



HAL
open science

Data-driven control of a weakly-instrumented excavator with deep learning

Nicolas Hoffmann, Max Cohen, Louis Roullier, Marius Preda, Titus Zaharia

► **To cite this version:**

Nicolas Hoffmann, Max Cohen, Louis Roullier, Marius Preda, Titus Zaharia. Data-driven control of a weakly-instrumented excavator with deep learning. 2024 IEEE 3rd International Conference on Intelligent Reality (ICIR 2024), IEEE, Dec 2024, Coimbra (Portugal), Portugal. hal-04916154

HAL Id: hal-04916154

<https://hal.science/hal-04916154v1>

Submitted on 28 Jan 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Data-driven control of a weakly-instrumented excavator with deep learning

N. Hoffmann, M. Cohen, L. Roullier, M. Preda, T. Zaharia

Abstract—This paper presents a data-driven approach for controlling a weakly-instrumented excavator within a Virtual Reality (VR) supervision environment. We address challenges related to non-linear dynamics and limited sensor data by focusing on arm movement control using both traditional and advanced strategies, including Proportional-Integral-Derivative (PID) controllers, Model Predictive Control (MPC), and Deep Reinforcement Learning (DRL). Our results demonstrate the effectiveness of these methods in achieving precise control despite non-linearities and limited instrumentation, contributing to the broader field of intelligent machine control.

Index Terms—Robotics, Digital twin, Weak instrumentation

I. INTRODUCTION

Controlling heavy machinery, such as excavators, is challenging due to complex dynamics and non-linear behavior, particularly when limited sensor data is available. The excavator arm, with its three hydraulically coupled joints, exemplifies these difficulties. Recent advances in Virtual Reality (VR) and Digital Twins, as shown in our previous work [Hof+22; Sau+23], offer new possibilities for remote supervision, enhancing spatial awareness and providing immersive operator experiences.

This paper tackles the problem of controlling a weakly-instrumented excavator arm within a Virtual Reality (VR)-based supervision environment. We evaluate and compare a range of control methods, from traditional approaches like Proportional Integral Derivative (PID) controllers, including enhancements with Look-Up Table (LUT) and inverse models, to more advanced techniques such as Model Predictive Control (MPC) and Deep Reinforcement Learning (DRL). Our approach seeks to balance simplicity, robustness, and computational efficiency, effectively managing the non-linear dynamics and limited sensing capabilities of the excavator arm.

An excavator, as shown in Figure 1, consists of a hydraulic manipulator mounted on a mobile base. This study concentrates on the control of the excavator arm, which presents significant challenges due to its complex structure. The arm comprises three geometrically coupled hydraulic joints (commonly referred to as the boom, stick, and bucket) that are powered by a shared hydraulic pump. This configuration introduces substantial non-linearities, making precise control particularly difficult for human operators.

A. Problem formulation

We define the state of the excavator arm at time t , denoted as x_t , as the vector of angular positions $\theta_t = (\alpha_t, \beta_t, \gamma_t)$ and velocities $\dot{\theta}_t = (\dot{\alpha}_t, \dot{\beta}_t, \dot{\gamma}_t)$:

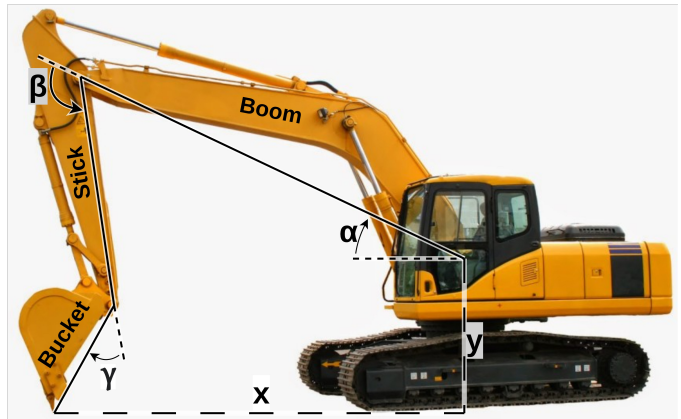


Fig. 1. Side-view of an excavator with articulation coordinates $\theta = (\alpha, \beta, \gamma)$ and operational coordinates $X = (x, y)$.

$$x_t = (\theta_t, \dot{\theta}_t) \quad (1)$$

The control input at time t , denoted as u_t , corresponds to the normalized joystick commands issued to the joints:

$$u_t = [u_t^{boom}, u_t^{stick}, u_t^{bucket}], \quad u_t \in [-1, 1]^3 \quad (2)$$

An estimation of the state, denoted as \hat{x}_t , is derived from the sensor measurements, y_t , provided by Inertial Measurement Unit (IMU) sensors mounted on the machine. This state estimation process involves a combination of low-pass filtering to mitigate sensor noise and numerical differentiation to estimate the angular velocities.

The objective is to compute control commands to drive the excavator arm to achieve the desired angular velocities, $\dot{\theta}_t^d$, as requested by a supervisor in a virtual reality VR environment. We implement the control strategy using a speed controller, as illustrated in Figure 2.

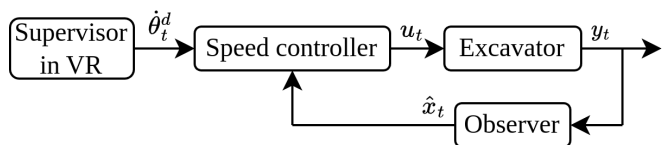


Fig. 2. Control diagram for achieving desired joint velocities based on VR inputs and sensor data.

B. Related work

1) *Supervision tools*: Various interfaces and technologies have been explored for earthworks supervision tools, yielding promising results [LH22]. These technologies include augmented reality [HWS18], 2D interfaces [Oki+19], digital twins [You+23], and advanced environment reconstructions [ZZD20]. Among these, we selected VR for its immersive experience and enhanced spatial awareness. By integrating VR with a digital twin, we aim to provide supervisors with a precise representation of the work environment and refined control capabilities.

2) *Model-free control*: The PID controller remains the most widely used control strategy in industry due to its simplicity and effectiveness. It generates the control command u_t based on the error signal e_t and the proportional, integral, and derivative gains K_p , K_i , and K_d , respectively:

$$u_t = K_p e_t + K_i \int e_t dt + K_d \frac{de_t}{dt} \quad (3)$$

Beyond linear systems, PID controllers can be enhanced through methods such as neural network tuning [MN19] or combined with an inverse model to compute the necessary commands for achieving desired joint speeds in an excavator arm [Lee+22]. However, the latter approach often requires extensive instrumentation to measure hydraulic cylinder pressures, which can be impractical when only joint positions are accessible. Alternatively, a PID controller can be coupled with a LUT (see Figure 3) to approximate the inverse model [Jud+19]. This straightforward method effectively mitigates hydraulic dead-zones but becomes less accurate as the system moves away from the neutral point around which the LUT linearizes the dynamics.

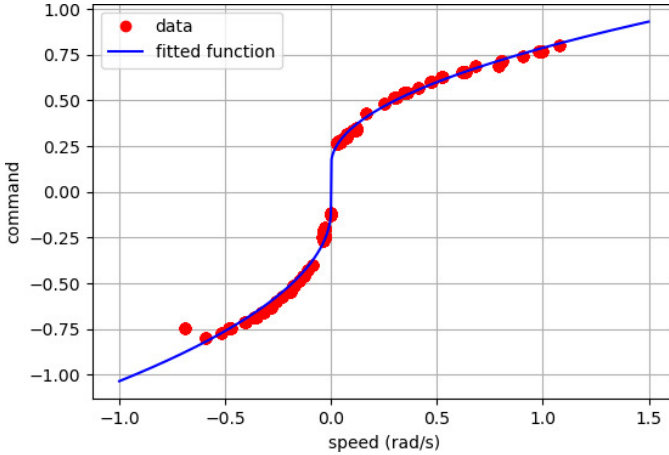


Fig. 3. Look-up table of the boom

3) *System identification*: While model-free control methods are easier to implement, they lack the predictive capabilities that make model-based approaches more effective. A forward

model is essential for predicting the future motion of the excavator based on its current state and input commands:

$$x_{t+1} = f(x_t, u_t) \quad (4)$$

White-box models, such as mathematical and multi-body models, have proven effective for controlling excavators but require significant expertise, frequent updates, and are specific to each machine [Mat+17]. To address these limitations, data-driven black-box models learn directly from input-output data, eliminating the need for explicit physical modeling [HW13]. For example, Multi-Layer Perceptron (MLP) has been used to estimate machine consumption and productivity from telematics data [JLO17; Kas+21].

The Non-linear Auto-Regressive with eXogenous inputs (NARX) formulation enables models to capture temporal features by predicting outputs based on a stack of the n past actions and m past states of the system [Hun+92; NP90]:

$$\hat{x}_t = f(u_{t-1}, \dots, u_{t-n}, x_{t-1}, \dots, x_{t-m}) \quad (5)$$

NARX models with MLP backbones have been effectively used to model excavator arm dynamics and to train position and speed controllers [EH22; EH20; Lee+22]. However, these methods often rely on data such as cylinder pressures or motor RPMs, necessitating costly sensors and increased maintenance, especially when equipping large fleets. Additionally, the finite input stack length limits the ability of the network to capture information beyond the specified time window.

Recurrent Neural Network (RNN), such as Long Short-Term Memory (LSTM) networks [HS97], offer an alternative by more effectively capturing temporal dependencies. Unlike NARX MLPs, which stack inputs over time, LSTMs maintain a hidden state that allows them to model complex temporal dependencies without a fixed time horizon. For instance, LSTMs have been used to model the activity of a Komatsu PC220LC excavator with 99.78% accuracy using only a few cellphone sensors and a Convolutional LSTM (Conv-LSTM) network [Mah+22].

4) *Model predictive control*: MPC optimizes a sequence of control commands over a finite horizon H by minimizing a cost function. At each time step, the first command from the optimized sequence is applied, and the optimization process is repeated:

$$U_t^* = u_t^*, \dots, u_{t+H}^* = \arg \min_{U_t} J(x_t, U_t) \quad (6)$$

The cost function $J(x_t, U_t)$ evaluates the desirability of the predicted trajectory starting from state x_t and following the input sequence U_t :

$$J(x_t, U_t) = \varphi(x_{t+H+1}) + \sum_{\tau=t}^{t+H} L(x_\tau, u_\tau) \quad (7)$$

This approach requires a dynamic model to simulate system trajectories in response to input sequences. Typically, this

model is a white-box mathematical model—a set of differentiable equations derived from a physical analysis of the system, which can be locally linearized for faster optimization. However, when using a neural network model that is difficult to linearize, a sampling-based MPC method can be employed [How+22], where the model serves as a simulator to compute trajectory rollouts. This approach has been validated in simulations using a model that accounts for the impact of internal and external forces on the system state [Jin+24].

5) *Deep reinforcement learning*: DRL combines Reinforcement Learning (RL) and Deep Learning (DL) to train neural networks through interactions without the need for labeled data. The problem is modeled as a Markov Decision Process (MDP), defined by the tuple (S, A, P, R, γ) , where S is the set of states, A is the set of actions, $P(s' | s, a)$ is the transition probability from state s to state s' when action a is taken, $R(s, a)$ is the reward function, and γ is the discount factor. The goal is to find a policy $\pi : S \rightarrow A$ that maximizes the expected cumulative discounted reward:

$$G = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \right] \quad (8)$$

Several studies have applied DRL techniques to control robotic arms and excavators, both in real-world and simulated environments, for tasks ranging from free-space manipulation to digging operations (see Table I). For instance, [Par+14; Par+17] developed online position controllers using Echo State Network (ESN), adjusting dynamically to system changes and demonstrating improved tracking on a real 21-ton excavator over traditional PD control. [Kur+20] combined multibody simulation with RL to train a PPO-CMA agent for efficient hopper filling, achieving high bucket loads in simulation. [Egl+22] created a compliant RL controller adaptable to various soil conditions, successfully tested on a real 12-ton excavator with high bucket fill ratios across different soil hardness levels. [SA21] learned a low-dimensional, causal linear model of excavator data using simulation with AgxTerrain. [LH22] introduced a physics-inspired data-driven model inversion method, enabling a 38-ton excavator to dig accurately and quickly with minimal tracking error. [ISR23] developed an RL position controller using Proximal Policy Optimization (PPO) in simulation, achieving low position tracking errors using only proprioceptive inputs. Lastly, [JYZ23] used imitation learning to teach an excavator to penetrate heterogeneous terrain without jamming, demonstrating improved performance over the training dataset on a real robotic arm equipped with a custom bucket.

C. Contributions

This work focuses on enhancing supervision capabilities for operators in the earthwork industry by evaluating and comparing control strategies, including PID, MPC, and DRL. Our approach is distinguished by two key aspects: the use of real machine data, avoiding reliance on simulators, and the

	Device		
Task	Simulation	Robot Arm	Excavator
Free-Space	[Hod18]		[EH20; EH22]
Digging	[Par+14; Kur+20] [Ran23]	[JYZ23]	[Par+17; Egl+22] [LH22]

TABLE I
DRL PUBLICATIONS SORTED BY TASK AND DEVICE.

emphasis on minimally instrumented, cost-effective machines. Furthermore, we focus on general methods that extend beyond excavators. Our findings demonstrate that DRL provides strong performance while maintaining a minimal computational load during operation.

D. Plan

The rest of this paper is structured as follows: Section II details the system identification and modeling techniques used to capture the excavator’s dynamics. Section III discusses the design of PID, MPC and DRL speed controllers. Section IV provides a comprehensive evaluation of the proposed methods, and Section V concludes with a discussion of the findings and potential future work.

II. SYSTEM IDENTIFICATION AND MODELING

Building on our previous work [Hof+24], we collected approximately one million transition samples $(\theta_t, u_t, \theta_{t+1})$ from 20 trajectories, each 5 minutes long, recorded at 100 Hz using a weakly instrumented CAT 323 excavator. To ensure dense sampling across the joint space, we oscillated the boom, stick, and bucket out of phase using a manually tuned PID controller (see Figure 4).

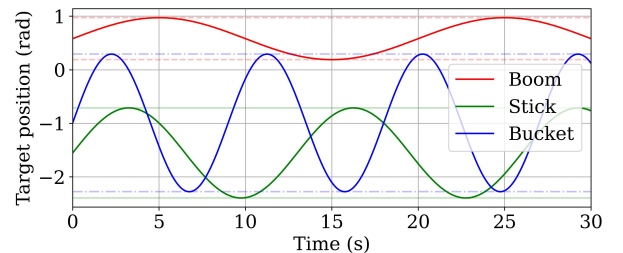


Fig. 4. Target joint positions oscillating out-of-phase to densely sample the joint space.

We evaluated MLP and LSTM networks, both with and without input stacking, to predict state variations $\Delta\theta_t = \theta_{t+1} - \theta_t$. The stacked-input LSTM architecture demonstrated the best performance, achieving the lowest validation loss and closely tracking the ground truth in free-run simulations over extended periods, as illustrated in Figure 5.

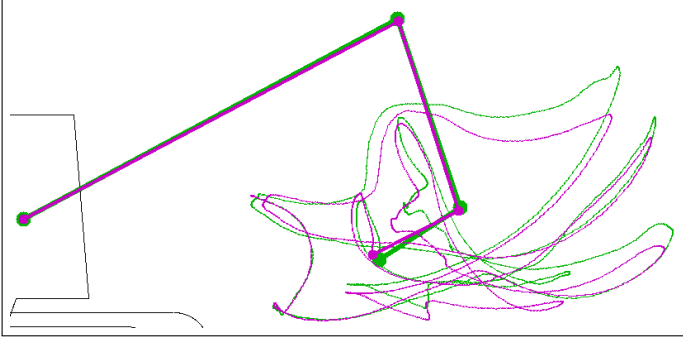


Fig. 5. Free-run simulation of the stacked-input LSTM over a 60-second trajectory.

A. Identification for control

Up to this point, our system modeling focused on maximizing prediction accuracy, as measured by the prediction loss in free-run simulations. However, this objective diverges from our primary goal of improving velocity control for the excavator operator. To address this, we trained an alternative, faster model that predicts 0.1 seconds into the future instead of 0.01 seconds, allowing for quicker inference over time. This faster model enables the controller to simulate more alternative control inputs within the same computational budget, enhancing performance as long as the model remains accurate.

Another approach, proposed by [Lee+22], involves using an inverse kinematic model to predict the control inputs required to achieve a desired velocity at a given position. The core idea is to employ a neural network to model the non-linear relationship between joint velocities and control inputs. While the original publication developed a sophisticated model incorporating force sensors, we adapt this method to a weakly instrumented setting (without force sensors) and we rely solely on the input-output data available from the neural network. We model this inverse relationship by redefining the state and actions as follows:

$$x_t^{\text{inv}} = (\alpha_{t+1}, \beta_{t+1}, \gamma_{t+1}, u_t^{\text{boom}}, u_t^{\text{stick}}, u_t^{\text{bucket}}) \quad (9)$$

$$u_t^{\text{inv}} = (\dot{\alpha}_{t+1}, \dot{\beta}_{t+1}, \dot{\gamma}_{t+1}) \quad (10)$$

The time offset in the velocities and positions maintains the causal relationship from inputs to motion, ensuring that the predicted control inputs are based on currently known states and affect future motions.

III. CONTROLLER DESIGN

This section details the environments used for training and validating our controllers, followed by the description of the PID, MPC, and DRL controllers. The performances of each controller are evaluated and analyzed at the end of this section.

A. Training and validation environments

We use the model derived in the previous section to simulate and evaluate various speed controllers presented in this section. As shown in Figure 6, we emulate the desired speed trajectory specified by the supervisor by following operational position trajectories, denoted as X_t^d .

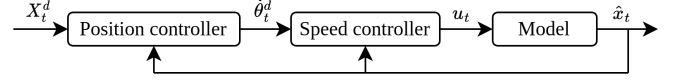


Fig. 6. Control strategy to train and validate speed controllers.

The desired operational position trajectories differ when training and validating the controllers:

- **Training Phase:** During training, the controller is tasked with reaching positions uniformly sampled in free space within uniformly sampled time intervals. The path between the current position of the arm and its target is a straight line. The operational position trajectory, X_t^d , progresses along this path at varying speeds, occasionally moving very quickly and other times more slowly.
- **Validation Phase:** Validation involves following predefined operational trajectories of interest, X_t^d , such as moving in a circle or scooping the ground to fill the bucket (see Figure 7) at different velocities.

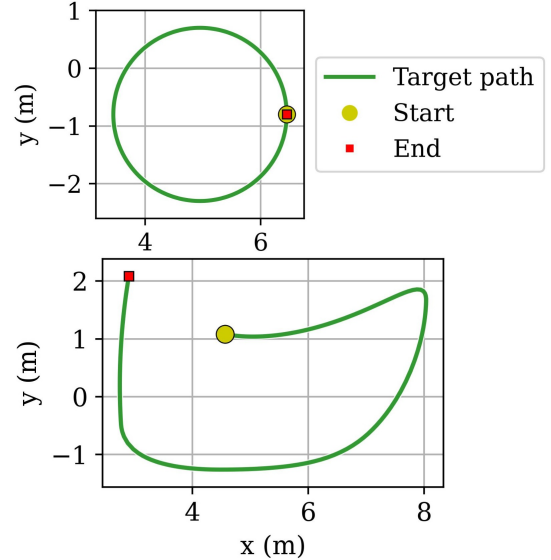


Fig. 7. Circular and scooping validation paths.

To compute the desired joint velocities, we use a proportional (P) position controller in combination with the inverse Jacobian matrix:

$$\dot{\theta}_t^d = J^{pinv}(X_t^d - X_t) \quad (11)$$

This method ensures that our controller is trained on a wide range of trajectories that the supervisor might consider and rigorously evaluates the controllers on the trajectories that are of primary interest.

B. PID controller design

As a baseline, we implemented a PID controller, even though our system exhibits significant non-linearities. We tuned the PID parameters (K_p , K_i , and K_d) using the Tree-structured Parzen Estimator (TPE) algorithm [Wat23], aiming to optimize a Pareto front that balances speed and precision. Figure 8 presents this Pareto front, along with the most promising PID controllers, under the label "PID," and compares them to other controllers.

Building on the state of the art, we further enhanced the PID controller by integrating it with a LUT (Look-Up Table) to linearize the system around its midpoint. In accordance with the approach described in [Lee+22], we also combined the PID controller with an inverse model of the environment, which calculates the commands required to achieve a desired speed. These two methods are also compared in Figure 8 under the labels "PID+LUT" and "PID+INV".

C. Model predictive control design

Our first model-based control method is sampling-based Model Predictive Control (MPC). Unlike other MPC methods that require a differentiable model of the system dynamics, the sampling-based approach only requires the capability to simulate the forward dynamics of the system. In this study, our objective is to minimize the Mean Squared Error (MSE) over the entire trajectory:

$$J(x_t, U_t) = \sum_{t=0}^H \|\dot{\theta}_t^d - \dot{\theta}_t\|_2 \quad (12)$$

Neural network models offer generality but have slower inference rates. We do not use the high-fidelity plant model in the MPC, but another model trained to predict farther in time, making it effectively faster, see section II-A. This faster internal model allows the MPC to simulate a few dozens trajectories each time-step, depending on the choice of hyperparameters presented in Table II. We also accelerate the optimization step by parameterizing the input trajectory using either constant functions, steps, or splines.

We optimize the hyperparameters with TPE and evaluate our most promising MPC controllers in Figure 8, labeled "MPC CST," "MPC STEP," and "MPC SPLINE".

Hyper-parameter	Range
Horizon	[0.1, 0.5]
Parametrization	[Constant, Step, Spline]

TABLE II
HYPER-PARAMETERS OF THE MPC CONTROLLERS.

D. Deep reinforcement learning design

Inspired by the approach in [EH22], we employ a DRL agent to control the excavator arm. The agent is trained in simulated environments and its performances are evaluated in our separate validation setting. The agent computes the

next control commands from the current joint positions and velocities, desired joint velocities, and previous control commands, as show in Table III.

Agent Inputs (12)

Joint positions (3)	$\theta_t = (\alpha_t, \beta_t, \gamma_t)$
Joint velocities (3)	$\dot{\theta}_t = (\dot{\alpha}_t, \dot{\beta}_t, \dot{\gamma}_t)$
Target joint velocities (3)	$\dot{\theta}_t^d = (\dot{\alpha}_t^d, \dot{\beta}_t^d, \dot{\gamma}_t^d)$
Previous control commands (3)	$u_{t-1} = (u_{t-1}^{\text{boom}}, u_{t-1}^{\text{stick}}, u_{t-1}^{\text{bucket}})$

Agent Outputs (3)

Control commands (3)	$u_t = (u_t^{\text{boom}}, u_t^{\text{stick}}, u_t^{\text{bucket}})$
----------------------	--

TABLE III
INPUTS AND OUTPUTS OF THE DRL CONTROLLER.

1) *Reward definition:* We define the reward function as a weighted sum of several terms, each designed to guide the agent toward desired objectives while mitigating undesirable behaviors:

$$r_t = \lambda_1 \cdot r_t^{\dot{\theta}} + \lambda_2 \cdot r_t^u + \lambda_3 \cdot r_t^{\Delta u} + \lambda_4 \cdot r_t^{\text{saf}} \quad (13)$$

where the individual terms are defined as follows:

- **Tracking error penalty:** $r_t^{\dot{\theta}} = -\|\dot{\theta}_t^d - \dot{\theta}_t\|_2$
This term penalizes deviations from the desired joint velocities, encouraging accurate tracking.
- **Control amplitude penalty:** $r_t^u = -\|u_t\|_2$
This term penalizes large control inputs to limit command amplitudes and reduce energy consumption.
- **Control variation penalty:** $r_t^{\Delta u} = -\|u_t - u_{t-1}\|_2$
This term penalizes abrupt changes in control commands, promoting smoother operation.
- **Hydraulics safety penalty:**

$$r_t^{\text{saf}} = -\|v_t\|_2, \text{ where } v_t^i = \begin{cases} u_t^i \cdot \dot{\theta}_t^i < 0, \\ 0, & \text{otherwise,} \end{cases}$$

for each joint i . This term penalizes control inputs that oppose the current joint velocities. While this strategy can improve tracking performance by decelerating joints more rapidly, it places excessive stress on the hydraulic components and can lead to mechanical damage.

After manual tuning and analysis of the results, we assigned the following weights to the reward terms: $\lambda_1 = 1.0$, $\lambda_2 = 0.05$, $\lambda_3 = 0.05$ and $\lambda_4 = 0.1$.

2) *Hyperparameter optimization:* We optimize the hyperparameters of PPO presented Table IV, with TPE and compare the performance of our most promising DRL controllers in Figure 8, labeled as "DRL".

E. Results

Our objective is to maximize the speed of the bucket while minimizing the tracking error. To evaluate the performance of the controllers, we define two key metrics:

Hyperparameter	Search Range
Policy network depth	1 to 6
Policy network width	8 to 512
Value network depth	1 to 6
Value network width	8 to 512
Learning rate	3×10^{-5} to 3×10^{-3}
Batch size	4 to 4096
Number of steps per update	4 to 256

TABLE IV

HYPERPARAMETERS OF THE PPO ALGORITHM OPTIMIZED USING TPE.

- **Average tracking error**

$$X_{e, \text{avg}} := \frac{1}{N} \sum_{t=1}^N \|X_t^d - X_t\|_2 \quad (14)$$

- **Average bucket speed**

$$|\dot{X}|_{\text{avg}} := \frac{1}{N} \sum_{t=1}^N |\dot{X}_t| \quad (15)$$

We evaluate all controllers presented in the previous sections, in the validation environment. Figure 8 illustrates the Pareto fronts and the most promising controllers.

1) *Analysis of PID Controllers:* We observe that the Proportional-Integral-Derivative (PID) controllers perform competitively, especially those utilizing Look-Up Table (LUT) linearization. The PID with inverse model (PID+INV) approach exhibited poor performance due to the inadequacy of the inverse model when trained as-is. In contrast, the original publication [Lee+22] proposing this method achieved impressive results (3cm of error at 66cm/s) by developing a detailed inverse model specific to their excavator and employing force sensors. Our study aims to maintain an approach that is applicable as-is to any minimally instrumented machine.

2) *Analysis of MPC Controllers:* The Model Predictive Control (MPC) approaches perform comparably to the PID controllers but are highly sensitive to the parameterization of the control signal. This sensitivity arises from the slow inference speed of our model, which limits the number of samples available for online optimization. Among the MPC methods, those using splines perform the best, as they strike a balance between input richness and a manageable optimization space.

3) *Analysis of DRL Controllers:* The Deep Reinforcement Learning (DRL) controllers outperform all other approaches except for the MPC with splines at specific velocities. They can achieve very high speeds exceeding 2m/s while maintaining a tracking error of approximately 60cm, depending on the requirements. Unlike MPC approaches, DRL controllers do not require intensive online computations, as they offload the computational burden to the offline training phase.

IV. DISCUSSION AND CONCLUSION

This study evaluates the effectiveness of data-driven control strategies for weakly-instrumented excavators, in a task critical for VR supervision environments. By leveraging real machine data and focusing on generalizable, cost-effective methods, we have shown that control strategies such as PID, MPC, and especially DRL can provide good speed control performance of earthwork machinery without imposing a heavy computational load during operation.

This work demonstrates the applicability of data-driven control techniques to assist earthwork operators. Our findings suggest several promising directions for future work. First, deploying these methods in real-world, uncontrolled environments will enable us to further test their robustness and adaptability under varying conditions. Second, a more thorough investigation of the reality gap under different external conditions, such as various soil types and terrains, would be necessary before considering widespread deployment. We would like to work on stronger safety guarantees and examine how the various control strategies perform in hazardous situations, including system failures or sensor malfunctions. Finally, the trade-off between MPC prediction capabilities and its computational demands could be further optimized.

REFERENCES

- [Egl+22] Pascal Egli et al. “Soil-Adaptive Excavation Using Reinforcement Learning”. In: *IEEE Robotics and Automation Letters* 7 (Oct. 2022), pp. 1–8. DOI: 10.1109/LRA.2022.3189834.
- [EH20] Pascal Egli and Marco Hutter. “Towards RL-Based Hydraulic Excavator Automation”. In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2020, pp. 2692–2697. DOI: 10.1109/IROS45743.2020.9341598.
- [EH22] Pascal Egli and Marco Hutter. “A General Approach for the Automation of Hydraulic Excavator Arms Using Reinforcement Learning”. In: *IEEE Robotics and Automation Letters* 7.2 (2022), pp. 5679–5686. DOI: 10.1109/LRA.2022.3152865.
- [Hod18] Benjamin J. Hodel. “Learning to Operate an Excavator via Policy Optimization”. In: *Procedia Computer Science* 140 (2018). Cyber Physical Systems and Deep Learning Chicago, Illinois November 5-7, 2018, pp. 376–382. ISSN: 1877-0509. DOI: <https://doi.org/10.1016/j.procs.2018.10.301>. URL: <https://www.sciencedirect.com/science/article/pii/S1877050918319744>.

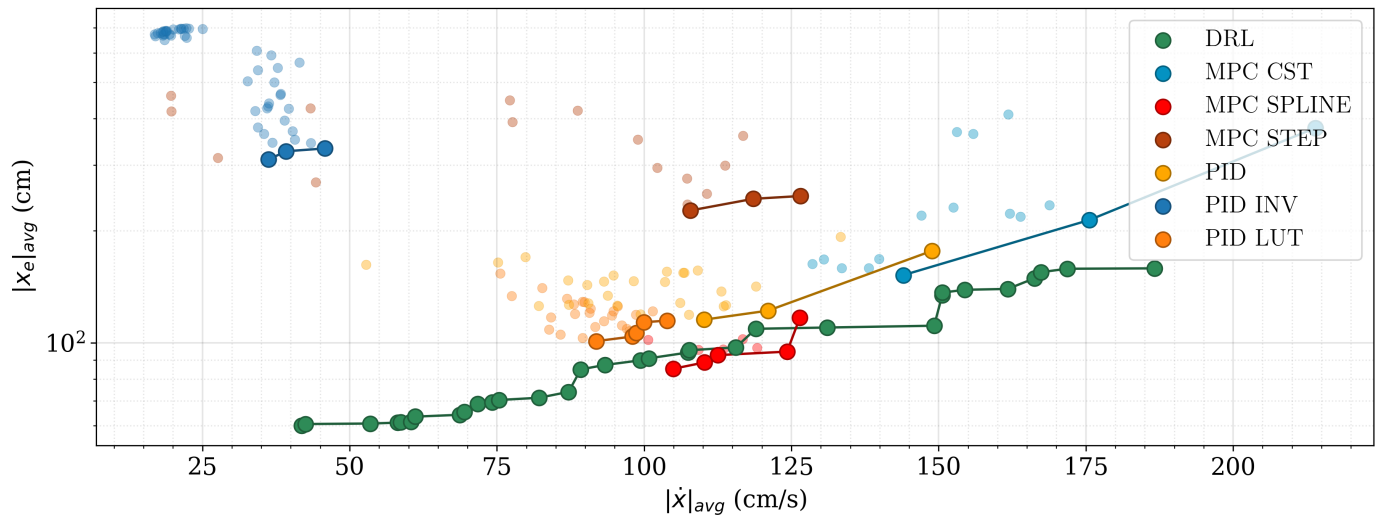


Fig. 8. Comparison of control performance across various methods. Pareto fronts indicate the trade-off between average tracking error and bucket speed for each control strategy.

[Hof+22] Nicolas Hoffmann et al. “Industrial Use-Case: Digital Twin for Autonomous Earthwork in Virtual-Reality”. In: *Proceedings of the 27th International Conference on 3D Web Technology*. Web3D '22. Evry-Courcouronnes, France: Association for Computing Machinery, 2022. ISBN: 9781450399142. DOI: 10.1145/3564533.3565803. URL: <https://doi.org/10.1145/3564533.3565803>.

[Hof+24] Nicolas Hoffmann et al. “Modeling weakly-instrumented excavator arm dynamics with stacked-input LSTM”. In: *The 12th International Conference on Robot Intelligence Technology and Applications* (2024).

[How+22] Taylor Howell et al. *Predictive Sampling: Real-time Behaviour Synthesis with MuJoCo*. 2022. arXiv: 2212.00541 [cs.RO]. URL: <https://arxiv.org/abs/2212.00541>.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. “Long Short-term Memory”. In: *Neural computation* 9 (Dec. 1997), pp. 1735–80. DOI: 10.1162/neco.1997.9.8.1735.

[Hun+92] K.J. Hunt et al. “Neural networks for control systems—A survey”. In: *Automatica* 28.6 (1992), pp. 1083–1112. ISSN: 0005-1098. DOI: [https://doi.org/10.1016/0005-1098\(92\)90053-I](https://doi.org/10.1016/0005-1098(92)90053-I). URL: <https://www.sciencedirect.com/science/article/pii/S000510989290053I>.

[HW13] Zhong-Sheng Hou and Zhuo Wang. “From model-based control to data-driven control: Survey, classification and perspective”. In: *Information Sciences* 235 (2013). Data-based Control, Decision, Scheduling and Fault Diagnostics, pp. 3–35. ISSN: 0020-0255. DOI: <https://doi.org/10.1016/j.ins.2012.07.014>. URL: <https://www.sciencedirect.com/science/article/pii/S0020025512004781>.

[HWS18] Hooman Hedayati, Michael Walker, and Daniel Szafrir. “Improving Collocated Robot Teleoperation with Augmented Reality”. In: *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*. HRI '18. Chicago, IL, USA: Association for Computing Machinery, 2018, pp. 78–86. ISBN: 9781450349536. DOI: 10.1145/3171221.3171251. URL: <https://doi.org/10.1145/3171221.3171251>.

[ISR23] Cendikia Ishmatuka, Indah Soesanti, and Ahmad Ataka Awwalur Rizqi. “Autonomous Pick-and-Place Using Excavator Based on Deep Reinforcement Learning”. In: Oct. 2023. DOI: 10.1109/ICITEE59582.2023.10317662.

[Jin+24] Zhong Jin et al. “MPC motion control of boom descending of unmanned excavator based on regenerative valve”. In: *Heliyon* 10.9 (2024), e29915. ISSN: 2405-8440. DOI: <https://doi.org/10.1016/j.heliyon.2024.e29915>. URL: <https://www.sciencedirect.com/science/article/pii/S2405844024059462>.

[JLO17] Hassanean S. H. Jassim, Weizhuo Lu, and Thomas Olofsson. “Predicting Energy Consumption and CO2 Emissions of Excavators in Earthwork Operations: An Artificial Neural Network Model”. In: *Sustainability* 9.7 (2017). ISSN: 2071-1050. DOI: 10.3390/su9071257. URL: <https://www.mdpi.com/2071-1050/9/7/1257>.

[Jud+19] Dominic Jud et al. “Autonomous Free-Form Trenching Using a Walking Excavator”. In: *IEEE Robotics and Automation Letters* PP (June 2019), pp. 1–1. DOI: 10.1109/LRA.2019.2925758.

- [JYZ23] Shiyu Jin, Zhixian Ye, and Liangjun Zhang. “Learning Excavation of Rigid Objects with Offline Reinforcement Learning”. In: (2023). arXiv: 2303.16427 [cs.LG].
- [Kas+21] Mohamad Kassem et al. “Measuring and benchmarking the productivity of excavators in infrastructure projects: A deep neural network approach”. In: *Automation in Construction* 124 (2021), p. 103532. ISSN: 0926-5805. DOI: <https://doi.org/10.1016/j.autcon.2020.103532>. URL: <https://www.sciencedirect.com/science/article/pii/S0926580520311122>.
- [Kur+20] Ilya Kurinov et al. “Automated Excavator Based on Reinforcement Learning and Multibody System Dynamics”. In: *IEEE Access* 8 (2020), pp. 213998–214006. DOI: 10.1109/ACCESS.2020.3040246.
- [Lee+22] Minhyeong Lee et al. “Precision Motion Control of Robotized Industrial Hydraulic Excavators via Data-Driven Model Inversion”. In: *IEEE Robotics and Automation Letters* 7.2 (2022), pp. 1912–1919. DOI: 10.1109/LRA.2022.3142389.
- [LH22] Jin Sol Lee and Youngjib Ham. “Exploring Human-Machine Interfaces for Teleoperation of Excavator”. In: *Construction Research Congress 2022*. 2022, pp. 757–765. DOI: 10.1061/9780784483961.079. eprint: <https://ascelibrary.org/doi/pdf/10.1061/9780784483961.079>. URL: <https://ascelibrary.org/doi/abs/10.1061/9780784483961.079>.
- [Mah+22] Elham Mahamedi et al. “Automating excavator productivity measurement using deep learning”. In: *Proceedings of the Institution of Civil Engineers - Smart Infrastructure and Construction* 174 (Apr. 2022), pp. 1–13. DOI: 10.1680/jsmic.21.00031.
- [Mat+17] Jouni Mattila et al. “A Survey on Control of Hydraulic Robotic Manipulators With Projection to Future Trends”. In: *IEEE/ASME Transactions on Mechatronics* 22.2 (2017), pp. 669–680. DOI: 10.1109/TMECH.2017.2668604.
- [MN19] Antonio Marino and Filippo Neri. “PID Tuning with Neural Networks”. In: *Intelligent Information and Database Systems* (Mar. 2019), pp. 476–487. DOI: 10.1007/978-3-030-14799-0_41.
- [NP90] K.S. Narendra and K. Parthasarathy. “Identification and control of dynamical systems using neural networks”. In: *IEEE Transactions on Neural Networks* 1.1 (1990), pp. 4–27. DOI: 10.1109/72.80202.
- [Oki+19] Shunsuke Okishiba et al. “Tablet interface for direct vision teleoperation of an excavator for urban construction work”. In: *Automation in Construction* 102 (2019), pp. 17–26. ISSN: 0926-5805. DOI: <https://doi.org/10.1016/j.autcon.2019.02.003>. URL: <https://www.sciencedirect.com/science/article/pii/S0926580517308294>.
- [Par+14] Jaemann Park et al. “Utilizing online learning based on echo-state networks for the control of a hydraulic excavator”. In: *Mechatronics* 24.8 (2014), pp. 986–1000. ISSN: 0957-4158. DOI: <https://doi.org/10.1016/j.mechatronics.2014.10.004>. URL: <https://www.sciencedirect.com/science/article/pii/S0957415814001627>.
- [Par+17] Jaemann Park et al. “Online Learning Control of Hydraulic Excavators Based on Echo-State Networks”. In: *IEEE Transactions on Automation Science and Engineering* 14.1 (2017), pp. 249–259. DOI: 10.1109/TASE.2016.2582213.
- [Ran23] J. Rankin. *Reinforcement Learning for Trench Excavation*. Loughborough University, 2023. URL: <https://books.google.fr/books?id=jQEy0AEACAAJ>.
- [SA21] Nicholas S. Selby and H. Harry Asada. “Learning of Causal Observable Functions for Koopman-DFL Lifting Linearization of Nonlinear Controlled Systems and Its Application to Excavation Automation”. In: *IEEE Robotics and Automation Letters* 6.4 (Oct. 2021), pp. 6297–6304. DOI: 10.1109/lra.2021.3092256. URL: <https://doi.org/10.1109%5C%2Flra.2021.3092256>.
- [Sau+23] Léa Saunier et al. “Virtual Reality Interface Evaluation for Earthwork Teleoperation”. In: *Electronics* 12.19 (2023). ISSN: 2079-9292. DOI: 10.3390/electronics12194151. URL: <https://www.mdpi.com/2079-9292/12/19/4151>.
- [Wat23] Shuhei Watanabe. *Tree-Structured Parzen Estimator: Understanding Its Algorithm Components and Their Roles for Better Empirical Performance*. 2023. arXiv: 2304.11127 [cs.LG]. URL: <https://arxiv.org/abs/2304.11127>.
- [You+23] Ke You et al. “Earthwork digital twin for teleoperation of an automated bulldozer in edge dumping”. In: *Journal of Field Robotics* 40.8 (2023), pp. 1945–1963. DOI: <https://doi.org/10.1002/rob.22234>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/rob.22234>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.22234>.
- [ZZD20] Tianyu Zhou, Qi Zhu, and Jing Du. “Intuitive robot teleoperation for civil engineering operations with virtual reality and deep learning scene reconstruction”. In: *Advanced Engineering Informatics* 46 (2020), p. 101170. ISSN: 1474-0346. DOI: <https://doi.org/10.1016/j.aei.2020.101170>. URL: <https://www.sciencedirect.com/science/article/pii/S1474034620301415>.