntos intermedios (2) y (3) como se muestra en la figura 8.

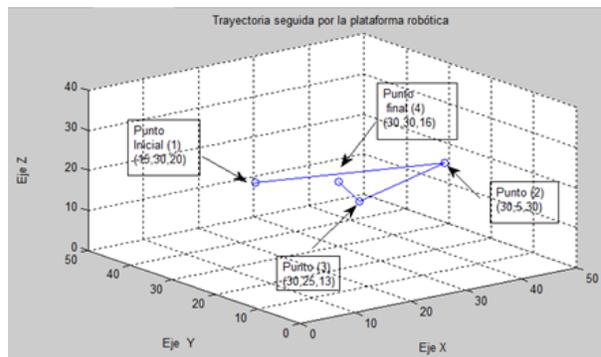


Figura 8. Trayectoria seguida por el elemento final de la plataforma, inicia en el punto (1) con coordenadas (15,30,20) luego continua al punto (2), (30,5,30), pasa al punto (3), (30,25,13) y termina en el punto final (4), (30,30,16). Para lograr esta trayectoria se debe introducir los valores en la interfaz gráfica de las posiciones de cada servomotor de acuerdo con la tabla 1.

Tabla No. 1 Valores de posición para generar la trayectoria de la figura 8.

Servo	Punto inicial (1)	Punto (2)	Punto (3)	Punto final (4)
1	650	650	650	650
2	200	300	300	200
3	500	320	340	150
4	300	360	150	260
5	400	500	430	630
6	800	380	620	710

Con referencia a las pruebas efectuadas a la plataforma se ha comprobado que posee un buen control del movimiento de todos los servomotores del

mecanismo, para mejorar el desempeño de la plataforma se incluyó en la interfaz una imagen con los límites del rango de valores que puede enviarse a cada servomotor, quedando como se indica en la tabla No. 2.

En la misma tabla No. 2 se aprecia en la última columna la amplitud del movimiento que realiza cada articulación en grados sexagesimales. Es importante mencionar que el fabricante especifica que cada línea del *encoder* tiene un valor igual a 0.24° por lo que se debe considerar por ejemplo que 100 líneas del *encoder* es igual 24° .

Tabla No. 2 Rango de valores para los servomotores.

SERVO	Límite inicial	Límite final	Amplitud (°)
1	200	700	120
2	100	900	192
3	0	700	168
4	0	1000	240
5	200	800	144
6	0	1000	240

Cabe mencionar que los límites son para proteger el mecanismo de posibles choques durante su funcionamiento.

En cuanto al tiempo del movimiento se obtuvo valores idénticos a los que menciona el fabricante, que van desde 0 hasta 3000 milisegundos; sin embargo, de acuerdo con las pruebas efectuadas no se recomienda llegar a esos valores puesto que genera mucha inestabilidad en el mecanismo y puede provocar golpes y/o caídas del mismo, es conveniente manejar el mecanismo de 100 hasta 500 milisegundos para mantener el equilibrio y la estabilidad.

En la figura 9. Se tiene una vista general de la conexión física del mecanismo robótico con las tarjetas y la interfaz gráfica instalada en la computadora ya terminada y funcionando.

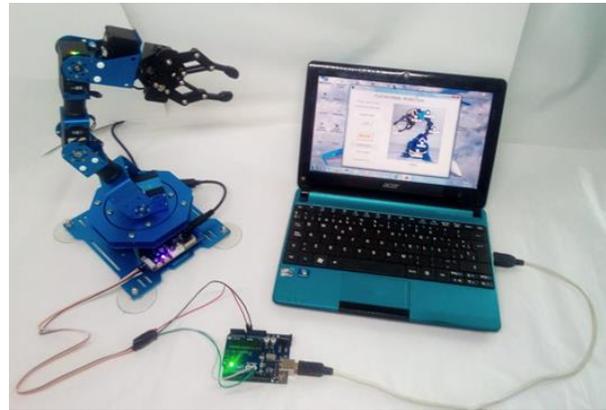


Figura 9. Vista general de la conexión de toda la plataforma ya funcionando.

4 TRABAJOS FUTUROS

Para lograr una mejor operación y funcionalidad en la plataforma se propone los siguientes trabajos a futuro.

- Diseñar y construir una base con superficie lisa, para fijar con un mayor agarre el mecanismo, ya que durante la prueba se detectó falta de estabilidad cuando el robot realiza movimientos con velocidad considerable y puede ocasionar golpes y/o desperfectos.
- Incrementar las funciones de control en los servomotores, como pueden ser la determinación de la aceleración mediante cálculo matemático en el Matlab.
- Mejorar las funciones de lectura de los servomotores, como el caso de la temperatura de operación.
- Diseñar un ambiente gráfico más agradable en la interfaz, que incluya de forma individual el control de cada servomotor.
- Elaborar un módulo que realice el despliegue gráfico y almacenamiento de los datos adquiridos para un análisis posterior.
- Desarrollar un módulo para el procesamiento matemático de los datos a fin de realizar un análisis cinemático completo, así como también un análisis que se relacione con la generación y control de trayectorias.

Como se ha podido apreciar en este trabajo se ha logrado integrar una plataforma completa y funcional mediante una interfaz gráfica en Matlab que se enlaza al mecanismo xArm 1s permitiendo visualizar y manipular las variables cinemáticas que se miden en las articulaciones.

5 CONCLUSIONES

Esta propuesta de plataforma representa una muy buena herramienta didáctica para los estudiantes de mecatrónica y robótica de nivel superior, por la relación costo-beneficio, comparada con equivalentes didácticos disponibles en el mercado es varias veces más económica.

Por los trabajos que pueden ejecutar los alumnos representa un gran modelo didáctico, ya que permite desarrollar competencias como: armar un mecanismo robótico completo (mecánica), aprender a conectar y programar el hardware electrónico para el control (electrónica). Desarrollar interfaces gráficas de usuario en un software adecuado para la robótica (informática). Representado un proceso de diseño mecatrónico muy conveniente para que los estudiantes de las universidades practiquen sus competencias.

Este trabajo es una base para efectuar prácticas relacionadas con resolver los dos principales problemas de la cinemática de un robot, el primero consiste en calcular la posición y orientación del efector final a partir de las coordenadas articulares y la geometría del robot, en la plataforma equivalen a todas las posiciones (ángulos) de cada servomotor (articulaciones) que forman el mecanismo robótico. Se podrá ingresar esos datos a Matlab que posee poder de cálculo suficiente para realizar las operaciones con las matrices de transformación homogénea y utilizando el método de Denavit-Hartenberg. A este problema se le conoce como cinemática directa.

El segundo problema se denomina cinemática inversa y consiste en calcular las coordenadas articulares conociendo la posición y orientación del efector final, de igual manera mediante el método de Denavit-Hartenberg y empleando Matlab. Incluso Matlab permite la representación gráfica e 3D de estos problemas.

Otras prácticas a desarrollar se relacionan con la generación de trayectorias del mecanismo robótico a partir de los datos calculados en la cinemática anterior y su implementación en otra interfaz de Matlab, ya que podría visualizarse el movimiento del efector final.

Otra aplicación para esta plataforma es realizar operaciones de movimiento de objetos o partes pequeñas y de poco peso que se encuentren en el área de trabajo del mecanismo, perfectamente puede manipular este tipo de objetos sin ningún problema, ya que posee la precisión y movimientos adecuados. Únicamente basta con realizar el programa y grabarlo en el dispositivo.

REFERENCIAS

- [1]. Lammer, L. Lepuschitz, W. Kynigos, C. Giuliano, A. Girvan, C. (2017) ER4STEM Educational Robotics for Science, Technology, Engineering and Mathematics, *Robotics in Education Research and Practices for Robotics in STEM Education*, Volume 457, 2017, 95-101.
- [2]. Sánchez, J. L. Juárez, C. (2018). Modelo de Robótica Educativa con el Robot Darwin Mini para Desarrollar Competencias en Estudiantes de Licenciatura, *RIDE Revista Iberoamericana Para La Investigación Y El Desarrollo Educativo*, 2018, 8(15), 877 - 897.
- [3]. Ruge, I. A. Jiménez, F. R. Hernández, O.M. (2017) Robot DARO: plataforma robótica para la enseñanza en ingeniería, *Ingenium*, 2017, 18(35), 58-74.
- [4]. Asín, G. Pastor, J. (2013) Low Cost Didactic Robotic Platform Based on Player/Stage Software Architecture and La Fondera Hardware, *IEEE REVISTA IBEROAMERICANA DE TECNOLOGIAS DEL APRENDIZAJE*, 2013, 8(3), 126-132.
- [5]. Torrente, O. (2018) *Arduino curso práctico de formación*, México: Alfaomega, 2018.
- [6]. Ganazhapa, B. (2017) *Arduino guía práctica*, México: Alfaomega, 2017.
- [7]. Reyes, F. Cid, J. (2017) *Arduino aplicaciones en robótica, mecatrónica e ingenierías*, México: Alfaomega, 2017.

[8]. Moreno, A. Córcoles, S. (2018) *Arduino curso práctico*, España: Ra-Ma, 2018.

[9]. Reyes, F. (2015) *MATLAB Aplicado a robótica y mecatrónica*, México: Alfaomega, 2015.

[10]. Liang, C. Ceccarelli, M. Carbone, G. (2011) Design and Simulation of Legged Walking Robots in MATLAB® Environment, Perutka K. (Ed.). *MATLAB for Engineers – Applications in Control, Electrical Engineering, IT and Robotics*. Croatia: In Tech, 2011, 459-492.

[11]. Yenorkar, R., Chaskar, U. M. (2018) "GUI Based Pick and Place Robotic Arm for Multipurpose Industrial Applications," *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, 2018, 200-203.

[12]. Marchand, P. Holland, O. T. (2003). *Graphics and GUIs with MATLAB*, FLORIDA: CRC press, 3rd ed. 2003.

[13]. Lent, C.S. (2013) *Learning to program with MATLAB: building GUI tools*, USA: Wiley, 2013.

Acerca de los autores



Darvi Echeverría Sosa. Recibió el grado de Ingeniero Mecánico en el Instituto Tecnológico de Mérida en 1996 con cédula profesional 2689731 es Maestro en Ingeniería Mecatrónica con cédula profesional 7136525 egresado de la Universidad

Modelo en el año 2011 en la ciudad de Mérida Yucatán México, actualmente es profesor de tiempo completo en el área de Electromecánica del Tecnológico Nacional de México, Campus Motul, sus intereses son el desarrollo tecnológico y la investigación en las áreas de Diseño y automatización de máquinas y mecanismos así como en el diseño mecatrónico.



Alejo Mosso Vázquez. Recibió el grado de Ingeniero en Comunicaciones y Electrónica de la Escuela Superior de Ingeniería Mecánica y Eléctrica del Instituto Politécnico Nacional (IPN) de México en 1975. Es maestro en ciencias con especialidad en Control por el CINVESTAV-

IPN en México 1986. Es también maestro en ciencias en sistemas de la manufactura con especialidad en Robótica por el ITESM-UT (USA) en 1993. También obtuvo el grado de Doctor en Ingeniería y Ciencias Aplicadas en Cuernavaca Morelos, México en 2012 por el CIICAP-UAEM. Sus intereses de investigación son Programación Matemática aplicada a la Robótica Humanoide, Control Automático y Redes Neuronales – Deep Learning.