2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Bridging the Sim-to-Real Gap with multipanda_ros2: A Real-Time ROS 2 Framework for Multimanual Systems

Jon Škerlj[1†], Seongjin Bien[2], Abdeldjallil Naceri[1] and Sami Haddadin[3]

***Abstract*— We present `multipanda_ros2`, a novel open-source ROS 2 architecture for multi-robot control of Franka Robotics robots. Leveraging `ros2_control`, this framework provides native ROS 2 interfaces for controlling any number of robots from a single process. Our core contributions address key challenges in real-time torque control, including interaction control and robot-environment modeling. A central focus of this work is sustaining a 1kHz control frequency, a necessity for real-time control and a minimum frequency required by safety standards. Moreover, we introduce a controllet-feature design pattern that enables controller-switching delays of $\leq$ 2 ms, facilitating reproducible benchmarking and complex multi-robot interaction scenarios. To bridge the simulation-to-reality (sim2real) gap, we integrate a high-fidelity MuJoCo simulation with quantitative metrics for both kinematic accuracy and dynamic consistency (torques, forces, and control errors). Furthermore, we demonstrate that real-world inertial parameter identification can significantly improve force and torque accuracy, providing a methodology for iterative physics refinement. Our work extends approaches from soft robotics to rigid dual-arm, contact-rich tasks, showcasing a promising method to reduce the sim2real gap and providing a robust, reproducible platform for advanced robotics research.**

I. Introduction

For robots to be able to operate in the challenging, dynamic and cluttered environments they are used in today, there are high requirements towards their control, among them: Torque control, Real-time capability, and the ability to quickly change between controllers to adapt to new tasks, constraints and environments. The foundation for robots to sense was laid with force and torque control [1], which represents a critical facet of robotics and automation, serving as a key enabler for robots to interact with their environments in a refined and intelligent manner. The capability to precisely control force and torque is indispensable for performing delicate tasks, ranging from assembly operations in industrial settings [2] to surgical procedures [3].

The early important work on giving robots a sense of touch and variable stiffness was performed by Salisbury [4] in 1980, which allowed the control of the stiffness of and all 6 degrees of freedom with a control rate of 60 Hz. In further research [5], this approach was extended to allow for a separation of directions where either force or position could be controlled. In these experiments, an improved force response could be observed by increasing the servo rate from 60 Hz to 120 Hz, but it was still unable to control the higher oscillations at about 280 Hz, due to computational limitations. With the development of more lightweight and flexible robots, the frequency of these oscillations further increased [6] leading to a necessity of control frequency of up to 1 kHz [7], [8].

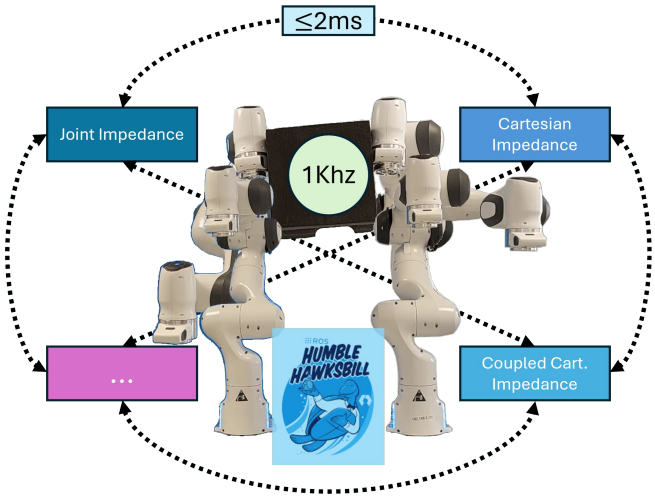


Fig. 1: An overview of `multipanda_ros2`, summarizing its core features.

Controlling the robots for safe and effective interaction now goes beyond traditional positional and force-based control, e.g. impedance control [9] in scenarios where the robot requires compliance, or the converse, admittance control [10], mapping force inputs to motion outputs. Even more complex tasks like wiping a surface require unified frameworks [11] for successful task execution. Other scenarios like hand-over or the manipulation of unwieldy objects involve two or more robots in synchronous, cooperative, or coupled movement [12], [13]. Naturally, scenarios in which a combination of such skills is required would inevitably call for the robot to be capable of rapidly and seamlessly switching from one skill to another.

Having discussed the different control types, the importance of maintaining a high control sample rate, and the necessity for ensuring flexible controller interchangeability, outlined in Fig. 1, the novel contributions of our work can be stated as follows:

- Review real-time torque control for single- and dual-arm robots, emphasizing interaction control and robot–environment modeling challenges.
- Introduce an open-source ROS 2 multirobot framework for Franka Robotics robots with $\leq$ 2 ms controllet-switching delays for reproducible benchmarking.
- Integrate MuJoCo simulations with metrics for kinematic accuracy and dynamic consistency (torques,

This work was supported by the Lighthouse Initiative Geriatronics by StMWi Bayern (Project X, grant no. IUK-1807-0007// IUK582/001) and LongLeif GaPa gGmbH (Project Y).
[1]Munich Institute of Robotics and Machine Intelligence (MIRMI), Technische Universität München (TUM), Germany.
[2] Technische Universität Nürnberg (UTN), Germany
[3] Mohamed Bin Zayed University of Artificial Intelligence, UAE
[†]Corresponding author. Email: `jon.skerlj@tum.de`

forces, control errors).
- Assess force-based fidelity, extending soft-robotics approaches [14] to rigid dual-arm, contact-rich tasks.
- Real-world inertial parameter identification reduces the sim-to-real gap, improving force/torque accuracy and enabling iterative physics refinement.

II. State of the Art

A. High Frequency Torque Control

High-frequency torque control is fundamental for safe and accurate physical human–robot interaction. A 1 kHz low-level torque command loop enables robots to be modeled according to Lagrangian dynamics, such that higher-level controllers can treat the joints as ideal torque sources [8]. At the same time, safety standards such as ISO 10218 [35] specify 1 kHz measurement frequencies for collision detection to minimize injury risk, which has driven widespread adoption of this frequency in collaborative robots (cobots). Prominent examples include the DLR Lightweight Robot series [11], [15], Franka Emika Panda [8], KINOVA Gen 3 [31], and Barrett WAM [20].

These systems often employ cascaded architectures [29], where an outer-loop controller generates task-level commands and inner loops regulate motor torques or velocities. This design improves precision and disturbance rejection while retaining fast safety responses. At the joint level, torque control is critical to guarantee safe human interaction [36]. However, as noted by [37], torque sensing alone is insufficient for fine-grained force control in contact-rich environments, motivating integrated strategies that combine sensing, impedance regulation, and high-rate control.

B. Existing Multimanual Architectures

Dual-arm and humanoid platforms impose stricter real-time requirements due to higher degrees of freedom and the need for coordinated control. Middleware frameworks such as ROS and ROS 2 [38], [39], extended with packages like `ros_control` [40], have become de facto standards for modular development, while OROCOS [41] and custom-built architectures provide low-latency real-time support.

ROS is widely used across many robot configurations, including dual-arm systems such as Baxter [19], GARMI [30], and Berkeley Blue [28]. The `ros_control` package [40] enables real-time, hardware-agnostic control, allowing the same code to run seamlessly on different robots. ROS 2 [39] improves on ROS by enhancing real-time capability, security, and robustness, introducing node composition [42] and leveraging data distribution service (DDS) [43] for high-performance data exchange to meet strict timing requirements in modern robotics.

Custom software enables 1 kHz control loops with minimal latency, critical for safety and for bridging the sim2real gap. DLR's bimanual Justin [17] uses a component-based Robot Development concept [44] with modular subsystems and a dedicated real-time system for precise control. Similarly, KIT's ARMAR-6 [26] employs ArmarX [45] for distributed parallel processing and tightly integrated real-time control. Table I summarizes humanoid setups, control frequencies, and software frameworks.

C. Robotic simulators

Simulation plays a crucial role in modern robotics, enabling researchers and engineers to design, test, and validate algorithms in a safe, cost-effective environment. It allows rapid prototyping without the risk of damaging hardware and accelerates development by supporting parallel testing of control strategies, perception pipelines, and interaction scenarios. Moreover, simulation makes it possible to explore edge cases and rare events that would be expensive or unsafe to reproduce on physical robots.

The field of robotics currently supports a wide range of simulators. A few of the most known are Gazebo [46], MuJoCo [47], Webots [48], CoppeliaSim [49], NVIDIA Isaac Sim [50], PyBullet [51], and Pinocchio [52], each designed with specific requirements in mind and varying in fidelity, ease of integration, and computational efficiency

Name	Structure	Year	Command τ_d freq.	Internal freq.	Sensor freq.	Middleware
DLR LWR I [15]	Single	2000	1 kHz	3 kHz	1 kHz	Custom
DLR LWR II [16]	Single	2001	1 kHz	3 kHz	1 kHz	Custom
DLR LWR III [7]	Single	2002	1 kHz	3 kHz	1 kHz	Custom
DLR Justin [17]	Humanoid	2006	1 kHz	3 kHz	1 kHz	Custom
DLR-Kuka LWR [18]	Single	2008	1 kHz	3 kHz	1 kHz	Sunrise.OS
Rethink Robotics Baxter [19]	Bimanual	2011	1 kHz	-	1 kHz	ros_control
Barrett WAM 7DoF [20]	Single	2011	1 kHz	-	$\geq$1 kHz	Custom
Boston Dynamics Atlas 1 [21]	Humanoid	2013	333 Hz	1 kHz	-	Custom
NASA Valkyrie [22]	Humanoid	2013	1 kHz	5 kHz	1 kHz	ros_control
DLR TORO [23]	Humanoid	2014	1 kHz	3 kHz	1 kHz	Custom
Rethink Robotics Sawyer [19]	Single	2015	1 kHz	-	1 kHz	ros_control
PAL Robotics TIAGo [24]	Humanoid	2016	1 kHz	-	1 kHz	ros(2)_control
PAL Robotics TALOS [25]	Humanoid	2017	1 kHz	-	1 kHz	ros_control
KIT ARMAR-6 [26]	Humanoid	2018	1 kHz	-	1 kHz	Custom
Doosan M-Series [27]	Single	2018	-	-	$\geq$1 kHz	Custom
Berkeley Blue [28]	Bimanual	2019	170 Hz	-	20 kHz (Current)	ros_control
Franka Robotics robot [8]	Single	2019	1 kHz	3 kHz	1 kHz	ROS, Custom
DLR SARA [29]	Single	2021	-	8 kHz	-	Custom
GARMI [30]	Humanoid	2021	1 kHz	3 kHz	1 kHz	ros_control
KINOVA Gen3 [31]	Single	2022	1 kHz	-	$\geq$1 kHz	Custom
1X Eve [32]	Humanoid	2023	-	-	-	Custom
Han's Robots Elfin Pro Series [33]	Single	2023	1 kHz	-	$\geq$1 kHz	Custom
Apptronik Apollo [34]	Humanoid	2024	-	-	-	Custom

TABLE I: A non-exhaustive list of torque-controlled commercial and research robots. To be included, a robot must explicitly state that it supports torque or effort command. For commercial robots, only those certified as cobots are considered. Unidentified values are marked with '-'.

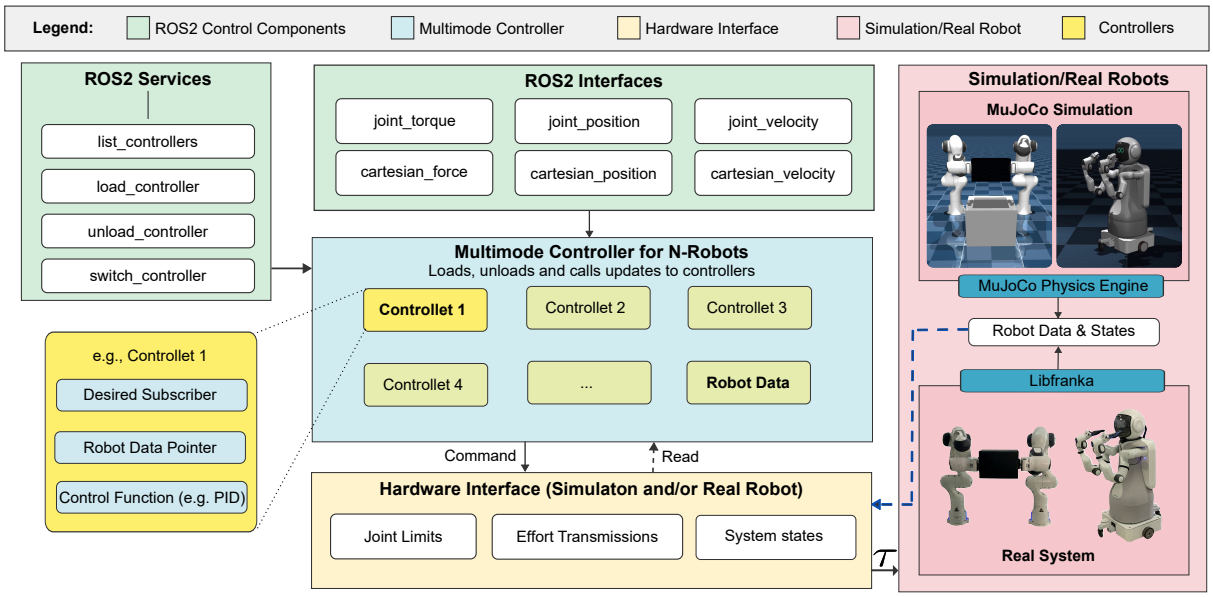


Fig. 2: Architecture diagram of the `multipanda_ros2` within `ros2_control` framework.

[53], [54]. MuJoCo and Pinocchio are often preferred for high-frequency, physics-accurate simulations: MuJoCo for control-intensive tasks and sim-to-real benchmarking, and Pinocchio for fast rigid-body dynamics computations, particularly in multi-body or humanoid systems, as summarized in Table I.

The sim-to-real gap is especially prevalent in the field of reinforcement learning, where policies trained in simulation often fail on real systems. Current work mainly tackles this by making policies robust to discrepancies, for example, through transfer learning and domain adaptation, without improving the simulator itself [55]. To the best of our knowledge, the examples of closing the gap from the simulation side, using real-world data to iteratively update and refine simulators, are scarce. One exception is the work of [14] which utilizes residual physics networks to learn from real data and reduce the sim-to-real gap. However, additional research is needed to see the full effect of this approach on the sim-to-real gap challenge.

III. Implementation

In this section, we present `multipanda_ros2`[1], a software framework developed for the coordinated control of multiple Franka Robotics robots. Our framework is built upon the `ros2_control` infrastructure, as illustrated in Fig. 2, and is fully compatible with the ROS 2 Humble distribution.

The primary goal of `multipanda_ros2` is to provide the existing community of Franka Robotics robot users with easy access to ROS 2, while also enabling the full list of features that are provided through the robot's Franka Control Interface (FCI) and `libfranka`, and to ease the entry into doing bimanual or multimanual torque-control research. The open-source repository includes extensive tutorials and examples to further aid this effort.

The `multimode_controller` (MC), shown in the blue block of Fig. 2, plays a central role in the proposed framework. It is implemented as a typical `ros2_control` controller, meaning it can be used alongside other controllers without issue. The MC itself does not implement a controller; instead, it uses *controllets*. Each controllet takes ownership of one or more robots in the `ros2_control` framework specified at runtime, and implements the actual command torque τ_{cmd} calculation for the loop. The MC can be instantiated with an arbitrary number of such controllets, and handles the robot assignment for all the controllets, ensuring that only one controllet has ownership of a robot at any given time.

The proposed framework has been designed with minimal overhead in mind, making heavy use of move operators and references to avoid unnecessary copying. The MC is instantiated with a vector of controllets upon initialization based on the user's configuration. Each element contains information about the robot that it needs to control, and references to the shared τ_{cmd} vector and robot data that are centrally managed by the MC. Such a design allows a controllet to have real-time access to the controlled robots' data, which is crucial for algorithms that require the full state and dynamic information of the involved robots. Additionally, this ensures that there are minimal copy operations in the framework, so that the τ_{cmd} calculation across controllets and the final commanded torque in the robot's `libfranka` loop stays consistent.

The 1 kHz control loop inside the MC calls the control loop function of the currently active controllets for each step. This enables the implementation of a switching function, which allows the MC to change the currently active controllet at runtime, with a delay of about 2ms (see section VI-A for more details). The controllet switching function is exposed as a ROS 2 service, and the MC ensures that the new set of activated controllers does not have conflicting resource requirements. While it is more desirable to create a single controller that is capable of handling all the tasks that a robot is expected to perform, custom controllers are written for different tasks in practice. As such, this feature allows for complex behaviors in scenarios that require rapid transitions

[1] https://github.com/tenfoldpaper/multipanda_ros2

between different controllers to be implemented effectively.

One additional benefit of the proposed structure is the significantly reduced amount of the required boilerplate coding. The controllets make heavy use of C++ templating and inheritance to achieve this. Typically, a communication-less instance of the controllet is implemented, which performs the desired $\boldsymbol{\tau}_{\text{cmd}}$ calculation for a given desired pose, and defines the parameter update function. Then, this class is inherited by a ROS 2-enabled instance, which implements the callback functions for desired pose subscription and parameter handling. This allows control algorithms to be only implemented once, then inherited to create a wide range of custom ROS 2 instances whose subscriptions match the particular task. Finally, these controllets can then be loaded into the MC with a simple `yaml` file update.

Given that `ros2_control` abstracts away the robot's interfaces, the proposed MC controller is, in fact, agnostic to the model of the robots it is controlling. This means that the proposed framework can be easily extended to other robots that already have their own `ros2_control` frameworks, such as the Baxter [19] and the TIAGo [24]. Additionally, while the framework is currently only implemented for the torque control mode of the Franka Robotic robot, extensions to the other control modes are planned.

A. Simulation Environment

Our primary research focus is physical human–robot interaction (pHRI), which involves high-frequency, control-intensive tasks. Accordingly, we selected the MuJoCo simulator [47] due to its efficient handling of dynamics and contact interactions at rates comparable to real hardware, enabling precise torque control and fast feedback loops essential for stable and responsive pHRI experiments.

The simulation is implemented as a plugin to the work of [56], which wraps the MuJoCo physics engine into a ROS 2 package. This allows development within the same ROS 2 framework used for real hardware, ensuring consistency between simulated and physical experiments. The setup also supports multi-arm simulations, enabling the same controllers to be run on both simulated and real robots. As with the real-hardware framework, control in simulation relies on `ros2_control`, facilitating straightforward extension to other robots that implement their own `ros2_control` interfaces. Following these design principles, a controller needs to be developed only once and can run on both simulation and real hardware without any modifications.

IV. Methods

To evaluate the multimode controller performance and sim-to-real gap in robotic manipulation, we implemented the Unified Force-Impedance Control (UFIC) framework from [57] and a classical Cartesian impedance controller (baseline) for both simulation and real-world experiments using Franka Robotics robots. This comparison quantifies performance differences in force tracking, stability, and energy management, assessing simulation fidelity for real-world deployment.

A. Controller Implementation

Both controllers were coded in C++ with real-time interfacing via the Franka Emika Fast Research Interface (FRI) at 1 kHz. The baseline impedance controller uses:

$$\boldsymbol{\tau}_i = \mathbf{J}^T(\boldsymbol{q})\Big[\mathbf{K}_c\tilde{\boldsymbol{x}} + \mathbf{D}_c\dot{\tilde{\boldsymbol{x}}} + \mathbf{M}_c(\boldsymbol{q})\ddot{\boldsymbol{x}}_d + \mathbf{C}_c(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{x}}_d + \mathbf{F}_g(\boldsymbol{q})\Big]. \tag{1}$$

where $\tilde{\boldsymbol{x}} = \boldsymbol{x}_d - \boldsymbol{x}$, with task torque

$$\boldsymbol{\tau}_{\text{task}} = \mathbf{J}^T(-\mathbf{K}_c\tilde{\boldsymbol{x}} - \mathbf{D}_c(\mathbf{J}\dot{\boldsymbol{q}})) \tag{2}$$

and nullspace torque

$$\boldsymbol{\tau}_{\text{null}} = (\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J})^T(\mathbf{K}_N(\boldsymbol{q}_{d_N} - \boldsymbol{q}) - \mathbf{D}_N\dot{\boldsymbol{q}}) \tag{3}$$

The UFIC extends this with force control:

$$\boldsymbol{\tau}'_m = \boldsymbol{\tau}'_i + \boldsymbol{\tau}'_f, \quad \boldsymbol{\tau}'_f = \mathbf{J}^T(\boldsymbol{q})\mathbf{F}'_f, \tag{4}$$

where $\mathbf{F}'_f = (\gamma_f + \alpha_f(1-\gamma_f))\mathbf{F}_f$ (PID output modulated by tank signals α_f and $\gamma_f = \exp(-\frac{\|\boldsymbol{x}-\boldsymbol{x}_c\|^2}{d^2_{\max}})$), and $\boldsymbol{\tau}'_i$ uses modulated $\tilde{\boldsymbol{x}}' = \boldsymbol{x}'_d - \boldsymbol{x}$ with $\boldsymbol{x}'_d = \boldsymbol{x}_d + \alpha_i(\boldsymbol{x}_d - \boldsymbol{x})$. Tank energies are initialized as $E_{f,0} = \frac{1}{2}\mathbf{K}_p\|\mathbf{F}_d\|^2$, $E_{i,0} = E_{f,0}$.

Dual-arm coordination applied a ± 15 cm y-axis offset to goal poses. Additional features included GJK-based self-collision avoidance ($\boldsymbol{\tau}_{\text{ca}}$ for distances < 0.05 m) and manipulability-based singularity avoidance:

$$m_{\text{kin}}(\boldsymbol{q}) = \sqrt{\det(\mathbf{J}(\boldsymbol{q})\mathbf{J}^T(\boldsymbol{q}))}, \tag{5}$$

$$V_{\text{sing}}(\boldsymbol{q}) = \begin{cases} k_m(m_{\text{kin}}(\boldsymbol{q}) - m_0)^2 & m_{\text{kin}}(\boldsymbol{q}) \le m_0 \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

$$\boldsymbol{\tau}_{\text{ma}} = -\frac{\partial V_{\text{sing}}(\boldsymbol{q})}{\partial \boldsymbol{q}}, \tag{7}$$

with $k_m = 10$, $m_0 = 0.1$. The final command was:

$$\boldsymbol{\tau}_{\text{cmd}} = \boldsymbol{\tau}_{\text{task}} + \boldsymbol{\tau}_{\text{null}} + \boldsymbol{\tau}_{\text{cor}} + \boldsymbol{\tau}_{\text{ca}} + \boldsymbol{\tau}_{\text{ma}}, \tag{8}$$

where $\boldsymbol{\tau}_{\text{cor}} = \mathbf{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \mathbf{g}(\boldsymbol{q})$. Parameters were: $\mathbf{K}_p = 200\mathbf{I}$ N/m, $\mathbf{K}_d = 10\mathbf{I}$ Ns/m, $\mathbf{K}_i = 50\mathbf{I}$ NRs/m; $\mathbf{K}_c = \text{diag}(100, 100, 50, 10, 10, 10)$ N/m; $\mathbf{D}_c$ for critical damping; tank limits $E_{\min} = 0$ J, $E_{\max} = 100$ J; $d_{\max} = 0.05$ m; no friction feedforward ($\mathbf{F}_{\text{FF}} = 0$).

V. Experimental Procedure

A. Controller performance

To evaluate the performance of the proposed framework, we conducted a series of experiments. For this purpose, we implemented a multimode controller with 2 distinct controllets: a dual Cartesian impedance controller (DC), and a coupled dual Cartesian impedance controller (CDC), as detailed in section IV, to command two Franka Robotics robots connected to the same network. To increase the complexity to the control loop calculation, we additionally implemented the self-collision avoidance algorithm from [58] and the manipulability potential algorithm from [59]. This creates 5 conditions that are being tested: 1) the DC controllet with no features (NF), 2) DC with self-collision avoidance (CA), 3) DC with manipulability potential (MA), 4) DC with both features (CA-MA), and 5) the CDC controllet with manipulability (C-MA). These acronyms are used for reference in Fig. 3.

For control cycle time calculation, we calculated the difference between the start and end of the control function, i.e. after the τ_{cmd} were calculated for both robots. The τ_{cmd} delay was measured by recording both the τ_{cmd} and the time at the moment the values are written to the shared memory between `ros2_control` and the robot's `libfranka` loop, and the same values within the `libfranka` control loop. The two datasets were then linearly interpolated across the union of the two datasets' timestamps, and then cross correlation was performed to identify the delay. Finally, for the computer resource measurement, the command `top -bn1` for the control program was executed every 0.3 seconds and its corresponding CPU and memory usage data were recorded. Additionally, we measured the controllet switching delay from 50 trials by measuring the time difference from when the controller receives the switch request to the first control loop of the new controllet. All experiments were conducted on the ASRock Industrial 4x4 BOX 7735U/D NUC mini-PC with 32GB of DDR5-5800 Mhz RAM.

With the above arrangements, the experiment was conducted by running each controllet for 30 seconds. At the same time, desired poses were calculated and published at 1 kHz on a separate node that implements the circular field motion generator from [13]. The metrics were measured at each cycle, resulting in 30000 data points each for the first 2 metrics, and 100 data points for the resource metric.

B. Simulation Validation

To evaluate the fidelity of the proposed simulation environment, we designed a series of experiments aimed at progressively challenging the kinematic and dynamic models of a serial manipulator. The rationale behind this staged approach is twofold. First, simple tasks allow us to isolate the effects of pure kinematic modeling errors. Second, gradually introducing dynamic phenomena, such as inertial effects and contact interactions, enables a structured evaluation of how the simulator captures increasingly complex physical behaviors.

To rapidly quantify deviations between simulated and real robot behavior, we employ a set of straightforward yet informative metrics. The primary measure is the root mean square error (RMSE) between the recorded simulation data and experimental measurements, selected for its simplicity and interpretability. Errors are evaluated over the following key variables: $\boldsymbol{q}$ (joint positions), $\dot{\boldsymbol{q}}$ (joint velocities), $\boldsymbol{\tau}$ (joint torques), $\boldsymbol{x}_{\text{EE}}$ (end-effector position), $\boldsymbol{F}_{\text{EE}}$ (end-effector force), and C_{err} (control error) [2], providing a comprehensive view of perfomance of both kinematics and dynamics.

We designed five validation tasks, each introducing an increasing level of kinematic and dynamic complexity. Each experiment was executed for 10 seconds and repeated 5 times, averaging the results across trials.

Task 1: Executes the default joint impedance example controller (following equation 1), provided by the manufacturer, controlling the robot in free space.

Task 2: Executes the default Cartesian impedance example controller (following equation 1), provided by the manufacturer, controlling the robot in free space.

Task 3: Executes a modified force example controller, provided by the manufacturer, which uses a variable desired contact force perpendicular to a flat surface, following a sinusoidal curve between 0 and 30 N for one period, instead of using a constant desired force.

Task 4: Executes a custom unified force impedance controller as formulated in equation 4, which maintains a constant 9.81 N contact force perpendicular to a flat surface (in Z-direction) while following a circular trajectory in X-Y plane with radius of 10 cm.

Task 5: Executes a custom coupled unified force impedance controller by extending the formulation of equation 4 for two arms in order to coordinate them in grasping a box from opposite sides. Each manipulator maintains a constant 20 N force perpendicular to the box surface while jointly following a 20 cm vertical trajectory.

Lastly, to illustrate the potential for improving the simulation fidelity using real-world data, we perform an identification procedure on the real system following the steps of [60] to update the simulated robot's inertial parameters before repeating task 4.

VI. Results

A. Controller performance results

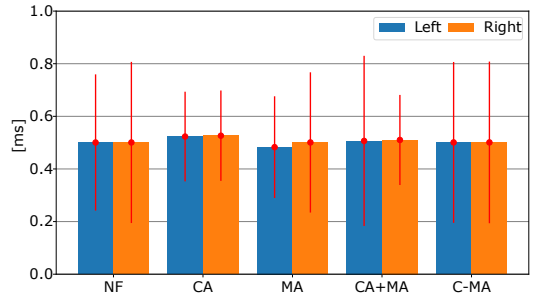
The results of the controller performance experiments are summarized in Fig. 3. We recorded the following metrics: 1) control loop cycle time, 2) time it takes for the computed command torque τ_{cmd} to reach the robot, and 3) system resource consumption. The above subfigure (3a) shows the average τ_{cmd} delay for each arm which is on average approximately 0.5 ms. This can be explained by the fact that the `ros2_control`'s and the `libfranka`'s control loops run on separate threads. From this, it can be ascertained that most new τ_{cmd} are consumed immediately by the robot's control loop promptly, with occasional single control cycle delays.

The average loop time is presented in Fig. 3b. The baseline DC controllet requires less than 25μs per cycle, while the collision avoidance algorithm introduces a noticeable computational overhead, as expected given the presence of two nested `for` loops (one per arm). In contrast, the manipulability potential calculation adds approximately 40–60 μs, depending on the condition. Nevertheless, the overall control loop remains sufficiently short to sustain the 1 kHz target frequency, leaving room for additional features before reaching this limit.

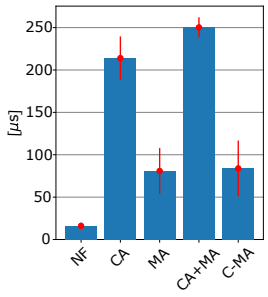
The CPU usage, shown in Fig. 3c, corresponds to a single core of the 16-core PC. Across all conditions, memory usage remained constant at 0.2%, constrained by the single-decimal precision of the `top` command. Overall, these results indicate that the proposed framework consumes only a modest amount of resources and does not hinder the execution of other tasks.

The controllet switching experiment yielded an average of 2.117ms with a standard deviation of 0.493ms, indicating that it takes about 2 control cycles before the new controllet is activated.

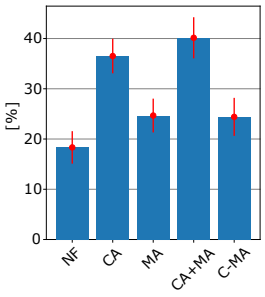
[2] Control error: defined as the difference between the commanded and achieved joint torque values.



(a) Average τ_{cmd} delay.



(b) Average loop time.



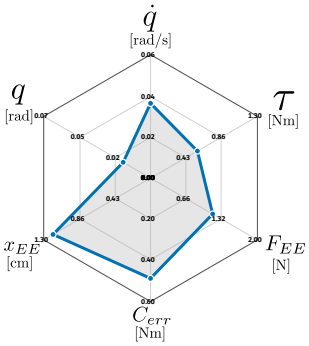
(c) Average 1-core CPU usage.

Fig. 3: The metrics measured to determine the performance of the proposed framework. The x-axis represents the controllet-feature condition.

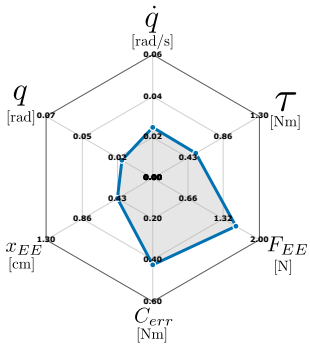
B. Simulation validation results

The simulation fidelity metrics for the single-arm experiments are shown in Fig. 4, while results for the bimanual task are reported in Fig. 5. The results show that the discrepancy between simulation and reality is up to 0.06 rad, 0.05 rad/s, and 1.3 cm for joint positions, joint velocities, and EE position, respectively. For dynamics, the differences are up to 1.3 Nm for joint torques, 1.7 N for detected external forces, and 0.6 Nm for the control error. It is important to note that the real system exhibited noticeable sensor noise during no-contact tasks (tasks 1 and 2), which introduces a certain error for the τ and F_{EE} metrics that is not considered by default in the simulation. For the bimanual task (task 5), Fig. 5 reveals pronounced discrepancies between the two manipulators, particularly in EE position, joint torques, and control error. Although both arms are of the same model, the differences indicate that simulation fidelity may vary substantially even among nominally identical systems. Likely causes include manipulator age and temperature-dependent effects on actuators and sensors. These observations underscore the necessity of incorporating real-system data to account for such variations when addressing the sim-to-real gap.

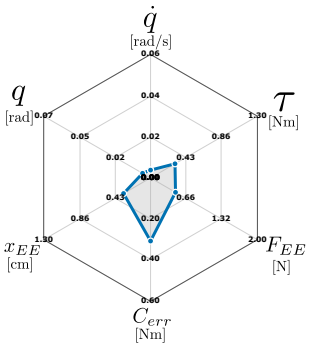
The final validation results, shown in Fig. 6, illustrate the effect of updating simulation parameters through a real-world identification procedure before repeating experiment 4. The red line indicates a reduced error between the simulation and the real system compared to the default experiment (blue line). For instance, the recorded external force error is reduced by nearly half. These results complement the findings from task 5 and suggest that incorporating identification-based parameter tuning into the simulation can be an effective approach to narrowing the sim2real gap.



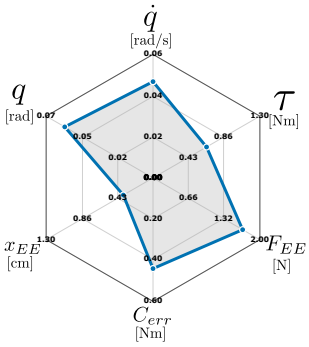
(a) Task 1



(b) Task 2

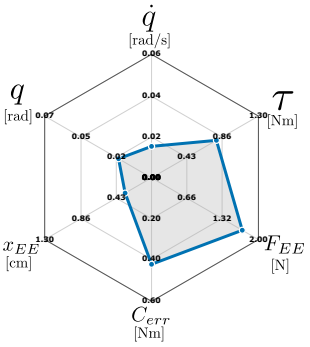


(c) Task 3

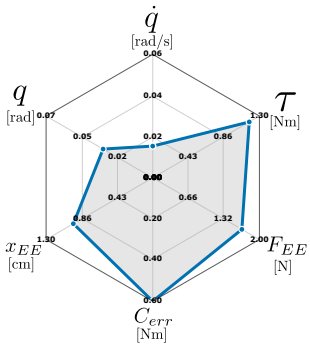


(d) Task 4

Fig. 4: Simulation error metrics results for the first four experiments: a) Free motion using a joint impedance example controller, b) Free motion using a Cartesian impedance example controller, c) Variable force holding, d) Circle trajectory following while maintaining constant contact force.



(a) Left manipulator



(b) Right manipulator

Fig. 5: Simulation error metrics results for the a) left and b) right manipulator during the bimanual box grasping task with constant contact force during lifting (task 5).

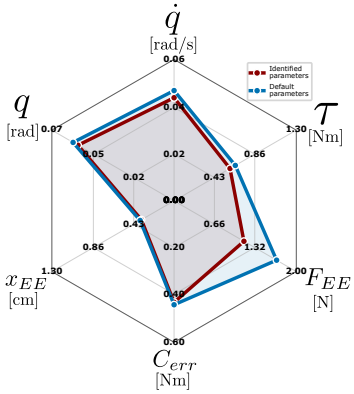


Fig. 6: Simulation error metrics results comparison between the default (blue) and updated (red) simulation parameters during task 4 (circle trajectory following with constant contact force).

VII. Discusion and Conclusion

In this work, we introduced `multipanda_ros2`, a novel architecture for real-time multi-robot control for the Franka Robotics robots within the ROS 2 framework. The framework integrates the full capabilities of the robot offered by `libfranka`, providing an easy entry to bimanual or multimanual torque control research to the community.

First, we introduce an architecture that provides reliable, low-latency control, with controllet-feature switching delays of $\leq$2 ms. This capability is crucial for high-frequency applications and aligns with the stringent requirements of physical human-robot interaction and robot-environment modeling, which require 1 kHz command loops for safety and accurate dynamics [8], [35]. Our framework not only simplifies the implementation of complex controllers but also serves as a benchmark for reproducible research, mirroring the component-based designs.

Second, we integrate a high-fidelity MuJoCo simulation environment to tackle the sim2real gap. We provide a comprehensive set of metrics to evaluate both kinematic accuracy and dynamic consistency, a contribution that extends beyond the common practice of focusing on kinematic data alone [55]. This approach allows researchers to assess simulation fidelity in terms of torques, forces, and control errors, which are critical for tasks involving physical contact.

Furthermore, we assess force-based fidelity by extending soft-robotics approaches [14] to rigid dual-arm, contact-rich tasks. Through real-world inertial parameter identification, we demonstrate that incorporating data from physical systems can significantly reduce the sim2real gap, leading to improved force and torque accuracy in the simulated environment. This iterative physics refinement process offers a promising method for continuously improving simulation models, directly addressing a key limitation in current research, akin to recent approaches using residual physics networks [14].

While the proposed architecture demonstrates notable benefits, we acknowledge its limitations. Currently, the framework is exclusively tailored to the Franka platform and evaluated across a restricted set of validation scenarios. Furthermore, our sim2real refinement relies entirely on offline physical parameter identification, lacking the online adaptive capabilities necessary to compensate for dynamic, real-time environmental changes.

As future work, we plan to explore more complex validation scenarios and develop additional validation metrics that remain accessible to the broader robotics community. We will also investigate advanced techniques to further reduce the sim2real gap, thereby accelerating progress in robotics research. Specifically, we intend to leverage this tight integration to deploy the framework in digital twin applications, enabling real-time synchronization, monitoring, and predictive control of physical systems, e.g., [61]. Finally, motivated by the strong interest already observed in the online repository, we will continue to expand and maintain this open-source framework to encourage wider adoption. Ultimately, we hope that `multipanda_ros2` will be utilized to drive tangible, real-world use cases beyond the laboratory setting.

References

[1] D. E. Whitney, "Force Feedback Control of Manipulator Fine Motions," *Journal of Dynamic Systems, Measurement, and Control*, vol. 99, no. 2, pp. 91–97, 1977.

[2] A. Gasparetto and L. Scalera, "A Brief History of Industrial Robotics in the 20th Century," *Advances in Historical Studies*, vol. 8, no. 1, pp. 24–35, 2019.

[3] E. I. George, T. C. Brand, A. LaPorta, J. Marescaux, and R. M. Satava, "Origins of Robotic Surgery: From Skepticism to Standard of Care," *JSLS: Journal of the Society of Laparoendoscopic Surgeons*, vol. 22, no. 4, p. e2018.00039, 2018.

[4] J. K. Salisbury, "Active stiffness control of a manipulator in cartesian coordinates," in *IEEE Conf. on Decision and Control*, 1980, pp. 95–100.

[5] M. H. Raibert and J. J. Craig, "Hybrid Position/Force Control of Manipulators," *Journal of Dynamic Systems, Measurement, and Control*, vol. 103, no. 2, pp. 126–133, 1981.

[6] I. Troch and P. Kopacek, "Control Concepts and Algorithms for Flexible Robots — An Expository Survey," *IFAC Proceedings Volumes*, vol. 21, no. 16, pp. 29–34, 1988.

[7] G. Hirzinger, N. Sporer, A. Albu-Schaffer, M. Hahnle, R. Krenn, A. Pascucci, and M. Schedl, "DLR's torque-controlled light weight robot III-are we reaching the technological limits now?" in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE, 2002, pp. 1710–1716.

[8] S. Haddadin, S. Parusel, L. Johannsmeier, S. Golz, S. Gabl, F. Walch, M. Sabaghian, C. Jähne, L. Hausperger, and S. Haddadin, "The Franka Emika Robot: A Reference Platform for Robotics Research and Education," *IEEE Robotics and Automation Magazine*, vol. 29, no. 2, pp. 46–64, 2022.

[9] N. Hogan, "Impedance Control: An Approach to Manipulation," in *American Control Conference*, 1984, pp. 304–313.

[10] W. S. Newman, "Stability and Performance Limits of Interaction Controllers," *Journal of Dynamic Systems, Measurement, and Control*, vol. 114, no. 4, pp. 563–570, 1992.

[11] A. Albu-Schäffer, C. Ott, and G. Hirzinger, "A Unified Passivity-based Control Framework for Position, Torque and Impedance Control of Flexible Joint Robots," *Int. Journal of Robotics Research*, vol. 26, no. 1, pp. 23–39, 2007.

[12] S.-J. Chung and J.-J. E. Slotine, "Cooperative Robot Control and Concurrent Synchronization of Lagrangian Systems," *IEEE Trans. on Robotics*, vol. 25, no. 3, pp. 686–700, 2009.

[13] R. Laha, M. Becker, J. Vorndamme, J. Vrabel, L. F. Figueredo, M. A. Müller, and S. Haddadin, "Predictive Multi-Agent-Based Planning and Landing Controller for Reactive Dual-Arm Manipulation," *IEEE Trans. on Robotics*, vol. 40, pp. 864–885, 2024.

[14] J. Gao, M. Y. Michelis, A. Spielberg, and R. K. Katzschmann, "Sim-to-real of soft robots with learned residual physics," *IEEE Robotics and Automation Letters (RA-L)*, vol. 9, no. 10, pp. 8523–8530, 2024.

[15] A. Albu-Schaffer and G. Hirzinger, "State feedback controller for flexible joint robots: a globally stable approach implemented on DLR's light-weight robots," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, vol. 2, 2000, pp. 1087–1093.

[16] G. Hirzinger, A. Albu-Schaffer, M. Hahnle, I. Schaefer, and N. Sporer, "On a new generation of torque controlled light-weight robots," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, vol. 4, 2001, pp. 3356–3363.

[17] C. Ott, O. Eiberger, W. Friedl, B. Bauml, U. Hillenbrand, C. Borst, A. Albu-Schaffer, B. Brunner, H. Hirschmuller, S. Kielhofer, R. Konietschke, M. Suppa, T. Wimbock, F. Zacharias, and G. Hirzinger, "A Humanoid Two-Arm System for Dexterous Manipulation," in *IEEE/RAS Intl. Conf. on Humanoid Robots*. IEEE, 2006.

[18] R. Bischoff, J. Kurth, G. Schreiber, R. Koeppe, A. Albu-Schaeffer, A. Beyer, O. Eiberger, S. Haddadin, A. Stemmer, G. Grunwald, and G. Hirzinger, "The KUKA-DLR Lightweight Robot arm - a new reference platform for robotics research and manufacturing," in *ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics)*, 2010, pp. 1–8.

[19] Rethink Robotics, "Arm Control System," 2022, accessed: 2024-12-10. [Online]. Available: https://support.rethinkrobotics.com/support/solutions/articles/80000980284-arm-control-system

[20] B. T. Inc., "WAM Arm data sheet," Barret Technology Inc., Tech. Rep., 2011, accessed: 2025-08-30. [Online]. Available: https://web.barrett.com/files/WAMDataSheet_02.2011.pdf

[21] S. Feng, E. Whitman, X. Xinjilefu, and C. G. Atkeson, "Optimization based full body control for the atlas robot," in *IEEE/RAS Intl. Conf. on Humanoid Robots*. IEEE, 2014, pp. 120–127.

[22] N. A. Radford, P. Strawser, K. Hambuchen, J. S. Mehling, W. K. Verdeyen, A. S. Donnan, J. Holley, J. Sanchez, V. Nguyen, L. Bridgwater, R. Berka, R. Ambrose, M. Myles Markee, N. J. Fraser-Chanpong, C. McQuin, J. D. Yamokoski, S. Hart, R. Guo, A. Parsons, B. Wightman, P. Dinh, B. Ames, C. Blakely, C. Edmondson, B. Sommers, R. Rea, C. Tobler, H. Bibby, B. Howard, L. Niu, A. Lee, M. Conover, L. Truong, R. Reed, D. Chesney, R. Platt, G. Johnson, C. Fok, N. Paine, L. Sentis, E. Cousineau, R. Sinnet, J. Lack, M. Powell, B. Morris, A. Ames, and J. Akinyode, "Valkyrie: NASA's First Bipedal Humanoid Robot," *Journal of Field Robotics*, vol. 32, no. 3, pp. 397–419, 2015.

[23] B. Henze, A. Werner, M. A. Roa, G. Garofalo, J. Englsberger, and C. Ott, "Control applications of toro – a torque controlled humanoid robot," in *IEEE/RAS Intl. Conf. on Humanoid Robots*. IEEE, 2014, pp. 841–841.

[24] J. Pagès, L. Marchionni, and F. Ferro, "Tiago: the modular robot that adapts to different research needs," 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:218478582

[25] O. Stasse, T. Flayols, R. Budhiraja, K. Giraud-Esclasse, J. Carpentier, J. Mirabel, A. Del Prete, P. Souères, N. Mansard, F. Lamiraux *et al.*, "TALOS: A new humanoid research platform targeted for industrial applications," in *IEEE/RAS Intl. Conf. on Humanoid Robots*, 2017, pp. 689–695.

[26] T. Asfour, L. Kaul, M. Wachter, S. Ottenhaus, P. Weiner, S. Rader, R. Grimm, Y. Zhou, M. Grotz, F. Paus, D. Shingarey, and H. Haubert, "ARMAR-6: A Collaborative Humanoid Robot for Industrial Environments," in *IEEE/RAS Intl. Conf. on Humanoid Robots*. IEEE, 2018.

[27] Doosan Robotics, "M0609: 6kg Payload, 900mm Reach, ±0.03mm Repeatability Robot," 2024, accessed: 2024-12-10. [Online]. Available: https://www.doosanrobotics.com/en/products/series/m0609

[28] D. V. Gealy, S. McKinley, B. Yi, P. Wu, P. R. Downey, G. Balke, A. Zhao, M. Guo, R. Thomasson, A. Sinclair, P. Cuellar, Z. McCarthy, and P. Abbeel, "Quasi-direct drive for low-cost compliant robotic manipulation," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2019, pp. 437–443.

[29] M. Iskandar, C. Ott, O. Eiberger, M. Keppler, A. Albu-Schaffer, and A. Dietrich, "Joint-Level Control of the DLR Lightweight Robot SARA," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 8903–8910.

[30] M. Trobinger, C. Jahne, Z. Qu, J. Elsner, A. Reindl, S. Getz, T. Goll, B. Loinger, T. Loibl, C. Kugler, C. Calafell, M. Sabaghian, T. Ende, D. Wahrmann, S. Parusel, S. Haddadin, and S. Haddadin, "Introducing GARMI - A Service Robotics Platform to Support the Elderly at Home: Design Philosophy, System Overview and First Results," *IEEE Robotics and Automation Letters (RA-L)*, vol. 6, no. 3, pp. 5857–5864, 2021.

[31] K. Inc., "Kinova Gen3," Kinova Inc., Tech. Rep., 2022, accessed: 2025-08-30. [Online]. Available: https://drive.google.com/file/d/1UbCKUgOjKqTvQNwXWN5gIdBiP9Q4hP5d/view

[32] 1X Technologies, "EVE by 1X Technologies: The Smarter Working Android," 2023, accessed: 2025-08-30. [Online]. Available: https://www.1x.tech/androids/eve

[33] H. Robots, "Han's Robots Elfin Pro," 2023, accessed: 2025-08-30. [Online]. Available: https://www.hansrobot.net/elfin-pro-collaborative-robot

[34] Apptronik, "Apollo: General Purpose Humanoid Robot," 2024, accessed: 2025-08-30. [Online]. Available: https://apptronik.com/apollo

[35] I. O. for Standardization (ISO), "ISO 10218-2:2025 – Robotics -– Safety Requirements for robot systems in an industrial environment —– Part 2: Robot systems, robot applications and robot cells integration," 2025.

[36] S. Haddadin, A. Albu-Schäffer, and G. Hirzinger, "Safety evaluation of physical human-robot interaction via crash-testing." in *Robotics: Science and systems*, vol. 3, 2007, pp. 217–224.

[37] R. J. Kirschner, A. Kurdas, K. Karacan, P. Junge, S. A. Baradaran Birjandi, N. Mansfeld, S. Abdolshah, and S. Haddadin, "Towards a Reference Framework for Tactile Robot Performance and Safety Benchmarking," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2021.

[38] A. Koubâa *et al.*, *Robot Operating System (ROS)*. Springer, 2017, vol. 1.

[39] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot Operating System 2: Design, architecture, and uses in the wild," *Science Robotics*, vol. 7, no. 66, p. eabm6074, 2022.

[40] S. Chitta, E. Marder-Eppstein, W. Meeussen, V. Pradeep, A. Rodríguez Tsouroukdissian, J. Bohren, D. Coleman, B. Magyar, G. Raiola, M. Lüdtke, and E. Fernández Perdomo, "ros_control: A generic and simple control framework for ROS," *The Journal of Open Source Software*, 2017.

[41] H. Bruyninckx, "Open robot control software: the OROCOS project," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, vol. 3, 2001, pp. 2523–2528.

[42] S. Macenski, A. Soragna, M. Carroll, and Z. Ge, "Impact of ROS 2 Node Composition in Robotic Systems," *IEEE Robotics and Automation Letters (RA-L)*, vol. 8, no. 7, pp. 3996–4003, 2023.

[43] Y. Liu, Y. Guan, X. Li, R. Wang, and J. Zhang, "Formal Analysis and Verification of DDS in ROS2," in *2018 16th ACM/IEEE International Conference on Formal Methods and Models for System Design (MEMOCODE)*. IEEE, 2018, pp. 1–5.

[44] B. Bauml and G. Hirzinger, "Agile Robot Development (aRD): A Pragmatic Approach to Robotic Software," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2006.

[45] N. Vahrenkamp, M. Wächter, M. Kröhnert, K. Welke, and T. Asfour, "The robot software framework ArmarX," *it - Information Technology*, vol. 57, no. 2, pp. 99–111, 2015.

[46] N. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, vol. 3, 2004, pp. 2149–2154.

[47] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2012, pp. 5026–5033.

[48] O. Michel, "Webots: Symbiosis Between Virtual and Real Mobile Robots," in *Virtual Worlds*. Springer Berlin Heidelberg, 1998, pp. 254–263.

[49] E. Rohmer, S. P. N. Singh, and M. Freese, "V-REP: A versatile and scalable robot simulation framework," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2013, pp. 1321–1326.

[50] NVIDIA, "Isaac Sim," accessed: 2025-08-30. [Online]. Available: https://github.com/isaac-sim/IsaacSim

[51] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," 2016.

[52] J. Carpentier, G. Saurel, G. Buondonno, J. Mirabel, F. Lamiraux, O. Stasse, and N. Mansard, "The Pinocchio C++ library: A fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives," in *2019 IEEE/SICE International Symposium on System Integration (SII)*, 2019, pp. 614–619.

[53] J. Collins, S. Chand, A. Vanderkop, and D. Howard, "A Review of Physics Simulators for Robotic Applications," *IEEE Access*, vol. 9, pp. 51 416–51 431, 2021.

[54] A. Farley, J. Wang, and J. A. Marshall, "How to pick a mobile robot simulator: A quantitative comparison of CoppeliaSim, Gazebo, MORSE and Webots with a focus on accuracy of motion," *Simulation Modelling Practice and Theory*, vol. 120, p. 102629, 2022.

[55] W. Zhao, J. P. Queralta, and T. Westerlund, "Sim-to-Real Transfer in Deep Reinforcement Learning for Robotics: a Survey," in *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2020, pp. 737–744.

[56] D. P. Leins, R. Haschke, and H. Ritter, "MuJoCo ROS: Integrating ROS with the MuJoCo Engine for Accurate and Scalable Robotic Simulation," in *2025 IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR)*, 2025, pp. 1–6.

[57] S. Haddadin and E. Shahriari, "Unified force-impedance control," *The International Journal of Robotics Research*, vol. 43, no. 13, pp. 2112–2141, 2024.

[58] H. Täubig, B. Bäuml, and U. Frese, "Real-Time Swept Volume and Distance Computation for Self Collision Detection," 2011.

[59] A. Dietrich, T. Wimbock, A. Albu-Schaffer, and G. Hirzinger, "Integration of reactive, torque-based self-collision avoidance into a task hierarchy," *IEEE Trans. on Robotics*, vol. 28, no. 6, pp. 1278–1293, 2012.

[60] M. Tröbinger, A. Naceri, X. Chen, H. Sadeghian, and S. Haddadin, "Identification of a Generalized Base Inertial Parameter Set of Robotic Manipulators Considering Mounting Configurations," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2023, pp. 11 502–11 508.

[61] J. Škerlj, M. Hamad, J. Elsner, A. Naceri, and S. Haddadin, "Safe-by-design digital twins for human-robot interaction: A use case for humanoid service robots," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 13 369–13 375.