Global Journal of Engineering and Technology Advances
eISSN: 2582-5003
Cross Ref DOI: 10.30574/gjeta
Journal homepage: https://gjeta.com/

(RESEARCH ARTICLE)

Check for updates

Design, Simulation, and Implementation of a 6-DOF Robotic Arm Using ROS2 and RViz with Hybrid CAD-Based Architecture

Sergio A. Galván-Chávez *, Edgar M. Gaona-García, Ariadna Barrera-González, JL Díaz-Huerta and Febe Jocabed Zavala-Mendoza

Department of Electrical and Electronic Engineering/ Technological Institute of Morelia, Morelia, Mexico.

Global Journal of Engineering and Technology Advances, 2025, 25(02), 144–155

Publication history: Received 08 October 2025; revised on 15 November 2025; accepted on 18 November 2025

Article DOI: https://doi.org/10.30574/gjeta.2025.25.2.0333

Abstract

This paper presents the design, modeling, and simulation of a hybrid mechanical six-degree-of-freedom (6-DOF) robotic arm that integrates off-the-shelf parts and purpose-designed elements. The mechanical design was developed in SolidWorks and converted to URDF (Unified Robot Description Format) for use in ROS2 (Robot Operating System 2). The robotic arm's control system was built on a ROS2 node for modularity and direct real-time communication. The motion and behavior of the robotic arm had been validated using RViz (ROS Visualization), a visual and analytical validation environment. Several trajectory planning workflow scenarios were tested to validate the system, all of which demonstrated sufficient certainty levels.

Keywords: Robot Simulation; Robot Operating System; Motion Planning; Educational Robotics

1. Introduction

Traditionally, the development and validation of robotic manipulators have imposed access to expensive hardware equipment, intricate control systems and niche-development environments [1]. Recent advancements in open-source robotic platforms and low-cost 3D modeling software are helping to reduce these barriers, lowering the costs associated with the rapid prototyping and testing of robotic systems [2].

Among available platforms, Robot Operating System 2 (ROS2) is a robust framework for developing modular robotic systems. Its improved communication and real-time capabilities position it perfectly for advanced robotic tasks. When combined with the visualization capabilities of tools like RViz and computer-aided design (CAD) software such as SolidWorks, ROS2 enables the quick prototyping and early verification of robotic models [3].

This paper proposes an innovative hybrid design for a 6-DOF robotic arm that uses commercial-off-the-shelf components and custom-manufactured parts. The arm was designed in SolidWorks and brought into the ROS2 ecosystem via a URDF file, enabling interactive simulation and visualization in RViz.

The objective of this work is to develop an inexpensive and reproducible robotic platform that can be utilized by students, and researchers for exploration and prototyping [4]. It provides a testbed for motion planning, joint control strategy and mechanical layout verification in an environment without hardware implementation.

The article details the full development workflow, including the mechanical design, ROS2 integration, and simulation outcomes. It also discusses the strengths and current limitations of the proposed method while outlining potential avenues for future development.

* Corresponding author: Sergio A. Galván-Chávez

Copyright © 2025 Author(s) retain the copyright of this article. This article is published under the terms of the Creative Commons Attribution License 4.0.

2. Methods

2.1. Related Work

Robotic arms with a greater number of degrees of freedom have been the object of attention in industry and academia for years. Over the past several years, there has been a push towards low-cost robotic arms, which are more relevant in an educational setting, as budget constraints are severe. For instance, S. Trubayev, E. Shehab and M. H. Ali [5] presented a 5-DOF robot manipulator with the aid of 3D printed parts assembled with servo motors. Their motivation was to offer a practical environment for educating embedded systems and control methodologies. Similarly, L. Čehovin Zajc et al. [6] developed an educational framework using affordable robotic platforms to introduce students to industrial robotics concepts. These studies highlight the growing interest in making robotic arms more accessible through open-source hardware and software integration.

Nevertheless, most of these works adopted simple mechanical structures or small-degree output. In addition, they usually adopt older software stacks, such as ROS1, which is not real-time controlled, and support multi-robot, scalable embedded systems.

2.2. ROS1 vs ROS2: Architectural Advancements

ROS1 has been popularly used in the past for writing robot applications; however, it adopts master slave architecture and utilizes TCP based Robot Operating System Protocol (TCPROS) to create inter-process communication, which introduces latency between sensors and actuators. ROS1 also does not provide built-in support for security and real-time execution necessary for advanced robotic systems. To address these issues, ROS2 was designed as a federated system using the Data Distribution Service (DDS) middleware.

ROS2 brings significant advancements, including node lifecycle management, improved inter-process communications, real-time safety, multi-robot support, etc. [7,8]. These features make ROS2 more suitable for scenarios with the need for modularity, scale, and integration to real-world systems such as autonomous vehicles and collaborative robots. Consequently, the robotics research community has started to use ROS2 in projects that demand high performance or distribution.

2.3. Simulation and Visualization Tools

The 3D modeling and simulation are essential in the designing and testing of robot systems. Gazebo and RViz are two of the most commonly used tools in the ROS world. Gazebo is a powerful 3D simulation environment with extensive support for physics-based interaction, collision dynamics, and sensor emulation. It is a great tool to experiment with control algorithms under realistic virtual conditions, such as for SLAM, navigation, and manipulation [9]. RViz, by comparison, emphasizes visualization over dynamic simulation. It provides users with an interactive visualization of a robot's model description and additionally can emulate sensor data, transformation matrix in the 3D world, and joint states. It is designed for monitoring information about your robot. Since RViz does not include physical modeling, due to its lightweight and fast processing, it is effective for debugging and validating motion planning in the early stages of development (development phase) [10].

While Gazebo is more popular in academic work because of its high-fidelity simulation, RViz has advantages over it when the focus lies on kinematics as opposed to dynamics (ease of use, performance, real-time visualization).

3. Hybrid cad-ros2 architecture

3.1. System overview

The proposed framework follows a modular, organized pipeline for combining mechanical design, robot description, simulation, and deployment. The proposed software architecture can perform the seamless transition from CAD models to real-time robotic control under ROS2. The end-to-end pipeline is demonstrated in Figure 1, including parametric modeling in SolidWorks, URDF generation, ROS2 integration, simulation with MoveIt and RViz, and hardware deployment.

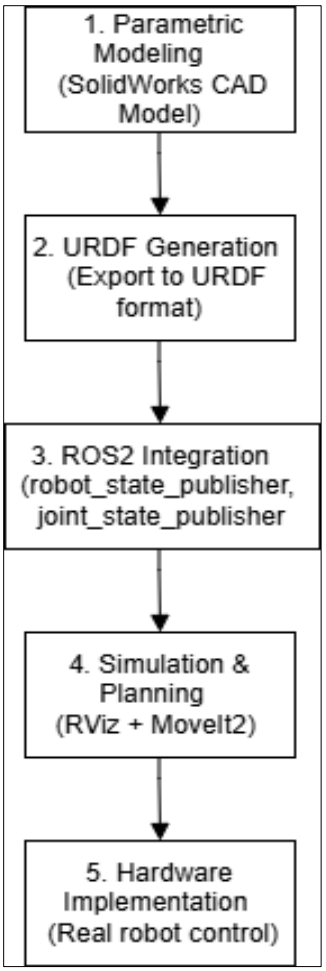

Figure 1 Overview of the Hybrid CAD-ROS2 Architecture Pipeline

The mechanical model of the 6-DOF robotic arm is parameterized into a CAD system. Key elements such as links, joints, and actuator placeholders are part of this model. Once a CAD design is complete, it can be exported as a URDF file using a SolidWorks-to-URDF exporter tool. These can be used to consume the kinematic and geometric information of the robotic arm for ROS2 packages.

The robot model is visualized using RViz, and it is simulated within a ROS2 environment with MoveIt2. Standard ROS2 packages like robot state publisher, joint state publisher, and moveit ros planning interface are used for kinematic calculations, trajectory planning, or joint-space control. The modular ROS2 node-level structure allows for future expansion with real-time feedback, sensor integration, and physical actuation.

The proposed hybrid workflow is a cost-effective, easy-to-scale, and reproducible alternative for robot manipulator design. Using open-source tools, commercial off-the-shelf components, and easily machined parts, the system is flexible in terms of both mechanical redesigns as well as control software, allowing for potential experimentation and educational applications. The introduction in this section sets the stage for writing about CAD modeling, ROS2 integration, and RViz simulation in the upcoming subsections.

3.2. Parametric design and cad modeling

The mechanical structure of the 6-DOF robot arm was modeled parametrically in SolidWorks. With this approach, the arm's geometry can be easily and efficiently altered by placing relationships and constraints between dimensions. Every link and joint was an individual part, which was then assembled in a tree (a branched chain) respecting the kinematic hierarchy and real-life mechanical restrictions, as presented in Figure 2.

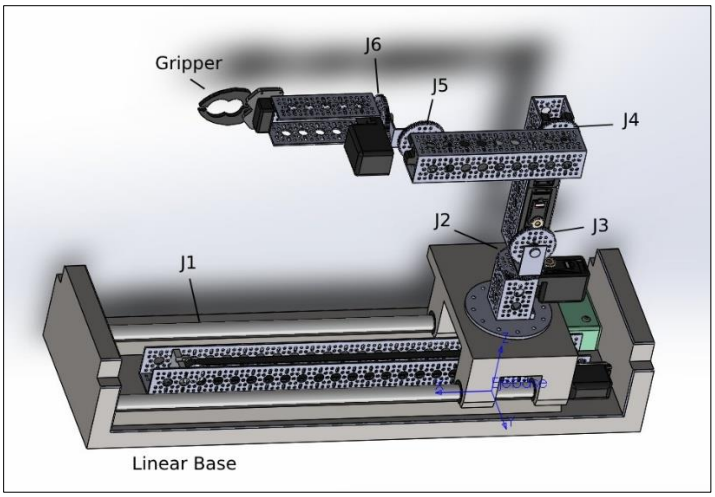

Figure 2 Exploded view of the 6-DOF robotic arm CAD model designed in SolidWorks

Parametric modeling allows rapid alteration of link lengths, joint ranges, and component locations during simulation and hardware prototyping. It has been designed with five revolute (R) and one linear prismatic (L) joint for each degree of freedom, with generalized interfaces for installing a servo motor or actuator at the end of the physical prototype.

For compatibility with ROS2, all components were given reference coordinate frames according to the Denavit–Hartenberg convention [11]. These were aligned with joint axes to ease the export to URDF. The frames of reference and their orientation are indicated in Figure 3.

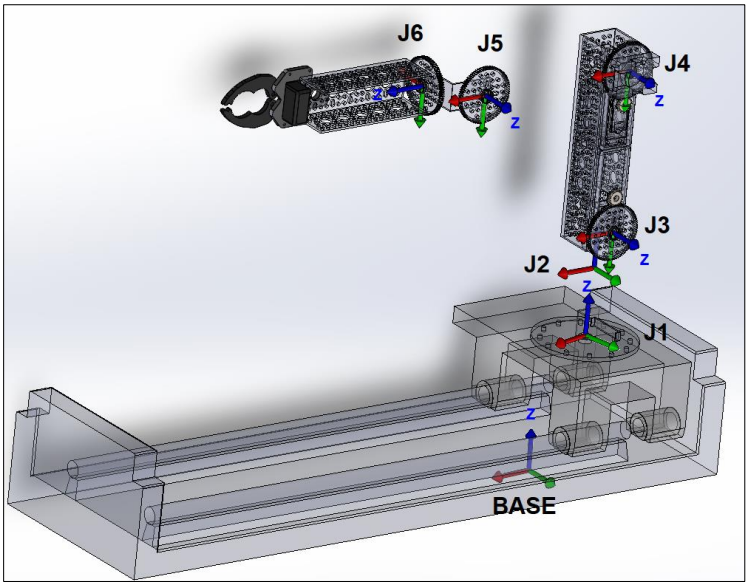

Figure 3 Coordinate frame assignment on each joint in the SolidWorks model

An example application of parametric dimensioning and design is presented in Fig. 4. This figure illustrates a detailed multi-view of the robotic manipulator, including isometric, top, front, and side views, as well as a section view of one of the joints. Both of the above views show how SolidWorks uses parametric constraints, whether they are geometric relationships between parts, dimension-driven features, or alignment constraints between sub-assemblies. Parametric modeling is used to adjust essential design parameters such as link lengths, joint offsets, actuator gaps, and end-effector mounts to adapt the design to a new application or fine-tune physical prototype.

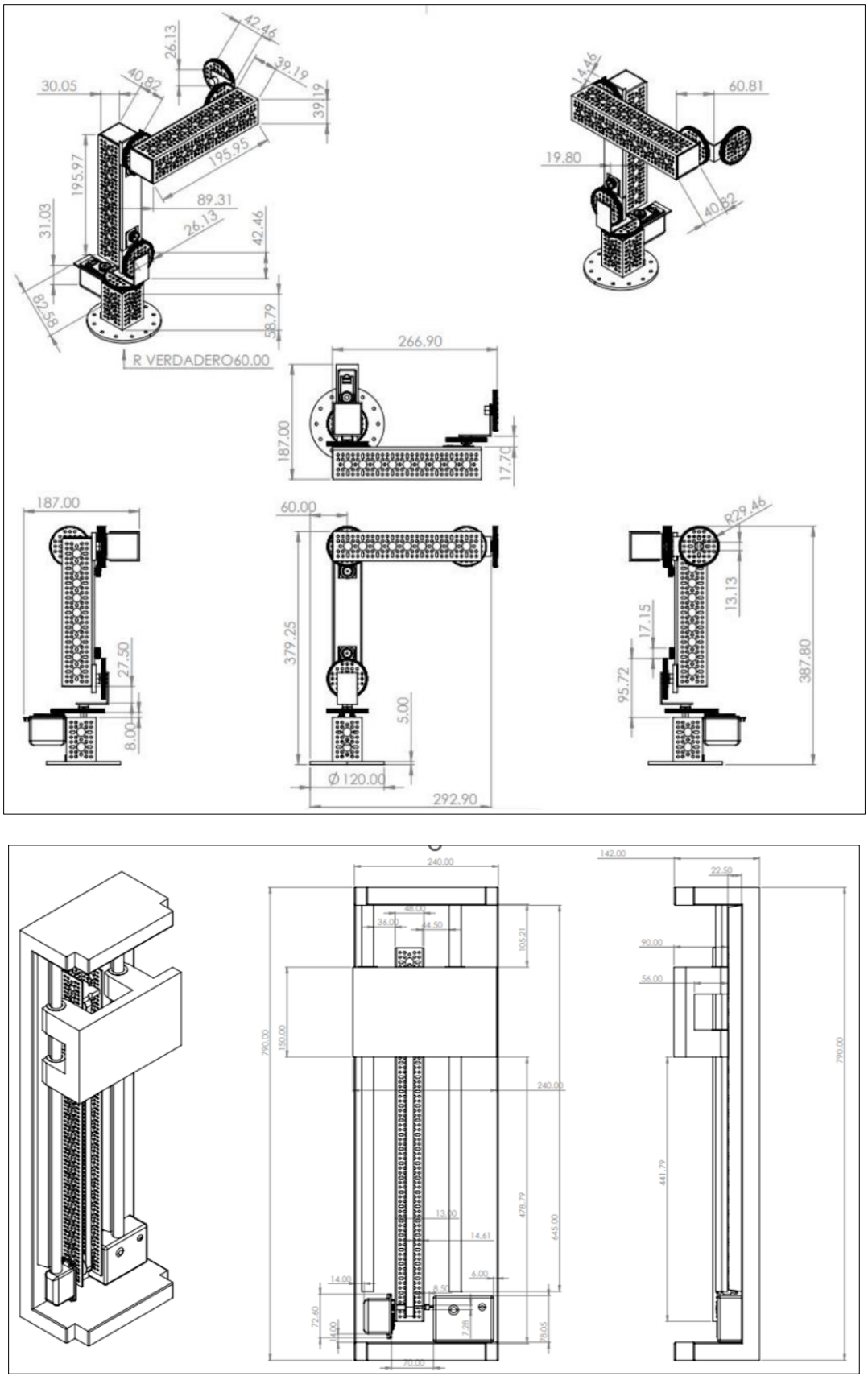

Figure 4 Parametric constraints and geometric relationships defined in the SolidWorks CAD model

The figure also shows how the robot arm can be constructed modularly, with each link and joint independently described, yet all parts constituting the same high-level assembly. Function definitions across modules are standardized to ensure compatibility with different actuators and to enable easy substitution/reconfigurability during the prototype/validation stages. It is worth highlighting the joint-axis orientation and the model followed (which must be according to the Denavit–Hartenberg notation) to guarantee a smooth integration with the URDF description in ROS2.

The resulting CAD model is the robot's digital twin in the ROS2 world. Its modular design enables the integration of sensors or end-effectors in the future, or the addition of any alternative joint mechanism, without redesigning the whole arm. Furthermore, correct modeling of mass and inertia properties was introduced to improve dynamic simulation and planning performance in MoveIt2.

3.3. URDF and robot description conversion

The 3D model is converted to a Unified Robot Description Format (URDF) file for the ROS2 framework, using a CAD-based robotic arm. This was carried out with the help of SolidWorks to URDF exporter, a plug-in that automates exporting native CAD assemblies to URDF. For each component (link) and its kinematic relationship(joint), the exporter generates structured XML representations of them, which contain mass properties, geometry, coordinate transforms, etc.

Each component of the robot, such as base_link, J1, J2, etc., is defined with

- Visual and collision mesh located under the meshes/ folder.
- Inertial parameters including mass and inertia tensors.
- Joints with types (fixed, revolute, prismatic, or continuous), axes of motion, limits, and transformations between parent and child links.

The tag <origin> in the URDF file contains the position and orientation (in xyz and rpy) of each link, which approximates how it's positioned in SolidWorks. Furthermore, a fixed joint is appended to the end to integrate a tool placeholder (tool_link), which serves as the robot's end-effector or Tool Center Point (TCP).

An excerpt from the URDF code defining a revolute joint “J3_J” between two links is shown in Figure 5. This URDF also enables the robot to be visualized in RViz and Gazebo and to perform motion planning and control within ROS2.

4. Implementation and validation

4.1. URDF model verification

```
<joint name="J3_J" type="revolute">
  <parent link="J2" />
  <child link="J3" />
  <origin xyz="0 0 0.14" rpy="0 0 0 " />
```

Figure 5 Example of URDF joint definition used to describe the kinematic relationship between two robot components.

The URDF model obtained from the parametric CAD model was validated to ensure a reliable, consistent representation of the original mechanical design using RVIZ. The length of each link, the types of joints, and the range of motion were verified against the SolidWorks model. Various tests have been conducted to confirm that the virtual robot accurately reflects the physical design (structure, shape, and kinematic capabilities).

The detailed CAD model, as shown in Figure 2, includes all mechanical parts and joint limits. The generated URDF model loaded in RViz (see Figure 6) verifies the correct embedding of geometry and kinematics in ROS2.

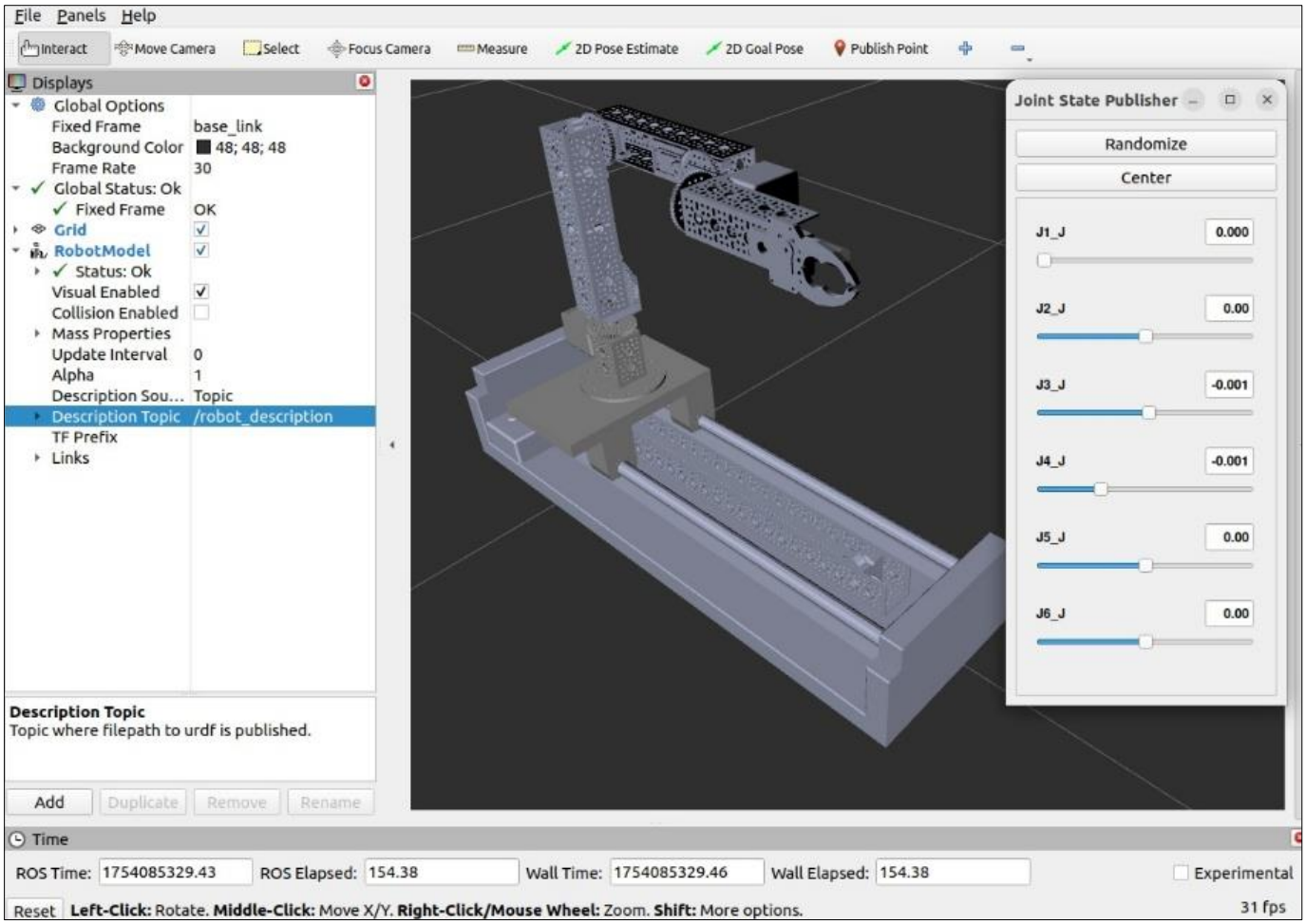

Figure 6 Visualization of the robotic arm URDF model loaded in RViz, illustrating joint axes and link connectivity consistent with the CAD design

4.2. Trajectory planning and validation

To validate the ability to plan a joint-type trajectory, a point-to-point trajectory was executed using the MoveIt framework in ROS2. Two positions were defined in Cartesian space, an initial one (gray) and a final one (orange), as can be seen in Figure 7, demonstrating that the planner generated a feasible joint-space trajectory that satisfied the kinematic constraints of the system. Through RIVZ's "Show Trace" function, it was possible to simulate movement more accurately by displaying the complete visual pattern, showing how the end effector moves from its starting point to its ending point, as shown in Figure 7. The trace indicates a continuous path from the beginning to the target position, which confirms the correct planning and execution by the motion planner.

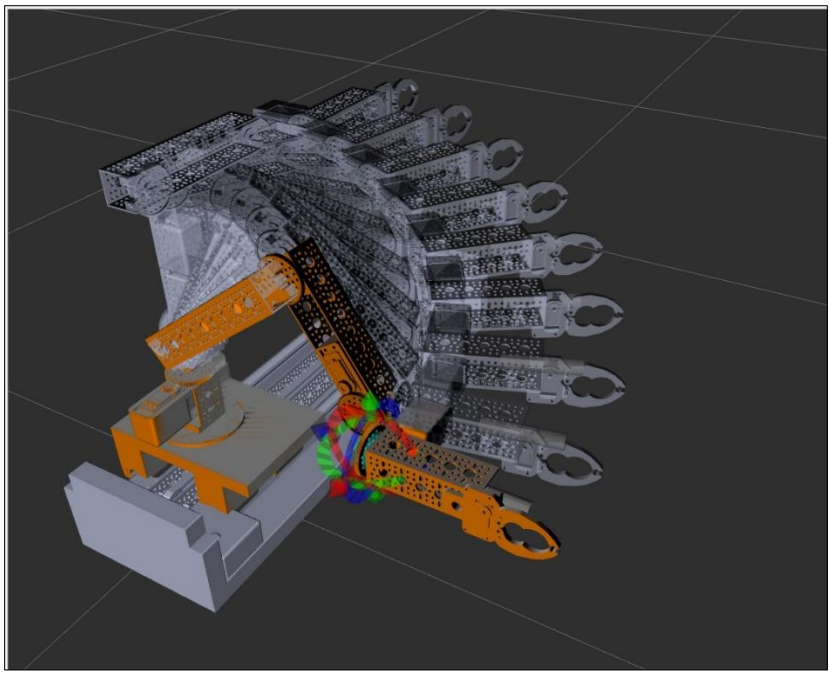

Figure 7 End-effector trajectory visualized using the "Show Trail" feature in RViz

4.3. Results

Throughout the development of this project, functional integration between ROS2 and physical implementation was achieved using an embedded system based on a Raspberry Pi Pico. Using MoveIt2 as the planning engine, it was possible to calculate collision-free trajectories. Figure 8 shows the interface built in conjunction with the Raspberry Pi for motor control.

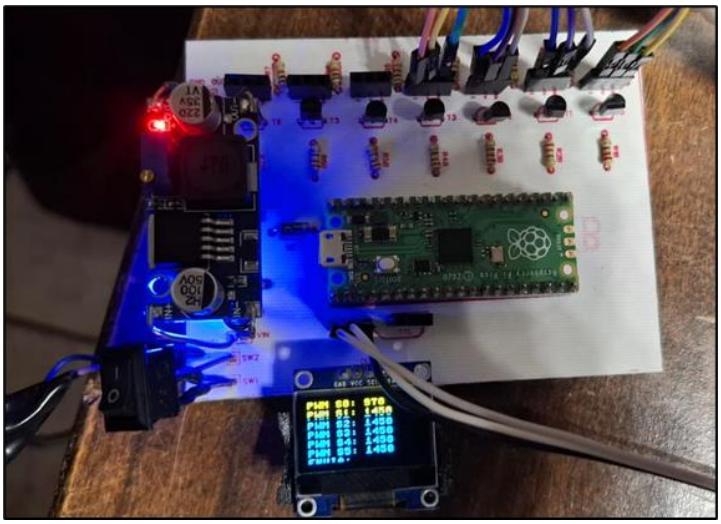

Figure 8 Robot Control Interface

In the embedded section, a control system was implemented capable of receiving instructions via UART, decoding 8-bit frames, and updating the PWM values of each servomotor in real time. This architecture enabled the physical reproduction of the planned movements, and preliminary tests showed a stable, synchronized response between the simulated environment and the mechanical action. Data visualization via an OLED screen on the PCB facilitated monitoring of the system's status during operation.

To export the robot model created in SOLIDWORKS, the SW2URDF plugin was used. This model was correctly loaded into RViz and configured for use in MoveIt2, where trajectories were successfully simulated and communication with the assembled real prototype was achieved via UART.

One of the tests involved implementing a seven-point routine. The first point was called "home." The remaining points simulate the robot's movement as if it were moving its mechanical gripper toward an object located to the side of its base, then lifting its gripper, moving to the end of the base's path, and then moving its gripper back down toward the front of the base, before returning to its starting position. The complete routine is shown in Figure 9.

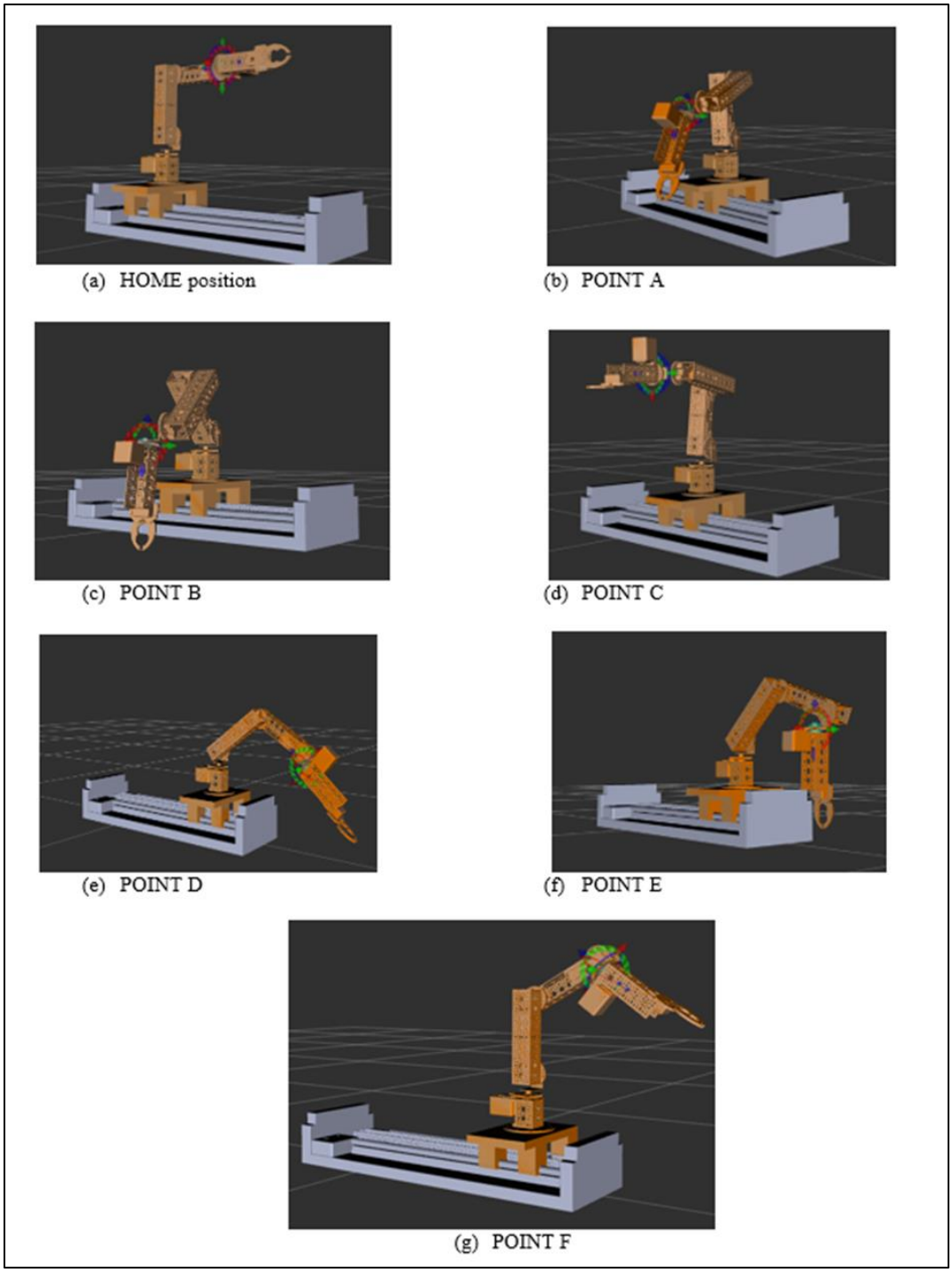

Figure 9 Simple Pick and Place Simulation

Figure 10 shows the robot's actual design during motion testing, along with the control board.

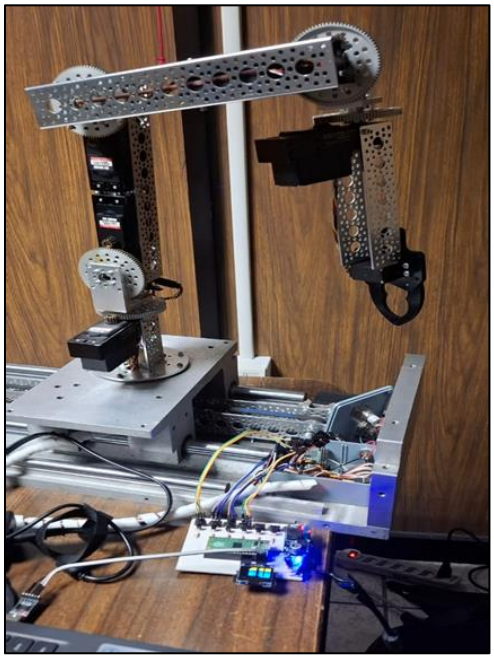

Figure 10 Actual view of the implemented robot

The correct movement of the real prototype was observed alongside its simulated digital twin in RVIZ. The robot correctly responded to data sent via UART to the Raspberry Pi Pico. A comparative table was created using this data, adjusting the simulation arrival times to match the actual movement time. The results, comparing the real robot to its digital twin, are shown in Table 1.

Table 1 Comparative tables of motion in simulation and real time

Real Robot				
Servomotor	**Movement Time(secs)**			**Average (°/secs)**
	displacement (0°-90°)	**displacement (90°-180°)**	**displacement (0°-180°)**	
J1	3.9	4.63	11.31	4.72833333
J2	4.52	4.86	10.9	4.94333333
J3	6.5	7.12	14.62	6.97666667
J4	3.25	3.25	7	3.33333333
J5	4.85	6.8	12.2	5.91666667
J6	1.5	2.1	3	1.7
Simulated Robot				
Servomotor	**Movement Time (secs)**			**Average (°/secs)**
	Displacement (0°-90°)	**Displacement (90°-180°)**	**Displacement (0°-180°)**	
J1	3.9	4.63	11.31	4.72833333
J2	4.52	4.86	10.9	4.94333333
J3	6.5	7.12	14.62	6.97666667
J4	3.25	3.25	7	3.33333333
J5	4.85	6.8	12.2	5.91666667
J6	1.5	2.1	3	1.7

5. Conclusion and future work

This work presented the design, simulation, and implementation of a 6-DOF robotic arm using a hybrid CAD-based architecture integrated with ROS2 and MoveIt. The parametric model developed in SolidWorks was successfully exported to URDF and validated within the ROS2 framework. The integration with MoveIt enabled accurate trajectory planning and execution, while RViz provided practical visualization tools such as Show Trail for analyzing point-to-point motions.

The results demonstrate that the proposed hybrid architecture provides a reliable workflow to transition from mechanical design to robotic simulation, ensuring consistency between the CAD model and its virtual representation. The successful execution of point-to-point trajectories validated the kinematic accuracy of the URDF model and confirmed the feasibility of the system for educational and research applications.

In a future version, the intention is to change the communication between ROS2 and the embedded system from UART to micro-ROS, enabling full control through ROS2.

Compliance with ethical standards

Acknowledgments

This work is supported by the Tecnológico Nacional de México through the research project 23268.25-P.

Disclosure of conflict of interest

The Authors confirm that the content of this manuscript has no conflict of interest.

References

[1] T. Singh, P. Anand and S. K. R. Development and Simulation of a Robotic Arm Using ROS2: A Comprehensive Study. 2025 International Conference on Pervasive Computational Technologies (ICPCT). 2025; 248-253.

[2] A. Sarajlić and L. Banjanović-Mehmedović. AI-Driven Human-Robot Interaction for a 3D-Printed Robotic Arm Using Gemini AI and ROS. 2025 24th International Symposium INFOTEH-JAHORINA (INFOTEH). 2025; 1-6.

[3] P. A. Wakem and T. E. Mamonova. Control of a Custom Urdf Robotic Arm in ROS. 2024 IEEE 3rd International Conference on Problems of Informatics, Electronics and Radio Engineering (PIERE). 2024; 600-603.

[4] D. Krūmiņš et al. Open Remote Web Lab for Learning Robotics and ROS With Physical and Simulated Robots in an Authentic Developer Environment. IEEE Transactions on Learning Technologies. 2024; 17:1313-1326.

[5] S. Trubayev, E. Shehab and M. H. Ali. Design and Development of a Small-Scale 5-DOF 3D Printed Robot Manipulator. 2022 IEEE 18th International Colloquium on Signal Processing and Applications (CSPA). 2022; 260-265.

[6] Zajc, L. Č., Rezelj, A., and Skočaj, D. Low-Cost Open-Source Robotic Platform for Education. IEEE Transactions on Learning Technologies. 2022; 16(1):18-25.

[7] Macenski, S., Foote, T., Gerkey, B., Lalancette, C., and Woodall, W. Robot operating system 2: design, architecture, and uses in the wild. Science Robotics. 2022; 7(66): eabm6074.

[8] Koubaa, A. Robot operating system (ROS): the complete reference. (Volume 5). 1st ed. Cham: Springer International Publishing; 2021.

[9] Gazebo Documentation [Internet]. Mountain View, California: Open Robotics; ©2025 [cited 2025 Sep 2]. Available from: https://gazebosim.org

[10] Rviz Documentation [Internet]. Mountain View, California: Open Robotics; ©2025 [cited 2025 Sep 2]. Available from: https://wiki.ros.org/rviz

[11] Tüske, I., and Hegedűs, G. Direct kinematics of a tilting table using Denavit: Hartenberg convention. Multidiszciplináris Tudományok. 2023; 13(4), 163-171.

[12] Ekrem, Ö., and Aksoy, B. Trajectory planning for a 6-axis robotic arm with particle swarm optimization algorithm. Eng. Appl. Artif. Intell. 2023;122,106099.

[13] Megalingam, R., Katta, N., Geesala, R., Yadav, P., and Rangaiah, R. Keyboard-Based Control and Simulation of 6-DOF Robotic Arm Using ROS. 2018 4th International Conference on Computing Communication and Automation (ICCCA). 2018; 1-5.

[14] Li, C., Hao, L., Cheng, H., and Nie, X. Research on Motion Planning System of Service Robot Based on ROS. 2017 IEEE 7th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER). 2017; 205-209.

[15] Li, C., Hao, L., Cheng, H., and Nie, X. Research on Motion Planning System of Service Robot Based on ROS. 2017 IEEE 7th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER). 2017; 205-209.