2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Reinforcement Learning Control for Autonomous Hydraulic Material Handling Machines with Underactuated Tools

Filippo A. Spinelli[1], Pascal Egli[1], Julian Nubert[1,2], Fang Nan[1], Thilo Bleumer[3], Patrick Goegler[3], Stephan Brockes[3], Ferdinand Hofmann[3], and Marco Hutter[1]

***Abstract*— The precise and safe control of heavy material handling machines presents numerous challenges due to the hard-to-model hydraulically actuated joints and the need for collision-free trajectory planning with a free-swinging end-effector tool. In this work, we propose an RL-based controller that commands the cabin joint and the arm simultaneously. It is trained in a simulation combining data-driven modeling techniques with first-principles modeling. On the one hand, we employ a neural network model to capture the highly nonlinear dynamics of the upper carriage turn hydraulic motor, incorporating explicit pressure prediction to handle delays better. On the other hand, we model the arm as velocity-controllable and the free-swinging end-effector tool as a damped pendulum using first principles. This combined model enhances our simulation environment, enabling the training of RL controllers that can be directly transferred to the real machine. Designed to reach steady-state Cartesian targets, the RL controller learns to leverage the hydraulic dynamics to improve accuracy, maintain high speeds, and minimize end-effector tool oscillations. Our controller, tested on a mid-size prototype material handler, is more accurate than an inexperienced operator and causes fewer tool oscillations. It demonstrates competitive performance even compared to an experienced professional driver.**

I. Introduction

Material handlers similar to the one in Fig. 1 find applications in diverse settings, including construction sites, recycling centers, ports, and warehouses. They are indispensable for efficiently maneuvering and sorting heavy materials such as scrap metal, bulk cargo, logs, and construction debris. Their most notable feature is the free-swinging end-effector tool. Compared to fixed attachments, it offers reduced manufacturing costs and operational advantages: gravity alignment facilitates grabbing piled material, and the swinging can be exploited to enlarge the reachable task space. While beneficial for specific tasks, this joint setup, combined with the hydraulic actuation, makes maneuvering extremely complex, even for trained operators. The hydraulic cabin-rotation motor is often characterized by extensive delays and binary braking dynamics, making accurate motion control, particularly stopping, challenging. Furthermore, if the tool oscillations are not adequately damped, they can cause severe damage.

*This work is supported in part by the NCCR digital fabrication and robotics, the Liebherr-Hydraulikbagger GmbH, and the Max Planck ETH Center for Learning Systems.

[1]The authors are with the Robotic Systems Lab, ETH Zürich, Zürich, Switzerland.

[2]The author is with the MPI for Intelligent Systems, Stuttgart, Germany.

[3]The authors are with the Liebherr-Hydraulikbagger GmbH, Kirchdorf an der Iller, Germany.

Corresponding author: Filippo A. Spinelli, fspinelli@ethz.ch

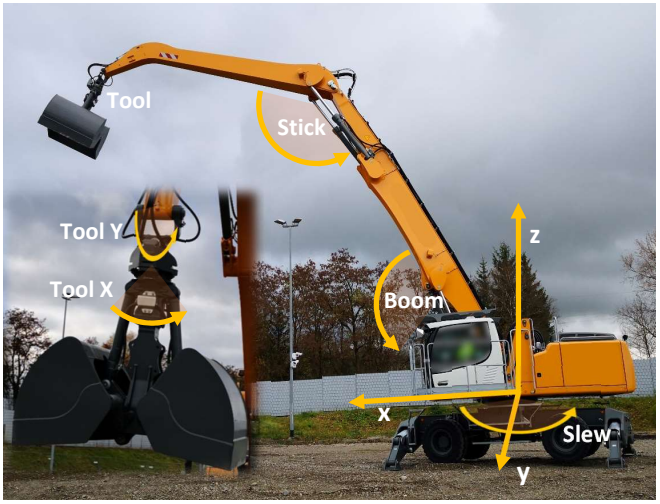

Fig. 1. The prototype material handler used in this work has an operational range of about 20 m and weighs more than 40 t. A 1.5 t grabbing shovel designed for loose material was employed, with a maximum load of 2 t.

Construction has become a focal point for robotic research in recent years [1], encompassing areas such as force control [2], [3], full arm motion control [4], [5], motion planning [6], [7], and state estimation [8]. Autonomous high-level tasks have been demonstrated, including earth-moving planning for bulldozers [9] and excavators [10], [11], or rock wall construction [12]. However, past research has rarely addressed the fast and efficient handling of material or the automation of large material handlers as the one we focus on, despite the long-recognized importance of material handling tasks in the industry [13]. Automating these machines would enable continuous operation and limit the need for human drivers in harsh conditions, thereby improving both efficiency and safety.

In this work, we propose a solution for autonomous large-scale material manipulation, filling the gap in the literature of control strategies for material handlers. Most state-of-the-art methods do not work reliably on such machines due to the complex actuators' velocity profile and the lack of specialized high-bandwidth servo valves. We present a Reinforcement Learning (RL) control scheme for material handling machines, learning to reach 3D Cartesian targets with high speeds while minimizing oscillations to facilitate safe material grasping. The proposed approach combines data-driven modeling for the highly non-linear and delayed cabin turn joint with modeling from first principles for the arm and the free-swinging end-effector tool. The evaluated

controller performs at a speed and accuracy comparable to those of an average operator and works under any load.

A. Related Work

1) Control of Hydraulic Machines: A variety of techniques for the control of hydraulic machines have been proposed. Traditional model-based control approaches [14], [15] rely the most on an accurate, often even analytic, model to handle delays and non-linearities, which is hard to obtain in practice. Research has shown that retrofitted high-performance hydraulic valves enable accurate force control [2]. Their practical use, however, is restricted by their high price and limited oil flow, often only allowing for slow arm motions. As a response, recent research has focused on incorporating machine learning techniques into the control loop. Nurmi et al. [16] work at the intersection of Machine Learning (ML) and model-based control, proposing a deep learning-based method to identify nonlinear velocity Feed Forward (FF) curves for pressure-compensated hydraulic valves. FF-based velocity control is also adopted in our work to control the two arm joints, even though without these learning improvements. Park et al. [17] present an online learning framework for position control of hydraulic excavators using echo-state networks. From time series of input and output data, they train an inverse model of the plant, which is later used to generate the control commands given a new trajectory reference. Online learning is, however, dangerous for heavy machinery tasks, so Lee et al. [5] propose a similar model inversion control approach working offline. In particular, they decompose the model into three physics-inspired components to learn the excavator dynamics more efficiently, considering force and pressure measurements. Recently, RL has emerged as an alternative to control hydraulic excavation machines. Egli and Hutter [18], [4] introduce a data-driven modeling approach of the coupled hydraulic cylinders. The excavator is controlled by training an RL agent in simulation, aiming for end-effector position- or velocity-tracking in free space and with weak ground contact. Compared to previous classical approaches, the learned controller handles non-linear dynamics and delays better and is more robust to disturbances. In this work, we build on top of [4] but split the modeling into two parts: *i)* the data-driven analogy is used for the cabin turn joint, with pressure and inertia as additional features, *ii)* while first principles modeling is adopted for the others. Dhakate et al. [19] propose a similar pipeline for a different machine. Their work captures the mapping between cylinders' displacements and joint variables of a small forest forwarder crane through a Neural Network (NN) model. They then train an RL position controller, which commands joint setpoints, simply treating the unactuated tool joint as a disturbance. In contrast, we explicitly model the unactuated tool in our simulation and aim for an active damping behavior through suitably chosen actions of all the controllable joints.

2) Control in the Presence of Passive Joints: Previous research in the construction domain has rarely addressed the free-swinging end-effector tool. Promising results on the safe control of tower cranes [20], [21] have been achieved, but they a have simpler structure and more restricted tasks. RL has been used on these machines to improve control performance under payload variation [22]. Andersson et al. [23] use RL to control a simulated forestry crane for log grasping. The arm and grapple kinematics resemble those of material handlers, and the agent learns to take advantage of the oscillations to complete the task. However, the motors are assumed to track velocity references reliably on every joint, and the work is not validated on physical hardware. Oktay and Sultan [24] explore the helicopter slung-load system, modeled using first principles. Simulation results show the dependence of the model-based controller on the exact dynamic parameters to operate reliably. Further studies have focused on trajectory optimization [25] and RL [26], addressing tracking with the suspended load by enabling the aerial system to exploit inertial forces for motion generation. In the robotic manipulation domain, Zimmermann et al. [27] developed a computational framework for the robotic animation of string puppets. These are coupled pendulum systems, sharing similar dynamics with a free-swinging grab. Ichnowski et al. [28] work on inertial transport for pick-and-place operations with robotic arms. Their approach is based on iterative convex optimization with end-effector acceleration constraints. Fictitious forces are included in the model to consider the inertial load during planning. Our work applies similar modeling techniques to build the training environment but solves the control problem via RL.

B. Contributions

We present the following contributions:

- A data collection routine and an NN-architecture to model hydraulic motors, accurately predicting delay effects by leveraging velocity and pressure evolutions.
- A combined modeling approach, partly consisting of a data-driven NN-model for the slew, partly of first-principle modeling for the arm and the unactuated tool.
- An RL agent entirely trained in simulation using the previously mentioned mixed model, actuating the slew motor and the two arm joints simultaneously. The controller tracks steady-state Cartesian position targets on the real machine, effectively reducing the tool oscillations while maintaining a high operational speed.
- Experimental validation on a prototype 40 t material handler, including comparisons with human operators of varying experience levels.

II. System Description

Our algorithm was validated on real hardware (Fig. 1). This research machine has three hydraulically actuated joints with one Degree of Freedom (DoF) each. They are independently controlled via steer-by-wire joystick commands and are subject to delays, dead zones, and non-linearities. Linear hydraulic cylinders actuate the boom and stick joints, while the slew joint uses a hydraulic motor to rotate freely. This motor presents a binary braking system to slow down rotations more aggressively, but its contribution cannot be actively regulated. The relationship between pressure and

velocity is highly intertwined, and the configuration's inertia plays a crucial role in shaping the speed curve. Such actuation complexity renders the previously deployed model-based control algorithms unreliable, resulting in significant overshoots and oscillations. Consequently, we use an ML approach for both the slew modeling and control. The tool consists of a chain of two unactuated revolute joints (pitch & roll) and one actuated joint for rotations around the vertical axis (yaw). Since our controller solely manages gripper positioning, we disregard the last joint as well as the clamshell opening, reducing the overall kinematics of the system to five DoFs.

A. Feedback

The machine used, specifically developed for autonomous purposes, is retrofitted at 50Hz with:

- Encoders on slew, boom, and stick joints, providing position and velocity measurements. As the velocity is derived from position data, it suffers from a delay of approximately 0.2 s.
- Pressure sensors on slew, boom, and stick, measuring fluid pressure on both sides of the piston.
- Inertial Measurement Units (IMUs) on cabin and tool, each supplying 3D angular velocities, 3D linear accelerations, and 2D angular orientation.
- An algorithm estimating the material load in the tool using least-squares regression on pressure data.

B. Arm Velocity Controller

A model-based velocity controller for the arm joints has been formulated, incorporating a FF component and PI feedback compensation as discussed in [29]. The FF model is based on a 25-point Look-Up Table (LUT) per joint and maps desired joint velocities to control commands. Such an approach can be deployed for simple isolated hydraulic cylinder control. While this kind of direct control cannot be applied on the slew hydraulic motor, it is tuned well enough for boom and stick simple hydraulic cylinders, allowing for a convenient decoupling of arm motion planning and low-level cylinder control as in [30].

III. Proposed Approach

Our research centers around the following aspects: *i)* utilizing ML to model hydraulic motor dynamics, *ii)* constructing a simulation environment with a reduced sim-to-real gap, and *iii)* training an RL controller to track task-space targets with the arm while accommodating hydraulic dynamics and tool oscillations.

A. Slew Actuator Model

We use ML to capture the turning dynamics as a first step toward RL-based slew control.

1) Data Collection: We aimed to excite the slew motor's main modes and explore the relevant state space. Data was collected in various ways according to Table I. The majority has been generated by applying artificial excitation signals consisting of regular periodic references. For each run, the arm configuration was randomized and kept static. We further collected data during manual random movements and real driving situations. For the final model, we included the deployment of an earlier version of the slew controller, trained only on the first two data sources. By recording a driver during operational cycles and our controller running in a closed loop, the model became more accurate at predicting the state evolution of typical tasks. We then built a unique dataset, as fine-tuning and transfer learning [31] do not benefit our simple architecture.

TABLE I
Characteristics of the Training Set.

%	Mins	Description
70	56	Periodic references: step, sinusoidal, trapezoidal, with static arm. Random references with arm motion.
9	7	Real driving conditions during common operations.
21	16	Closed-loop slew controller, with static arm.
100	79	Total data collected.

2) Data Augmentation: From experiments and mechanical analysis, we concluded that the slew joint has symmetric rotation dynamics. Leveraging this, we augmented the dataset via mirroring, which doubles the amount of data and improves the model accuracy due to the attenuation of the recorded noise effects.

3) Neural Network Model: We use a Multilayer Perceptron (MLP) to capture the slew dynamics. The framework operates on a vectorized input history of states and control commands and predicts the pressure and velocity evolution one step into the future. Unlike previous work [4], we explicitly include measured pressures in our formulation to better deal with the large delays from command input to velocity response. In particular, we learn both pressure (Eq. (1)) and speed (Eq. (2)) dynamics via two different MLPs, with trainable weights θ_p and θ_ω. The prediction and measurement rates are 0.1 s. Both use the ReLU activation function inspired by [16], with the loss defined as the single-step prediction error. Other hyperparameters are summarized in Table II. We assume a deterministic mapping between the past inputs and the next pressure values and model it as:

$$\begin{aligned}\left[p_{l[k]}, p_{r[k]}\right] = \mathcal{F}_p\Big(&u_{[k-9,k]}, p_{l[k-10,k-1]}, p_{r[k-10,k-1]},\\ &\omega_{[k-10,k-1]}, I_{z,[k-1]}; \theta_p\Big),\end{aligned} \tag{1}$$

where $\mathcal{F}_p$ denotes the neural network with trainable parameters θ_p, u denotes the control input, p_l and p_r represent the pressures of the left and right chambers, ω is the angular velocity of the cabin and I_z the configuration-dependent inertia around the z-axis. The notation $\cdot[i, j]$ denotes a discrete time series from time step i to time step j of the given quantity. Using our double architecture, 1 s history access

TABLE II
Neural Network Design Hyperparameters.

Hyperparam	Pressure	Velocity
Input dim	41	41
Output dim	2	1
Layers	$[128, 128, 128, 128, 32]$	$[128, 128, 128, 32]$

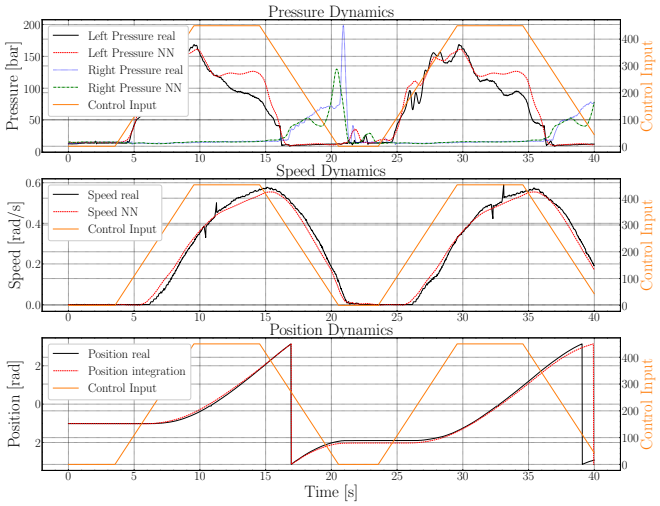

Fig. 2. Open-loop prediction using the NN model for a 40-second trapezoidal reference. This shape approximates a control profile while maintaining regularity to mitigate noise effects.

is enough for generalizable learning. Similarly, the second neural network models the velocity evolution as follows:

$$\left[\omega_{[k]}\right] = \mathcal{F}_\omega\Big(u_{[k-9,k]}, p_{l[k-9,k]}, p_{r[k-9,k]}, \omega_{[k-10,k-1]}, I_{z,[k-1]}; \theta_\omega\Big). \tag{2}$$

Note that Eq. (2) takes the output of Eq. (1) as an input. Finally, the position at step k is computed via integration. All buffers are initialized with zeros. The configuration-dependent inertia I_z is computed from the arm position and the nominal link weights, approximating the tool as a point mass. This simple approach is sufficient to capture the effects of the arm extension on the acceleration. We report in Table III the Mean Absolute Error (MAE) obtained with different input features but the same history access. Notably, inertia plays the most critical role, and the additional pressure inputs are useful to better learn the velocity dynamics.

Fig. 2 shows the qualitative open-loop performance of our model on a 40 s-long test trajectory. The integrated position is subject to drift over time. Still, we must ensure accuracy only for a training episode, during which the RL agent achieves its goal. Based on real-world manipulation tasks, we identify 10 s as a good trade-off between sufficiently long and precise.

B. Simulation Environment

A mixed environment was developed to train RL controllers. It simulates the full machine *i)* using the actuator model for the slew joint dynamics and *ii)* using an analytic model derived from first principles to simulate the arm and the tool dynamics. While the first is machine-specific, the mathematical models can be used more generally and adapted with parameter tuning. For the boom and stick joints, we use a first-order system with delay to approximate the velocity tracking performance of the arm controller:

$$\dot{q}[k] = \dot{q}[k-1] + P\big(\hat{\dot{q}}[k-d] - \dot{q}[k-1]\big), \tag{3}$$

where $\hat{\dot{q}}$ is the velocity reference, d represents the delay (larger than the one produced by the encoders), and P specifies a hand-tuned time constant. This equation produces simulated trajectories as in Fig. 3.

TABLE III
ACTUATOR MODEL FEATURE CRAFTING.
METRICS ARE COMPUTED OVER 10s OPEN-LOOP PREDICTIONS.

Features	**Velocity MAE** [°/s]	**Position MAE** [°]
Proposed	**1.616**	**5.707**
No Pressure	1.805	6.509
No Inertia	2.080	7.517

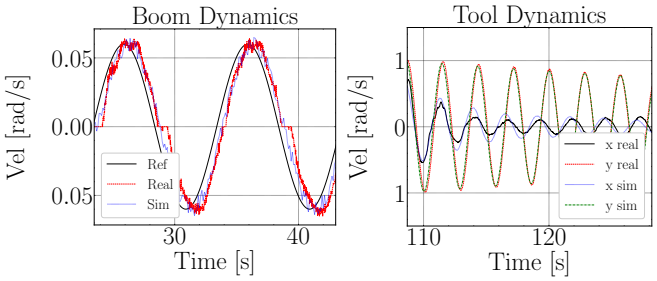

Fig. 3. Arm controller dynamics modeled as first-order systems with delay (left), and tool dynamics modeled via Lagrange and dissipation (right).

The tool is modeled using the Lagrange principle as a pendulum with decoupled x and y rotations. This makes state propagation via Forward Euler integration ($\Delta t = 0.02$ s) easier and more stable. The moving reference frame produces fictitious forces; we account only for the Euler and the centrifugal ones, as the Coriolis force would introduce a coupling between the axes. Furthermore, we include a dissipative term with the Rayleigh's dissipation function [32]. These choices lead to a system response as illustrated in Fig. 3. The following equations describe the mathematical formulation:

$$\dot{\theta}_{y[k+1]} = \Bigg(\Bigg(\frac{v_{x[k+1]} - v_{x[k]}}{\Delta t}\cos\theta_{y[k]} - \underbrace{g\sin\theta_{y[k]}}_{F_g} - \underbrace{\dot{\theta}^2_{slew[k]} r_y}_{F_{slew}}\Bigg)/l_y - \underbrace{b_{fy}\dot{\theta}_{y[k]}}_{\text{dissipation}}\Bigg)\Delta t + \dot{\theta}_{y[k]}, \tag{4}$$

$$\dot{\theta}_{x[k+1]} = \Bigg(\Bigg(-\frac{v_{y[k+1]} - v_{y[k]}}{\Delta t}\cos\theta_{x[k]} - \Big(\underbrace{g\cos\theta_{y[k]}}_{F_g} + \underbrace{\dot{\theta}^2_{y[k]} l_y}_{F_{roty}}\Big)\sin\theta_{x[k]}\Bigg)/l_x - \underbrace{b_{fx}\dot{\theta}_{x[k]}}_{\text{dissipation}}\Bigg)\Delta t + \dot{\theta}_{x[k]}. \tag{5}$$

Here, $\theta_{x,y}$ denote the angles between the two unactuated joints and the main frame axes, $v_{x,y}$ the linear velocities of the tool attachment points, $l_{x,y}$ the corresponding tool lengths, r_y the distance between the tool and the slew rotation axis, g the gravity constant, and $b_{fx,y}$ the dissipative coefficients. An illustration is shown in Fig. 4.

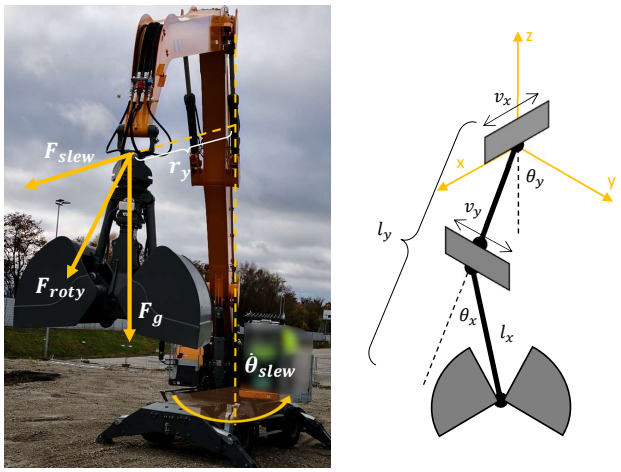

Fig. 4. The tool is modeled as a double pendulum with linearly oscillating support. In the left figure, we show the forces accounted for. The adopted approximations for each DoF are shown on the right.

C. RL End-Effector Controller

Inspired by the prior success of learning-based control of hydraulic excavators [18], [4], we tackled the end-effector task-space control by training a high-level RL policy that *i)* deals with the nonlinear machine dynamics (mostly stemming from the hydraulic slew motor), and *ii)* actively stabilizes the tool oscillations. This trained policy directly outputs joystick commands for the slew joint and speed references for boom and stick low-level velocity controllers.

1) Control Formulation: The controller is represented by an MLP with trainable weights θ_c as follows:

$$\begin{aligned}\mathbf{u}_{[k]} &= \left[u_{\text{slew}[k]}, \hat{\dot{q}}_{\text{boom}[k]}, \hat{\dot{q}}_{\text{stick}[k]}\right]^\top \\ &= \pi_{\theta_c}\Big(\mathbf{u}_{[:]}, \dot{\mathbf{q}}_{[:]}, \mathbf{q}_{[:]}, I_{z,[k-1]}, \hat{x}, \hat{y}, \hat{z}\Big),\end{aligned} \tag{6}$$

where the buffer is $[:] \coloneqq [k-H, k-1]$, and

$$\begin{aligned}\mathbf{q} &= \left[q_{\text{slew}}, q_{\text{boom}}, q_{\text{stick}}, q_{\text{tool},x}, q_{\text{tool},y}\right]^\top, \\ \dot{\mathbf{q}} &= \left[\dot{q}_{\text{slew}}, \dot{q}_{\text{boom}}, \dot{q}_{\text{stick}}, \dot{q}_{\text{tool},x}, \dot{q}_{\text{tool},y}\right]^\top,\end{aligned}$$

denote the measured angular position and velocity of all joints. Given a history of actions $\mathbf{u}$ and states $[\mathbf{q}, \dot{\mathbf{q}}]$, the inertia I_z, and the task-space target $[\hat{x}, \hat{y}, \hat{z}]$, the RL controller produces u_{slew} (directly applied to the machine), $\hat{\dot{q}}_{boom}$ and $\hat{\dot{q}}_{stick}$ (arm velocity references). During training, the Cartesian target is constant throughout an entire 10 s episode. The joystick input is limited to feasible values, and speed references are clipped within $[-0.2, 0.2]$ rad/s to match the LUT steady-state velocity assumption. The history length H is a trade-off between learning to exploit the full range of dynamics and limiting the mismatch of simulation and reality, which increases with H.

2) Policy Gradient RL: We train the agent using model-free policy gradient RL. Specifically, we employ the PPO [33] learning scheme with 50 Hz simulation and 10 Hz control rate. Our policy and value function networks have dimensions of $[256, 128, 128]$, using $\tanh$ activation and a linear output layer.

3) Domain Randomization: We use domain randomization to bridge the sim-to-real gap [34], [35]. At each step, uniform noise is added to the observations. At each environment initialization, we randomly sample the initial joint positions (avoiding collision configurations), the load, the arm controller and tool model parameters, and the target. Parameter randomization is particularly important for control robustness during deployment, and helps alleviate the simulation inaccuracies. The tool starts vertically, with zero initial velocity. Based on the assumption that the slew joint behavior is independent of position, the Cartesian target is randomly generated only in the x-z plane, with $\hat{y} = 0$ and $\hat{\theta}_{\text{slew}} = 0$. This facilitates learning by reducing the number of active observations. When deploying for different targets, the slew position feedback is then converted to an error.

4) Termination Conditions: Besides the low-level velocity controller enforcing boom and stick position limits, we also include termination conditions during training to avoid any safety hazards, i.e., if the gripper gets too low or too close to the cabin. This way, self-collisions and collisions with flat ground are avoided precautionary. All episodes last 10 s, unless the mentioned termination occurs.

5) Reward: Our proposed reward at timestep k consists of seven terms:

$$\begin{aligned}R_k = r_k^{\text{balance}} + r_k^{\text{target}} + r_k^{\text{action}} \\ + r_k^{\text{overshoot}} + r_k^{\text{oscillation}} + r_k^{\text{decouple}} + r_k^{\text{one-shot}}.\end{aligned} \tag{7}$$

These are defined as follows:

$$\begin{aligned}r_k^{\text{bal.}} &\propto \Big(\exp\big(-\|\tilde{\varepsilon}_k\|_1\big) - 1\Big), \quad r_k^{\text{tar.}} \propto \exp\Big(-\|\tilde{\varepsilon}_k\|_2^2\Big), \\ r_k^{\text{act.}} &\propto -\|\Delta\mathbf{u}_k/\sigma_u\|_2^2, \quad r_k^{\text{over.}} \propto \Big(\exp\big(-|q_k^{\text{ovs}}|\big) - 1\Big), \\ r_k^{\text{osc.}} &\propto -\|\tilde{\varphi}_k\|_1, \quad r_k^{\text{dec.}} \propto -|\dot{q}_{\text{slew}[k]}|\big(|\dot{q}_{\text{boom}[k]}| + |\dot{q}_{\text{stick}[k]}|\big), \\ r_k^{\text{o-s}} &\propto -\|\mathbf{u}_k/\sigma_u\|_2^2 \cdot \mathcal{I}\big[\|\tilde{\varepsilon}_k\|_2 < 0.5 \wedge |\dot{q}_{\text{slew}[k]}| < 0.02\big]\end{aligned}$$

with quantities

$$\begin{aligned}\tilde{\varepsilon}_k &= \left[\hat{x} - x_{[k]}, \hat{y} - y_{[k]}, \hat{z} - z_{[k]}\right], \\ \mathbf{u}_k &= \left[u_{\text{slew}[k]}, \hat{\dot{q}}_{\text{boom}[k]}, \hat{\dot{q}}_{\text{stick}[k]}\right], \ \tilde{\varphi}_k = \left[\dot{q}_{\text{tool},x[k]}, \dot{q}_{\text{tool},y[k]}\right] \\ q_k^{\text{ovs}} &= \begin{cases}\max\left(0, q_{\text{slew}[k]}\right) & \text{if} \quad q_{\text{slew}[0]} < 0, \\ \min\left(0, q_{\text{slew}[k]}\right) & \text{if} \quad q_{\text{slew}[0]} > 0\end{cases}.\end{aligned}$$

a) Core Reward Terms: We use the r^{balance} penalty to promote a fast target approach while limiting its maximum magnitude early in the episode. We also define a positive reward for reaching the target proximity: r^{target}. Both these terms are scaled and reshaped with curriculum learning [36], a technique which has become extremely popular in RL [37] because it allows agents to acquire complex skills by being tasked with environments of increasing difficulty. Two separate rewards allow for easier and more effective tuning. To reduce the aggressiveness of the policy, we introduce r^{action}, computed on the normalized delta action vector.

b) Decorating Reward Terms: Additional reward terms further shape the behavior towards material handling tasks. The overshoot penalty $r^{\text{overshoot}}$ allows for additional arm motion to cope with the tool oscillations but enforces that the

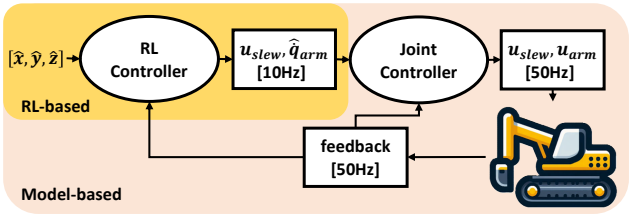

Fig. 5. Schematic of the ROS 2 interface. Nodes are oval, and the communication interfaces are represented in rectangular boxes with message rates. The *RL Controller* outputs three actions $[u_{slew}, \hat{q}_{boom}, \hat{q}_{stick}]$ at 10Hz, interpreted by the *Joint Controller* to provide arm joystick inputs at 50Hz, using FF and PI compensation, while maintaining a constant zero-order hold slew joystick signal for 5 iterations.

slew angle target is reached with proper braking. We penalize the tool velocity oscillations via $r^{\text{oscillation}}$. This term prevents the tool from reaching dangerous configurations during the motion and promotes damping upon target achievement. Further, we include a loss, r^{decouple}, which aims to reduce the coupling between slew and arm motions. As the simulated slew actuator model does not consider inertia variations, this term limits model mismatches. The one-shot penalty $r^{\text{one-shot}}$ prevents additional actions once a target neighborhood has been reached, so that the machine can initiate the grasping phase. We slowly introduce it via curriculum to prevent the agent from diverging during the initial training iterations.

D. RL Slew-Only Controller

To validate the RL-based approach and assess the quality of the slew joint model, we first conducted tests using policies trained to actuate only the rotational slew motor (without the arm joints). This controller primary objective is to learn how to accurately command the nonlinear rotation dynamics to reach target joint positions, neglecting the tool. We utilized some of the reward components outlined in Eq. (7), specifically r^{balance}, r^{target}, r^{action}, $r^{\text{overshoot}}$, $r^{\text{one-shot}}$, but adapted to operate in the joint space.

IV. Evaluation

We deployed the controllers on a prototype machine using a ROS 2 interface as shown in Fig. 5. Tests consist of steady-state position references with transitions every 15 s, similar to typical working routines. We compared policies with differently-tuned rewards and observation vectors to human operators under variable load conditions. The method is benchmarked using four metrics: *i)* average slew speed until steady-state, *ii)* average of the maximum slew overshoots and *iii)* of the errors at steady state for each target position, and *iv)* average of the tool's angular velocity. These are chosen to validate the suitability of our controller for a material dumping routine, which needs to be efficient but also satisfy hard error constraints. Specifically, the grab needs to approach a ship or a dump truck reliably without collisions.

A. Slew-Only Control

Our slew controller incorporates a five-element history access, allowing the agent to learn how to shape the system dynamics with smooth actions and minimal corrections. As shown in Table IV, history makes the controller faster and more accurate, with an average steady-state error of less than 2°. The RL approach can accurately control the slew joint, whereas a model-free PI controller fails. However, this simplified formulation does not address the goal of minimizing the tool oscillations. Although a slower and smoother rotation is helpful, active damping with all joints is necessary to achieve competitive operation speed.

TABLE IV
RL Performance for Slew-only Control.
As a baseline, we use a PI controller fine-tuned to quickly reduce the error within the 15 s target period. We test 2 RL controllers with and without observations history.

Policy	Speed [°/s]	Overshoot [°]	Error [°]	Tool [°/s]
PI control	14.04	38.10	12.38	-
No History	11.75	6.42	**0.46**	36.50
History	**14.44**	**2.98**	1.43	**25.44**

B. Full End-Effector Control

Figure 6 compares the performance of our controller to an operator with 15 years of experience. Resulting trajectories are similar, but the autonomous controller better exploits the simultaneous actuation of all three joints. As reported in Table V, with comparable slew speeds, the RL agent strongly reduces the oscillation of the tool while exhibiting more significant steady-state errors. However, a less experienced driver (1/2 years) controls the machine with an error similar to our policy and much larger oscillations. In Fig. 7, we show experiments without access to the load estimation, demonstrating the controller's robustness to variations due to the randomization applied during training. Our approach can handle both empty and full buckets with comparable performance, as reported in Table VI. This suggests that it could be extended to the manipulation of heavy objects. As seen in Fig. 8, the controller can adjust the final trajectory from observing the dynamics online: with a large load (even if not measured), the gripper is controllable more efficiently, and a direct path can be followed.

TABLE V
RL Performance **with** Load Estimation vs. Human Driver.
We recorded data from different drivers and controllers and averaged the runs to compute the metrics.

Policy	Speed [°/s]	Overshoot [°]	Error [m]	Tool [°/s]
Driver	11.12	**0.57**	**0.593**	20.23
Drv. Inex.	**11.46**	1.49	1.145	21.83
Controller	11.17	7.68	1.078	**10.49**
Drv. Load	**11.12**	**1.55**	**0.470**	12.26
Ctrl. Load	10.77	8.88	1.677	**6.47**

TABLE VI
RL Performance **without** Load Estimation.
Metrics are obtained by averaging different runs.

Load	Speed [°/s]	Overshoot [°]	Error [m]	Tool [°/s]
Empty	**12.43**	7.79	0.843	**11.06**
Robust	11.06	**6.02**	**0.727**	13.29

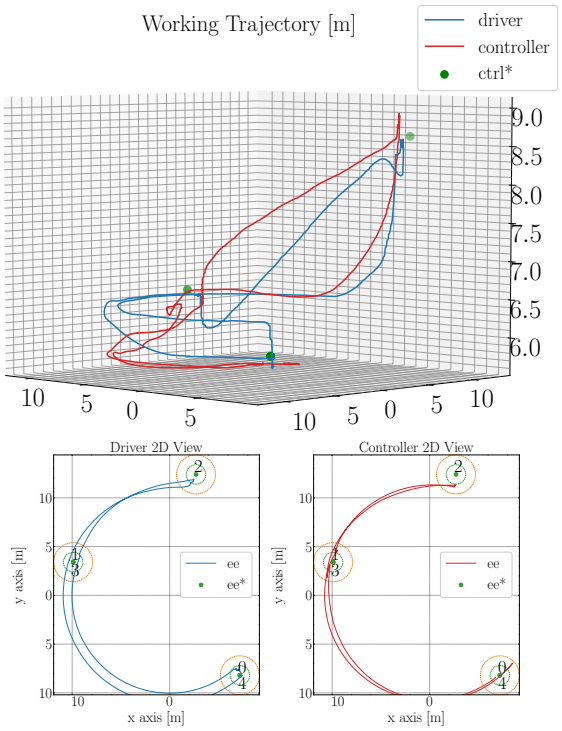

Fig. 6. A sequence of four targets at varying heights and distances is provided to both the controller and the driver. While the driver exhibits greater accuracy, the controller utilizes the entire actuation space better. Targets are numbered from 0 (starting position) to 4 (final position).

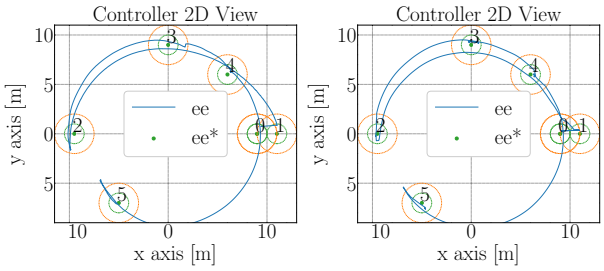

Fig. 7. We evaluate five target positions with tolerances of 1 (green circles) and 2 m (orange). On the left is a controller with a 0.5 s history without a load in the bucket. On the right, a 0.5 s history controller is deployed with an unknown load, performing similarly.

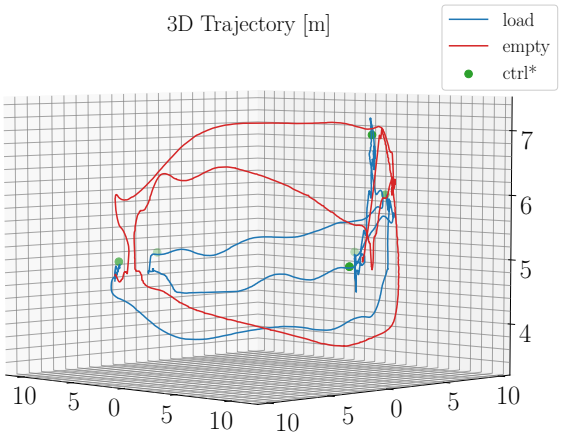

Fig. 8. The controller can adopt different strategies according to the state evolution. When carrying a load, a large height variation to reduce tool oscillations is not required because of the increased gripper inertia.

V. Discussion

Our experiments reveal that stabilizing the tool requires to superimpose additional motions: our controller implicitly learns to reduce tool oscillations through vertical transitions, using inertia to help damping (Fig. 8). This behavior is enabled by the observation history of 0.5 s, which allows to leverage the full machine dynamics. With a shorter observation history, a more conservative and short-sighted control strategy based on tool orientation usually emerges. In practice, these trajectories must be performed accounting for obstacles, a problem which falls beyond the scope of the current work.

Compared to human operators, one drawback of our approach is the increased overshoot. However, when controlling the slew joint only we were able to achieve competitive results for this metric (Table IV). We attribute this problem to the increased task complexity and the additional tool-damping objective, which lead the algorithm to converge to slower and less accurate policies, primarily due to the reward shape: the designed training environment tends to solve the trade-off by prioritizing safety. Other factors contributing to the lowered accuracy are the model mismatch arising from the fast-varying inertia, and the limiting assumptions of the FF arm controllers.

VI. Conclusions & Future Work

In this work, we developed a novel control algorithm to address the automation of material handlers equipped with free-swinging end-effector tools. For this purpose, we used RL to learn a 3D arm position controller, which properly aligns with the task requirements. Our agent controls all DoFs simultaneously, but with different strategies. The slew hydraulic motor was modeled via ML and integrated into the simulation environment, allowing the controller to directly actuate it by learning a correlation between joystick input and velocity output. The arm joints are actuated via simpler velocity controllers, operating over RL-provided velocity references. Our approach simultaneously handles implicit trajectory planning, grab oscillation, and hydraulic joint control, allowing for a trade-off between tracking accuracy, operational speed, and minimization of the end-effector tool oscillations.

Our research, a first-of-its-kind control algorithm for large material handling machines, significantly narrows the gap toward deploying autonomous controllers for material handling. We demonstrated that RL can execute simple tasks competitively compared to average human operators. Despite being less accurate than a very experienced driver, our controller matches their speed and damps the tool under any load conditions more reliably. The investigation of more powerful architectures, such as Temporal Convolution Networks (TCNs) [38] and transformers [39], is part of our future research to address the sim-to-real gap. Additionally, we plan to use low-level controllers trained independently of a specific hydraulic model [40] to improve the velocity tracking performance of the arm joints. To tackle collision avoidance with external bodies, we are currently working

on incorporating multiple dynamic targets into the tracking objective to develop a path-following tool controller.

REFERENCES

[1] Q. Chen, B. G. de Soto, and B. T. Adey, "Construction automation: Research areas, industry concerns and suggestions for advancement," *Automation in Construction*, vol. 94, pp. 22–38, 2018.

[2] D. Jud, P. Leemann, S. Kerscher, and M. Hutter, "Autonomous free-form trenching using a walking excavator," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3208–3215, 2019.

[3] J. Koivumäki and J. Mattila, "Stability-guaranteed force-sensorless contact force/motion control of heavy-duty hydraulic manipulators," *IEEE Transactions on Robotics*, vol. 31, no. 4, pp. 918–935, 2015.

[4] P. Egli and M. Hutter, "A general approach for the automation of hydraulic excavator arms using reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5679–5686, 2022.

[5] M. Lee, H. Choi, C. Kim, J. Moon, D. Kim, and D. Lee, "Precision motion control of robotized industrial hydraulic excavators via data-driven model inversion," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1912–1919, 2022.

[6] E. Jelavic, K. Qu, F. Farshidian, and M. Hutter, "Lstp: Long short-term motion planning for legged and legged-wheeled systems," *IEEE Transactions on Robotics*, vol. 39, no. 6, pp. 4190–4210, 2023.

[7] D. Lee, I. Jang, J. Byun, H. Seo, and H. J. Kim, "Real-time motion planning of a hydraulic excavator using trajectory optimization and model predictive control," 2024, pp. 1–17.

[8] J. Nubert, S. Khattak, and M. Hutter, "Graph-based multi-sensor fusion for consistent localization of autonomous construction robots," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 10 048–10 054.

[9] M. Hirayama, J. Guivant, J. Katupitiya, and M. Whitty, "Path planning for autonomous bulldozers," *Mechatronics*, vol. 58, pp. 20–38, 2019.

[10] L. Terenzi and M. Hutter, "Towards autonomous excavation planning," 2023.

[11] L. Zhang, J. Zhao, P. Long, L. Wang, L. Qian, F. Lu, X. Song, and D. Manocha, "An autonomous excavator system for material loading tasks," *Science Robotics*, vol. 6, no. 55, p. eabc3164, 2021.

[12] R. L. Johns, M. Wermelinger, R. Mascaro, D. Jud, I. Hurkxkens, L. Vasey, M. Chli, F. Gramazio, M. Kohler, and M. Hutter, "A framework for robotic excavation and dry stone construction using on-site materials," *Science Robotics*, vol. 8, no. 84, p. eabp9758, 2023.

[13] M. J. Skibniewski and S. C. Wooldridge, "Robotic materials handling for automated building construction technology," *Automation in Construction*, vol. 1, no. 3, pp. 251–266, 1992.

[14] J. Mattila, J. Koivumäki, D. G. Caldwell, and C. Semini, "A survey on control of hydraulic robotic manipulators with projection to future trends," *iEeE/ASME Transactions on Mechatronics*, vol. 22, no. 2, pp. 669–680, 2017.

[15] Y. Yamamoto, J. Qiu, T. Doi, T. Nanjo, K. Yamashita, and R. Kikuuwe, "A position controller for hydraulic excavators with deadtime and regenerative pipelines," *IEEE Transactions on Automation Science and Engineering*, 2024.

[16] J. Nurmi, M. M. Aref, and J. Mattila, "A neural network strategy for learning of nonlinearities toward feed-forward control of pressure-compensated hydraulic valves with a significant dead zone," in *Fluid Power Systems Technology*, vol. 51968. American Society of Mechanical Engineers, 2018, p. V001T01A023.

[17] J. Park, B. Lee, S. Kang, P. Y. Kim, and H. J. Kim, "Online learning control of hydraulic excavators based on echo-state networks," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 1, pp. 249–259, 2016.

[18] P. Egli and M. Hutter, "Towards rl-based hydraulic excavator automation," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 2692–2697.

[19] R. Dhakate, C. Brommer, C. Bohm, H. Gietler, S. Weiss, and J. Steinbrener, "Autonomous control of redundant hydraulic manipulator using reinforcement learning with action feedback," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 7036–7043.

[20] L. Ramli, Z. Mohamed, A. M. Abdullahi, H. I. Jaafar, and I. M. Lazim, "Control strategies for crane systems: A comprehensive review," *Mechanical Systems and Signal Processing*, vol. 95, pp. 1–23, 2017.

[21] F. Rauscher and O. Sawodny, "Modeling and control of tower cranes with elastic structure," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 1, pp. 64–79, 2020.

[22] J. Zhang, C. Zhao, and J. Ding, "Deep reinforcement learning with domain randomization for overhead crane control with payload mass variations," *Control Engineering Practice*, vol. 141, p. 105689, 2023.

[23] J. Andersson, K. Bodin, D. Lindmark, M. Servin, and E. Wallin, "Reinforcement learning control of a forestry crane manipulator," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 2121–2126.

[24] T. Oktay and C. Sultan, "Modeling and control of a helicopter slung-load system," *Aerospace Science and Technology*, vol. 29, no. 1, pp. 206–222, 2013.

[25] K. Sreenath, N. Michael, and V. Kumar, "Trajectory generation and control of a quadrotor with a cable-suspended load - a differentially-flat hybrid system," in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 4888–4895.

[26] I. Palunko, A. Faust, P. Cruz, L. Tapia, and R. Fierro, "A reinforcement learning approach towards autonomous suspended load manipulation using aerial robots," in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 4896–4901.

[27] S. Zimmermann, R. Poranne, J. M. Bern, and S. Coros, "Puppetmaster: robotic animation of marionettes," *ACM Trans. Graph.*, vol. 38, no. 4, jul 2019.

[28] J. Ichnowski, Y. Avigal, Y. Liu, and K. Goldberg, "Gomp-fit: Grasp-optimized motion planning for fast inertial transport," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 5255–5261.

[29] D. Jud, S. Kerscher, M. Wermelinger, E. Jelavic, P. Egli, P. Leemann, G. Hottiger, and M. Hutter, "Heap-the autonomous walking excavator," *Automation in Construction*, vol. 129, p. 103783, 2021.

[30] P. Egli, D. Gaschen, S. Kerscher, D. Jud, and M. Hutter, "Soil-adaptive excavation using reinforcement learning," *IEEE robotics and automation letters*, vol. 7, no. 4, pp. 9778–9785, 2022.

[31] L. Torrey and J. Shavlik, "Transfer learning," in *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*. IGI global, 2010, pp. 242–264.

[32] E. Minguzzi, "Rayleigh's dissipation function at work," *European Journal of Physics*, vol. 36, no. 3, p. 035014, 2015.

[33] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017.

[34] 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)*. IEEE, 2020, pp. 737–744.

[35] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, "Learning agile and dynamic motor skills for legged robots," *Science Robotics*, vol. 4, no. 26, p. eaau5872, 2019.

[36] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 41–48.

[37] P. Soviany, R. T. Ionescu, P. Rota, and N. Sebe, "Curriculum learning: A survey," pp. 1526–1565, 2022.

[38] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[39] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is All you Need," in *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., 2017.

[40] F. Nan and M. Hutter, "Learning adaptive controller for hydraulic machinery automation," *IEEE Robotics and Automation Letters*, vol. 9, no. 4, pp. 3972–3979, 2024.