

Middleware entre ROS 2 y ACL para el control de robots SCORBOT-ER V+

Pérez García, M., Muñoz-Rodríguez, M.*, Moreno Úbeda, J.C., Berenguel Soria, M.

Departamento de Informática, ceiA3, CIESOL, Universidad de Almería, Ctra. Sacramento s/n, La Cañada de San Urbano, Almería, 04120, España.

Resumen

El robot SCORBOT-ER V+ ha sido ampliamente utilizado en entornos educativos para la enseñanza de la robótica industrial. No obstante, su sistema de programación basado en ACL presenta limitaciones en cuanto a escalabilidad e integración con tecnologías modernas. Este trabajo propone una nueva arquitectura de control basada en ROS 2 (Robot Operating System 2), que sustituye el entorno ACL manteniendo el hardware original. El sistema incorpora nodos ROS 2 para el control del robot, la gestión de periféricos y la monitorización de sensores, comunicados mediante el middleware DDS (Data Distribution Service). La solución fue validada en un entorno académico real, permitiendo el control distribuido y la ejecución de prácticas colaborativas. Además de mejorar la funcionalidad técnica, esta arquitectura proporciona a los estudiantes un entorno de aprendizaje alineado con los estándares de la Industria 4.0, fomentando competencias en programación modular, integración de sistemas y comunicación distribuida. La plataforma es extensible y replicable, contribuyendo a la modernización y reutilización de sistemas educativos existentes.

Palabras clave: Robótica educativa, ROS 2, Middleware DSS, integración de sistemas

Middleware between ROS 2 and ACL for the control of SCORBOT-ER V+ robots

Abstract

The SCORBOT-ER V+ robot has been widely used in educational environments for teaching industrial robotics. However, its programming system based on ACL presents limitations in terms of scalability and integration with modern technologies. This paper proposes a new control architecture based on ROS 2 (Robot Operating System 2), which replaces the original ACL environment while preserving the hardware. The system incorporates ROS 2 nodes for robot control, peripheral management, and sensor monitoring, communicating via the DDS (Data Distribution Service) middleware. The solution was validated in a real academic setting, enabling distributed control and collaborative practical sessions. In addition to improving technical functionality, this architecture provides students with a learning environment aligned with Industry 4.0 standards, promoting skills in modular programming, system integration, and distributed communication. The platform is extensible and easily replicable, contributing to the modernization and reuse of existing educational robotics systems.

Keywords: Educational Robotics, ROS 2, Middleware DSS, systems integration

1. Introducción

La robótica se ha convertido en una herramienta fundamental tanto en el ámbito industrial como en el educativo. En particular, el desarrollo de sistemas robóticos modulares, conectados y flexibles es esencial para dar respuesta a los requisitos de la llamada Industria 4.0. En este contexto, la actualización de entornos tradicionales de robótica educativa representa una necesidad clave para garantizar que los futuros ingenieros puedan enfrentarse a sistemas reales basados en arquitecturas distribuidas, interoperables y fácilmente escalables.

El robot SCORBOT-ER V+ es un manipulador ampliamente utilizado en entornos formativos por su facilidad de uso y configuración mecánica. Sin embargo, su sistema de control basado en el lenguaje propietario ACL (Advanced Control Language) presenta limitaciones considerables. ACL requiere software y hardware obsoletos (como ejecutables diseñados para Windows XP y entornos Java antiguos), carece de capacidad para integrarse con otros dispositivos en red y no permite una programación modular o reutilizable. Estas restricciones hacen que el SCORBOT, pese a su valor como plataforma didáctica,

*Autor para correspondencia: mmm411@ual.es

ca, no pueda aprovecharse plenamente en entornos tecnológicos actuales.

Una revisión de la literatura actual (Deshpande and George, 2012; Balestrino et al., 2004) pone de manifiesto que la mayoría de los trabajos que utilizan el SCORBOT en entornos docentes o de investigación siguen apoyándose en ACL como único lenguaje de control, lo que limita las posibilidades de innovación, expansión o reutilización del hardware en nuevos contextos. A diferencia de robots con soporte nativo en plataformas como ROS o ROS 2, el SCORBOT-ER V+ queda aislado de muchos de los avances recientes en robótica colaborativa, simulación, visión artificial o integración con sistemas ciberfísicos.

Para superar estas limitaciones, se propone en este trabajo la sustitución del lenguaje ACL por una arquitectura de control basada en ROS 2 (Robot Operating System 2). ROS 2 es un framework de desarrollo robótico de código abierto diseñado para satisfacer requisitos industriales como comunicación distribuida, interoperabilidad, seguridad y soporte en tiempo real. La migración a ROS 2 permite transformar una célula robótica cerrada en un sistema flexible, compatible con estándares abiertos y fácilmente ampliable con sensores, visión artificial u otras herramientas. ROS 2 ha sido ampliamente adoptado en entornos de investigación y comienza a consolidarse como herramienta educativa en robótica gracias a su arquitectura modular y distribuida (Lages, 2022; Horváth et al., 2024).

Desde el punto de vista funcional, el sistema desarrollado permite controlar el SCORBOT-ER V+ y sus periféricos desde nodos ROS 2 escritos en Python, facilitando su integración en entornos distribuidos. Esta solución mantiene el hardware original, pero reemplaza completamente el entorno de programación, lo que extiende significativamente su vida útil y posibilidades de uso.

Además del interés técnico de esta migración, la propuesta tiene un claro valor educativo, ya que permite a los estudiantes familiarizarse con tecnologías y herramientas actuales de desarrollo robótico. La arquitectura basada en ROS 2 favorece el aprendizaje de conceptos como la comunicación middleware, el diseño modular de sistemas, la integración de sensores y actuadores, y el trabajo colaborativo en entornos conectados. Por tanto, este sistema no solo resuelve las limitaciones del entorno ACL, sino que mejora la calidad formativa de las prácticas de robótica en el ámbito universitario.

En este artículo se presenta el desarrollo técnico del sistema propuesto. En la Sección 2 se describen los materiales y métodos utilizados, incluyendo el diseño de la arquitectura y su implementación. La Sección 3 expone los resultados obtenidos tras la validación del sistema en un entorno de laboratorio real. Finalmente, la Sección 4 recoge las principales conclusiones y las líneas futuras de mejora.

2. Materiales y métodos

2.1. Hardware utilizado

El sistema es desarrollado sobre una célula robótica educativa basada en el robot SCORBOT-ER V+, un manipulador de cinco grados de libertad más pinza (ver Figura 1(b)), controlado mediante un controlador industrial de la marca Eshed Robotec (ver Figura 1(a)). Este equipo se encuentra ampliamente distribuido en entornos universitarios por su facilidad de

uso, robustez y disponibilidad de periférico. El controlador del SCORBOT dispone de salidas y entradas digitales que permiten la conexión de actuadores y sensores externos. En esta implementación se integraron los siguientes dispositivos adicionales:

- Cinta transportadora, activada a través de una salida digital, utilizada para simular el movimiento de piezas en la célula.
- Mesa giratoria, accionada por un motor paso a paso controlado por el SCORBOT, que permite modificar la orientación de las piezas.
- Sensores digitales (microinterruptores y sensores fotoeléctricos), conectados a entradas del controlador, usados para la detección de piezas y eventos.

La comunicación entre el PC y el controlador del robot se realiza mediante puerto serie RS232, lo que permite enviar comandos ACL directamente desde un equipo externo.

2.2. Arquitectura del sistema

La nueva arquitectura (ver Figura 2) se basa en una implementación modular desarrollada sobre ROS 2 Foxy Fitzroy, organizada en nodos independientes que se comunican mediante el middleware DDS. Esta arquitectura facilita la ejecución distribuida, la integración progresiva de nuevos dispositivos y la reutilización de componentes.

El sistema se compone de los siguientes niveles funcionales:

- Nivel de control del robot: incluye un nodo principal (bridge) que recibe comandos de movimiento y los traduce a instrucciones ACL que son enviadas al SCORBOT vía RS232.
- Nivel de periféricos: incluye nodos encargados de accionar la cinta, la mesa giratoria y de leer los sensores conectados al controlador.
- Nivel de interacción: formado por nodos opcionales para interfaces gráficas, terminales de usuario, teleoperación o integración futura con visión artificial.

La arquitectura distribuida permite ejecutar nodos en diferentes equipos conectados en red, favoreciendo la escalabilidad del sistema y la colaboración entre estudiantes en prácticas multipuesto. La arquitectura basada en ROS 2 permite diseñar sistemas educativos distribuidos y escalables, alineados con los principios de Educación 4.0 (Moraes et al., 2023; Pozo et al., 2022).

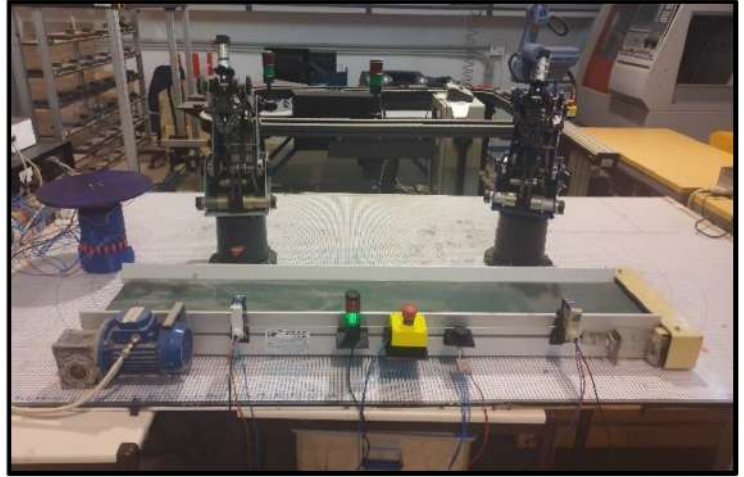
2.3. Implementación del software

El software fue desarrollado en Python 3, utilizando la API `rclpy` de ROS 2 para la gestión de nodos, tópicos, servicios y acciones. Además, se utilizó la librería `pyserial` para establecer la comunicación con el controlador a través del puerto serie. La modularidad del sistema y la implementación basada en ROS 2 permiten una estructura educativa escalable, como se ha aplicado también en kits de formación en robótica (Almeida et al., 2024).

El nodo principal se encarga de:



(a) Puesto de control



(b) Puesto manipulador

Figura 1: Célula robótica final.

- Suscribirse a comandos de movimiento articulado o cartesiano.
- Traducirlos al formato ACL correspondiente.
- Enviar las instrucciones al controlador SCORBOT-ER V+.
- Escuchar y publicar respuestas o estados, si son requeridos.

Los nodos secundarios se encargan de:

- Activar o desactivar la cinta transportadora.
- Posicionar la mesa giratoria en ángulos predeterminados.
- Detectar el estado de los sensores (por ejemplo, presencia de pieza).
- Publicar eventos en ROS 2 para ser utilizados en lógica de control distribuida.
- La estructura de nodos modular permite una programación flexible y la ejecución paralela de tareas, replicando el comportamiento de sistemas industriales modernos.

A continuación, se muestra un ejemplo representativo de la función desarrollada en Python para traducir un comando ROS 2 a una instrucción ACL enviada al SCORBOT-ER V+ a través del puerto serie:

```
def move_joint(joint, value):
    acl_cmd = f"MOVE {joint},{value}\r"
    serial_port.write(acl_cmd.encode())
```

2.4. Entorno de desarrollo

Las pruebas del sistema se realizaron sobre equipos con Ubuntu 20.04 LTS, con ROS 2 Foxy instalado desde paquetes oficiales. El código fue organizado en paquetes ROS estándar, con dependencias mínimas para facilitar su despliegue en distintos equipos.

Las conexiones físicas entre el SCORBOT y el ordenador se establecieron mediante un adaptador USB-RS232. El sistema fue probado en modo local (todos los nodos en un solo PC) y en modo distribuido (con nodos ejecutados desde diferentes equipos conectados por red interna), sin requerir cambios en el código base.

La validación incluyó tanto pruebas de funcionamiento autónomo como prácticas cooperativas entre estudiantes, que ejecutaban y controlaban distintos nodos simultáneamente.

3. Resultados

La arquitectura desarrollada fue evaluada mediante una serie de pruebas funcionales realizadas en el laboratorio docente de la Universidad de Almería, utilizando el robot SCORBOT-ER V+ original, sus periféricos, y un entorno de prácticas real. El objetivo de estas pruebas fue verificar la correcta ejecución de los comandos desde ROS 2, la integración efectiva de los periféricos, y la capacidad del sistema para ser operado de forma distribuida y colaborativa.

3.1. Validación funcional

El nodo principal de control ROS 2-ACL demostró una comunicación estable con el controlador del SCORBOT, permitiendo el envío de comandos de movimiento (articulares y cartesianos) y la ejecución de rutinas básicas como apertura/cierre de la pinza o retorno a posición inicial. La latencia en la transmisión de comandos fue mínima y no se observaron pérdidas de mensajes o bloqueos en el flujo de control.

En la Figura 3 se muestran las interfaces utilizadas durante la validación funcional del sistema. A la izquierda se aprecia el ejecutable Python que permite enviar comandos al robot, mientras que en el centro y derecha se observan los nodos ROS 2 nodo1 y nodo2, responsables del control distribuido de periféricos.

Los nodos de periféricos permitieron encender y apagar la cinta transportadora y accionar la mesa giratoria con precisión.

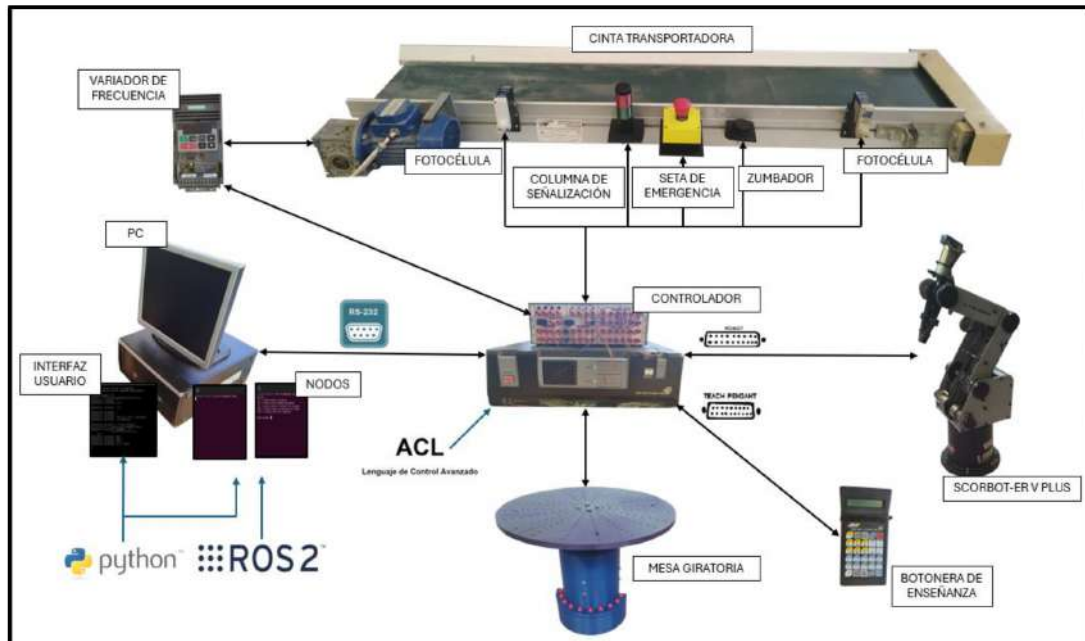


Figura 2: Esquema de la arquitectura de comunicación entre ROS 2, el nodo Python y el controlador ACL del SCORBOT-ER V+.

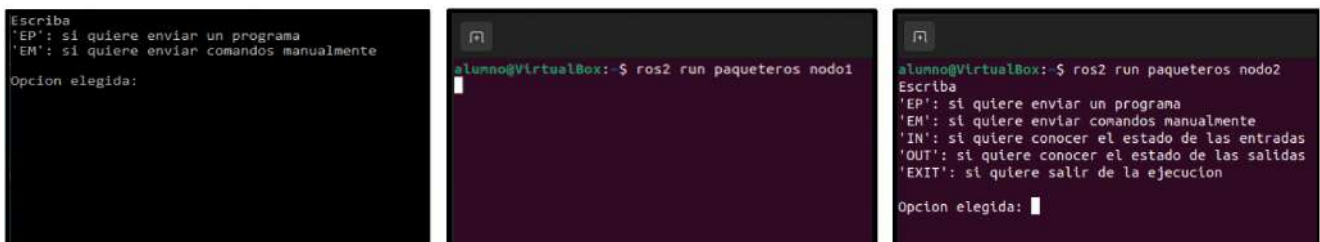


Figura 3: Interfaces utilizadas durante la validación: ejecutable Python (izquierda), nodo1 (centro) y nodo2 (derecha) ejecutándose en ROS 2.

La lectura de sensores digitales se realizó correctamente, publicando eventos en tiempo real mediante tópicos ROS 2, lo que permitió sincronizar acciones entre distintos nodos (por ejemplo, detener la cinta al detectar una pieza).

3.2. Ejecución de prácticas

El sistema fue utilizado para desarrollar prácticas formativas donde distintos estudiantes controlaban los componentes de la célula desde nodos separados:

- Un alumno programaba el nodo del brazo robótico.
- Otro gestionaba los periféricos (cinta y mesa).
- Un tercero leía sensores y supervisaba el sistema.

Estas prácticas demostraron el potencial del sistema para fomentar el trabajo colaborativo y la comprensión del flujo de datos entre procesos concurrentes. La arquitectura modular permitió modificar o reiniciar componentes sin afectar al conjunto, lo que favorece la exploración segura del entorno por parte del alumnado. A diferencia del entorno ACL original, la solución propuesta promueve competencias en comunicación distribuida y control colaborativo, pilares de los programas educativos modernos basados en Industria 4.0 (Ahmad et al., 2024).

3.3. Comparativa frente al entorno ACL

A diferencia del entorno ACL original, donde la programación se realizaba de forma secuencial y aislada, la nueva arquitectura permitió una programación reactiva y distribuida. Los estudiantes no solo ejecutaron movimientos, sino que diseñaron flujos lógicos con decisiones condicionadas por sensores o eventos del sistema.

Además, el sistema desarrollado eliminó la necesidad de utilizar software obsoleto (como ATS sobre Windows XP), facilitando la instalación en cualquier equipo moderno con Linux. Esto redujo significativamente los tiempos de preparación y puesta en marcha del sistema, así como los problemas derivados de compatibilidad o estabilidad.

4. Conclusiones

Este trabajo ha demostrado la viabilidad de actualizar una célula robótica educativa tradicional, basada en el robot SCORBOT-ER V+, mediante la integración de una arquitectura moderna de control basada en ROS 2. La solución desarrollada permite mantener el hardware existente, pero sustituir completamente el entorno de programación legado (ACL) por un sistema abierto, modular y escalable.

Desde el punto de vista técnico, la arquitectura diseñada ha permitido:

- Controlar el robot y sus periféricos mediante nodos ROS 2 escritos en Python.
- Establecer una comunicación estable y funcional entre ROS 2 y el controlador original del SCORBOT.
- Diseñar un sistema distribuido, con nodos independientes y reutilizables, capaces de ejecutarse en distintos equipos conectados en red.
- Integrar sensores, actuadores y lógica de control sin necesidad de modificar el firmware del robot.

En cuanto a su impacto práctico, el sistema desarrollado ha simplificado la puesta en marcha de las prácticas, ha eliminado la dependencia de software obsoleto y ha abierto la puerta a futuras ampliaciones, como la incorporación de visión artificial o el trabajo con múltiples células robóticas conectadas entre sí.

Además de su valor técnico, esta propuesta tiene una clara proyección en el ámbito educativo. La migración a ROS 2 permite a los estudiantes trabajar en un entorno realista y actual, alineado con los requerimientos de la Industria 4.0. Los alumnos no solo aprenden a mover un robot, sino que adquieren experiencia en programación distribuida, integración de sistemas, gestión de eventos y trabajo colaborativo en red. Esto refuerza competencias clave en automatización, control y diseño de soluciones robóticas.

Agradecimientos

Este trabajo se ha realizado en el marco del proyecto LIFE-ACCLIAMTE (LIFE23-CCA-ES-LIFE-ACCLIMATE/101157315), cofinanciado por el Programa LIFE de la Unión Europea, Proyecto CyberGreen, PID2021-

1225600B-I00, y las instalaciones 'AgroConnect.es', subvención EQC2019-006658-P, ambos financiados por MCIN/AEI/10.13039/501100011033 y por FEDER A way to make Europe.

Referencias

- Ahmad, I., Sharma, S., Singh, R., Gehlot, A., Gupta, L. R., Thakur, A. K., Priyadarshi, N., Twala, B., 2024. Inclusive Learning Using Industry 4.0 Technologies: Addressing Student Diversity in Modern Education. *Cogent Education* 11 (1), 2330235.
DOI: <https://dx.doi.org/10.1080/2331186X.2024.2330235>
- Almeida, F., Leão, G., Sousa, A. J., 2024. An Educational Kit for Simulated Robot Learning in ROS 2. In: *Proceedings of the 10th Iberian Robotics Conference (ROBOT 2023)*. Coimbra, Portugal.
DOI: https://doi.org/10.1007/978-3-031-59167-9_42
- Balestrino, A., Bicchi, A., Caiti, A., Cecchini, T., Pallottino, L., Pisani, A., Tonietti, G., 2004. A Robotic Set-Up With Remote Access for 'Pick and Place' Operations Under Uncertainty Conditions. In: *Proceedings of the 1st Int. Workshop on e-Learning and Virtual and Remote Laboratories (VIRTUAL-LAB)*. pp. 144–149.
- Deshpande, V. A., George, P. M., 2012. Analytical Solution for Inverse Kinematics of SCORBOT-ER V+ Robot. *International Journal of Emerging Technology and Advanced Engineering* 2 (3), 478–482.
- Horváth, E., Ignézi, G., Markó, N., Krecht, R., Unger, M., 2024. Teaching Aspects of ROS 2 and Autonomous Vehicles. *Engineering Proceedings* 79 (1), 49.
DOI: <https://doi.org/10.3390/engproc2024079049>
- Lages, W. F., 2022. Remote Teaching of Dynamics and Control of Robots Using ROS 2. *IFAC-PapersOnLine* 55 (17), 279–284.
DOI: <https://doi.org/10.1016/j.ifacol.2022.09.292>
- Moraes, E. B., Kipper, L. M., Kellermann, A. C. H., Austria, L., Leivas, P., Moraes, J. A. R., Witczak, M. V. C., 2023. Integration of Industry 4.0 Technologies with Education 4.0: Advantages for Improvements in Learning. *Interactive Technology and Smart Education* 20 (2), 271–287.
DOI: <https://dx.doi.org/10.1108/ITSE-11-2021-0201>
- Pozo, E., Patel, N., Schrödel, F., 2022. Collaborative Robotic Environment for Educational Training in Industry 5.0 Using an Open Lab Approach. *IFAC-PapersOnLine* 55 (17), 314–319.
DOI: <https://doi.org/10.1016/j.ifacol.2022.09.298>