IEEE Transactions on Field Robotics

Received 17 October 2023; revised 24 March 2024 and 20 June 2024; accepted 4 July 2024.
Date of publication 31 July 2024; date of current version 14 October 2024.
This article was recommended by Executive Editor Sanjiv Singh.

Digital Object Identifier 10.1109/TFR.2024.3432508

Reinforcement Learning-Based Bucket Filling for Autonomous Excavation

PASCAL EGLI, LORENZO TERENZI, AND MARCO HUTTER (Member, IEEE)

Robotic Systems Laboratory, ETH Zürich, 8092 Zürich, Switzerland

CORRESPONDING AUTHOR: PASCAL EGLI (pasegli@ethz.ch)

This work was supported by the Swiss National Science Foundation through the National Centre of Competence in Digital Fabrication (NCCR dfab).

(Regular Article)

ABSTRACT This article presents a bucket-filling controller for autonomous excavation. The key innovation of this controller is that it can react to the encountered soil conditions and adapt the excavation behavior online without the explicit knowledge of soil properties while respecting machine limitations to avoid stalling. At the same time, the controller takes into account the current terrain elevation and adheres to a maximum-depth constraint to achieve a desired design. The controller is trained entirely in simulation with reinforcement learning (RL). A simple analytical soil model based on the fundamental equation of Earth moving (FEE) is used to simulate ground interactions. To learn an appropriate excavation strategy for a wide variety of scenarios, soil parameters, as well as other properties of the environment, are randomized extensively during training. We test and evaluate the controller on a 12-ton excavator with a conventional two-stage hydraulic system in a wide range of different soil conditions. In addition, we show the excavation of a complete trench by integrating the controller into an autonomous excavation planning system. The experiments demonstrate that the controller can robustly adapt the excavation trajectory based on the encountered conditions and shows competitive performance compared to a professional machine operator. A video is available at https://youtu.be/rQKUk9nKaCk.

INDEX TERMS Autonomous excavation, hydraulic actuators, reinforcement learning (RL).

NOMENCLATURE

$\boldsymbol{\tau}_t$	Joint torques.
$\mathbf{q}_t$, $\dot{\mathbf{q}}_t$	Joint positions and velocities.
$\mathbf{a}_t$	Policy actions (normalized joint-velocity references).
φ_t	Bucket AoA.
${}_{\mathcal{G}}\mathbf{r}_{B,t}$	Bucket-edge position.
${}_{\mathcal{G}}\mathbf{v}_{B,t}$	Bucket-edge linear velocity.
${}_{\mathcal{G}}\phi_{B,t}$	Bucket orientation.
${}_{\mathcal{G}}\boldsymbol{\omega}_{B,t}$	Bucket angular velocity.
${}_{\mathcal{G}}\mathbf{v}_{C,t}$	Excavator base linear velocity.
${}_{\mathcal{G}}z_S({}_{\mathcal{G}}\mathbf{r}_B)$	Soil height at bucket position.
${}_{\mathcal{G}}z^*_S({}_{\mathcal{G}}\mathbf{r}_B)$	Maximum-depth height base at bucket position.
$\mathbf{n}_S({}_{\mathcal{G}}\mathbf{r}_B)$	Soil normal vector at bucket position.
V_t	Bucket-fill ratio.
PD	Sampled pullup distance.
c_f	Curriculum level.
$V_T(c_f)$	Terminal fill ratio.
$V_{C,T}(c_f)$	Terminal fill ratio bucket closing.
$\mathrm{PZ}_T(c_f)$	Terminal pullup zone width.
$H_T(c_f)$	Terminal height above lowest trajectory point.

I. INTRODUCTION

AUTOMATION in the construction industry, one of the largest industry sectors worldwide, will become increasingly important in the near future to alleviate the prevalent shortage of trained workforce [3] and thus improve the productivity [52]. In addition, automation on construction sites can help reduce the high numbers of fatal work injuries [68]. In the longer run, also space applications seem to become increasingly relevant [21], [67].

This work addresses the automation of excavation, one of the most common tasks on construction sites. Given a desired state of the terrain, e.g., a pit or a trench represented as a georeferenced map, the mission consists of achieving the goal state with an excavator fully autonomously. A common way of solving this task is to split it into tractable submodules,

FIGURE 1. Same controller adaptively excavates in different conditions without having prior knowledge about soil properties. (a) In soft soil, the controller exhibits a *penetrate and rotate* behavior. In (b) and (c), the controller adapts the excavation trajectory to the terrain slope to avoid excavating in the air or penetrating too deep toward the end of the trajectory (soft soil sloped upward from the excavator and soft soil sloped downward from the excavator, respectively). (d) and (e) In hard soil, the controller performs a *penetrate and drag* behavior to avoid pulling the machine toward the bucket or lifting it off the ground and closes the bucket only in front of the excavator to maximize the worked area (concrete floor, and hard and compacted stony soil, respectively). A large bucket AoA results in a large pressure exerted on the ground to loosen up the soil. (f) Maximum-depth constraint can be set to achieve the desired excavation shape.

which are coordinated by some form of state machine [33], [64], [66], [70], [74], [79]. Individual states consist of driving the machine to a sequence of positions so that the entire design can be covered, finding and moving the bucket to the excavation point, excavating, finding a dump location, dumping the excavated soil, and monitoring the progress. In this article, we present a controller for the excavation phase of a complete autonomous excavation system. This phase is especially demanding because it requires controlling a heavy hydraulic machine with highly nonlinear dynamics as well as its interaction with a priori unknown soil. In addition, task-specific requirements need to be satisfied to successfully excavate the desired designs.

Soil properties on an excavation site can vary significantly, even on a small area and at different depths, such that it becomes intractable to identify soil parameters of an entire site (Fig. 1). This requires a controller that can react to the encountered soil properties and adapt the excavation behavior

online without prior information about the soil conditions. On one side of the spectrum, in extremely hard soil, the bucket can barely penetrate the soil without stalling or lifting the machine off the ground. The desired behavior is to scrape over the ground while applying as much pressure as possible to loosen up the soil, which results in a long excavation path at a shallow depth. On the opposite side, in very soft soil that offers little resistance, the most efficient strategy is to perform a short and deep trajectory. These two extremes are referred to as *penetrate and drag* and *penetrate and rotate (scoop)* [9]. However, depending on the soil properties, the optimal excavation path lies somewhere in between these extremes. Also, the controller needs to take into account the current terrain elevation to avoid penetration that is too deep or too shallow. Both cases are not ideal as they can lead to overfilling or stalling, or inefficient, incomplete filling. In addition, to achieve the desired terrain shape, the controller should excavate freely if the bucket is far away from the desired depth for maximal efficiency but follow the design when the bucket reaches it. Combining these criteria optimally is difficult even for human operators and requires many years of practice. Therefore, we propose a reinforcement learning (RL)-based approach that allows us to flexibly combine these disparate objectives and different sensor modalities. The RL agent discovers the ideal behavior in a wide range of different soil conditions by interacting with the system through trial and error.

A. RELATED WORK

Over the past decades, the automation of excavators has been addressed in numerous contributions. In particular, also the automation of the bucket-filling phase has been the focus of many works. However, as outlined in the following, autonomous excavators still lag behind the performance and versatility of human operators, which hinders the adoption by the industry [29].

A natural method for automating the bucket-filling process is to handcraft a reasonable bucket trajectory through the soil [36], [60], [74]. This method is effective if soil properties are consistent but fails if soil conditions change.

The excavation phase can be further broken down into segments consisting of penetration, dragging, scooping, and lifting. To reduce manual tuning effort, Sandzimier and Asada [55] proposed a controller that switches automatically from dragging to scooping to achieve a more precise bucket filling. The key parameters, particularly the excavation depth of the trajectory, are still hand-tuned for a specific soil type. To alleviate this issue, Sotiropoulos and Asada [62] proposed a method to regulate the bucket height during the dragging phase to apply maximum power to the soil, which is important for efficiency and avoids stalling. However, the remaining parts of the trajectory are still manually calibrated, and it is unclear if the method works on a real hydraulic excavator with low control bandwidth and dead zones, where an accurate estimation of the force gradient is challenging. Also, how to use this controller to excavate a desired design has not yet been addressed. Sotiropoulos and Asada [63] also proposed a model predictive control (MPC) to track the desired shape. The MPC uses a soil-bucket model based on the Koopman theory that is trained on data from excavation experiments in a single soil type. However, the controller relies on accurate force control, which requires expensive modifications to the hydraulic system [25]. Also, how the method can be extended to different soil types is left for future work. Jud et al. [30], [33] achieved a soil-adaptive behavior in different soils on a full-sized excavator by defining a bucket-force trajectory for each of the excavation phases, and a target shape could be accurately excavated by switching to position control if the bucket comes close to the desired height. Accurate force control was achieved by replacing the standard main-stage valves with high-performance servo valves. Another approach to avoid stalling when following a predefined bucket trajectory is to use impedance control to track the desired bucket force [51]. However, this leads to incomplete bucket filling in soil that is harder than assumed. This issue was addressed by Maeda et al. [43], who proposed an iterative learning controller with a disturbance observer. It assumes, however, similar (near-repetitive) soil responses, which is not the case in general. Similarly, Park et al. [48] used online learning to improve the tracking performance over time in a repetitive excavation task.

Rule-based methods are yet another approach to avoid stalling by applying handcrafted correction behaviors if the interaction forces become too large [47] or the tracking error increases [13]. Some researchers have elaborated on the idea of defining a database of motion primitives that are selected or combined by a rule-based system to solve autonomous excavation tasks [9], [19], [49], [57], [70]. While such systems were successfully deployed on real machines, creating the motion primitives requires substantial engineering effort and heavy domain knowledge, especially for more complex tasks.

To transition away from handcrafted bucket paths, trajectory optimization (TO) techniques have been extensively studied for autonomous excavation. Purely kinematic approaches have been proposed to optimize objectives such as the desired amount of scooped soil volume, time, smoothness, or bucket AoA [71], [77]. While these methods can take into account the current and desired elevation and are real-time capable, they cannot guarantee that the trajectory is feasible because soil reaction forces are not considered. Therefore, Lee et al. [39] proposed a dynamics-aware MPC with a disturbance observer to track a kinematically optimized trajectory and tested it in simulation. However, if the disturbances become larger, i.e., if the soil is harder than the expected one, the deviation from the optimized trajectory increases, leading to incomplete bucket filling and, thus, reduced efficiency because the trajectory is not updated online. Other proposed TO approaches consider soil properties by integrating soil models into the optimization,

which, however, increases the computation time and renders them incapable of running in real time [35], [72], [75], [82]. To mitigate long computation times during deployment, Yao et al. [73] distilled the results of TO into a neural network that can be inferred in only a few milliseconds. The challenge of TO methods, including soil models, is that soil parameters must be known in advance. One way of obtaining these parameters is through laborious geological site inspection [82]. Another way is to optimize the model parameters by minimizing the error between predicted and measured soil reaction forces during manual operation of the machine [35] or controlled laboratory experiments [1]. Zhao et al. [80] trained a model in a supervised fashion to predict soil parameters directly from the measured soil resistance while excavating preclassified soils. These soil parameter estimation methods heavily depend on the amount and quality of the datasets, which are time-consuming to collect. It remains to be clarified whether the predictions generalize to different machines or if new data need to be collected each time.

Instead of using TO to generate an excavation trajectory, another stream of research focuses on leveraging expert demonstrations. These were reparameterized [78] or directly optimized stochastically [20] for kinematic objectives such as smoothness or speed, however, without considering soil reaction forces, which can lead to stalling. This can simply be avoided by defining a force threshold and limiting the admissible excavation area but can lead to inefficient excavation [24]. Son et al. [61] proposed modulating the trajectory depth to avoid stalling, which, however, leads to incomplete bucket filling if the interaction forces become larger because the target endpoint remains fixed. Zhu et al. [81] used offline RL and defined a cost to avoid large reaction forces in the optimization of expert demonstrations, which in general leads to trajectories with lower torques but does not guarantee the feasibility of the trajectory. In the same way, Jin et al. [28] used offline RL to reduce forces and maximize the bucket depth during the penetration phase. Not explicitly relying on *expert* demonstrations, but still requiring data collected on the real system [56] or in simulation [40], visual prediction models of the excavation scene were trained and then leveraged with sampling-based optimization to find actions that achieve the desired state of the scene, however, without considering interaction forces. Tahara et al. [65] proposed an improved imitation learning method to leverage also nonoptimal demonstrations and applied it to a small-scale excavator model for excavating granular soil. The common challenge of these methods is that they heavily depend on the quality, quantity, and variety of demonstrations, and it is unclear if or how the datasets can be used to synthesize controllers for different excavators.

To avoid the dependence on time-consuming and expensive data collection on real machines, RL in simulation provides a tempting alternative. Lu et al. [41] used RL in simulation to train a policy for excavating rigid objects with a Franka Panda arm based on a visual representation of the excavation scene. However, because soil interaction forces were not considered, the policy would often get stuck during deployment due to the mismatch between simulation and reality. High-fidelity simulators can simulate the excavation process more accurately but are computationally expensive, which limits their practicability to train excavation policies for a wide variety of soils, and a sim-to-real transfer has not yet been demonstrated [38], [45], [46].

RL-based controllers have shown state-of-the-art performance in other domains of robotics such as legged locomotion [22] or drone racing [34]. Due to the large amount of required training data, the prevalent approach is to train the controllers in simulation and subsequently deploy them on the real hardware. However, successful sim-to-real transfers for ordinary construction machines are still scarce due to the tradeoff between simulating such machines accurately enough and computational complexity. Besides excavation, simulation-only results of RL-based controllers for construction machines have been reported for wheel loaders [5], [23], walking excavators [4], or forestry cranes [2]. A successful sim-to-real transfer of an RL-based controller for driving a full-scale forestry machine with active suspension across uneven terrain was shown by Wiberg et al. [69]. A key element in this work was an accurate machine model identification and low-level controllers that were tuned independently for simulation and the real machine. However, slow learning was reported due to the accurate and, hence, computationally expensive simulation, which limited the achievable complexity of the task. Also, deployed on the real machine, but for a relatively constrained task, Samtani et al. [54] trained an RL controller with two binary actions to push against a rock and trigger the activation of a hammer tool attached to an excavator for rock breaking. The sim-to-real gap was bridged by incorporating actuation delays into the simulation and by filtering measurements during deployment. Dadhich et al. [11] used RL directly on the machine to refine a bucket-filling controller for a wheel loader that has been trained from expert demonstrations, avoiding simulation completely. The method, however, relies on expert data, which is time-consuming to obtain in sufficient quantities for complex tasks. In our previous work [15], we used RL to accurately control an excavator arm for surface leveling (grading) operations. Critical for bridging the sim-to-real gap and achieving competitive accuracy was augmenting the simulation with an actuator network trained on real-world data. Similarly, Dhakate et al. [12] demonstrate a sim-to-real transfer of an RL-based inverse kinematics (IK) controller for a redundant hydraulic manipulator but relied on manually tuned low-level controllers, which compromised the tracking accuracy. For full-fledged excavation, training an actuator network would require much more extensive data collection in different types of soils to avoid limiting the generalizability of the controller.

In our previous work on excavation [14], we used a simple, analytical soil model based on the fundamental equation of Earth-moving (FEE) to train an RL controller for excavation. The computational affordability of the model allowed for compensating for the reduced accuracy with heavy domain randomization and for training a soil-adaptive policy in only a few hours on a single workstation. The policy could be transferred to the real machine and showed robust excavation in a wide range of different soils. Using joint velocities as controller outputs made a more accurate machine model redundant. However, this policy has limitations that prevent it from being used to excavate a multiscoop task. Specifically, the policy is blind and assumes flat terrain. Also, it does not consider the desired excavation shape, and joint torque limits are assumed to be constant, which is not true because of the nonlinear piston-joint linkage configuration and compromises the performance.

B. CONTRIBUTION

The contribution of this work is a bucket-filling controller for an autonomous excavation system that can adapt online to varying soil hardness without requiring explicit knowledge about soil parameters. The excavation behavior is also adapted according to the current terrain elevation, which is mapped online with a light detection and ranging (LiDAR) sensor. In addition, the controller adheres to a maximum-depth constraint used to excavate the desired terrain shape and respects machine limitations to avoid stalling or lifting the machine off the ground. The controller is trained in simulation using RL and deployed on a full-sized hydraulic excavator with a conventional two-stage hydraulic actuation, demonstrating a successful sim-to-real transfer of an RL policy for an ordinary construction machine. We show extensive experiments in diverse terrains as well as the excavation of an entire trench by integrating the bucket-filling controller presented here into a complete autonomous excavation system. An in-depth comparison to a professional human operator indicates the competitive efficiency and precision of the proposed approach.

II. METHOD

The bucket-filling controller is trained with RL entirely in simulation. Therefore, the control problem is formulated as a discrete-time Markov decision process (MDP), where the agent interacts with an environment. In our setup, *agent* stands for the control policy and *environment* stands for the excavator and the terrain. The complete state of the environment at time step t, including soil parameters, is represented by $\mathbf{s}_t \in \mathcal{S}$. At every time step, the agent observes a subset of the complete state, e.g., no soil parameters, $\mathbf{o}_t \in \mathcal{O} \subseteq \mathcal{S}$, takes action $\mathbf{a}_t \in \mathcal{A}$ and receives a scalar reward $r_t(\mathbf{s}_t, \mathbf{a}_t) \in \mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. The agent acts according to a stochastic policy $\pi_{\boldsymbol{\theta}}(\mathbf{a}|\mathbf{o}_t) \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{o}_t), \boldsymbol{\sigma}^2)$, where $\boldsymbol{\mu}_{\boldsymbol{\theta}}$ is represented as a neural network with parameters $\boldsymbol{\theta}$ and $\boldsymbol{\sigma}^2$ is the observation-independent variance. Its objective is to learn a policy that maximizes the infinite-horizon return $\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$ by interacting with the environment. The discount factor $\gamma \in [0, 1)$ trades off between current and future rewards and renders the return numerically finite. The remainder of this section outlines the details of the MDP and the training procedure.

A. SIMULATION SETUP

1) EXCAVATOR

The excavator is simulated as a serial manipulator with a floating base in NVIDIA's Isaac Gym 4, a GPU-accelerated physics engine [44], which allows for a fast, parallelized data collection (Fig. 2). The agent actuates the four arm joints of the excavator. All the other joints, in particular the cabin turn, and since we use a walking excavator in this study also the leg joints, are fixed. The machine is supported by flat ground with a friction coefficient of $\mu = 0.8$. The agent outputs joint-velocity references. These are tracked with an inverse dynamics (ID) controller according to (1) that is manually tuned for each joint in simulation. Thereby, $\boldsymbol{\tau}$ are the applied joint torques; $\mathbf{M}(\mathbf{q})$, $\mathbf{g}(\mathbf{q})$, $\mathbf{J}_p(\mathbf{q})$, and $\mathbf{J}_r(\mathbf{q})$ are the configuration-dependent generalized mass matrix, gravity terms, translational, and rotational Jacobian, respectively; $\mathbf{F}_{\text{ext}}$ and $\mathbf{M}_{\text{ext}}$ are external forces and momentum resulting from the soil interaction, respectively; $\dot{\mathbf{q}}$ and $\dot{\mathbf{q}}^*$ are measured and desired joint velocities, respectively; and $\mathbf{K}_d$ is a diagonal matrix with tunable gain parameters. Coriolis and centrifugal terms are omitted for simplicity, as they are small for the relatively low velocities used for excavation

$$\begin{aligned} \boldsymbol{\tau} &= \mathbf{M}(\mathbf{q})\,\ddot{\mathbf{q}}^* + \mathbf{g}(\mathbf{q}) + \mathbf{J}_p^T(\mathbf{q})\,\mathbf{F}_{\text{ext}} + \mathbf{J}_r^T(\mathbf{q})\,\mathbf{M}_{\text{ext}} \\ \ddot{\mathbf{q}}^* &= \mathbf{K}_d\left(\dot{\mathbf{q}}^* - \dot{\mathbf{q}}\right). \end{aligned} \tag{1}$$

Joint torques and velocities are limited according to the cylinder specifications, which are geometrically transformed into joint space. This highly nonlinear conversion results in significantly different limits depending on the arm configuration, which can be seen later in Section III.

2) SOIL MODEL

The excavation behavior heavily depends on the soil characteristics. Instead of relying on a high-fidelity and, hence, computationally expensive soil simulation, we use a fast analytical soil model, which computes the most relevant excavation forces. The reduced complexity is then leveraged and compensated for by generating a vast amount of training data with heavily randomized soil parameters.

The forces resulting from the interaction with the soil and acting on the excavator's bucket are computed based on Park's model [8], which includes two mechanisms: *separation* and *penetration*.

Separation describes the process of breaking and displacing soil. Its computation is based on the FEE introduced by Reece [50]. The original FEE models the separation force for a wide blade moving horizontally through the soil. This model was adapted and improved by various researchers to account for narrow tools, representing the excavation

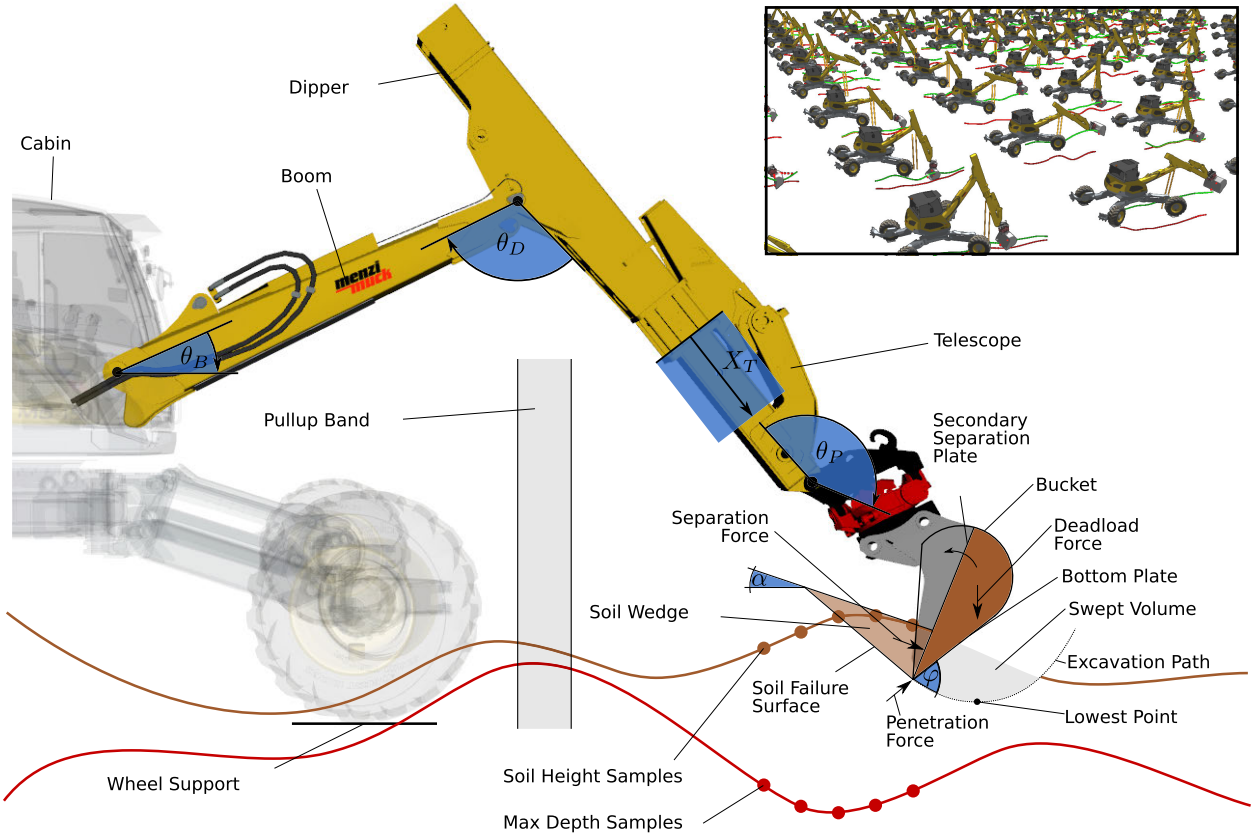


FIGURE 2. Simulation setup. The policy controls the four arm joints θ_B, θ_D, X_T, and θ_P. α is the terrain's slope at the location of the bucket edge, and φ is the bucket's AoA, i.e., the angle between its velocity and the bottom plate. The soil model computes forces for penetration, separation, and the dead load, which act on the bucket and depend on the bucket configuration and soil properties. For separation, the soil failure shape is modeled as a wedge. The swept soil volume is modeled to be evenly distributed in the bucket, resulting in a rising secondary separation plate as the excavation progresses. If the bucket cannot be filled completely, it should be closed within the pullup band. The agent observes height samples from the soil and depth constraint. Top right: batched experience collection in NVIDIA's Isaac Gym.

process with a bucket more accurately [18]. Park [8] models the separation forces according to Perumpral et al. [26], who proposed a simplified failure shape, i.e., a simple wedge, whose shape is determined based on the bucket state and soil parameters. These models can approximate the excavation forces to an accuracy of ∼20% as verified in extensive experiments by Cannon and Singh [10] and Luengo et al. [42].

Penetration describes the movement of the bucket edge straight into the soil. The penetration resistance is computed based on frictional and adhesive forces acting on the bottom plate of the bucket and the cutting edge or teeth. The forces for the bottom plate depend on the passive and lateral Earth pressures, which we compute according to Bennett et al. [7] and JAKY [27]. For the cutting edge, the pressure is higher because the soil is actively deformed and forms a cavity. We simplify Park's model based on the finding that the cavity pressure reaches a limit pressure [76] and assume that this pressure is immediately reached. The factor p_t represents how much larger the cavity limit pressure is compared to the passive Earth pressure and has been determined in field tests [76].

Instead of defining a discrete set of excavation modes such as Park, where one combination of penetration and separation is active at a time, we superimpose both mechanisms smoothly. We scale the penetration resistance with the cosine of the bucket's AoA, which is 0 if the bucket's motion is perpendicular to the bottom plate and 1 if it is parallel.

When the bucket intrudes the soil, we integrate the bucket path and assume that the entire swept volume accumulates in the bucket. For the bucket-filling process, we further assume that the soil evenly distributes within the bucket along a secondary separation plate and neglect compaction. The separation resistance is then computed for this secondary separation plate, which builds up the more the bucket is filled. The shape of the bucket is approximated with a flat bottom plate and a cylindrical back. If the bucket exits the soil and the bottom plate points downward, all the soil is spilled at once, i.e., the bucket fill volume is reset to 0. Since this is undesired and should be avoided by the agent, the simulation is reset if this event occurs, as will be explained later in detail. Therefore, adding the spilled soil to the terrain is not modeled.

TABLE 1. Soil parameters and randomization ranges.

Symbol	Description	Unit	Min	Max
c	Cohesion [27]	kPa	0	105
c_a	Adhesion [17]	kPa	0	c
Φ	Soil internal friction angle [27]	rad	0.3	0.8
γ	Unit weight [27]	kN/m^3	17	22
δ	Soil-bucket friction angle [17]	rad	0.2	0.4
p_t	Cavity pressure factor [76]	-	0	300
l	Terrain shape RBF length scale	-	0	0.5

The terrain shape is modeled by computing randomized soil heights along the boom of the excavator. Thereby, we assume that the height of the soil is constant over the width of the bucket. The height is sampled using RBFs with a Gaussian kernel $\exp(-l\, d(x_i, x_j)^2)$, where $l \geq 0$ is the length scale that controls how strongly different parts of the terrain are correlated and $d(x_i, x_j)$ is the Euclidean distance between two points. The smaller l, the more the samples are correlated, and for $l = 0$, the function collapses to a straight line, i.e., flat soil. For $l = 0.5$, the maximal terrain inclination is $\sim$45°. We chose RBFs for their smoothness, which avoids abrupt changes in the soil force response, and their simplicity of implementation. We anticipate that alternative methods, such as Perlin noise, may yield comparable results.

The relevant components of the soil model are illustrated in Fig. 2. All the soil parameters are summarized in Table 1. The randomization ranges of the soil parameters, except for the terrain's length scale l, are obtained from geotechnical databases.

B. TASK DESCRIPTION

The main objective for the RL agent is to fill the bucket with soil as quickly as possible, starting at the point in the workspace where it is initialized. If the bucket cannot be filled completely, it should finish the excavation at a safe distance from the machine, which we call *pullup distance*. This can occur if the excavation starts close to the excavator's base or the soil is very hard such that only shallow penetration is possible. Finding a potentially better point of attack (PoA) is not part of the learning objective and is assumed to be provided by another system. In addition, a maximum-depth constraint should not be violated, i.e., the agent should excavate freely if far away from the depth constraint but follow it closely if soil properties allow until the bucket is completely filled or inside the final pullup zone.

C. EPISODE INITIALIZATION

The environment is initialized in a randomized state at the beginning of every training episode. A terrain shape is generated by sampling the soil height and soil parameters uniformly within the ranges listed in Table 1. A maximum-depth constraint, upper bounded by the terrain, is sampled also using RBFs with the same length scale l. The pullup distance is sampled within $[2, 3.5]$ m to account for different leg configurations of the excavator if excavating with a rotated cabin. The arm is then initialized with randomized joint states such that the bucket does not violate the depth and pullup constraints and lies within 0.4 m of the terrain surface. If the bucket pose is sampled within the soil, the bucket fill ratio is uniformly sampled between $[0, 1]$. In addition, the machine is pitched in the direction of the boom within $\pm 5°$.

D. REWARDS

To achieve the desired behavior, we formulate a set of reward terms. We distinguish between *per-time-step* and *terminal* rewards, which are received at every time step or only once when a termination condition is satisfied, respectively. Thereby, termination conditions can lead to a positive as well as a negative reward to encourage or discourage reaching certain states. If a termination condition is met, the episode is reset to avoid exploration beyond that state. We use termination conditions to seek or avoid certain states and the per-time-step rewards to guide the agent to or away from these states as well as to shape its behavior. The total reward consists of the sum of all the reward terms explained in detail next. The exact definitions are listed in Tables 2–4.

1) POSITIVE TERMINATION REWARDS

a) T1 Full Bucket

If the bucket is filled enough, pulled up, and closed enough to retain the scooped soil, the agent is awarded $+10$. The desired height is thereby defined as the distance of the bucket edge relative to the lowest point of the excavated trajectory.

b) T2 Partially Filled Bucket

The agent can also receive a positive, however, lower reward of $+5$, if it does not fill the bucket but has at least filled it a bit, pulled up, and closed within the allowed pullup zone.

2) NEGATIVE TERMINATION REWARDS

The agent is penalized with -1, and the episode is terminated if any of the following conditions are met to discourage reaching those states.

TABLE 2. Per-time-step reward definition.

Reward		Definition	$\neq 0$ If ...
R1	Move down	$-0.1\,{}_{\mathcal{G}}\mathbf{v}_{B,t} \cdot \mathbf{n}_S({}_{\mathcal{G}}\mathbf{r}_{B,t})$	$(V_t < 0.01) \wedge (\neg II)$
R2	Filling	$V_t - V_{t-1}$	$\neg(I \vee II)$
R3	Power	$1 \times 10^{-6}(\boldsymbol{\tau} \cdot \dot{\mathbf{q}})$	$\neg(I \vee II)$
R4	Max-depth tracking	$0.04 \exp\left(-0.01({}_{\mathcal{G}}z^*_S({}_{\mathcal{G}}\mathbf{r}_{B,t}) - {}_{\mathcal{G}}\mathbf{r}_{B,t,z})^2\right)$	$\neg(I \vee II)$
R5	Close	$0.1\,{}_{\mathcal{G}}\boldsymbol{\omega}_{B,t,y}$	$(II \vee V) \wedge (\neg IV)$
R6	Smooth action	$-0.0075\,\|\mathbf{a}_t - \mathbf{a}_{t-1}\|_1$	Always

TABLE 3. Termination criteria definition.

Termination		Condition	Reward
T1	Full bucket	$I \wedge III \wedge IV$	10
T2	Partially filled bucket	$II \wedge III \wedge IV \wedge (V_t > 0.01)$	5
T3	Bucket velocity	$\|{}_{\mathcal{G}}\mathbf{v}_{B,t}\|_2 < 0.75\ \mathrm{m\,s^{-1}}$	-1
T4	Bucket AoA	$(\varphi_t < 0.0\ \mathrm{rad}) \wedge ({}_{\mathcal{G}}\mathbf{r}_{B,t,z} - {}_{\mathcal{G}}z_S({}_{\mathcal{G}}\mathbf{r}_{B,t}) < 0\ \mathrm{m})$	-1
T5	Base motion	$\|{}_{\mathcal{G}}\mathbf{v}_{C,t}\|_2 > 0.1\ \mathrm{m\,s^{-1}}$	-1
T6	Max depth	${}_{\mathcal{G}}z^*_S({}_{\mathcal{G}}\mathbf{r}_{B,t}) - {}_{\mathcal{G}}\mathbf{r}_{B,t,z} > 0.05\ \mathrm{m}$	-1
T7	Pullup distance	${}_{\mathcal{G}}\mathbf{r}_{B,t,x} < PD$	-1
T8	Soil spillage	$(V_t = 0) \wedge (V_{t-1} > 0)$	-1
T9	Bucket height above soil	$({}_{\mathcal{G}}\mathbf{r}_{B,t,z} - {}_{\mathcal{G}}z_S({}_{\mathcal{G}}\mathbf{r}_{B,t}) > 0.4\ \mathrm{m}) \wedge (V_t = 0)$	-1
T10	Self-collision	in collision	-1

TABLE 4. Different conditions used to define the termination criteria and reward terms.

Condition		Definition
I	Full enough	Fill ratio $V_t > V_T(c_f)$
II	Close enough	Bucket-edge distance to base $< PD + PZ_T(c_f)$
III	Bucket edge high enough	Bucket-edge height above deepest excavation point $> H_T(c_f)$
IV	Closed enough	$-0.1 < \phi_{B,t,y} < 0.2$ rad
V	Full enough closing	$V_t > V_{C,T}(c_f)$

a) T3 Bucket Velocity

The bucket velocity is constrained to a maximum of 0.75 m·s^{-1}. This value is tuned to achieve reasonably good velocity tracking on the real machine while still maintaining an efficient working speed. This constraint provides a simple way of addressing the hydraulic coupling through a common supply that hinders moving all the joints at full speed simultaneously.

b) T4 Bucket AoA

We define the AoA (φ) as the angle between the bucket edge's motion and its bottom plate (Fig. 2). If the bucket is inside the soil, the AoA must not be negative to avoid pushing the soil with the lower side of the bottom plate.

c) T5 Base Motion

The excavator's base must not move during excavation, especially not lifted off the ground or pulled toward the bucket. Therefore, the episode is terminated if the norm of the linear base velocity exceeds 0.1 m·s^{-1}.

d) T6 Maximum Depth

If the maximum depth is overshot by more than 5 cm, the episode is terminated. This tolerance allows for accurate tracking of the desired depth, if possible, which is further encouraged with a tracking reward defined next. Perfect tracking would not be possible without this tolerance because of the exploration noise added to the agent's actions inherent to RL, which leads to the agent staying at a safe distance from the constraint.

e) T7 Pullup Distance

The episode is terminated if the bucket gets closer to the excavator's base than the sampled pullup distance.

f) T8 Soil Spillage

According to the soil model, all the soil is spilled if the bucket is pulled out of the soil and points downward. If this happens, the episode is terminated.

g) T9 Bucket Height

To limit the search space and accelerate training, the episode is terminated if the agent moves the empty bucket more than 0.4 m above the terrain.

h) T10 Self-Collisions

If self-collisions are detected, particularly between the bucket and the boom, the episode is terminated.

3) PER-TIME-STEP REWARDS

a) R1 Bucket Down

To quickly start the excavation at the initialization point, a reward is given proportional to the velocity in the direction perpendicular to the terrain surface if the bucket is empty and the bucket is not already close enough to the excavator.

b) R2 Bucket Filling

Bucket filling is encouraged with a reward proportional to the scooped soil volume per time step. It is set to 0 if the bucket is full or if the bucket is close enough to the desired pullup distance.

c) R3 Power

In addition to the bucket-filling reward, we add a reward for maximizing the power, the dot product of joint velocities and joint torques, applied to the ground to encourage deep and efficient excavation [62]. The reward is set to 0 if the bucket is full or close enough to the excavator.

d) R4 Maximum-Depth Tracking

Tracking of the maximum depth is incentivized with a reward proportional to $\exp(-ce^2)$, where c is a tuning parameter and e is the distance between the bucket tip and the maximum depth. This reward is also set to 0 if the bucket is full or close enough to the excavator.

e) R5 Bucket Closing

Bucket closing is encouraged with a reward proportional to the bucket's curling velocity if the bucket is full enough or already close enough and not already closed enough.

f) R6 Action Rate

To ensure smooth actions, the first-order approximation of its derivative is penalized proportionally.

TABLE 5. Linearly interpolated environment parameters according to the curriculum factor c_f, where N_T is the total number of episodes terminated, and N_{T1} and N_{T2} are the number of episodes terminated with criteria $T1$ and $T2$, respectively.

Parameter		Symbol	Bounds]
C1	Terminal fill ratio	$V_T(c_f)$	[0.5, 1.0]
C2	Terminal bucket height	$H_T(c_f)$	[0, 0.8] m
C3	Fill ratio bucket closing	$V_{C,T}(c_f)$	[0.4, 0.9]
C4	RBF length scale	l	[0, 0.5]
C5	Pullup zone width	$PZ_T(c_F)$	[1.0, 0.3] m

$$c_f \in [0,1] \mathrel{+}= \begin{cases} \frac{1}{250}, & \text{if } \frac{N_{T1}+N_{T2}}{N} > 0.5 \\ 0, & \text{otherwise} \end{cases}$$

E. CURRICULUM

Curriculum learning [6] has established itself to train complex RL tasks. The idea is to make the task easier at the beginning of training when the agent's performance is poor, i.e., merely random actions, and increase the difficulty gradually. Therefore, we define a difficulty level $c_f \in [0, 1]$. The level is increased by $1/250$ after an algorithm iteration if more than 50% of the environments terminated with a positive reward and otherwise kept constant. The aggressiveness of the curriculum is a tuning factor that trades off learning speed versus stability. With the reported settings, it reaches the maximal difficulty after around 500 algorithm iterations. Based on c_f, five environment or reward parameters are linearly interpolated within the bounds specified in Table 5.

a) C1 Terminal Fill Ratio

This is the minimum bucket fill ratio for the agent to complete the task with maximal reward (T1). It is increased from half full to completely full.

b) C2 Terminal Bucket Height

The bucket's minimum height above the deepest point of the excavation trajectory is increased from 0 to 0.8 m.

c) C3 Fill Ratio Bucket Closing

The agent is rewarded for closing the bucket if it has at least filled the bucket with this ratio. It is increased over time to encourage initiating closing later when the bucket is more filled.

d) C4 Terrain and Maximum-Depth Shape

The difficulty of the task is increased by generating terrains and maximum-depth constraints with larger variations. Therefore, the length scale factor l used to generate the RBF is increased from 0, i.e., flat terrain, to 0.5.

TABLE 6. Policy observations and actions. Dimension in brackets.

Observations o_t (41)	Actions a_t (4)
Arm joint torques (4)	Arm joint vel. (4)
Arm joint kinematics (pos., vel.) (8)	
Previous joint-vel. command (4)	
Bucket fill ratio (1)	
Soil height (5)	
Max. depth (5)	
Pullup distance (1)	
Cabin pitch ang., pitch ang. rate in arm direction (2)	
Bucket depth (1)	
Bucket lin./ang. pos./vel. in cabin frame (6)	
Bucket lin. vel. norm (1)	
Bucket angle of attack (φ) (1)	
Bucket joint lin. vel. (2)	

e) C5 Pullup Band

The pullup band defines the range in which the bucket needs to be placed to be considered close enough. It is narrowed over the course of training to force the agent to maximize the usage of the available workspace.

F. ACTIONS AND OBSERVATIONS

Most excavators are equipped with a load sensing (LS) system, which increases the fuel efficiency by only raising the system pressure when the load increases. LS also facilitates the operation because it decouples joystick commands and cylinder speed from the load. The cabin turn joint is usually not used during excavation, as it is relatively weak but fast and optimized for quick dumping of the soil to the side. Therefore, a natural choice of policy actions ($\mathbf{a}_t$) is velocity commands for the arm joints, specifically joint-velocity references, to avoid the conversion from cylinder to joint space during training.

As observations ($\mathbf{o}_t$), the agent receives noise-free kinematic joint states and joint torques directly from the rigid-body simulation. Joint torque observations are particularly important for learning a soil-adaptive behavior and allow for an implicit estimation of soil parameters as discussed in the following. Furthermore, the soil height information is provided with five values sampled starting at the bucket position and moving along the boom in the direction of the cabin with a spacing of 20 cm. The information about the maximum depth is supplied in the same way. As we want the policy to finish the scoop at the desired distance from the base, the pullup distance is also provided as an observation. In addition, information about the bucket state, obtained from forward kinematics, is also provided to accelerate learning.

The observations and actions are normalized with empirical means and standard deviations to accelerate convergence and are summarized in Table 6.

G. TRAINING

The agent is trained with an implementation of proximal policy optimization (PPO) [59] with GAE [58] by Rudin et al. [53]. Policy and value functions are approximated with two separate neural networks, both receiving the same input and having linear output layers. The training hyperparameters are listed in Table 7. Training and experience collection are executed on a single NVIDIA GeForce RTX 3090 GPU and occupy around 5.5 GB of VRAM in total with the reported settings. The policy is trained for 5k iterations, which lasts around 3.5 h.

To investigate the assumption that joint torque observations are necessary for learning a soil-adaptive excavation behavior, we train policies with different combinations of omitting joint torques or adding perfect knowledge about soil parameters explicitly according to Table 8. Fig. 3 shows the mean policy return averaged over five training runs with different random seeds per combination. If neither torques nor soil parameters are observed, the performance is the worst (red). The agent converges to a conservative and inefficient policy, which, even in soft soil, only excavates shallowly because it tries to avoid stalling in all conditions but has no means of inferring how close to the torque limits it operates. When torques and/or soil parameters are observed, the return reaches higher values, and the average maximum penetration depth in soft soil is much larger and similar for the three combinations. In terms of convergence speed, observing joint torques and soil parameters (orange) is the fastest. Observing only the joint torques (blue) leads to faster convergence than only the soil parameters (green). We hypothesize that the mapping from soil parameters and kinematic arm state to the torque limits is more difficult to learn than the mapping from the measured torques and kinematics to the limits. Nevertheless, this shows that knowledge about soil properties is essential for efficient excavation and that the relevant soil

TABLE 7. PPO training hyperparameters.

Parameter	Value
Policy hidden layers	128×128, Leaky Rectified Linear Unit (ReLU)($\alpha = 0.01$)
Value hidden layers	128×128 , Leaky ReLU ($\alpha = 0.01$)
Actor initial noise std.	0.4
Discount factor γ	0.95
Max. episode length	20.0 s
Batch size	$196.6k$
Entropy coefficient	0.01
Learning rate	$5e^{-4}$
Value function coefficient	0.5
Max. grad norm	0.5
GAE λ	0.95
Mini-batches	4
Optimization epochs	5
Clip range	0.2

TABLE 8. Comparison of policies with different observations. The return is averaged over five runs per setting with different random seeds and the last 200 algorithm iterations. The maximum penetration depth is averaged over ∼12.5k randomly initialized episodes, with soft soil and the bucket starting always above the ground. The proposed combination (bold) does not require explicit soil parameter observations and reaches almost identical performance as the policies with access to the soil parameters.

Joint Torques	Soil Parameters	Mean Return	Max Bucket Depth
✔	✘	**2.70**	**27.4 cm**
✓	✓	2.74	28.0 cm
✗	✓	2.66	28.1 cm
✗	✗	2.38	5.0 cm

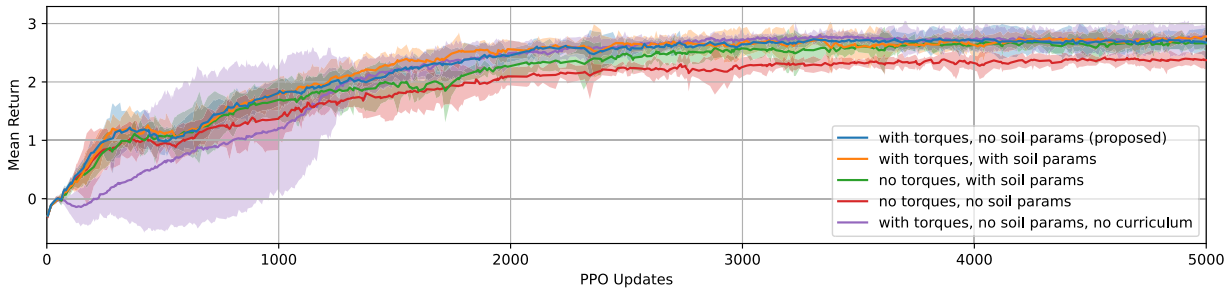


FIGURE 3. Mean return, averaged over five training runs with different random seeds and 95% confident bounds.

parameters can be inferred from joint torque observations. Observing soil properties, explicitly or implicitly, allows the agent to disambiguate the causes for stalling at different penetration depths, hence not avoiding deep excavation in soft soils.

The impact of the curriculum on the training is analyzed by turning off the curriculum and training from the beginning on the most difficult level, i.e., $c_f = 1$. Fig. 3 (purple) shows that the agent is able to fulfill the task successfully and achieves a return of 2.74, similar to the other successful policies. However, while at the beginning of training some policies learning faster than without curriculum, others struggle to achieve positive returns more than 1000 iterations. The policies trained with curriculum show a small dip in return at around iteration 500, where the curriculum reaches the final difficulty level to which the agent has to adapt. We assume that in some cases, adaption to the new level is harder than learning the hardest task directly, which explains why

some policies without curriculum train faster than those with it. Nonetheless, the curriculum stabilizes training, which facilitates the development of the MDP.

III. EXPERIMENTAL RESULTS

In this section, we present the experimental results on a full-sized excavator. First, we analyze the behavior of the proposed controller in various soil conditions and with different constraints. Furthermore, we show autonomous trenching as a practical application case of the bucket-filling controller and compare the performance of the autonomous system to a professional human operator. All the following experiments are performed using the same excavation controller, which in particular, does not observe soil parameters explicitly. A video of the experiments can be found at https://youtu.be/rQKUk9nKaCk.

A. HARDWARE DESCRIPTION AND PREREQUISITES

The experiments are performed on HEAP [32]. HEAP is a Menzi Muck M545, a 12-ton walking excavator retrofitted with various systems and sensors to enable autonomous operation. In the following, we highlight the prerequisites to train the controller in simulation and deploy it on the real machine.

1) MACHINE MODEL

The training of the controller in simulation requires a rigid-body model of the excavator. As opposed to the kinematics, the dynamic parameters are only accurate to a certain degree because the model does not contain parts such as hydraulic hoses or the 220 L of hydraulic fluid in detail. In particular, inertia parameters are difficult to estimate accurately. However, due to the relatively low joint velocities and accelerations during excavation, the predominant contribution to the joint load, besides external forces, stems from the gravitational load, which is independent of the link inertia. Also, friction terms are omitted because dry friction is smaller than 1% of the maximal force for large-scale cylinders, as shown in experiments by Hutter et al. [25].

2) OBSERVATIONS

Critical for deploying the controller is that all the observations provided during training in the simulation are also available on the real machine. Kinematic states of the arm joints and the cabin are obtained through a Leica state estimation system using inertial measurement units (IMUs) on each link. Joint torques $\boldsymbol{\tau}$ are obtained according to (2), where $\mathbf{P}_1$ and $\mathbf{P}_2$ are the hydraulic pressures on either side of the cylinders, $\mathbf{A}_1$ and $\mathbf{A}_2$ are the inner surfaces of the pistons, and $\mathbf{E}(\mathbf{q})$ is the configuration-dependent geometric conversion from piston to joint space

$$\boldsymbol{\tau} = \mathbf{E}\left(\mathbf{q}\right)\left(\mathbf{A}_1\mathbf{P}_1 - \mathbf{A}_2\mathbf{P}_2\right). \tag{2}$$

The terrain is perceived using a LiDAR sensor installed on the roof of the excavator. The point cloud output from the LiDAR is then converted into a 2.5-D elevation map [16] with a resolution of 0.1 m from which the soil height observation is sampled. The scooped soil volume is computed by integrating the proprioceptive bucket path through the soil because the excavation path is occluded from the sensor by the bucket during excavation. In addition, even when moving the bucket away, parts of the terrain might remain occluded, especially when scooping deep close to the machine's base. Therefore, we integrate the proprioceptive information into the elevation map after a scoop. The maximum-depth and pullup-distance constraints are inputs that have to be provided by the user or the system utilizing the bucket-filling controller. The remaining observations can be derived geometrically using the joint-state measurements.

3) JOINT-VELOCITY CONTROL

The M545 is equipped with a two-stage hydraulic system featuring a mechanical pilot stage that steers the main-stage proportional valves and is actuated with joysticks. In addition, the machine is equipped with a hydraulic LS system that only raises the system pressure if required. To allow for automatic control, an electrical pilot stage has been retrofitted to steer the original main-stage valves. Joint-velocity commands provided by the excavation controller are converted geometrically into piston-velocity commands. These are then tracked with proportional–integral–derivative (PID) plus feedforward valve-flow controllers that output valve-current commands at 100 Hz. The feedforward part consists of a lookup table that maps the desired piston velocities to valve-current commands. The lookup table has been collected in a nominal configuration of the excavator's arm and contains 13 points per joint. The LS decouples oil flow, thus speed, from the load such that this approach provides sufficiently accurate velocity tracking also during excavation. Experiments have shown that the velocity control bandwidth lies around 0.5 Hz [32].

B. SINGLE-SCOOP EXPERIMENTS

In this section, we analyze the performance of the proposed controller in different soil conditions and with different constraints. The bucket is placed manually at the desired excavation point and enabled by the operator on the push of a button.

1) DIFFERENT SOIL CONDITIONS

We test the control policy on the real machine in different scenarios, as shown in Fig. 1. The results are shown in the left column of Fig. 4. Even though achieving exactly identical experimental conditions is impossible, we conduct five consecutive trials for each scenario to assess the controller's consistency. The plots show the state of the bucket and the terrain, as well as the bucket fill ratio and the individual joint states for the bold bucket path (the data of the other trials are omitted for the clearness of the plots). The dashed lines in the joint-state plots indicate the joint torque and velocity limits. The joint limits are not constant due to the cylinder-linkage configurations and vary depending on the current arm configuration.

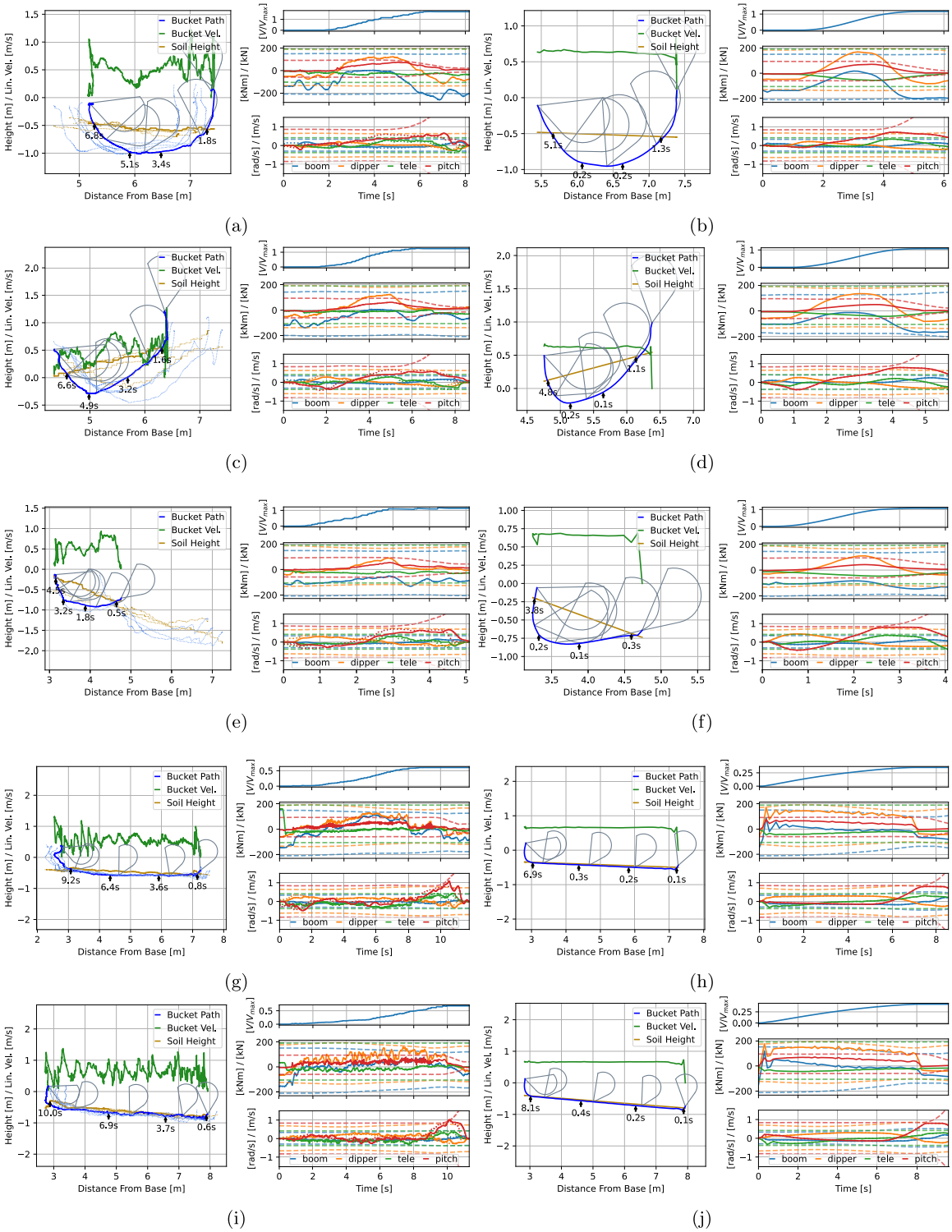


FIGURE 4. Excavation in different conditions on the real machine (left) and in simulation (right). The dashed lines indicate the joint limits, and the dotted lines are the joint-velocity reference provided by the learned controller. (a) Soft soil [Fig. 1(a)]. (b) Soft soil in simulation. (c) Soft soil with a slope toward the excavator [Fig. 1(b)]. (d) Soft soil with a slope toward the excavator in simulation. (e) Soft soil with a slope away from the excavator [Fig. 1(c)]. (f) Soft soil with a slope away from the excavator in simulation. (g) Concrete ground [Fig. 1(d)]. (h) Hardest soil parameters in simulation. (i) Hard ground [Fig. 1(e)]. (j) Hard soil in simulation.

In the right column of Fig. 4, we approximate the real situation in simulation by choosing appropriate soil parameters.

The control policy shows the desired *penetrate and rotate* and *penetrate and drag* in soft soil [Fig. 4(a) and (b)] and hard

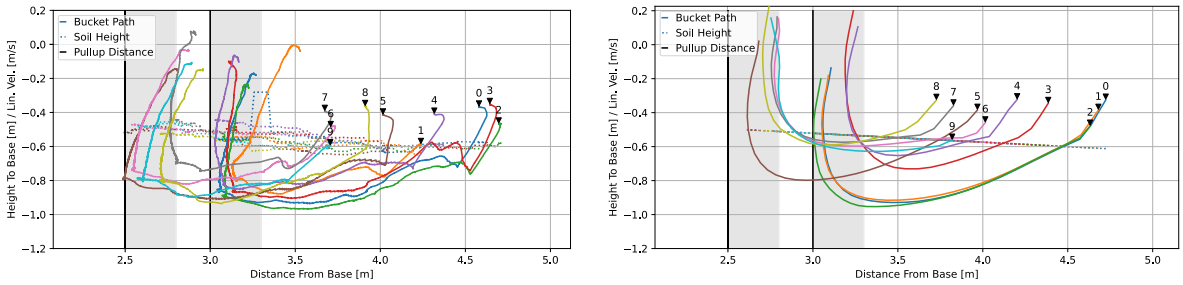


FIGURE 5. Pullup-distance constraints on the real machine (left) and in simulation (right). Scoops 0–4 and 5–9 have a pullup distance of 3 and 2.5 m, respectively. The excavation starts at the number.

soil [Fig. 4(g)–(j)], respectively. In soft soil, the controller achieves complete filling of the bucket and even overfills the bucket slightly. This is the intended behavior since overfilling is not explicitly penalized with the presented reward scheme. In hard soil, the policy cannot fill the bucket completely and pulls up to terminate before colliding with the excavator. Also, the controller maintains a large bucket AoA to maximize the pressure exerted on the ground, facilitating soil loosening. The loosened soil is pushed toward the machine and then finally scooped up by curling the bucket while keeping the edge on the ground to minimize spillage. Moreover, the bucket path is also adjusted according to the terrain slope to avoid excavating in the air or penetrating too deep toward the end of the scoop [Fig. 4(c)–(f)]. The simulation results show good agreement with the real-world experiments, indicating that the simple soil model can sufficiently well approximate the real situation. In particular, also the joint torques are similar and reach values close to the limits, which is important for efficient excavation. Excavation in extremely hard soil [Fig. 4(i)] leads to vibrations with a lower frequency compared to the homogeneous, concrete floor [Fig. 4(g)] due to the larger rocks contained in the soil. Even though this effect is not simulated, the controller generalizes to this scenario. Furthermore, the policy is robust to the inferior joint-velocity tracking performance on the real machine.

2) PULLUP DISTANCE

Here, we test the pullup-distance constraint explicitly by starting the excavation relatively close to the machine. The desired pullup distance is provided as an additional input to the control policy. It is important to avoid self-collisions and can be changed within the range of randomization depending on the configuration of the machine. The controller is trained to finish the excavation with 30 cm in front of the pullup distance.

The results on the real machine and in simulation are shown in Fig. 5. The average bucket fill ratios are 76% (std: 19%) and 39% (std: 36%) for reality and simulation, respectively. This demonstrates that the policy successfully prioritizes pulling up over continuing to fill the bucket (the fill ratio has to be 100% for the maximal reward for a full scoop, see the Nomenclature). Furthermore, in simulation, the scoops always terminate within the allowed band, however, sometimes at the very boundary of the zone. On the real machine, the pullup constraint is only minimally violated once (scoop 5). However, due to the imperfect velocity tracking, it does not always finish precisely within the desired zone. Therefore, we slightly relax the condition when a scoop is considered done for deployment.

3) MAXIMUM-DEPTH TRACKING

In this experiment, we test the tracking capability of a randomized desired depth in soft soil. Soft soil conditions allow the controller to reach the depth constraint without stalling. The agent has been trained to track the desired depth if the bucket edge is in its proximity until the bucket is filled or approaches the pullup distance. Fig. 6 shows the results on the real machine and in simulation. While the tracking performance in simulation is almost perfect with a maximum overshoot of 2 cm, it is slightly worse on the real machine with 5 cm. This is caused by the sim-to-real gap related to joint-velocity tracking, which is also visible at the beginning of, e.g., scoops 2–4, where the bucket moves up after an initially too-fast motion downward.

C. TRENCH EXCAVATION

To test the practicality of the proposed controller, we integrate it into a full-fledged autonomous excavation planning system based on the work of Terenzi and Hutter [66]. Given the desired geometry of an excavation site, the system plans a strategy to achieve the final design fully autonomously. We chose to excavate a trench with a length of 7 m and a depth of 0.75 m. The situation is shown in Fig. 7. In the following, we provide a summary of the excavation planning system, explain the modifications needed to integrate the learned excavation controller, show experimental results on HEAP, and compare the performance of the autonomous system to a professional human operator.

1) AUTONOMOUS EXCAVATION PLANNING SYSTEM

The excavation planning system consists of two main modules: a *global planner* that computes a sequence of excavator poses to cover the desired geometry and a *local planner* that controls the excavation of the current local

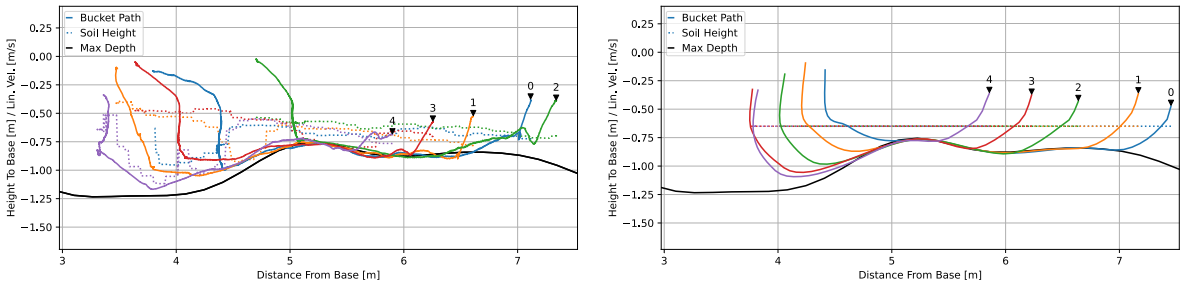


FIGURE 6. **Maximum-depth tracking on the real machine (left) and in simulation (right) in soft soil. Soft soil conditions allow the bucket to reach the maximum depth without stalling. The maximum depth is then tracked until the bucket is filled. The excavation starts at the number.**

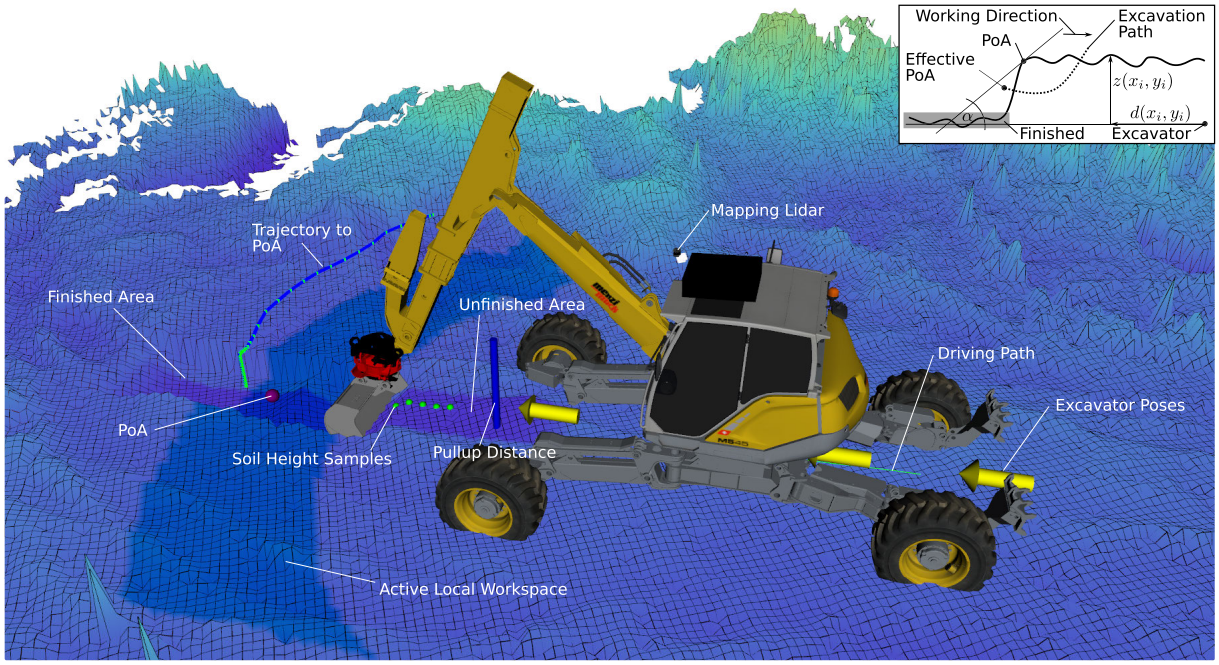


FIGURE 7. **Overview of the autonomous excavation planning system. A georeferenced map of the terrain is prerecorded with a LiDAR sensor and continuously updated. The global planner splits the desired design into arc-shaped local workspaces, which can be excavated without driving. The PoA at which the bucket-filling controller is enabled is found according to the drawing in the top right (adapted from [30]).**

workspace segment. The latter consists of a state machine, where one state consists of the actual soil excavation. It is this state that is replaced with the RL-based controller presented in this work.

In the first step, a georeferenced 2.5-D elevation map of the excavation site is recorded. Therefore, we use the LiDAR sensor mounted on the roof of the excavator. In addition, we use two Global Navigation Satellite System (GNSS) receivers with real-time kinematic (RTK) correction to complete the state estimation of the machine and thus georeference the map. Based on the desired geometry that is drawn into the map, the global planner subdivides the design into local workspaces and computes machine poses from which the local workspaces can be excavated without driving. During excavation, the map is continuously updated at a rate of 1 Hz.

The excavation starts with a rapidly exploring random tree (RRT)-based path planner that computes a feasible driving path to the first excavator pose, which is executed by a pure-pursuit controller. The next step is to find an appropriate point to start the excavation. In [66], the excavation point is found by maximizing the volume of soil that can be scooped. The excavation trajectory is optimized in conjunction using Bayesian optimization (BO), which requires online simulation of excavation trajectories. Besides being computationally demanding, as discussed earlier, this can only lead to efficient excavation paths if the soil conditions are known. Therefore, we use an approach presented by Jud et al. [30] to find the PoA, which is purely geometric and independent of the excavation trajectory. The main idea is to maintain a certain slope (α), and we choose $\alpha = 30°$, at the border between finished and unfinished

terrain (Fig. 7, top right). This prevents collapsing soil and avoids starting the excavation at the bottom of steep sections, which would quickly lead to large excavation forces. To this end, the excavation point is selected according to (3), where (x_i, y_i) are point coordinates in the elevation map, $\mathbf{M}(x_i, y_i)$ is a mask that filters points outside the current workspace or that are already excavated to the required accuracy, $z(x_i, y_i)$ is the terrain height at a coordinate, and $d(x_i, y_i)$ is the Euclidean distance of a coordinate from the location of the excavator. This approach can also be extended to three dimensions [31] if required by the design

$$(x_{\mathrm{PoA}}, y_{\mathrm{PoA}}) = \arg\max_{x_i, y_i} \mathbf{M}(x_i, y_i)\,(z(x_i, y_i) + d(x_i, y_i)\tan\alpha). \tag{3}$$

The PoA is then projected to the closest point on the center line of the trench, and its height, $z(x_i, y_i)$, corresponds to the terrain height. We apply offsets to the PoA to avoid terrain collisions and start the excavation at the *effective PoA*. The bucket orientation is set constant to 0.6 rad. An arm trajectory planner computes a collision-free bucket path to the effective PoA, which is then tracked using a hierarchical IK controller [32]. At this point, the learned excavation controller is enabled and executed until it reaches a desired terminal state, T1 or T2. As opposed to Terenzi and Hutter [66], who find the optimal dumping point by considering no-dump zones and the global working direction, we chose to dump at a fixed distance from the trench for simplicity. The same arm trajectory planner and IK tracking controller are used for dumping. At the end of an excavation cycle, the local planner checks whether the local workspace is completed by comparing the current terrain elevation (z) to the desired geometry (z_{des}). It continues excavating until more than 90% of the elevation map cells are excavated to the desired depth, including an accuracy margin ($\Theta > 0$), i.e., $z - z_{\mathrm{des}} < \Theta$. Both parameters are tuning factors trading off accuracy versus excavation speed. If a local workspace is considered done, the excavator retracts the arm, plans a path to the next global pose, and drives to it. These steps are repeated until the entire design is excavated.

2) MANUAL EXCAVATION BENCHMARK

To better understand the performance of the autonomous system, we asked a foreman and an excavator operator to excavate a trench manually, as it is usually done on construction sites. Both have more than seven years of experience in underground construction. The procedure for excavating such a trench usually starts with marking the trench on the ground with a marking spray or strings. Depending on the requirements of accuracy, the trench is approximately marked according to the construction plan or, more accurately, with GNSS. For simplicity, we mark the trench using the state estimator of the excavator bucket on the ground with a marking spray and define the desired depth relative to a pole next to the trench. The excavation is then a two-person job: one person operates the excavator, and the second person monitors the trench depth with a measuring stick and a bubble level and visually signals the operator to excavate deeper or shallower [Fig. 8(b), top left and middle]. According to these professionals, 3-D guidance systems or total stations, which make manual surveying redundant gain ground, are, however, mostly only used when the excavation task is larger and more complex. For a simple trench, the two-person approach is most prevalent.

3) PERFORMANCE EVALUATION

We show the results of excavating the trench autonomously, with three different values for the accuracy threshold Θ, and compare it to the performance of a professional human operator. To attain comparable results, all the trenches are excavated in the same location, in soil that has already been dug up and, therefore, is relatively homogeneous and loose.

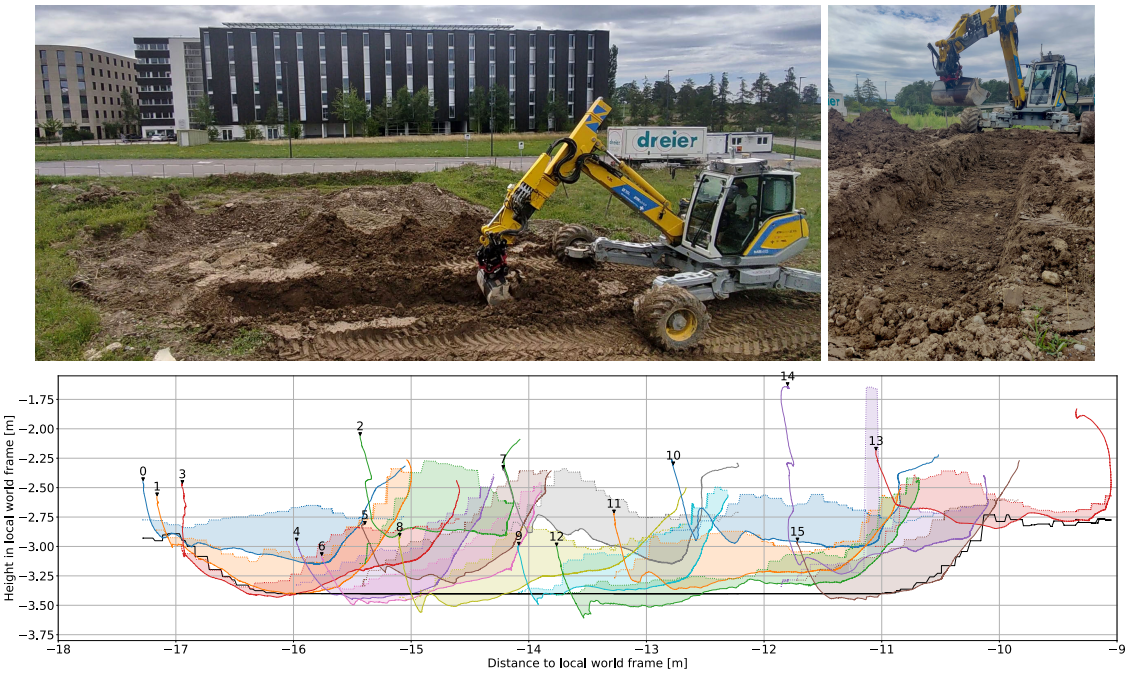
a) Bucket Path

The result of excavating the trench with $\Theta = 15$ cm is shown in Fig. 8(a). It shows a cross section of the trench and the bucket trajectories of the individual scoops, which are numbered in a chronological order and start at the number. The colored solid lines show the bucket-edge path, and the dashed lines show the terrain height before the scoop. The desired depth is drawn in black. The controller adapts the bucket path according to the uneven terrain surface created by preceding scoops (e.g., scoop 1) and the maximum-depth constraint given by the desired design (e.g., scoops 3, 13, and 15) such that the excavation planning system can successfully complete the trench. Also, the excavation controller shows robust behavior to mapping errors, which can occur through soil spillage, and caused the excavation controller to start much higher above the soil than intended (scoop 14). The PoA is selected to maintain a sloped excavation. This can be observed, for example, in scoop 2, which starts closer to the excavator to remove the steep excavation edge created by the preceding scoop.

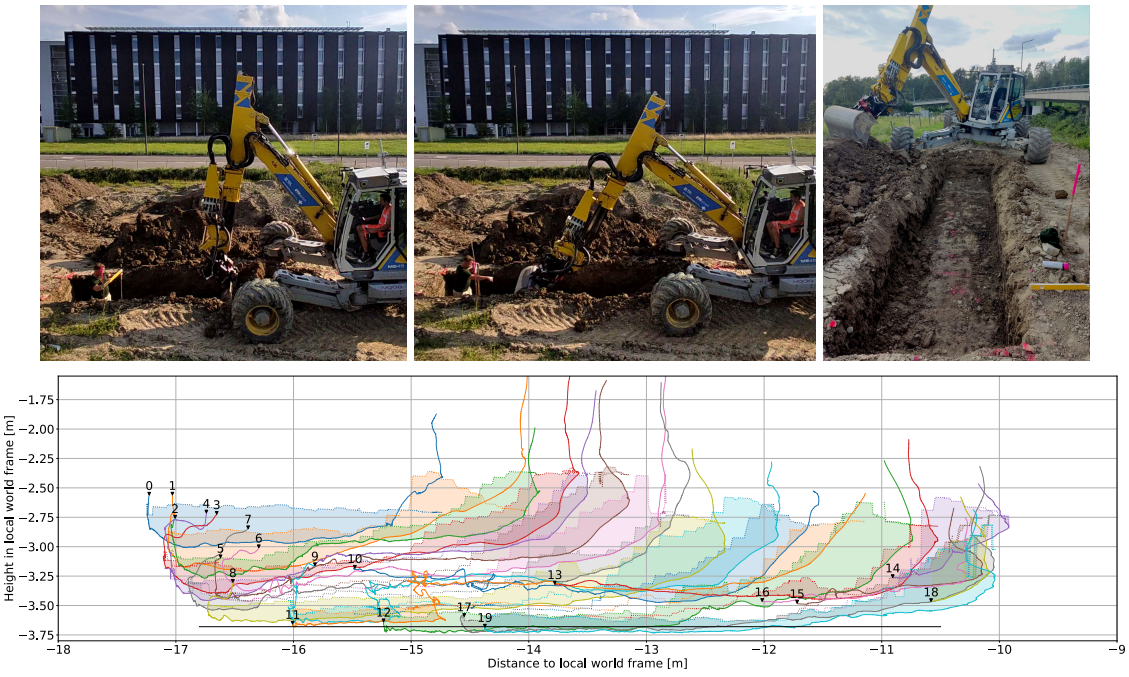
For comparison, the human operator proceeds in a more layered fashion [Fig. 8(b)]. The individual scoops are long and shallow, even though the ground would be soft enough to excavate deeper without stalling. The reason for this is that the operator wants to avoid overshooting the desired depth indicated by the surveyor. Also, it can be observed in the top-right image of Fig. 8(b) that the bottom of the trench is smooth and slightly compacted. The operator achieved this by sliding the bucket with the edge pointing upward, i.e., with a negative AoA ($\varphi < 0$), over the bottom of the trench.

b) Duration Breakdown

Table 9 lists the total and relative time spent in each excavation state. Also, the cycle time, i.e., the sum of *Move to PoA*, *Excavate*, and *Dump*, is further analyzed. For the autonomous system, as expected, the number of scoops and the excavated volume, and therefore also the overall time to complete the trench, increases with lower accuracy thresholds. The relative time spent in each state stays similar.



(a)



(b)

FIGURE 8. Trench excavation. The plots show a cross section of the trench. The individual scoops are numbered in chronological order and start at the number. The colored solid lines show the bucket-edge path, and the dashed lines show the terrain height before the scoop. The desired depth is drawn in black. (a) Autonomous trench excavation with an accuracy threshold $\Theta = 0.15$ m (person in the cabin for safety). (b) Manual trench excavation. In the left and middle images, the surveyor is measuring and signaling the excavator operator to adjust the bucket path with hand signs. The smooth and compacted bottom of the trench was achieved by sliding the back of the bucket over the ground.

Compared to the human operator, the time spent for excavation is significantly shorter for the proposed autonomous solution, while the time to move to the PoA and for dumping is longer. Note that this part was not optimized as part of the present work. The average cycle time, however, is only slightly longer for the autonomous system. Also, the human

TABLE 9. Trench excavation duration breakdown. For the cycle time, mean and std (in brackets) are reported. In this work, only the excavation phase (bold) has been optimized.

	$\Theta = 5$ cm	$\Theta = 10$ cm	$\Theta = 15$ cm	Human Operator
Check Workspace	0.06 min (0.4 %)	0.06 min (0.5 %)	0.05 min (0.5 %)	-
Drive	0.29 min (1.9 %)	0.21 min (1.8 %)	0.37 min (4.0 %)	0.27 min (2.3 %)
Move to PoA	4.93 min (32.8 %)	3.38 min (29.6 %)	2.54 min (27.3 %)	3.00 min (25.9 %)
Excavate	**2.86 min (19.0 %)**	**2.45 min (21.4 %)**	**1.82 min (19.6 %)**	**5.13 min (44.3 %)**
Dump	6.66 min (44.3 %)	5.09 min (44.5 %)	4.28 min (46.0 %)	1.95 min (16.9 %)
Retract Arm	0.23 min (1.5 %)	0.25 min (2.2 %)	0.23 min (2.5 %)	0.20 min (1.7 %)
Survey	-	-	-	1.02 min (8.8 %)
Total Time	15.03 min	11.43 min	9.3 min	11.57 min (100.0 %)
Move to PoA	11.8 (3.7) s	10.7 (2.4) s	9.5 (1.9) s	8.6 (3.4) s
Excavate	**6.9 (0.7) s**	**7.7 (1.0) s**	**6.8 (0.9) s**	**14.7 (7.2) s**
Dump	16.0 (1.4) s	16.1 (1.5) s	16.0 (1.5) s	5.6 (2.0) s
Cycle Time	34.7 (3.8) s	34.5 (3.4) s	32.4 (2.7) s	28.8 (9.15) s
Num. Scoops	25	19	16	20
Volume	10.8 m^3	9.2 m^3	8.4 m^3	11.5 m^3

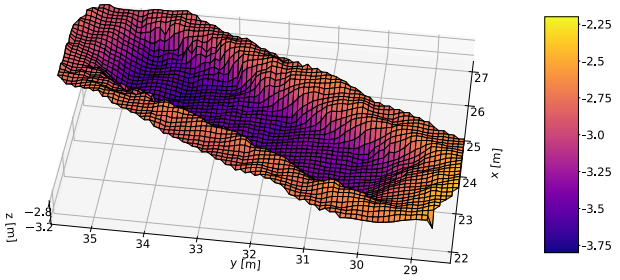


FIGURE 9. LiDAR scan of the completed trench with the autonomous system for an accuracy threshold $\Theta = 0.15$ m.

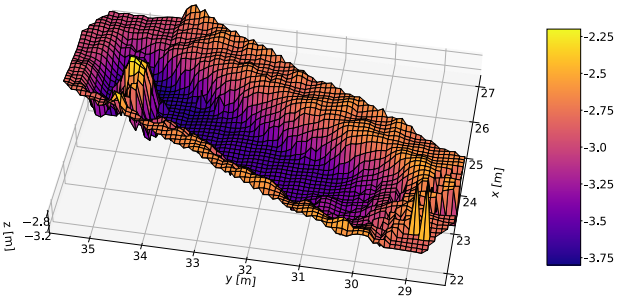


FIGURE 10. LiDAR scan of the completed trench by the human operator. The surveyor was captured by the LiDAR at $y = 35$ m as he stood inside the trench to instruct the machine operator during excavation.

operator had to interrupt excavation for 9% of the total time to wait for the surveyor to monitor the progress. The autonomous system can perform monitoring in negligible 0.5% of the time.

c) Accuracy

Figs. 9 and 10 show the elevation map of the completed trenches by the autonomous system and the human operator, respectively. The excavation accuracy (Fig. 11) is computed

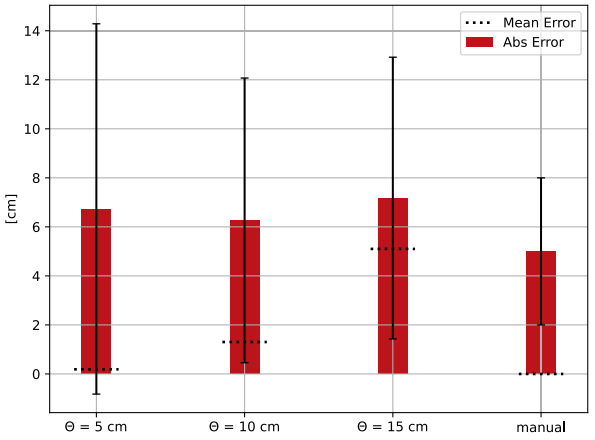


FIGURE 11. Trench excavation accuracy for different accuracy thresholds Θ and manual operation. The error bars show the standard deviation.

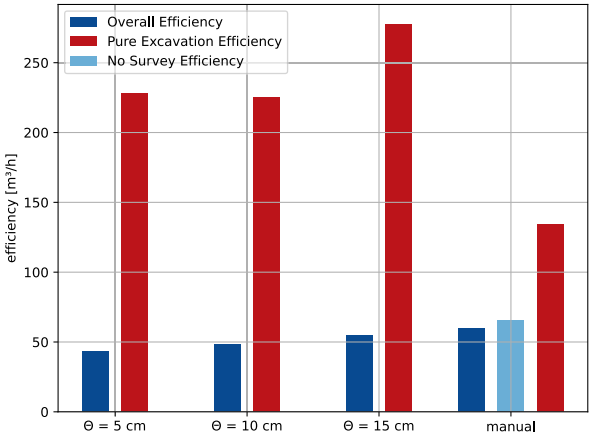


FIGURE 12. Trench excavation efficiency overall and only during excavation for different accuracy thresholds Θ and manual operation.

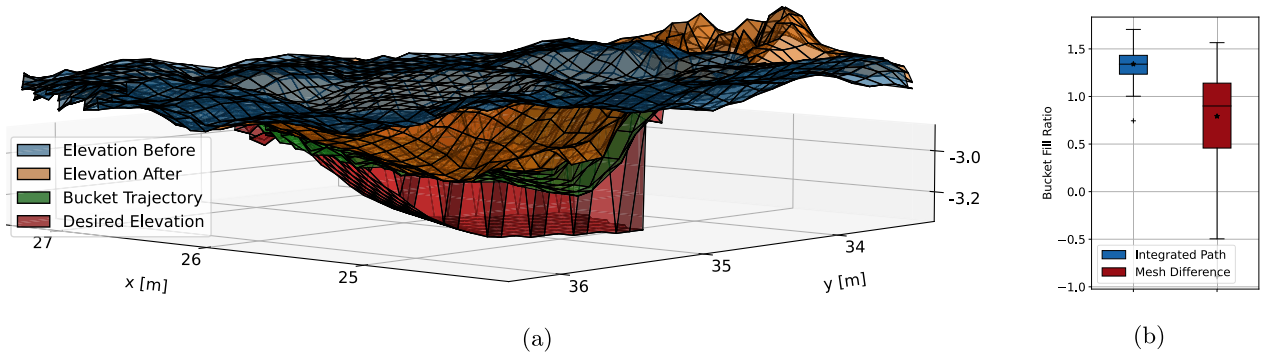


FIGURE 13. Estimated and measured bucket fill ratio during autonomous trench excavation for one scoop. Not all of the moved material ends up in the bucket. Mapping errors can lead to outliers in the measured bucket fill volume. (a) Terrain patch including the excavation trajectory used to evaluate the estimated bucket fill ratio, obtained by integrating the bucket trajectory, and measured, obtained through the difference between the elevation before and after a scoop. (b) Statistics.

by comparing the measured to the desired terrain elevation and averaged over the number of grid map cells. While the average absolute error for the autonomous system is consistent for the three accuracy thresholds and lies in between 6 and 7 cm, the mean error decreases with lower accuracy thresholds. This is expected as the accuracy threshold defines when a grid cell is considered finished by the excavation planner. However, the standard deviation of the accuracy increases with a lower accuracy threshold. With a lower accuracy threshold, more excavation cycles start close to the desired design. This leads to more overshoot because there is less soil that dampens the initially larger bucket velocity, which is caused by imperfect tracking of the joint-velocity references (Fig. 8(a), e.g., scoops 8, 9, and 12).

For comparison, the average absolute error for the manually excavated trench is 5 cm and hence 1–2 cm lower compared to the autonomous system (Fig. 11). Furthermore, the standard deviation is 3 cm, roughly half, which can also be observed visually in Fig. 10, which, compared to the autonomous system, shows a slightly smoother trench bottom.

d) Efficiency

In Fig. 12, we report the efficiency, i.e., excavated soil volume per time, of the entire trenching operation and only during the excavation phase. For manual trenching, we further report the efficiency without surveying to approximate the utilization of a 3-D guidance system. The volume is estimated by comparing the measured to the original terrain elevation (not the integrated bucket path). It can be observed that higher accuracy thresholds lead to higher efficiency, with 54.5 $m^3 \cdot h^{-1}$ for $\Theta = 0.15$. Lower accuracy thresholds lead to more shallow bucket trajectories, close to the desired design, where not all of the moved soil ends up in the bucket.

The overall efficiency of the manual operator is slightly higher: with surveying 59.7 $m^3 \cdot h^{-1}$ and without 65.4 $m^3 \cdot h^{-1}$. However, the efficiency during the excavation phase is almost half compared to the autonomous system. Since the maximum depth is known to the autonomous system, the bucket can be filled much quicker. In contrast, the human operator moves the bucket slower to avoid overshooting the depth constraint. Also, the motions of the bucket in the air for moving to the PoA and the dumping location have a lot of room for improvement, which will substantially increase the efficiency of the autonomous system.

e) Bucket Fill Ratio

The excavation controller needs an online estimation of the current bucket fill state. Because the map update is slower than the control rate, the bucket fill state is obtained by integrating the bucket trajectory through the soil, assuming that all the soil ends up in the bucket. In simulation, this is done similarly due to the lack of a more accurate soil simulation. However, parts of the soil might be pushed away and not actually end up in the bucket. This can be measured by comparing the elevation map patch containing the excavation trajectory before and after a scoop [Fig. 13(a)]. Averaged over all the scoops of the three autonomously excavated trenches, the bucket path integration leads to 55% higher estimate of the bucket fill ratio than the difference of measured elevation [Fig. 13(b)]. Mapping errors caused, for example, by spilling soil, lead to outliers in the measurement of the bucket fill ratio.

Due to the surveyor walking around the bucket during manual excavation, the elevation maps captured during excavation are corrupted such that we cannot report reliable numbers for manual operation (Fig. 10).

IV. CONCLUSION AND DISCUSSION

In this study, we present an RL-based, soil- and constraint-adaptive, perceptive bucket-filling controller for an autonomous excavation system. The controller is trained entirely in simulation and subsequently deployed on a full-sized machine with standard hydraulics. Through extensive randomization of soil parameters in simulation, the controller learns to operate in a wide range of different soils. The agent does not observe soil parameters explicitly but can implicitly estimate them from measurements of onboard sensors and

adapt the excavation path online. We show that, in particular, pressure measurements are essential to learn an efficient soil-adaptive behavior. Thanks to the computationally lightweight simulation setup, training only lasts around 3.5 h on a single-GPU workstation.

Experiments in different soil types show that the controller achieves the desired *penetrate and rotate* and *penetrate and drag* behaviors in soft and hard soil, respectively. Interaction forces with the soil, which are measured through joint torques, can be accurately simulated with the presented soil model, even for the most extreme case: scraping on a concrete floor. The controller also takes into account the current terrain elevation, a maximum-depth constraint, and the desired pullup distance, which is essential for practical excavation use cases. This is demonstrated by integrating the bucket-filling controller into a full-fledged excavation planning system for excavating a trench.

The overall efficiency for excavating a trench is 54 $m^3 \cdot h^{-1}$, which is only slightly below 59 $m^3 \cdot h^{-1}$ achieved manually by a professional machine operator and a surveyor executing the same task. However, the manual operator achieves a smoother bottom of the trench with 5-cm average absolute error with and std of 3 cm, whereas the autonomous system reaches 7 cm on average with an std of 6 cm. Terenzi and Hutter [66] report an efficiency of 42.7 $m^3 \cdot h^{-1}$ with the same machine with an average accuracy of 7 cm, for the excavation of a square pit. These numbers should, however, be compared carefully because of the different excavation geometry.

The autonomous system is slightly lagging behind the professional operator in overall efficiency. It should, however, be noted that the focus of this work is laid on improving the excavation phase and not the motions in the air for moving to the PoA and for dumping. When analyzing the cycle times, it can be observed that the excavation phase of the autonomous system is much shorter compared to manual operation (6.8/14.7 s). Since the maximum depth is known to the system, it can fill the bucket more aggressively. While the time spent in approaching the PoA is similar, dumping provides a major point for improving the efficiency of the autonomous system. Compared to the human, the average dumping time is three times larger (16.0/5.6 s). The human operator combines cabin turning and dumping and operates the machine much closer to the speed limits, whereas the autonomous system operates sequentially: move to the side, open the bucket, and wait for the soil to fall out. To improve the accuracy of the autonomous system, a *refinement state* can be introduced. Thereby, the last few centimeters are excavated with a grading controller optimized for accurate tracking, such as Egli and Hutter [15], where average errors of 2 cm were demonstrated.

In the future, the excavation controller should be further improved. A current limitation of the soil model is that it cannot simulate inhomogeneous grounds that can lead to vibrations such as in Fig. 4(i). For larger boulders up to large, buried, immovable rocks, the policy might not generalize out of the box, and the behavior of the policy should be further investigated. Also, the soil model does not simulate how the terrain changes when the bucket interacts with it. In particular, pushing the soil away rather than scooping it is not simulated and leads to an overestimation of bucket filling. Especially in hard soil, where the ground has to be loosened up first, and material is pushed in front of the bucket, an appropriate bucket-closing strategy is crucial to maximize the amount of scooped soil. Augmenting the simulation with rocks and a more realistic soil behavior can lead to more efficient and general policies. Another point for improvement lies in augmenting the excavator model in simulation to capture real-world effects such as imperfect joint-velocity tracking or sensor noise more realistically. Currently, inaccurate depth tracking or finishing within the pullup zone is caused by the sim-to-real gap. In particular, the overshooting at the beginning of a scoop can be avoided by incorporating tracking delays in the simulation, resulting in higher accuracy and better constraint satisfaction.

REFERENCES

[1] K. Althoefer, C. P. Tan, Y. H. Zweiri, and L. D. Seneviratne, "Hybrid soil parameter measurement and estimation scheme for excavation automation," *IEEE Trans. Instrum. Meas.*, vol. 58, no. 10, pp. 3633–3641, Oct. 2009.

[2] J. Andersson, K. Bodin, D. Lindmark, M. Servin, and E. Wallin, "Reinforcement learning control of a forestry crane manipulator," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Jun. 2021, pp. 2121–2126.

[3] Associated Builders Contractors (ABC). (2023). *Construction Workforce Shortage Tops Half a Million in 2023*. Accessed: Jun. 20, 2023. [Online]. Available: https://www.abc.org/News-Media/News-Releases/entryid/19777/construction-workforce-shortage-tops-half-a-million-in-2023-says-abc

[4] A. Babu and F. Kirchner, "Terrain adaption controller for a walking excavator robot using deep reinforcement learning," in *Proc. 20th Int. Conf. Adv. Robot. (ICAR)*, Dec. 2021, pp. 64–70.

[5] S. Backman, D. Lindmark, K. Bodin, M. Servin, J. Mork, and H. Löfgren, "Continuous control of an underground loader using deep reinforcement learning," *Machines*, vol. 9, no. 10, p. 216, 2021.

[6] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proc. Int. Conf. Mach. Learn. (ICML)*, vol. 2, 2009, pp. 1–8.

[7] N. Bennett, A. Walawalkar, M. Heck, and C. Schindler, "Integration of digging forces in a multi-body-system model of an excavator," *Proc. Inst. Mech. Eng. K, J. Multi-Body Dyn.*, vol. 230, no. 2, pp. 159–177, Jun. 2016.

[8] B. Park, "Development a virtual reality excavator simulator: A math model excavator digging a calculation methodology," Ph.D. thesis, Virginia Polytech. Inst. State Univ., Blacksburg, VA, USA, 2002.

[9] D. A. Bradley and D. W. Seward, "The development, control and operation of an autonomous robotic excavator," *J. Intell. Robot. Syst., Theory Appl.*, vol. 21, no. 1, pp. 73–97, 1998.

[10] H. Cannon and S. Singh, "Models for automated earthmoving," in *Experimental Robotics VI*. Springer, 1999, pp. 163–172.

[11] S. Dadhich, F. Sandin, U. Bodin, U. Andersson, and T. Martinsson, "Adaptation of a wheel loader automatic bucket filling neural network using reinforcement learning," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2020, pp. 1–9.

[12] R. Dhakate, C. Brommer, C. Bohm, H. Gietler, S. Weiss, and J. Steinbrener, "Autonomous control of redundant hydraulic manipulator using reinforcement learning with action feedback," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2022, pp. 7036–7043.

[13] M. Dunbabin and P. Corke, "Autonomous excavation using a rope shovel," in *Tracts in Advanced Robotics* (Springer Tracts in Advanced Robotics). 2006, pp. 555–566.

[14] P. Egli, D. Gaschen, S. Kerscher, D. Jud, and M. Hutter, "Soil-adaptive excavation using reinforcement learning," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 9778–9785, Oct. 2022.

[15] P. Egli and M. Hutter, "A general approach for the automation of hydraulic excavator arms using reinforcement learning," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 5679–5686, Apr. 2022.

[16] P. Fankhauser, M. Bloesch, and M. Hutter, "Probabilistic terrain mapping for mobile robots with uncertain localization," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 3019–3026, Oct. 2018.

[17] Fine (2023). *Geotechnical Software GEO5*. Accessed: Jan. 1, 2023. [Online]. Available: https://www.finesoftware.eu/help/geo5/en/earth-pressures-01/

[18] R. D. Grisso and J. V. Perumpral, "Review of models for predicting performance of narrow tillage tool," *Trans. ASAE*, vol. 28, no. 4, pp. 1062–1067, 1985.

[19] T. Groll, S. Hemer, T. Ropertz, and K. Berns, "Autonomous trenching with hierarchically organized primitives," *Autom. Construct.*, vol. 98, pp. 214–224, Feb. 2019.

[20] Q. Guo, Z. Ye, L. Wang, and L. Zhang, "Imitation learning and model integrated excavator trajectory planning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2022, pp. 5737–5743.

[21] D. Höber, A. Taschner, and E. Fimbinger, "Excavation and conveying technologies for space applications," *Berg-Und Huttenmännische Monatshefte*, vol. 166, no. 2, pp. 95–103, Feb. 2021.

[22] D. Hoeller, N. Rudin, D. Sako, and M. Hutter, "ANYmal parkour: Learning agile navigation for quadrupedal robots," *Sci. Robot.*, vol. 9, no. 88, pp. 1–20, Mar. 2024.

[23] J. Huang, D. Kong, G. Gao, X. Cheng, and J. Chen, "Data-driven reinforcement-learning-based automatic bucket-filling for wheel loaders," *Appl. Sci.*, vol. 11, no. 19, p. 9191, Oct. 2021.

[24] J. Huh et al., "Deep learning-based autonomous excavation: A bucket-trajectory planning algorithm," *IEEE Access*, vol. 11, pp. 38047–38060, 2023.

[25] M. Hutter et al., "Towards optimal force distribution for walking excavators," in *Proc. Int. Conf. Adv. Robot. (ICAR)*, Jul. 2015, pp. 295–301.

[26] J. V. Perumpral, R. D. Grisso, and C. S. Desai, "A soil-tool model based on limit equilibrium analysis," *Trans. ASAE*, vol. 26, no. 4, pp. 0991–0995, 1983.

[27] G. Mesri and T. M. Hayat, "The coefficient of Earth pressure at rest," *Can. Geotech. J.*, vol. 30, no. 4, pp. 647–666, Aug. 1993.

[28] S. Jin, Z. Ye, and L. Zhang, "Learning excavation of rigid objects with offline reinforcement learning," 2023, *arXiv:2303.16427*.

[29] K. Johnson, "The elusive dream of fully autonomous construction vehicles," Tech. Rep., 2023.

[30] D. Jud, G. Hottiger, P. Leemann, and M. Hutter, "Planning and control for autonomous excavation," *IEEE Robot. Autom. Lett.*, vol. 2, no. 4, pp. 2151–2158, Oct. 2017.

[31] D. Jud, I. Hurkxkens, C. Girot, and M. Hutter, "Robotic embankment," *Construct. Robot.*, vol. 5, no. 2, pp. 101–113, Jun. 2021.

[32] D. Jud et al., "HEAP—The autonomous walking excavator," *Autom. Construct.*, vol. 129, Sep. 2021, Art. no. 103783.

[33] D. Jud, P. Leemann, S. Kerscher, and M. Hutter, "Autonomous free-form trenching using a walking excavator," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 3208–3215, Oct. 2019.

[34] E. Kaufmann, L. Bauersfeld, A. Loquercio, M. Muller, V. Koltun, and D. Scaramuzza, "Champion-level drone racing using deep reinforcement learning," *Nature*, vol. 620, no. 7976, pp. 982–987, Aug. 2023.

[35] Y. B. Kim, J. Ha, H. Kang, P. Y. Kim, J. Park, and F. C. Park, "Dynamically optimal trajectories for earthmoving excavators," *Autom. Construct.*, vol. 35, pp. 568–578, Nov. 2013.

[36] A. J. Koivo, M. Thoma, E. Kocaoglan, and J. Andrade-Cetto, "Modeling and control of excavator dynamics during digging operation," *J. Aerosp. Eng.*, vol. 9, no. 1, pp. 10–18, Jan. 1996.

[37] A. Koliji. (2023). *Geotech Data*. Accessed: Jan. 1, 2023. [Online]. Available: https://www.geotechdata.info/parameter/

[38] I. Kurinov, G. Orzechowski, P. Hämäläinen, and A. Mikkola, "Automated excavator based on reinforcement learning and multibody system dynamics," *IEEE Access*, vol. 8, pp. 213998–214006, 2020.

[39] D. Lee, I. Jang, J. Byun, H. Seo, and H. J. Kim, "Real-time motion planning of a hydraulic excavator using trajectory optimization and model predictive control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2021, pp. 2135–2142.

[40] Q. Lu and L. Zhang, "Excavation learning for rigid objects in clutter," *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 7373–7380, Oct. 2021.

[41] Q. Lu, Y. Zhu, and L. Zhang, "Excavation reinforcement learning using geometric representation," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 4472–4479, Apr. 2022.

[42] O. Luengo, S. Singh, and H. Cannon, "Modeling and identification of soil-tool interaction in automated excavation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. Innov. Theory, Pract. Appl.*, vol. 3, Jul. 1998, pp. 1900–1906.

[43] G. J. Maeda, I. R. Manchester, and D. C. Rye, "Combined ILC and disturbance observer for the rejection of near-repetitive disturbances, with application to excavation," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 5, pp. 1754–1769, Sep. 2015.

[44] V. Makoviychuk, "Isaac gym: High performance GPU based physics simulation for robot learning," in *Proc. Conf. Neural Inf. Process. Syst. (NeurIPS) Datasets Benchmarks Track (Round 2)*, 2021, pp. 1–22.

[45] K. Matsumoto et al., "Simulation-based reinforcement learning approach towards construction machine automation," in *Proc. Int. Symp. Autom. Robot. Construct. (IAARC)*, Oct. 2020, pp. 457–464.

[46] T. Osa and M. Aizawa, "Deep reinforcement learning with adversarial training for automated excavation using depth images," *IEEE Access*, vol. 10, pp. 4523–4535, 2022.

[47] H.-S. Park, D.-V. Dang, T.-T. Nguyen, and N.-T. Le, "Implementation of a virtual autonomous excavator," *Trans. FAMENA*, vol. 41, no. 3, pp. 65–80, Oct. 2017.

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

[49] Q. Ha, M. Santos, Q. Nguyen, D. Rye, and H. Durrant-Whyte, "Robotic excavation in construction automation," *IEEE Robot. Autom. Mag.*, vol. 9, no. 1, pp. 20–28, Mar. 2002.

[50] A. R. Reece, "The fundamental equation of Earth-moving mechanics," *Proc. Inst. Mech. Eng.*, vol. 179, pp. 16–22, 1964.

[51] N. Reginald, J. Seo, and M. Cha, "Integrative tracking control strategy for robotic excavation," *Int. J. Control, Autom. Syst.*, vol. 19, no. 10, pp. 3435–3450, Oct. 2021.

[52] M. J. Ribeirinho et al., *The Next Normal in Construction*. Chicago, IL, USA: McKinesy & Company, 2020.

[53] N. Rudin, D. Hoeller, P. Reist, and M. Hutter, "Learning to walk in minutes using massively parallel deep reinforcement learning," in *Proc. Conf. Robot Learn. (CoRL)*, 2022, pp. 91–100.

[54] P. Samtani, F. Leiva, and J. Ruiz-del-Solar, "Learning to break rocks with deep reinforcement learning," *IEEE Robot. Autom. Lett.*, vol. 8, no. 2, pp. 1077–1084, Feb. 2023.

[55] R. J. Sandzimier and H. H. Asada, "A data-driven approach to prediction and optimal bucket-filling control for autonomous excavators," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 2682–2689, Apr. 2020.

[56] C. Schenck, J. Tompson, S. Levine, and D. Fox, "Learning robotic manipulation of granular media," in *Proc. Conf. Robot Learn.*, 2017, pp. 239–248.

[57] D. Schmidt, M. Proetzsch, and K. Berns, "Simulation and control of an autonomous bucket excavator for landscaping tasks," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 5108–5113.

[58] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," 2015, *arXiv:1506.02438*.

[59] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*.

[60] H. Shao, H. Yamamoto, Y. Sakaida, T. Yamaguchi, Y. Yanagisawa, and A. Nozue, "Automatic excavation planning of hydraulic excavator," in *Proc. Int. Conf. Intell. Robot. Appl. (ICIRA)*, vol. 5315, 2008, pp. 1201–1211.

[61] B. Son, C. Kim, C. Kim, and D. Lee, "Expert-emulating excavation trajectory planning for autonomous robotic industrial excavator," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 2656–2662.

[62] F. E. Sotiropoulos and H. H. Asada, "A model-free extremum-seeking approach to autonomous excavator control based on output power maximization," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1005–1012, Apr. 2019.

[63] 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. Autom. Lett.*, vol. 7, no. 1, pp. 151–158, Jan. 2022.

[64] A. Stentz, J. Bares, S. Singh, and P. Rowe, ''Robotic excavator for autonomous truck loading,'' *Auto. Robots*, vol. 7, no. 2, pp. 175–186, 1999.
[65] H. Tahara, H. Sasaki, H. Oh, B. Michael, and T. Matsubara, ''Disturbance-injected robust imitation learning with task achievement,'' in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2022, pp. 2466–2472.
[66] L. Terenzi and M. Hutter, ''Towards autonomous excavation planning,'' 2023, *arXiv:2308.11478*.
[67] J. Thangavelautham, K. Law, T. Fu, N. A. E. Samid, A. D. S. Smith, and G. M. T. D'Eleuterio, ''Autonomous multirobot excavation for lunar applications,'' *Robotica*, vol. 35, no. 12, pp. 2330–2362, Dec. 2017.
[68] U.S. Bureau of Labor Statistics. (2021). *Number and Rate of Fatal Work Injuries, by Private Industry Sector*. Accessed: Jun. 20, 2023. [Online]. Available: https://www.bls.gov/charts/census-of-fatal-occupational-injuries/number-and-rate-of-fatal-work-injuries-by-industry.htm
[69] V. Wiberg et al., ''Sim-to-real transfer of active suspension control using deep reinforcement learning,'' 2023, *arXiv:2306.11171*.
[70] X. Shi, P. J. A. Lever, and F.-Y. Wang, ''Experimental robotic excavation with fuzzy logic and neural networks,'' in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 1, Aug. 1996, pp. 957–962.
[71] Y. Yang, P. Long, X. Song, J. Pan, and L. Zhang, ''Optimization-based framework for excavation trajectory generation,'' *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 1479–1486, Apr. 2021.
[72] Y. Yang, J. Pan, P. Long, X. Song, and L. Zhang, ''Time variable minimum torque trajectory optimization for autonomous excavator,'' 2020, *arXiv:2006.00811*.
[73] Z. Yao, S. Zhao, X. Tan, W. Wei, and Y. Wang, ''Real-time task-oriented continuous digging trajectory planning for excavator arms,'' *Autom. Construct.*, vol. 152, Aug. 2023, Art. no. 104916.
[74] H. Yoshida, T. Yoshimoto, D. Umino, and N. Mori, ''Practical full automation of excavation and loading for hydraulic excavators in indoor environments,'' in *Proc. IEEE 17th Int. Conf. Autom. Sci. Eng. (CASE)*, Aug. 2021, pp. 2153–2160.
[75] T. Yoshida, T. Koizumi, N. Tsujiuchi, Z. Jiang, and Y. Nakamoto, ''Digging trajectory optimization by soil models and dynamics models of excavator,'' *SAE Int. J. Commercial Vehicles*, vol. 6, no. 2, pp. 429–440, Sep. 2013.
[76] H. S. Yu and G. T. Houlsby, ''Finite cavity expansion in dilatant soils: Loading analysis,'' *Géotechnique*, vol. 41, no. 2, pp. 173–183, Jun. 1991.
[77] Y. Zhang, Z. Sun, Q. Sun, Y. Wang, X. Li, and J. Yang, ''Time-jerk optimal trajectory planning of hydraulic robotic excavator,'' *Adv. Mech. Eng.*, vol. 13, no. 7, Jul. 2021, Art. no. 168781402110346.
[78] J. Zhao, Y. Hu, C. Liu, M. Tian, and X. Xia, ''Spline-based optimal trajectory generation for autonomous excavator,'' *Machines*, vol. 10, no. 7, p. 538, Jul. 2022.
[79] J. Zhao and L. Zhang, ''TaskNet: A neural task planner for autonomous excavator,'' in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 2220–2226.
[80] Y. Zhao, J. Wang, Y. Zhang, and C. Luo, ''A novel method of soil parameter identification and force prediction for automatic excavation,'' *IEEE Access*, vol. 8, pp. 11197–11207, 2020.
[81] Y. Zhu, L. Wang, and L. Zhang, ''Excavation of fragmented rocks with multi-modal model-based reinforcement learning,'' in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2022, pp. 6523–6530.
[82] Z. Zou, J. Chen, and X. Pang, ''Task space-based dynamic trajectory planning for digging process of a hydraulic excavator with the integration of soil–bucket interaction,'' *Proc. Inst. Mech. Eng. K, J. Multi-Body Dyn.*, vol. 233, no. 3, pp. 598–616, Sep. 2019.