Universidad Tecnológica Nacional
AJEA – Actas de Jornadas y Eventos Académicos de UTN
Trabajo Completo Publicado en AJEA – Full Text Published in AJEA
https://rtyc.utn.edu.ar/index.php/ajea/article/view/1567
Libro de actas de resúmenes DOI: https://doi.org/10.33414/ajea.1640.2024
Colección JIT en AJEA https://rtyc.utn.edu.ar/index.php/ajea/JITs

Jóvenes Investigadores Tecnológicos
Facultad Regional Rafaela 26 y 27 de octubre de 2023

Área: Aplicaciones mecánicas y mecatrónica / **Categoría:** Trabajo en desarrollo – estudiantes de grado / **Regional:** Facultad Regional Santa Fe

Actualización de un brazo manipulador robótico a partir de la implementación de ROS.

Update of a manipulator robot arm based on the implementation of ROS.

Presentación: 17/10/2023

Matías Maglianesi
Centro de I+D en Ingeniería Eléctrica y Sistemas Energéticos, Universidad Tecnológica Nacional, Facultad Regional Santa Fe, Argentina
mmaglianesi@frsf.utn.edu.ar

Resumen

Los robots manipuladores de uso comercial trabajan bajo sistemas cerrados, propietarios de cada empresa. El software ROS (Robot Operating System) se ha desarrollado con el propósito de facilitar la comunicación entre los diferentes componentes de robots a partir de librerías de código abierto. En el presente trabajo se busca realizar una actualización a un brazo manipulador didáctico controlado a través de una placa de desarrollo Arduino a partir de este software para mejorar el control y las capacidades del robot.

Palabras clave: Sistema operativo de robots, Brazo robot, Python, Arduino, Control

Abstract

Commercial use manipulator robots work under proprietary systems form each company. The ROS (Robot Operating System) software have been developed to facilitate the communication between different robot components using open-source libraries. In the present work we will seek to update a didactic manipulator arm controlled through a development Arduino board using this software to enhance the robot control and capabilities

Keywords: Robot operating system, Robot arm, Python, Arduino, Control

Introducción

El sector comercial de la robótica está dominado por sistemas cerrados donde se prioriza la dependencia del proveedor sobre la innovación. Cada robot dispone de su propio sistema operativo y lenguaje de programación. (Suárez et al, 2022). El sistema operativo de robots, llamado en inglés Robot Operating System (ROS), representa una gran contribución al mundo de la robótica. Es un software libre de código abierto que permite la comunicación y programación fluida de los diversos accionamientos y sensores que conforman un robot. Con más de 2000 paquetes de software, ROS se convierte en una plataforma versátil y colaborativa que no solo agiliza el desarrollo de robots, sino que también fomenta la colaboración entre investigadores, ingenieros y entusiastas. Con una compatibilidad que abarca aproximadamente 80 robots comerciales.

AJEA – Actas de Jornadas y Eventos Académicos de UTN
Libro de actas de resúmenes DOI: https://doi.org/10.33414/ajea.1640.2024
Texto Completo Publicado en AJEA – Full Text Published in AJEA
https://rtyc.utn.edu.ar/index.php/ajea/article/view/1567

El laboratorio de sistemas de control (LSC) de la Universidad Tecnológica Nacional, Facultad Regional Santa Fe, posee un robot manipulador didáctico en tres dimensiones, controlado por una placa Arduino. En la Figura 1, se puede apreciar el robot manipulador.

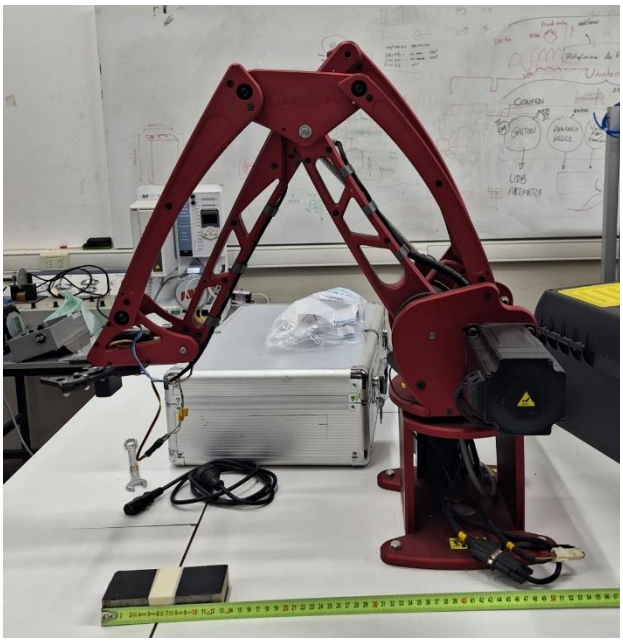

Figura 1 – Brazo manipulador sobre el que se trabajó

Este robot consta de 3 motores paso a paso, los cuales poseen codificadores rotatorios incorporados. Para su operación cada motor está conectado a un controlador propietario, el cual se encarga de gobernar los pasos del motor a partir de la entrada de un tren de pulsos que representa la cantidad de pasos a realizar (curva superior de la figura 2) y una señal la cual indica el sentido de giro deseado (curva de dirección de la figura 2). Finalmente, en el extremo del brazo se tiene un servomotor que permite el agarre de objetos.

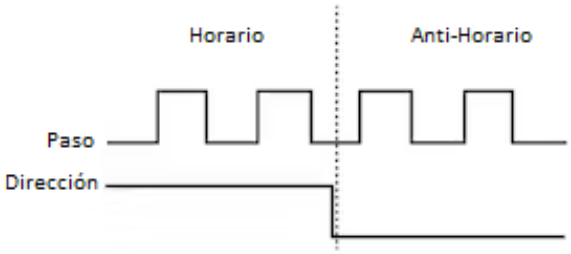

Figura 2 – Esquema de control de los motores paso a paso

El control se realiza a lazo abierto, ya que el controlador no dispone de ninguna salida de información, adicionalmente en el programa original los movimientos se realizaban a partir de la operación manual de un mando incorporado, ya que el objetivo del robot es didáctico, y la precisión es un elemento secundario.

El fabricante de este robot ya no se encuentra en el mercado, por lo que se buscó realizar una actualización de este manipulador a partir de la implementación de la distribución de ROS Noetic Ninjemys para la ampliación de sus funcionalidades, como es la implementación de rutinas preprogramadas.

AJEA – Actas de Jornadas y Eventos Académicos de UTN
Libro de actas de resúmenes DOI: https://doi.org/10.33414/ajea.1640.2024
Texto Completo Publicado en AJEA – Full Text Published in AJEA
https://rtyc.utn.edu.ar/index.php/ajea/article/view/1567

Metodología

ROS opera mediante la estructura fundamental de tópicos (topics), que permite la interacción entre diversos dispositivos al posibilitar el intercambio de información a partir de la publicación y suscripción a estos tópicos. Para comenzar la modernización del manipulador se planteó qué señales se tienen que registrar y enviar a través de la placa Arduino para poder operar a este. Los resultados pueden observarse en la figura 3.

Figura 3 – Mapa de entradas y salidas del controlador.

Los resultados obtenidos en el análisis previo sientan las bases para la planificación de los nodos de comunicación que serán implementados en el sistema. En la fase inicial de desarrollo, el enfoque se dirige hacia los componentes esenciales de movimiento, dejando de lado el control externo de variables como velocidad, posición mediante mando analógico, funciones extra activadas por botones, indicador sonoro, señalización lumínica y accionamiento de los servomotores accesorios. El énfasis recae exclusivamente en el dominio del control de posición a través de los 3 motores paso a paso que controlan el robot.

Para la gestión de la posición objetivo, se introduce un tópico denominado "pos_deseada", donde tanto el usuario como el programa de la rutina tienen la capacidad de publicar la posición deseada de los motores paso a paso. Una placa de desarrollo Arduino Mega y su microcontrolador Atmega 328 es la encargada del control global del robot, se suscribe a este tópico y, mediante la comparación entre su posición actual y la posición deseada, inicia el accionamiento de los motores pertinentes. Este proceso se desarrolla moviéndose a una velocidad predeterminada que está preestablecida en la placa. De manera simultánea, el Arduino también publica su posición actual en un tópico denominado "pos_real", lo cual proporciona una retroalimentación acerca del estado de los motores, lo que permite determinar la llegada a la posición objetivo.

La ubicación en el espacio es posible obtenerla a partir de la posición de las diferentes junturas, esto se conoce como cinemática directa. Esto se puede lograr a partir de aplicar algebra matricial para obtener la transformación la cual asocia la posición del extremo de un eslabón a la base de otro. Para el proyecto, la opción por la que se optó fue la herramienta incorporada en ROS llamada RViz. la cual permite realizar estos cálculos matriciales, y graficar la posición a partir de los ángulos publicados, a partir de un archivo de texto el cual define todas las características del robot, entre estas se encuentran la forma y tamaño de todos los eslabones y el tipo de juntas que los unen.

Para la obtención precisa de la posición real, se ejecuta una rutina de calibración al inicio cada vez que el manipulador se activa, o por medio del tópico “calibración”. Esta rutina se encarga de localizar con precisión la ubicación física de los finales de carrera de cada motor, logrando así establecer la referencia espacial del brazo robótico con respecto a su base fija, y por consiguiente al entorno circundante. Este proceso de calibración inicial es esencial para garantizar un funcionamiento confiable y preciso del manipulador en sus tareas y operaciones subsiguientes. Otro factor para la implementación de esta calibración se da por la deficiente calidad constructiva del robot, la cual permite mantener esta precisión por medio de subsiguientes calibraciones tras realizar múltiples operaciones.

Durante la operación habitual, se debe asegurar que el robot se mueva dentro de los límites establecidos, tanto de manera física por la posición de los finales de carrera, como los que surgen por el espacio de trabajo, como lo son la mesa donde está colocado. Para esto se implementan controles que eviten que se acerque a zonas prohibidas, como paradas de emergencia que detienen la ejecución del robot en caso de que entre a estas. Inicialmente el control de posición se realiza sobre los pasos de los diferentes motores, y no la posición en el espacio, esto lleva a que las limitaciones se impongan en el programa al no permitir que se continúen realizando pasos al estar cercano a estos límites. Para lograr

AJEA – Actas de Jornadas y Eventos Académicos de UTN
Libro de actas de resúmenes DOI: https://doi.org/10.33414/ajea.1640.2024
Texto Completo Publicado en AJEA – Full Text Published in AJEA
https://rtyc.utn.edu.ar/index.php/ajea/article/view/1567

Resultados y discusión

Después de llevar a cabo la fase inicial de implementación, se logró exitosamente establecer una comunicación efectiva del robot a través de los tópicos definidos en la sección anterior. Este avance permitió ejercer un control sobre los movimientos del robot al publicar manualmente los pasos necesarios. La representación visual del flujo de información se encuentra plasmada en la figura 4.

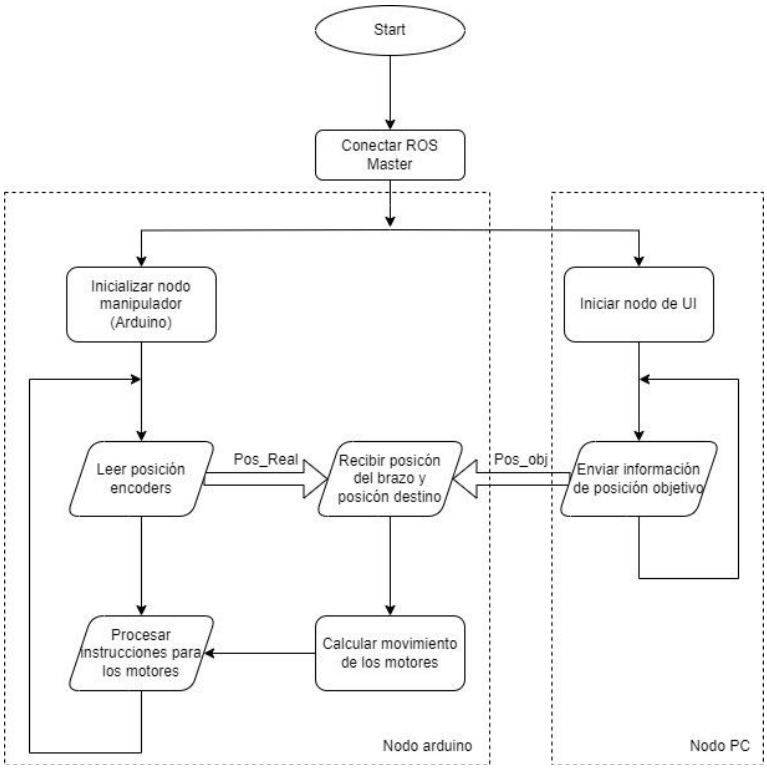

Figura 4 – Diagrama de flujo de las acciones.

En una segunda instancia se optó por emplear Python, el cual, gracias a su amplia adopción, cuenta con un gran rango de librarías compatibles, las cuales posibilitaron realizar rutinas cíclicas más complejas, preparando el manipulador para implementar funciones de mayor nivel en instancias futuras.

Conclusiones

ROS, junto con la posibilidad de la adopción de Python como lenguaje de programación, permiten un control de gran precisión y personalización en el robot manipulador. Para este robot se logró mejorar la interfaz de control y la retro alimentación que recibe el usuario.

En las fases futuras del proyecto, la integración con Python abre oportunidades para incorporar capacidades de procesamiento de imágenes. Este avance permitiría que el manipulador identifique su objetivo y se desplace hacia este, o la detección de obstáculos de manera eficiente.

AJEA – Actas de Jornadas y Eventos Académicos de UTN
Libro de actas de resúmenes DOI: https://doi.org/10.33414/ajea.1640.2024
Texto Completo Publicado en AJEA – Full Text Published in AJEA
https://rtyc.utn.edu.ar/index.php/ajea/article/view/1567

Referencias bibliografías

Gerkey, B., Quigley, M., & Smart, W. D. (2015). *Programming Robots with ROS: A Practical Introduction to the Robot Operating System*. O'Reilly Media, Incorporated.

Open Robotics. (s.f.). ROS. https://www.ros.org/

Suárez et ál. (2022). *Robot Operating System (ROS)*. Grupo de Trabajo de Innovación de la Asociación Española de Robótica y Automatización (AER).. Accessed: Nov, 28.