



Universidad de
los Andes
Colombia

Departamento de Ingeniería
Eléctrica y Electrónica

Taller 3

Diseño de un robot manipulador

1st Cristian Alejandro Ríos
School of Electrical and
Electronic Engineering
Universidad de los Andes, Bogotá
ca.rios@uniandes.edu.co

2nd Juan Diego Parada
School of Electrical and
Electronic Engineering
Universidad de los Andes, Bogotá
jd.parada@uniandes.edu.co

3rd Jose David Chavez
School of Electrical and
Electronic Engineering
Universidad de los Andes, Bogotá
jd.chavez@uniandes.edu.co

4th Alejandro Silva
School of Electrical and
Electronic Engineering
Universidad de los Andes, Bogotá
a.silvah@uniandes.edu.co

Resumen—En el siguiente taller se determinó y se desarrolló el diseño del robot manipulador. Se determinó que el robot manipulador tendrá 3 grados de libertad, el primero corresponde al movimiento en el plano xy, el segundo y el tercero son para el movimiento en el eje z para tener una mayor movilidad por parte del robot. Adicionalmente, se determinó que no habría necesidad de una PCB en la construcción del robot. Finalmente, mediante ROS se verificó el correcto funcionamiento del robot que incluye el movimiento de sus 4 motores mediante las teclas 'y','h','u','j','i','k','o','l' que le permitirán a un usuario controlar el robot, la interfaz que permite visualizar la posición en tiempo real del end-effector y la posibilidad llevar el end-effector del robot a una posición destino deseada.

Index Terms—ROS, Arduino, diseño mecánico, Python, robot.

Se determinará la configuración de este, como son sus dimensiones y grados de libertad, evaluando ventajas y desventajas de cada configuración. Por otro lado, se determinarán los elementos que constituirán el robot y que ayudan a su movimiento y comunicación con ROS. También, se desarrollarán los planos mecánicos que incluyen las diferentes partes del brazo y el mecanismo de movimiento que usará tanto para el desplazamiento en el espacio como para la tarea de agarrar un objeto. Finalmente, se verifica el funcionamiento del robot verificando que pueda moverse utilizando el teclado, además de observar la posición en tiempo real del end-effector y llevar el end-effector del robot a una posición destino deseada.

I. INTRODUCCIÓN

Una de las partes más importantes en la construcción del robot es el diseño mecánico de este. Un buen diseño mecánico está relacionado con la adecuada planeación de la construcción del robot, la cual debe estar orientada al problema que se desea resolver [1]. Debido a su relevancia, el siguiente taller se enfoca en el diseño del robot, incluyendo tanto la parte mecánica como eléctrica.

II. METODOLOGÍA

Como el presente taller pretende implementar el diseño mecánico y eléctrico del robot, la metodología utilizada tiene los siguientes pasos.

1. Definir los grados de libertad del brazo.
2. Definir el alcance del brazo.
3. Definir las dimensiones del brazo.

- a diferencia de otros tipos de diseños, este es más fácil de controlar e implica una configuración más sencilla de los motores en el código sin ser una configuración obsoleta. Adicionalmente, este diseño otorga una estabilidad suficiente en el robot para que desempeñe las tareas necesarias, fijándolo mediante tornillos a la base del robot diseñado en el taller anterior.

III. RESULTADOS Y ANÁLISIS DE RESULTADOS

Para la construcción del robot, es necesario tener claro cuál va a ser la tarea o función que este va a desempeñar. En este caso, el end-effector del robot debe ser capaz de desplazarse hasta cierto lugar y tomar una bola de pin pong. Para lograr este objetivo, se definieron las etapas necesarias en el diseño del robot explicadas a continuación.

Dado que el brazo va a estar sobre el robot diseñado en el taller, se decidió que el robot tendrá 3 grados libertad, ya que el equipo consideró que estos serían suficientes para el movimiento del brazo en determinado espacio sin complicar más el manejo del robot, agregándole más grados de libertad. Estos 3 grados de libertad le permitirían al robot moverse angularmente en el plano xy para determinar la dirección que va a tomar en este plano, el segundo y tercer grado de libertad son para facilitar el movimiento del robot en el plano xz, teniendo x apuntando en la dirección escogida en con el primer grado de libertad. No se consideró que el uso de más grados de libertad fueran necesarios, ya que con estos 3 el robot tiene una adecuada locomoción para la tarea que se le va a enmendar a este.

Para el diseño de las piezas se optó por un diseño simple y ligero, de tal forma que el robot diseñado en el taller 2 no tenga que cargar con un peso mucho mayor al que tiene actualmente, en consecuencia, se usó el material de PLA porque este es ligero y lo suficientemente resistente para los propósitos mecánicos de este brazo. Esta configuración consta de trece partes diseñadas en Inventor para posteriormente imprimirlos en el material mencionado y ensamblarlo.

Technical drawing of a robotic gripper assembly. The drawing shows a base with a circular platform (Ø150.00) and a gripper arm. Dimensions are provided in millimeters (mm). Key dimensions include: 120.00 (total height), 40.00 (arm height), 130.00 (base width), 211.00 (base depth), 150.00 (base radius), 133.75 (gripper width), 100.00 (gripper arm length), 30.00 (gripper arm thickness), 10.00 (gripper arm radius), 39.75 (gripper arm offset), 40.00 (gripper arm offset), 61.25 (gripper arm offset), 70.00 (base width), 150.00 (base radius), 211.00 (base depth), 150.00 (base radius), 211.00 (base depth), 150.00 (base radius), 211.00 (base depth).

Diseño electrónico

En esta caso se utilizó una mini protoboard para las conexiones, su hace una conexión entre el puerto de 5v del Arduino a un path de la protoboard y de este

path se conectan los 5v que van a los servomotores, del puerto de GND se conecta un cable que va a otro path de la protoboard y de este path se conectan las tierras de los servos, finalmente se usaron los pines digitales 9, 10, 11 y 12 del Arduino. El pin 9 para el servo de abajo, el pin 10 para el servo de la primera junta, el pin 11 de la segunda junta y el pin 12 para el servo de la garra. A continuación se presenta el plano electrónico.

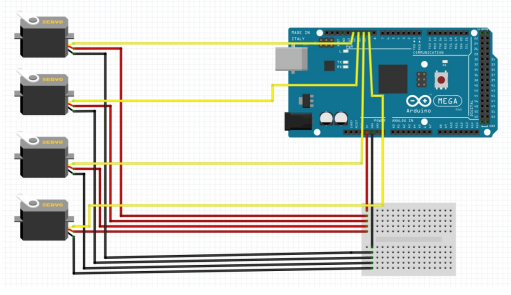


Figura 2: Plano electrónica del brazo robótico

Material y Costos

En la tabla I se muestran todos los componentes usados en la construcción del robot junto con sus respectivos costos.

Cuadro I: Costos de materiales

Material	Cantidad	Precio por unidad (COP)
Servomotor	4	\$12.000,00
Tornillos	17	\$100,00
Arduino Mega	1	\$85.000,00
Jumpers	20	\$4.000,00
PLA	-	-

SOLUCIÓN

/robot_manipulator_teleop:

Este es el nodo encargado de controlar las juntas del robot mediante las teclas. Este nodo envía un string al Arduino y el Arduino lo identifica para mover determinados grados del servomotor correspondiente a esta junta. El control del movimiento de sus 4 motores se hace mediante las teclas 'y', 'h', 'u', 'j', 'i', 'k', 'o', 'l' que le permitirán a un usuario controlar el robot. Las teclas 'y' y 'h' son para controlar el primer servomotor que corresponde al que está en la base del brazo robótico, las teclas 'u' y 'j' controlan el movimiento del segundo motor que está en la primera junta del brazo, las teclas 'i' y 'k' permiten controlar el movimiento del tercer motor el cual está en la segunda junta y finalmente con las teclas 'o' y 'l' se controla el motor que corresponde a la garra del brazo.

/angulo:

Este nodo envía al brazo robótico el ángulo que desea el usuario en cada servo, esto se hace a través de una interfaz gráfica en la que el usuario puede poner los ángulos que requiere en cada servomotor y enviarlo para que el end-effector del brazo robótico llegue a una posición en el espacio. Para esto se hizo una caracterización que está adjunta en la carpeta enviada junto a este pdf.

/robot_manipulador_planner:

Este tópico se encarga de enviar un mensaje tipo twist con la posición final lineal en x & y por medio del serial entre ROS y Arduino. El mensaje de la posición será indicado desde el tópico por medio de una interfaz, y dicho mensaje será recibido por el arduino para así, realizar el siguiente cálculo que permitirá que cada servo se mueva en un ángulo especificado:

$$\cos(\theta_1) = \frac{1}{x^2 + y^2} (x(a_1 + a_2 \cos(\theta_2)) \pm ya_2 \sqrt{1 - \cos^2(\theta_2)})$$

$$= \frac{\arccos}{x^2 + y^2} (x(a_1 + a_2 \cos(\theta_2)) \pm ya_2 \sqrt{1 - \cos^2 \theta_2})$$

$$\cos(\theta_2) = \frac{1}{2a_1a_2} ((x^2 + y^2) - (a_1^2 + a_2^2))$$

$$= \frac{\arccos}{2a_1a_2} ((x^2 + y^2) - (a_1^2 + a_2^2))$$

De esta manera, se caracterizó los valores ingresados en el tópico en un intervalo de $0 < x \leq 1$, $0 < y \leq 1$. Esto tras realizar diferentes pruebas que permitan ubicar la garra en los lugares deseados.

/robot_manipulador_camara:

Manda un string dependiendo del frame en que se encuentre el objeto localizado. El objeto se localiza dependiendo de su color que se configura según lo deseado.

Para esto, se toma una cámara y se divide su rango de visión en 35 frames de la siguiente manera:

1	6	11	16	21	26	31
2	7	12	17	22	27	32
3	8	13	18	23	28	33
4	9	14	19	24	29	34
5	10	15	20	25	30	35

Figura 3: Frames tomados para dividir el rango de visión de la cámara

Como se puede observar, se tienen 7 divisiones en el eje x y 5 en el eje y. Esto, con el objetivo de

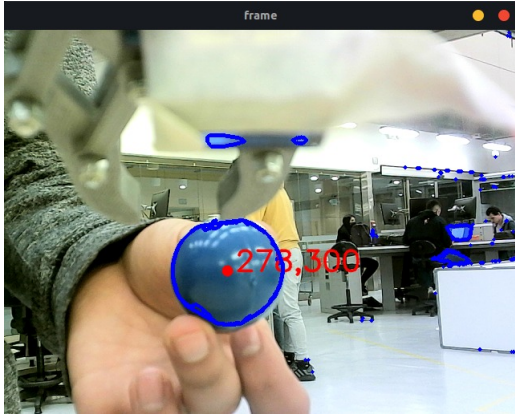


Figura 4: Cámara reconociendo el color azul de un ping pong

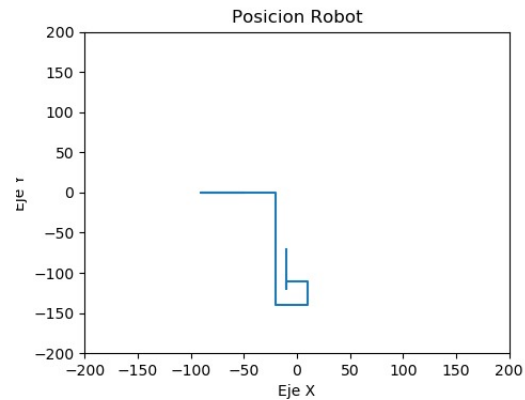


Figura 5: Interfaz gráfica de la posición del end-effector del manipulador

ajustar el movimiento de los servomotores por cada frame encontrado. Es decir, si la cámara reconoce el color deseado, los servos lograrán ubicar el brazo en la posición x y y deseada.

- Eje X : Se tienen las siguientes divisiones:

- Columna 1: Ángulo = 60°
- Columna 2: Ángulo = 54°
- Columna 3: Ángulo = 47°
- Columna 4: Ángulo = 40°
- Columna 5: Ángulo = 33°
- Columna 6: Ángulo = 26°
- Columna 7: Ángulo = 20°

- Eje Y : Se tienen las siguientes divisiones:

- Fila 1: Ángulo = 120°
- Fila 2: Ángulo = 135°
- Fila 3: Ángulo = 140°
- Fila 4: Ángulo = 140°
- Fila 5: Ángulo = 140°

Como se puede observar, el ángulo se fija en 140° , pues, es en el cual alcanza la altura de 10 cm a la que se encontrarán los objetos a alcanzar.

/turtlebot_posicion:

Este nodo envía la posición actual del end-effector a través de un mensaje tipo twist para poder visualizarla a través de una interfaz gráfica. El mensaje es emitido por el nodo del Arduino, el cual envía el movimiento de los servomotores que se desplazan en el eje x y en el eje y , para así recibirlos en el tópicos como un mensaje tipo twist y lograr así generar la gráfica.

IV. MONTAJE

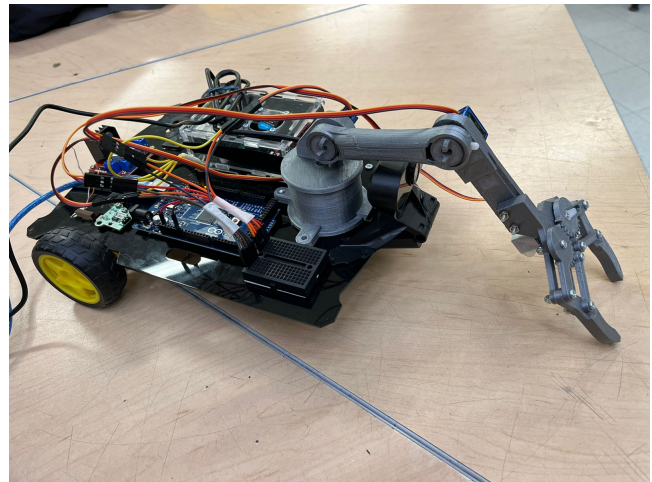


Figura 6: Montaje del manipulador sobre el robot

V. CONCLUSIONES

En conclusión, se observa que el diseño del robot cumple con los requerimientos dispuestos por el equipo para que no ocupe un volumen muy grande sobre la base diseñada en el taller 2 y que no tenga mucho peso de tal forma que los motores sea capaces de mover esta carga. Además, cumple con todas las funcionalidades de movimiento mediante el teclado, visualización del end-effector del brazo robótico y la posibilidad de llevar el end-effector del robot a una posición destino deseada..

REFERENCIAS

- [1] Capitulo 3 Diseño mecánico de los robots [Online]. Available: http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/betanzos_m_w/capitulo3.pdf

- [2] G. Villasenor, E. Gorrostieta, C. Pedraza, J. Ramos, I. Collazo, A. González, A. Romero and J. Serrano, Diseño Mecatrónico de un Robot Móvil (Configuración Diferencial) in 8º Congreso Nacional de Mecatrónica, Veracruz, 2009.