Soil-Adaptive Excavation Using Reinforcement Learning

Journal Article

Author(s):
Egli, Pascal; Gaschen, Dominique; Kerscher, Simon; Jud, Dominic; Hutter, Marco

Publication date:
2022-10

Permanent link:
https://doi.org/10.3929/ethz-b-000557541

Originally published in:
IEEE Robotics and Automation Letters 7(4), https://doi.org/10.1109/LRA.2022.3189834

Soil-Adaptive Excavation Using Reinforcement Learning

Pascal Egli, Dominique Gaschen, Simon Kerscher, Dominic Jud, Marco Hutter

***Abstract*—In this letter, we present an excavation controller for a full-sized hydraulic excavator that can adapt online to different soil characteristics. Soil properties are hard to predict and can vary even within one scoop, which requires a controller that can adapt online to the encountered soil conditions. The objective is to fill the bucket with excavation material while respecting machine limitations to prevent stalling or lifting of the machine. To this end, we train a control policy in simulation using Reinforcement Learning (RL). The soil interactions are modeled based on the Fundamental Equation of Earth-Moving (FEE) with heavily randomized soil parameters to expose the agent to a wide range of different conditions. The agent learns to output joint velocity commands, which can be directly applied to the standard proportional valves of the real machine. We test the controller on a 12-ton excavator in different types of soils. The experiments demonstrate that the controller can adapt online to changing conditions without the explicit knowledge of the soil parameters, solely from proprioceptive observations, which are easily measurable.**

***Index Terms*—Autonomous Excavation, Reinforcement Learning, Sim-to-Real**

I. INTRODUCTION

HYDRAULIC excavators are omnipresent in various areas of application such as construction sites, forest businesses, or mines due to their great versatility. The automation of such machines has a huge potential to improve efficiency, productivity, and safety [1].

In this work, we focus on one of the most fundamental tasks for a hydraulic excavator, which is the excavation of soil. This is a particularly challenging problem since it includes the control of the highly nonlinear machine dynamics and its interaction with soil, which can have very different properties even within one digging cycle. A prerequisite for efficiently excavating is minimizing the number of time-consuming loading cycles. Thus, the amount of excavated soil per scoop needs to be maximized. Since an excavator has limited capabilities in terms of forces that it can apply to the soil, the optimal digging path heavily depends on the properties of the ground. In hard soil, for example, the bucket can only penetrate to a small depth without stalling or lifting up the machine, resulting in a shallow but long digging trajectory, i.e., merely scraping on the ground, also referred to as *penetrate and drag* [2]. Contrary, in very soft soil, the most efficient digging path is deep and short because the excavator can penetrate much deeper without reaching its force limits, i.e., *penetrate and scoop* [2]. Soil characteristics are difficult to measure or predict, especially because they can vary a lot on a site or even within the same scoop (see Fig. 1). Therefore, a controller is required that can adapt online to the encountered soil properties, similar to how human operators proceed, without explicitly knowing soil parameters.

Fig. 1. Excavating soil with abruptly changing properties during the same scoop with a 12-ton hydraulic excavator. A block of granite is buried in the first part of the scoop to emulate extremely hard soil.

A. Related Work

Various researchers have addressed the problem of automating excavation over the past decades. The proposed solutions are manifold, covering trajectory optimization, model-free approaches, learning from demonstration, and RL-based methods. However, current solutions still have deficiencies that make them fall short compared to the efficiency of human operators. In particular, there is a lack of methods that can adapt online to varying soil conditions while using the machine's full capabilities and that do not rely on expensive hardware modifications.

Trajectory optimization is a popular approach to finding the optimal digging path. Purely kinematic solutions were proposed that optimize for time efficiency, bucket filling, and other kinematic considerations [3], [4]. While the computation is quick, the lack of taking into account machine dynamics and external forces resulting from the interaction with the ground makes these approaches incapable of digging in different soils without stalling or being inefficient. Lee et al. [5] combined kinematic path optimization with dynamics-aware Model Predictive Control (MPC) to track the kinematic

Manuscript received: February, 24, 2022; Revised April, 17, 2022; Accepted June, 19, 2022.

This paper was recommended for publication by Editor Youngjin Choi upon evaluation of the Associate Editor and Reviewers' comments.

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

Authors are with the Robotic Systems Lab, ETH Zurich, Switzerland. Contact: pasegli@ethz.ch

Digital Object Identifier (DOI): see top of this page.

trajectory. Paired with a disturbance observer that estimates external forces, the MPC could track the trajectory accurately in simulation. The joint torques, however, were far below the physical limitations of the machine during digging, such that the executed trajectory only minimally deviated from the plan, which indicates inefficient digging [6], [7]. To account for different soil properties, dynamics-based trajectory optimization approaches that incorporate soil dynamics were proposed. However, these methods are computationally expensive and cannot be executed in real-time, i.e., the computation time lies in the order of minutes to compute one trajectory [8]. Also, soil parameters need to be known for the optimization. They can be determined by analyzing the site geologically [9] or by fitting the soil parameters of the model to measurements collected during manual operation of the machine [10]. This makes these approaches incapable of adapting to unexpected soil properties online.

Another stream of research divides the digging trajectory into a sequence of four segments based how human operators perform excavation work: sloped penetration, horizontal dragging, scooping, and vertical lifting. If soil conditions on the test site are homogeneous, the segments of the parameters can be manually calibrated [11]–[13]. However, in most applications, soil conditions vary. Therefore, Sotiropoulos et al. [6] proposed a controller that maximizes the power transmitted to the soil during the dragging phase by controlling the boom joint. This resulted in adaptive digging in different grounds, required, however, still manually designing the remaining excavation phases and was not tested on a hydraulic machine with low control bandwidth and deadzones. Maeda et al. [14] designed an iterative learning controller with a disturbance observer to follow a human-inspired digging trajectory. This approach assumes that disturbances are similar (near-repetitive) between excavation cycles, which is generally not the case. To handle the event of stalling if the required forces to track the trajectory segments exceed the capabilities of the machine or an obstacle is encountered, rule-based methods have been proposed [15], [16]. If stalling is detected, the digging path is modified with hand-crafted corrections, which need to be specifically designed for the particular platform and use case. Instead of following a kinematic trajectory, Jud et al. [17], [18] defined a sequence of end effector force references for each digging phase. This resulted in soil-adaptive digging while respecting machine limits. However, it required retrofitting expensive high-performance servo valves to track the desired force references.

Other researchers have proposed behavior-based approaches. Depending on the state of the excavator, an appropriate action is selected from a library of predefined motion primitives. Tested in simulation [19] and also on a real machine [2], [7], [20], [21], behavior-based methods could successfully dig. However, it requires a substantial engineering effort, specific to a particular machine, to create the behavior database. Bradley et al. [2], for example, designed 80 different rules for autonomous excavation.

Also, learning-based approaches have been proposed. Son et al. [22] used learning from demonstration to find the parameters for dynamic motion primitives to imitate the trajectories of human operators, which were then modulated online to avoid excessive forces. The modulation, however, only affected the excavation depth. The endpoint of the trajectory remained fixed. Especially in hard soil, this leads to inefficient digging because the bucket is not filled up.

Based on visual representations of the excavation scene, convolutional neural networks and RL policies have been trained to find optimal trajectories to excavate rigid objects [23], [24] or to manipulate granular material accurately [25]. These approaches, however, are purely geometric. Hence, they cannot adapt to different types of soils if force limits are exceeded. Park et al. [26] used an echo-state network, pretrained with a conventional PD controller and then updated online during digging operation to track a desired trajectory. The adaptation, however, takes multiple dig cycles and assumes that the soil conditions remain similar. More recently, the feasibility has been investigated to use RL for training excavation policies in full-fledged physics simulators, which combine meshes and particles to simulate realistic soil properties [27]–[29]. While such simulators can model soil interactions accurately, they are computationally expensive. Therefore, to keep training times reasonable, the investigated scenarios were simplified with fixed soil parameters such that the agent does not learn to adapt to different soil properties. Also, the deployment on real excavators has not yet been demonstrated.

B. Contribution

The contribution of this work is an excavation controller that can adapt online to varying soil conditions, also if they change within the same scoop. We leverage RL and an analytical soil model to train a control policy in simulation that can be transferred to the real machine. Experiments on a full-sized hydraulic excavator demonstrate that the controller consistently achieves bucket filling without stalling or lifting the machine in different types of soil, while still applying large forces to the ground. If bucket filling is not possible due to initializing the bucket too close to the machine or extremely hard soil, the controller prioritizes avoiding self-collisions over bucket filling. Our approach does not require soil parameters to be known explicitly. They are inferred through proprioceptive measurements from hydraulic pressure and kinematic sensors, which are readily available on modern excavators. The controller actuates standard proportional valves such that an expensive modification of the hydraulics is not required.

II. Method

We train an RL excavation control policy in simulation that we can deploy on the real machine. Therefore, we use the classical RL setup and model the problem as a discrete-time Markov Decision Process. The agent, in our case the control policy, interacts with an environment that consists of the excavator in a simulator and a soil model. The state of the environment at time step t is represented by $s_t \in \mathcal{S}$. At every time step, the agent observes $o_t \in \mathcal{O} \subseteq \mathcal{S}$, takes action $a_t \in \mathcal{A}$ and receives a scalar reward $r_t(s_t, a_t, s_{t+1}) \in \mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$. The agent acts according to a stochastic policy

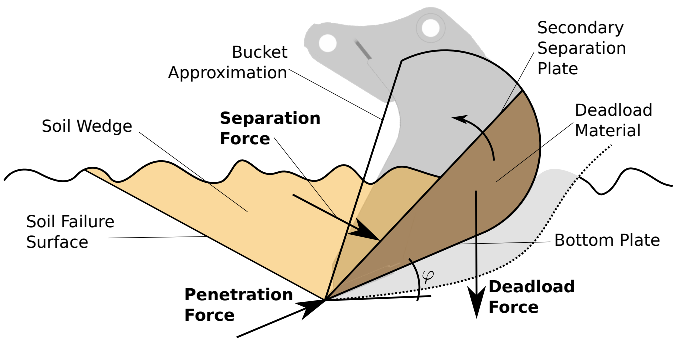


Fig. 2. Forces from the analytical soil model acting on the bucket. The angle between the bottom plate and its velocity is denoted with φ.

$\pi(a_t|o_t)$. Its objective is to learn a policy that maximizes the infinite-horizon reward $\mathbb{E}\big[\sum_{t=0}^{\infty}\gamma^t r_t\big]$ by interacting with the environment. The discount factor $\gamma \in (0,1)$ trades off between current and future rewards.

A. Simulation

1) Excavator: The rigid-body dynamics of the excavator are simulated with RaiSim [30], a fast physics engine developed for robotics and RL. The excavator's dynamic model is exported from CAD, which is only accurate to some extent, as many parts such as hoses or hydraulic oil are not included in the model.

As explained later in Section II-C, the agent outputs joint velocity commands for the arm joints. Therefore, we implement an explicit PID controller for each joint that outputs joint torques, given a velocity reference. The PID parameters and update rate are tuned to achieve practically perfect velocity tracking. Forces acting on the bucket are compensated by applying the corresponding torques at the joints, which can be computed using the translational Jacobians. Torque and velocity limits are obtained from the real machine and assumed to be constant over the joint range. The outputs of the PID controllers are clipped accordingly. This leads to the behavior that the joint motion is stopped if the torque required to overcome resistance, e.g., from the soil, exceeds the joint limit. Additionally, a particularity of hydraulic actuators is that they can absorb much more force in the direction opposite to the command. We implement this by changing the torque limits depending on the desired joint velocity. The remaining joints, in particular the legs and the cabin turn, are not actuated and kept in a fixed configuration. The machine is simulated with a floating base. Its wheels are supported by solid ground with a friction coefficient of 0.8.

2) Soil: Training the agent to excavate different soils requires a soil model. Existing soil models can be broadly categorized as particle-based or analytical. While particle-based models are relatively accurate and simulate many physical effects, they are computationally expensive, which makes them hitherto unsuited for training complex policies with RL [27]–[29]. For this reason, we use a computationally lightweight analytical soil model, which expresses the main excavation forces in terms of geotechnical parameters. We heavily randomize the soil parameters to simulate a wide variety of different soils and thus compensate for the reduced accuracy of the model.

TABLE I
SOIL PARAMETERS AND RANDOMIZATION RANGES.

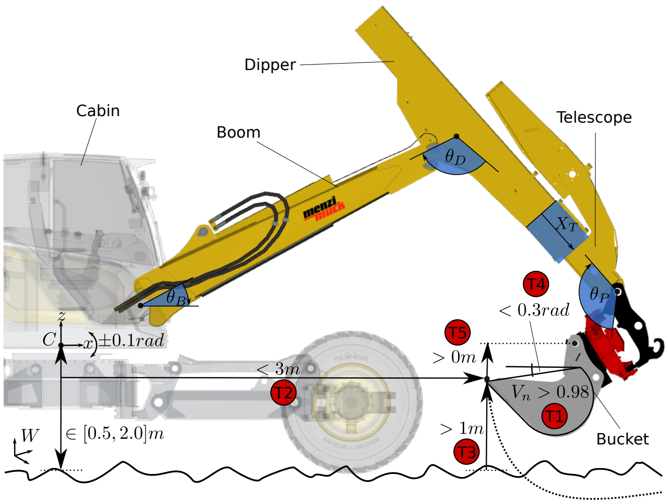
Parameter	**Unit**	**Min**	**Max**
Cohesion (c) [32]	kPa	0	105
Adhesion (c_a) [33]	kPa	0	c
Soil internal friction angle (Φ) [32]	rad	0.3	0.8
Unit weight (γ) [32]	$\mathrm{kN/m^3}$	17	22
Soil-bucket friction angle (δ) [33]	rad	0.2	0.4
Cavity pressure factor (p_t) [34]	-	0	300

We base our implementation on the thesis of Park [31], whose model comprises two mechanisms: separation and penetration.

Separation: Separation describes the process of breaking and displacing soil. Its computation is based on the assumption of a flat blade moving horizontally through the soil and was first introduced by Reece [35] in 2 dimensions as the Fundamental Equation of Earth-Moving (FEE) and later extended to 3 dimensions, which is more appropriate for excavation [36]. The FEE computes the forces required to break the soil along a failure surface, determined based on the bucket geometry, and soil parameters (see Fig. 2). We assume a simplified bucket geometry, consisting of a triangle and a semicircle, and compute the separation forces for the secondary separation plate that builds up as the bucket is filled with soil. Thereby, we assume that the excavated material is evenly distributed in the bucket and not compacted. The excavated soil volume is obtained by integrating the bucket path through the soil. In addition to Park's model, we add the gravitational force of the accumulated material, which acts on its centroid. Even though Park's model accounts for the slope of the soil, we assume in this work horizontal ground for simplicity.

Penetration: Penetration refers to cutting the soil with the edge or teeth of the bucket. It consists of frictional and adhesive forces acting on the bottom plate and the bucket edge. For the bottom plate, the friction is proportional to the passive earth pressure, as given by Bennett et al. [37]. We use the model for the coefficient of the lateral earth pressure proposed by Jaky [38]. The resistive forces acting on the edge can be computed similarly, except that the pressure is higher because the soil is actively deformed. As proposed by Park, this can be modeled using the cavity expansion theory. However, the exact pressure can only be determined in an iterative fashion. Therefore, we simplify the model by making use of the fact that the cavity pressure does not increase indefinitely as the cavity expands but reaches a limit pressure [34]. We then assume that the limit pressure is directly reached and introduce a factor that defines how much larger the cavity pressure is compared to the passive earth pressure (p_t). This factor is an additional soil parameter that has been determined through field tests [34].

Instead of defining discrete digging modes like Park, where either separation, penetration, or both are active, we follow a similar approach as Bennett et al. [37] and superimpose both mechanisms. Additionally, we scale the resistance from the bucket edge with the cosine of the angle φ between the bottom plate and the bucket motion (see Fig. 2), i.e., if the bucket moves parallel to the bottom plate, the factor is 1, if it moves



Termination		Definition
T1	Full enough	Filling ratio $V_{n,t} \in [0,1] > k_j 0.68 + 0.3$
T2	Too close	Edge distance to base < 3.0 m
T3	Edge high enough	Edge height above soil $> k_j 0.7 + 0.3$ m
T4	Curled enough	Angle to horizon < 0.3 rad
T5	Origin high enough	Origin height $>$ Edge height
Terminate if		(T1 $\vee$ T2) $\wedge$ T3 $\wedge$ T4 $\wedge$ T5
Curriculum factor $\in [0,1]$		$k_j = 1 - \exp(-0.01j)$

Fig. 3. Desired terminal state conditions (T1-T5). The policy actuates the four main arm joints θ_B, θ_D, X_T, θ_P. The cabin height remains fixed, while the cabin pitch and soil height are randomized during training. The wheels are supported by patches of solid ground. The cabin frame is denoted with C and the world frame with W.

perpendicular, the factor is 0. Spillage of soil is simulated by setting the bucket fill volume to 0 if the bottom plate of the bucket points downwards while not being inside the soil. The soil parameters' randomization ranges are obtained from existing databases (see Table I). Computing the forces from the analytical soil model is roughly 40 times faster than the rigid body dynamics of the excavator.

B. Learning Objective

The agent's objective is to start digging at the location of initialization and filling the bucket with soil as quickly as possible while respecting certain constraints. This is achieved by formulating appropriate rewards and termination conditions. The agent receives a reward once per time step, whereas if a termination condition is reached, the training episode is terminated and reset, and the agent receives once a terminal reward.

1) Positive Termination: To encourage the agent to fill the bucket as quickly as possible, we define a terminal state (see Fig. 3) where we award the agent with +10 if the bucket is full enough (T1) or +5 if it is too close to the machine (T2). Additionally, the bucket has to be high enough above the soil (T3), curled sufficiently to avoid spillage (T4) and the bucket origin has to be higher than the bucket edge to prevent over curling (T5). To facilitate learning, we make it at the beginning of the training easier for the agent to achieve the desired final state by introducing a curriculum (k_j) [39], which is changed based on the RL update count (j).

TABLE II
REWARD TERMS FOR POLICY TRAINING.

Reward		Definition	$\neq 0$ If C5 $\wedge$
R1	Move down	$-0.1 v^W_{t,z}$	$V_{n,t} < 5\%$
R2	Filling	$V_{n,t} - V_{n,t-1}$	$\neg$C2
R3	Move up	$0.1 v^W_{t,z}$	(C1 $\vee$ C2) $\wedge$ $\neg$C3
R4	Curl	$0.05 \omega^W_{t,y}$	(C1 $\vee$ C2) $\wedge$ $\neg$C4
R5	Smooth action	$-0.005 \|a_t - a_{t-1}\|_1$	Always
C1	Full eough	Filling ratio $V_{n,t} \in [0,1] > k_j 0.6 + 0.3$	
C2	Too close	Edge distance to base < 3.0 m	
C3	Edge high enough	Edge height above soil > 1.0 m	
C4	Curled enough	Angle to horizon < 0.3 rad	
C5	Slow enough	Edge vel $< 0.4\ \mathrm{m\,s^{-1}}$	

2) Negative Termination: To discourage the agent from reaching certain states, we define a set of conditions for which we terminate the episode and give a reward of -1.

Bucket Velocity: Without further optimization, we set the maximum bucket velocity to $0.5\ \mathrm{m\,s^{-1}}$, which corresponds to a reasonable digging speed.

Bucket Motion and Orientation: To avoid pushing the soil with the lower side of the bottom plate, the angle between the bucket motion and the bottom plate (φ) has to be positive (see Fig. 2).

Excavator Base Motion: The episode is terminated if any of the components of the linear velocity of the excavator's base exceed $0.1\ \mathrm{m\,s^{-1}}$ to prevent stalling by pulling the machine towards the bucket or lifting it up.

Empty Bucket Above Soil: To restrict the search space of the agent and focus on exploration close or inside the soil, the episode is terminated if the bucket is empty and more than 0.5 m above the ground.

Self-Collisions: If the arm collides with any part of the excavator, the episode is terminated. The legs are spread such that self-collisions with the legs cannot occur.

Too Deep or Flat: The secondary separation plate must exit the soil before reaching the bucket's back. If this is not enforced, the bucket can penetrate very deep or dig at a very flat angle (see Fig. 2).

3) Reward: The total reward is given by the sum of the terms R1-R5. Their definitions and conditions for which they are non-zero are listed in Table II. To encourage the agent to start filling the bucket, it receives a reward proportional to the linear velocity of the bucket edge towards the soil, i.e., moving down ($v^W_{t,z} < 0$) if the bucket is filled less than 5 % (R1). Bucket filling is motivated by a reward proportional to the normalized amount of added soil volume if the bucket is not too close to the base (R2, C2). The incentive to reach the terminal desired state is then provided by giving a reward for moving the bucket upwards ($v^W_{t,z} > 0$) if it is not already high enough (R3, C3), and for curling ($\omega^W_{t,y}$), if not already curled enough (R4, C4), once the bucket is full enough (C1) or too close to the base (C2). To incentivize the agent to maintain a margin to the maximal bucket velocity defined by the negative terminal conditions, we set all the reward terms to 0 if the bucket velocity exceeds $0.4\ \mathrm{m\,s^{-1}}$ (C5). To ensure smooth commands, the L_1-norm between two consecutive actions (a_t, a_{t-1}) is penalized (R5).

C. Observations and Actions

We chose the agent's actions to be joint velocities instead of joint torques. Since the dynamic model of the excavator is known to be inaccurate, a torque-based policy would not be easily transferable to the real machine. Additionally, to control forces, valves operate around zero oil flow. Standard valves have a large deadband which leads to a bad force tracking performance such that a more advanced hydraulic setup would be required [17], [40]. Also, human operators control the oil flow to the cylinders through the joysticks, which comes close to controlling the cylinder velocities. We actuate the four main arm joints (see Fig. 3), hence restricting the digging motion to 2 dimensions as human operators mainly do it. As observations, we use quantities that are easily measurable also on the actual machine. Most importantly, the agent does not receive any information about the soil properties but infers the characteristics from proprioceptive measurements. In particular, joint torque observations enable the agent to sense the properties of the soil and adapt its behavior accordingly. We also trained a policy without providing torque observations, resulting in a very conservative and inefficient controller operating far below the torque limits. The reason for this is that the agent avoids deep penetration because it might stall depending on the soil properties and lacks information to infer the cause of stalling, i.e., reaching torque limits, which cannot be deduced from kinematic observations. The position and velocity of the bucket can be inferred through the joint states, are, however, provided to accelerate learning and represented in the cabin frame (see Fig. 3) to be independent of the cabin turn position. Since we assume flat ground, we can estimate the height of the soil relative to the cabin by averaging the wheel heights, which are known from leg kinematics. The bucket fill volume is then approximated by integrating the bucket path through the soil. The observations are noise-free, and actions and observations are normalized with constant empirical means and standard deviations to accelerate training. The actions and observations are summarized in Table III.

D. Episode Initialization

At the beginning of every training episode, we initialize the environment in a randomized state to expose the agent to different conditions. Specifically, we sample uniform random soil parameters within the ranges listed in Table I, including the soil height in the range of $[-2.0, -0.5]$ m relative to the cabin (see Fig. 3). The arm is initialized in a random state within the joint limits and less than 0.4 m above the ground. In 25 % of the cases, the bucket is initialized inside the soil and we sample a random bucket fill ratio. This helps the agent explore the state space and accelerates the convergence of the policy. The excavator itself is initialized with a pitch angle in the range of ± 0.1 rad.

E. Training

We train the agent using a custom implementation of Proximal Policy Optimization (PPO) [41] with General Advantage Estimation (GAE) [42], which allows for parallelized rollout collection[1]. The policy and value functions are approximated using two separate neural networks which receive the same observations and have linear output layers. The training converges after ~3K updates (~200M samples) and takes around 7 h[2] (see Fig. 4).

TABLE III
POLICY OBSERVATIONS AND ACTIONS. DIMENSION IN BRACKETS.

Observations o_t (27)	**Actions** a_t (4)
Arm joint torques (4)	Joint vel. (4)
Arm joint kinematics (pos., vel.) (8)	
Previous joint vel. command (4)	
Soil height (1)	
Bucket fill ratio (1)	
Bucket lin./ang. pos./vel. in cabin frame (6)	
Cabin pitch ang., pitch ang. rate in arm direction (2)	
Angle between bucket vel. and bottom plate (φ) (1)	

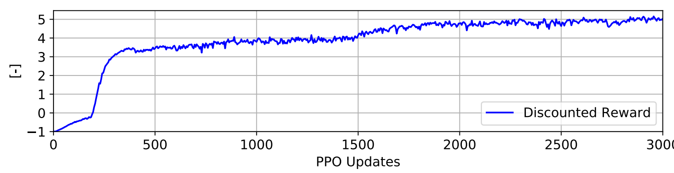


Fig. 4. Discounted reward during training. *Parameters*: Policy, value function hidden layers: 128, 128; Actor initial noise std.: 0.4, lower bound until update 1500: 0.1; Activation: LeakyReLU ($\alpha = 0.01$); Discount factor: 0.99; Control Δt: 0.15 s; Max. episode length: 19.95 s; Batch size: 68.1K; Entropy coeff.: 0.0; Learning rate: 5e-4; Value loss coeff: 0.5; Max. grad norm: 0.5; GAE λ: 0.95; Mini-batches: 4; Optimization epochs: 5; Clip range: 0.2

III. EXPERIMENTAL RESULTS[3]

A. Simulation Experiments

To validate the controller in simulation, we test the policy in different types of soils, ranging from very soft to very hard (see Fig. 5). The joint velocity commands from the agent (dashed) are almost perfectly tracked (solid) due to the well-tuned joint velocity controllers and perfect actuation with clipped maximum torque.

Fig. 5a depicts the digging trajectory in soft soil. Since the ground offers only little resistance, the bucket can be quickly filled by penetrating deep and curling without reaching machine torque limits. As per the reward definition, the linear velocity of the bucket always stays below 0.4 $\mathrm{m\,s^{-1}}$.

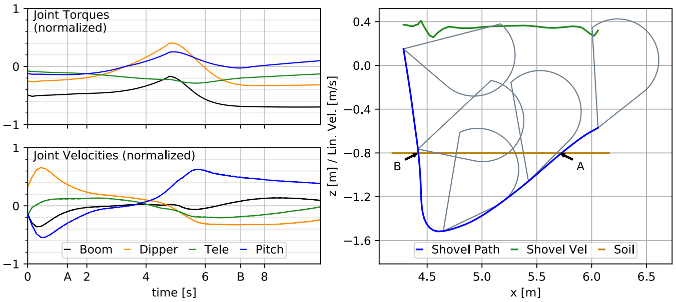
In medium soil (see Fig. 5b), the agent learns to penetrate less deep to prevent stalling or lifting up the machine. Therefore, the agent has to drag the bucket at a certain depth to fill the bucket. During the dragging phase, the dipper joint operates at its torque limit, which indicates efficient digging [6]. This is also intuitive, as the dipper joint contributes most to the horizontal motion of the bucket.

In hard soil (see Fig. 5c), the bucket can penetrate even less deep. The agent only manages to fill the bucket up to 80 %, even when starting with the arm entirely extended and dragging over the whole range of motion. It then pulls up

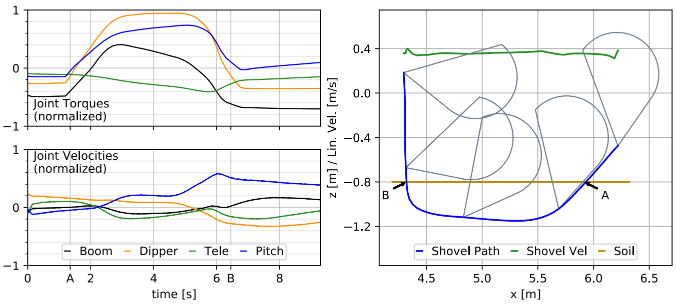
[1]https://github.com/leggedrobotics/rsl_rl

[2]We use a PC with an AMD Ryzen9 3950x CPU (@4.05GHz), 32GB of RAM, and an Nvidia RTX 2080s GPU. Experience generation takes place on the CPU and policy training on the GPU.

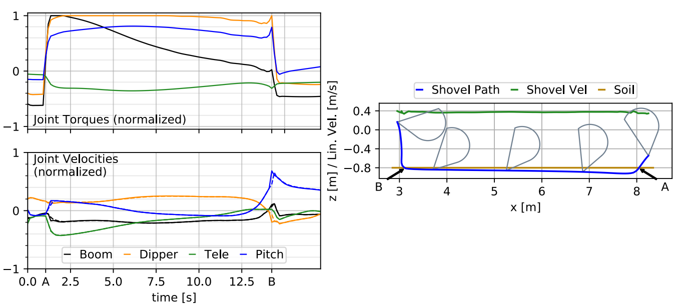
[3]Video of the experiments: https://youtu.be/0TJ6pFBb2kU.



(a) Soft soil. The final bucket fill ratio is 1. *Elapsed time*: Total: 10.0 s; In soil: 5.9 s. *Soil parameters*: $c = 0$, $c_a = 0$, $\phi = 0.4$, $\gamma = 19\text{e}3$, $\delta = 0.4$, $p_t = 1$.



(b) Medium soil. The final bucket fill ratio is 1. *Elapsed time*: Total: 9.3 s; In soil: 5.1 s. *Soil parameters*: $c = 15\text{e}3$, $c_a = 8\text{e}3$, $\phi = 0.3$, $\gamma = 20\text{e}3$, $\delta = 0.3$, $p_t = 50$.



(c) Hard soil. The final bucket fill ratio is 0.8. The agent pulls up the bucket before it is full to avoid self-collision. *Elapsed time*: Total: 17.3 s; In soil: 13.2 s. *Soil parameters*: $c = 100\text{e}3$, $c_a = 50\text{e}3$, $\phi = 0.8$, $\gamma = 21\text{e}3$, $\delta = 0.5$, $p_t = 250$.

Fig. 5. Simulation experiments. *Limits*: Force/Torque: $[-200, 140]$ kNm, $[-130, 170]$ kNm, $[-100, 190]$ kN, $[-50, 85]$ kNm; Velocity: $[-0.3, 0.3]$ rad/s, $[-0.6, 0.6]$ rad/s, $[-0.4, 0.4]$ m/s, $[-0.8, 0.8]$ rad/s

to prevent self-collision as per the definition of the desired terminal state. All experiments show very smooth velocity commands with minimal direction changes, which is critical for a successful transfer to the actual machine.

B. Hardware Requirements and Description

To validate the proposed method in reality, we show experiments on a modified Menzi Muck M545, a 12-ton hydraulic walking excavator [43], and, to begin with, briefly recapitulate the requirements. Training the controller in simulation relies on a kinematic and an approximate dynamic model of the machine. The agent outputs joint velocity commands, which requires two different joint velocity controllers for simulation and deployment. On the M545, we use the standard proportional valves in the main stage of the hydraulics. Automatic operation is enabled by retrofitting electrically driven valves in the low-pressure pilot stage. Joint velocity control is achieved with PID plus feed-forward valve-flow controllers, which output currents that are applied to the solenoids of the valves. Joint torques are estimated by measuring the cylinder pressures and knowing the surface areas of the pistons. The machine's orientation with respect to the world is measured with an Inertial Measurement Unit (IMU). Kinematic joint states of the arm are measured with IMUs and of the legs with magnetostrictive sensors. Today's modern excavators are already equipped ex works with the required sensors and valves. Hence, retrofitting parts is unnecessary for such machines.

C. Hardware Experiments

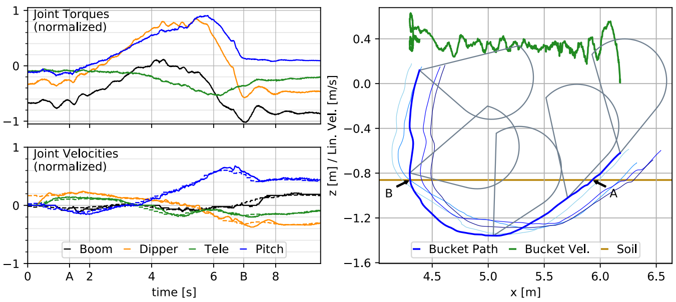
Fig. 6 shows the results of testing the controller on the real machine in different scenarios. The bucket is placed manually above the ground before activating the controller. We repeat each experiment multiple times to verify the consistency and report the final bucket fill factor and the elapsed time. Note that the shovel fill factor is only an approximation as it is computed by integrating the bucket path through the soil and assumes that all the material ends up in the bucket. Joint torques and desired and measured velocities are shown for the bold bucket path. The other runs are shown in thin lines and show a similar course (torques and velocities are omitted for clearness of the plots). The variations arise from differences in the initialization and soil conditions which slightly change after each dig.

Fig. 6a shows digging in soft to medium soil. The trajectories lie in between soft and medium soil in simulation (see Fig. 5a and Fig. 5b). Since the joint velocities do not perfectly follow the commands due to delays and deadzones in the hydraulic actuation [43], the bucket velocity slightly exceeds 0.4 $\mathrm{m\,s^{-1}}$.

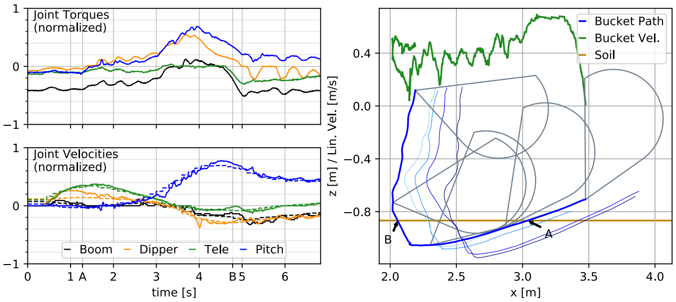
In Fig. 6b we place the bucket very close to the excavator. Since the agent was trained to avoid self-collisions, thus pulling the bucket up if it is too close to the machine, the bucket is only partially filled.

To test the controller in hard soil, we buried a block of granite at a shallow depth because the soil on the test site is relatively soft. As is shown in Fig. 6c, the bucket can only penetrate to the depth of the rock and subsequently drags the bucket over the rock while still applying large forces also downwards with the boom joint, without lifting up the machine from the ground, similar to the simulation experiment (see Fig. 5c). This is essential to rip open hard ground and excavate efficiently. At the end of the block, the bucket reaches softer soil and penetrates deeper to build up again large forces to fill up the bucket quickly. The bucket velocity stays below the threshold of 0.4 $\mathrm{m\,s^{-1}}$ except for the moment of transition between hard and soft soil where the joint velocity controllers cannot react fast enough.

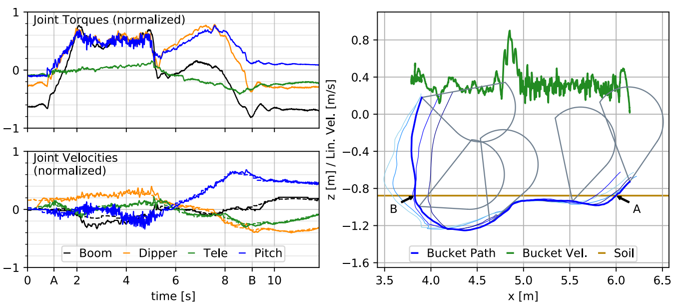
Finally, we demonstrate that the machine is indeed capable of lifting itself up if the digging trajectory is not adjusted to the encountered soil conditions. We manually push the boom downwards while pulling the dipper towards the machine on the hard soil, as if we were to excavate a deep, round trajectory intended for soft soil (see Fig. 7). The joint torques, in particular also at the boom joint, are so high that the front wheels of the machine are lifted. The torque of the boom joint



(a) Soft soil. The bucket is completely filled with each scoop. *Elapsed time mean/std*: Total: 9.6/0.53 s; In soil: 6.0/0.20 s.



(b) Soft soil, close to the base. The agent pulls the bucket up before it is full to avoid self-collision. *Bucket fill ratio mean/std*: 0.32/0.068. *Elapsed time mean/std*: Total: 6.4/0.46 s; In soil: 3.4/0.26 s.



(c) Abruptly changing soil conditions (see Fig. 1). A rock is buried in the first part of the excavation path. The bucket is completely filled with each scoop. *Elapsed time mean/std*: Total: 12.0/0.17 s; In soil: 8.0/0.30 s.

Fig. 6. Hardware experiments. *Limits*: As per Fig. 5.

exceeds 100 % because we neglect the fact that the torque limits are position-dependent due to the cylinder-joint linkage mechanisms.

IV. Conclusion and Discussion

In this work, we presented a controller that can dig adaptively in different soil types without stalling or lifting up the machine. The controller is trained in simulation with RL, where it learns to excavate by interacting with a large variety of different soils, just like human operators gain experience. For fast simulation and massive data generation with different soils, we use an analytical soil model that computes the major soil characteristics based on geotechnical parameters, which we heavily randomize during training. Unlike other approaches, the proposed method does not require prior knowledge about the working site's soil conditions. The controller can adapt online to varying soil conditions only based on proprioceptive measurements readily available on modern excavators. The desired behavior is shaped by defining simple and intuitive rewards and terminal conditions, which

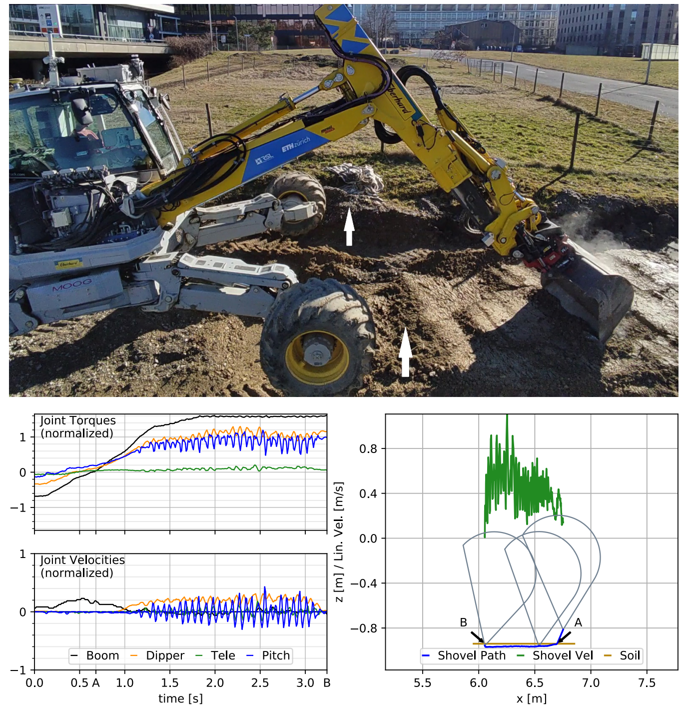


Fig. 7. Manually trying to excavate a deep round trajectory in hard soil. We aborted the experiment because the machine was lifted up, which never happened using the proposed excavation controller.

can be easily adapted depending on particular requirements or a different platform. The execution of the trained controller for deployment is computationally extremely cheap, such that resources can be used for other operations.

This work relies on some assumptions and simplifications. However, the proposed approach is flexible enough to address these in future developments. In particular, we assume horizontal ground, which can be addressed by adding perception sensors and providing this information as additional observation to the agent [44]. The analytical soil model used in this work can account for sloped terrains [31]. Also, the bucket fill volume can then be estimated more accurately [17]. Even though we already observed that the agent could react to changing soil conditions within one scoop, obstacles such as buried objects can be simulated explicitly by defining areas with hard soil properties to train the agent to contour or excavate them. Another simplification concerns the torque limits, which are assumed to be constant. However, due to the joint linkage configurations, the actual torque limits are position-dependent, which can be easily implemented. The cabin turn position should also be randomized in the future to account for changing machine stability depending on the digging direction. These extensions increase the state space, which will result in an increased training time. However, recent developments in parallel computing make it possible to train complex RL policies for real-world robotic problems within minutes [45].

Besides improving the generalizability, future work will require using the presented excavation controller in a practical real-world application, such as for example excavating an entire building pit. Adaptive digging, as shown in this letter, can then be combined with other controllers, which have been designed to track desired trajectories with high accuracy [46]. Furthermore, excavation speed was not a particular focus of this work. It should be optimized to maximize the working efficiency, which becomes important in larger applications.

REFERENCES

[1] V. Chacko, H. Yu, S. Cang, and L. Vladareanu, "State of the Art in Excavators," in *Proceedings of ICAMechS*, 2014, pp. 481–488.

[2] D. A. Bradley and D. W. Seward, "The Development, Control and Operation of an Autonomous Robotic Excavator," *JINT: Theory and Applications*, vol. 21, no. 1, pp. 73–97, 1998.

[3] Y. Zhang, Z. Sun, Q. Sun, Y. Wang, X. Li, and J. Yang, "Time-jerk optimal trajectory planning of hydraulic robotic excavator," *AIME*, vol. 13, no. 7, pp. 1–13, 2021.

[4] Y. Yang, P. Long, X. Song, J. Pan, and L. Zhang, "Optimization-Based Framework for Excavation Trajectory Generation," *IEEE RA-L*, vol. 6, no. 2, pp. 1479–1486, 2021.

[5] 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 *Proceedings of IEEE/RSJ IROS*, 2021, pp. 2135–2142.

[6] F. E. Sotiropoulos and H. H. Asada, "A Model-Free Extremum-Seeking Approach to Autonomous Excavator Control Based on Output Power Maximization," *IEEE RA-L*, vol. 4, no. 2, pp. 1005–1012, 2019.

[7] H. Cannon, "Extended Earthmoving with an Autonomous Excavator," Master's thesis, Carnegie Mellon University, Pittsburgh, PA, 1999.

[8] Y. Yang, J. Pan, P. Long, X. Song, and L. Zhang, "Time Variable Minimum Torque Trajectory Optimization for Autonomous Excavator," *arXiv*, 2020.

[9] 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," *IMechE, Part K: Journal of Multi-body Dynamics*, vol. 233, pp. 598–616, 2019.

[10] Y. B. Kim, J. Ha, H. Kang, P. Y. Kim, J. Park, and F. C. Park, "Dynamically Optimal Trajectories for Earthmoving Excavators," *Automation in Construction*, vol. 35, pp. 568–578, 2013.

[11] H. Yoshida, T. Yoshimoto, D. Umino, and N. Mori, "Practical Full Automation of Excavation and Loading for Hydraulic Excavators in Indoor Environments," in *Proceedings of IEEE CASE*, 2021, pp. 2153–2160.

[12] R. J. Sandzimier and H. H. Asada, "A Data-Driven Approach to Prediction and Optimal Bucket-Filling Control for Autonomous Excavators," *IEEE RA-L*, vol. 5, no. 2, pp. 2682–2689, 2020.

[13] H. Shao, H. Yamamoto, Y. Sakaida, T. Yamaguchi, Y. Yanagisawa, and A. Nozue, "Automatic Excavation Planning of Hydraulic Excavator," in *Proceedings of ICIRA*, vol. 5315, 2008, pp. 1201–1211.

[14] 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 TCST*, vol. 23, no. 5, pp. 1754–1769, 2015.

[15] H. S. Park, D. V. Dang, T. T. Nguyen, and N. T. Le, "Implementation of a Virtual Autonomous Excavator," *Transactions of Famena*, vol. 41, no. 3, pp. 65–80, 2017.

[16] M. Dunbabin and P. Corke, "Autonomous Excavation Using a Rope Shovel," *STAR*, vol. 25, pp. 555–566, 2006.

[17] D. Jud, P. Leemann, S. Kerscher, and M. Hutter, "Autonomous Free-Form Trenching Using a Walking Excavator," *IEEE RA-L*, vol. 4, no. 4, pp. 3208–3215, 2019.

[18] D. Jud, G. Hottiger, P. Leemann, and M. Hutter, "Planning and Control for Autonomous Excavation," *IEEE RA-L*, vol. 2, no. 4, pp. 2151–2158, 2017.

[19] D. Schmidt, M. Proetzsch, and K. Berns, "Simulation and Control of an Autonomous Bucket Excavator for Landscaping Tasks," in *Proceedings of IEEE ICRA*, 2010, pp. 5108–5113.

[20] T. Groll, S. Hemer, T. Ropertz, and K. Berns, "Autonomous trenching with hierarchically organized primitives," *Automation in Construction*, vol. 98, pp. 214–224, 2019.

[21] Quang Ha, M. Santos, Quang Nguyen, D. Rye, and H. Durrant-Whyte, "Robotic Excavation in Construction Automation," *IEEE RAM*, vol. 9, no. 1, pp. 20–28, 2002.

[22] B. Son, C. Kim, C. Kim, and D. Lee, "Expert-Emulating Excavation Trajectory Planning for Autonomous Robotic Industrial Excavator," in *Proceedings of IEEE/RSJ IROS*, 2020, pp. 2656–2662.

[23] Q. Lu and L. Zhang, "Excavation Learning for Rigid Objects in Clutter," *IEEE RA-L*, vol. 6, no. 4, pp. 7373–7380, 2021.

[24] Q. Lu, Y. Zhu, and L. Zhang, "Excavation Reinforcement Learning Using Geometric Representation," *IEEE RA-L*, 2022.

[25] C. Schenck, J. Tompson, S. Levine, and D. Fox, "Learning Robotic Manipulation of Granular Media," in *Proceedings of CoRL*, 2017, pp. 239–248.

[26] 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 T-ASE*, vol. 14, no. 1, pp. 249–259, 2017.

[27] 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.

[28] K. Matsumoto, A. Yamaguchi, T. Oka, M. Yasumoto, S. Hara, M. Iida, and M. Teichmann, "Simulation-based Reinforcement Learning Approach towards Construction Machine Automation," *ISARC*, pp. 457–464, 2020.

[29] I. Kurinov, G. Orzechowski, P. Hamalainen, and A. Mikkola, "Automated Excavator Based on Reinforcement Learning and Multibody System Dynamics," *IEEE Access*, vol. 8, pp. 213 998–214 006, 2020.

[30] J. Hwangbo, J. Lee, and M. Hutter, "Per-Contact Iteration Method for Solving Contact Dynamics," *IEEE RA-L*, vol. 3, no. 2, pp. 895–902, 2018.

[31] Borinara Park, "Development of a Virtual Reality Excavator Simulator: a Mathematical Model of Excavator Digging and a Calculation Methodology," Ph.D. dissertation, Virginia Polytechnic Institute and State University, 2002.

[32] "Geotech Data," https://www.geotechdata.info/parameter/, accessed: 2022-02-01.

[33] "Fine Geo Software," https://www.finesoftware.eu/help/geo5/en/earth-pressures-01/, accessed: 2022-02-01.

[34] H. S. Yu and G. T. Houlsby, "Finite Cavity Expansion in Dilatant Soils: Loading Analysis," *Géotechnique*, vol. 41, no. 2, pp. 173–183, 1991.

[35] A. R. Reece, "The Fundamental Equation of Earth-Moving Mechanics," *IMechE*, vol. 179, pp. 16–22, 1964.

[36] J. V. Perumpral, R. D. Grisso, and C. S. Desai, "A Soil-Tool Model Based on Limit Equilibrium Analysis," *Transactions of the ASAE*, vol. 26, no. 4, pp. 991–995, 1983.

[37] N. Bennett, A. Walawalkar, M. Heck, and C. Schindler, "Integration of Digging Forces in a Multi-Body-System Model of an Excavator," *IMechE, Part K: Journal of Multi-body Dynamics*, vol. 230, no. 2, pp. 159–177, 2016.

[38] J. JAKY, "The Coefficient of Earth Pressure at Rest," *Society of Hungarian Architects and Engineers*, pp. 355–358, 1944.

[39] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum Learning," in *Proceedings of ICML*, vol. 2, no. 3, New York, New York, USA, 2009, pp. 1–8.

[40] M. Hutter, P. Leemann, S. Stevsic, A. Michel, D. Jud, M. Hoepflinger, R. Siegwart, R. Figi, C. Caduff, M. Loher, and S. Tagmann, "Towards Optimal Force Distribution for Walking Excavators," in *Proceedings of ICAR*, 2015, pp. 295–301.

[41] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," *arXiv*, 2017.

[42] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-Dimensional Continuous Control Using Generalized Advantage Estimation," *arXiv*, 2015.

[43] D. Jud, S. Kerscher, M. Wermelinger, E. Jelavic, P. Egli, P. Leemann, G. Hottiger, and M. Hutter, "HEAP - The Autonomous Walking Excavator," *Automation in Construction*, vol. 129, 2021.

[44] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning Robust Perceptive Locomotion for Quadrupedal Robots in the Wild," *Science Robotics*, vol. 7, no. 62, 2022.

[45] N. Rudin, D. Hoeller, P. Reist, and M. Hutter, "Learning to Walk in Minutes Using Massively Parallel Deep Reinforcement Learning," in *Proceedings of CoRL*, 2022, pp. 91–100.

[46] P. Egli and M. Hutter, "A General Approach for the Automation of Hydraulic Excavator Arms Using Reinforcement Learning," *IEEE RA-L*, vol. 7, no. 2, pp. 5679–5686, 2022.