UMEÅ UNIVERSITY

High-performance autonomous wheel loading

- a computational approach

Koji Aoshima

THESIS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF PHYSICS
UMEÅ UNIVERSITY
SWEDEN 2025

High-performance autonomous wheel loading - a computational approach
Koji Aoshima
ISBN 978-91-8070-567-7 (Print)
ISBN 978-91-8070-568-4 (Digital)

Department of Physics
Umeå University
SE-907 36 Umeå
Sweden
Telephone: +46 (0)90-786 50 00

Cover: Diggability map colored by a loading performance prediction model.

Electronic version available at: `https://www.diva-portal.org`
Printed by: Scandinavian Print Group, Umeå, Sweden 2025

Abstract

Smart and autonomous earthmoving equipment enhances energy efficiency, productivity, and safety at construction sites and mines. The innovations provide means to reach high-set sustainability goals and be profitable despite increasing labor shortages. In addition, recent technological breakthroughs in artificial intelligence highlight the potential of superhuman capabilities to further enhance operations. This thesis presents a computational approach to end-to-end optimization of autonomous wheel loaders operating in a dynamic environment. Wheel loaders are mainly used for repeatedly loading material and carrying it to load receivers in quarries and mines. The difficulty lies in that each loading action alters the state of the material pile. The resulting state affects the possible outcomes of the subsequent loading process and, ultimately, the total performance. Thus, the challenge is to achieve both autonomous and high-performance wheel loading over a sequence of tasks. Achieving this requires the ability to predict future outcomes and account for the cumulative effect of loading actions. The thesis constructs a real-time wheel loader simulator, develops world models for sequential loading actions with evolving pile states, formulates the end-to-end optimization problem, and introduces a look-ahead tree search method to solve the problem. These contributions provide insights into utilizing physics-based simulation in combination with machine learning to further improve sustainability in mining and construction.

List of publications

This thesis is based on the following papers:

I **Simulation-Based Optimization of High-Performance Wheel Loading**
K. Aoshima, E. Wadbro, M. Servin
Proceedings of the 38th ISARC, 688-695, Dubai, UAE (2021).

II **Examining the simulation-to-reality gap of a wheel loader digging in deformable terrain**
K. Aoshima, M. Servin
Multibody System Dynamics (2024).

III **World Modeling for Autonomous Wheel Loaders**
K. Aoshima, A. Fälldin, E. Wadbro, M. Servin
Automation 5(3), 259-281 (2024).

IV **Optimizing wheel loader performance: an end-to-end approach**
K. Aoshima, E. Wadbro, M. Servin
Manuscript (2024).

Contents

1 Introduction

Smart and autonomous earthmoving equipment has enhanced energy efficiency, productivity, and safety at construction sites and mines. In harsh mining environments, fully automated dump trucks operate continuously. Compared to manually operated dump trucks, the automated trucks are managed by fewer operators from safe control rooms. Semi-automated excavators and bulldozers can accurately shape the terrain, regardless of the operator's skill. The planning of earthmoving operations is enhanced by integrating simulation and site model data from drone scanning of the terrain. This eliminates manual measurement processes, and re-planning can be done more regularly. These examples highlight how site optimization through automation and digital transformation provides a sustainable way for industries to maximize productivity and address labor shortages. In addition, recent research demonstrates that robotic excavators can handle complex tasks robustly [1, 2]. These studies highlight the potential of superhuman capabilities by leveraging artificial intelligence technologies to further enhance smart construction.

This thesis focuses on one particular type of machine, the wheel loader, which is mainly used for loading materials and carrying them to load receivers in quarries and mines. The loading process includes the task of bucket filling. The task requires coordinated actions of lifting and tilting the bucket while also carefully adjusting traction for an efficient interaction with the soil. This energy-consuming and skill-intensive operation has already been semi-automated to reduce operator fatigue. Recent research has demonstrated that the robustness of bucket filling controllers can be increased using reinforcement learning to adapt loading actions to different soil conditions [3–8]. However, these studies have been limited to single bucket fillings and have not considered the task of sequential loading from a pile. In practice, bucket filling actions are repeated throughout the sequential loading cycle [9]. The difficulty lies in that each loading action alters the pile state. The resulting state affects the possible outcomes of the subsequent loading process and, ultimately, the total performance. If not managed properly, this could lead to unsafe situations. Thus, the challenge is to achieve both autonomous and high-performance wheel loading over a sequence of tasks, which requires the ability to predict future outcomes and account for the cumulative effect of loading actions.

This thesis presents a computational approach to end-to-end optimization of sequential loading tasks in a dynamic environment. The approach leverages physics-based simulation to create a large dataset, subsequently used to train deep neural networks as world models to predict the outcome of a single loading action based on the initial pile state. The total performance of a sequential loading task is predicted through repeated model inferences. Future predictions of various loading scenarios enable optimization of the action sequence by tree search.

1.1 Research focus and aim

The thesis explores how the combination of simulation and machine learning can be applied to autonomous wheel loader operations in the construction and mining industries.

The approach requires large datasets of state-action-outcome triplets from the loading cycle, involving interactions between the machine and the soil in a pile. Although data from field experiments are authentic, collecting such data in natural environments is impractical, especially if we are to include rare, sub-optimal, and hazardous situations. Therefore, we turn to simulation to systematically collect data in controllable virtual environments. The simulation model needs to accurately capture the physical behavior of the machine and the terrain it interacts with. For the soil dynamics, it is common to use the discrete element method (DEM); however, its computational speed is too slow for extensive use due to the large number of particles and contacts it involves. To address this, a wheel loader simulator is developed that employs a multiscale terrain model with real-time performance capability.

Considering the time of a single loading cycle, a wheel loader system has about 30 s to compute a plan of actions for the next loading cycle [10] while completing the current one. However, since an optimization process is likely to require numerous evaluations, real-time simulation is not sufficient for solving the problem. Therefore, we explore learning computationally efficient world models from simulated data and performing efficient searches with them.

Based on the current state-of-the-art, we have taken four steps to advance the science of autonomous wheel loaders in dynamic environments.

Paper I: Formulating an optimization problem for the sequential loading task to approach it systematically and understanding the problem to divide it into sub-problem for focused attention. Creating a physics-based simulator to study which causal relationships and modeling aspects are important.

Paper II: Testing the simulator, including a multiscale terrain model, to ensure it captures the outcome of real machine operations. Investigating the dependency of the sim-to-real gap on the level of model fidelity.

Paper III: Developing world models, as surrogates to the physics-based simulator, for fast predictions of the loading outcome. Investigating the feasibility of using the world models to solve the optimization, considering their accuracy and speed.

Paper IV: Solving the optimization problem by introducing a look-ahead tree search using world models. Studying the effect of using different prediction horizons.

1.2 Outline

The remainder of this thesis is organized as follows: Chapter 2 describes the wheel loader's working cycle and the requirements for high-performance operation. Chapter 3 presents our computational approach, which involves formulating end-to-end optimization using 3D multibody dynamics to model the machine and terrain, employing deep neural networks for world models, and implementing the search method. Chapter 4 presents the results and summarizes the four papers that form the basis of this thesis. Chapter 5 concludes with an overview of the contributions to the research field and outlines future work. The four papers are included at the end of the thesis.

2 Smart wheel loader operation

2.1 Wheel loader and the system

Essentially, the wheel loader is a versatile machine for loading and moving materials. Equipped with a suitable attachment tool, it can handle various materials, such as soil, rock, logs, waste, slag, and snow [11]. Although there is a wide range of machine sizes, ranging from around 1 to over 200 tonnes in machine weights, they share a common mechanical structure. The machine consists of a front and a rear frame, connected by a revolute joint for articulated steering. Rotational power is transmitted from the engine through a driveline to the four wheels for traction and velocity control, together with a brake system. A linkage mechanism controls the attachment tool with linear actuators for lifting and tilting. The machine can be autonomous if the actuators are electronically controlled using systems for localization and interoceptive and exteroceptive sensors, such as GNSS, LiDAR, and joint encoders.

2.2 Loading cycle

With a bucket attachment, the machine is typically operated on construction sites and quarries, where it repeatedly fills the bucket with soil from a pile and dumps it onto load receivers. The dump truck is placed close to the dig location, making it convenient for the wheel loader to approach the truck while it waits to be filled. The efficient loading pattern is called V-shape loading, due to the shape of the wheel loader transportation trajectory. The loading cycle is illustrated in Fig. 1.

A loading cycle can be seen as carrying out a sequence of sub-actions [12]. Assuming that the wheel loader is located at the receiver after having dumped its load, it starts going back in reverse while turning, brakes, and then goes forward while turning in order to approach the pile at a selected dig location. Then, the machine digs into the pile to scoop the material with a coordinated motion of the boom and bucket cylinders and the wheels for driving forward or in reverse [13]. No articulated turning is applied while digging. It is recommended in training sessions [14, 15] to approach the pile with the bucket lowered horizontally and touching the ground to clear it from residual soil. Once the machine has thrust into the pile, the traction should be adjusted to prevent slipping. This usually involves lifting the loading mechanism after penetration to increase the normal force on the wheels. Before breaking out from the pile, the bucket is flipped to spill excess load on the pile. After the breakout, the loader approaches the receiver with the bucket raised high before emptying the bucket into the body. This cycle is repeated to fill the body of the load receiver. For the subsequent loading cycles, it is recommended

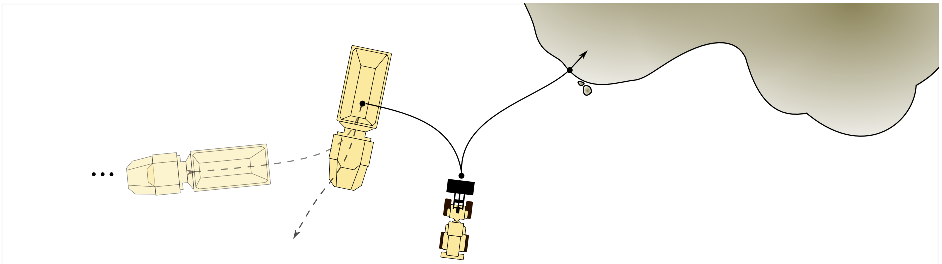

Figure 1: Illustration of the loading cycle with the characteristic V-shaped path.

to choose different dig locations, such as a concave-shaped pattern [16], to fill the bucket more easily [14, 15].

The loaded mass on the truck body should be adjusted by trimming it to comply with regulations and policies for the rated load [17]. When approaching the load receiver with a filled bucket, material can spill on the ground. The spillage negatively impacts tractive performance and control. Therefore, extra actions, such as clearing the ground or improving the pile shape, are sometimes necessary.

2.3 Loading control

2.3.1 Hierarchical planning and control

The loading tasks can be divided into sub-actions hierarchically. The machine is assumed to have a high-level planning and low-level control system, which work as the hierarchical planner architecture described in Refs. [18] and [19]. The high-level planning selects a sub-action, such as bucket filling and transportation, according to the environment state and then triggers the low-level control system to perform it with a motion plan or set of action parameters as input.

2.3.2 Bucket filling control

The loading performance depends on soil conditions [20], influenced by factors, such as particle size and shape distribution, chemical nature, and the geometric pile shape [21]. Although the "slicing cheese" motion for bucket filling appears optimal for conditions with lower digging resistance, other strategies might be better in other situations [22]. Much of the previous research has relied on bucket trajectories planned using kinematical models of the wheel loader. A problem is that although the kinematics may be realizable, they may be dynamically infeasible. The dig resistance is not known in advance and may exceed what is achievable with the actuators' limitations.

For this reason, force-based control is preferred for automatic bucket filling [23]. The admittance controller from [24] regulates the bucket actuator velocity using the measured boom cylinder force. That research was conducted for a load-haul-dump machine loading fragmented rock. Wheel loaders, however, require a somewhat different strategy because of a lower breakout force than the load-haul-dump machine [25].

2.4 High-performance wheel loader operations

Although each operated site puts different demands on the wheel loader, high performance means essentially bringing profit in a sustainable way. The essential performance metric is

$$\text{Performance} = \frac{\text{Production}}{\text{Cost}}. \tag{2.1}$$

For wheel loaders working in quarries or mines, production is typically measured by the loaded mass—the amount of material carried by the bucket at the end of each loading cycle. The main costs are the loading time, which is the time elapsed between the start and the end of each loading cycle, and the fuel, which depends on the energy consumed by the boom and bucket actuators and the forward drive. The energy includes the work required for filling the bucket, breaking out, raising the bucket, and accelerating both the vehicle and soil. Work is lost to substantial frictional dissipation internally in the soil and at the bucket-soil interface. The measurements of mass, time, and work are often seen as key factors for the metrics of productivity and efficiency [23]. Another aspect of the cost is related to slippage and tire wear. Large slips tend to happen at bucket filling when traction is insufficient for the digging resistance. Slippage is not only associated with energy consumption but also the increased cost of changing tires more frequently. Another risk is that material is spilled from the bucket onto the ground, causing poor traction, excessive vibrations, and tire damage. Moreover, the spilled material must be cleared away, adding additional time and energy costs.

Considering the total outcome over a sequence of cycles, one needs to account for the terrain alterations caused by each loading operation. There is a risk that the shape of the pile gradually deteriorates into a poor shape, making it difficult to load efficiently or possibly leading to dangerous situations.

When emptying the bucket, the wheel loader should adjust the unloading position at every cycle with consideration to the distribution of the loaded mass on the truck body [1, 26–29] and the height of the pile [30]. If these extra tasks can be reduced, the overall efficiency is improved. Evaluating the pile state quality might be useful to evaluate the performance of the loading action or analyze the results.

3 A computational approach

Our approach to high-performance autonomous loading is computational. The first step is to give the problem a mathematical formulation in the form of an optimization problem. That gives us a tool to approach it systematically and break it down into sub-problems. Some of the sub-problems already have their solutions and others are addressed in this thesis with different computational techniques. We use a physics-based simulation of a wheel loader and its dynamic environment to study which causal relationships and modeling aspects are particularly important. To ensure that the simulator adequately reflects the real physical system, we investigate the so-called reality gap and how it depends on the level-of-fidelity used. The developed simulator runs approximately at real-time speed. That enables systematic exploration of how the wheel-loader performance depends on the local pile state and the precise loading action. However, the optimization problem requires a large number of evaluated loading cycles, far more than the simulator performance allows. Therefore, we introduce a form of surrogate models using deep learning and synthetic training data produced with the simulator. The learned models predict the outcome of a specific loading, including the new state of the pile, the amount of mass ending up in the bucket, and how much time and energy it costs. The resulting *world model* is partially differentiable and many orders of magnitude faster to evaluate than the simulator. This makes it possible to search for near-optimal solutions to the optimization problem over longer horizons than a single loading. We explore a tree-search method and compare the obtained result to what a greedy strategy produces.

3.1 Problem formulation

Our goal is autonomous loading that is high-performing over time. This means we seek to find a sequence of actions $\{\boldsymbol{a}_n\}_{n=1}^N$ that renders a sequence of pile states $\{\boldsymbol{H}_n\}_{n=1}^N$ and performance measurements $\{\boldsymbol{\mathcal{P}}_n\}_{n=1}^N$ that are near-optimal given the capabilities of the machine. Note that these cannot be computed independently due to the strong dependency on the evolving pile state. We pose the end-to-end optimization problem of N sequential loading cycles as the problem of finding the sequence of actions that satisfies

$$\underset{\{\boldsymbol{a}_1,\dots,\boldsymbol{a}_N\}}{\operatorname{argmin}} \sum_{n=1}^{N} \boldsymbol{w}^\mathsf{T} \boldsymbol{\mathcal{P}}_n, \tag{3.1}$$

where $\boldsymbol{w}$ is a vector of positive weighting coefficients for controlling the trade-off between different performance measures. The input to the problem is some form of world model and its initial state. That includes the initial pile state, $\boldsymbol{H}_1$. The world model predicts the new state the world will transition into, given the previous state and a selected action. It also outputs observations for computing the performance. A world model can be an advanced physics-based simulator run at high spatial and temporal resolution, a simplified analytical model, or a data-driven model. We consider them all.

Each loading cycle has a duration T_n, requires some amount of work W_n, and results in a certain amount of mass M_n being removed from the pile and added to the receiver. Each sub-task contributes to the cycle time and work, while only the loading sub-task produces a mass measurement. We attribute each loading cycle with a performance that we measure by the performance vector

$$\boldsymbol{\mathcal{P}}_n = \left[\frac{M_0}{M_n}, \frac{T_n}{T_0}, \frac{W_n}{W_0}\right]^{\mathsf{T}}, \tag{3.2}$$

where M_0, T_0, and W_0 are some characteristic values used for normalization. The performance vector is a function of the pile state and the selected action, i.e., $\boldsymbol{\mathcal{P}}_n(\boldsymbol{H}_n, \boldsymbol{a}_n)$.

We consider the short loading cycle, where the wheel loader's task is to repeatedly load soil from a pile onto a load receiver. Each cycle can be divided into a sequence of sub-tasks: a first V-turn from the load receiver to the pile, bucket filling at the pile, a second V-turn to return to the load receiver, and finally dumping the load. These are illustrated in Fig. 2. These sub-tasks can be further divided, e.g., into motion planning and continuous control. We assume that the wheel loader is equipped with a low-level control system for each sub-task that is activated by selecting a set of sub-task action parameters so that $\boldsymbol{a}_n = [\boldsymbol{a}^{\mathrm{V1}}, \boldsymbol{a}^{\mathrm{load}}, \boldsymbol{a}^{\mathrm{V2}}, \boldsymbol{a}^{\mathrm{dump}}]$. Two actions that we pay special attention to are selecting the particular location where to dig into the pile and how to adapt the loading technique to the local shape of the pile.

3.2 Simulator

A wheel loader simulator was created through physics-based modeling. The wheel loader model was developed to approximately match the dimensions and physical properties of a Komatsu WA320-7. The simulator includes the vehicle's drivetrain, articulation, loading mechanism, and a 3D representation of the environment that includes the dynamics of the soil when interacting with it. A challenge in modeling is to select an appropriate fidelity level. We aimed for a model that could be run at roughly real-time speed while still capturing the essential features that allow us to measure the performance and track pile state transitions.

The wheel loader is therefore modeled as a rigid multibody system consisting of a front and rear frame, connected by a revolute joint for articulated steering, four wheels, and a parallel Z-bar linkage system for controlling the bucket relative to the

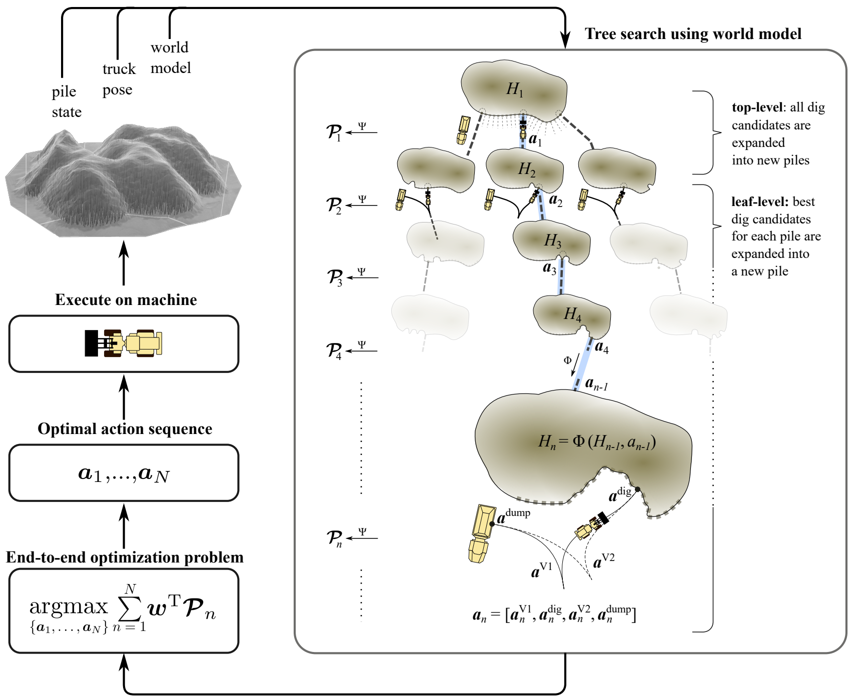

Figure 2: Overview of the tree search method for a wheel loader moving soil in short loading cycles from the evolving pile to a dump truck.

front frame. The driveline is modeled by a revolute motor transmitting rotational power to the front and rear wheel pairs via a main, front, and rear shaft, coupled with differentials. Each wheel is connected to the frame with a revolute joint and consists of a tire-hub pair, with finite elasticity with respect to radial, lateral, bending, and torsional displacements. The interaction with the environment is modeled in terms of frictional contacts.

Vehicle-terrain interaction is often performed using either the so-called DEM or semi-empirical models for the soil. DEM accurately captures many modes of soil dynamics but comes at a high computational cost. In terramechanics, there are two main semi-empirical models: the Bekker-Wong model [31] for wheel-terrain interaction and the Fundamental Equation of Earth-Moving Mechanics (FEE) [32]. These models are computationally efficient but limited to stationary conditions and idealized geometries. Therefore, we consider a new, but rather untested, method referred to as multiscale deformable terrain [33].

The simulated wheel loader is equipped with an admittance controller for automatic bucket filling. The admittance control, a type of force-feedback control, regulates the actuator speed according to the resistant force [24]. We parameterize the controller by four action parameters that determine the sensitivity of lifting and tilting the bucket to the dig resistance observed by the lift cylinders. This control helps prevent tire slip and getting stuck in the pile by limiting the traction or breakout force, adjusting the bucket lifting or tilting in response to the dig resistance. Admittance control is also utilized to make the bucket filling adaptable to different material types by changing the sensitivity [34]. Different values of the four action parameters render different dig trajectories while always respecting the system's force limits.

3.2.1 Multibody dynamics framework

We use non-smooth contacting multibody dynamics for modeling the vehicle and the soil, introduced in [35] and supported by AGX Dynamics [36]. Specifically, we use a maximal coordinate representation in terms of rigid bodies and kinematic constraints for joints, motors, and frictional contacts. The governing equations are

$$\boldsymbol{M}\dot{\boldsymbol{v}} = \boldsymbol{f} + \boldsymbol{G}_{\mathrm{j}}^T\boldsymbol{\lambda}_{\mathrm{j}} + \boldsymbol{G}_{\mathrm{c}}^T\boldsymbol{\lambda}_{\mathrm{c}}, \tag{3.3}$$

$$\varepsilon_{\mathrm{j}}\boldsymbol{\lambda}_{\mathrm{j}} + \eta_{\mathrm{j}}\boldsymbol{g}_{\mathrm{j}} + \tau_{\mathrm{j}}\boldsymbol{G}_{\mathrm{j}}\boldsymbol{v} = \boldsymbol{u}_{\mathrm{j}}, \tag{3.4}$$

$$\boldsymbol{\lambda}_{\mathrm{min}} \leq \boldsymbol{\lambda}_{\mathrm{j}} \leq \boldsymbol{\lambda}_{\mathrm{max}}, \tag{3.5}$$

$$\text{contact\_law}(\boldsymbol{g}_{\mathrm{c}}, \boldsymbol{v}_{\mathrm{c}}, \boldsymbol{\lambda}_{\mathrm{c}}), \tag{3.6}$$

with system mass matrix $\boldsymbol{M} \in \mathbb{R}^{6N_{\mathrm{b}}\times 6N_{\mathrm{b}}}$, external force $\boldsymbol{f} \in \mathbb{R}^{6N_{\mathrm{b}}}$, and velocity $\boldsymbol{v} \in \mathbb{R}^{6N_{\mathrm{b}}}$ that is the time derivative of the world frame maximal coordinates $\boldsymbol{x} \in \mathbb{R}^{7N_{\mathrm{b}}}$ (using quaternions for the orientation). The constraint forces in the Newton-Euler equation of motion (3.3) are represented in terms of the Lagrange multiplier $\boldsymbol{\lambda}$ and Jacobian $\boldsymbol{G}$. The constraints are divided into those for joints and motors, labeled with j, and those for contacts, labeled with c. Eq. (3.4) is a generic constraint equation. An ideal joint can be represented with $\varepsilon_{\mathrm{j}} = \tau_{\mathrm{j}} = \boldsymbol{u}_{\mathrm{j}} = 0$, in which case Eq. (3.4) expresses a holonomic constraint, $\boldsymbol{g}_{\mathrm{j}}(\boldsymbol{x}) = 0$. A non-ideal joint is modeled using finite compliance ε_{j} and viscous damping rate τ_{j}. A motor may be represented by a velocity constraint $\boldsymbol{G}_{\mathrm{j}}\boldsymbol{v} = \boldsymbol{u}_{\mathrm{j}}(t)$ with target speed $\boldsymbol{u}_{\mathrm{j}}(t)$, by setting $\varepsilon_{\mathrm{j}} = \eta_{\mathrm{j}} = 0$ and $\tau_{\mathrm{j}} = 1$. Range limits on the motor constraint forces may be imposed by Eq. (3.5). With N_{j} constrained and actuated degrees of freedom we have $\boldsymbol{\lambda}_{\mathrm{j}} \in \mathbb{R}^{N_{\mathrm{j}}}$ and $\boldsymbol{G}_{\mathrm{j}} \in \mathbb{R}^{N_{\mathrm{j}}\times 6N_{\mathrm{b}}}$.

Contact laws are imposed as inequality and complementarity conditions on the contact multiplier, $\boldsymbol{\lambda}_{\mathrm{c}} \in \mathbb{R}^{3N_{\mathrm{c}}}$, and relative contact velocity, $\boldsymbol{v}_{\mathrm{c}}$. The contact laws state that bodies should not interpenetrate in the normal direction and that there should be no relative slip in the tangential direction unless the tangential force reaches the Coulomb friction limit. Each contact multiplier is split $\boldsymbol{\lambda}_{\mathrm{c}} = [\lambda_{\mathrm{n}}; \boldsymbol{\lambda}_{\mathrm{t}}]$ into the normal and tangential components that are conditioned by the Coulomb law, $|\boldsymbol{\lambda}_{\mathrm{t}}| \leq \mu_{\mathrm{t}}\lambda_{\mathrm{n}}$.

The dynamic system is time-integrated using the SPOOK stepper [35], a first-order accurate discrete variational integrator developed particularly for fixed time-step simulation of multibody systems with non-ideal constraints and non-smooth dynamics. This enables real-time simulation of moderately sized systems using a time-step around $1 - 20$ ms. The time-discrete equations form a mixed complementarity problem (MCP), which is solved using the direct-iterative split solver in AGX [36]. The direct solver is applied to the vehicle system and its external contacts. It solves with high accuracy even for very ill-conditioned and over-constrained systems but does not scale well to large systems with many interconnections. Large contact systems, as arising in DEM simulations, are solved using an iterative solver based on the projected Gauss-Seidel (PGS) algorithm [37].

3.2.2 Discrete element method

The discrete element method (DEM) [38] is a popular method for simulating granular materials, such as soils. It is a versatile method for modeling soil dynamics in its different phases, which range from solid, dense liquid to gas, depending on the level of agitation. The soil is treated as a large collection of contacting particles, often of simplified shape and much larger than the actual particles. In full 3D, DEM simulations often involve hundreds of thousands or millions of particles, which is computationally demanding and time-consuming. We use a time-implicit, or nonsmooth, DEM following [39, 40]. It allows for large time-steps and strong coupling with the multibody dynamics, since the contacts are modeled in terms of kinematic constraints via Eqs. (3.3)-(3.6). The contact model is mapped to the Hertz's model for contact elasticity and also includes Coulomb friction and Newton's impact law. Rolling resistance constraints are included to capture the effect of the real particles having a non-spherical shape while the simulated particles are spherical bodies.

3.2.3 Multi-scale terrain model

For computationally efficient simulation of full-system dynamics, we use the multiscale deformable terrain model described in [33] and illustrated in Fig. 3. It can be regarded both as a reduced order model of DEM and as a multibody dynamics generalization of the FEE [32]. The idea is to resolve only the part of the terrain inside a well-localized region of shear failure. The terrain is assigned a set of bulk mechanical parameters for its physical behavior in its nominal bank state. When a bucket comes in contact with the terrain surface, the zone of active soil displacement is predicted. Only inside the active zone is the soil resolved by particles. These are modeled using DEM with specific mass density and contact parameters that ensure a bulk mechanical behavior consistent with the set bulk parameters. The reaction force on the bucket consists of contact with an aggregate body, which inherits the physical properties of the soil in the active zone, and a penetration resistance constraint, which is a function of the bucket geometry, soil parameters, and the soil pressure at the cutting edge.

3.3 The reality gap

A simulator is an idealized replica of a real system. It is unavoidable that it behaves somewhat differently although fed with identical control signals. The

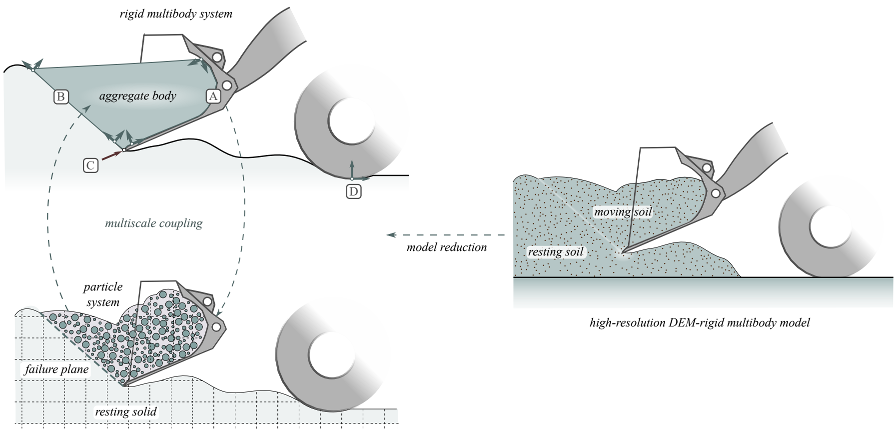

Figure 3: Illustration of the multiscale terrain model (left) adapted from [33]. It can be regarded as a heavily reduced version of the full DEM model (right). The predicted region of active soil is substituted with an aggregate body that couples with the vehicle dynamics (upper left). The soil dynamics inside the active region are modeled using DEM (lower left).

potential causes for the mismatch can broadly be categorized into model errors, numerical errors, and implementation errors. Model errors include unmodeled or oversimplified geometry and physics, inaccurate model parameters, and initial conditions. Actuator latency and noise are reportedly major sources of model errors when the system involves feedback control [41]. Common sources of numerical errors include using a spatial and temporal resolution that is too coarse. When simulations run over a long time, compared to the integration time-step, it is important to use numerically stable algorithms that prevent locally small errors from accumulating into large global errors.

In robotics and deep learning, the discrepancy between a simulated and real system is usually denoted as the reality gap or sim-to-real gap [42]. If the gap is significant, a solution developed in simulation will exhibit a simulation bias causing it to perform differently, and usually poorly, when transferred to the real system [43]. The gap is severe if the effort required to adapt the solution to the real domain is greater than its conception in the simulated environment. The reality gap may be considered small when it is less than the natural variations in different instances of the real system. Hence, there is no objective measure for the sim-to-real gap. It depends on the task that the system is intended to perform and is relative to the naturally occurring variations.

To quantify the sim-to-real gap, we consider time-discrete signals $f_{0:T} = [f_0, f_1, ..., f_N]$ and trajectories $\boldsymbol{x}_{0:T} = [\boldsymbol{x}_0, \boldsymbol{x}_1, ..., \boldsymbol{x}_N]$. For each scalar signal f_n, with a real reference $\hat{f}_n$, the instantaneous error at discrete time index n is denoted

$\varepsilon_n^f = f_n - \hat{f}_n$, and we compute the normalized mean absolute error (MAE) by

$$\mathcal{E}_f = \frac{1}{N}\sum_{n=0}^{N} \frac{\left|\varepsilon_n^f\right|}{f_{\text{norm}}}, \tag{3.7}$$

where f_{norm} is a normalizing reference value. For trajectories, a natural choice is the normalized mean Euclidean error (MEE)

$$\mathcal{E}_{\boldsymbol{x}}^{\text{MEE}} = \frac{1}{N}\sum_{n=0}^{N} \frac{\|\varepsilon_n^{\boldsymbol{x}}\|_2}{L_{\text{norm}}}, \tag{3.8}$$

with the Euclidean norm of the momentary trajectory error $\varepsilon_n^{\boldsymbol{x}} = \boldsymbol{x}_n - \hat{\boldsymbol{x}}_n$ and a normalizing length L_{norm}. However, if two trajectories are subject to relative phase shift or difference in speed but are otherwise similar, this is picked up by the momentary error and accumulated along the entire trajectory. This produces a large error, although they represent systems of very similar dynamics. One way to handle this is to use the similarity measure based on dynamic time warping (DTW) distance introduced in [44].

3.4 World models learned from data

In the context of reinforcement learning, a world model is a type of model that predicts an environment's future states based on current observations and actions [45]. This is usually accomplished by learning from data using deep neural networks. For this, Variational Autoencoders (VAEs) are often preferred due to their ability to effectively learn compact representations of complex data distributions.

We draw inspiration from [46], where a convolutional autoencoder was used for learning a reduced representation of the terrain surface combined with recurrent long short-term memory to learn the time-evolution of the surface during an earthmoving task. Based on this, a pile state predictor model was developed

$$\boldsymbol{H}_{n+1} = \boldsymbol{\Phi}(\boldsymbol{H}_n, \boldsymbol{a}_n). \tag{3.9}$$

First, an encoder network compresses the initial pile state into a lower-dimensional latent representation. Next, the latent state and the loading action are input to a multilayer perceptron (MLP), which predicts the latent representation of the new pile state. Finally, a decoder network constructs the resulting pile state from the predicted latent state. A performance predictor model

$$\boldsymbol{\mathcal{P}}_n^{\text{load}} = \boldsymbol{\Psi}(\boldsymbol{H}_n, \boldsymbol{a}_n) \tag{3.10}$$

was similarly developed as a convolutional neural network, where the MLP outputs the performance instead of a latent vector to be decoded.

3.5 Tree-search optimization

Tree search algorithms are discrete optimization methods that explore a structured search space, such as game trees, where possible moves are represented as a tree of sequential decisions. Essentially, tree searches can explore all possible actions in the search space, guaranteeing an optimal solution at an impractically high computational cost (exhaustive search), or they can exploit the option that looks best in the current situation, with the hope of finding the global optimum at a minimal computational cost (greedy strategy).

A balance between exploration and exploitation is effectively managed by the Upper Confidence Bound for Trees (UCT) in Monte Carlo Tree Search (MCTS) [47]. It evaluates each state by exploring various possible action sequences, allowing it to gather more information about potential outcomes and reduce short-term biases. MCTS executes numerous random simulations to evaluate action sequences probabilistically, considering that the reward is influenced not only by the immediate action but also by external factors, such as the opponent's actions in multiplayer games.

Our problem can be seen as a single-player game since the future pile state is determined only by the actions of the wheel loader. Therefore, we use a computationally faster and simpler algorithm, a look-ahead search. This search is a hybrid approach that builds on the greedy strategy by adding a limited exploration of some future steps and their consequences to the current evaluation. It helps to avoid some of the pitfalls of the greedy strategy, making it more suitable for problems where immediate choices are not always the best in the long run, while maintaining computational efficiency. Assuming that any action will neither change the states globally nor critically impact the performance, a few steps of future prediction can give the problem a greedy property. Therefore, a look-ahead search-based algorithm is well-suited for our problem.

4 Results: Summary of papers

This chapter presents the results and discusses the findings of our research by summarizing each paper.

Paper I: Simulation-Based Optimization of High-Performance Wheel Loading

This paper proposes a framework for utilizing a simulated environment to explore wheel loading actions that maximize performance in a task.

Although the loading task is repetitive, the changing environment after every loading cycle prevents physical experiments from consistently achieving high-performance loading actions. This motivates us to conduct systematic and repeatable experiments utilizing a simulated environment. The DEM could be a primary option for the simulation model; however, it seems difficult to simulate with various pile shapes and soil types due to computational costs. To simplify the model, either kinematic loading trajectories or quasi-static soil models can be used, but this may optimize an unrealistic trajectory or fail to capture the actual soil displacement. To support the complex dynamics and variable soil displacement in machine-soil interactions with real-time capability, we developed a wheel loading simulator that combines contact 3D multibody dynamics with a hybrid continuum-particle terrain model.

Our simulator demonstrated running a total of 270,000 simulations with different loading action parameters for a force-based controller to analyze their performances, including productivity and energy-efficiency, in relation to the pile slopes ranging from 10° to 40° across different soil properties, such as gravel, sand, and dirt. The results indicate that the preferred digging actions should be adjusted to the slope angle and take advantage of a steep pile slope. High digging speed favors high productivity, while energy-efficient loading requires a lower digging speed.

We conclude that our simulator can support executing a vast number of loading simulations to identify the combined actions and pile states that significantly affect loading performance.

Paper II: Examining the simulation-to-reality gap of a wheel loader digging in deformable terrain

This paper investigates how effectively a physics-based simulator can replicate a real wheel loader performing bucket filling, as well as the transferability of simulation results.

Simulators play a crucial role in heavy equipment development, offering a safe and efficient way to test in the early development stages. They are also expected to generate large amounts of annotated synthetic training data necessary for leveraging deep-learning methods. However, the computational speed of more finely resolved models, despite their high fidelity, remains a limiting factor. Additionally, there is little knowledge about how the reality gap depends on the simulator's level of resolution. To address this, we extended the wheel loader simulator developed in our previous work (Paper I) to vary the levels of fidelity. This allowed us to examine the differences in results between each other and real bucket filling operations. Two types of terrain models were prepared for the simulators: one using DEM for the terrain as finely resolved in particles and another employing a reduced multiscale model. We examined both types of simulators with different spatiotemporal resolutions, ranging between 50-400 mm and 2-500 ms. The sim-to-real error is assessed by comparing time-series data of vehicle motion and actuation forces, loaded mass, and mechanical work using the same sensing information as employed in the field tests. The operator's control of the vehicle is simulated using feedforward control across three different operational scenarios from the field tests. The mean of the sim-to-real error was found to be approximately 10 %, and there is a weak but positive trend with increasing the resolution on average compared to the significant difference in computational speed ranging between 10^{-4} and 5 times faster than real-time. Furthermore, we investigated the domain sensitivity of a force-feedback controller optimized in a real-time simulator when transferred to a much higher-fidelity simulator. The domain bias was observed to cause a 5 % reduction in bucket filling performance, even though the domain gap was approximately 15 %, suggesting the optimal control parameters may be transferable between the domains.

Overall, this research provides valuable insights that help bridge the gap between simulation and reality. It is expected to more effectively support the development of autonomous controllers in the construction and mining industries.

Paper III: World Modeling for Autonomous Wheel Loaders

This paper presents a method of learning world models for wheel loaders performing automatic loading actions on a pile of soil.

Previous research has focused on improving loading performance and robustness to adapt to soil properties while being limited to a single bucket filling. However, the practical task involves sequential loading, where each loading action alters the pile state. These state changes affect the possible outcomes of subsequent loading actions and, ultimately, the overall performance. Thus, an end-to-end approach requires the ability to instantaneously predict the cumulative effect of loading actions over a sequence of tasks. We developed two kinds of deep neural network models. One predicts loading performance, including loaded mass, loading time, and work for a single loading cycle, while the other, formulated as a VAE, predicts the resulting pile state after that cycle. Both models use inputs such as the initial heightmap of the pile and action parameters for an automatic bucket filling controller. The net outcome of a sequence of loading actions can then be predicted by repeated inferences of the model on the predicted pile state, thereby predicting its sequential evolution as well. To learn the models, we utilized a wheel loader simulator developed in our previous work (Paper II). The simulator's real-time capability enables us to generate a dataset comprising more than 10,000 random loading actions on various gravel pile shapes. The trained models achieved around 95 % accuracy in predicting loading performance in approximately 1.2 ms, and 97 % accuracy in predicting the resulting pile state in about 4.5 ms. The performance prediction was found to be even faster at the expense of accuracy by reducing the model size with the lower dimensional representation of the pile state using its slope and curvature.

The feasibility of long-horizon predictions was demonstrated with 40 sequential loading actions at a large pile. We found that the evolution of the pile state was accurately predicted under the assumption that the loading outcome depends only on the local state. The rate of accumulated prediction error increased starting around the 15th cycle, when the pile evolved into a state featuring steeper slopes, leading to avalanches that caused changes extending beyond the local region. This limitation can be addressed using a physics-based model for longer horizons. We conclude that the proposed models are capable of rapid evaluations to find near-optimal action plans.

Paper IV: Optimizing wheel loader performance: an end-to-end approach

This paper presents a method for solving the wheel loader end-to-end optimization problem in sequential loading.

To the best of our knowledge, this problem has not been scientifically investigated before. In Paper III, we developed world models capable of predicting loading performance and the resulting pile state within milliseconds. These models were designed to evaluate the cumulative effects of loading actions over a sequential loading task. Using this, we developed a look-ahead tree search algorithm. The look-ahead search identifies the optimal action sequence by evaluating a large number of action candidates for a loading cycle and recursively expanding them into new action candidates for the subsequent cycles based on the evolving pile state. The performance of the action candidates at a cycle is predicted by the world model and added to the transportation cost for the V-paths computed using numerical integration along B-splines. The computational time for evaluating a loading cycle was 73.5 ms on average.

The look-ahead search was tested on ten randomized piles. A search depth of 1 corresponds to a greedy strategy, which exploits the actions that maximizes the immediate loading performance. With a search depth of 4 (and above), the look-ahead search was found to improve the total performance by 6 % over the greedy strategy and 14 % better that optimizing for the transportation cost only.

The look-ahead search continuously select dig locations relatively close to the dump truck where pile shape was not necessarily attractive for the immediate loading action but improves after digging there by getting steeper. This paper concludes that the look-ahead search leveraging world models utilize the ability of future prediction to maximize the total performance of sequential loading task.

5 Conclusions

The thesis presents a computational approach to high-performance autonomous wheel loading. The approach consists of formulating the end-to-end optimization problem (Paper I and IV), constructing a real-time wheel loader simulator (Paper I and II), developing world models for sequential loading action in a changing pile (Paper III), and proposing a tree search method using the world models for look-ahead to solve the problem (Paper IV).

The thesis provides insights into utilizing real-time physics simulation in combination with machine learning. The real-time simulation contributes to the development of autonomous control in the early development stages. This thesis provides insights that help bridge the gap between simulation and reality. The methodology helps overcome the impracticality of relying solely on physical experiments to develop solutions.

Planning high-performance actions involves considering their consequences for future loadings, multiple objectives and constraints, and huge data sets that humans cannot easily understand. This has the potential to maximize the site performance and address labor shortages. The computational approach requires information about the machine's position and pose, the heightfield of the work area, and the lift cylinder force for the bucket filling admittance control. Therefore, it can be applied to autonomous loading equipment without adding any special sensors that are not already used for the system. Since the single loading cycle is divided into sub-tasks with sub-action parameters, the controllers for each sub-task are exchangeable. The thesis can support future research in improving site optimization, further refining route planning, and advancing the use of autonomous ground vehicles in construction and mining operations.

5.1 Future work

The current assumptions define the conditions under which our approach works effectively. The approach could be adapted for a wider range of practical applications by eliminating these assumptions.

The current limitation is the assumption of homogeneous and non-cohesive soil. In real sites, the machine would also face inhomogeneous and cohesive soil. However, it has been challenging to overcome the computational cost of simulating a high-resolution terrain model. As discussed in Paper II, our simulator demonstrated the ability to generate nearly optimal solutions for a higher-resolution domain, maintaining fidelity while preserving real-time capabilities through the model order reduction technique. Therefore, the next question is whether we can utilize the homogeneous soil while minimizing the domain sensitivity to actual material variations.

The pile state predictor in our world model assumes that pile avalanches only occur within the local region around the dig locations. This assumption does not hold for very steep piles, where large avalanches can form and affect areas beyond the local region. This becomes an obstacle for predicting the outcome of longer loading sequences. As researched in Paper III, a physics-based avalanching method, such as cellular automata, may be necessary, although it would increase computational cost. The question, then, is how we can efficiently integrate such an avalanching method with our predictor.

Our research focused on optimizing the actions of a wheel loader with the assumption of a fixed dump location. In this situation, the overall task outcome depends solely on the wheel loader's action sequence, framing the problem as a single-player game, as researched in Paper IV. In practice, a typical work site involves a wheel loader and dump trucks, which arrive sequentially after being filled with material. Depending on the constraints, the optimization of dig and dump locations, as well as the actions of both the wheel loader and the dump trucks, can be treated as a multiplayer game. While solving the multiplayer game problem becomes more complex with the increasing number of decision variables, it offers the potential for achieving a global optimum in site performance.

Contributions

Short description of the contributions by the author to each of the papers included in the thesis.

Paper I

The author developed the simulator, collected the data, and performed the analysis. He also contributed to the study conception, writing and reviewing the article, and approved the final manuscript.

Paper II

The author developed the simulation scene, collected the data, developed the models, and performed the analysis. He also contributed to the study conception, writing and reviewing the article, and approved the final manuscript.

Paper III

The author collected the field data, developed the wheel loader simulation model. He also contributed to the study conception, simulations, data analysis, writing and reviewing the article, and approved the final manuscript.

Paper IV

The author developed the single loading cycle outcome predictor and the search method, collected the data, and performed the analysis. He also contributed to the study conception, writing and reviewing the article, and approved the final manuscript.

Acknowledgments

First and foremost, I am extremely grateful to my supervisors, Martin Servin, Eddie Wadbro, and Takehiro Komatsu, for their invaluable advice, continuous support, and patience during our doctoral project. Their immense knowledge and plentiful experience have encouraged me throughout my academic research and daily life. Martin-san, thanks for always trying to synchronize our vision and motivating the research work. You always believe in ourselves. That has made it possible for us to get good findings on the long journey. Eddie-san, when I struggled to get the mathematical insight, you supported me by writing down precise algorithms and mathematical equations on the whiteboard, which always worked out on my computer without any warning. Komatsu-san, thanks for being patient and trying not to make me rush and lose myself. I couldn't concentrate on my work without your gentleness. Everyone else at the office, especially Erik-san and Arvid-san, I couldn't manage our research without your technical support on my study. You saved me! Finally, all of my family also got used to living in Umeå well with making many friends, acquiring the language, and enjoyed life by their own way. Your incredible development really encouraged me that we can do what we want to do overcoming what we saw as a difficulty before. What we made here lives endlessly even after going back my home country.

Bibliography

[1] Liangjun Zhang, Jinxin Zhao, Pinxin Long, Liyang Wang, Lingfeng Qian, Feixiang Lu, Xibin Song, and Dinesh Manocha. An autonomous excavator system for material loading tasks. *Science Robotics*, 6(55):eabc3164, 2021.

[2] Ryan Luke Johns, Martin Wermelinger, Ruben Mascaro, Dominic Jud, Ilmar Hurkxkens, Lauren Vasey, Margarita Chli, Fabio Gramazio, Matthias Kohler, and Marco Hutter. A framework for robotic excavation and dry stone construction using on-site materials. *Science Robotics*, 8(84):eabp9758, 2023.

[3] Siddharth Dadhich, Fredrik Sandin, Ulf Bodin, Ulf Andersson, and Torbjörn Martinsson. Adaptation of a wheel loader automatic bucket filling neural network using reinforcement learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–9, 2020.

[4] Osher Azulay and Amir Shapiro. Wheel loader scooping controller using deep reinforcement learning. *IEEE Access*, 9:24145–24154, 2021.

[5] Heshan Fernando and Joshua Marshall. What lies beneath: Material classification for autonomous excavators using proprioceptive force sensing and machine learning. *Automation in Construction*, 119:103374, 2020.

[6] Sofi Backman, Daniel Lindmark, Kenneth Bodin, Martin Servin, Joakim Mörk, and Håkan Löfgren. Continuous control of an underground loader using deep reinforcement learning. *Machines*, 9(10), 2021.

[7] Nataliya Strokina, Wenyan Yang, Joni Pajarinen, Nikolay Serbenyuk, Joni Kämäräinen, and Reza Ghabcheloo. Visual rewards from observation for sequential tasks: Autonomous pile loading. *Frontiers in Robotics and AI*, 9, 2022.

[8] Eric Halbach, Joni Kämäräinen, and Reza Ghabcheloo. Neural network pile loading controller trained by demonstration. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 980–986, 2019.

[9] Sanjiv Singh and Reid G Simmons. Task planning for robotic excavation. In *IROS*, volume 92, pages 1284–1291, 1992.

[10] Reno Filla. *Quantifying operability of working machines*. PhD thesis, Linköping University Electronic Press, 2011.

[11] Volvo Construction Equipment. Everything you need to know about wheel loaders, 2021. Available from https://www.volvoce.com/asia/en-as/about-us/news/2021/everything-you-need-to-know-about-wheel-loaders/. Accessed: 2024-12-01.

[12] Ahmad Hemami and Ferri Hassani. An Overview of Autonomous Loading of Bulk Material. In Carlos H Caldas, William J O'Brien, Seokoho Chi, Jie Gong, and Xiaowei Luo, editors, *Proceedings of the 26th International Symposium on Automation and Robotics in Construction*, pages 405–411, Austin, USA, jun 2009. International Association for Automation and Robotics in Construction (IAARC).

[13] Cat Mining. Cat K Series Large Wheel Loader Operator Training, 2016. Available from https://www.youtube.com/watch?v=a55VzpybLjQ. Accessed: 2024-12-01.

[14] Volvo Construction Equipment. Bucket handling – Volvo Wheel Loaders H-series – Basic operator training, 2019. Available from https://www.youtube.com/watch?v=MAJ_6RKybiE&list=PLJ93Sr2jvwvt5P53gEOkx2zaMOJ31LsMN&index=5. Accessed: 2024-12-01.

[15] Volvo Construction Equipment. Loading/transporting – Volvo Wheel Loaders H-series – Basic operator training, 2019. Available from https://www.youtube.com/watch?v=H7bMgyx8stA&list=PLJ93Sr2jvwvt5P53gEOkx2zaMOJ31LsMN&index=6. Accessed: 2024-12-01.

[16] Sanjiv Singh. and H. Cannon. Multi-resolution planning for earthmoving. In *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*, volume 1, pages 121–126 vol.1, 1998.

[17] Caterpillar. CAT MINING TRUCKS PAYLOAD MANAGEMENT GUIDELINEs, August 2021. Available from https://caterpillar.scene7.com/is/content/Caterpillar/CM20201116-02db9-39c55. Accessed: 2024-12-01.

[18] Paul J. A. Lever. An automated digging control for a wheel loader. *Robotica*, 19(5):497–511, 2001.

[19] Liyang Wang, Zhixian Ye, and Liangjun Zhang. Hierarchical planning for autonomous excavator on material loading tasks. In *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, volume 38, pages 827–834, Dubai, UAE, 2021. IAARC Publications.

[20] David A. Bradley and Derek W. Seward. The development, control and operation of an autonomous robotic excavator. *Journal of Intelligent and Robotic Systems*, 21(1):73–97, jan 1998.

[21] S. P. Singh and R. Narendrula. Factors affecting the productivity of loaders in surface mines. *International Journal of Mining, Reclamation and Environment*, 20(1):20–32, 2006.

[22] Reno Filla and Bobbie Frank. Towards finding the optimal bucket filling strategy through simulation. In *Proceedings of 15: th Scandinavian International Conference on Fluid Power, June 7-9, 2017, Linköping, Sweden*, pages 402–417. Linköping University Electronic Press, 2017.

[23] Siddharth Dadhich, Ulf Bodin, and Ulf Andersson. Key challenges in automation of earth-moving machines. *Automation in Construction*, 68:212–222, 2016.

[24] Andrew A. Dobson, Joshua A. Marshall, and Johan Larsson. Admittance control for robotic loading: Design and experiments with a 1-tonne loader and a 14-tonne load-haul-dump machine. *Journal of Field Robotics*, 34(1):123–150, 2017.

[25] Siddharth Dadhich, Fredrik Sandin, Ulf Bodin, Ulf Andersson, and Torbjörn Martinsson. Field test of neural-network based automatic bucket-filling algorithm for wheel-loaders. *Automation in Construction*, 97:1–12, 2019.

[26] Jirapat Jongluxmanee and Masaki Yamakita. Improved soil shape on the dump truck for soil loading operation of excavator. In *2019 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 845–850, 2019.

[27] Jinwoo Kim, Seokho Chi, and Jongwon Seo. Interaction analysis for vision-based activity identification of earthmoving excavators and dump trucks. *Automation in Construction*, 87:297–308, 2018.

[28] Jeonghwan Kim, Dong eun Lee, and Jongwon Seo. Task planning strategy and path similarity analysis for an autonomous excavator. *Automation in Construction*, 112:103108, 2020.

[29] Anthony Stentz, John Bares, Sanjiv Singh, and Patrick Rowe. A robotic excavator for autonomous truck loading. *Autonomous Robots*, 7:175–186, 1999.

[30] Jungho Yoon, Jeonghwan Kim, Jongwon Seo, and Sangwook Suh. Spatial factors affecting the loading efficiency of excavators. *Automation in Construction*, 48:97–106, 2014.

[31] Jo Yung Wong. *Terramechanics and Off-Road Vehicle Engineering: terrain behaviour, off-road vehicle performance and design*. Elsevier/Butterworth-Heinemann, 2010.

[32] Edward McKyes. *Soil cutting and tillage*. Elsevier, 1985.

[33] Martin Servin, Tomas Berglund, and Samuel Nystedt. A multiscale model of terrain dynamics for real-time earthmoving simulation. *Advanced Modeling and Simulation in Engineering Sciences*, 8(1):1–35, 2021.

[34] Heshan Fernando, Joshua A Marshall, and Johan Larsson. Iterative learning-based admittance control for autonomous excavation. *Journal of Intelligent & Robotic Systems*, 96(3):493–500, 2019.

[35] Claude Lacoursière. *Ghosts and machines: regularized variational methods for interactive simulations of multibodies with dry frictional contacts*. PhD thesis, Umeå University, SE-901 87 Umeå, June 2007.

[36] Algoryx Simulations. AGX Dynamics, 2023. Available from `https://www.algoryx.se/products/agx-dynamics/` Accessed: 2024-12-01.

[37] Martin Servin, Da Wang, Claude Lacoursière, and Kenneth Bodin. Examining the smooth and nonsmooth discrete element approaches to granular matter. *International Journal for Numerical Methods in Engineering*, 97(12):878–902, 2014.

[38] Peter A Cundall and Otto DL Strack. A discrete numerical model for granular assemblies. *Géotechnique*, 29(1):47–65, 1979.

[39] Martin Servin, Da Wang, Claude Lacoursiere, and Kenneth Bodin. Examining the smooth and nonsmooth discrete element approaches to granular matter. *International Journal for Numerical Methods in Engineering*, 97(12):878–902, 2014.

[40] Viktor Wiberg, Martin Servin, and Tomas Nordfjell. Discrete element modelling of large soil deformations under heavy vehicles. *Journal of Terramechanics*, 93:11–21, 2021.

[41] Julian Ibarz, Jie Tan, Chelsea Finn, Mrinal Kalakrishnan, Peter Pastor, and Sergey Levine. How to train your robot with deep reinforcement learning: lessons we have learned. *The International Journal of Robotics Research*, 40(4-5):698–721, 2021.

[42] Wenshuai Zhao, Jorge Peña Queralta, and Tomi Westerlund. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 737–744, 2020.

[43] Christopher G Atkeson and Stefan Schaal. Robot learning from demonstration. In *Proceedings of the Fourteenth International Conference on Machine Learning*, volume 97, pages 12–20, 1997.

[44] Donald J Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In *Proceedings of the 3rd international conference on knowledge discovery and data mining*, pages 359–370. AAAI Press, 1994.

[45] Yutaka Matsuo, Yann LeCun, Maneesh Sahani, Doina Precup, David Silver, Masashi Sugiyama, Eiji Uchibe, and Jun Morimoto. Deep learning, reinforcement learning, and world models. *Neural Networks*, 152:267–275, 2022.

[46] Yuki Saku, Masanori Aizawa, Takeshi Ooi, and Genya Ishigami. Spatiotemporal prediction of soil deformation in bucket excavation using machine learning. *Advanced Robotics*, 35(23):1404–1417, 2021.

[47] Cameron B. Browne, Edward Powley, Daniel Whitehouse, Simon M. Lucas, Peter I. Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(1):1–43, 2012.