

Taller 2 de Robótica

Ivan David Perez Moreno
Universidad de los Andes
id.perez@uniandes.edu.co

Juan José Arévalo
Universidad de los Andes
jj.arevaloh@uniandes.edu.co

Diego Felipe Rodriguez
Universidad de los Andes
df.rodriguezg@uniandes.edu.co

May 10, 2022

Introducción

1 Descripción del robot

1.1 Materiales

- 4 Servos MG-90S.
- 1 Servo SG-90.
- 1 Protoboard.
- Jumpers Macho-Hembra x10.
- Jumper Macho-Macho x10.

1.2 Planos

1.2.1 Planos mecánicos

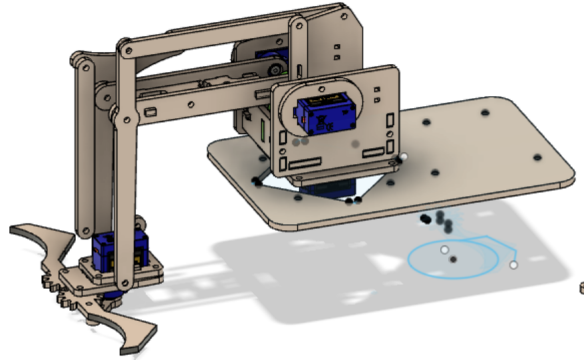


Figure 1: Modelo 3D

Los planos 2D se pueden encontrar adjuntos a este documento.

1.2.2 Planos electrónicos

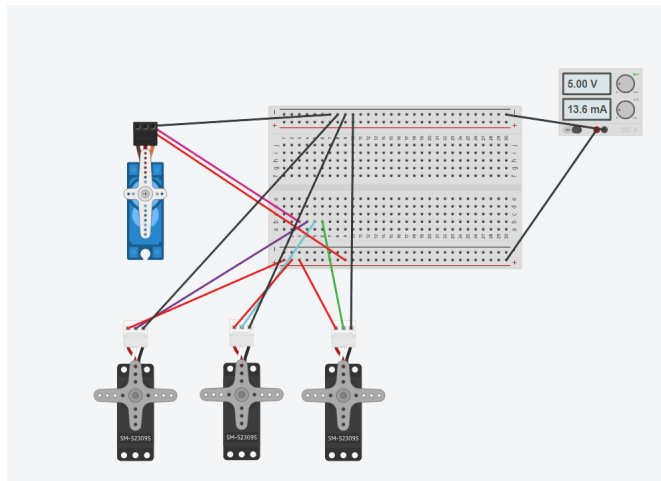


Figure 2: Plano Electrónico

1.3 Integración

Para la integración de la parte mecánica con la parte electrónica se tomo el chasis del brazo diseñado para finalmente integrar los motores y todas las conexiones generales a través del diseño planteado. A continuación se muestra la integración final del robot donde se crean las conexiones entre el cerebro de la maquina y su parte física.

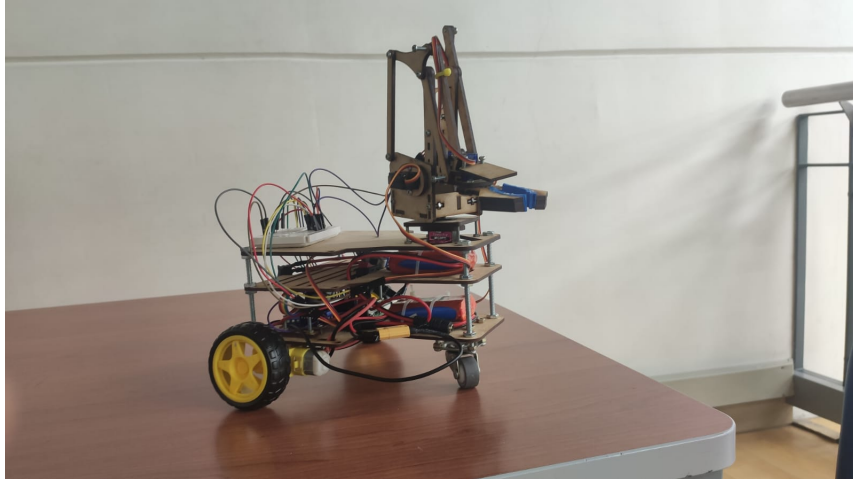


Figure 3: Integración física

1.4 Funcionamiento general

Para el funcionamiento general como se puede observar tanto en los planos electrónicos como físicos del robot, se implemento un manipulador de 3 grados de libertad con un actuador correspondiente a un gripper. Este robot se compone por 4 motores servo donde los primeros 3 controlan los grados de libertad (x,y,z) y el ultimo la pinza.

El cerebro del robot es la raspberry pi , la cual recibe los datos del servidor, computador donde se procesa toda la información para depues transmitirla mediante ROS de manera sencilla y asi ejecutar las acciones físicas.

2 Solución

Para la solución de este problema se definieron 5 nodos a desarrollar , cada uno encargado de resolver los requerimientos necesarios.

- robot_manipulator_listener
- robot_manipulator_teleop
- robot_manipulator_planer
- robot_manipulator_interface
- robot_manipulator_ping-pong

En rasgos generales, el nodo teleop envia las señales de teclado para el movimiento del manipulador, el listener ejecuta dichos comandos en el robot fisico, el nodo planer genera y ejecuta una ruta de cinematica inversa , el nodo interface grafica la posición actual del gripper y el nodo ping pong es utilizado para el analisis de imagen. A continuación se describen a detalle cada uno de los nodos.

2.1 Nodo Manipulator Teleop

Se creo un nodo publisher llamado */robot_manipulator_teleop* (ver codigo adjunto) el cual permite al usuario controlar el manipulador del robot mediante el uso del teclado. Este nodo se inicializa mediante la libreria *rospy* e incluye un listener que se encarga de detectar las pulsaciones del teclado , dicho listener proviene de la libreria *keyboard*.

Este nodo cuenta con dos funciones , la primer funcion es definida como *talker()* funcion que se inicia con el main y se encarga de recibir la informacion por consola de la velocidad angular de cada uno de los joints, de igual manera inicia el listener *keyboard.hook(key_press)* el cual llama a la segunda funcion cada vez que se presiona o suelta una tecla y finalmente publica el mensaje de tipo *String* el cual contiene la informacion del motor , direccion y velocidades del actuador .Esta segunda funcion es definida como *key_press(key)* cumple el papel de leer la informacion de la tecla presionada y a partir de ella construir el mensaje *String*.Este mensaje sigue la siguiente estructura *motor, direccion, velA, velB, VelC* donde la variable motor corresponde al motor a mover (a,b,c,d), la variable direccion a la dirección del motor(-1,1) y las velocidades a la velocidad angular de cada motor.

Las velocidades recibidas corresponden al numero de ángulos que cada uno de los servos se va a desplazar con cada pulsación θ/ref .A continuación se muestra el diagrama de flujo del funcionamiento del nodo.

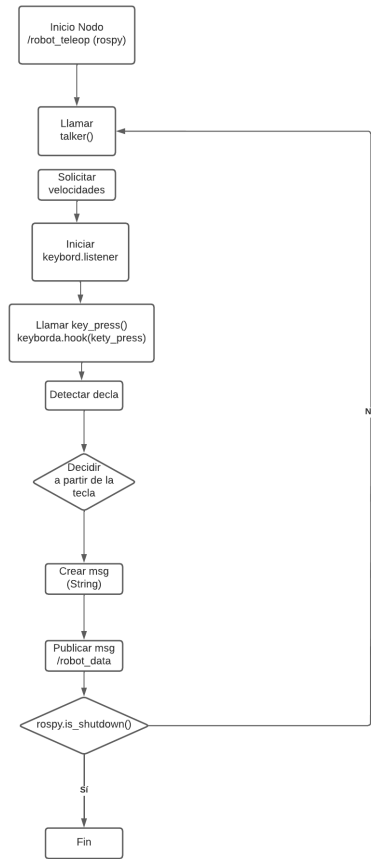


Figure 4: Diagrama */robot_manipulator_teleop*

2.2 Nodo manipulator Interface

Se creó el nodo */robot_interface* de tipo suscribir, el cuál recibe la posición del nodo teleop y se encarga de graficarla en tiempo real la posición del endeffector en el marco global de referencia (x, y, θ) . Este nodo se suscribe a un topico de tipo *String* para los datos de ángulo de los motores. Datos que son escritos por el nodo teleop.

El nodo interface recibe la información de el ángulo de los motores y realiza la conversión a coordenadas cilíndricas para calcular la posición del endeffector para después transformarlas al marco de referencia local. Posteriormente se crearon 3 arreglos donde se guardan las coordenadas x,y,z.

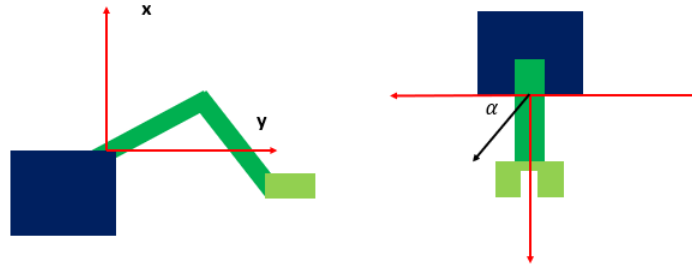


Figure 5: Marco global de referencia

Para graficar se utiliza un plot que se va actualizando cada vez que cambia la posición y este se muestra en una ventana creada con tkinter en donde se muestran las posiciones previas. A continuación se muestra los resultados obtenidos.

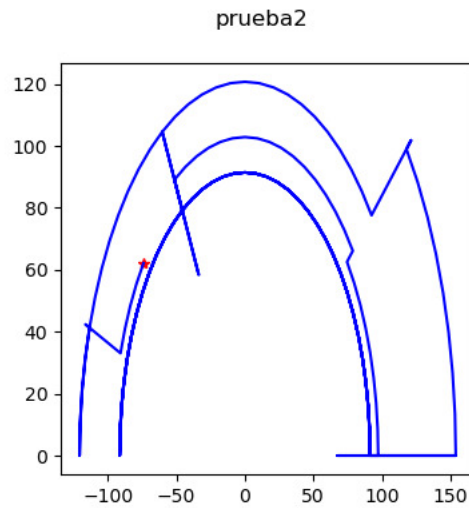


Figure 6: Trayectoria

A continuación se muestra el diagrama de flujo del nodo.

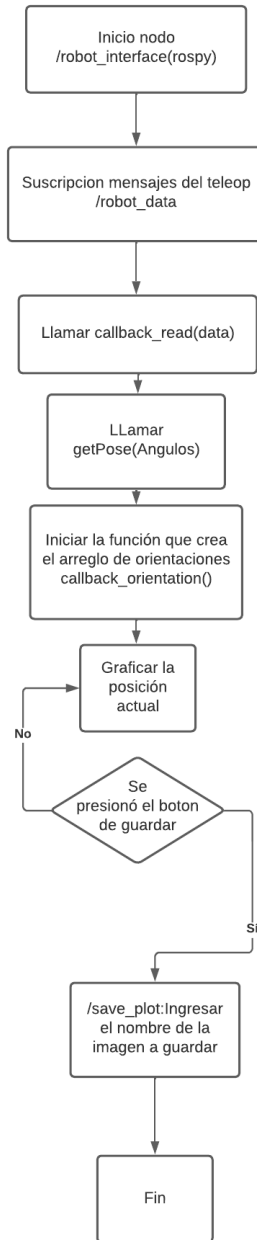


Figure 7: Diagrama */robot_manipulador_interface*

2.3 Nodo manipulator planer

Se creo el nodo */robot_manipulator-planer* el cual toma un punto meta (x, y, θ) dentro del volumen definido a continuación y planea la trayectoria para alcanzar dicho punto, finalmente la ejecuta y la publica en el topico */robot_manipulator_goal*.

Este volumen esta compuesto por un plano en x,y el cual es rotado alrededor de z generando el solido de revolucion.

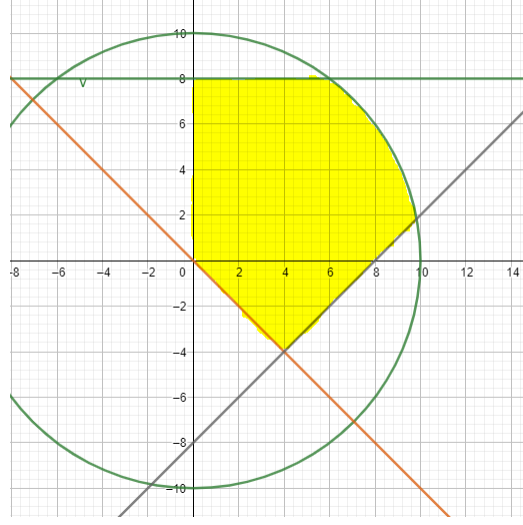


Figure 8: Volumen definido

Para realizar el planeo de trayectoria se tiene un código de cinemática inversa el cual calcula el ángulo de cada motor con respecto al eje de coordenadas en el marco de referencia, para así finalmente setear el robot en dicha posición.

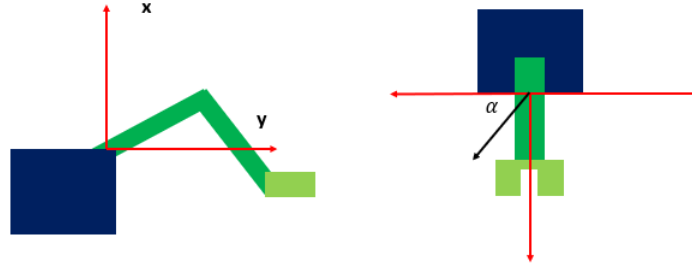


Figure 9: Marco de referencia

Para calcular los angulos se utilizaron las siguientes ecuaciones de cinematica:

$$\begin{aligned}\theta_A &= \theta \\ \theta_C &= -\cos^{-1}\left(\frac{x^2 + y^2 + a_1^2 + a_2^2}{2a_1a_2}\right) \\ \theta_B &= \tan^{-1}\left(\frac{y}{x}\right) + \tan^{-1}\left(\frac{a_2\sin(\theta_C)}{a_1 + a_2\cos(\theta_C)}\right)\end{aligned}$$

Donde:

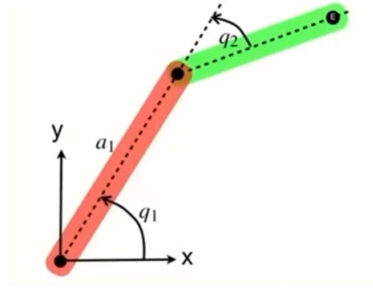


Figure 10: Modelo cinemática inversa

A continuación se muestra el diagrama de flujo del nodo creado.

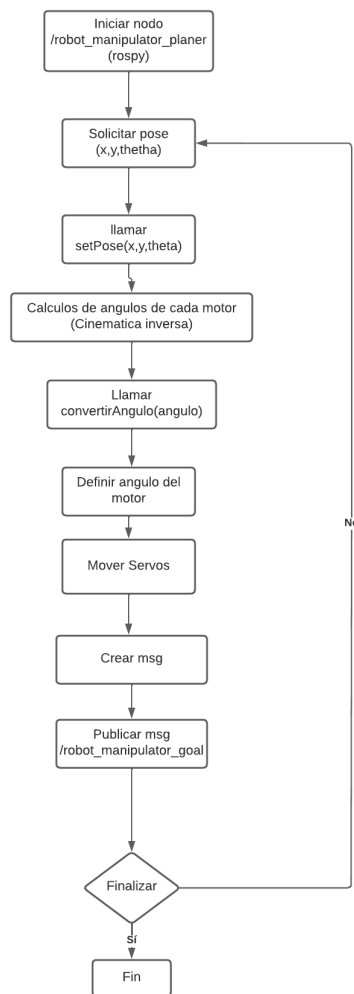


Figure 11: Diagrama */robot_manipulador_planer*

2.4 Nodo manipulator Listener

Adicionalmente se creo un nodo llamado *robot_manipulator_listener* el cual se encarga tomar la informacion del nodo *robot_manipulator_teleop* y traducirla al robo fisico mediante la *Raspberry* utilizando la libreria *RPi.GPIO* y *gpio.zerode* igual manera calcula y actualiza la posicion actual de cada una de las juntas del brazo.

El nodo cuenta con el main , donde se inicializa la funcion *listener()* en esta funcion se inicializa el nodo mediante *rospy*. La funcion principal de codigo es la funcion *callback_read(data)* en esta funcion se recibe la informacion de las velocidades de cada motor , asi como el motor y la direccion en la que se desea mover, una vez se tiene esta infromacion se calcula la señal PWM que debe enviarse los servos mediante la funcion *convertirAngulo(angulo)*. Finalmente se verifica la direccion del moviento y se llaman la funcion *moverMotor(motor, dirrecion, velA, velB, velC)* las cuales definen la direccion de giro de cada motor y envian la señal fisica a cada uno de ellos con el fin de que lleguen al angulo objetivo.

A continuacion se presenta el diagrama del flujo del nodo listener.

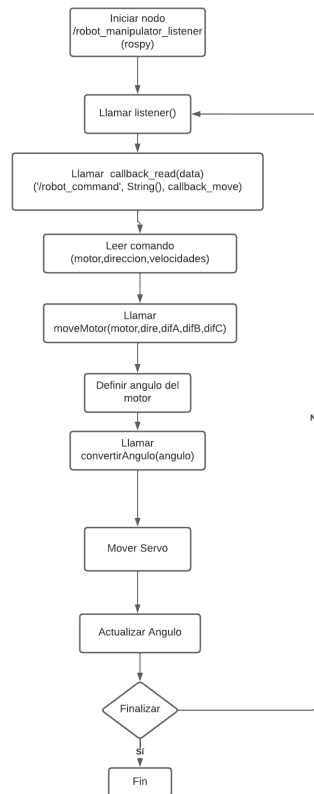


Figure 12: Diagrama /*robot_manipulator_listener*

2.5 Nodo robot_manipulator_ping_pong

Finalmente se creo el nodo `robot_manipulator_ping_pong` para realizar el análisis de imagen. Para este analisis se crearon las mascaras necesarias para dos requerimientos.

El primero es identificar el color objetivo, y el segundo identificar su contorno. Tras el desarrollo del código se hicieron diversas pruebas de análisis de imagen con el fin de identificar los ping pongs , a continuación se muestran los resultados.

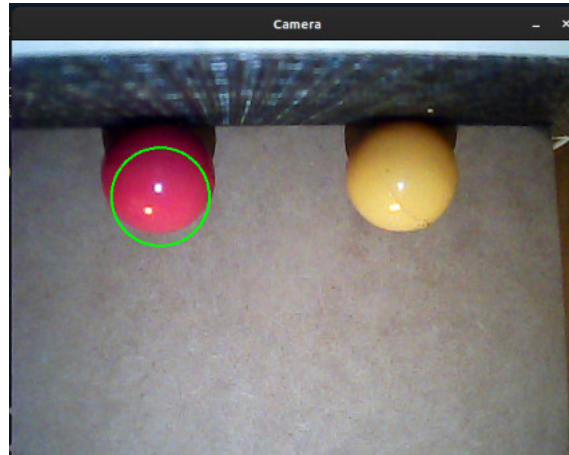


Figure 13: Identificación de imagen

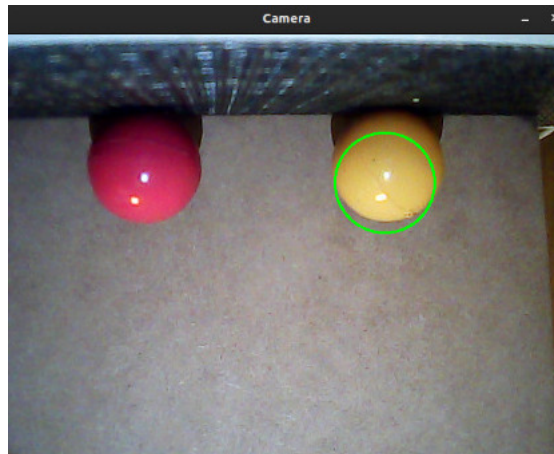


Figure 14: Identificación de imagen

2.6 Conexión entre nodos y tópicos

A continuación se muestra la iteración entre cada uno de los nodos y topicos. El primer diagrama rqt podemos observar como el nodo `teleop` publica la información al tópico `/robot_data`. A dicho topico se conectan los nodos de interface y listener con el fin de leer la información y ejecutar a partir de ella.



Figure 15: Diagrama RQT 1

En el siguiente diagrama rqt se observa como el nodo planer recibe la información de trayectoria y la publica en el tópico */robot_manipulator_goal..*

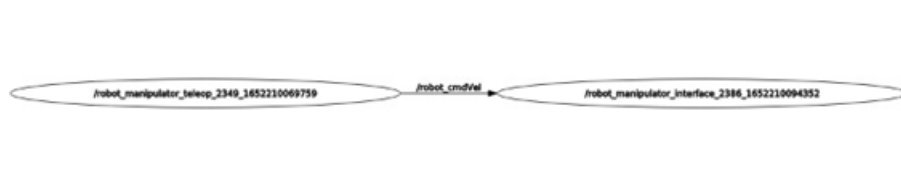


Figure 16: Diagrama RQT 2