

Ensamblaje y Control de un Robot con Arquitectura Antropomórfica de 5 Grados de Libertad

Miguel Páez
Automatización y Control Industrial
Escuela Politécnica Nacional
Quito- Ecuador
miguel.paez@epn.edu.ec

Evelyn Curicho
Automatización y Control Industrial
Escuela Politécnica Nacional
Quito- Ecuador
evelyn.curicho@epn.edu.ec

Nelson Sotomayor
Automatización y Control Industrial
Escuela Politécnica Nacional
Quito- Ecuador
nelson.sotomayor@epn.edu.ec

Resumen—El presente artículo explica el ensamblaje y control de un robot antropomórfico de 5 grados de libertad para seguimiento de ruta, el control se lo realizó con la ayuda de programación en lenguaje de alto y bajo nivel, los cuales a su vez serán enlazados por medio de una memoria compartida con la ayuda del software MATLAB. En el caso del control de bajo nivel se lo realiza usando la placa Arduino, la misma que se encarga de la comunicación bidireccional con los motores, el control en alto nivel utiliza la cinemática del robot, algoritmo de seguimiento de ruta e interfaz gráfica, este control se desarrolló en MATLAB.

Keywords—Robot, antropomórfico, memoria compartida, MATLAB, 5 grados de libertad.

I. INTRODUCCIÓN

En la actualidad, la robótica y automatización cada día están más presentes en la vida cotidiana con el propósito de precautelar y mejorar la calidad de vida de los seres humanos, ya sea para aumentar su fuerza, mejorar su precisión, manipular carga y/o materiales peligrosos, entre otros [1].

Por esta razón se plantea el ensamblaje y control de un robot antropomórfico capaz de realizar trazado de figuras predeterminadas por medio del control de sus cinco grados de libertad, para conseguir dicha meta, uno de los pasos primordiales tiene que ver con la cinemática directa e inversa, pues de esto dependerá la posición que tomará el efector final y los valores que deben enviarse a cada uno de sus cinco grados de libertad para lograr el movimiento del robot antropomórfico. Es necesario mencionar que para la estructura física se tomó como referencia el diseño mecánico desarrollado en un trabajo de titulación como parte de las actividades del proyecto de investigación PIMI-14-04 de la Escuela Politécnica Nacional [2].

Para realizar el control del brazo se utiliza un controlador en bajo nivel usando un sistema embebido, un controlador en alto nivel colocado en un PC y un bloque de memoria compartido que se encarga de enlazar los dos controladores anteriores.

II. METODOLOGÍA

Primero se realiza un estudio de la cinemática para continuar con el ensamblaje mecánico, el diseño de hardware y de software

A. Cinemática

Para el movimiento del robot antropomórfico sin tomar en cuenta la fuerza aplicada es necesario el estudio de la

cinemática, esta busca describir el movimiento espacial del robot con relación a la posición del efector final y los valores articulados del mismo. La cinemática directa tiene como finalidad conocer la posición del efector final con relación a la base del robot cuando se tiene conocimiento de la posición y orientación relativa de un eslabón con el siguiente, mientras que en la cinemática inversa se conoce la posición del efector final con relación a la base, para obtener la posición y orientación relativa de cada eslabón con el siguiente.

Para el análisis de un robot antropomórfico de cinco grados de libertad, se procede con la asignación de ejes de coordenadas fijas y móviles [4] como se muestra en la Figura 1.

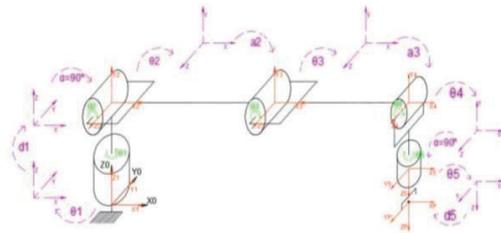


Figura 1. Asignación de ejes de coordenadas fijas y móviles.

Existen diversos métodos para obtener la cinemática de un sistema, para este caso se optó por el método matricial Denavit-Hartenberg para obtener de manera sistemática un sistema de coordenadas [1]. De este análisis con sus consideraciones se desarrolla la Tabla 1.

TABLA 1: PARÁMETROS D-H

Eslabón	θ	d	α	a
1	θ_1	d_1^*	90	0
2	θ_2	0	α_2	a_2
3	θ_3	0	α_3	a_3
4	θ_4	d_4^*	90	0
5	θ_5	d_5^*	α_5	0

La cinemática directa del robot antropomórfico de 5 grados de libertad se aprecia en la Ec. (1).

$$\begin{cases} x = [a_2 \cos(\theta_2) + a_3 \cos(\theta_2 + \theta_3)] \cos \theta_1 \\ y = [a_2 \cos(\theta_2) + a_3 \cos(\theta_2 + \theta_3)] \sin \theta_1 \\ z = a_3 \sin(\theta_2 + \theta_3) + a_2 \sin \theta_2 \end{cases} \quad (1)$$

B. Emsablaje mecánico

Para el modelo de las piezas se utiliza el Software SOLIDWORDKS, este diseño se encuentra dividido en dos subcomponentes:

- a) *Base*: sirve como soporte del brazo, aquí se encuentran ubicados tres actuadores, además de 2 eslabones.
- b) *Brazo*: se encuentra conformado por tres eslabones, además de 3 actuadores.

No se hace énfasis en la pinza pues los 5 grados de libertad no se incluyen en la apertura y agarre de esta.

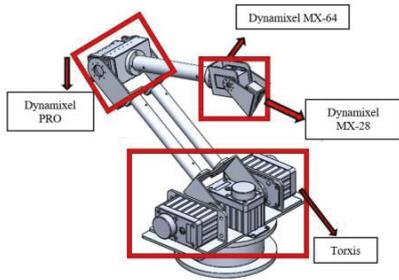


Figura 2. Diseño brazo de cinco grados de libertad

En la Figura 2 se puede observar el diseño final del brazo, con la ubicación de los actuadores, el brazo se manufacturo usando acero de transmisión.

Los grados de rotación que deben tener cada uno de los eslabones son:

- Eslabón 1: rotación entre 0° a 180°.
- Eslabón 2: rotación entre 0° a 90°, el rango de este movimiento se debe a la forma constructiva del robot
- Eslabón 3: rotación entre 0° a 150°, se debe a que el trazado de figuras se realiza en la parte frontal.
- Eslabón 4: rotación entre 0° a 180°: al igual que el eslabón 2.
- Eslabón 5: rotación entre 0° a 360°: este eslabón no posee ninguna restricción física para el trazado de figuras.

C. Diseño electrónico

La arquitectura del robot antropomórfico a implementarse se encuentra constituida por el sistema embebido Arduino Mega, el módulo RS-485, los motores, la PC y la fuente de alimentación, como se observa en la Figura 3.

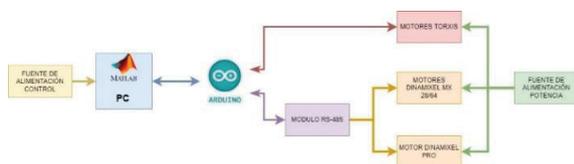


Figura 3. Arquitectura del Sistema Electrónico

Para el diseño electrónico en lo que respecta a hardware, el Arduino mega sirve como lazo entre los actuadores con sus sensores y el algoritmo de control que se encuentra en la PC, este sistema embebido se encarga de enviar las acciones que comandan el posicionamiento de los motores y recibir la información de la posición de los actuadores. En la Figura 4 se muestra el esquema de conexión del diseño electrónico que incluye el sistema embebido y los periféricos.

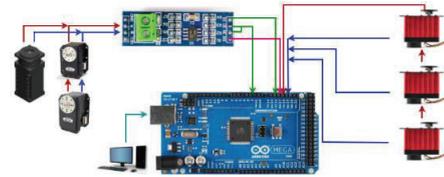


Figura 4. Esquema general del Sistema Electrónico

D. Diseño de software

Antes de empezar con el diseño de software es necesario realizar la configuración de los parámetros internos de los motores del robot, debido a que se utilizan motores de la marca Dynamixel y Torxis, es necesario utilizar los softwares Dynamixel WIZARD 2.0 y Jrk Configuration Utility, respectivamente, para configurar los parámetros como numeración, velocidad de movimiento, controladores internos, entre otros.

a) *Lenguaje en alto nivel*: Se lo realizó en Matlab, este posee el algoritmo para el trazado de figuras, este algoritmo entrega los puntos en los ejes X, Y, Z gracias a las ecuaciones paramétricas que representan dichas figuras, posteriormente con ayuda de la cinemática inversa se transforma esos puntos de las tres dimensiones en grados para cada una de las articulaciones con el fin de que el brazo pueda posicionarse en las coordenadas deseadas. Adicional a la programación mencionada se tiene un esquema de bloques en Simulink para mostrar datos, acondicionar valores, controlar tiempos, etc. El diagrama de bloques implementado se puede observar en la Figura 6.

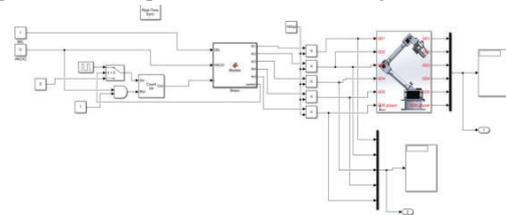


Figura 6. Diagrama de bloques Simulink.

Para completar esta sección, se realizó una interfaz gráfica con el fin de que sea fácil operar al robot, en la Figura 7 se puede observar la interfaz gráfica.



Figura 7 Interfaz hombre máquina

b) Lenguaje en bajo nivel:

Este es el encargado de comandar directamente a los motores, se lo realizó en la plataforma Arduino, su objetivo principal es recibir la posición deseada desde la memoria compartida, enviar esa información a los motores, recibir la realimentación de la posición real y mandar esa posición nuevamente hacia la memoria compartida. Para este

objetivo se utilizó las librerías propias de las marcas de los motores tanto Torxis como Dynamixel, además de comunicación serial con la PC y con el módulo RS-485.

c) Memoria compartida:

La memoria compartida consta de dos partes, la primera es un bloque de Simulink llamado “Generic Share Memory” cuya estructura interna se puede observar en la Figura 8. Este bloque se encarga de definir la cantidad y el tipo de datos que serán enviados y recibidos por Matlab.

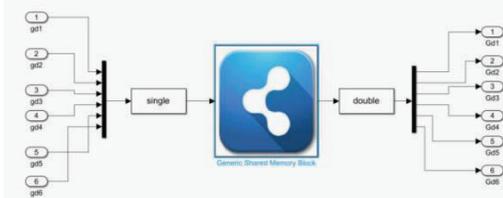


Figura 8. Bloque memoria compartida

La segunda parte es un algoritmo de comunicación realizado con ayuda del software Visual Studio, su objetivo principal es enlazar la comunicación entre Matlab y Arduino, este programa controla el inicio y fin de la comunicación, envío y recepción de datos, acondicionamiento, tiempos, entre otros.

III. RESULTADOS Y DISCUSIÓN

El resultado es un robot antropomórfico de cinco grados de libertad para seguimiento de rutas, el robot final se muestra en la Figura 9, el cual puede alcanzar una altura máxima de 83.5 [cm] y pesa aproximadamente 20 [kg].



Figura 8. Brazo final ensamblado

Para el análisis de la función de seguimiento de ruta se optó por contrastar la figura realizada por el robot con la simulación obtenida con ayuda de Matlab (Figura 10 y Figura 11).

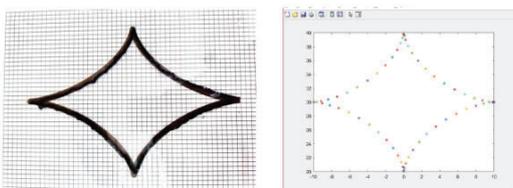


Figura 10. Comparación Asteroide

Como se puede observar la figura Asteroide realizada por el robot es muy similar al patrón, sin embargo, si tiene leves desviaciones, uno de los motivos que lleva a estas fallas es la resolución de los motores, en especial de la base ya que estos están orientados mayormente a trabajos de fuerza.

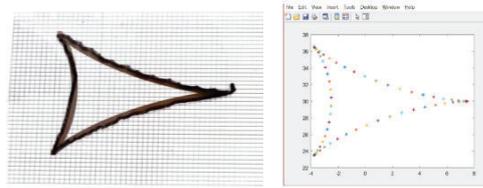


Figura 11. Comparación Hipocicloide de 3 puntas.

Al trazar la figura Hipocicloide, de manera similar al caso anterior las figuras son similares, se pueden notar ciertas desviaciones especialmente en la punta ubicada a la derecha, debido a que en este punto es donde se inicia el trazado de la figura.

IV. CONCLUSIONES

La implementación de una memoria compartida para enlazar dos lenguajes de programación y dos softwares diferentes como son Matlab y Arduino es de gran ayuda ya que evita saturar el sistema embebido entregándole pocas y simples tareas, y dándole las tareas más demandantes al computador el cual posee una capacidad de procesamiento mucho mayor.

El error que se produce durante el trazado de figuras de manera física es causada por los motores Torxis ubicados en la base, ya que estos motores poseen un gran torque lo cual le permite ejercer mucha fuerza, esto le permite mover y manipular toda la estructura del brazo, en cambio, su exactitud al ubicarse en determinada posición posee un error de consideración, además que el software encargado de configurar su tarjeta de control y la tarjeta de control en sí, son considerados obsoletos por los mismos fabricantes los cuales a su vez recomiendan su cambio.

Una vez entendido el funcionamiento de la cinemática inversa de un robot de cinco grados de libertad y gracias a su implementación en líneas de código, la implementación de cualquier figura se vuelve sencilla si se conoce sus puntos en los ejes X, Y, Z.

AGRADECIMIENTOS

Nos complace expresar nuestra gratitud por la generosidad recibida por parte del EPN, a través del proyecto de investigación PIMI-14-04 al colaborar con la obtención de los motores usados en el robot antropomórfico.

REFERENCIAS

- [1] B. Ollero, de Robótica, manipuladores y robots móviles, Barcelona, Alfaomega, 2001.
- [2] V. Chávez y F. López, Diseño y simulación de un brazo manipulador de objetos en zonas de alto riesgo para seres humanos, Quito: Escuela Politécnica Nacional, 2016.
- [3] «DataSheet Dynamixel Driver,» Robotics e-Manual, 2018. [En línea]. Available: https://emanual.robotis.com/docs/en/software/dynamixel/dynamixel_sdk/device_setup/. [Último acceso: 01 08 2022].
- [4] M. V. Oramas Granja, Modelación y análisis de la cinemática directa e inversa del manipulador Standford de seis grados de libertad, Quito: Escuela Politécnica Nacional, 2014.