Precision Motion Control of Robotized Industrial Hydraulic Excavators via Data-Driven Model Inversion

Minhyeong Lee, *Graduate Student Member, IEEE*, Hyelim Choi, *Graduate Student Member, IEEE*, ChangU Kim, Jihyun Moon, Dongmok Kim, and Dongjun Lee, *Member, IEEE*

***Abstract*—This work proposes a novel precision motion control framework of robotized industrial hydraulic excavators via data-driven model inversion. Rather than employing a single neural network to approximate the whole excavator dynamics, including input delays and dead-zones, we construct a physics-inspired data-driven model with a modular structure. The data-driven model is then inverted in a modular fashion which benefits the training speed. The data-driven model and its inversion are trained offline in a supervised manner using the real operational data since online learning methods can damage the machine and surroundings. The entire motion control framework consists of the data-driven model inversion that compensates for the excavator dynamics and the proportional control that determines the input of the model inversion to enhance the robustness. The framework is experimentally validated with a commercial 38-ton class hydraulic excavator for digging and grading tasks, achieving a precise control performance (i.e., root-mean-square of the path following error under 2 $[\mathrm{cm}]$) even under severe soil interactions.**

***Index Terms*—Industrial hydraulic excavator, data-driven, model inversion, input delay, dead-zone, soil interaction.**

I. Introduction

Hydraulic actuator systems are widely employed in various engineering domains due to their high power-to-weight ratio, reliability, and affordability. In particular, hydraulic excavators are essential in construction, demolition, mining, and forestry where large operating forces are required. However, multiple joints of the excavator must be manipulated simultaneously while maximizing the operation efficiency, making the manipulation a demanding task that must be performed by skilled operators with many years of experience. In this regard, the automation of the excavators has drawn a great interest [1] to reduce human-associated costs (e.g., fatigue, safety, etc.). Central to this automation is a precise motion control, but it still remains a challenging problem in practical settings [2], particularly with complex soil interactions.

In this paper, we propose a novel precision motion control framework of robotized industrial hydraulic excavators based on a data-driven model inversion. The data-driven inversion is challenging for typical data-driven approaches (e.g., methods using a recurrent neural network (RNN), a multilayer perceptron (MLP), etc.), as it is expensive to represent the inverse of time-related behaviors and to train discontinuous relations. The hydraulic excavators, however, are under the effect of input delays and dead-zones, which intensify in the presence of complex hydraulic circuits (e.g., a main control valve (MCV) [3]) commonly found in industrial hydraulic settings. To address these distinct features, we introduce a physics-inspired data-driven model with a modular structure composed of the following neural network modules: 1) an infinite impulse response (IIR) unit, which accommodates the input delays; 2) a piecewise linear (PL) map, which deals with the state-dependent dead-zones; and 3) MLP networks, which capture the remaining nonlinear and coupled dynamics. The environmental impacts (e.g., soil interaction forces) and the hydraulic states (e.g., hydraulic pressures) are taken into account by including the measurements in the network input.

Learning the data-driven model and its inversion online can endanger the excavator and the environments, thus the learning is done offline in a supervised manner using the operational data of the real machine. We then design our control to consist of the following two layers: 1) the data-driven model inversion control, constructed as an inversion of the data-driven model in a modular fashion that significantly enhances the training speed; and 2) the proportional (P) control, implemented on top of the data-driven model inversion control to enhance the robustness. The stability and robustness of the control framework are theoretically established. Even in the presence of intense soil interactions, the proposed control framework accomplishes a remarkable performance (i.e., the path following root-mean-square error (RMSE) less than 2 $[\mathrm{cm}]$) for digging and grading operations of a commercial 38-ton class hydraulic excavator Doosan DX380LC.

Model-based methods have been proposed [3]–[6] for the control of hydraulic excavators, but they adopted simplifications in modeling which necessarily compromise the control

Manuscript received September 9, 2021; accepted December 25, 2021. Date of publication January 13, 2022; date of current version January 21, 2022. This letter was recommended for publication by Associate Editor Gianluca Garofalo and Editor Youngjin Choi upon evaluation of the reviewers' comments. This work was supported by the Hyundai Doosan Infracore. *(Minhyeong Lee and Hyelim Choi contributed equally to this work.) (Corresponding author: Dongjun Lee.)*

Minhyeong Lee, Hyelim Choi, ChangU Kim, and Dongjun Lee are with the Department of Mechanical Engineering, IAMD and IER, Seoul National University, Seoul 08826, South Korea (e-mail: minhyeong@snu.ac.kr; helmchoi@snu.ac.kr; kcu11@snu.ac.kr; djlee@snu.ac.kr).

Jihyun Moon and Dongmok Kim are with the Hyundai Doosan Infracore, Incheon 22502, South Korea (e-mail: jihyun3.moon@hyundai-di.com; dongmok.kim@hyundai-di.com).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2022.3142389, provided by the authors.

Digital Object Identifier 10.1109/LRA.2022.3142389

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

performance. To avoid the difficulties of deriving accurate mathematical models, data-driven methods were introduced for the hydraulic excavator control [7]–[10]. Reinforcement learning (RL) approaches were presented in [7], [8], where the dynamics were approximated by a single large MLP. However, the large number of trainable parameters, which arises from importing the data history as an input of the MLP to handle the delays, substantially slows down the learning speed (e.g., 57476 parameters leading to 10 hours of training with 0.72 million data for [7] as compared to 9160 parameters and 2 hours with 2.6 million data for our plant model). Further, they were presented only in slow operation speeds to commercialize (e.g., average speed of 10 $[\mathrm{cm/s}]$ to 20 $[\mathrm{cm/s}]$ for a 12-ton class excavator), limiting the practical usefulness of the control. On the other hand, an RNN was employed to learn the controller of hydraulic excavators online in [9], [10]. Their performances, however, exhibited rather large tracking errors (e.g., RMSE greater than 1 $[\mathrm{m}]$) in digging operations. The works on data-driven methods did not consider soil interactions [7], [8], [10], and the dead-zone compensation was simply defined by constant control input offsets [8], [10], further limiting them from precision motion control.

For a single hydraulic actuator, a data-driven force control was proposed in [11], where the controller was configured as an MLP. The force controller network was fed with a large dimensional history of the actuator position and force, which again put a strain on the training, and the dead-zone compensation was not considered in the control learning. The dead-zone compensation was studied in [12], [13] with a trainable tailored map, but they could not represent the state-dependent nature of the dead-zones. In [14], a soil interaction model of the excavator was suggested without an examination of the hydraulics and various soil properties for the industrial applications. In contrast to these previous results, our proposed framework can address the complex hydraulic excavator dynamics including the input delays and the state-dependent dead-zones without especially increasing the network size due to the modular structure, while fully considering the interaction with the soil. We also believe that our proposed framework would be advantageous for other hydraulically-actuated robotic systems with multiple actuators and complex environmental interactions.

The rest of the paper is organized as follows. Section II describes the autonomous hydraulic excavator adopted for the experimental validation. Section III introduces the modular design for the proposed data-driven model inversion. The entire offline process that derives data-driven model inversion is depicted in Section IV. Experimental results are presented in Section V, and then Section VI concludes the paper.

II. System Description

This work employs Doosan DX380LC, an industrial hydraulic excavator, to validate our data-driven control strategy. The excavator is customized using sensors to measure states and environmental impacts, as shown in Fig. 1. Inertial measurement unit (IMU) sensors are attached to the boom, arm, and bucket links to estimate the joint configuration. Swing angle is also measurable, but we only consider the motion within the sagittal plane as visualized in Fig. 2 because the swing action is not involved in the excavation. Apart from the joint configuration, hydraulic pressure sensors are located in pumps and cylinders to consider the hydraulic behavior. The pumps and the cylinders are connected through the MCV as detailed in [3], which consists of spool valves that distribute the pump flow rate to generate the cylinder velocity (i.e., the joint angular rate). The spool positions are controlled by electronic proportional pressure reducing (EPPR) valves commanded by the joystick signal. The joystick signal also affects the pump control provided by the manufacturer, which implies that the joystick signal must be regarded as the control input of our industrial excavator. Meanwhile, the soil interactions are evaluated using the momentum-based wrench estimator [15] since we cannot attach a force/torque sensor to the bucket joint due to reliability and cost concerns. A LiDAR sensor scans the point cloud data (PCD) of the terrain for the reference trajectory planning. The communication is via controller area network (CAN), where the sensing and control frequency is set to 100 $[\mathrm{Hz}]$.

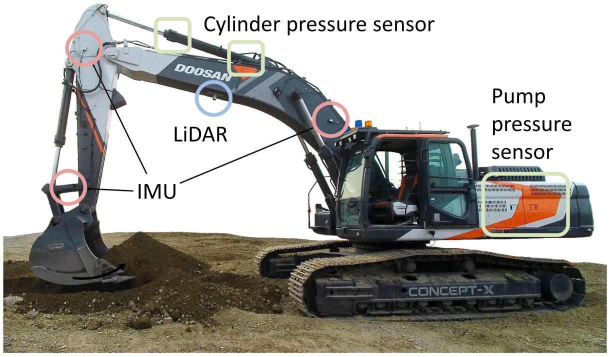

Fig. 1. Doosan DX380LC, a commercial 38-ton class industrial hydraulic excavator. The excavator is customized with IMUs, pressure sensors, and a Velodyne Puck VLP-16 LiDAR. The IMUs are attached to each link, and the pressure sensors are located in all pumps and cylinders.

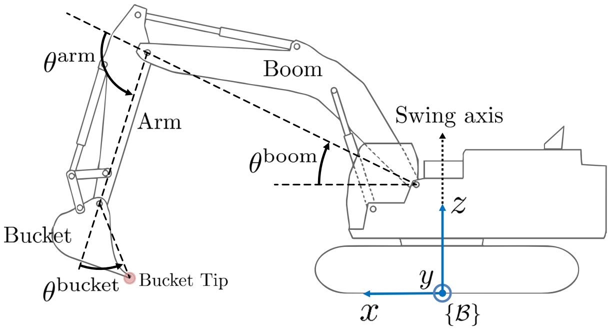

Fig. 2. Joint configuration of the excavator. The bucket tip position is observed in the frame $\{\mathcal{B}\}$ attached to the base of the excavator.

III. Designing Data-Driven Model Inversion

This section introduces the concept of the data-driven model inversion illustrated in Fig. 3. First, we propose a data-driven, physics-inspired, and easy-to-control model with a modular structure that provides an approximate of the excavator dynamics. The data-driven model can cope with the distinct features of the excavator dynamics, including the input delays, the state-dependent dead-zones, and the soil interactions. Then, the inversion control of the data-driven model is configured to compensate for the excavator dynamics.

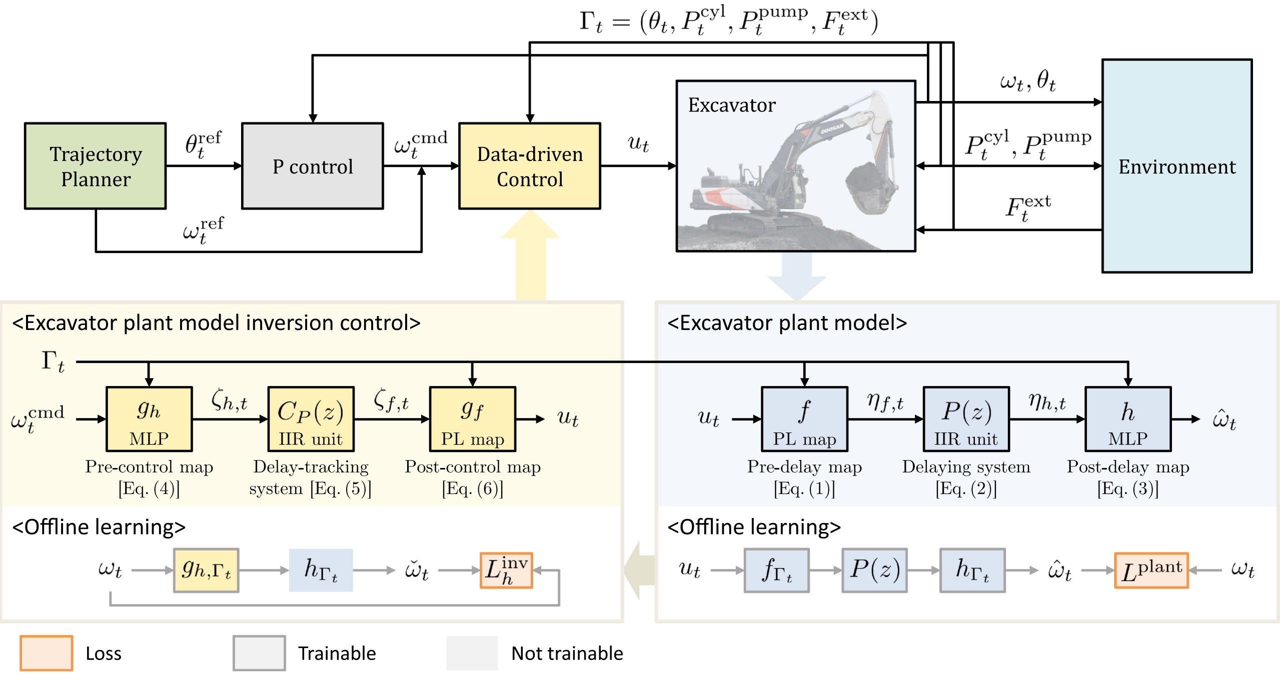

Fig. 3. Framework of the data-driven model inversion and its offline learning schema. First, the excavator plant model (right) is proposed to approximate the excavator dynamics. Then, the inversion control (left) of the excavator plant model, constructed in a modular manner, compensates for the excavator dynamics including the input delays, the state-dependent dead-zones, and the soil interactions.

A. Excavator Plant Model

Assuming that the time-related behavior (i.e., the spool dynamics and the hydraulic delays) can be approximated by a linear time-invariant (LTI) system, the resulting network, namely the *excavator plant model* shown in the right-hand side of Fig. 3, predicts the joint angular rate by

$$\eta_{f,t} = f_{\Gamma_t}(u_t) \tag{1}$$

$$\mathcal{Z}\{\eta_{h,t}\} = P(z)\mathcal{Z}\{\eta_{f,t}\} \tag{2}$$

$$\hat{\omega}_t = h_{\Gamma_t}(\eta_{h,t}) \tag{3}$$

where $t \in \mathbb{Z}$ is the time step identified by the subscript of a time signal $\star_t := \star(t)$, $\mathcal{Z}\{\star_t\} := \sum_{t=0}^{\infty} \star_t / z^t$ is the z-transform, and $P(z)$ is the delaying system, a stable z-domain $n_h \times n_f$ LTI transfer function matrix which captures the multiple and different delays of the hydraulic excavator. The nonlinear nature of the hydraulic circuit is accommodated in pre-delay map $f_{\Gamma_t} : [-1, 1]^3 \to \mathbb{R}^{n_f}$ and post-delay map $h_{\Gamma_t} : \mathbb{R}^{n_h} \to \mathbb{R}^3$ with a simplified expression of a Γ_t-dependent map $\star_{\Gamma_t}(\cdot) := \star(\Gamma_t, \cdot)$. There are two intermediate variables, the pre-delay state $\eta_{f,t} \in \mathbb{R}^{n_f}$ and the post-delay state $\eta_{h,t} \in \mathbb{R}^{n_h}$, to integrate the LTI system and the nonlinear maps. The control input (i.e., joystick signal) is denoted by $u_t \in [-1, 1]^3$, the joint angular rate and its prediction are denoted by $\omega_t, \hat{\omega}_t \in \mathbb{R}^3$, and the excavator state is denoted by

$$\Gamma_t := (\theta_t, P_t^{\text{cyl}}, P_t^{\text{pump}}, F_t^{\text{ext}}) \in \mathbb{R}^{13}$$

where $\theta_t := (\theta_t^{\text{boom}}, \theta_t^{\text{arm}}, \theta_t^{\text{bucket}}) \in \mathbb{R}^3$ is the joint angle, $P_t^{\text{cyl}} \in \mathbb{R}^6$ is the pressure of head- and rod-side chambers of the cylinders, $P_t^{\text{pump}} \in \mathbb{R}^2$ is the pressure of two pumps that supply the hydraulic fluid, and $F_t^{\text{ext}} \in \mathbb{R}^2$ is the horizontal and vertical external force acting on the bucket tip which captures the soil interactions. Refer to Section IV-A for neural network module architectures and offline learning methods for the proposed excavator plant model. We would like to comment that a state-dependent delaying system is available for (2), but the LTI transfer function matrix $P(z)$ works satisfactorily in our application with and without large soil interactions.

B. Excavator Plant Model Inversion Control

From the command joint angular rate $\omega_t^{\text{cmd}} \in \mathbb{R}^3$, the *excavator plant model inversion control* shown in the left-hand side of Fig. 3 computes the joystick signal as

$$\zeta_{h,t} = g_{h,\Gamma_t}(\omega_t^{\text{cmd}}) \tag{4}$$

$$\mathcal{Z}\{\zeta_{f,t}\} = C_P(z)\mathcal{Z}\{\zeta_{h,t}\} \tag{5}$$

$$u_t = g_{f,\Gamma_t}(\zeta_{f,t}) \tag{6}$$

where $C_P(z)$ is the delay-tracking system, a stable z-domain $n_f \times n_h$ LTI transfer function, $g_{h,\Gamma_t} : \mathbb{R}^3 \to \mathbb{R}^{n_h}$ is the pre-control map, and $g_{f,\Gamma_t} : \mathbb{R}^{n_f} \to [-1, 1]^3$ is the post-control map. The pre-control state $\zeta_{h,t} \in \mathbb{R}^{n_h}$ and the post-control state $\zeta_{f,t} \in \mathbb{R}^{n_f}$ are intermediate variables. The reference (e.g., $n_r = 0$ for the step-reference and $n_r = 1$ for the ramp-reference) tracking condition of the delay-tracking system $C_P(z)$ is written as

$$\lim_{z \to 1} (z - 1)\left(P(z)C_P(z) - I_{n_h}\right)\mathcal{Z}\{t^{n_r}\} = 0_{n_h \times n_h} \tag{7}$$

from the final value theorem, where $I_a \in \mathbb{R}^{a \times a}$ is an identity matrix and $0_{a \times b} \in \mathbb{R}^{a \times b}$ is a zero matrix. Two nonlinear maps g_h, g_f satisfy the pseudo-inverse relation s.t. $\star_{\Gamma_t} \circ g_{\star,\Gamma_t}$ is an identity function on the domain of $g_{\star,\Gamma_t}$ given Γ_t. Note that the exact inverse of the pre- and post-delay maps (i.e., $g_{\star,\Gamma_t} \circ \star_{\Gamma_t}$ is also an identity function) may be out of existence because of many-to-one relations such as the dead-zones. The inversion method for each module is illustrated in Section IV-B. The following Proposition 1 provides the properties of our data-driven inversion control (4), (5), and (6).

Proposition 1: Consider the excavator plant model (1), (2), and (3) under the data-driven inversion control (4), (5), and (6). Assume that a) errors of the model prediction $\delta_{\omega,t} := \hat{\omega}_t - \omega_t \in \mathbb{R}^3$ and the errors of the pseudo-inverse relations $\delta_{h,t} := (h_{\Gamma_t} \circ g_{h,\Gamma_t})(\omega_t^{\mathrm{cmd}}) - \omega_t^{\mathrm{cmd}} \in \mathbb{R}^3, \delta_{f,t} := (f_{\Gamma_t} \circ g_{f,\Gamma_t})(\zeta_{f,t}) - \zeta_{f,t} \in \mathbb{R}^{n_f}$ are bounded; b) the post-delay map h_{Γ_t} is a Lipschitz continuous function; and c) the pre-control map g_{h,Γ_t} is a bounded function. Then if the joint angular rate ω_{t_0} and its command $\omega_{t_0}^{\mathrm{ref}}$ at the initial time step $t_0 \in \mathbb{Z}$ are bounded, the difference between the joint angular rate and its command $\nu_{\omega,t} := \omega_t - \omega_t^{\mathrm{cmd}} \in \mathbb{R}^3$ is bounded $\forall t \geq t_0$.

Proof: The triangle inequality provides two inequalities s.t.

$$\begin{aligned}\|\nu_{\omega,t}\| &\leq \|\delta_{\omega,t}\| + \|\hat{\omega}_t - \omega_t^{\mathrm{cmd}}\| \\ &\leq \|\delta_{\omega,t}\| + \|\delta_{h,t}\| + \|h_{\Gamma_t}(\eta_{h,t}) - h_{\Gamma_t}(\zeta_{h,t})\|\end{aligned}$$

where $\|\delta_{\omega,t}\|$ and $\|\delta_{h,t}\|$ are bounded from the first assumption. For the second inequality, see the definition of the post-delay map (3) and the pseudo-inverse error $\delta_{h,t} = h_{\Gamma_t}(\zeta_{h,t}) - \omega_t^{\mathrm{cmd}}$. From the Lipschitz continuity of h_{Γ_t}, there $\exists L \in \mathbb{R}_{\geq}$ s.t.

$$\|h_{\Gamma_t}(\eta_{h,t}) - h_{\Gamma_t}(\zeta_{h,t})\| \leq L\|\eta_{h,t} - \zeta_{h,t}\|$$

where L is referred to as a Lipschitz constant. The delaying system $P(z)$ and its tracking control $C_P(z)$ is rearranged as

$$\begin{aligned}&\mathcal{Z}\{\eta_{h,t} - \zeta_{h,t}\} \\ &\quad = (P(z)C_P(z) - I_{n_h})\,\mathcal{Z}\{\zeta_{h,t}\} + P(z)\mathcal{Z}\{\delta_{f,t}\}\end{aligned}$$

where $P(z)C_P(z) - I_{n_h}$ is a stable linear system satisfying the reference tracking condition (7). From the bounded-input bounded-output (BIBO) property, the error converges as

$$\begin{aligned}\|\eta_{h,t} - \zeta_{h,t}\| \leq{}& \beta\left(\|\eta_{h,t_0} - \zeta_{h,t_0}\|, t - t_0\right) \\ &+ \gamma_1\left(\sup\nolimits_{t_0 \leq \tau \leq t}\|\zeta_{h,\tau}\|\right) + \gamma_2\left(\sup\nolimits_{t_0 \leq \tau \leq t}\|\delta_{f,\tau}\|\right)\end{aligned}$$

where $\gamma_\star : [0, a) \to [0, \infty)$ is a class $\mathcal{K}$ function (i.e., $\gamma_\star$ is strictly increasing with $\gamma_\star(0) = 0$), and $\beta : [0, a) \times [0, \infty) \to [0, \infty)$ is a class $\mathcal{KL}$ function (i.e., $\beta(r, s)$ for each fixed s belongs to class $\mathcal{K}$ and $\beta(r, s)$ for each fixed r is decreasing with $\lim_{s\to\infty} \beta(r, s) = 0$). The pre-control state $\zeta_{h,\tau}$ and the pseudo-inverse error $\delta_{f,\tau}$ are bounded due to the last and the first assumptions, respectively. Bounded properties of the initial conditions lead $\|\eta_{h,t_0} - \zeta_{h,t_0}\|$ to be bounded, which implies that $\nu_{\omega,t}$ is also bounded. ■

On top of the excavator plant model inversion control, the joint angle P control added to the feedforward reference angular rate enhances the robustness of the entire framework with the command joint angular rate:

$$\omega_t^{\mathrm{cmd}} := \omega_t^{\mathrm{ref}} - Ke_{\theta,t} \in \mathbb{R}^3 \tag{8}$$

where $\omega_t^{\mathrm{ref}} \in \mathbb{R}^3$ is the reference joint angular rate, $\theta_t^{\mathrm{ref}} \in \mathbb{R}$ is the reference joint angle, $e_{\theta,t} := \theta_t - \theta_t^{\mathrm{ref}} \in \mathbb{R}^3$ is the joint angle error, and $K \in \mathbb{R}^{3\times3}$ is the P gain which is a positive-definite matrix. Theorem 1 then concludes the entire control framework. Note that the command joint angular rate (8) can be determined independently of the data-driven inversion control. For instance, a velocity field control or a proportional-integral (PI) control can replace the P control in (8).

Theorem 1: Consider the excavator plant model (1), (2), and (3) under the data-driven inversion control (4), (5), and (6) with the P control (8). Following the assumptions of Proposition 1, the joint angle error $e_{\theta,t}$ is ultimately bounded.

Proof: Let us first consider the following Lyapunov function:

$$V := \frac{1}{2}e_{\theta,t}^T e_{\theta,t}$$

for the error convergence in continuous-time domain. The time derivative of the Lyapunov function yields

$$\dot{V} = e_{\theta,t}^T \dot{e}_{\theta,t} = -e_{\theta,t}^T K e_{\theta,t} + e_{\theta,t}^T \nu_{\omega,t}$$

with $\dot{e}_{\theta,t} + Ke_{\theta,t} = \nu_{\omega,t}$ where $\dot{e}_{\theta,t} = e_{\omega,t} := \omega_t - \omega_t^{\mathrm{ref}} \in \mathbb{R}^3$ is the joint angular rate error. From the inequality $\dot{V} \leq -\lambda_{\min}(K)\|e_{\theta,t}\|^2 + \|e_{\theta,t}\|\|\nu_{\omega,t}\|$ with the minimum eigenvalue operator $\lambda_{\min}(\cdot)$, the joint angle error is ultimately bounded by a closed ball of radius $\|\nu_{\omega,t}\|/\lambda_{\min}(K)$ where $\|\nu_{\omega,t}\|$ is bounded by Proposition 1. ■

In Proposition 1, the first assumption stems from reliable learning performances, and the second assumption is based on the continuous and bounded dynamic behavior of the excavator. The last assumption can be enforced by choosing a bounded output activation, such as a hyperbolic tangent, an arc-tangent, or a logistic function, for the pre-control map g_h. Theorem 1 theoretically establishes the robustness of the entire control system, which is not provided in other data-driven controls of hydraulic excavators (e.g., [7]–[10]).

IV. Learning Data-Driven Model Inversion

As schematized in Fig. 3, constructing the data-driven model inversion consists of two steps. The first step is to learn the excavator plant model, made up of the delaying system $P(z)$ and the pre- and post-delay maps f, h; and the second step is to obtain the inversion of each component to constitute the excavator plant model inversion control. The learning steps are detailed in the following Section IV-A and IV-B.

For the offline learning process, we assemble the measurements of Doosan DX380LC to capture complex nonlinear dynamics and soil interactions. The data is collected from autonomous digging/grading operations (with various depths and bucket speeds) and sinusoidal joystick signals (at frequencies 0.25 [Hz] to 0.5 [Hz] and amplitudes 0.3 to 0.5) near the initial and final configurations of the operations. The reference path of the autonomous operation is obtained by length-scaling the nominal bucket configuration extracted from the pattern of human experts [15]. The reference joint angle is computed by inverse kinematics, and the trajectory is time-scaled by the bang-bang approach on joint angular rate considering the hardware limits (e.g., the workspace of the excavator, a rough range of the joint angular rate, and the maximum excavation volume). For the control during the data collection, we employ the manufacturer-provided control and the proposed control trained with a small amount of data. In this work, the main focus is digging/grading tasks, but we believe that the proposed framework can be easily extended to the entire workspace by collecting sufficient data.

We use the data of 2.6 million time steps at a frequency of 100 [Hz] (i.e., 7.2 hours of data) to train the controller. The data is randomly split into training, validation, and test sets at a ratio of 80:15:5. Using the data sets, the offline learning is performed on a computer with an AMD Ryzen 5 3600X 3.8 [GHz] CPU, a 16 [GB] RAM, and an NVIDIA GeForce GTX 1660 Ti GPU. Note that the proposed controller requires at least 1.2 million time steps (i.e., 3.3 hours) of data to obtain good enough performance under the nominal operating condition. However, we include data as extensively as possible on various operating conditions (e.g., soil properties and weather conditions) to address the diverse circumstances of the machines commercialized by the manufacturer.

A. Learning Excavator Plant Model

The first step, a supervised learning of the excavator plant model, exploits the following loss function:

$$L^{\mathrm{plant}} := \|\hat{\omega}_t - \omega_t\|^2$$

where $P(z)$, f, and h are all trainable. The preexisting neural network architectures, however, cannot effectively address the unique properties of the excavator dynamics. For this reason, we propose new neural network modules: an IIR unit for the delaying system $P(z)$ and a monotonically non-decreasing PL map for the pre-delay map f.

Infinite Impulse Response Unit: The delaying system $P(z)$ is a multi-input multi-output (MIMO) transfer function configured as a matrix of single-input single-output (SISO) transfer functions. To construct a neural network for the transfer function learning, let us first consider a z-domain SISO LTI transfer function written as

$$H(z) := \frac{b_0 + b_1 z^{-1} + \cdots + b_{n_b} z^{-n_b}}{a_0 + a_1 z^{-1} + \cdots + a_{n_a} z^{-n_a}}$$

where $b_0, b_1, \ldots, b_{n_b} \in \mathbb{R}$ and $a_0, a_1, \ldots, a_{n_a} \in \mathbb{R}$ are constant coefficients with $a_0 \neq 0$. The transfer function is equivalent to a recursive filter in t-domain, described in terms of the difference equation $y_t = (\sum_{i=0}^{n_b} b_i x_{t-i} - \sum_{i=1}^{n_a} a_i y_{t-i})/a_0$ where $x_t \in \mathbb{R}$ is the input signal and $y_t \in \mathbb{R}$ is the output signal. The difference equation then can be rearranged to

$$y_t = \sum_{i=1}^{n_b} \bar{b}_i (x_{t-i} - x_t) - \sum_{i=1}^{n_a} \bar{a}_i (y_{t-i} - DC_H x_t) + DC_H x_t$$

where $\bar{b}_i := b_i/a_0 \in \mathbb{R}\ \forall i \in \{1, 2, \ldots, n_b\}$ and $\bar{a}_i := a_i/a_0 \in \mathbb{R}\ \forall i \in \{1, 2, \ldots, n_a\}$ are normalized coefficients and $DC_H := H(1) \in \mathbb{R}$ is the low-frequency (DC) gain. Now the IIR unit can be written as an $n_h \times n_f$ matrix s.t.

$$P(z) := \Big[P_{jk}(z)\Big]_{j \in \{1,2,\ldots,n_h\} \text{ and } k \in \{1,2,\ldots,n_f\}}$$

where $P_{jk}(z)\ \forall j, k$ is a SISO transfer function whose orders of the numerator and the denominator are denoted by $n_b^{jk}, n_a^{jk} \in \mathbb{Z}_{\geq}$ and normalized coefficients are denoted by $\bar{b}_i^{jk} \in \mathbb{R}\ \forall i \in \{1, 2, \ldots, n_b^{jk}\}$ and $\bar{a}_i^{jk} \in \mathbb{R}\ \forall i \in \{1, 2, \ldots, n_a^{jk}\}$. The IIR unit belongs to the recurrent neural network family with the given network size $n_h, n_f, n_b^{jk}, n_a^{jk}$ and trainable variables $\bar{b}_i^{jk}, \bar{a}_i^{jk}\ \forall i, j, k$. The DC gain can also be trainable, but here, we choose $DC_P := P(1)$ as a $n_h \times n_f$ matrix with ones on the main diagonal and zeros on the off-diagonal so that $\operatorname{rank} P(1) = \min(n_h, n_f)$. Any $n_h \times n_f$ transfer function matrix whose DC gain rank is $\min(n_h, n_f)$ can be transformed into the IIR unit with row and column matrix operations.

Piecewise Linear Map: The post-control map g_f must deal with jump discontinuities or large slopes to compensate the dead-zones. However, the compensation map is not well trainable with a vanilla MLP because (ω_t, u_t) pairs have one-to-many relations in the dead-zone intervals. For this reason, the pre-delay map f learning is conducted with a monotonically non-decreasing n-segment PL map $\mathrm{PL}_{(X,Y)} : [X_0, X_n] \to [Y_0, Y_n]$ s.t.

$$\mathrm{PL}_{(X,Y)}(x) := \begin{cases} ((Y_i - Y_{i-1})x + X_i Y_{i-1} - X_{i-1} Y_i) \,/\, (X_i - X_{i-1}) \\ \qquad\qquad \text{if } x \in [X_{i-1}, X_i)\ \forall i \in \{1, 2, \ldots, n\} \\ Y_n \qquad\qquad \text{if } x = X_n \end{cases}$$

where $X_i, Y_i \in \mathbb{R}\ \forall i \in \{0, 1, \ldots, n\}$ are breakpoints of the map and $X := \{X_i\}_{i=0}^n, Y := \{Y_i\}_{i=0}^n$ are the non-decreasing sequences of the breakpoints. The PL map is continuous at $x = X_i$ with $\mathrm{PL}_{(X,Y)}(X_i) = Y_i$ if $X_{i-1} < X_i$, while the map can represent the jump discontinuity at $x = X_i$ if $X_{i-1} = X_i$. To apply distinct PL maps to the boom, arm, and bucket joystick signals, a tuple of multiple PL maps is expressed as $(y_1, y_2, \ldots, y_m) := \mathrm{PL}_{(\mathbf{X},\mathbf{Y})}(x_1, x_2, \ldots, x_m)$ where $X_i^k, Y_i^k \in \mathbb{R}\ \forall i \in \{0, 1, \ldots, n^k\}$ are the breakpoints of the n^k-segment k-th PL map $\forall k \in \{1, 2, \ldots, m\}$. Lists of the breakpoints are denoted by $\mathbf{X} := \{X^k := \{X_i^k\}_{i=0}^{n^k}\}_{k=1}^m$ and $\mathbf{Y} := \{Y^k := \{Y_i^k\}_{i=0}^{n^k}\}_{k=1}^m$. Here, we choose the boundary breakpoints as $X_0^k = Y_0^k = -1$ and $X_{n^k}^k = Y_{n^k}^k = 1\ \forall k$, so that $\mathrm{PL}_{(\mathbf{X},\mathbf{Y})} : [-1, 1]^m \to [-1, 1]^m$. Note that the pseudo-inverse of the PL map is defined as $\mathrm{PL}_{(\mathbf{X},\mathbf{Y})}^{+} = \mathrm{PL}_{(\mathbf{Y},\mathbf{X})}$ owing to its monotonicity. The non-decreasing sequences of an interval $[a, b]$ can be trained with the custom activation function $\sigma_{[a,b]} : \mathbb{R}^n \to [a, b]^{n+1}$ written as $\sigma_{[a,b]}(c) = d$ s.t.

$$d_i := a + (b - a) \frac{\sum_{j=1}^{i} \exp(c_j)}{\sum_{j=1}^{n} \exp(c_j)}\ \forall i \in \{1, 2, \ldots, n\}$$

where $c := (c_1, c_2, \ldots, c_n) \in \mathbb{R}^n$ is the activation input and $d := (a, d_1, \ldots, d_n) \in [a, b]^{n+1}$ is the partition of the interval with $a \leq d_1 \leq d_2 \leq \cdots \leq d_n = b$. The activation function can also be extended to a two-dimensional input (e.g., $\sigma_{[a,b]} : \mathbb{R}^{n(\times 2m)} \to [a, b]^{(n+1)\times 2m}$ for m distinct n-segment PL maps) by applying the function to every column of the input.

Employing the proposed neural network modules, the delaying system $P(z)$ is configured as a 3×3 IIR unit whose order is chosen by $n_b^{jk} = n_a^{jk} = 3$ if $j = k$ and $n_b^{jk} = n_a^{jk} = 0$ if $j \neq k\ \forall j, k$ (i.e., 3-rd order transfer functions on the diagonal and zeros on the off-diagonal). This is based on the assumption that the pre- and post-delay maps f, h can adjust the input coupling. Fig. 4 shows the characteristics of the delaying system trained with the pre- and post-delay maps. Here, we would like to mention that the trained controller without the IIR unit causes fatal oscillations in the real machine.

The pre-delay map f is replaced with PL maps as $f_{\Gamma_t} := \mathrm{PL}_{\Phi_{f,t}} : [-1, 1]^3 \to [-1, 1]^3$. Breakpoints of the PL map are assumed to depend on Γ_t by an auxiliary network $\Phi_{f,t} :=$

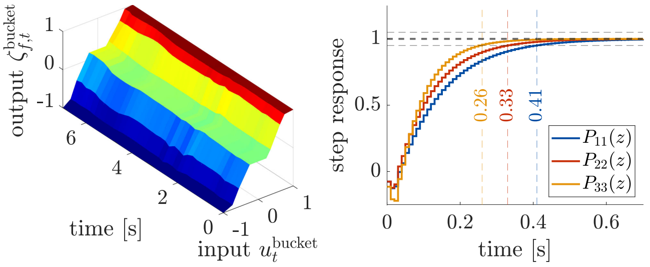

Fig. 4. Visualization of the pre-delay map f_{Γ_t} (left) and the delaying system $P(z)$ (right). The PL map landscape visualizes the state-dependent pre-delay map for the bucket joystick signal during a trial of grading operation. The IIR step response shows the input delays of the boom, arm, and bucket, with settling times to within 5% are $(0.41, 0.33, 0.26)$ [s].

$\phi_f(\Gamma_t) \in [-1,1]^{9\times 6}$ which consists of a hidden layer of 64 ReLU nodes and an output layer of 8 × 6 nodes with the custom output activation function $\sigma_{[-1,1]} : \mathbb{R}^{8\times 6} \to [-1,1]^{9\times 6}$, for the three distinct 8-segment PL maps. The trained PL map is visualized in Fig. 4. Notice that the PL map enables the dead-zone compensation, shown in Fig. 5 as jumps of joystick signals. The pre-delay map can be further customized with constraints or other designs (e.g., applying a constraint to pass through the origin, training boundary breakpoints, or combining the PL map with an additional MLP), but this is not necessary in our case.

To address the remaining nonlinear properties and couplings, the post-delay map h is expressed as an MLP, with a single hidden layer of 256 nodes with a ReLU activation and a linear output layer. Training the excavator plant model (1), (2), and (3) takes only around 2 hours to converge using the Adam optimizer. Fig. 5 visualizes two examples of the prediction, while the prediction RMSE of the test data set is $(0.51, 0.66, 1.16)$ [deg/s] for each boom, arm, and bucket joint angular rate, which is small enough to justify the presented neural network architecture.

B. Learning Excavator Plant Model Inversion Control

The second step configures the modular inversion of the data-driven excavator plant model. The delay-tracking system $C_P(z)$, the inversion of the delaying system $P(z)$, is a 3×3 MIMO LTI transfer function. Although we can train $C_P(z)$ with another IIR unit, we analytically compose the delay-tracking system since obtaining the tracking control for the diagonal transfer function matrix is relatively simple. A stable and exact inverse of the delaying system (i.e., $P^{-1}(z)$) does not exist because the trained delaying system has unstable zeros characterized by inverse responses as shown in Fig. 4. Thus, the delay-tracking system is constructed to meet the reference tracking condition (7) $\forall n_r \in \{0,1\}$ where the poles are empirically optimized to 0.82 with multiplicity 2. We choose the delay-tracking system with the minimum numerator order, which is a proper transfer function matrix. The post-control map g_f, the pseudo-inverse of the pre-delay map f, does not require any offline learning process as the post-control map can be easily computed as $g_{f,\Gamma_t} := \text{PL}^{+}_{\Phi_{f,t}}$.

On the other hand, the pre-control map g_h (i.e., the pseudo-inverse of the post-delay map h) cannot be analytically obtained since the post-delay map is an MLP. For the offline learning of the pre-control map, a distal learning approach [16] is introduced. The loss function is defined as

$$L_h^{\text{inv}} := \|\breve{\omega}_t - \omega_t\|^2$$

where $\breve{\omega}_t := (h_{\Gamma_t} \circ g_{h,\Gamma_t})(\omega_t) \in \mathbb{R}^3$ to realize the pseudo-inverse relation. The MLP network of the pre-control map g_h has a hidden layer of 256 nodes with a ReLU activation and an output layer with a hyperbolic tangent activation. The pre-control map can also be customized with additional PL maps to tolerate large slopes or jump discontinuities. After some trials, however, we found that an MLP is enough for the pre-control map g_h, implying that the dead-zones are all captured in the pre-delay map f while training the plant model. The pre-control map training takes less than 5 minutes, owing to the modular inversion method. Reconstruction of the joystick signal is compared to the recorded signal in Fig. 5.

V. Experiment

The proposed control framework is verified with the commercial 38-ton class excavator Doosan DX380LC in digging and grading tasks. The reference trajectories for both operations are generated using the same planning algorithm as described in Section IV. The control frequency is 100 [Hz], though the inversion control can be easily implemented with higher frequencies. The P gain is chosen as $K = 1.5I_3$ to determine the model inversion input. We implemented PI feedback as well, yet, found that only P control suffices as the control error is already fairly small without the I feedback.

The bucket tip position $p_t := (p_{x,t}, p_{z,t}) \in \mathbb{R}^2$ is calculated to evaluate the control performance, where $p_{x,t}, p_{z,t} \in \mathbb{R}$ are the horizontal and vertical tip positions as shown in Fig. 2. The path following error is denoted by $e_{p,t}^{\text{path}} := \min_{t_0 \le \tau \le t_f} \|p_t - p_\tau^{\text{ref}}\| \in \mathbb{R}$, which indicates the error of the excavated ground geometry. The trajectory error, or the bucket tip position error, is written as $e_{p,t}^{\text{traj}} := \|p_t - p_t^{\text{ref}}\| \in \mathbb{R}$. Here, we calculate the RMSE from one second after the initial time. This is to assess the performance/precision of our proposed control in the steady-state as typically done in control literature. Note from Figs. 6, 7, and the supplementary video that, at the initial time, we have a non-zero initial error due to our using of the (less accurate) manufacturer-provided PI control until then. See to it also that our proposed control well-behaves (e.g., with no sudden dipping) during this transition. This initial performance can further be improved by turning on our proposed data-driven controller before the initial operation or by re-planning the path using the measured configuration as the initial condition (i.e., ensuring the controller in its steady-state). The control performance is compared to the manufacturer-provided PI control, which determines the joystick signal by a (ω_t, u_t) pair look-up table with a joint angle PI feedback and an angular rate feedforward. The manufacturer manually fine-tuned the control gain and the look-up table using the air digging data (i.e., data without soil interactions).

Digging: The digging operation is the removal of soil from the current terrain to achieve the target ground shape. Due to the excavation capacity limit, multiple digging operations may be required to reach the final target ground geometry. Fig. 6

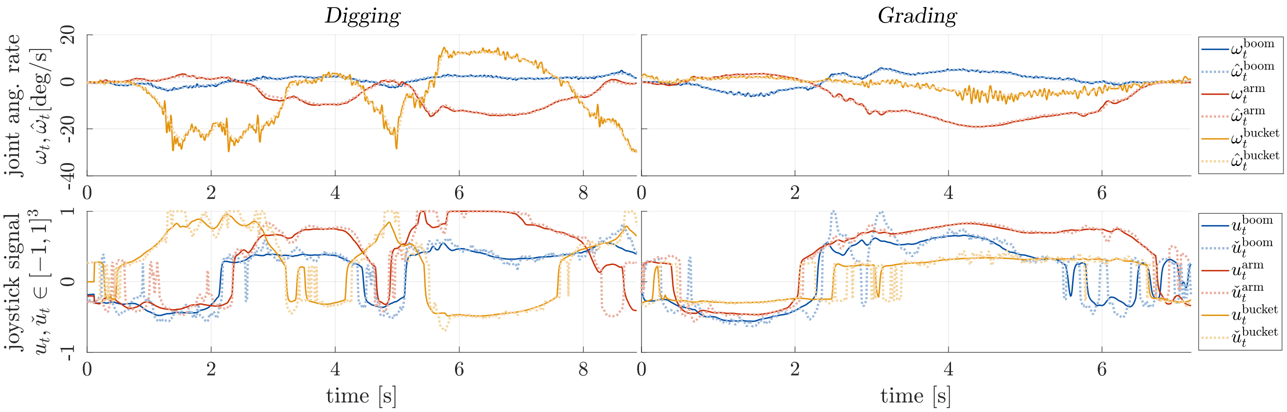

Fig. 5. Prediction of the joint angular rate (top) and reconstruction of the joystick signal (bottom) for the digging (left) and grading (right) data within the test set. The predicted joint angular rate $\hat{\omega}_t$ is computed using the excavator plant model given the same joystick signal u_t. The reconstructed joystick signal $\check{u}_t$ is the output of the model inversion taking the measured joint angular rate ω_t as the inversion input. About the discrepancies between the recorded/reconstructed joystick signals, we would like to note that: 1) there is no exact inverse of the dead-zones, that is, different joystick signals in the dead-zones can lead to the same excavator response; and 2) perfect tracking control of the non-minimum phase delays does not exist.

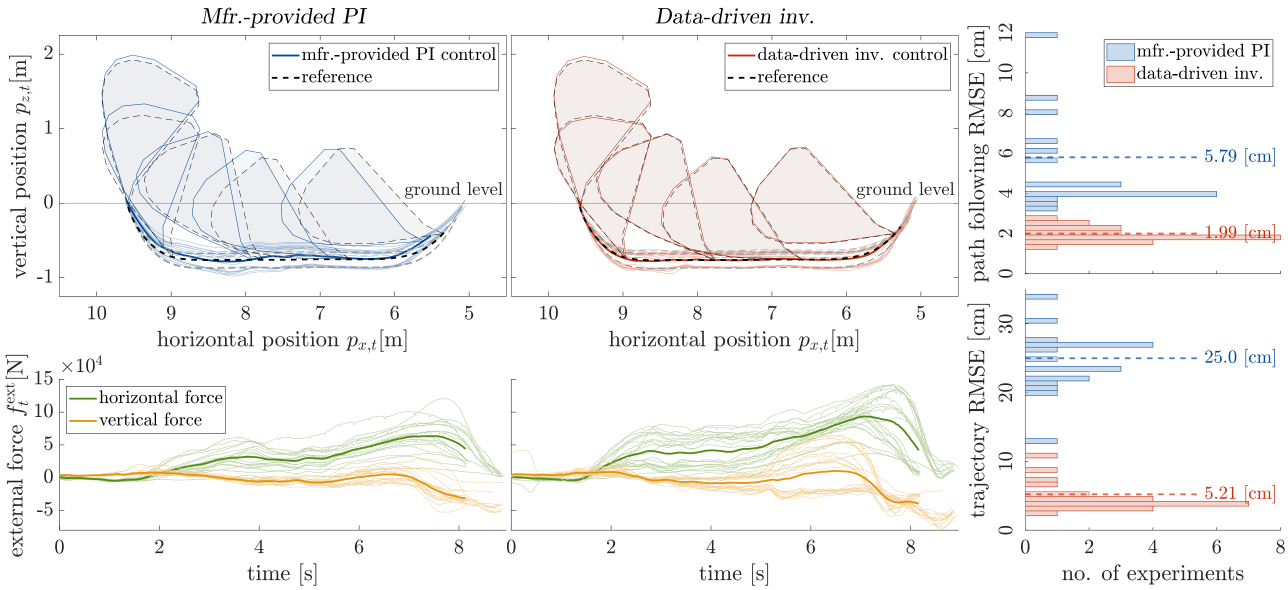

Fig. 6. Bucket tip position (top-left), external force (bottom-left), and performance evaluation (right) during the digging operations. Repeated experiments are visualized as thin lines where the reference trajectories are planned for every repetition considering the current/target terrain and the hardware limit. The bold lines are the averages of the trials, and the shaded bucket plots are the average bucket configuration for every 2 [s]. Histograms of the path following and trajectory RMSEs are also provided for the performance comparison.

visualizes bucket trajectories, soil interactions, and error distributions of repeated experiments on various excavation depths and volumes. The experimental results of the manufacturer-provided PI control have a path following RMSE of 5.79 [cm] and a trajectory RMSE of 25.0 [cm]. The PI control results have an RMS reference bucket tip velocity of 66.2 [cm/s] and an RMS external force of 4.31×10^4 [N]. The proposed control framework outperforms the manufacturer-provided PI control, where a path following RMSE is 1.99 [cm] and a trajectory RMSE is 5.21 [cm] with an RMS reference bucket tip velocity of 66.3 [cm/s] and an RMS external force of 6.41×10^4 [N]. The operation speed and the external force are large enough for industrial applications.

Grading: The grading operation is to level the ground surface after the digging operations, where its experimental results are shown in Fig. 7. The manufacturer-provided PI control has a path following RMSE of 5.28 [cm], trajectory RMSE of 15.1 [cm], an RMS reference bucket tip velocity of 87.2 [cm/s], and an RMS external force of 2.69×10^4 [N]. The excavator plant model inversion control with the P control attains a path following RMSE of 1.83 [cm] and a trajectory RMSE of 3.17 [cm] with an RMS reference bucket tip velocity of 87.3 [cm/s] and an RMS external force of 2.23×10^4 [N]. The errors are evenly small both with and without intense soil interactions since the inversion captures and compensates for the effect of the external force.

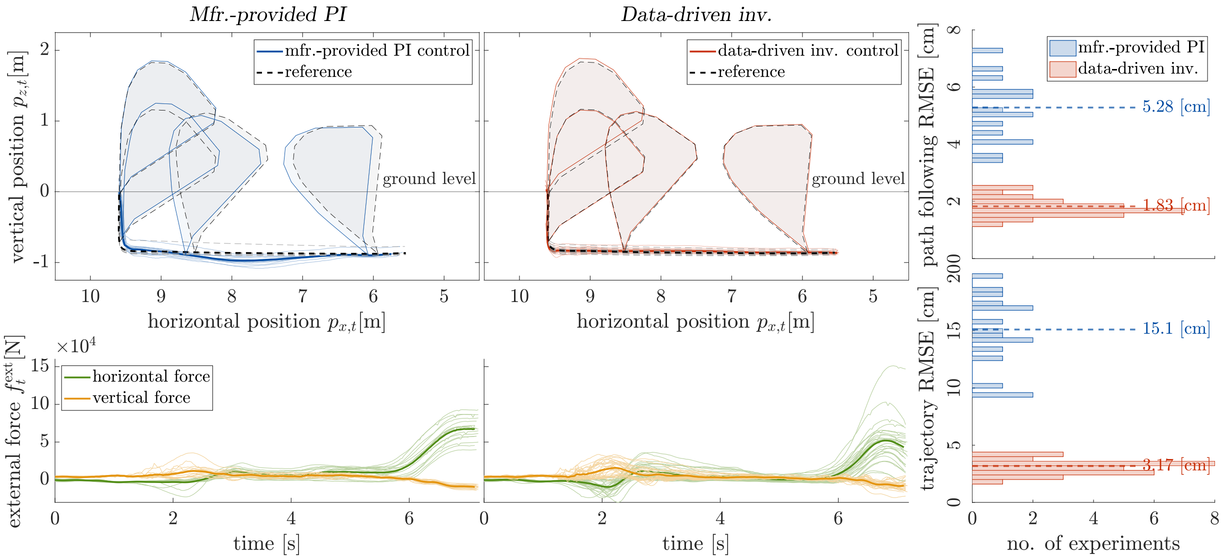

Fig. 7. Experimental results of the repeated grading operations which is to level the ground surface after the digging operations shown in Fig. 6.

VI. Conclusion

This work presents a precision motion control of robotized industrial hydraulic excavators via data-driven model inversion. Considering distinct features that hinder the learning-based control methods (i.e., input delays and dead-zones), we propose a data-driven model with a physics-inspired modular structure to approximate the excavator dynamics. We then derive the inversion of the plant model in a modular manner which considerably promotes the learning speed. To prevent injuries of the machine and surroundings, the model and its inversion are trained offline in a supervised fashion using the measurements of the Doosan DX380LC, a 38-ton class industrial hydraulic excavator. Our proposed control framework is composed of the data-driven model inversion control, which compensates for the excavator dynamics, and a P control that computes the model inversion input and enhances the robustness. The stability and robustness of the control framework are theoretically proven, and experimental results are presented in comparison with the manufacturer-provided PI control. The proposed control framework significantly outperforms the PI control and shows a precise control performance (i.e., path following RMSE under 2 [cm]) even in the presence of intense soil interactions.

Some possible future research directions include: 1) generalization of the excavator plant model using the state-dependent delaying system; 2) incorporation of the expert-emulating planning [15]; 3) implementation of the over-the-air programming to effectively collect the measurements and update the control; 4) rigorous comparison with other control strategies (e.g., [17]); and 5) application of our framework to other systems with delays and dead-zones.

References

[1] L. Zhang *et al.*, “An autonomous excavator system for material loading tasks,” *Sci. Robot.*, Amer. Assoc. Advancement of Sci., vol. 6, no. 55, 2021, Art. no. eabc3164.

[2] 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 Trans. Mechatronics*, vol. 22, no. 2, pp. 669–680, Apr. 2017.

[3] K. Kim, M. Kim, D. Kim, and D. J. Lee, “Modeling and velocity-field control of autonomous excavator with main control valve,” *Automatica*, vol. 104, pp. 67–81, 2019.

[4] S. Tafazoli, S. E. Salcudean, K. Hashtrudi-Zaad, and P. D. Lawrence, “Impedance control of a teleoperated excavator,” *IEEE Trans. Control Syst. Technol.*, vol. 10, no. 3, pp. 355–367, 2002.

[5] Q. Ha, M. Santos, Q. Nguyen, D. Rye, and H. Durrant-Whyte, “Robotic excavation in construction automation,” *IEEE Robot. Automat. Mag.*, vol. 9, no. 1, pp. 20–28, May 2002.

[6] P. Chang and S. Lee, “A straight-line motion tracking control of hydraulic excavator system,” *Mechatronics*, vol. 12, no. 1, pp. 119–138, 2002.

[7] P. Egli and M. Hutter, “Towards RL-based hydraulic excavator automation,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 2692–2697.

[8] P. Egli and M. Hutter, “A general approach for the automation of hydraulic excavator arms using reinforcement learning,” *IEEE Robot. Automat. Lett.*, early access. 2021, doi: 10.3929/ethz-b-000487440.

[9] J. Park, D. Cho, S. Kim, Y. Kim, P. Kim, and H. Kim, “Utilizing online learning based on echo-state networks for the control of a hydraulic excavator,” *Mechatronics*, vol. 24, no. 8, pp. 986–1000, 2014.

[10] J. Park, B. Lee, S. Kang, P. Kim, and H. Kim, “Online learning control of hydraulic excavators based on echo-state networks,” *IEEE Trans. Automat. Sci. Eng.*, vol. 14, no. 1, pp. 249–259, Jan. 2017.

[11] S. Kim, B. Cho, S. Shin, J. Oh, J. Hwangbo, and H. Park, “Force control of a hydraulic actuator with a neural network inverse model,” *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 2814–2821, Apr. 2021.

[12] R. R. Selmic and F. L. Lewis, “Deadzone compensation in motion control systems using neural networks,” *IEEE Trans. Autom. Control*, vol. 45, no. 4, pp. 602–613, Apr. 2000.

[13] 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*, New York, NY, USA: American Society of Mechanical Engineers, 2018.

[14] F. E. Sotiropoulos and H. H. Asada, “Dynamic modeling of bucket-soil interactions using Koopman-DFL lifting linearization for model predictive contouring control of autonomous excavators,” *IEEE Robot. Automat. Lett.*, vol. 7, no. 1, pp. 151–158, Jan. 2022.

[15] B. Son, C. Kim, C. Kim, and D. J. Lee, “Expert-emulating excavation trajectory planning for autonomous robotic industrial excavator,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 2656–2662.

[16] M. I. Jordan and D. E. Rumelhart, “Forward models: Supervised learning with a distal teacher,” *Cogn. Sci.*, vol. 16, no. 3, pp. 307–354, 1992.

[17] M. Neumann-Brosig, A. Marco, D. Schwarzmann, and S. Trimpe, “Data-efficient autotuning with bayesian optimization: An industrial control study,” *IEEE Trans. Control Syst. Technol.*, vol. 28, no. 3, pp. 730–740, May 2020.