

A MATLAB/Simulink-Based Sim2Real Control Framework for the Unitree G1 Using ROS 2 and MuJoCo

Leffer Trochez¹, Nicanor Quijano¹, Jorge Lopez-Jimenez¹, and Carlos F. Rodriguez²

¹Department of Electrical and Electronic Engineering, Universidad de los Andes, Bogotá D.C., Colombia

²Department of Mechanical Engineering, Universidad de los Andes, Bogotá D.C., Colombia

Abstract—Humanoid robot control research requires development environments in which control strategies can be designed, organized, debugged, validated, and transferred to real hardware with minimal friction. In practice, however, this workflow is often fragmented across separate tools for dynamic modeling, simulation, middleware communication, visualization, and deployment. Such fragmentation increases implementation effort, slows down controller iteration, complicates closed-loop debugging, and makes it difficult to preserve a consistent architecture when transitioning from simulation to the real robot. This limitation is especially relevant for commercial humanoid platforms such as the Unitree G1, where practical research often requires combining multiple software layers and execution environments.

This work presents a MATLAB/Simulink-based Sim2Real control framework for the Unitree G1 that integrates MuJoCo and ROS 2 into a unified and control-oriented workflow. The framework is organized around a modular Variant Subsystem that switches between a MuJoCo simulation backend and a ROS 2 real-robot backend while preserving compatible interfaces across both domains. This enables reuse of the same high-level control organization for simulation-based validation and hardware execution without redesigning the overall implementation structure. As a representative example, a standard ankle-motion command task was validated in simulation and executed on the real Unitree G1 through the same workflow. The resulting framework can be understood as a research-oriented integration layer that supports rapid prototyping, structured debugging, modular controller development, and future extensions toward estimation, stabilization, perception, and higher-level humanoid autonomy.

I. MOTIVATION AND OBJECTIVE

Humanoid robots are complex systems whose control development rarely takes place in a single environment. In many practical cases, controller logic is designed in one tool, tested in another, connected to middleware through separate scripts, and then adapted again for real-robot deployment. Although this fragmented process is common in robotics research, it introduces substantial friction. The most immediate consequences are duplicated implementation effort, poor traceability between simulation and hardware versions, and additional difficulty when debugging closed-loop behavior across the full system.

This issue becomes particularly relevant for the Unitree G1, a platform with strong research potential but for which practical controller development still benefits from a clean and

organized integration workflow. From a control-engineering perspective, researchers need more than a simulator or a communication interface in isolation. They need an environment in which controllers, signals, interfaces, and execution backends can be structured in a modular and reusable way. The main objective of this work is therefore to provide a unified Sim2Real framework that reduces fragmentation and offers a clearer path from controller design to simulation-based validation and real-robot execution. Rather than proposing a new low-level control law, this work focuses on the integration layer needed to accelerate experimentation. In that sense, the contribution is architectural and methodological: the framework provides a reusable MATLAB/Simulink-based environment for implementing, testing, monitoring, and extending humanoid control strategies on the Unitree G1.

II. FRAMEWORK PURPOSE AND ARCHITECTURE

The purpose of the framework is to unify the principal stages of a control-oriented humanoid robotics workflow inside a single development environment. More specifically, it is intended to support: i) modular controller implementation in Simulink, ii) simulation-based validation using MuJoCo, iii) communication with the real Unitree G1 through ROS 2, iv) structured monitoring and debugging of relevant signals, and v) reuse of the same high-level organization across simulation and real-robot execution. These capabilities are especially valuable in robotics control because they reduce the amount of rework usually required when a controller moves from concept to experiment.

The central architectural element is a Sim2Real Variant Subsystem implemented in MATLAB/Simulink. This subsystem switches between two execution backends: a MuJoCo-based simulation backend and a ROS 2-based real-robot backend. A key design decision is that both backends preserve compatible input/output interfaces, allowing the same high-level control logic, signal flow, and monitoring structure to be reused when moving from simulation to hardware. This reduces the need to reorganize or rewrite the controller when changing the execution layer. As summarized in Fig. 1, modeling in Simulink, validation in MuJoCo, and deployment through



Fig. 1. Unified Sim2Real development pipeline implemented in the proposed framework.

ROS 2 are treated as parts of a unified workflow rather than as disconnected stages.

The repository organization follows the same principle by including the Unitree G1 MuJoCo model, ROS 2 communication files, the custom `unitree_hg` message workflow, configuration scripts, signal buses, and constraints that support a modular and extensible implementation.

III. CONTRIBUTIONS AND PRACTICAL USE

The main contributions of this work can be summarized in three points. First, it provides a practical MATLAB/Simulink-based Sim2Real integration framework for the Unitree G1 that connects simulation, communication, visualization, and deployment within a unified workflow. This is relevant because it gives control researchers a structured environment for experimentation without forcing them to separate design, testing, and execution into disconnected stages.

Second, the framework introduces a backend-switching architecture based on a Variant Subsystem that preserves interface compatibility between MuJoCo and ROS 2 execution. This directly addresses one of the most common inefficiencies in robotics development: reorganizing or rewriting high-level logic when transitioning from simulation to hardware. By maintaining a common interface structure, the framework reduces implementation overhead and improves continuity between development phases.

Third, the project is organized as a reusable research base rather than as a single isolated example. The inclusion of configuration files, signal buses, constraints, simulation resources, and communication components makes it suitable as a starting point for future developments in control and estimation. In practical terms, the framework is useful for rapid prototyping, controller validation, structured debugging, and extension toward more advanced humanoid functionalities.

IV. REPRESENTATIVE SIM2REAL DEMONSTRATION

To illustrate the proposed workflow, a standard ankle-motion command task was selected as a representative example. The purpose of this case study is not to maximize task complexity, but to validate the central claim of the framework under a clear and interpretable motion command. By choosing a focused demonstration, the contribution remains centered on workflow consistency rather than on the difficulty of a specific benchmark. The demonstration shows that the controller organization can be built and tested in Simulink, validated under the MuJoCo backend, and then executed on the real robot through the ROS 2 backend while preserving the same general structure. This is precisely the transition that the framework is designed to simplify. From a control research

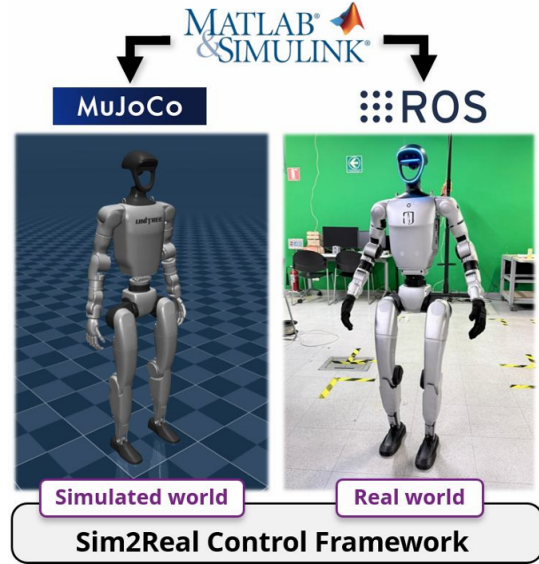


Fig. 2. Representative Sim2Real view of the proposed framework in simulation and on the real Unitree G1.

perspective, the practical value lies in maintaining the same model organization, signal flow, and monitoring logic while changing only the execution backend. Figure 2 illustrates this principle by showing the same general framework concept connecting the simulated world and the real robot through a common Simulink-centered development structure.

Although the example is intentionally simple, it is methodologically important. Many robotics workflows appear functional in isolation but become difficult to scale when additional sensing, estimation, or supervisory modules are introduced. In contrast, the proposed framework emphasizes structural consistency from the beginning, which makes it more suitable as a basis for continued development rather than as a one-time demonstration.

V. CONCLUSION

This work presented a MATLAB/Simulink-based Sim2Real control framework for the Unitree G1 humanoid robot. By organizing MuJoCo simulation and ROS 2 real-robot communication around a common Variant Subsystem with compatible interfaces, the framework enables a unified workflow for controller development, simulation-based validation, debugging, visualization, and deployment. The representative ankle-motion example demonstrated the practical simulation-to-hardware transition supported by the proposed environment. Overall, the framework provides a clean and extensible basis for future humanoid control research on the Unitree G1.