Building a ROS 2 - Isaac Sim Framework for Dual Arm Manipulation of Rigid Objects and Textiles

Jonas Gschnell
Institute of Robotics
Johannes Kepler University Linz
4040 Linz
jonas.gschnell@jku.at

Alexander Kitzinger
Institute of Robotics
Johannes Kepler University Linz
4040 Linz
alexander.kitzinger@jku.at

Hubert Gattringer
Institute of Robotics
Johannes Kepler University Linz
4040 Linz
hubert.gattringer@jku.at

Andreas Müller
Institute of Robotics
Johannes Kepler University Linz
4040 Linz
a.mueller@jku.at

Abstract

This paper presents a framework that connects ROS 2 with NVIDIA Isaac Sim to support perception-driven dual-arm manipulation, evaluated on rigid objects and textiles. While rigid object handling is achieved after careful parameter tuning, textile manipulation exposes limitations. The paper discusses key integration challenges such as interface alignment, temporal and spatial synchronization, and coordinated dual-arm motion planning.

1 Introduction

Configuring robotic systems to perform complex manipulation tasks reliably is challenging, particularly under cost and time constraints. The interaction of sensors, robots and objects to be manipulated is often difficult to predict, and extensive experimentation is often needed [1]. At the same time powerful consumer-level GPUs enable high-fidelity simulations that promise rapid prototyping of robotic workflows. Photorealistic rendering and physically based simulation suggest that complex perception based manipulation setups can be validated before deployment in real systems. However, in reality integration is still often complex and integration-details are often omitted in favor of high-level insights and results. Practical system constraints—such as controller interface alignment, temporal synchronization, and consistent transformation management—are frequently underreported in favor of high-level task performance. This work investigates these integration challenges through the development of a pipeline using NVIDIA's Isaac Sim as the simulation environment in combination with Robot Operating System 2 (ROS 2). As a representative scenario, dual-arm manipulation of a rigid body and a deformable body, specifically cloth manipulation, is used. Cloth manipulation is introduced as a test case that exposes simulation limitations, while rigid object handling serves to validate the framework. By analyzing recurring integration issues and system-level constraints, the paper aims to provide practical design guidelines that improve reproducibility and Sim-to-Real workflows for perception-driven robotic manipulation.

2 System architecture and application examples

To investigate the integration effort for linking ROS 2 and Isaac Sim, a dual-arm manipulation cell is implemented. ROS 2 provides a large ecosystem of reusable packages covering perception, motion

The Third Austrian Symposium on AI, Robotics, and Vision (AIROV26).

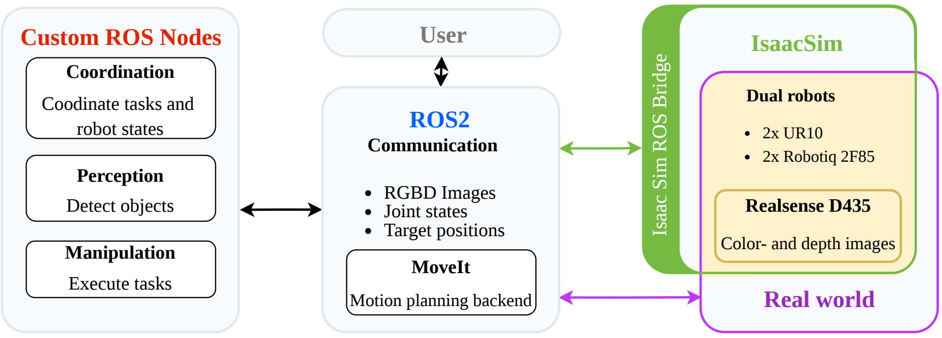

Figure 1: System architecture for simulated and real world setup.

planning, control, and many more [2]. Within the ROS framework Universal Robot Description Format (URDF) files are used to describe robot kinematics, geometry, inertial properties, and sensors. Further, they define the structural and control interfaces expected by downstream tooling. Reusable URDF configurations are parameterized using Xacro, where macros are used to set up a complete robot cell. To assemble the same cell in NVIDIA Isaac Sim Universal Scene Description (USD) assets are composed to a complete scene using Python. The ROS package MoveIt2 is used as a motion planning backend, it handles inverse kinematics, collision checking, trajectory generation, and execution management [3]. Semantic robot information such as planning groups, end effectors, and allowed collisions is specified via Semantic Robot Description Format (SRDF) files. The simulated workcell consists of two Universal Robots UR10 arms equipped with Robotiq 2F-85 grippers. They are placed on a table and a spatial fixed RGB-D camera (Intel RealSense D435) is used for visual observation. An overview of the complete hardware and software setup is shown in fig. 1.

Dual-arm manipulation of rigid objects: To validate the framework the virtual setup is tested by manipulating a simple rigid object with the dual arms. The procedure includes a cooperative movement with both grippers holding the object. Reliably grasping the object in Isaac Sim is challenging, as small changes to the grippers stiffness, damping, contact offsets, or friction coefficients can switch a grasp from stable to completely failing. Simulating realistic contact interactions is difficult due to discretization limits, geometry approximations, friction modeling and contact solver stability. Discrete collision checks with practical time steps sometimes lead to tunneling and interpenetration for fast or small-contact grasps, unless the number substeps and solver iterations is increased significantly. Further, the use of convex geometry approximations leads to higher performance, but misaligned contact patches, especially for thin fingers and complex objects simulation stability suffers. For rigid objects these problems can be reduced by using small time steps and decreasing the grippers stiffness.

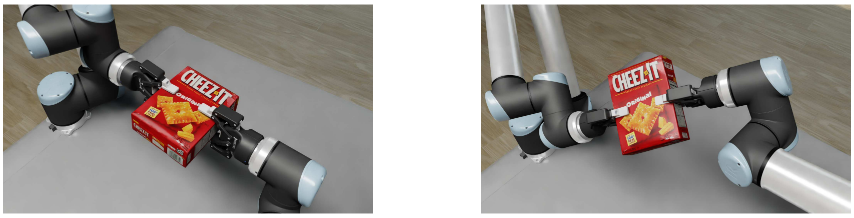

Figure 2: Cooperative dual-arm manipulation. (left) Initial configuration. (right) Final configuration.

Textile manipulation: Isaac Sim also enables more advanced physics simulations, including particle-based models that can represent deformable objects such as cloth. Since textile manipulation is becoming increasingly important, the simulation of cloth manipulation gets explored as a challenging test case. To set up the simulation, a mesh of a standard T-shirt is imported into Isaac Sim. Once imported in Isaac Sim a particle-based model is applied, transforming mesh vertices into particles connected by virtual springs and dampers. Additional material parameters such as mass, density, friction, adhesion, drag, and lift can be specified to control the behavior of the simulated cloth. Despite extensive parameter tuning, from simulation settings such as solver iterations and simulation step time to cloth parameters such as mass, friction, stiffness or contact offsets, stable

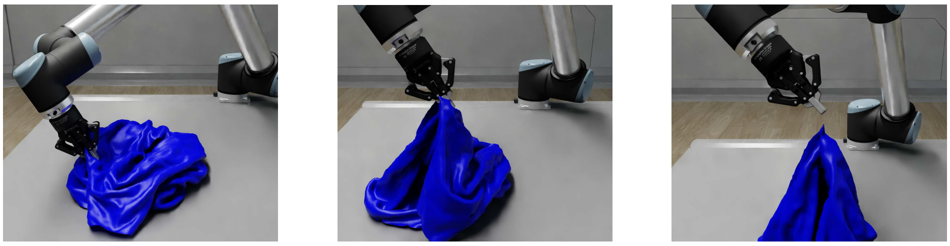

Figure 3: Attemt to lift cloth from table. (center) Visible tunneling of gripper through fabric.

grasping of the fabric could not be achieved. As illustrated in fig. 3, the cloth repeatedly slips out or tunnels through the grippers. Because the cloth mesh represents a single layer, the contact issues discussed above become particularly pronounced, making reliable grasping difficult to achieve under the tested conditions.

3 Integration Issues

In this section common integration challanges are discussed, and possible solutions are presented. The topics covered include interface mismatches, as well as time and spatial synchronization issues between Isaac Sim and ROS. Finally, the particular implementation for cooperative dual arm movement is discussed.

Interface alignment: Communication between ROS 2 and the simulation environment is realized via the Isaac Sim ROS Bridge, which exposes simulation data and takes in actuator commands through ROS Topics. A robot setup using ROS controllers needs specific command interfaces and state interfaces. The controllers then use this abstracted interfaces to send commands to the robot and receive state updates, regardless of whether the robot is simulated or real. The ROS Bridge does not provide such interfaces, it only enables the simulation articulation controllers to both publish states to and receive commands from ROS Topics. As a consequence, an additional adaptation layer is required to align controller expectations between the two ecosystems. The specific hardware interfaces that are initialized on launch must be defined in the URDF file. In particular, the *topic_based_control*-plugin can be used in the URDF files. It makes hardware interfaces available, translates commands to standard ROS Topics and returns state feedback to the hardware interface.

Temporal consistency: Physics steps and rendering in Isaac Sim vary in duration depending on the current scene complexity. Therefore, Isaac Sim does not necessarily simulate in real time. However, control systems, trajectory execution and perception depend on a consistent timeline. As a consequence, it is necessary to synchronize ROS Nodes with Isaac Sim's simulation time. The simulation time can be published on a ROS Topic via the Isaac Sim ROS Bridge. Per default, Nodes take the system time to synchronize, but by changing Node parameters an external time source can be used as reference.

Spatial consistency The photorealistic camera streams generated by Isaac Sim can be used for generating synthetic data for deep neural networks or reinforcement learning algorithms, as well as validating perception-based control pipelines. Usually in such pipelines object poses detected in camera coordinates must be transformed into a common reference frame. ROS 2 provides such capabilities with the build in transformation framework TF2, which maintains a dynamically updated tree of coordinate transforms between all links. Robot joint states are already published from Isaac Sim to ROS as part of the control feedback loop. These joint states are used by TF2 to update the robot kinematic chain. In contrast, a fixed camera mounted in the environment can be defined by a static transform relative to the world frame. Such static transforms are derived from the URDF file if not defined otherwise. A common source of integration errors arises from differences in coordinate frame conventions between Isaac Sim and ROS. As a result, additional transformations may be required when publishing camera poses through the ROS bridge. Careful verification of coordinate frame conventions is therefore necessary to ensure consistent perception results.

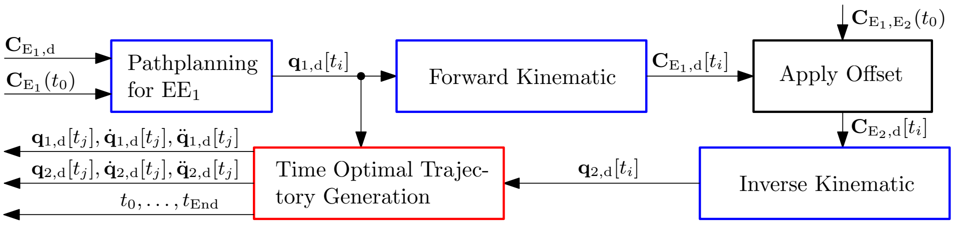

Figure 4: Workflow to generate a time optimal trajectory for cooperative dual-arm movement.

Dual arm movement: For cooperative manipulation tasks, the relative position and orientation between the end effectors must remain constant for the whole trajectory. MoveIt exposes several ROS Services and Actions to plan and execute movements, however, it does not directly expose a planning interface that enforces a fixed relative transform between multiple end effectors. The approach proposed here, allows to remain closely integrated with the MoveIt framework. Consequently, collision checking, trajectory validation, and time parameterization remain within MoveIt. This preserves MoveIt's safety mechanisms and maintains a single source of truth for the robot configuration. Fig. 4 illustrates the workflow used to generate synchronized joint trajectories for both end effectors E_1 and E_2. Given the initial configuration $\mathbf{C}_{\mathrm{E}_1}(t_0)$ and the desired configuration $\mathbf{C}_{\mathrm{E}_1,\mathrm{d}}$ of end effector E_1 the desired trajectory is first planned and converted to joint space resulting in $\mathbf{q}_{1,d}[t_i]$. Using forward kinematics and the relative configuration of E_2 w.r.t. E_1, $\mathbf{C}_{\mathrm{E}_1,\mathrm{E}_2}(t_0)$, the corresponding configurations $\mathbf{C}_{\mathrm{E}_2}[t_i]$ are computed and converted to joint paths. Finally, a time-optimal trajectory generator produces synchronized position $\mathbf{q}_{\mathrm{d}}[t_j]$, velocity $\dot{\mathbf{q}}_{\mathrm{d}}[t_j]$, and acceleration $\ddot{\mathbf{q}}_{\mathrm{d}}[t_j]$ profiles for both robots. The blue steps in the figure are accessible via Actions or Services, the red marked time optimal trajectory generation [4] is available as a c++ library. However, by writing a simple ROS Node it can be wrapped in a ROS Service.

4 Conclusion

Although Isaac Sim allows rapid advances in perception driven - and RL workflows, setting up a robust framework is not a rudimentary task. Nevertheless, with some know-how and experience it is possible to utilize the high fidelity simulation to reduce integration time. The dual arm manipulation of rigid objects was successful after carefully tuning the parameters. However, when investigating the manipulation of textiles, the limitations of current cloth simulation became apparent, particularly with regard to reliable grasping. Future work will focus on experimental validation on a real robot system and on improving textile manipulation strategies, with the long-term goal of enabling tasks such as pursued by the ICRA Cloth Competition [5].

Acknowledgments and Disclosure of Funding

This work has been supported by the "LCM – K2 Center for Symbiotic Mechatronics" within the framework of the Austrian COMET-K2 program.

References

[1] Ester Martinez-Martin and Angel P Del Pobil. Vision for robust robot manipulation. *Sensors*, 19 (7):1648, 2019.

[2] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Ng. Ros: an open-source robot operating system. volume 3, 01 2009.

[3] Sachin Chitta, Ioan Sucan, and Steve Cousins. Moveit![ros topics]. *IEEE Robotics & Automation Magazine*, 19(1):18–19, 2012.

[4] Tobias Kunz and Mike Stilman. Time-optimal trajectory generation for path following with bounded acceleration and velocity. *Robotics: Science and Systems VIII*, pages 1–8, 2012.

[5] Victor-Louis De Gusseme et al. A dataset and benchmark for robotic cloth unfolding grasp selection: The icra 2024 cloth competition. *International Journal of Robotics Research*, 2026.