Control of a Robotic Telehandler for Sandwich Panel Assembly: High-Fidelity Digital Twin and Fractional Predictive Controller

Lucas P. Costa[1], **João P. Alves**[2], **Nuno M. F. Ferreira**[2], **Micael S. Couceiro**[1]

[1]Ingeniarius, Lda., R. Nossa Senhora da Conceição, 146, 4445-147, Alfena, Portugal
[2]Polytechnic University of Coimbra, Rua Pedro Nunes, Quinta da Nora 3030-199 Coimbra, Portugal

lucas@ingeniarius.pt, jpalves@isec.pt, nunomig@isec.pt, micael@ingeniarius.pt

Abstract -

On-site sandwich-panel assembly demands safe, precise manipulation by heavy-duty equipment operating under dynamic, partially known constraints. We present motion-control design and validation for a retrofitted rotating telehandler. A ROS2-compatible Unity digital twin is used for safe, repeatable tuning. The twin embeds joint-level non-linearities (backlash, friction, compliance, saturation) to reduce sim-to-real mismatch. On top of this simulator, we implement two joint-space controllers: a baseline PID and a fractional-order PID with a light predictive term, here denoted as FOPID-P. Controllers are tuned in simulation and evaluated on step and frequency response, followed by a zero-shot transfer of the controller to the real telehandler. Results indicate that twin-first tuning supports faster, safer commissioning and reproducible controller comparison in a construction-relevant setting. This workflow aligns with sustainable, smart, and lean construction by reducing risky on-site iterations and unnecessary machine idling.

Keywords -

Construction robotics; Robotic telehandler; ROS; Digital twin; Unity; Fractional-order control; predictive controller; PID; Sandwich panel assembly; Sim-to-real.

1 Introduction

On-site assembly of prefabricated envelopes, such as sandwich panels, increasingly relies on heavy-duty equipment operating in close proximity to humans and under changing, partially known constraints. This context concentrates several enduring challenges of construction robotics, including the need for safe human-machine collaboration in unstructured environments, the difficulty of achieving precise manipulation with high-inertia actuation, and the persistent gap between laboratory validation and field deployment [1]. Recent initiatives have shown the feasibility and benefits of introducing robotic capabilities into construction workflows, including our previously introduced pilot that positions a rotating telehandler as the main manipulator for façade panel assembly [2]. Yet, moving from proof-of-concept to reliable on-site operation demands control strategies that explicitly account for the non-linearities of heavy machinery, such as backlash, joint flexibility, friction, and saturation.
While construction robotics has progressed rapidly in perception and task planning [3], reliable low-level control for heavy machines remains under-explored. We therefore compare a standard PID baseline against a fractional controller [4, 5] in a high-fidelity ROS2 digital twin, and validate zero-shot transfer to a real telehandler. The paper addresses the specific and industrially relevant problem of controlling a retrofitted rotating telehandler (Manitou MRT 2260) for the precise, safe handling of sandwich panels. Rather than attempting the entire human-robot collaboration (HRC) assembly pipeline (task allocation, perception-driven grasping, human-robot interaction protocols), we focus on the motion-control layer as a prerequisite for any trustworthy deployment. Concretely, we (i) introduce a ROS2-compatible digital twin developed in Unity that emphasises joint-level realism by modelling backlash, hysteresis, flexibility, friction, and saturation; (ii) implement two joint-space controllers as MoveIt2 controllers, a baseline PID and a fractional-order predictive PID (FOPID-P), tuned in simulation via trial-and-error; and (iii) evaluate both in several experiments using standard metrics, followed by zero-shot transfer to the real machine with the same tuned parameters. The proposed approach shifts the most hazardous and wasteful parts of commissioning into a high-fidelity digital twin, reducing on-site iterations (lean), enabling data-driven supervision and controller selection (smart), and lowering unnecessary machine idling and rework (sustainable).

2 Related Work

Broad surveys of construction robotics document a shift toward HRC and on-site autonomy, while also noting that field results emphasise perception and task-level planning rather than low-level control [6]. Heavy equipment, such as telehandlers, concentrate the core control challenges: hydraulic actuation with strong non-linearities, long kinematic chains, and coupled dynamics that degrade tracking and amplify risks near humans. Most validation of control architectures within these machines remains in standardised cycles or simplified testbeds [7].

Parallel advances in "robotising" machinery show that retrofitting is a practical route to research translation [3]. Beyond platform enablement, surveys of emerging technologies for heavy-duty machines emphasise perception stacks, connectivity, and digital twinning; they also recognise the difficulty of closing the loop from high-level information to low-level motion, where actuator physics and structural compliance dominate error budgets [8].

On the control side, PID remains the *de facto* baseline due to transparency and ease of commissioning. However, fractional-order control has repeatedly shown improved robustness to modelling errors and joint non-idealities [5]. In hydraulic settings, impedance/time-delay strategies have also been explored to mitigate unmodelled dynamics and oscillations, but their adoption hinges on reliable plant models and repeatable commissioning procedures [7].

Reducing the sim-to-real gap is therefore central to safe deployment. Surveys emphasise the need for high-fidelity structured modelling, domain randomisation and disturbance injection to expose controllers to variability during tuning before field trials [9]. In construction, several works emphasise layout/progress monitoring rather than joint-level dynamics fidelity critical for controller transferability [10]. This is a key limitation for heavy-duty robots, where gear trains, long booms, and hydraulic effects dominate error and safety margins.

In summary, the literature establishes (i) the importance of retrofitting heavy equipment for on-site robotics, (ii) the promise of robust controllers, and (iii) the need for high-fidelity digital twins to support safe commissioning and transfer. To the best of our knowledge, a telehandler-centric study that combines an explicit modelling of joint-level non-linearities, a ROS2-compatible digital twin, and a MoveIt2-integrated comparison of PID vs. fractional-order PID, has not yet been reported.

3 Problem Statement

Modern façade assembly with prefabricated sandwich panels requires heavy-duty machines to manipulate long loads in proximity to humans and partially known constraints. Reliable joint-space tracking must remain safe under non-linearities, variable inertias, and delays. At the same time, commissioning must be de-risked through credible simulation so that controller parameters tuned off-site transfer with minimal effort to the field. This paper therefore asks whether a fractional-order controller, tuned in a high-fidelity digital twin that explicitly embeds joint-level non-idealities, can deliver robust, zero-shot motion control for a retrofitted rotating telehandler, thereby laying a dependable foundation for subsequent perception, HRI, and task-level automation.

3.1 The CRIARTE R&D Project

The CRIARTE project[1] is a 36-month industry-academia R&D initiative focused on Construction 5.0, uniting a construction company, a construction collaborative laboratory, a robotics SME, and two higher-education institutes to push on-site human-robot collaboration from proof-of-concept towards operational relevance. CRIARTE targets perception, safe co-manipulation, mixed-reality interaction, scan-to-BIM updating, and a ROS2-based digital-twin/fleet stack for on-site HRC. The project builds upon European pilots and testbeds, including FORTIS[2], but goes beyond them by explicitly addressing construction-specific challenges, technologies, and pipelines - this includes joint-level dynamics, integrating high-fidelity simulation into the development loop.

3.2 Use Case: Sandwich Panel Assembly

The use case concerns on-site façade assembly with sandwich panels using a robotic telehandler. A ground worker attaches a vacuum cup to the panel and the telehandler, positioned and stabilised with outriggers, hands over to an autonomous mode. The control-critical phases addressed in this paper are controller responses where the robot regulates the panel's joint-space motion under uncertain inertial properties and non-linear actuation. Interaction channels and higher-level task planning are outside scope; here we assume reference trajectories and basic panel properties are available under controlled conditions.

3.3 Robot: Retrofitted Rotating Telehandler

The telehandler was retrofitted to operate as a ROS2-enabled robotic platform, following a proven pathway for heavy-duty machines [3]. An embedded gateway bridges the electro-hydraulic controller over the J1939 CAN bus to ROS2 nodes, enabling joint-space commands and streaming of time-synchronised telemetry. Two industrial CANopen encoders were added on previously uninstrumented joints, the basket (single-turn) and winch (multi-turn), and integrated via custom firmware into the machine CAN stack (Fig. 1). A modular sensing and compute payload (3D LiDAR and stereo vision) is installed for artificial perception but is not exercised in the present study; here, the critical signals are joint positions, with actuation routed through the OEM hydraulics under ROS2 control. This configuration yields full joint observability and a deterministic actuation/measurement path for the controllers evaluated later, while keeping the perception stack available for future HRC and planning layers.

[1]https://www.p2030criarte.com/
[2]https://fortis-project.eu/

Figure 1. Robotic telehandler retrofitted into a ROS2-enabled robot with multimodal sensing. Other sensors shown besides the additionally added encoders are not exercised in this control study.

4 Digital Twin

High-fidelity digital twins are central to commissioning heavy-duty, highly non-linear machines without incurring site risk. For telehandlers with long, compliant kinematic chains and hydraulic actuation, a simulator that resolves joint-level effects enables safe tuning, repeatable regression tests, and credible zero-shot transfer to the field.

4.1 Unity-ROS2 Framework

We implement the digital twin in Unity, leveraging the PhysX engine (ArticulationBody joints) for stable, constraint-consistent multibody dynamics [11]. Unity exchanges messages with ROS2 through the ROS-TCP Connector, which instantiates a lightweight TCP server and auto-generates C# bindings for ROS2 messages [12]. To mirror the machine's ROS2–CAN interface, Unity subscribes to joint command topics (e.g., `telehandlerjoint_cmds` as `std_msgsmsgInt8MultiArray`) and publishes encoder feedback on `telehandlerjoint_states` (`sensor_msgsmsgJointState`). This 1:1 topic/messaging parity allows controllers integrated in MoveIt2 to be exercised against the twin or the real telehandler without code changes, and facilitates deterministic logging and repeatable closed-loop tests.

4.2 Telehandler Digital Model

The telehandler is represented in URDF as a 9-joint articulated mechanism anchored at `base_link`: two proximal revolutes (`base_to_cylinder_joint`, `cylinder_to_arm1_joint`); three prismatic sections forming the telescopic boom (`arm_link1_to_arm2_joint`, `arm_link2_to_arm3_joint`, `arm_link3_to_arm4_joint`); two distal revolutes at the fork/basket interface (`arm_link4_to_armFork_joint`, `basket_link_to_armFork_joint`); and a basket assembly with one revolute and one prismatic axis (`basket_crane_link_to_basket_link`, `winch_link_to_basket_crane_link`) (Fig. 2). In the retrofitted machine, the three prismatic boom joints are driven by the native hydraulic controller as a single extension axis; consequently, the ROS2 control layer exposes one virtual *boom extension* DoF. This yields 6 commanded DoF overall (excl. `base_link`). The same joint names and frames are used in MoveIt2 (kinematics/planning) and in Unity (via `ArticulationBody`), ensuring that the same ROS2 controllers operate seamlessly in simulation and on the physical platform.

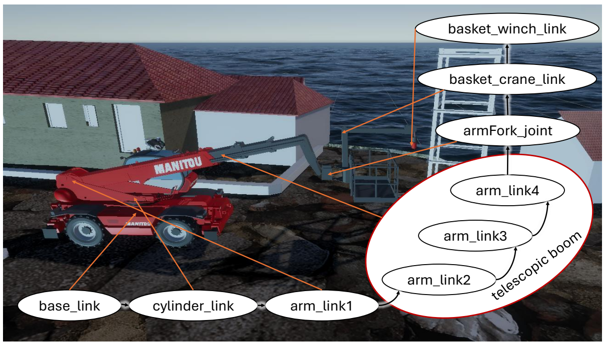

Figure 2. Digital model of telehandler and construction testbed designed in Unity with overlaid joint parent-child relationship.

4.3 Non-linearities

Heavy-duty manipulators exhibit non-idealities that materially affect closed-loop performance and safety. In the digital twin we embed the dominant joint-level effects to reduce the sim-to-real gap and enable controller tuning.

Backlash and hysteresis: Gear trains present directional deadbands and impact-like transients at reversals [13]. We use a backlash operator with a gap δ_b and re-engagement via a Newtonian restitution coefficient $\varepsilon \in [0, 1]$. Around a contact event, the post-impact velocities of link ($\dot{q}$) and motor ($\dot{q}_m$) are updated as

$$\dot{q}^+ = \frac{\dot{q}\,(J_\ell - \varepsilon J_m) + \dot{q}_m\, J_m(1+\varepsilon)}{J_\ell + J_m}, \qquad \dot{q}_m^+ = \frac{\dot{q}\, J_\ell(1+\varepsilon) + \dot{q}_m\,(J_m - \varepsilon J_\ell)}{J_\ell + J_m}, \tag{1}$$

with $\dot{q}^+, \dot{q}_m^+$ denoting post-impact velocities and J_ℓ, J_m the link/motor inertias. Within the backlash gap ($|q_m - q| < \delta_b$) we latch $\dot{q} = 0$ until re-engagement; (1) captures the observable "clunk" and energy loss on tooth re-contact.

Joint flexibility / structural compliance: Long booms are elastic; transmissions add torsional compliance [14].

A standard two-inertia spring-damper coupling (motor-link) reproduces load-dependent deflection and ringing:

$$\begin{aligned} \tau_m &= J_m \ddot{q}_m + B_m \dot{q}_m + K_m (q_m - q), \\ K_m (q_m - q) &= J_\ell \ddot{q} + B_\ell \dot{q} + \tau_{\text{ext}}, \end{aligned} \tag{2}$$

where K_m is the equivalent stiffness, B_m, B_ℓ viscous losses on the motor, and τ_{ext} aggregates gravity/contact effects.
Non-linear friction: To avoid bias and chatter at low speeds, we employ a composite law with a set-valued stiction region and smooth Stribeck decay [15]:

$$\tau_f(\dot{q}) = \begin{cases} \tau \in [-\tau_s, +\tau_s], \dot{q} = 0, \\ \left(\tau_c + (\tau_s - \tau_c) e^{-(|\dot{q}|/v_s)^\gamma}\right) \operatorname{sgn}(\dot{q}) \;+\; B\,\dot{q}, \dot{q} \neq 0, \end{cases} \tag{3}$$

with τ_s (stiction), τ_c (Coulomb), B (viscous), and (v_s, γ) shaping the Stribeck transition. A gate around $\dot{q} = 0$ reproduces stick-slip observed on sliding sections and bearings.
Actuator saturation and rate limits: The commanded effort is clamped to the physical envelope and, optionally, slew-rate limited to emulate hydraulic dynamics [16]:

$$\tau = \operatorname{sat}(\tau_{\text{cmd}}; -\tau_{\max}, \tau_{\max}) = \begin{cases} -\tau_{\max}, & \tau_{\text{cmd}} \le -\tau_{\max}, \\ \tau_{\text{cmd}}, & |\tau_{\text{cmd}}| < \tau_{\max}, \\ \tau_{\max}, & \tau_{\text{cmd}} \ge \tau_{\max}. \end{cases} \tag{4a}$$

$$\dot{\tau} = \operatorname{sat}(\dot{\tau}_{\text{cmd}}; -r_{\max}, r_{\max}). \tag{4b}$$

which aligns the simulation with the real machine and supports anti-windup logic in the controllers.

The numerical parameters in (1)-(4) were empirically calibrated through an iterative sim-to-real procedure. Starting from physically plausible values, we executed representative motion primitives in both the digital twin and the real telehandler under different controller gains and compared salient response features (e.g., overshoot, settling time, etc). The parameters were then progressively adjusted until the simulated responses reproduced the observed real-machine trends with consistent qualitative agreement. A formal parameter identification method will be addressed in future work through data-driven optimisation over logged telemetry.
Configuration- and load-dependent effects: Gravity and payload torques grow with boom elevation and extension, shifting equilibria and effectively amplifying frictional losses along prismatic sections. We include configuration-dependent bias and extension-scaled friction to emulate these couplings, aligning with observations from heavy-equipment studies where geometry and load profoundly shape closed-loop behaviour [14, 7].
With these non-linearities embedded and I/O parity to the machine established, we now detail the controller designs.

5 Control Architecture

The telehandler is controlled as a set of joint loops driven by a motion pipeline: high-level commands (waypoints or joint trajectories) are converted by MoveIt2 into joint-space references, which are then tracked by per-joint position controllers running at a fixed sampling rate f_s. Each controlled joint j receives a reference $q_j^*[n]$ and publishes a measured position $q_j[n]$; the loop error is $e_j[n] \triangleq q_j^*[n] - q_j[n]$. The closed-loop control problem decomposes into the **6** channels/joints introduced in the previous section, with one controller per channel. In revolute joints the controller regulates angular position; in prismatic joints it regulates linear displacement. Gains are first identified in the digital twin and then transferred *as-is* to the retrofitted machine for zero-shot validation.

5.1 MoveIt2 Framework

Controllers are implemented as a MoveIt2 ROS2 plugin that consumes `trajectory_msgs/JointTrajectory` and `sensor_msgs/JointState` and advances point-by-point within a given tolerance. For each waypoint, the node computes per-joint position errors $e_j[n]$, wrapping angles for revolute joints via the shortest angular distance and using direct differences for prismatic joints. A generic *controller core* then maps $\{e_j[n]\}$ to control commands $\{u_j[n]\}$. Around it, the stack applies non-linearities (Section 4.3) and collision checks. Joint feedback is read from `/telehandler/joint_states` and velocity commands are converted to the telehandler's low-level interface through the compact `/telehandler/joint_cmds` drive vector. Emergency stop, pause/resume, and cancel are handled consistently via a ROS2 action server. Replans, when triggered, replace the active trajectory seamlessly. This arrangement yields a reproducible joint-space follower in which controllers differ only in how $u_j[n]$ is computed from $e_j[n]$.

5.2 Baseline: PID Controller

We adopt a standard parallel PID as the reference controller, with a first-order filtered derivative and anti-windup around the actuator limits. In the Laplace domain the law reads

$$U(s) \;=\; \left(K_p \;+\; \frac{K_i}{s} \;+\; K_d\,\frac{s}{1+\tau_f s}\right) E(s), \tag{5}$$

where K_p, K_i, K_d are the proportional, integral and derivative gains, and $\tau_f > 0$ is the derivative filter time constant (a standard practice to mitigate noise amplification and high-frequency spillover in real actuators [16]). For real-time execution within the MoveIt2/ROS2 controller plugin, we use a simple, numerically robust discrete realisation with sampling period h and $e_n \triangleq e(nh)$:

$$\text{(Integrator)} \quad I_n = I_{n-1} + K_i \frac{h}{2} \left(e_n + e_{n-1}\right), \tag{6}$$

$$\text{(Filtered derivative)} \quad v_n = \alpha\, v_{n-1} + (1-\alpha)\, \frac{e_n - e_{n-1}}{h}, \quad \alpha = \frac{\tau_f}{\tau_f + h}, \tag{7}$$

$$\text{(Unsaturated output)} \quad u_n = K_p\, e_n + I_n + K_d\, v_n, \tag{8}$$

$$\text{(Saturation)} \quad u_n^{\text{sat}} = \text{sat}\big(u_n;\, u_{\min}, u_{\max}\big), \tag{9}$$

$$\text{(Back-calculation)} \quad I_n \leftarrow I_n + K_{aw}\, h\, \big(u_n^{\text{sat}} - u_n\big), \tag{10}$$

where e_n is the current tracking error, v_n is the filtered discrete derivative, $u_{\min}, u_{\max}$ are the joint-level limits, and $K_{aw} \geq 0$ is the anti-windup gain. The trapezoidal integral in (6) improves bias rejection without excessive phase lag; the first-order derivative filter (7) is the discrete counterpart of $s/(1+\tau_f s)$ in (5).

This means that we expose three per-joint free parameters $\{K_p,\, K_i,\, K_d\}$, yielding a 6×3 parameter state (18 parameters). The first-order derivative filter constant τ_f and the anti-windup gain K_{aw} are treated as global implementation settings to minimise the tuning burden. This baseline is implemented as a MoveIt2 controller and interfaces the telehandler through `sensor_msgs/JointState`, driving joint commands at $f_s = 1/h$. At this stage, the gains were obtained through a staged manual tuning in the digital twin. Starting from conservative gains, K_p was first increased to obtain a responsive closed-loop behaviour without excessive oscillation; K_d was then adjusted to reduce overshoot; finally, K_i was introduced gradually to compensate steady-state offsets. This process was repeated across representative motion primitives and operating configurations until a stable compromise was reached between rise time, overshoot, settling, and residual oscillation.

5.3 Fractional-Order PID Controller

To improve robustness against the non-linearities of heavy-duty actuation previously introduced, we adopt a fractional-order PID ($\text{PI}^{\lambda}\text{D}^{\mu}$), which enables non-integer integration and differentiation orders $\lambda, \mu \in (0, 1]$ [17, 18]. Fractional controllers have shown favourable phase margins and disturbance rejection under model uncertainty [5, 19]. In our design, we further include a light predictive term that extrapolates the near-future error to attenuate small effective delays in the loop (e.g., sensing/communication latency), yielding an FOPID-P law:

$$u(t) = K_p e(t) + K_i I_t^{\lambda}\{e\}(t) + K_d D_t^{\mu}\{e\}(t) + K_{\text{pred}}\big(\hat{e}(t+\Delta) - e(t)\big), \tag{11}$$

where I_t^{λ} and D_t^{μ} denote the fractional integral and derivative of orders λ and μ, $\hat{e}(t+\Delta)$ is a short-horizon prediction of the error, and $K_p, K_i, K_d, K_{\text{pred}} \geq 0$ are gains. For real-time execution inside MoveIt2/ROS2, we use a Grünwald-Letnikov (GL) discrete approximation, which realises fractional operators as finite-memory convolutions with binomial weights [20], leading to

$$I_t^{\lambda}\{e\}(t_n) \approx h^{\lambda} \sum_{k=0}^{N} w_k^{(\lambda)} e_{n-k}, \qquad D_t^{\mu}\{e\}(t_n) \approx h^{-\mu} \sum_{k=0}^{N} (-1)^k w_k^{(\mu)} e_{n-k}, \tag{12}$$

with $e_n = e(t_n)$, sampling period h, $t_n = nh$, truncation (memory length) N, and recursively computed weights $w_0^{(\alpha)} = 1$, $w_k^{(\alpha)} = w_{k-1}^{(\alpha)} \frac{\alpha-(k-1)}{k}$ for $\alpha \in \{\lambda, \mu\}$, which avoids repeated Gamma-function evaluations and is numerically stable for online use [20]. The predictive term uses a one-step linear extrapolation,

$$\hat{e}_n = e_n + \Delta\, \frac{e_n - e_{n-1}}{h}, \tag{13}$$

with small horizon Δ to remain conservative. The discrete control update becomes

$$u_n = K_p e_n + K_i h^{\lambda} \sum_{k=0}^{N} w_k^{(\lambda)} e_{n-k} + K_d h^{-\mu} \sum_{k=0}^{N} (-1)^k w_k^{(\mu)} e_{n-k} + K_{\text{pred}}(\hat{e}_n - e_n). \tag{14}$$

It is noteworthy that the additional degrees of freedom λ and μ shape memory and phase lag, i.e., choosing $\lambda < 1$ mitigates integral wind-up and improves low-frequency robustness; $\mu < 1$ filters high-frequency noise [18, 19]. We adopt a simulation-guided tuning: (i) start from the classical PID ($\lambda = \mu = 1$) introduced before to obtain a baseline; (ii) reduce λ, μ in $[0.6, 1.0]$ while re-balancing K_i, K_d to meet time-domain targets; (iii) introduce a small predictive gain K_{pred} with $\Delta \ll h$ to compensate mild delays. Anti-windup and actuator saturation are enforced around equation (14). Considering the presented control architecture, this means that, for the fractional-order controller each joint carries $\{K_p,\, K_i,\, K_d,\, \lambda,\, \mu,\, K_{\text{pred}}\}$, i.e., a 6×6 parameter state (36 parameters), while the prediction horizon Δ, discretisation step h, GL memory length N, and any safety clamping are shared.

We next quantify, in the twin and on hardware, how the baseline PID and the proposed FOPID-P behave under a series of experiments.

6 Experimental Results

We evaluate the two joint-space controllers, baseline PID and the proposed FOPID-P, first in the Unity-ROS2 digital twin and then on the real retrofitted telehandler without re-tuning. The study is organised to isolate effects and quantify transferability.

6.1 Setup

We harmonise simulation and field protocols to enable like-for-like comparisons. Controller parameters are obtained by trial-and-error in simulation as depicted in the Table 1. Data logs contain time-stamped joint states, controller internals (error, control effort), and event flags (saturation/stall), from which the metrics are computed offline.

Table 1. Controller tuning parameters

Controller	K_p	K_i	K_d	λ	μ
PID	1.0	3.0	0.0737	–	–
FOPID-P	5.98	4.38	0.088	1.03	1.47

6.2 Step and Frequency Response

To characterise joint behaviour and controller robustness, each controlled joint is excited with bounded, symmetric position steps sized to a moderate fraction of its travel, with randomised initial configurations to vary gravity/payload bias. For each step, we record the response under PID and FOPID-P. Figure 3 shows the step response of the two controllers, where it can be seen that the FOPID-P reaches the desired set point faster.

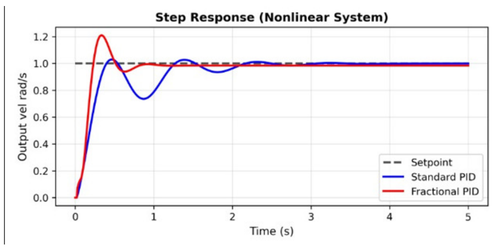

Figure 3. Step response.

To assess the robustness of the controllers in the face of disturbances, an impulse disturbance was simulated and introduced into the system. Figure 4 shows the behaviour of both controllers, with FOPID-P standing out as it has a lower stabilisation time due to its higher damping component than PID, which exhibits more oscillations.

Figure 5 shows the behaviour of the controllers when following a given trajectory. Due to the smooth response of the FOPID-P, it once again stands out when compared

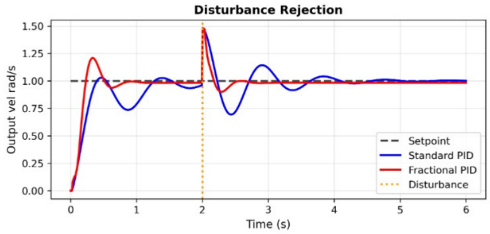

Figure 4. Controller behaviour under disturbance.

to the baseline PID. This is complemented with a quantitative analysis to assess the robustness of each controller, as depicted in Table 2. The table shows that FOPID-P outperforms the PID in all metrics.

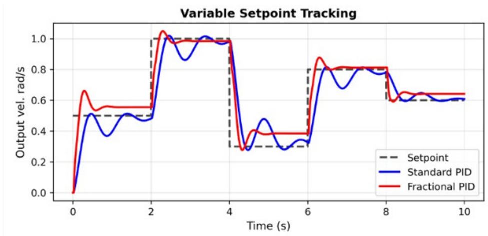

Figure 5. Response to a given trajectory.

Table 2. Non-linear system results.

Test	PID	FOPID-P	Imp. (%)
Step response	0.795	0.573	27.9
Disturbance rejection	0.945	0.679	28.1
Variable set point	1.000	0.763	23.7

6.3 Zero-shot Real-World Assessment

Finally, we deploy both controllers on the real telehandler, using the same parameters obtained in simulation. The platform executes a construction-relevant motion sequence: picking a sandwich panel from the pile, transporting it towards the façade, and returning it to the pile. Figure 6 reports the MoveIt2 reference along with the measured joint responses for both controllers. It is noteworthy that there was no need to control the `basket_crane_link_to_basket_link` (therefore omitted from the plot), since the winch joint provides sufficient distal flexibility to complete the task while respecting the imposed kinematic and safety constraints.

The results show that FOPID-P achieves tighter tracking and a more prompt transient than the baseline PID, which exhibits a pronounced lag and residual error during phases with stronger coupling. This becomes particularly evident after the payload is engaged (from the second vertical grey line around the 300 s mark), where the PID tracking slows

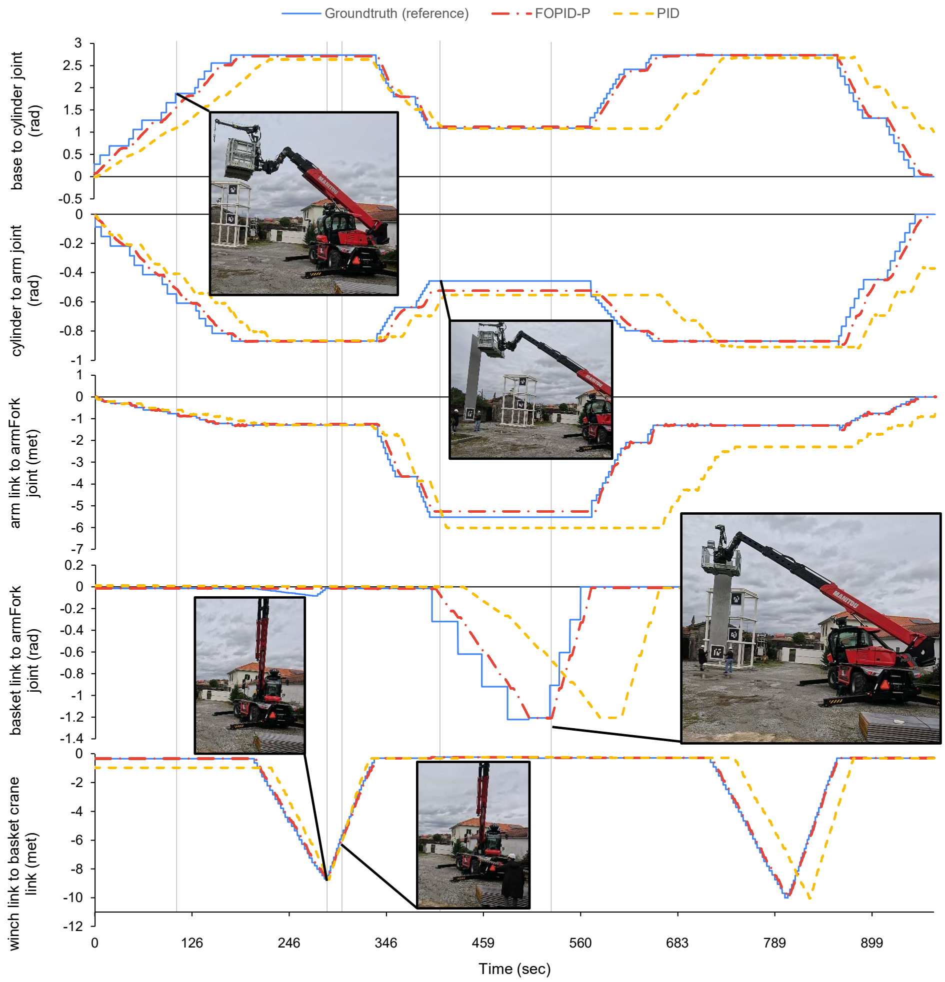

Figure 6. Joint-space response to a full pick-and-place assembly trajectory executed by the real robotic telehandler.

down and deviates more from the reference. To complement the qualitative comparison in Fig. 6, we computed the normalised root mean squared error (NRMSE) and the normalised mean absolute error (NMAE) from the real-world trajectories. FOPID-P reduces the mean NRMSE from 22.0% to 4.3% and the mean NMAE from 13.6% to 2.5%, corresponding to improvements of 80.5% and 81.9%, respectively. At joint level, the NRMSE reduction ranges from 67.1% to 90.2%, indicating that the tighter tracking achieved by FOPID-P is not limited to a single axis. Since these results are based on one representative execution per controller, they should be interpreted as descriptive case-study evidence rather than as a statistical comparison, which is foreseen as future work.

7 Conclusions and Future Work

This paper presented a motion-control workflow for a retrofitted rotating telehandler targeting sandwich panel handling in construction. A ROS2-Unity digital twin was developed and enriched with joint-level non-linearities to reduce sim-to-real mismatch and de-risk commissioning. On this foundation, we implemented and evaluated two MoveIt2-integrated joint-space controllers: a baseline PID and a fractional-order PID with a light predictive term (FOPID-P). Both simulated an real-world experiments depicted the superiority of the FOPID-P, particularly after payload engagement. These findings indicate that high-fidelity digital twins with explicit joint-level effects can make controller tuning more repeatable and transferable for construction robots. Future work will automate the tuning process and expand the evaluation to longer assembly routines.

Acknowledgments

This document is issued within the frame and for the purpose of the CRIARTE project. This project is co-funded by the programme Portugal 2030, under the reference COMPETE2030-FEDER-01192200.

References

[1] Ming Zhang, Rui Xu, Haitao Wu, Jia Pan, and Xiaowei Luo. Human–robot collaboration for on-site construction. *Automation in Construction*, 150: 104812, 2023.

[2] Micael S Couceiro, Beril Yalcinkaya, Carlos Pizzino, and Rui B Garcia. Establishing an on-site construction pilot for collaboration between humans and heavy-duty robots. In *Proceedings of the 3rd Future of Construction Workshop at the International Conference on Robotics and Automation (ICRA 2024), Yokohama, Japan*, pages 13–17, 2024.

[3] David Portugal, Maria Eduarda Andrada, André G Araújo, Micael S Couceiro, and João Filipe Ferreira. Ros integration of an instrumented bobcat t190 for the semfire project. In *Robot Operating System (ROS) The Complete Reference (Volume 6)*, pages 87–119. Springer, 2021.

[4] Igor Podlubny. Fractional-order systems and pi/sup/spl lambda//d/sup/spl mu//-controllers. *IEEE Transactions on automatic control*, 44(1):208–214, 2002.

[5] NM Fonseca Ferreira and JA Tenreiro Machado. Fractional-order hybrid control of robotic manipulators. In *Proceedings of the 11th International Conference on Advanced Robotics*, volume 398. Piscataway, Japan: IEEE Press, 2003.

[6] Stefana Parascho. Construction robotics: From automation to collaboration. *Annual Review of Control, Robotics, and Autonomous Systems*, 6(1):183–204, 2023.

[7] Weiwei Liu, Xuan Luo, Jingwen Zhang, Dongdong Niu, Jianyang Deng, Weiqi Sun, and Jie Kang. Review on control systems and control strategies for excavators. In *Journal of Physics: Conference Series*, volume 2301, page 012023. IOP Publishing, 2022.

[8] Asmat Ullah Khan, Lizhen Huang, Erling Onstein, and Yongping Liu. Overview of emerging technologies for improving the performance of heavy-duty construction machines. *Ieee Access*, 10:103315–103336, 2022.

[9] Abhishek Kadian, Joanne Truong, Aaron Gokaslan, Alexander Clegg, Erik Wijmans, Stefan Lee, Manolis Savva, Sonia Chernova, and Dhruv Batra. Sim2real predictivity: Does evaluation in simulation predict real-world performance? *IEEE Robotics and Automation Letters*, 5(4):6670–6677, 2020.

[10] Xi Wang, Ci-Jyun Liang, Carol Menassa, and Vineet Kamat. Real-time process-level digital twin for collaborative human-robot construction work. In *Proceedings of the International Symposium on Automation and Robotics in Construction (IAARC)*, 2020.

[11] Jonathan Platt and Kenneth Ricks. Comparative analysis of ros-unity3d and ros-gazebo for mobile ground robot simulation. *Journal of Intelligent & Robotic Systems*, 106(4):80, 2022.

[12] Jordan Allspaw, Gregory LeMasurier, and Holly Yanco. Comparing performance between different implementations of ros for unity. 2023.

[13] Mattias Nordin and Per-Olof Gutman. Controlling mechanical systems with backlash—a survey. *Automatica*, 38(10):1633–1649, 2002.

[14] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. *Robotics: modelling, planning and control*. Springer, 2009.

[15] Brian Armstrong-Hélouvry, Pierre Dupont, and Carlos Canudas De Wit. A survey of models, analysis tools and compensation methods for the control of machines with friction. *Automatica*, 30(7):1083–1138, 1994.

[16] Karl Johan Åström and Tore Hägglund. *Advanced PID control*. ISA-The Instrumentation, Systems and Automation Society, 2006.

[17] I Podlubny, L Dorcak, and I Kostial. On fractional derivatives, fractional-order dynamic system and pid-controllers. In *Proc. of the 36th 1997 IEEE Conference on Decision and control*, pages 4985–5990, 1999.

[18] Concepción A Monje, YangQuan Chen, Blas M Vinagre, Dingyü Xue, and Vicente Feliu. *Fractional-order systems and controls: fundamentals and applications*. Springer, 2010.

[19] YangQuan Chen, Ivo Petras, and Dingyu Xue. Fractional order control-a tutorial. In *2009 American control conference*, pages 1397–1411. IEEE, 2009.

[20] BM Vinagre, I Podlubny, A Hernandez, and V Feliu. Some approximations of fractional order operators used in control theory and applications. *Fractional calculus and applied analysis*, 3(3):231–248, 2000.