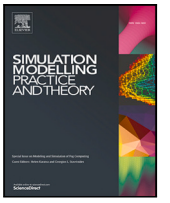
Simulation Modelling Practice and Theory 145 (2025) 103208

Contents lists available at ScienceDirect

Simulation Modelling Practice and Theory

journal homepage: www.elsevier.com/locate/simpat



Physics-informed data-driven modeling of rock motion dynamics in excavation using a high-fidelity simulator

Mohammad Heravi *, Amirmasoud Molaei, Reza Ghabcheloo

Faculty of Engineering and Natural Sciences, Tampere University, Korkeakoulunkatu 7, Tampere, 33720, Finland

ARTICLE INFO

Keywords:
Autonomous excavation
Rock motion dynamics
Data-driven modeling
Physics engine simulation
Predictive model
Physics-informed neural networks

ABSTRACT

In this paper, the problem of predicting the motion of large rocks during excavation is addressed. During excavation, complex interactions are observed among the excavator bucket, rock, and soil, which are not effectively captured using analytical models due to nonlinearities and unknown phenomena. To address this, a physics-informed, data-driven framework is proposed, in which a predictive model of the rock dynamics is learned using data obtained from a high-fidelity physics-based simulator. Specifically, a physics-informed neural network is employed, structured as a multilayer perceptron that receives the state variables and control inputs from a fixed-length temporal window. A kinematic constraint is incorporated into the loss function to enforce physical consistency. The model is trained and evaluated using data from 200 experiments. The effect of the look-back window length is examined, and a window length of two is found to yield the minimum prediction error. The prediction error distributions are statistically evaluated for different soil and rock scenarios, as well as across different prediction horizons (1–20). The model's accuracy is shown to be within the desired threshold.

1. Introduction

Construction is regarded as one of the largest industries worldwide, supporting the development of roads, foundations, vital infrastructures, as well as mining operations [1,2]. In this field, excavators are frequently used as primary earth-moving machinery owing to their versatility and multi-tasking abilities [3]. A standard hydraulic excavator, depicted in Fig. 1, is mainly powered by a hydraulic transmission system and consists of three principal components: the traveling body, swing body, and front manipulator. This manipulator comprises a boom, an arm, and a bucket interconnected by three revolute joints [4]. The versatility of excavators proves particularly crucial for large-scale open-pit mining, where entire rock faces may be blasted, and also on construction sites, where large rocks may be partially embedded among smaller stones and soil [5]. Removing and relocating these sizable rocks—either to dump trucks or other designated areas—is a challenging operation requiring the skill and knowledge of experienced operators [6]. Therefore, automation in excavation tasks offers the potential to mitigate operator shortages while decreasing operational costs as well as reducing the high fatality rates commonly associated with heavy machinery usage in construction sites [2,7]. The process of rock excavation can be divided into three distinct stages, as depicted in Fig. 2. In the first stage, the bucket is placed behind the rock. During the second stage, the distance r_R, defined as the distance between the center of bucket and the rock's center of mass, is progressively decreased. Excavation is considered successful once the distance r_R is reduced below the threshold r_S, after which the rock is scooped in the third stage. In this study, the entire sequence is referred to as "rock excavation", with particular emphasis given to the second stage. Experimental data collected from human-operated excavations are utilized to train predictive models that estimate the motion of rock during excavation tasks.

* Corresponding author.
E-mail address: mohammad.heravi@tuni.fi (M. Heravi).

https://doi.org/10.1016/j.simpat.2025.103208
Received 25 July 2025; Received in revised form 8 September 2025; Accepted 19 September 2025
Available online 25 September 2025

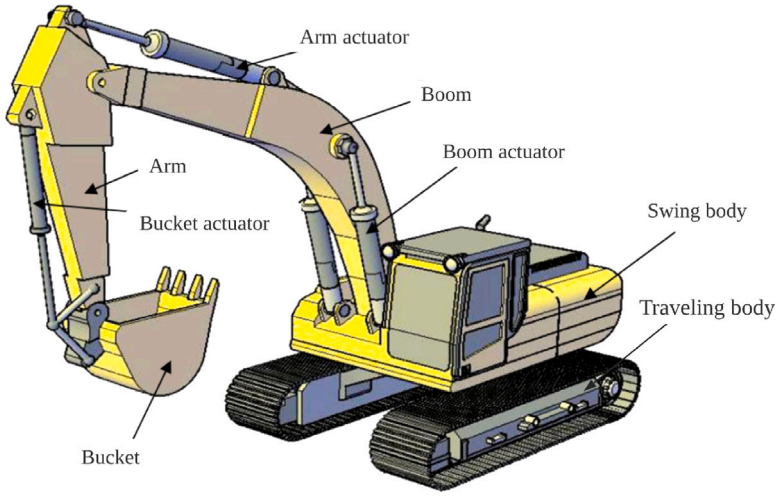


Fig. 1. A standard hydraulic excavator and its different parts [4].

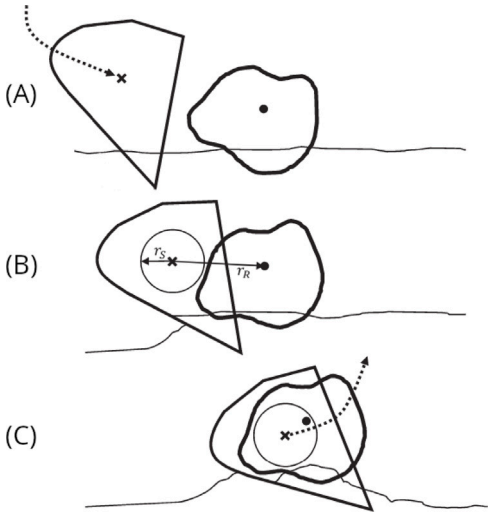


Fig. 2. The rock excavation task proceeds in three stages: **(A)** positioning the bucket behind the rock, **(B)** using the data-driven model to precisely control bucket motion, reducing the distance r_R between bucket and rock, and **(C)** executing a scooping action once r_R decreases below the threshold r_S [6].

In the literature on autonomous excavation, a range of methodologies have been proposed to improve productivity, enhance energy efficiency, and optimize excavator control across diverse soil conditions [8–11]. However, most research efforts focus primarily on efficient bucket-filling practices, and only a limited number of studies directly investigate excavation tasks involving rocks [6]. The behavior of large rocks significantly differs from that of homogeneous soil or gravel, thus necessitating different control objectives. Unlike homogeneous materials, which permit continuous excavation, individual rocks require precise bucket positioning and careful handling.

Although excavation of single rocks using excavators is recognized as a challenging task [5,12,13], to the best of our knowledge, aside from the methods proposed in [6,14], no other approaches have been introduced. In [14], autonomous rock excavation is implemented using if–then rules. These rules mimic skilled operators and are calibrated on a one-fifth-scale LUCIE prototype in mostly homogeneous sandy soil. No explicit model of bucket–soil interaction is used. Rule-based controllers are not robust and lack adaptability to changing conditions. Also, the effect of different soil types is not systematically investigated. In [6], a Gaussian process regression model combined with an unscented Kalman filter is proposed. The approach is implemented on a laboratory UR10e manipulator fitted with a 3D-printed bucket designed specifically for rocks sized 7.5–15 cm. The evaluation is conducted using a single type of granular soil. However, the model is limited to planar motion, the robustness to varying soil properties is not examined, and performance in multi-step-ahead prediction–essential for trajectory planning–is neither quantified nor discussed.

A significant challenge in autonomous rock excavation arises from the complex interactions occurring between the bucket and the soil, which are extremely difficult to model accurately. Moreover, the fundamental physical mechanisms underlying soil-rock excavation are still not fully understood, thereby highlighting substantial barriers that need to be addressed for achieving reliable autonomous systems [12]. Additional complexity of this problem lies in the presence of unknown parameters, such as properties of the rocks and soil. While approximate rock volume estimates might sometimes be available, properties such as the rock's precise shape and mass are typically unknown. Furthermore, terrain unevenness and unknown characteristics of the soil introduce further uncertainties into the excavation scenario.

As stated in [15], capturing a rock using a bucket can be considered analogous to a non-prehensile manipulation problem in robotics. In non-prehensile manipulation, a bucket or another comparable tool guides the object, in this case, a rock, along a desired trajectory through interactions with the surrounding environment. In recent years, data-driven approaches have been demonstrated to effectively address challenges associated with non-prehensile manipulation problems [16]. Specifically, a model-based control approach is introduced in [17], where the dynamics of planar pushing are learned through Gaussian processes and integrated into a model predictive control (MPC) framework. With this approach, accurate trajectory tracking is achieved, requiring as few as 10 training data points. Additionally, [18] proposes a recurrent neural network trained with visual inputs, effectively capturing the history of push interactions to estimate the object's center of mass and determine suitable push actions. This method achieves over 97% success in simulations and consistently succeeds during experiments on real robots after only a limited number of attempts.

Deep learning techniques are widely used in data-driven dynamical modeling for different applications such as non-prehensile manipulation [16]. Due to the short duration of a rock excavation task, it is advantageous to use models that perform better on a limited temporal horizon. Different studies have found that, when only a short history is available, multilayer perceptrons (MLPs) exhibit lower prediction errors compared to recurrent architectures like LSTMs [19–21]. MLPs also offer greater computational efficiency, resulting in shorter training and inference times compared to more complex RNN architectures [22]. This computational advantage makes MLPs especially well-suited to deployment in real-time scenarios [23]. Furthermore, numerous studies have demonstrated that embedding physical priors into neural networks—commonly referred to as physics-informed neural networks (PINNs)—can constrain and enhance model training within a feasible solution space [24–27].

However, training data-driven models typically demands extensive data collection, which proves challenging when performed directly on physical robotic platforms. To mitigate this issue, several studies turn to physics-based simulation environments to provide sufficient training data [16]. Popular examples of these simulators include Bullet Physics, MuJoCo, the Open Dynamics Engine (ODE), NVIDIA PhysX, and Havok [28]. While simulators like PhysX, Havok, and ODE often lack adequate precision for high-accuracy applications [28], Bullet Physics and MuJoCo also face limitations when modeling interactions involving soft or granular materials [29].

It has been demonstrated by [30] that a multi-scale terrain simulation approach based on AGX Dynamics [31] accurately reproduces excavation forces and soil displacements within an error margin of 10–25% compared to high-resolution models while maintaining real-time computational performance. Additionally, a simulation environment known as TERA, created using Unity3D and AGX Dynamics, has been developed by [32], delivering accurate modeling of excavator–terrain interactions and supporting scalable simulations. Furthermore, in bucket-filling scenarios, a simulation-to-reality gap of only approximately 10% has been reported by [33], and the integration of AGX Dynamics-based simulations with learned world models has enabled optimized sequential loading operations, as shown by [34]. Taken together, these studies confirm that AGX Dynamics effectively provides the necessary accuracy and computational performance needed for the simulation of autonomous excavation activities.

Rock excavation involves significant challenges arising from uncertain rock shapes and masses, unknown soil characteristics, and intricate interactions among rock, soil, and bucket, which are difficult to analytically model. In response to these challenges, a PINN is proposed in this paper, leveraging high-fidelity rock excavation scenarios simulated using AGX Dynamics. The primary contributions of this paper are as follows:

- A PINN is designed to predict rock motion dynamics during excavation in 3D space, with kinematic-consistency constraints embedded in the loss function; thus, the out-of-plane motion neglected by the planar Gaussian-process surrogate in [6] is recovered.
- The model is evaluated under different soil conditions and rock geometries; error-propagation over horizons up to 20 steps is analyzed, and the prediction accuracy is shown to remain below the desired threshold. The effect of different soil conditions for this application have not been studied before [6,14].

The remainder of this paper is structured as follows. In Section 2, the problem formulation is presented and the architecture of the proposed data-driven model is detailed. In Section 3, the implementation, data collection, validation, and analysis of the developed model under several rock and soil scenarios are discussed. Finally, conclusions drawn from the study and recommendations for future research are provided in Section 4.

2. Methodology

2.1. Problem formulation

In this section, a framework for predicting rock motion in excavation tasks is introduced. Rock motion prediction during excavation is considered challenging, as traditional physical models, such as terramechanics-based methods [35], often fail to

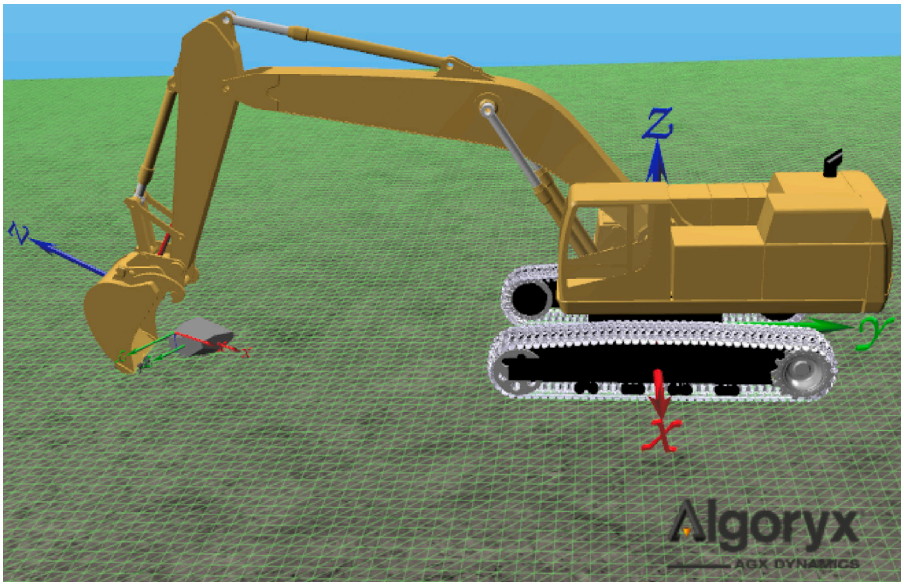


Fig. 3. The rock removal setup in AGX Dynamics.

accurately capture the complex interactions and uncertain parameters involved. Additionally, the excavation process is intrinsically stochastic; under similar conditions, the outcome of rock capture is not deterministic, and the success of capturing a particular rock may vary across trials [6]. It is important to emphasize that this study is dedicated to the development of a model for the motion dynamics of the rock, without the purpose of presenting any planning or control algorithms.

For simplicity, our research study focuses on scenarios involving a single rock whose volume is smaller than that of the bucket and whose geometric shape fits entirely within the bucket. These rocks are considered larger than the individual soil grains surrounding them, though still smaller than the bucket itself. While some rocks might be deeply buried and thus require excessive digging forces, the present work is limited to cases where rocks are sufficiently exposed to allow direct excavation without extreme effort. It should be acknowledged that the excavation of multiple small rocks is regarded as a scooping task [36]. Consequently, the modeling of the motion of small rocks during excavation is beyond the scope of this discussion.

The simulation setup implemented in AGX Dynamics is illustrated in Fig. 3, where the excavator is placed to initiate stage (B) of the previously discussed excavation process (see Fig. 2). Within this simulation environment, all relevant variables are measured according to the "base coordinate frame", which is attached to the excavator's base as indicated in Fig. 3. While the rock is allowed to freely move in 3D space, the bucket's motion is confined within the y–z plane in the present study. To formalize the dynamics describing rock motion, the following state variables are introduced:

$$\mathbf{s} = \begin{bmatrix} x_{rb} & y_{rb} & z_{rb} & r_x & r_y & r_z \end{bmatrix}^T \tag{1}$$

The state vector comprises two primary components. The first component, $\begin{bmatrix} x_{rb} & y_{rb} & z_{rb} \end{bmatrix}^T$, indicates the position of the rock's center of mass with respect to the center of the bucket, expressed in the base coordinate frame. The second component is the unit vector $\mathbf{r} = \begin{bmatrix} r_x & r_y & r_z \end{bmatrix}^T$, representing the orientation along the rock's longest dimension, also expressed in the base coordinate frame. Describing the rock's rotation using this unit vector (whose components satisfy the constraint defined in Eq. (2)) presents a simplifying assumption: any rotation around the rock's longest axis is neglected. The individual elements of the state vector are depicted schematically in the x–y and y–z plane views in Fig. 4. The unit vector orientation is constrained to satisfy:

$$\left\| \begin{bmatrix} r_x & r_y & r_z \end{bmatrix}^T \right\|_2 = 1, \tag{2}$$

where $\|\cdot\|_2$ denotes the Euclidean norm (i.e., the ℓ_2 norm). In this study, the bucket motion is restricted to the y–z plane, resulting in a total of three degrees of freedom. Consequently, the control input vector is introduced as follows:

$$\mathbf{u} = \begin{bmatrix} \Delta y_b & \Delta z_b & \Delta \theta_b \end{bmatrix}^T. \tag{3}$$

In this definition, the terms Δy_b and Δz_b denote displacements of the bucket along the y- and z-axes in the base coordinate frame, respectively, while $\Delta \theta_b$ refers to the change in the bucket's orientation angle θ_b, defined as rotation around the x-axis. These variables are visually illustrated in Fig. 4. The aim of this work is the development and training of a PINN capable of predicting the state vector $\mathbf{s}$ based on the state and control inputs from a set of previous time steps.

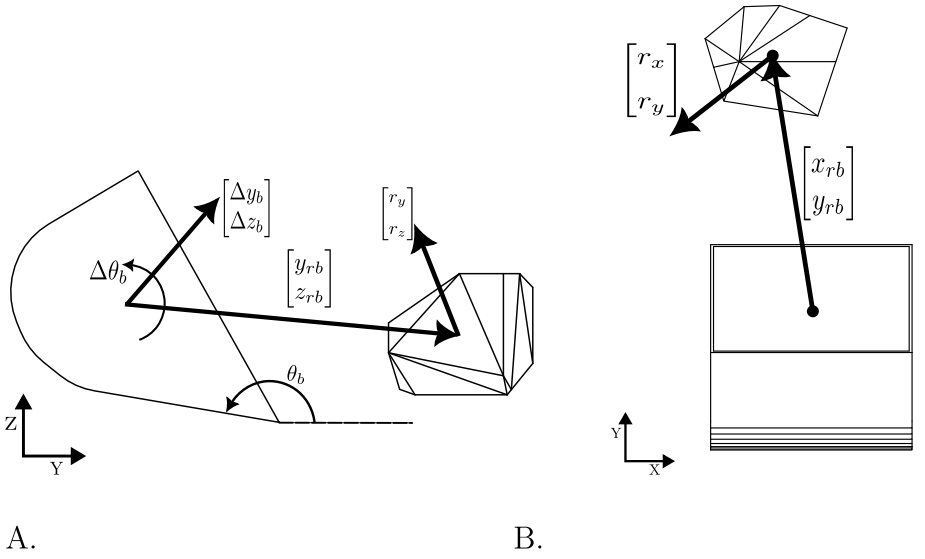


Fig. 4. State and control variables introduced in Eqs. (1) and (3): **(A)** y–z plane; **(B)** x–y plane views.

2.2. Terminology and definitions

Key terms used throughout the paper are defined below to establish terminology and maintain consistent usage across sections.

Look-back window: the look-back window refers to the historical data that the model receives as input for prediction. It consists of a fixed number of the most recent states and control inputs that provide background regarding the evolution of the system up to the current time.

Prediction horizon: the prediction horizon refers to the extent into the future that the model is required to predict.

Time step: throughout the paper, "step" refers to a time step of the process/simulator, i.e., one discrete move forward in time.

Ground truth: ground truth refers to the reference data considered accurate and utilized to supervise training processes and validate model predictions. This data may originate from direct measurements, expert annotations, or verified simulations.

Penalty term: during the training phase, a penalty term is incorporated into the cost function to promote favorable outcomes beyond simply fitting the data. Such penalties can enhance regularity (e.g., smoothness), ensure physical or geometric coherence, or discourage unrealistic predictions.

2.3. PINN architecture for rock motion dynamics

A PINN is used to model the complex, nonlinear relationships that characterize rock movement during excavation. The proposed PINN is an MLP augmented with a custom loss function to enforce a physical constraint. The proposed PINN architecture is specifically designed to incorporate temporal dependencies by employing a carefully selected look-back window spanning several previous time steps. Let $\mathbf{x}_t$ represent the input vector at time step t, consisting of the state variables and corresponding control inputs collected over a look-back window of length ℓ:

$$\mathbf{x}_t = \begin{bmatrix}\mathbf{s}_{t-\ell:t} & \mathbf{u}_{t-\ell:t}\end{bmatrix}^T, \tag{4}$$

where $\mathbf{s}_{t-\ell:t}$ corresponds to the set of state variables between time $t-\ell$ to t, while $\mathbf{u}_{t-\ell:t}$ corresponds to the control inputs over the same interval. It should be noted that a look-back window of $\ell = 0$ implies that only the current states and control inputs are employed. The vector $\mathbf{y}_t$ represents the changes in state variables at step t, as defined in Eq. (5).

$$\mathbf{y}_t = \begin{bmatrix}\Delta x_{rb,t} & \Delta y_{rb,t} & \Delta z_{rb,t} & \Delta r_{x,t} & \Delta r_{y,t} & \Delta r_{z,t}\end{bmatrix}^T. \tag{5}$$

The relationship between successive state vectors $\mathbf{s}_{t+1}$, $\mathbf{s}_t$ and the state change $\mathbf{y}_t$ is described by Eq. (6). Given that the ℓ_2-norm of the last three elements of the state vector $\mathbf{s}$ must remain equal to 1, these components are normalized as expressed in Eq. (6).

$$\mathbf{s}_{t+1} = \begin{bmatrix} x_{rb,t}+\Delta x_{rb,t}\\ y_{rb,t}+\Delta y_{rb,t}\\ z_{rb,t}+\Delta z_{rb,t}\\ \frac{r_{x,t}+\Delta r_{x,t}}{m}\\ \frac{r_{y,t}+\Delta r_{y,t}}{m}\\ \frac{r_{z,t}+\Delta r_{z,t}}{m}\end{bmatrix}, \qquad m = \left\| \begin{bmatrix}(r_{x,t}+\Delta r_{x,t}) & (r_{y,t}+\Delta r_{y,t}) & (r_{z,t}+\Delta r_{z,t})\end{bmatrix}^T \right\|_2. \tag{6}$$

Table 1
Key parameters of SAND_1 and DIRT_1 materials used in the study.

Parameter	SAND_1	DIRT_1	Unit
Cohesion (bulk)	0.0	2100.0	Pa
Density	1474.0	1474.0	kg/m^3
Maximum Density	1800.0	2000.0	kg/m^3
Internal Friction Angle	0.68	0.70	rad

The input and output vectors of the PINN are represented by $\mathbf{x}_t$ and $\hat{\mathbf{y}}_t$, respectively, where $\hat{\mathbf{y}}_t$ is the estimate of $\mathbf{y}_t$. The neural network function, parameterized by Θ, is denoted by f_Θ and is defined in Eq. (7):

$$\hat{\mathbf{y}}_t = f_\Theta(\mathbf{x}_t). \tag{7}$$

The objective of the PINN training process is formulated as the minimization of a loss function consisting of two distinct components: a primary component that quantifies prediction errors and a secondary penalty term introduced to enforce physical constraints. The Mean Squared Error (MSE) is selected as the primary loss and is defined as follows:

$$\begin{aligned} \boldsymbol{\delta}_t &:= \mathbf{y}_t - \hat{\mathbf{y}}_t, \qquad \boldsymbol{\delta}_t \in \mathbb{R}^6, \\ \mathbf{Q} &:= \operatorname{diag}\left(\tfrac{1}{\ell_b}, \tfrac{1}{\ell_b}, \tfrac{1}{\ell_b}, 1, 1, 1\right), \\ \mathcal{L}_{\text{primary}} &= \frac{1}{6}\,\boldsymbol{\delta}_t^{\mathsf{T}}\,\mathbf{Q}^2\,\boldsymbol{\delta}_t, \end{aligned} \tag{8}$$

where $\boldsymbol{\delta}_t$ represents the prediction error, with $\mathbf{Q}$ making the first three translational components unit-less by dividing them by the bucket length $\ell_b = 1.3\,\text{m}$. Also, using ℓ_b for scale normalization removes any hard-coding of rock-specific dimensions. Since the PINN outputs change of the state variables, a loss is introduced to enforce that these variables obey the constraint in Eq. (2). According to [37], small changes applied to a unit vector must remain orthogonal to the vector itself. Therefore, the penalty term is defined as:

$$\mathcal{L}_{penalty} = (\mathbf{r}^\top \Delta\mathbf{r})^2, \tag{9}$$

where $\Delta\mathbf{r} = \begin{bmatrix}\Delta r_{x,t} & \Delta r_{y,t} & \Delta r_{z,t}\end{bmatrix}^T$ represents change in orientation vector $\mathbf{r}$ at time step t. This penalty ensures constancy of the ℓ_2-norm of $\mathbf{r}$ over successive predictions. The final loss function is a weighted combination of the primary prediction error and the penalty term:

$$\mathcal{L}_{total} = (1-\alpha)\mathcal{L}_{primary} + \alpha\mathcal{L}_{penalty}\,, \tag{10}$$

where the weight $\alpha \in [0, 1]$ controls the relative trade-off between achieving prediction accuracy and maintaining adherence to physical constraints.

3. Results

In this section, first, the experimental setup used for model training and evaluation is described. Then, the impact of selecting different values of the look-back window parameter, ℓ, is systematically analyzed to determine its optimal value for predictive accuracy. Next, the performance of the model on the training conditions as well as different types of rock and soil scenarios is evaluated. Finally, the main observations and implications of these results are discussed.

3.1. Experiment setup

As previously discussed, a substantial amount of data is required for training models based on data-driven techniques; however, data collection directly from a real excavator is costly and time-consuming. To overcome these practical limitations, AGX Dynamics is employed as a simulation platform, within which relevant datasets are generated through manual operation (Fig. 5). The simulated excavator is modeled after a CAT® 365 machine, with a total mass of 66 t, a maximum operating reach of 14.4 m, and a bucket capacity of 3.8 m^3. In this setup, the distance between the center of the bucket and the tip of the bucket is about 1.3 m. Hence, the scooping threshold, r_S in Fig. 2, of 0.5 m ensures the successful excavation. In this study, two distinct soil types and two rock geometries are selected for the experiments. The two considered rock shapes, henceforth labeled Rx1 and Rx2, are shown in Fig. 6. The mass and maximum dimension of rock Rx1 are 558.59 kg and 1.04 m, respectively, whereas rock Rx2 weighs 611.84 kg and measures 1.27 m in its largest dimension. Furthermore, compared to the rounder shaped Rx1, Rx2 exhibits a flatter geometry. Regarding soil types, two conditions ("SAND_1" and "DIRT_1") from the terrain model library included in AGX Dynamics [38] are selected and illustrated in Fig. 7. In AGX Dynamics, the SAND_1 and DIRT_1 terrains are calibrated to match bulk behavior, and Table 1 reports important bulk properties. The calibration approach and details are described in [30,39,40].

To facilitate the training and evaluation of the proposed PINN, a total of 200 manual experiments are conducted across different rock geometries and soil conditions, designated as D1, D2, D3, and D4. A summary description of these datasets is provided in Table 2. Dataset D1, which includes Rx1 and SAND_1 conditions, is employed for training the model and determining the optimal

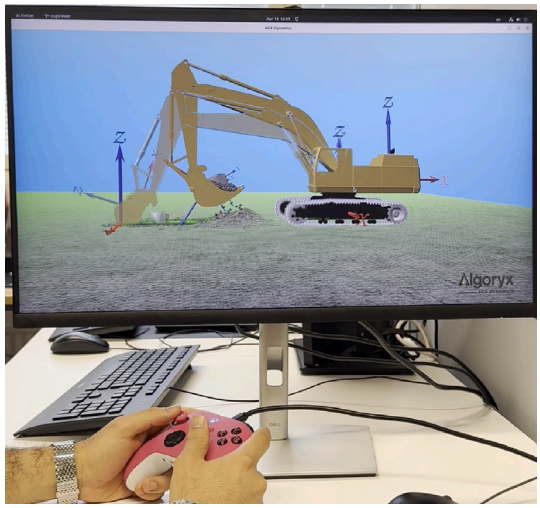


Fig. 5. The high-fidelity AGX Dynamics simulation setup employed for dataset generation.

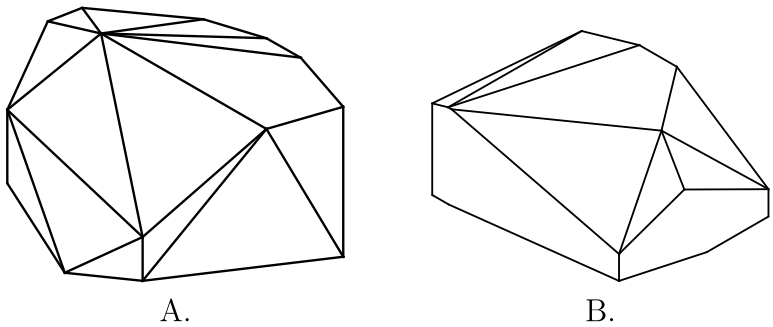


Fig. 6. Two rock geometries utilized in simulation experiments: **(A)** Rx1, a round-shaped rock (558.59 kg), and **(B)** Rx2, a flatter rock (611.84 kg), demonstrating different shapes and mass distributions.

Table 2
A summary of the four datasets employed for model training and validation is provided, detailing the soil and rock types, number of trials, and experimental purpose.

Dataset Name	Soil Type	Rock Type	Number of Trials	Purpose
D1	SAND_1	Rx1	170	Training the PINN and optimization of look-back window ℓ
D2	SAND_1	Rx1	10	Validation under the same rock and soil conditions as training
D3	SAND_1	Rx2	10	Validation under a different rock condition
D4	DIRT_1	Rx1	10	Validation under a different soil condition

look-back parameter, ℓ. Dataset D2, which shares similar conditions with D1, is utilized to assess the trained model's performance under identical training conditions. Conversely, datasets D3 and D4 — incorporating different rock and soil conditions, respectively — are used to evaluate the model's accuracy in conditions not encountered during training. Throughout the experiment, both the state vector and the control inputs are collected at a constant sampling frequency of 10 Hz. The average duration of each experiment is approximately 6.2 s. To ensure diversity of experimental conditions and enhance dataset richness, not all experiments conclude with successful rock excavation. Also, at the beginning of each experiment, the initial orientation of the rock is randomized.

The proposed PINN comprises 6 linear layers, with hidden layers consisting of 128, 512, 256, and 128 neurons, respectively. Layer normalization is applied between these hidden layers, and the Rectified Linear Unit (ReLU) activation function is employed throughout the entire network. Dataset D1, which corresponds to rock type Rx1 and soil type SAND_1, is used to train and test the

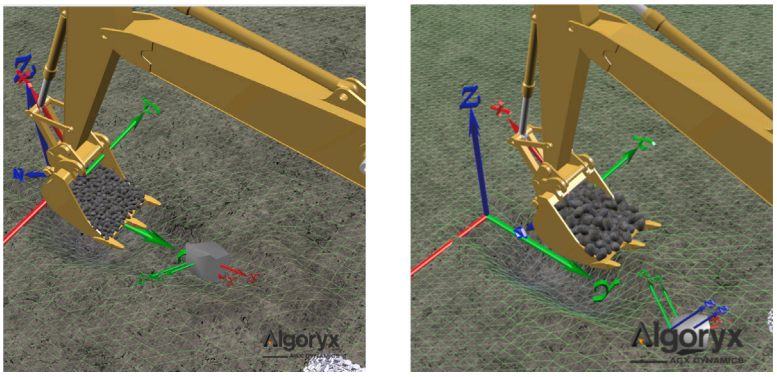


Fig. 7. Soil types used in the experiments: **(A)** SAND_1 and **(B)** DIRT_1 both from the AGX Dynamics library [38].

Table 3
Summary of hyper-parameters used for training.

Hyperparameter	Value
Batch size	64
Optimizer	Adam
Learning rate	0.001
Number of epochs	500
Penalty term coefficient α in Eq. (10)	0.1

model. The dataset is split, with 80% reserved for training the model and the remaining 20% utilized for determining the optimal look-back window (ℓ) and monitoring training convergence. Data are processed using a batch size of 64 to increase computational efficiency and enhance resource utilization. The learning rate is set to 0.001, and the training is carried out for 500 epochs. The weight α, controlling the penalty term in Eq. (10), is set to 0.1. This particular setting is determined through a grid search applied to dataset D1. Models are trained using $\alpha \in \{0, 0.05, 0.1, 0.2, 0.3\}$ while maintaining consistent hyper-parameters, with $\mathcal{L}_{total}$ serving as the criterion to select the most suitable α value. Network parameters are optimized by Adam optimizer. A summary of these hyper-parameters is provided in Table 3. PyTorch is utilized for model implementation, benefiting from its effective and flexible support for complex neural network designs [41]. The training process is accelerated by GPU computations performed on an Ubuntu workstation equipped with an Intel Core i7-13700 CPU and an NVIDIA GeForce RTX 4080 GPU.

In this paper, multi-step (recursive) prediction is employed to assess model performance, as it highlights error accumulation. Let $h \in \mathbb{N}$ denote the number of prediction steps ahead. At each time step, state predictions and control inputs of the current time step as well as those of a window of ℓ steps back are provided as inputs for the next time step. Thus, each new prediction depends on the prior outputs of the model, as follows:

$$\begin{aligned}
&(\mathbf{s}_{t-\ell:t}, \mathbf{u}_{t-\ell:t}) \rightarrow \hat{\mathbf{s}}_{t+1},\\
&([\mathbf{s}_{t-\ell+1:t}, \hat{\mathbf{s}}_{t+1}], \mathbf{u}_{t-\ell+1:t+1}) \rightarrow \hat{\mathbf{s}}_{t+2},\\
&\vdots\\
&\begin{cases} ([\mathbf{s}_{t+h-1-\ell:t}, \hat{\mathbf{s}}_{t+1:t+h-1}], \mathbf{u}_{t+h-1-\ell:t+h-1}) \rightarrow \hat{\mathbf{s}}_{t+h}, & \text{if } h \le \ell + 1\\ (\hat{\mathbf{s}}_{t+h-1-\ell:t+h-1}, \mathbf{u}_{t+h-1-\ell:t+h-1}) \rightarrow \hat{\mathbf{s}}_{t+h}, & \text{if } h > \ell + 1. \end{cases}
\end{aligned} \tag{11}$$

During the second stage of rock excavation, the goal is to minimize the distance between the bucket's center and the rock's mass center (see Section 1), while the orientation of the rock is not important. To focus specifically on how accurately the model predicts this relative position, a new metric is introduced. This metric calculates the ℓ_2-norm between the predicted position and the actual position of the rock after h steps as follows:

$$e_{t+h} = \sqrt{\left(\hat{y}_{rb,t+h} - y_{rb,t+h}\right)^2 + \left(\hat{z}_{rb,t+h} - z_{rb,t+h}\right)^2}, \tag{12}$$

where $y_{rb,t+h}$ and $z_{rb,t+h}$ denote the distances between the bucket and the rock along the y- and z-axes at time step $t+h$, respectively, while $\hat{y}_{rb,t+h}$ and $\hat{z}_{rb,t+h}$ denote corresponding predictions provided by the neural network.

3.2. Choosing optimal look-back parameter

To accurately capture the temporal dependencies in rock motion dynamics, different look-back window sizes, denoted by ℓ, are tested. Specifically, window sizes ℓ ranging from 0 to 5 are considered. For each of these values, a PINN is trained using dataset

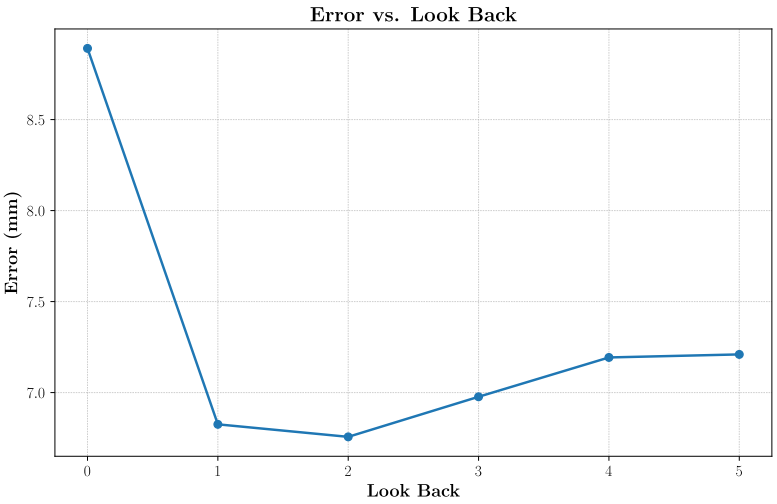


Fig. 8. Effect of varying look-back window size (ℓ) on the prediction error.

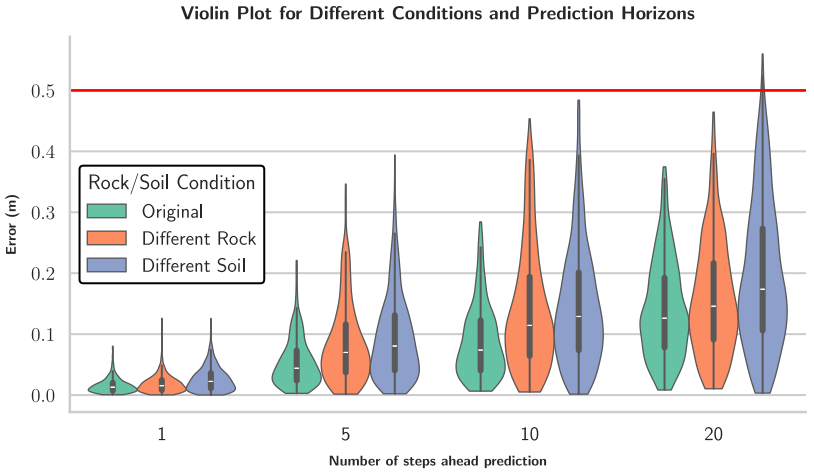


Fig. 9. Violin plots comparing prediction error distributions for each time-step horizon (1, 5, 10, 20) under three different scenarios—original training conditions (dataset D2), altered rock geometry (dataset D3), and different soil condition (dataset D4). The red line at 0.5 m indicates the threshold used to determine successful rock capture.

D1, keeping hyper-parameters such as the number of hidden layers, learning rate, and batch size consistent across all tests. After training, the prediction performance is measured using the prediction error defined in Eq. (12), specifically at a prediction horizon of $h = 1$. The mean prediction errors for the different look-back window sizes are presented in Fig. 8. Based on these results, a window size of $\ell = 2$ is found to be optimal. This choice provides enough historical information about the recent system states and control inputs without unnecessarily increasing model complexity. Therefore, $\ell = 2$ is selected for use in all subsequent experiments due to its higher accuracy.

3.3. Model performance on different scenarios

By utilizing datasets D2, D3, and D4, each consisting of 10 experiments, the accuracy of the proposed model is assessed under three distinct scenarios: (i) original rock and soil conditions identical to those employed during model training, (ii) previously unseen rock geometry and mass, and (iii) significantly different soil characteristics. For each scenario, the prediction errors for various horizons, specifically $h \in \{1, 5, 10, 20\}$, are computed using Eq. (12). These error distributions across multiple prediction horizons and scenarios are illustrated through violin plots in Fig. 9. Each violin-shaped graphic illustrates the distribution of prediction errors during the 10 experiments for the corresponding scenario (datasets D2, D3, and D4) and prediction horizon ($h \in \{1, 5, 10, 20\}$). The width of each violin plot at a given error magnitude indicates its probability density; broader sections reflect a higher density of data points at those specific error values. The embedded vertical boxes within the violins represent the interquartile range (IQR),

encompassing the central 50% of the data (ranging from the 25th to the 75th percentile). Horizontal lines within these boxes denote the medians, highlighting the central tendency of the data. Furthermore, vertical lines, known as whiskers, extend beyond the boxes to illustrate the range of data, excluding extreme outliers. Lastly, the horizontal red line positioned at 0.5 m denotes the threshold utilized for determining successful rock capture.

A central observation common across all scenarios is that lower prediction errors are consistently maintained by the model at shorter prediction horizons (from 1 to 5 steps ahead); however, an increase in errors is observed at extended horizons (from 10 to 20 steps ahead). This pattern is evident in Fig. 9, where the median errors for predictions of 1 and 5 steps ahead are comparatively lower, and the error distributions are notably denser at smaller values than those at longer horizons. An additional observation derived from Fig. 9 is that variations in soil conditions result in greater discrepancies compared to changes in rock geometry or mass. Specifically, in this figure, higher errors consistently appear in scenarios involving different soil conditions across all prediction horizons. These scenarios demonstrate higher median errors and distributions shifted towards larger values relative to the other examined cases. According to Fig. 9, the prediction errors remain below the designated threshold across all scenarios and horizons, except for predictions made 20 steps ahead under varying soil conditions for few time steps during the 10 corresponding experiments.

3.3.1. Temporal error analysis: Baseline

In Fig. 10, the predicted and actual distances between the bucket and the rock along the y- and z-axes, as well as the corresponding prediction errors for various horizons, are illustrated for a randomly selected episode out of 10 from dataset D2. In this figure, y_{rb} and z_{rb} represent the distances between the bucket and the rock over time, displaying predictions for multiple horizons alongside the ground truth. These predictions are generated recursively according to the scheme in Eq. (11). As depicted in Figs. 10(A) and 10(B), these distances progressively approach zero, indicating successful rock capture by the end of the experiment. Additionally, predictions for the initial 3 time steps are excluded from the figures due to the chosen look-back window size of $\ell = 2$. These initial data points instead provide the basis for generating predictions 1, 5, 10, or 20 steps ahead. Given that the data collection frequency is 10 Hz, the respective prediction horizons correspond to intervals of 0.1 s, 0.5 s, 1 s, and 2 s. As shown in Fig. 10(C), prediction errors for shorter horizons (1- and 5-step-ahead) remain smaller compared to those at longer horizons (10- and 20-step-ahead). Furthermore, the prediction errors consistently remain below 0.2 m for all prediction horizons.

3.3.2. Temporal error analysis: Unseen rock

In Fig. 11, the actual versus predicted trajectories of the rock relative to the bucket along the y- and z-axes, as well as their corresponding prediction errors across multiple horizons, are presented for a single, randomly selected episode from dataset D3 featuring an unseen rock geometry. According to Figs. 11(A) and 11(B), an increasing trend in the distance between the bucket and the rock over time is evident, signifying that the rock progressively moves away from the bucket during this particular experiment. Additionally, as indicated in Fig. 11(C), shorter-term predictions (1- and 5-step-ahead) demonstrate relatively lower errors compared to longer horizons (10- and 20-step-ahead). Despite this increased uncertainty over longer prediction horizons, the error magnitude consistently remains below 0.4 m for all examined horizons. Figs. 10 and 11 show that the model prediction error is below the threshold across both successful and unsuccessful episodes.

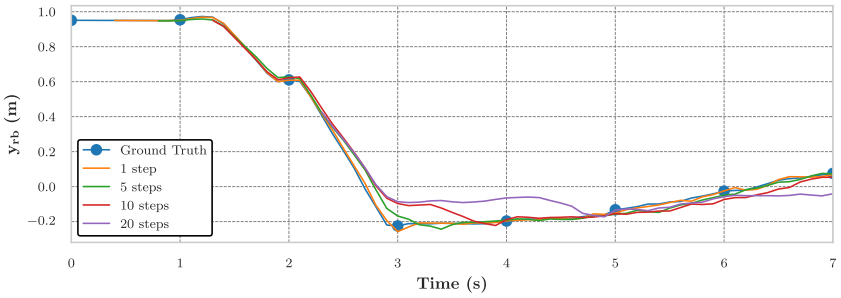
3.3.3. Temporal error analysis: Different soil

In Fig. 12, the predicted and actual trajectories of the rock relative to the bucket along the y- and z-axes, as well as the corresponding errors for multiple prediction horizons, are illustrated for a randomly selected experiment from dataset D4, conducted under different soil conditions. According to Figs. 12(A) and 12(B), the distances do not converge to zero, meaning that the rock is not captured successfully. Also, the distances between the bucket and the rock fluctuate notably throughout the experiment, reflecting higher variability and more challenging dynamic interactions introduced by the altered soil properties. Moreover, it can be observed in Fig. 12(C) that predictions at shorter horizons (1- and 5-step-ahead) exhibit notably lower errors in comparison to longer horizons (10- and 20-step-ahead). Despite these increases, the prediction errors remain below 0.5 m except for one sample time for 20-step-ahead horizon.

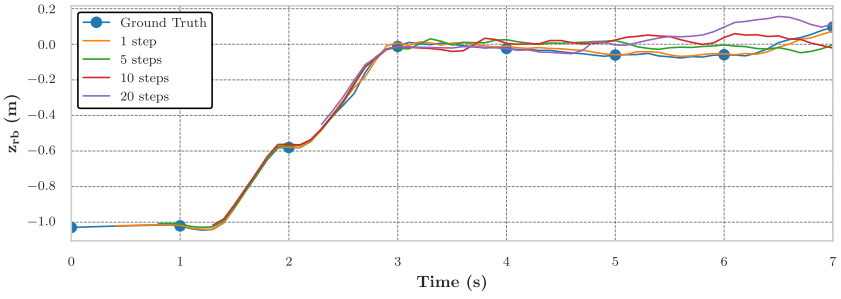
3.4. Discussion

In Fig. 9, the rock–bucket relative motion is shown to be accurately predicted by the proposed PINN for all evaluated horizons and operating conditions. The prediction errors are consistently maintained well below the 0.5 m capture threshold, with the sole exception being at the 20-step prediction horizon under the different-soil scenario, where a limited number of outliers modestly surpass this threshold. Moreover, the observed reliability across previously unseen rock shapes and soil conditions indicates that the PINN captures a generalizable representation of rock dynamics rather than merely memorizing specific training trajectories.

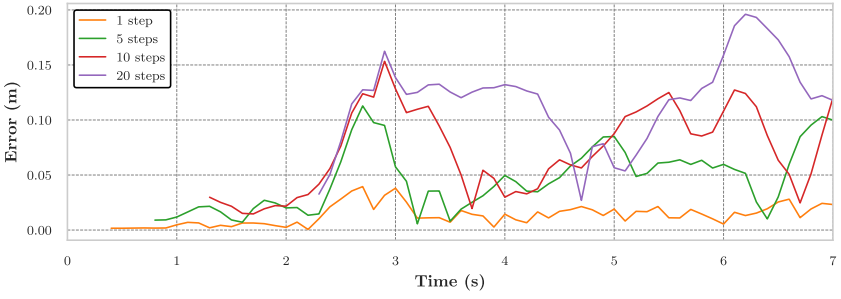
The presented prediction accuracy carries two immediate implications in the context of control engineering. First, the developed model can directly serve as a computationally efficient yet accurate simulator suitable for extensive virtual experimentation. In particular, reinforcement learning (RL) agents can be trained using this simulation environment to identify excavation strategies without incurring safety hazards or the wear-and-tear expenses associated with real-world trials, thereby accelerating the iteration cycle involved in policy development and hyperparameter optimization. Second, the proposed predictive approach naturally aligns with the principles of MPC. In this framework, state predictions provided by the trained PINN can be utilized by an optimization layer to systematically generate control actions. Since the proposed PINN inherently captures nonlinear interactions, the result-



(A)



(B)



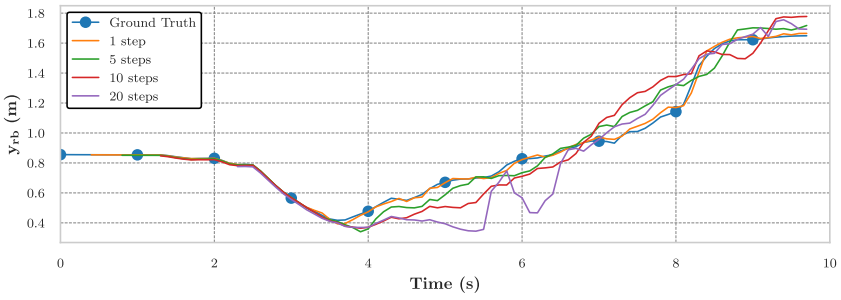
(C)

Fig. 10. Evaluation of the model under the same rock and soil conditions as in the training dataset, across different time horizons. **(A)** Prediction of position of rock relative to the bucket along the y-axis compared to ground truth, **(B)** Prediction of position of rock relative to the bucket along the z-axis compared to ground truth, and **(C)** Magnitude of the prediction error e_{t+h}, defined in Eq. (12), for horizons $h \in \{1, 5, 10, 20\}$ time steps.

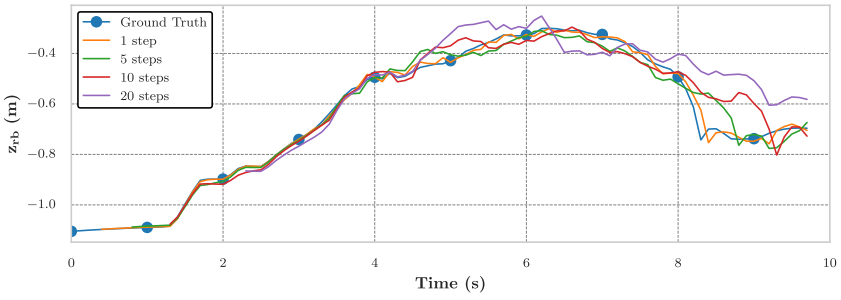
ing MPC formulation explicitly considers transient contact phenomena typically overlooked by traditional terramechanics-based approximations.

As previously mentioned, higher prediction errors are observed for the different-soil scenario (dataset D4). This quantitative finding emphasizes the critical role of bucket–soil interactions, a point that was qualitatively recognized in earlier autonomous excavation research [8,9]. While varying rock characteristics were the primary concern of the study conducted by [6], the current results reveal that variability in soil conditions may exert an even more significant effect on rock dynamics and, consequently, on overall excavation controller performance.

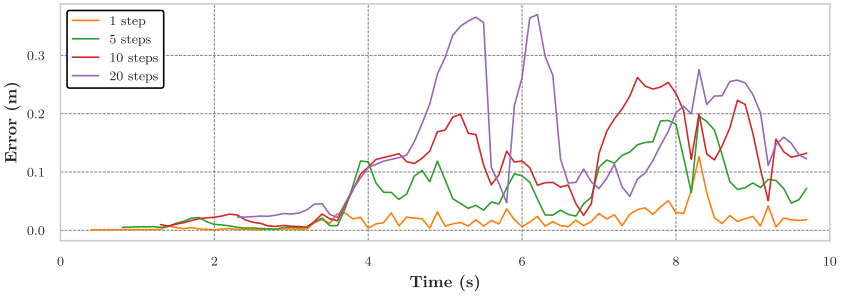
Regarding the transferability of the trained model, several factors should be highlighted. First, the synthetic dataset is generated by the AGX Dynamics simulator, which is able to reproduce bucket-filling forces and trajectories within about 10% of what is measured in real machines [33]. Secondly, a considerable extent of randomization is implemented during data collection through the use of random initial orientations for the rock and a variety of manual experiments—some of which did not result in successful rock excavation. Third, a sensitivity analysis is carried out with a different rock shape and soil type. In this analysis, the median error for 20-step-ahead predictions increased by less than approximately 0.05 m, as shown in Fig. 9. These results suggest that the model



(A)



(B)



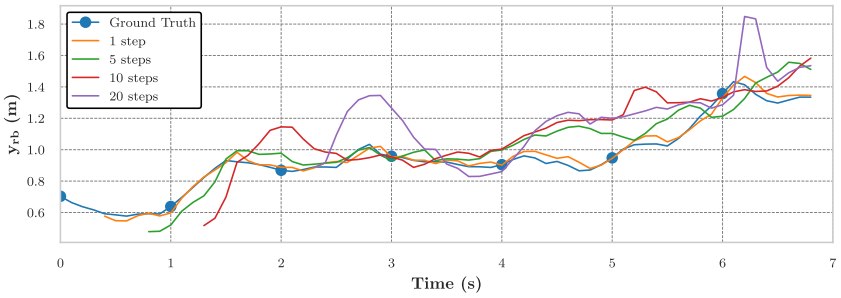
(C)

Fig. 11. Evaluation of the model for different rock condition over different time horizons. **(A)** Prediction of position of rock relative to the bucket along the y-axis compared to ground truth, **(B)** Prediction of position of rock relative to the bucket along the z-axis compared to ground truth, and **(C)** Magnitude of the prediction error e_{i+h}, defined in Eq. (12), for horizons $h \in \{1, 5, 10, 20\}$ time steps.

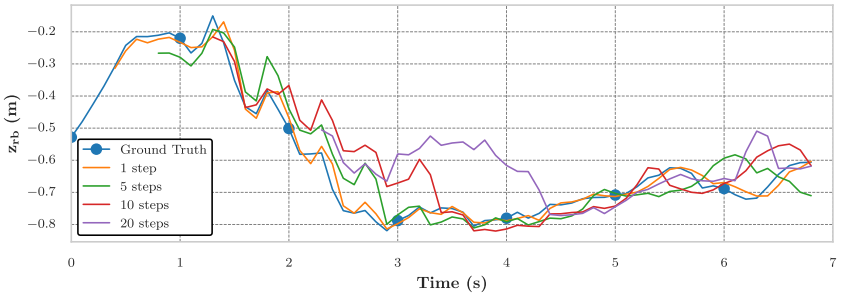
is able to generalize beyond the original training data and, by design, is rock- and soil-agnostic, rather than tied to prespecified rock sizes, geometries, or soil types. However, it should be noted that a full evaluation in real-world conditions is left for future work.

The precise determination of the center of mass of a rock presents significant challenges in practice due to factors such as occlusions and soil coverage, and this study does not aim to address the estimation of this variable. In future research, the proposed learned model may be utilized as the process model within a Kalman-filter–based estimator (e.g., EKF/UKF), which integrates visual data with machine signals (such as force measurements) to enhance the estimation of the position of the rock.

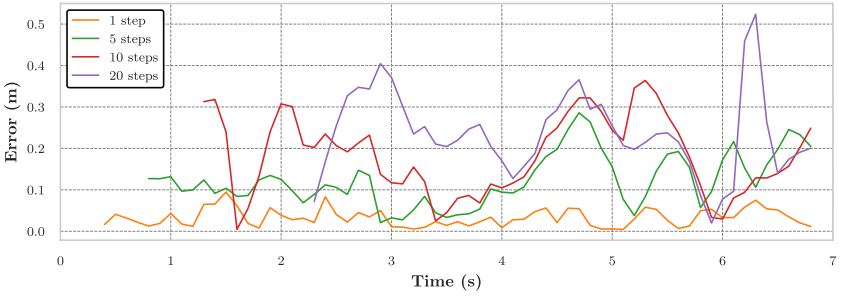
For clarity, the present study deliberately narrows its scope to excavation cases involving a single rock whose volume is smaller than the bucket and whose geometry can fit completely inside the bucket. Such rocks are larger than the surrounding soil grains yet remain smaller than the bucket itself. Also the attention is limited to rocks that are sufficiently exposed so they can be removed without extreme digging forces; deeply buried rocks that would demand excessive breakout forces are not considered. Moreover, future work should address planning and control aspects, such as rock reorientation and the selection of scooping angles. Finally, because the modeling framework is purely kinematic, constraints related to energy consumption or hydraulic power are outside the scope of this work.



(A)



(B)



(C)

Fig. 12. Evaluation of the model for different soil condition over different time horizons. **(A)** Prediction of position of rock relative to the bucket along the y-axis compared to ground truth, **(B)** Prediction of position of rock relative to the bucket along the z-axis compared to ground truth, and **(C)** Magnitude of the prediction error e_{t+h}, defined in Eq. (12), for horizons $h \in \{1, 5, 10, 20\}$ time steps.

4. Conclusion

This study focuses on modeling the dynamics of large rock excavation using a data-driven approach, based on experiments carried out in a high-fidelity physics engine. The primary objective is to develop a PINN for estimating rock motion dynamics during excavation. This model should be robust under various uncertainties, including differences in rock characteristics and soil conditions. To that end, a custom penalty is incorporated into the training loss function to ensure compliance with physical constraints, and the influence of varying look-back window lengths is systematically explored to optimally represent key temporal dependencies. The obtained results demonstrate that, even when predicting the rock-bucket distance up to 20 time steps ahead, the model's prediction errors typically remain below the threshold of 0.5 m—with only a few exceptional outliers observed under different soil conditions. Overall, the attained accuracy provides a solid groundwork for subsequent research aimed at autonomous excavation planning and control. Although based on a compact feedforward MLP, our approach's novelty lies in embedding a physics-informed constraint within the loss function and employing an optimized two-step look-back window.

Future research should focus on further refining the model's applicability to real-world excavation scenarios through investigation of more complex and diversified conditions. Additionally, the incorporation of online learning methods that continually update model parameters based on real-time sensor feedback could enhance the performance of the framework in uncertain excavation

environments. In parallel, future research should address the development of vision systems capable of reliably tracking the position of the rock during excavation.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work the authors used ChatGPT in order to improve language clarity. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The work was funded in part by Horizon Europe Project XSCAVE under Grant 101189836. Also, the authors gratefully acknowledge financial support from the Research Council of Finland through the PROFI 7 grant.

Data availability

Data will be made available on request.

References

[1] N. Franceschini, P. Thangeda, M. Ornik, K. Hauser, Autonomous excavation of challenging terrain using oscillatory primitives and adaptive impedance control, 2024, http://dx.doi.org/10.48550/arxiv.2409.18273, arXiv.org.
[2] P. Egli, L. Terenzi, M. Hutter, Reinforcement learning-based bucket filling for autonomous excavation, IEEE Trans. Field Robot. 1 (2024) 170–191, http://dx.doi.org/10.1109/TFR.2024.3432508.
[3] O.M.U. Eraliev, K.-H. Lee, D.-Y. Shin, C.-H. Lee, Sensing, perception, decision, planning and action of autonomous excavators, Autom. Constr. 141 (2022) http://dx.doi.org/10.1016/j.autcon.2022.104428, 104428–104428.
[4] A. Molaei, A. Kolu, N. Haaraniemi, M. Geimer, Automatic estimation of excavator's actual productivity in trenching and grading operations using building information modeling (BIM), Actuators 12 (11) (2023) http://dx.doi.org/10.3390/act12110423.
[5] X.D. Huang, L.E. Bernold, Robotic rock handling during backhoe excavation, in: 10th International Symposium on Automation and Robotics in Construction, Houston, TX, USA, 1993, pp. 355–362, http://dx.doi.org/10.22260/ISARC1993/0046.
[6] F.E. Sotiropoulos, H.H. Asada, Autonomous excavation of rocks using a Gaussian process model and unscented Kalman filter, IEEE Robot. Autom. Lett. 5 (2) (2020) 2491–2497, http://dx.doi.org/10.1109/LRA.2020.2972891.
[7] L. Terenzi, M. Hutter, Toward autonomous excavation planning, IEEE Trans. Field Robot. 1 (2024) 292–317, http://dx.doi.org/10.1109/TFR.2024.3485037.
[8] F. Sotiropoulos, H. Asada, A model-free extremum-seeking approach to autonomous excavator control based on output power maximization, IEEE Robot. Autom. Lett. 4 (2) (2019) 1005–1012, http://dx.doi.org/10.1109/LRA.2019.2893690.
[9] D. Jud, G. Hottiger, P. Leemann, M. Hutter, Planning and control for autonomous excavation, IEEE Robot. Autom. Lett. 2 (4) (2017) 2151–2158, http://dx.doi.org/10.1109/LRA.2017.2721551.
[10] A.A. Dobson, J.A. Marshall, J. Larsson, Admittance control for robotic loading: Design and experiments with a 1-tonne loader and a 14-tonne load-haul-dump machine, J. Field Robot. 34 (1) (2017) 123–150, http://dx.doi.org/10.1002/rob.21654.
[11] G.J. Maeda, I.R. Manchester, D.C. Rye, Combined ILC and disturbance observer for the rejection of near-repetitive disturbances, with application to excavation, IEEE Trans. Control Syst. Technol. 23 (5) (2015) 1754–1769, http://dx.doi.org/10.1109/TCST.2014.2382579.
[12] X. Huang, L.E. Bernold, Toward an adaptive control model for robotic backhoe excavation, Transp. Res. Rec. 1406 (1993) 20–24, https://api.semanticscholar.org/CorpusID:108001066.
[13] C. McKinnon, J.A. Marshall, Automatic identification of large fragments in a pile of broken rock using a time-of-flight camera, IEEE Trans. Autom. Sci. Eng. 11 (3) (2014) 935–942, http://dx.doi.org/10.1109/TASE.2014.2308011.
[14] D.A. Bradley, D.W. Seward, The development, control and operation of an autonomous robotic excavator, J. Intell. Robot. Syst. 21 (1) (1998) 73–97, http://dx.doi.org/10.1023/a:1007932011161.
[15] M.T. Mason, Progress in nonprehensile manipulation, Int. J. Robot. Res. 18 (11) (1999) 1129–1141, http://dx.doi.org/10.1177/02783649922067762.
[16] J. Stüber, C. Zito, R. Stolkin, Let's push things forward: A survey on robot pushing, Front. Robot. AI 7 (2020) http://dx.doi.org/10.3389/frobt.2020.00008.
[17] M. Bauza, F.R. Hogan, A. Rodriguez, A data-efficient approach to precise and controlled pushing, 2018, http://dx.doi.org/10.48550/arXiv.1807.09904, arXiv:1807.09904.
[18] J. Li, W. Sun Lee, D. Hsu, Push-Net: Deep planar pushing for objects with unknown physical properties, in: Robotics: Science and Systems XIV, Robotics: Science and Systems Foundation, 2018, http://dx.doi.org/10.15607/RSS.2018.XIV.024.
[19] S. Bai, J.Z. Kolter, V. Koltun, An empirical evaluation of generic convolutional and recurrent networks for sequence modeling, 2018, http://dx.doi.org/10.48550/arXiv.1803.01271, arXiv:1803.01271.
[20] Y. Xie, Y. Ueda, M. Sugiyama, A two-stage short-term load forecasting method using long short-term memory and multilayer perceptron, Energies 14 (18) (2021) 5873, http://dx.doi.org/10.3390/en14185873.
[21] H. Tchoketch-Kebir, A. Madouri, Research leadership and high standards in economic forecasting: Neural network models compared with etalon arima models, Bus. Ethics Leadersh. 8 (1) (2024) 220–233, http://dx.doi.org/10.61093/bel.8(1).220-233.2024.
[22] A. Lazcano, M.A. Jaramillo-Morán, J.E. Sandubete, Back to basics: The power of the multilayer perceptron in financial time series forecasting, Mathematics 12 (12) (2024) 1920, http://dx.doi.org/10.3390/math12121920.
[23] C.J.B.V. Guimarães, M.A.C. Fernandes, Real-time neural networks implementation proposal for microcontrollers, Electronics 9 (10) (2020) 1597, http://dx.doi.org/10.3390/electronics9101597.
[24] X. Jia, J. Willard, A. Karpatne, J.S. Read, J.A. Zwart, M. Steinbach, V. Kumar, Physics-guided machine learning for scientific discovery: an application in simulating lake temperature profiles, ACM/IMS Trans. Data Sci. 2 (3) (2021) 1–26, http://dx.doi.org/10.1145/3447814.

[25] R. Zhang, Y. Liu, H. Sun, Physics-informed multi-LSTM networks for metamodeling of nonlinear structures, Comput. Methods Appl. Mech. Engrg. 369 (2020) 113226, http://dx.doi.org/10.1016/j.cma.2020.113226.
[26] N. Muralidhar, M.R. Islam, M. Marwah, A. Karpatne, N. Ramakrishnan, Incorporating prior domain knowledge into deep neural networks, in: 2018 IEEE International Conference on Big Data (Big Data), IEEE, Seattle, WA, USA, 2018, pp. 36–45, http://dx.doi.org/10.1109/bigdata.2018.8621955.
[27] T. Fu, T. Zhang, Y. Cui, X. Song, Novel hybrid physics-informed deep neural network for dynamic load prediction of electric cable shovel, Chin. J. Mech. Eng. 35 (1) (2022) 147, http://dx.doi.org/10.1186/s10033-022-00817-x.
[28] T. Erez, Y. Tassa, E. Todorov, Simulation tools for model-based robotics: Comparison of Bullet, Havok, MuJoCo, ODE and PhysX, in: 2015 IEEE International Conference on Robotics and Automation, ICRA, IEEE, Seattle, WA, USA, 2015, pp. 4397–4404, http://dx.doi.org/10.1109/ICRA.2015.7139807.
[29] J. Collins, S. Chand, A. Vanderkop, D. Howard, A review of physics simulators for robotic applications, IEEE Access 9 (2021) 51416–51431, http://dx.doi.org/10.1109/ACCESS.2021.3068769.
[30] M. Servin, T. Berglund, S. Nystedt, A multiscale model of terrain dynamics for real-time earthmoving simulation, Adv. Model. Simul. Eng. Sci. 8 (1) (2021) 11, http://dx.doi.org/10.1186/s40323-021-00196-3.
[31] Algoryx, AGX dynamics - algoryx, 2025, https://www.algoryx.se/agx-dynamics/.
[32] C. Aluckal, R.V.K. Lal, S. Courtney, Y. Turkar, Y. Dighe, Y.-J. Kim, J. Gemerek, K. Dantu, TERA: A simulation environment for terrain excavation robot autonomy, 2024, http://dx.doi.org/10.48550/arXiv.2501.01430, arXiv:2501.01430.
[33] K. Aoshima, M. Servin, Examining the simulation-to-reality gap of a wheel loader digging in deformable terrain, Multibody System Dynamics 64 (2025) 121–148, http://dx.doi.org/10.1007/s11044-024-10005-5.
[34] K. Aoshima, High-Performance Autonomous Wheel Loading: A Computational Approach (Ph.D. thesis), Department of Physics, Umeå University, Umeå, 2025.
[35] S. Blouin, A. Hemami, M.G. Lipsett, Review of resistive force models for earthmoving processes, J. Aerosp. Eng. 14 (3) (2001) 102–111, http://dx.doi.org/10.1061/(asce)0893-1321(2001)14:3(102).
[36] J.A. Marshall, P.F. Murphy, L.K. Daneshmend, Toward autonomous excavation of fragmented rock: Full-scale experiments, IEEE Trans. Autom. Sci. Eng. 5 (3) (2008) 562–566, http://dx.doi.org/10.1109/tase.2007.912709.
[37] M.D. Weir, J. Hass, G.B. Thomas, Thomas' Calculus: Early Transcendentals, thirteenth ed., Pearson, Boston, 2014.
[38] Algoryx, AGX terrain, 2025, https://www.algoryx.se/documentation/complete/agx/html/doc/UserManual/source/agxTerrain.html#terrain-properties.
[39] M. Pogulis, M. Servin, Local particle refinement in terramechanical simulations, J. Terramechanics 120 (2025) 101083, http://dx.doi.org/10.1016/j.jterra.2025.101083.
[40] V. Wiberg, M. Servin, T. Nordfjell, Discrete element modelling of large soil deformations under heavy vehicles, J. Terramechanics 93 (2021) 11–21, http://dx.doi.org/10.1016/j.jterra.2020.10.002.
[41] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, PyTorch: An imperative style, high-performance deep learning library, 2019, http://dx.doi.org/10.48550/arXiv.1912.01703, arXiv:1912.01703.