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, we construct a physics-inspired data-driven model with a modular structure to deal with prominent features of the hydraulic excavator dynamics, including input delays and dead-zones. 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 such heavy machines and surroundings. The data-driven model is inverted in a modular fashion which benefits the training speed, and the entire motion control framework consists of the inversion control and the proportional (P) control that determines the input of the model inversion to enhance 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 (RMS) of the path following error under 2 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. Experimental valida-
tion is conducted with a commercial 38-ton class hydraulic excavator for digging and grading tasks, accomplishing a precise control performance (i.e., root-mean-square (RMS) of the path following error under 2 [cm]) even in the presence of intense soil interactions.

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 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., average speed of 10 [cm/s] to 20 [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 in [9], [10]. Their performances, however, exhibited rather large tracking errors (e.g., RMS error (RMSE) greater than 1 [in]) in digging operations. Moreover, 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 actuator with input delays and dead-zones, 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 controller input, state-dependent dead-zones, and the soil interactions. Then, a data-driven model inversion illustrated in Fig. 3. First, we propose a data-driven, state-dependent dead-zones, and the soil interactions. Then, 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.

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
\[ \eta_{f,t} = f_{\Gamma_t}(u_t) \]  
\[ Z\{\eta_{h,t}\} = P(z)Z\{\eta_{f,t}\} \]  
\[ \hat{\omega}_t = h_t(\eta_{h,t}) \]   
where \( t \in \mathbb{Z} \) is the time step identified by the subscript of a time signal \( *_t := *t \), \( Z\{*_t\} := \sum_{t=0}^{\infty}*_t/z^t \) 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 circuitry is accommodated in pre-delay map \( f_{\Gamma_t} : [-1,1]^3 \rightarrow \mathbb{R}^{n_f} \) and post-delay map \( h_t : \mathbb{R}^{n_h} \rightarrow \mathbb{R}^3 \) with a simplified expression of a \( \Gamma_t \)-dependent map \( *_{\Gamma_t}(\cdot) := *(\Gamma_t,\cdot) \). 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 \( \hat{\omega}_t, \hat{\omega}_t \in \mathbb{R}^3 \), and the excavator state is denoted by
\[ \Gamma_t := (\theta_t, P_t^{cyl}, P_t^{pump}, F_t^{ext}) \in \mathbb{R}^{13} \]
where \( \theta_t := (\theta_{boom}, \theta_{arm}, \theta_{bucket}) \in \mathbb{R}^3 \) is the joint angle, \( P_t^{cyl} \in \mathbb{R}^6 \) is the pressure of head- and rod-side chambers of the cylinders, \( P_t^{pump} \in \mathbb{R}^2 \) is the pressure of two pumps that supply the hydraulic fluid, and \( F_t^{ext} \in \mathbb{R}^2 \) 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.

B. Excavator Plant Model Inversion Control

From the command joint angular rate \( \omega_t^{cmd} \in \mathbb{R}^3 \), the excavator plant model inversion control shown in the left-hand side of Fig. 3 computes the joystick signal as
\[ \hat{\eta}_{h,t} = g_{h_{\Gamma_t}}(\omega_t^{cmd}) \]  
\[ Z\{\hat{\eta}_{f,t}\} = C_P(z)Z\{\hat{\eta}_{h,t}\} \]  
\[ u_t = g_{f_{\Gamma_t}}(\hat{\eta}_{f,t}) \]   
where \( C_P(z) \) is the delay-tracking system, a stable \( z \)-domain \( n_f \times n_h \) LTI transfer function, \( g_{h_{\Gamma_t}} : \mathbb{R}^3 \rightarrow \mathbb{R}^{n_h} \) is the pre-control map, and \( g_{f_{\Gamma_t}} : \mathbb{R}^{n_h} \rightarrow [-1,1]^3 \) is the post-control map. The pre-control state \( \hat{\eta}_{h,t} \in \mathbb{R}^{n_h} \) and the post-control state \( \hat{\eta}_{f,t} \in \mathbb{R}^{n_f} \) 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
\[ \lim_{z \rightarrow 1} [z^r (z) (P(z)C_P(z) - I_{n_h}) Z\{t^{r_r}\}] = 0_{n_h \times n_h} \]   
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. \( *_{\Gamma_t} \circ g_{h_{\Gamma_t}} \) is an identity function on the domain of \( g_{h_{\Gamma_t}} \), given \( \Gamma_t \). Note that the exact inverse of the pre- and post-delay maps (i.e., \( g_{h_{\Gamma_t}} \circ *_{\Gamma_t} \) 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 \(\delta_{\omega,t} := \omega_t - \omega_t^* \in \mathbb{R}^3\) and the errors of the pseudo-inverse relations \(\delta_{h,t} := (h_t \circ g_{\nu,t})(c_{\nu,t}) - h_t^* \in \mathbb{R}^3\), \(\delta_{f,t} := (f_t \circ g_{\nu,t})(c_{\nu,t}) - f_t^* \in \mathbb{R}^3\) are bounded; b) the post-delay map \(h_t\) is a Lipschitz continuous function; and c) the pre-control map \(g_{\nu,t}\) is a bounded function. Then if the joint angular rate \(\omega_{t_0}\) and its command \(\omega_{t_0}^\text{cmd}\) at the initial time step \(t_0 \in \mathbb{Z}\) are bounded, the difference between the joint angle \(\omega_t\) and its command \(\omega_{t}^\text{cmd}\) in \(\mathbb{R}^3\) is bounded \(\forall t \geq t_0\).

Proof. The triangle inequality provides two inequalities s.t.
\[
\|\omega_{t} - \omega_{t}^\text{cmd}\| \
\leq \|\omega_{t} - \omega_{t}^*\| + \|\omega_{t}^* - \omega_{t}^\text{cmd}\|
\]
where \(\delta_{\omega,t}\) and \(\|\delta_{\omega,t}\|\) are bounded as the first assumption. For the second inequality, see the definition of the post-delay map \((3)\) and the pseudo-inverse error \(\delta_{\omega,t} = h_t^*(\zeta_{\omega,t}) - \omega_{t}^\text{cmd}\). From the Lipschitz continuity of \(h_t^*\), there \(\exists \lambda \in \mathbb{R} \geq s.t.
\]
\[
\|h_t^*(\eta_{\omega,t}) - h_t^*(\zeta_{\omega,t})\| \leq \lambda \|\eta_{\omega,t} - \zeta_{\omega,t}\|
\]
where \(L\) is referred to as a Lipschitz constant. The delaying system \(P(z)\) and its tracking control \(C_P(z)\) is rearranged as
\[
\mathcal{Z}\{\eta_{\omega,t} - \zeta_{\omega,t}\} = (P(z)C_P(z) - I_{\nu_0})\mathcal{Z}\{\zeta_{\omega,t}\} + (P(z)\delta_{f,t})
\]
where \(P(z)C_P(z) - I_{\nu_0}\) is a stable linear system satisfying the reference tracking condition \((7)\). From the bounded-input bounded-output (BIBO) property, the error converges as
\[
\|\eta_{\omega,t} - \zeta_{\omega,t}\| \leq \beta(\|\eta_{\omega,t} - \zeta_{\omega,t}\|, t \geq t_0) + \gamma_1(\sup_{t_0 \leq \tau \leq t} \|\zeta_{\omega,t}\|) + \gamma_2(\sup_{t_0 \leq \tau \leq t} \|\delta_{f,t}\|)
\]
where \(\gamma = [0, a) \rightarrow [0, \infty)\) is a class \(K\) function (i.e., \(\gamma_\ast\) is strictly increasing with \(\gamma_\ast(0) = 0\), and \(\beta : [0, a) \times [0, \infty) \rightarrow [0, \infty)\) is a class \(K\mathcal{L}\) function (i.e., \(\beta(r, s)\) for each fixed \(s\) belongs to class \(K\) and \(\beta(r, s)\) for each fixed \(r\) is decreasing with \(\lim_{s \rightarrow \infty} \beta(r, s) = 0\)). The pre-control state \(\zeta_{\omega,t}\) and the pseudo-inverse error \(\delta_{f,t}\) are bounded due to the last and the first assumptions, respectively. Bounded properties of the initial conditions lead \(\|\eta_{t_0} - \zeta_{t_0}\|\) to be bounded, which implies that \(\nu_{t_0}\) is also bounded.

On top of the excavator plant model inversion control, the joint angle \(P\) control added to the angular rate feedforward enhances the robustness of the entire framework with the command joint angular rate:
\[
\omega_{t}^\text{cmd} := \omega_{t}^\text{ref} - Ke_{\theta,t} \in \mathbb{R}^3
\]
where \(\omega_{t}^\text{ref}\) is the reference joint angular rate, \(e_{\theta,t} := \theta_t - \theta_t^\text{ref}\) is the joint angular error, and \(K \in \mathbb{R}^{3 \times 3}\) is the \(P\) gain which is a positive-definite matrix. Then Theorem 1 concludes the entire control framework via the data-driven model inversion. The command joint angular rate \((8)\) can be determined independent 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
\[
\dot{V} \leq -\lambda_{\min}(K)\|e_{\theta,t}\|^2 + \|e_{\theta,t}\|\|\nu_{t}\|\] where \(\|\nu_{t}\|\) 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 \(g_{\nu,t}\). 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. To capture the complex nonlinear dynamics and the soil interactions, we assemble the measurements of 2.6 million time steps at a frequency of 100\([Hz]\) (i.e., 7.2 hours of data) on the Doosan DX380LC. The data is collected using the autonomous digging/grading tasks with various depths and bucket speeds. During the autonomous data collection, we use the manufacturer-provided control and the proposed control trained with a small amount of data. In addition, data with sinusoidal joystick signals (at frequencies 0.25\([Hz]\) to 0.5\([Hz]\) and amplitudes 0.3 to 0.5) is also gathered near the initial and final configurations of the digging/grading.

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 on various operating conditions (e.g., soil properties and weather conditions) as extensively as possible to address the possibilities of the machines being commercialized by the manufacturer. 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 a NVIDIA GeForce GTX 1660 Ti GPU.
The difference equation then can be rearranged to

\[ P(z) = \left[ P_{jk}(z) \right]_{j \in \{1, 2, \ldots, n_j \}} \text{and } k \in \{1, 2, \ldots, n_f \} \]

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

\textbf{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 \( (\bar{\omega}_i, n_i) \) pairs have one-to-many relations in the dead-zone intervals. From this reason, the pre-delay map \( f \) learning is conducted with a monotonically non-decreasing \( n \)-segment PL map \( \text{PL}(X, Y) : [X_0, X_n] \to [Y_0, Y_n] \) s.t.

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

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 \( \text{PL}(X, Y)(X_i) = Y_i \) 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 \( (y_1, y_2, \ldots, y_m) := \text{PL}(X, Y)(x_1, x_2, \ldots, x_m) \) where \( X^k, Y^k \in \mathbb{R} \forall i \in \{0, 1, \ldots, n^k\} \) 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 \( X := \{X^k := \{X^k_i\}_{i=0}^{n^k} \}_k=1 \) and \( Y := \{Y^k := \{Y^k_i\}_{i=0}^{n^k} \}_k=1 \). Here, we choose the boundary breakpoints of \( X^k = Y^k = 0 \) and \( n^k = 1 \forall k \), so that \( \text{PL}(X, Y) : [-1, 1]^m \to [-1, 1]^m \). Note that the pseudo-inverse of the PL map is defined as \( \text{PL}^t(X, Y) = \text{PL}(Y, X) \) 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.

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

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 \( a \leq d_1 \leq a_2 \leq \cdots \leq d_n \equiv b \). 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 \( 3 \times 3 \) IIR unit whose order is chosen by \( n_b^j = n_f^j = 3 \) if \( j = k \) and \( n_b^j = n^j = 0 \) if \( j \neq k \). Breakpoints of the PL map are assumed to depend on an auxiliary network \( \Phi_{f,t} := \phi_{f}(\Gamma_t) \in [-1,1]^{30} \), which consists of a hidden layer of 64 ReLU nodes and an output layer of 8 \( \times \) 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. \( \text{[4]} \). Due to the jump discontinuities of the model inversion, the inversion training without the PL map has a 31.9\% greater test loss. 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 an output layer node. Training the excavator plant model \( \text{[1]}, \text{[2]}, \text{[3]} \) takes only around 2 hours to converge using the Adam optimizer. Fig. \( \text{[5]} \) 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 \( 3 \times 3 \) MIMO LTI transfer function. Although we can train the delay-tracking system 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. \( \text{[6]} \). Thus, the delay-tracking system is constructed to meet the reference tracking condition \( \forall n_r \in \{0,1,2,16\} \) 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 \( g_t \), 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 \( g_{f,t} := \text{PL}_{\Phi_{f,t}} \).

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 \( \text{[16]} \) is introduced to enhance the learning performance. The loss function is defined as

\[
L_h^\text{inv} := \| \hat{\omega}_t - \omega_t \|^2
\]

where \( \hat{\omega}_t := (h_{f,t} \circ g_{h,t})(\omega_t) \in \mathbb{R}^3 \) 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. \( \text{[5]} \).
Fig. 6. Bucket tip position (top-left), external force (top-bottom), 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.

Fig. 7. Experimental results of the repeated digging operations. Details are the same as Fig. 6.

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. Remark that we can use other feedback laws such as PI, but the integral feedback does not empirically improve the performance.

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 $e_{p,t} := \min_{0 \leq \tau \leq t_f} \|p_t - p_t^{\text{ref}}\| \in \mathbb{R}$, which indicates the error of the excavated ground geometry. The trajectory error, or the bucket tip position error, is written as $e_{p,t}^{\text{tra}} := \|p_t - p_t^{\text{ref}}\| \in \mathbb{R}$. The RMS of the data is calculated one second after the initial time to evaluate the control performance. The reason is that the bucket is placed in the primary position of the excavation using the manufacturer-provided PI control, resulting in a nonzero initial error. Although the proposed controller works well in transient-states even if the initial error is not zero, it is common to compare the steady-state performance in the control literature. The control performance is compared to the manufacturer-provided PI control, which determines the joystick signal by a $(\omega_t, u_t)$ pair look-up table with a joint angle PI feedback and an angular rate feedforward. Note that
the manufacturer manually fine-tuned the control gain and the look-up table using the air digging data (i.e., data without soil interactions). Automatic tuning methods such as [17] are not implemented because there is no guarantee that the online gain adjustment works well with the dead-zones, the delays, and various soil properties.

**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 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]. Note that the errors are evenly small in both cases with and without intensive soil interactions since the inversion captures and compensates for the effect of the external force.

**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. To prevent injuries of the machine and surroundings, the model is trained offline in a supervised fashion, using the measurements of the Doosan DX380LC, a 38-ton class industrial hydraulic excavator, during its operations. Then, we derive the inversion of the plant model, constructed with inverse networks obtained for each module, where this modular structure considerably promotes the learning speed. 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 linear time-varying 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; and 4) application of our framework to other systems with delays and dead-zones.

**References**


