

Article

# Automating the Short-Loading Cycle: Survey and Integration Framework

Carl Borngrund <sup>1,\*</sup> , Ulf Bodin <sup>1</sup> , Henrik Andreasson <sup>2</sup>  and Fredrik Sandin <sup>1</sup> 

<sup>1</sup> Department of Computer Science, Electrical and Space Engineering, Luleå University of Technology, 97187 Luleå, Sweden; ulf.bodin@ltu.se (U.B.); fredrik.sandin@ltu.se (F.S.)

<sup>2</sup> School of Science and Technology, Örebro University, Fakultetsgatan 1, 70182 Örebro, Sweden; henrik.andreasson@oru.se

\* Correspondence: carl.borngrund@ltu.se

**Abstract:** The short-loading cycle is a construction task where a wheel loader scoops material from a nearby pile in order to move that material to the tipping body of a dump truck. The short-loading cycle is a vital task performed in high quantities and is often part of a more extensive never-ending process to move material for further refinement. This, together with the highly repetitive nature of the short-loading cycle, makes it a suitable candidate for automation. However, the short-loading cycle is a complex task where the mechanics of the wheel loader together with the interaction between the wheel loader and the environment needs to be considered. This must be achieved while maintaining some productivity goal and, concurrently, minimizing the used energy. The main objective of this work is to analyze the short-loading cycle, assess the current state of research in this field, and discuss the steps required to progress towards a minimal viable product consisting of individual automation solutions that can perform the short-loading cycle well enough to be used by early adopters. This is achieved through a comprehensive literature study and consequent analysis of the review results. From this analysis, the requirements of an MVP are defined and some gaps which are currently hindering the realization of the MVP are presented.

**Keywords:** short-loading cycle; automation; wheel loader; construction; data-driven approaches



**Citation:** Borngrund, C.; Bodin, U.; Andreasson, H.; Sandin, F.

Automating the Short-Loading Cycle: Survey and Integration Framework. *Appl. Sci.* **2024**, *14*, 4674. <https://doi.org/10.3390/app14114674>

Academic Editor: Edyta Plebankiewicz

Received: 18 April 2024

Revised: 16 May 2024

Accepted: 24 May 2024

Published: 29 May 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

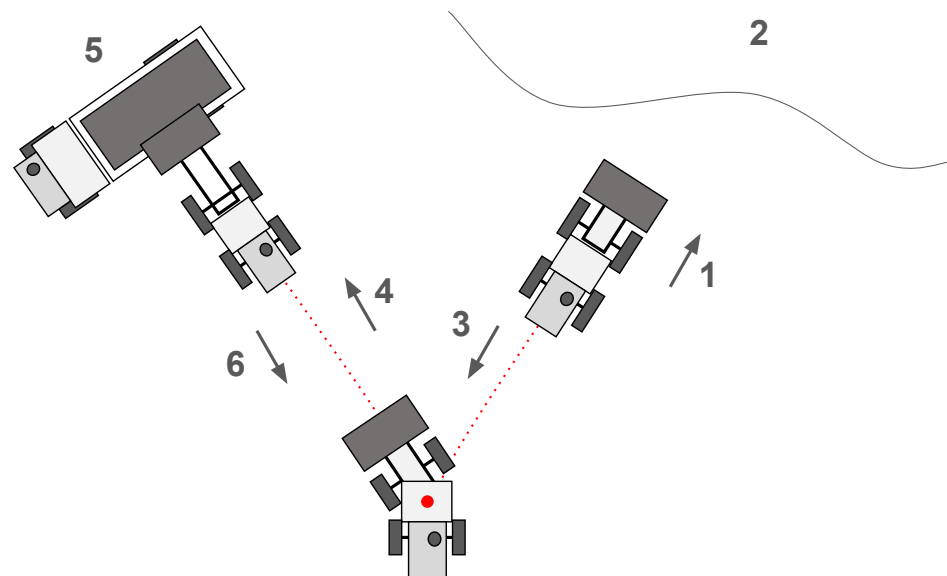
## 1. Introduction

Over the last few decades, automation attempts in the construction industry have increased in frequency. This is because automation of, for example, the used heavy-duty machinery can bring with it a wide set of potential benefits. These benefits include but are not limited to avoiding having humans working in harsh environments [1], more predictable performance and efficiency in operating the machine [2], and redefinition of wheel loader geometry [3].

One of the processes employed in the construction industry is the short-loading cycle where a wheel loader moves material such as rocks, gravel, or dirt from a nearby pile and transports that material to the tipping body of a stationary dump truck. The material is then transported longer distances by the dump truck operator. In the literature, the short-loading cycle is commonly divided into six different steps [4]. These steps include Approaching the pile (1), Loading (2), Retracting from the pile (3), Approaching the dumper (4), Dumping (5), and Retracting from the dumper (6), as seen in Figure 1.

The short-loading cycle is often performed in, for example, an open pit mine, where material continuously flows from a crusher or conveyor belt into a pile, resulting in a never-ending process where there is always material to be scooped. This material is then moved elsewhere for further refinement. Thus, the short-loading cycle is typically performed early on in a larger process, resulting in the need of high performance as any inefficiency has a cascading effect.

The short-loading cycle is a repetitive task performed in high quantities by human operators, making it a great option for automation. The short-loading cycle is also highly representative of the type of work for which wheel loaders will be used for [5]. This includes but is not limited to scooping, dumping, and transporting smaller amounts of materials. The experience from the short-loading cycle can potentially be used while attempting to automate other tasks relevant to wheel loaders, but also similar tasks in other domains.



**Figure 1.** An overview of the short-loading cycle. The leftmost vehicle is a stationary dump truck ready to have material placed in its tipping body. The rightmost vehicle is the wheel loader starting position ready to navigate towards the pile. The numbers indicate the steps of the short-loading cycle as defined in [4]. These steps include the following: (1) Approach the pile, (2) Loading, (3) Retract from the pile, (4) Approach the dumper, (5) Dumping, (6) Retract from the dumper. The red dot shows the reversal point.

Due to the highly repetitive nature of the short-loading cycle, the work can be considered dull for the operators. Operators handle wheel loaders in dirty and dangerous environments that can often be unsuitable for humans, such as underground mines. Automation of wheel loaders could alleviate these negatives while also allowing for enhanced operator engagement where operators can focus on performing more complex tasks. Furthermore, a well-designed driver assistance or automated system can lead to less variance in the efficiency of the cycle [2]. Lastly, automation can allow for a complete transformation of the geometry of the wheel loader, which in turn allows the vehicle easy access deep into, for example, mines and also increases the number of types of environments in which the machine can be used.

Even with decades worth of research, there exists no general autonomous solution that can perform the short-loading cycle task to an acceptable level. The short-loading cycle has been demonstrated in full by a rule-based system [6]; however, very low productivity was demonstrated, where data-driven approaches have successfully addressed some sub-steps such as scooping [7]. Previous standalone surveys have focused on a single type of technology, such as computer vision [8], a single type of problem, such as examining how we can assure safety for autonomous solutions [9], how the construction industry could be automated in general [10], or examining the problem of the lack of adoption of automation in the construction industry [11]. Most of these works have examined the challenges of automation of different types of construction equipment or relevant technology assuming a longer time horizon to realize an autonomous system. In addition, only a handful of survey articles examined the automation of specifically wheel loaders in the short-loading task, and many works focused on a diverse set of construction equipment.

The lack of progress outlined above in the domain of the automation of the short-loading cycle is surprising as the potential benefits of an autonomous solution are high. This serves as motivation to explore the reason behind the slow progress and from this identify a way forward towards a plausible defined minimal viable product (MVP) [12]. An MVP is the idea of creating a product with a minimal amount of features to solve a given problem, allowing for early adopters to provide feedback for further development. In this case, an MVP should be comprised of a set of automation solutions for individual steps in the short-cycle loading operation that are in reach for development and can be combined into a coherent system capable of performing the entire cycle without human intervention.

The main objective of this work is to examine the following question: “From the current status of the automation of the short-loading cycle, what is a reasonably defined MVP and what challenges are currently hindering the realization of an MVP?”, with a secondary question being the following: “Which challenges facing the automation of the short-loading cycle can benefit from the use of deep learning methods?”.

In this work, identification and discussion of a set of automation challenges hindering automating the short-loading cycle are presented. A systematic survey of relevant literature is conducted in terms of the previously identified steps to structure and reason about published solutions for automating the complete cycle. An MVP for automating the short-loading cycle is proposed based on the literature presented in the systematic review. A conceptual automation framework is introduced, leveraging finite-state machines (FSMs) as a model to organize individual automation into a system for experimentation with various solutions for the different steps, contributing to the realization of the proposed MVP. Lastly, the literature synthesis prompts a discussion on future challenges for automating the short-loading cycle.

The main contributions of this paper are as follows:

1. MVP identification based on the currently published literature together with best practices performing the short-loading cycle.
2. An FSM-based high-level framework as a model for combining individual automation solutions into a complete system and facilitating experiments on the automation of the short-loading cycle.
3. Identification of open issues and gaps facing the automation of the short-loading cycle and the realization of a reasonably defined MVP. The identified gaps relate to the abstraction level due to rule-based interfaces, enforcement of safe behavior, the effect of assumptions, and the considerations required when using data-driven and classical control solutions in conjunction.

The remainder of the paper is structured as follows: Section 2 identifies and discusses some of the challenges that are required to overcome to automate the short-loading cycle. Section 3 presents the survey methodology, survey results, and a qualitative analysis of said results. Section 4 proposes a set of requirements for an MVP capable of performing the short-loading cycle autonomously. Section 5 presents a framework aimed to facilitate experiments for the automation of the short-loading cycle. Section 6 discusses some of the future challenges that need to be overcome to realize the defined MVP. Section 7 concludes the paper and presents a clear direction for future research.

## 2. Automation Challenges

Automating construction tasks, such as the short-loading cycle, presents distinct challenges related to vehicle dynamics, environmental interactions, and the unpredictable movements of other agents around the autonomous system. Not only should an autonomous system be capable of comprehending, managing, and responding to these challenges, but it should do so while also achieving a level of performance comparable to that of expert operators regarding productivity, fuel efficiency, and overall effectiveness. The challenges facing the automation of these types of construction tasks include but are not limited to the environmental variables, the mechanics of the wheel loader, long-term dependencies important during the short-loading cycle, and safety challenges.

### 2.1. Environmental Challenges

The types of environments in which the short-loading cycle occurs consist of a series of challenges that complicate the automation process. First, the wheel loader needs to gather material from a pile using a bucket attachment. This interaction holds high significance as an insufficient fill factor would necessitate another round of the cycle to fill the tipping body. Consequently, an unnecessary amount of fuel is used while increasing the fill time of the tipping body, resulting in a worse performance. Modeling the interaction between the bucket and the pile during scooping is challenging due to the large amount of intra-pile forces where each grain of material exerts force on another while also experiencing the force from the bucket [4].

Another challenging interaction to model is the behavior of the wheel loader while steering due to the tyre–terrain interaction. This challenge arises due to the large set of variables that influence the steering mechanics during actuation. These include but are not limited to terrain type, terrain conditions, and tyre thread [13]. The interaction between tyre and terrain is further complicated due to the varying material weight in the bucket, as different load distributions on the wheels affect the overall stability of the wheel loader [14].

Not only are the interactions between the wheel loader and environment complex, but the environment itself is rapidly changing due to operations. For instance, the shape of the pile changes as a result of the scooping process, where the material might fall from the bucket during navigation, and there may be some spillage during dumping. Furthermore, the driving surface, often composed of softer materials, is likely to change over time as the operator repeatedly performs the cycle. To address this, the operator of the wheel loader cleans the work area to simplify operations and ensure that the terrain maintains a good condition to minimize wear on the tyres. This is all achieved while awaiting a new dump truck to load with the material.

### 2.2. Wheel Loader Mechanics

Wheel loaders are complex vehicles where the majority of actions are executed through a hydraulic system. These actions include lifting the boom, tilting the bucket, and changing the steering angle, which are all actions that are directly coupled together. The dynamics of the hydraulic system within a wheel loader are stiff and highly nonlinear, posing challenges in developing precise mathematical simulation models [15]. Due to the stiffness of the hydraulics, there is a significant time delay between when the operator initiates an action, such as the lift, and when the boom begins to move. This time delay is a consequence of the way in which hydraulic systems work, where fluid has to move to create high enough pressure in the cylinders, resulting in the boom moving. The combination of action coupling, time delays, and nonlinearity of actions contribute to the complex nature of automating the short-loading cycle.

Wear and tear on the machine parts such as tyre wear pose additional complications for automation, as even regular wear can affect the amount of propulsion the machine can generate. Wear on the tyres relates to cleaning the work area. If the work area is left uncleaned, there is a greater likelihood that the tyre will wear faster or, in the worst case, be damaged to such an extent to necessitates costly replacements [16]. This means that any autonomous solution has to be capable of adapting to changes in the mechanics of the wheel loader due to wear.

In addition, the choices that operators make during operation can affect the amount of maintenance required for the vehicle due to wear. For example, a lower-skilled operator might incorrectly leverage the lift and tilt functionality during the scooping of material which leads to higher wheel spin that wears down the tyres. Changing the tyres is a large expense [17] and will take the wheel loader out of effective work for some time, halting the process.

### 2.3. Long-Term Dependencies

The short-loading cycle encompasses numerous long-term dependencies crucial for the optimization of the work cycle. As the short-loading cycle is typically performed early in a larger process, such as ore refinement, any inefficiencies cascade downwards affecting the downstream process. Because of this, during the cycle, operators attempt to maintain a given productivity goal while minimizing the environmental impact. The operator's behavior can have a massive impact on both productivity and the environmental impact, with a theoretical fuel efficiency increase of 200% and a theoretical productivity increase of 700% [18]. Even for experienced drivers, there can be a large difference in productivity and fuel efficiency, where it can be difficult to determine whether the performance difference is due to skill or minimization of personal workload [19].

To fill the dump truck's tipping body, multiple rounds of the short-loading cycle are necessary. To perform this task correctly, the operator needs to understand how the material disperses within the tipping body during the dumping process. This can be achieved by dumping in the front, middle, and back of the tipping body which, if performed correctly, ensures even loading of material in the tipping body, facilitating stable navigation. This means that information gathered from the first round of the cycle contains information necessary to optimize performance of the cycle in subsequent rounds.

Lastly, the scooping and dumping position chosen by the operator during the cycle affects the position of the reversal point. The operator must select the reversal point to maximize productivity while minimizing the environmental impact.

### 2.4. Safety Concerns

When attempting to automate these large vehicles, we must pay close attention to the relevant safety challenges. These safety challenges encompass various aspects, including human-and-machine interaction, emergency handling, collision avoidance, and safety guarantees for certification.

Heavy-duty construction machinery, like wheel loaders, operate in close vicinity of both other vehicles with operators and other workers. Any autonomous system must be not only capable but has assurances to prevent damage to vehicles and injuries to humans. This is essentially emergency handling and collision avoidance. However, it is crucial to emphasize the significance of safety guarantees for regulatory bodies to certify the solution's widespread deployment.

This means that any deployable autonomous system must be provably correct, which can be challenging due to the technology solutions used within the autonomous system.

For example, when leveraging data-driven approaches, it can be difficult to reflect on how a system will behave in different situations. Currently, incorrect behavior is often part of the learning process leading to embedding the decision-making of safe behavior into the policy. The consequence of this is that there is no guarantee of safe behavior, nor is it possible to prove the solution safe, which leads to difficulties in certifying a solution for use.

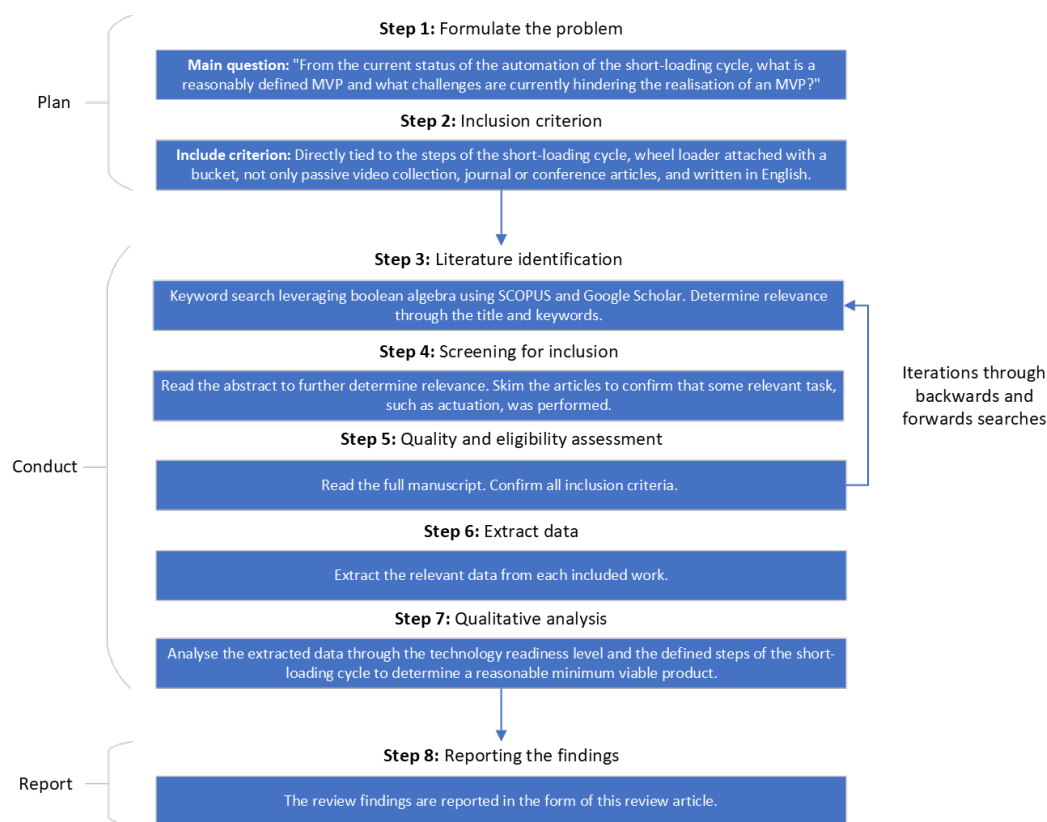
## 3. Automation of Wheel Loaders

To define a plausible MVP for the automation of the short-loading cycle, together with future challenges in improving said MVP, the currently published literature in this domain was reviewed. This section aims to present the methodology, the results, and a qualitative analysis of the review.

### 3.1. Review Methodology

The methodology used for this systematic literature review is based on the process presented in [20], where the review is divided into 8 different steps, in three overarching categories. The methodology is visualized in Figure 2. The first category of planning the review includes Steps 1 and 2, where the problem is formulated and a review protocol is developed. In this work, the background for the problem formulation is presented in Section 1, resulting in the main question: "From the current status of the automation of the

short-loading cycle, what is a reasonably defined MVP and what challenges are currently hindering the realization of an MVP?”. The last step reporting the findings (Step 8) is performed in the form of this review article.



**Figure 2.** An eight-step review methodology used to perform the survey. The figure is based on the review methodology figure presented in [20].

Hence, this section and the following sections aim to cover Steps 2 to 7. This is structured in the following way: Section 3.1.1 presents the inclusion criterion used in the survey (Step 2). Section 3.1.2 presents how the initial literature identification was performed (Step 3). Section 3.1.3 presents how the identified literature was screened for potential inclusion in the survey (Step 4). Section 3.1.4 presents how the literature was screened for quality and final eligibility in the survey (Step 5). Section 3.2 presents the extracted data from each of the included works (Step 6). Lastly, Section 3.3 presents the qualitative analysis of the extracted data (Step 7).

This study aims to investigate the state of the art directly related to operations to assess the plausible possibility of defining an MVP capable of performing the short-loading cycle autonomously. This makes works that are not directly tied to the automation of the short-loading cycle relatively diminished in relevance, which is reflected in the inclusion criterion and search terms defined below.

### 3.1.1. Inclusion Criterion

Only the literature that is directly tied to the automation of single or multiple steps of the short-loading cycle, as defined in Figure 1, was included. This encompassed both rule-based and data-driven approaches. The literature that exclusively examined passive vision-based information extraction was excluded, for example, finding the bounding box of different parts of the dump truck from a camera [21]. However, studies that used vision to perform some tasks such as finding the scooping point and then navigating to it were still included.

If a set of continuous articles were identified, the most complete version was included.



In this work, the short-loading cycle is assumed to be performed by a wheel loader with the bucket, meaning that other types of vehicles were not included in the survey. This is because the structure and mechanics of the wheel loader are quite unique compared to other vehicles used in the construction industry. Because of this, it can be difficult to transfer the knowledge from, for example, an excavator to a wheel loader.

Lastly, only optimal control works that were directly relevant to performing the cycle were included in the survey. This means studies that, for example, compared different types of vehicles were outside the scope of this work.

These inclusion criteria will hopefully result in a literature synthesis that provides a good reflection of the current status of the automation of the short-loading cycle and offers a clear direction for future work.

### 3.1.2. Literature Identification

To identify the literature relevant to this work, a set of general domain search terms were defined, such as “Automation of Wheel loaders” or “Automation of the short-loading cycle”. These results were used to collect a set of relevant keywords that could be used for literature identification. The keywords were divided into three categories: (1) Hardware-specific, (2) Task-specific, and (3) Subtasks. The hardware-specific keywords are used to capture the type of vehicle and machine that is being used within the task, such as wheel loaders or construction equipment. The task-specific keywords capture the task of the short-loading cycle and its set of synonyms that have been used historically within this domain. Lastly, the sub-task-specific keywords try to encapsulate all the sub-tasks that we are interested in such as automating the short-loading cycle, performing navigation, or carrying out pathfinding.

To capture the relevant literature as effectively as possible, Boolean algebra was leveraged, supported by many search engines, and a search query was built that fully includes all three categories defined above. From these results, the title was used to gauge the relevance of the paper. If the paper appeared relevant from the title and was written in English, the full reference was included for further screening. In this work, Scopus was used for the literature identification and to refine the keywords using Scopus filtering functionality. In addition, Google Scholar was used for literature identification, and forward and backward searches were utilized to decrease the chance of important literature being missed. Table 1 shows the set of keywords and synonyms used to identify the literature in this work.

This search resulted in 89 articles for further screening.

**Table 1.** The search terms used while performing literature identification.

Category	Search Terms
Hardware synonyms	Wheel loader, Construction equipment/machinery, Loader, Mining machinery, Hydraulic machinery/equipment, Earthmoving machinery
Task synonyms	Short-loading cycle, Y cycle, V cycle, Short-cycle loading, Loading cycles, Y-shaped, Y path, V-shaped, Y path
Subtask	Autonomous, Automation, Controllers, Robots, Robotics, Plan, Planning, Detection, Follow, Following, Scoop, Scooping, Approach, Approaching, Reverse, Reversing, Dump, Dumping, Fill, Filling, Navigation, Shovel, Shoveling, Bucket-filling, Path

### 3.1.3. Screening for Inclusion

Screening the identified set of articles for preliminary inclusion was achieved by reading the abstract. From the abstract, the reviewer deemed whether the article was relevant enough for full-text review or not. When reading the abstract, the keywords defined in Table 1 were verified to at least be partially present. Furthermore, the work was

verified to include some type of actuation control or real-world data. Uncertainties were presented and discussed to ensure minimal biases.

This resulted in 65 articles for continued assessment.

### 3.1.4. Quality and Eligibility Assessment

As mentioned, to assess the quality of each identified article, the full text was skimmed. Peer-reviewed journal articles together with peer-reviewed conference articles were deemed to have the highest quality of the identified articles. However, some non-peer-reviewed journal and conference articles were included if they had high relevancy and the work seemed to be of high quality. Books, online presentations, and patents were excluded.

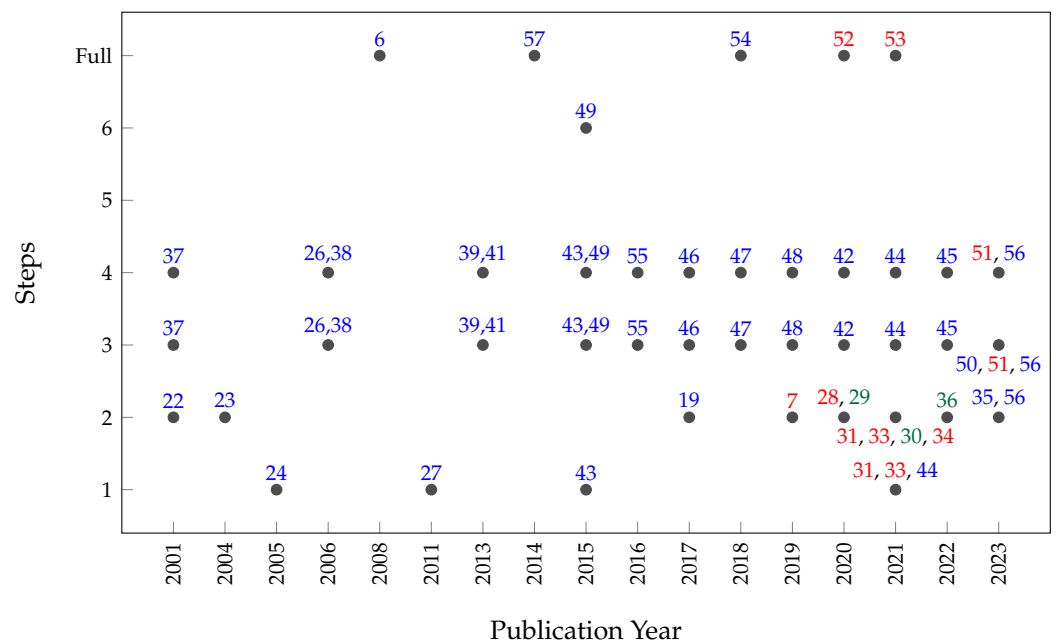
From this last screening, a total of 37 articles were fully read, resulting in 17 being included in this work. During the full manuscript assessment, all inclusion criteria were verified, with real-world data or actuation control being essential. This included pure simulation work showing some tasks being performed, such as scooping. If no task was performed, such as in the case of performance analysis, there was a requirement for it to be based on real-world data.

### 3.1.5. Iterations

To minimize the risk of important literature not being captured within the search due to keyword choices, both a backward and a forward search were performed on the included literature. The articles identified from the forward and backward search were also included or excluded based on the same procedure as all other literature. The forward and backward search produced articles that were screened for inclusion. Together with the 17 from the previous section, 36 articles were included in the survey.

## 3.2. Survey

As previously mentioned, the short-loading cycle can be divided into six different steps. These six steps have received different levels of investigation over time. For example, as shown in Table 2 and Figure 3, 12 of the survey works focused on scooping, 15 focused on navigation, and 0 fully focused on the dumping step.



**Figure 3.** The steps that each surveyed work examined. If one work examines multiple steps, there are multiple entries in the graph. The individual numbers coincide with a single citation. The bluemarked citations are rule-based, the red citations are data-driven, and the green citations use a mix of both.



The survey is split into two parts. The first investigates what has been performed in terms of the automation of the short-loading cycle by examining the bucket-filling, the navigation, and the full cycle. This is achieved by including both data-driven and rule-based approaches to investigate what steps have been examined using the two types of approaches and whether one type of approach outperforms the other. In this case, data-driven approaches encompass algorithms where data can be automatically leveraged to create solutions, whereas rule-based solutions are built through direct observation by the developer. In addition, investigations regarding the performance of human operators are presented to obtain an understanding of what high performance is.

The second part of the survey is a qualitative analysis of the works to obtain an overview of the types of approaches used for each step in the short-loading cycle and the potential of using said approaches in an MVP capable of completing the task of the short-loading cycle.

A summary of all the surveyed works, together with the addressed steps, can be found in Table 2. The advantages and disadvantages of each work come from the perspective of how these solutions or how knowledge from these solutions can be used when creating a solution that correctly performs the short-loading cycle. Furthermore, the advantages and disadvantages also try to encompass the importance of some type of optimization towards productivity and energy efficiency.

### 3.2.1. Bucket-Filling

Bucket-filling, or scooping, has seen the most research activity when it comes to the automation of the short-loading cycle. One explanation for this is that during the bucket-filling phase, it is possible to receive direct feedback from the interactions between material and bucket. Furthermore, the phase is short and also complex. Lastly, modeling the pile is difficult because of the internal forces acting on each other when there is interaction with the pile [4].

Rule-based solutions attempt to mimic observed human operator behavior through handwritten rules. One of these behaviors is lifting the bucket from the ground to the exit height while simultaneously tilting the bucket [19]. This was found to be the optimal strategy when examining bucket-filling in simulations. However, this could unfortunately not be verified to be the optimal bucket-filling strategy for expert operators on real machines. Tentatively, the authors still concluded that this is the best strategy for human operators when filling the bucket.

Fuzzy logic was used to formulate another observed operator behavior programmatically to perform the scooping of material [22] together with an FSM that executed the programmed behavior, successfully matching operator performance in terms of energy and payload. This seems to indicate that there is not one unique strategy that will result in meeting the recommended fill factor after scooping on real machines.

Following this, a three-step algorithm was proposed based on the three observed bucket-moving phases during scooping [23]. It is concerned with the way the bucket moves inside the pile throughout the bucket-filling process. This solution was tested on a miniature wheel loader and the fill factor was estimated through stereo vision and a column model of the pile, but it was not compared to real operators.

Multiple early attempts were made to generate the path towards the scooping point using different algorithms such as a mixture of clothoid curves and straight lines [24], where a clothoid is a curve whose curvature is proportional to the length of the path [25]. This, together with the column model and the stereo vision setup to detect the pile, was used to scoop material from a pile. The authors used a method similar to that which has been used to generate the V path between the scooping point and the dump truck [26]. Not comparing the performance of the autonomous system to expert operators or the recommended fill factor for the machine is problematic due to the importance of matching the performance of expert operators in the short-loading cycle.

As previously mentioned, the first step of scooping material from a pile is to determine a spot from where to scoop. Some attempts have been made to create a system that leverages Light Detection and Ranging (LIDAR) to find the optimal scooping point [27]. Here, four algorithms were tested. These include column-model estimates, the two-part bucket model, the three-part bucket model, and quadric fitting. The authors concluded that the quadratic fitting was the best-performing solution for determining a scooping position. However, the authors found that it seems to be difficult to create a dataset or determine the ground truth of the best scooping position. The authors found that when operators of the same skill level were presented with a pile, they offered multiple different answers for the optimal scooping point, making it difficult to determine a true optimal scooping point for each pile shape.

In later years, deep learning solutions were applied to the problem of autonomous bucket-filling since modeling the internal forces within a pile during the bucket-filling phase is difficult, for example, when teaching a time-delayed neural network to perform regression by classification and using that solution to autonomously scoop material using a real wheel loader [7]. This time-delayed neural network was trained on 100 bucket-filling examples by an expert operator and could match the operator in terms of the fill factor. However, the solution could still not beat the operator in terms of bucket-filling time. This time-delayed neural network was later adapted to learn how to scoop another type of material through the use of reinforcement learning [28]. Within 40 bucket-filling trials, the bucket-filling solution improved the productivity by 5–10% over directly deploying the old network.

A simple random forest controller has successfully performed bucket-filling in summer conditions, outperforming a neural network controller in the same environment [29], trained on data collected from non-expert operators and using both low-level signals and vision-based sensors as input. However, when the complexity of the environment increased such as when the temperature decreased, a random forest controller started failing, hence requiring a more complex controller such as that that adapting the attention mechanism [30]. Furthermore, adding pressure sensors as input to the controller also increased the performance of the controller. The results presented in the above two works indicate that data-driven approaches are capable of effectively operating in more complex environments under challenging conditions.

Both of these solutions were tested on real machines in realistic environments, showing good performance. Furthermore, the former could match the expert operator in terms of fill factor, which shows that this type of approach has the potential to result in a higher-performing solution.

Lastly, reinforcement learning (RL) has been used to attempt to perform bucket-filling. RL has the benefit of allowing us training of an agent to perform the bucket-filling without having to collect data from a real operator, as this can be expensive. Furthermore, in the future, there might not exist human data to train from, hence RL might be the only way to train an agent.

A deep RL controller has been trained in simulation to perform the approach towards the pile and scooping of material [31]. The controller was trained using the actor–critic DPPG algorithm [32], purely by exploring the environment. Both the actor and the critic had three hidden layers, mapping a 12 d vector containing sensor data to the wheel loader’s tilt, lift, and velocity. The controller was transferred to a miniature wheel loader and tested with no additional training. The solution showed good performance compared to the simulation in terms of the weight of the material within the bucket and the bucket-filling time.

Two sequential deep learning agents learned how to perform the approach and scooping of material in a high-fidelity 3D simulation [33]. The two sequential agents were a mucking position agent which determined the scooping point from camera input and a mucking agent which approached and scooped from the determined scooping point. The mucking agent used a variety of sensors such as position, velocity, force, and LIDAR to

navigate towards the scooping point and scoop. This type of solution achieved an average of 75% of the max loading capacity.

Q-learning has also been applied to the bucket-filling problem. This was achieved by removing the physical model of the wheel loader and instead using a statistical model predicting the state of the wheel loader at some time step [34]. From this model, an agent was trained to perform the bucket-filling process. The trained agent showed good performance in terms of fuel consumption, adaptability, and convergence even with the absence of a physical model. Furthermore, the trained agent could, with transfer learning, be adapted to other types of material.

Many of the solutions presented above, rule-based and data-driven alike, relied on the reaction forces between bucket and pile, or pressure in the hydraulics, as input to the system. This seems like a reasonable input choice for an autonomous bucket-filling system, even if it does not translate to what human operators use generally, as with this information autonomous systems can outperform human operators at least in terms of fuel efficiency [35]. The optimal depth that the bucket should be pushed into in the pile while preserving the fill factor was around 400 mm [36], which decreased the operational time by 15% and fuel consumption by 5%.

In terms of bucket-filling, there exists a large, varied solution pool consisting of different approaches using a wide set of different sensor setups. This gives us a good idea of what is the highest-performing solution and what type of solution could be a part of an MVP in the domain of the short-loading cycle. From the literature synthesis above, it appears that for scooping, deep learning-based approaches outperform rule-based systems in terms of matching expert operator performance. This seems to lead to using some deep learning-based system for scooping in the MVP, and currently, the network should probably be trained through collected human data and adapted to new material using deep reinforcement learning.

### 3.2.2. Navigation

When it comes to navigation between the pile and the tipping body, very little has been achieved in terms of examining how data-driven approaches can be used for navigation; however, quite a lot of rule-based solutions have been proposed. Very few of these solutions have been demonstrated on a real wheel loader in a realistic environment.

Clothoid-based path generation and tracking have been used not only to navigate towards the pile but also to navigate between the pile and the dump truck [37]. The generated path consists of two symmetrical clothoids connected at the reversal point, where the first clothoid is tracked during reversing and the second is tracked during forward motion. Due to wheel loaders being a type of actuated vehicle, the tracking point on the wheel loader has to change throughout navigation. While reversing, the tracking point is the centre of the back axle and while moving forward, the tracking point is the centre of the front axle.

This clothoid solution is extended to include a larger amount of types of path generation situations, including the generating a path between pile and dump truck [38]. A canonical path skeleton is defined between a set of points costing three line segments and two clothoids, which is then optimized using two different methods. The first method is a genetic algorithm and the second is a form of quadratic programming. The methods search through the variables and parameters for a set amount of time and once finished, the semi-optimal path is retrieved, where it is optimally defined as the length of the path. A similar optimization algorithm is based on a genetic algorithm, where the path is assumed to include a straight line, a clothoid curve, and a circle line [39]. The optimization criterion is the path length, and the genetic algorithm runs for 1000 iterations before returning the semi-optimal path. Both of these solutions are validated on a miniature wheel loader, showing that both solutions have potential.

An extended version of the Reeds–Shepp [40] algorithm has been used to generate the trajectory between the pile and the tipping body [41]. The generated optimal path, where

the optimality criteria are based on distance and cusp minimization, consists of circular arcs and straight segments. To perform the path following the generated trajectory, a PID is used to control the lateral position with constant velocity. Ref. [42] used an algorithm based on RRT\* and CC steer to plan the trajectory between the pile and the tipping body. The generated trajectory was tracked using an AMPC tracking system that considers the curvature disturbances in the trajectory resulting from the wheel loader's acceleration and deceleration. Neither of these solutions considered the lift speed of the boom in the proposed solutions.

Most of the previous work has examined cases where a trajectory needs to be generated between a single loading and unloading point. However, the short-loading cycle consists of multiple scooping and unloading points as it takes more than a single bucket to fill the tipping body of a dump truck. Ref. [43] proposed an algorithm that calculates multiple scooping points for a single unload point and tested it in a simulation. This was achieved by creating a score space containing the length of each path in the global space and generating, from this score space, all paths using CC paths. The solution was validated on a miniature wheel loader.

Furthermore, ref. [44] proposed another solution to generate paths between multiple arbitrary loading points and a single unloading point. This was achieved by decomposing the problem into three distinct tasks. The first task revolves around finding a set of candidate scooping points from a cost map. The second task ranks these points using reinforcement learning to obtain the actual set of scooping points. From this, the last task uses CC paths to generate the continuous path to excavate the gravel pile and transfer the material to the tipping body of the dump truck.

Lastly, optimal control has been used to perform path generation for both pile-to-truck and truck-to-pile [45]. The authors of the aforementioned reference tested a set of different controllers to control the articulation angle (steering) in a 3D simulation and found that the LVP-MPC controller performed the best. Due to a high time complexity, the path had to be generated offline, while tracking could be performed online. To track the generated paths, in addition to the articulation controller, a Stanley controller for acceleration, deceleration, and shifting was used.

Multiple different techniques have been applied to examine the optimal trajectory during the short-loading cycle. Based on 2D space discretization, ref. [46] formulated the path planning problem as an optimal control problem, where optimality was defined by fuel efficiency and environmental impact. To find the optimal path, a discrete dynamic programming approach was used. The best performing was the free path, where no working area limitation existed, and adding an obstacle increased the fuel usage; however, compared to measured data, the optimal path reduced the fuel consumption by 14%. Furthermore, this solution was later extended to optimize a path within a known construction environment while also considering the terrain information [47].

From an a priori optimal path, the optimal tracking and switching time from reversing to forward was examined [48]. The wheel loader was modeled using three modes: reversing, forward, and stopping. Adaptive dynamic programming was used to solve the optimal control problem where a neural network was trained to approximate the optimal cost.

A driver-in-the-loop model was proposed to examine trajectory traversal optimization in terms of energy flow and energy loss during the short-loading cycle [49]. The analyzed human behavior was embedded into the driving model using a linear quadratic regulator and model predictive control. By applying the driving model to three predefined paths, the authors examined where losses and gains could be found. For example, a longer path increased the travelled length but decreased the steering actuation amount.

Therefore, it seems that very few of the mentioned solutions considered the lifting speed of the boom, which is an important variable due to the coupled nature of the wheel loaders. The boom speed dictates the absolute minimum time the navigation can take as the boom has to be raised high enough to make sure the bucket clears the rim of the tipping body before dumping. When examining energy optimization for navigation during the short-loading cycle, boom speed and correct steering movements were found to be essential [50].

Reinforcement learning has been applied to the problem of navigation [51]. The authors used two agents: one that performed the reversal and another that performed the approach towards the dump truck. The agents controlled the vehicle through lift, velocity, and steering. The task was possible for two agents to learn, whereas a single agent trained on the same reward functions failed to learn. The results indicate that task decomposition can facilitate the automation of the short-loading cycle, especially when attempting to leverage deep learning techniques.

It is not yet clear what approach is superior due to a small amount of research using data-driven approaches and the lack of validation on real vehicles. However, what is clear is that validating on either miniature or real wheel loaders appears to accelerate the development and is something that should be leveraged for the development of an MVP.

### 3.2.3. The Full Cycle

In terms of the attempts to automate the full short-loading cycle, only a handful of solutions have been proposed. Furthermore, only a single solution demonstrated the full cycle on a real machine, whereas the majority of works only examined the theoretical benefits of this type of automation.

The demonstrated solution was based on three different subsystems: measuring and modeling the environment, task planning, and motion control [6]. This solution was capable of detecting the pile, navigating towards the pile, scooping, navigating to the tipping body, and dumping material in the tipping body. This was performed in a semi-realistic environment; however, it was at low productivity, probably due to safety concerns of controlling such a large vehicle autonomously and not pushing the limits of the solution. All of the subsystems were first tested on miniature wheel loaders, further showing that the development pipeline of transferring the solution from miniature wheel loaders to real wheel loaders can be successful.

In terms of data-driven approaches, long-short-term memory (LSTM) has been used to predict the brake pedal aperture [52]. The model was trained on real data from an expert operator and the model could successfully predict the brake pedal aperture. Furthermore, a set of sequential LSTMs has been used to predict the throttle and state variables of the wheel loader throughout the short-loading cycle [53]. The state variable includes tilt pressure, lift pressure, engine speed, and velocity of the wheel loader.

The theoretical benefits of automation of the short-loading cycle are, however, substantial, such as a 14% increased fuel efficiency compared to real operators [54]. This makes the pursuit of an autonomous solution worthwhile. Some of the behaviors required for optimality can be difficult for human operators to maintain during long periods. For example, one optimal control formulation showed that it is optimal to perform very fast steering changes throughout the short-loading cycle [55], something that is difficult to maintain over a long time. Furthermore, a multistage optimization problem has been defined where both the transport and the bucket-loading operation were included [56]. The control solution showed a fuel usage decrease of 42.1% over human operators.

The theoretical benefits of automation depend not only on the operator choices but also on external variables. This includes the positioning of the dump truck, where a 20-degree change in the orientation of the dump truck increases the fuel consumption during the short-loading cycle by 18% [57]. In addition, if the dump truck is positioned optimally, the trajectory for both time and fuel efficiency is identical.

From these attempts at automation of different parts of the short-loading cycle, it is unclear which is the best approach; however, it is clear that there is a requirement of some type of structured pipeline for development. This includes either decomposing the problem into smaller parts having different systems be in charge of different functionality or having some platform that allows for fast testing. This includes having, for example, an LSTM be responsible for the braking aperture while other systems are responsible for the other actions, or creating a development platform such as a miniature wheel loader.



**Table 2.** Summary, advantages, and disadvantages of the surveyed articles. Each article is categorized based on the steps of the short-loading cycle. The steps are the following: (1) Approach the pile, (2) Loading, (3) Retract from the pile, (4) Approach the dumper, (5) Dumping, (6) Retract from the dumper [4].

Publ.	Steps	Summary	Advantages	Disadvantages
[24]	1	Path generation towards scooping point. Clothoid-based. Determining the scooping direction.	High explainability due to being a rule-based solution. The solution is evaluated on a real miniature wheel loader.	The solution makes a set of assumptions for the generated path that might not hold in all situations. Does not discuss the fill factor or similar metrics.
[27]	1	Selecting attack point. Achieved by estimating convexity and sload of the pile. Point cloud data. Outperforms the previously published techniques.	Identifies good attack poses from both simulated and real data. The proposed method does not require creating an elevation map, leading to good time complexity.	Difficulties in finding the ground truth for the optimal attack pose due to the complexity of the task. The solution is only tested on a single gravel pile.
[31]	1, 2	Deep deterministic policy gradient [32]. Approach and scooping. Simulation-to-scale-model transfer. 65% fill factor.	Simulation-to-reality transfer with comparable performance shows that this type of pipeline has potential. The average cycle time is quite low.	Low fill factor compared to the usually desired fill factor (100–110% depending on material).
[33]	1, 2	Soft Actor–Critic [58]. Teaching one agent to find the scooping point. The second agent performs approach and scooping. Trained in a high-fidelity simulation.	Shows the potential of a purely data-driven solution in this use case. Embedded energy usage in the reward function.	No validation towards real data or real vehicles. Low fill factor compared to the desired fill factor during operation.
[22]	2	Automated digging. Fuzzy logic for behavior formulation. Finite-state machine to create the formulated behavior.	The solution uses only production sensors. Tested on multiple different pile materials. Good performance compared to expert operators in terms of energy/payload.	Worse performance than operators in terms of the most important metric—productivity. A faster pulling rate might have allowed for even closer performance compared to operators.
[23]	2	Automated digging. Based on acting bucket forces. A three-step algorithm. Stereo vision for scooped volume estimation.	Tested on a different set of pile slopes. Validated on a scale model. High explainability due to the white-box nature of the proposed solution.	Fill factor or similar performance not reported. Pile modeling is difficult due to the intra-pile forces. Only tested on a single material.
[19]	2	Examined different bucket-filling strategies using DEM simulation. A wide set of different trajectory types is examined. Qualitatively “slicing cheese” is the top-performing strategy in both simulation and real-world applications.	The optimal control is compared to real-world expert operators for validation. Provides insight into the specifics of how operators perform bucket-filling.	Issues in measuring setup make it difficult to conclude the best bucket-filling strategy. Not all type-E trajectories can be tested due to computing limitations. Only gravel was considered as the material.
[7]	2	Autonomous scooping. Uses a time-delayed neural network. Trained using imitation learning on 96 examples. Scooped from a pile containing medium coarse gravel. Matches human performance.	Matches performance in terms of weight in the bucket while only having a 26% longer fill time compared to teleoperated. Trained offline and tested on real machines.	Only trained on a single type of material. The throttle is to a constant value, meaning the network cannot optimize this variable.
[28]	2	Adapts a previous machine-learning model [7]. From medium coarse gravel to cobble gravel. Deterministic actor–critic reinforcement learning algorithm. Tested on a real wheel loader.	The proposed solution is tested on a real vehicle. Requires a low amount of trials to adapt the solution to new material. Not learning from human operators allows for the model to learn tasks where human data might not be available.	No test of whether the model retains the knowledge of shovelling medium coarse gravel after adaptation. Trained online, leading to high safety risk as training of an RL model requires exploration. The productivity-based reward function fails to improve productivity.



Table 2. Cont.

Publ.	Steps	Summary	Advantages	Disadvantages
[29]	2	Compares a neural network and random forest for autonomous scooping. Trained using human demonstrations. Data was collected during the summer months. Random forest using low-level signals and vision performs the best.	A simple RF controller successfully learns and performs scooping from a pile. Tested over multiple weather conditions, daytime, and two types of pile material.	Uncertainty of the skill of the operator whose demonstrations are used. The performance is reported in terms of success classes rather than the fill factor.
[30]	2	Based on previous work [29]. Examines how the control would perform during winter months. The simple RF controller that functions during the summer months struggles during the winter months.	Shows that the difficulty differences due to seasons can impact what type of controller performs the best. The new proposed controller has higher robustness to changing conditions and a superior success rate compared to the author's previous work.	Same as above.
[34]	2	Q-learning to perform bucket-filling. The model also covers the working state of the wheel loader. Outperforms human operators in terms of fuel consumption. The trained model can be transferred to other materials.	Due to being based on a predictive model, there is no need for direct interaction with the real environment during training. Improves the fuel efficiency compared to the operator data.	Not tested on an actual vehicle; however, it is based on real data. Because it is not being tested on an actual vehicle, it is difficult to say how well the predictive model performs compared to a dynamical model or online training.
[36]	2	Develops a test platform to test the operational performance of wheel loaders. Tests and analyzes 9 different bucket trajectories during bucket-filling. The optimal shovel depth is found to be 400 mm.	Data and model based on a real wheel loader. The model shows that the optimal scooping depth is 400 mm, outperforming human operators in terms of fuel usage and operational time.	Total energy consumption is higher for the given solution compared to the human operator. Unclear whether an autonomous solution can have mm preciseness when scooping in a real situation.
[35]	2	Formulates the bucket filling as a control problem. Proposes a bucket-filling strategy using optimal control in simulation.	The proposed method reduces fuel consumption by around 30%. The model is validated using a real wheel loader. The proposed algorithm is compared to real drivers. The proposed algorithm can be used online.	The skill level of the human drivers is unclear. Unclear whether the proposed algorithm is tested on multiple different types of material.
[50]	3	Examines the path planning from the scooping point to the reversal point. Formulates the problem as a control of a switching system at some time. Uses approximate dynamic programming to solve this formulation. Finds a near-optimal solution.	Compared to other work, the authors consider the lifting action as it is vital for the efficiency of the cycle. The proposed closed-loop solution is insensitive to different initial conditions.	The solution is not tested on a real wheel loader or compared to real operator data. The model does not seem to be validated towards a real wheel loader, making the results very dependent on the model used.
[37]	3, 4	Clothoid-based path generation and path tracking. Tested on a miniature wheel loader. The tracking point is changed during navigation, the rear axle during reversing, and the front axle during forward motion.	Highly explainable due to the low randomness in the solution. The solution is validated on a miniature wheel loader.	Because it is a clothoid-based solution, assumptions might not hold in every situation. The lift is not considered in this work, where the lift speed is one of the important factors determining the path.

Table 2. Cont.

Publ.	Steps	Summary	Advantages	Disadvantages
[38]	3, 4	Presents a semi-optimal path generation scheme. Clothoid-based solution. Optimizes a path skeleton using genetic optimization and quadratic programming optimization. Optimality is defined as the wheel loader's moving distance.	Still a clothoid-based solution; however, there is some type of optimization from the generated skeleton path. Tests a wide set of different setups, not only paths during the short-loading cycle.	No real-world tests. Optimality for the optimization is defined as the total moving distance rather than productivity or energy efficiency. The lift is not considered in this work, where the lift speed is one of the important factors determining the path.
[26]	3, 4	Models the pile as a set of columns. Approximates all the forces that would act on the bucket and uses that approximation to decide the scooping direction. Plans a V-shaped trajectory consisting of two symmetrical clothoid and three line segments. Minimizes the V-shaped trajectory in terms of length. Tested on a miniature wheel loader.	Considers path generation together with a scooping point detection procedure. The solution is tested on a miniature wheel loader.	Modeling the pile is difficult because of the intra-pile forces, meaning that the column model might not result in the optimal scooping point. Clothoid-based solution without any type of optimization. The lift is not considered in this work, where the lift speed is one of the important factors determining the path.
[39]	3, 4	Performed path planning between the scooping point and the unloading point. Uses GA to optimize the path by having each chromosome include the wheel loader's motion in one term. Optimizes the formulation over 1000 generations. Tested on a miniature wheel loader.	Optimizes the path using GA resulting in a shorter path compared to the typical path generation strategies. Quasi-optimization as finding the true optimum is too time-consuming, leading to a reasonable execution time. The solution is tested on a miniature wheel loader.	Distance optimization might not offer the best result in terms of important metrics such as productivity and energy efficiency. The lift is not considered in this work, where the lift speed is one of the important factors determining the path.
[41]	3, 4	Extended Redd and Shepps algorithm [40] to perform path planning during the short-loading cycle. Uses a PID to follow the generated path, while the vehicle velocity is constant. The algorithm finishes when the minimum path is found.	The given solution does not require the path to be symmetrical as that assumption does not always hold depending on the scooping and unloading points. Optimizes the path in terms of distance while still validating all the constraints.	The lift is not considered in this work, where the lift speed is one of the important factors determining the path. Optimizing distance might not lead to the most efficient path in terms of productivity or energy efficiency. The solution is only tested in simulation.
[55]	3, 4	Compares data from different skilled operators with an improved optimal control formulation. Examines the effect different variables have on total productivity. The optimal control solution shows very fast steering inputs, difficult for humans to perform for long periods.	The wheel loader model is validated against real-world data, showing that the model could estimate fuel consumption closely. Functionality regarding the lifting operation is included in the model. The results show the potential of autonomous solutions.	No attempt to test the proposed optimal control on a real vehicle using some set of controllers.
[46]	3, 4	Examines the optimal path during the short-loading cycle. Models the wheel loader. Two-dimensional space discretization. Dynamic programming to find the optimal path through search. Optimally is defined in terms of fuel efficiency and environmental impact.	Optimality is defined as fuel efficiency and environmental impact which are very relevant metrics. The model includes the lifting operation. The model is validated against a real wheel loader.	No attempt is made to validate the suggested optimal control by implementing it on a real vehicle. As the author states, it is unclear how discretization influences the control.

Table 2. Cont.

Publ.	Steps	Summary	Advantages	Disadvantages
[47]	3, 4	Extension of previous work [46]. Path optimization based on topological information from the construction site. Compares three lift strategies. To solve the optimization problem, a grid-based dynamic programming search is used.	The given solution takes the terrain of the construction site into account, which is an important factor during operation. Solutions are analyzed in terms of important metrics such as production rate and fuel consumption.	No attempt to test the proposed optimal control on a real vehicle using some set of controllers. The lift functionality is only considered using 3 strategies, none of which seem similar to what operators offer.
[48]	3, 4	Examines the optimal switching time instant between backwards, stopping, and forward while tracking an a priori path. Approximate dynamic programming is used together with a neural network. The optimal switching time is found to be after 2.86 s.	The model is validated on data from a real wheel loader. Lift included in the model.	The optimal path is an a priori and it is unknown in what sense this path is optimal. The given solution is optimal in terms of switching times; however, it is unclear how this relates to metrics such as productivity and energy efficiency.
[42]	3, 4	An algorithm is proposed based on RRT* and CC steer to plan the trajectory between the scooping and the unloading point. Uses adaptive model predictive control. Does not control the lift or tilt.	Performs both path planning and path following. Considers the changes in the path depending on the velocity of the vehicle.	The solution does not seem to take productivity or energy efficiency into account during path planning. Uncertain whether the model has been validated using data from a real wheel loader. Lift action is not considered.
[45]	3, 4	Proposes a two-step algorithm for trajectory planning and trajectory tracking. Offline trajectory planning is performed using MPC due to computational requirements. Linear parameter varying model predictive control is used to follow the planned path. The solution does not consider tilt or lift.	The tracking algorithm tracks the generated path with a low error and fast online computation time. The solution is tested in a high-fidelity simulator, verifying the model. High explainability due to low reliance on randomness.	The solution does not consider the lift action which is important for an efficient cycle. Uncertainty of how realistic it is to perform the path generation offline in terms of adoption in real operators.
[51]	3, 4	Trains reinforcement learning agents to perform the navigation between the scooping point and the dumping point. This is achieved by having one agent perform the reversal and another agent performs the approach to the dumping point. This is carried out in a low-fidelity simulation.	Results indicate that task decomposition can aid in the automation of the short-loading cycle, especially when attempting to automate it using deep learning techniques.	Very low fidelity simulation where the agent can actuate the vehicle much faster than possible in reality. Does not consider productivity or energy efficiency in the reward function. No tests on real machines.
[43]	1, 3, 4	Generates paths between multiple arbitrary loading points and a single unloading point. An algorithm based on CC paths to generate a set of paths. The set of paths is scored using a novel scoring system to find the optimal path. Does not consider lift or tilt.	Considers multiple scooping points with a single unloading point and considers the multiple rounds an operator has to perform throughout the full short-loading cycle. The effectiveness of the solution is validated on a miniature wheel loader.	The solution does not consider the lift action which is important for an efficient cycle. The solution is based on minimizing the distance of the paths which, as mentioned earlier, is not a good metric in terms of productivity and/or energy efficiency.

Table 2. Cont.

Publ.	Steps	Summary	Advantages	Disadvantages
[44]	1, 3, 4	Novel solution of finding multiple scooping points with a single unloading and reversal point. Achieved by decomposing the task into 3 different tasks, where a cost map is used to find the scooping points, RL is used to rank said points, and CC paths are used to find the path.	Minimizes mileage of the vehicle rather than travelled distance where mileage is believed to be closely tied to fuel consumption. Considers the multiple scooping points for the entire task of the short-loading cycle. The solution ranks all possible scooping points and is capable of doing so until the entire pile is expedited.	The solution does not appear to consider the lift during path generation. The solution is built using a column model for the pile, where it is unclear whether it is a good enough approximation. The solution is not tested on some type of real vehicle or real data.
[56]	2, 3, 4	Examines optimal fuel usage and productivity during the short-loading cycle. Formulates a multistage optimization problem to capture multiple steps of the short-loading cycle.	Unifies the scooping and transport phase for optimization. Shows a fuel consumption reduction of 42.1% compared to real operators. Includes automatic gear shifting.	Solution not used to control a real vehicle. Appears to assume a singular dumping and loading position. Cycle time is used as a substitute for productivity.
[49]	3, 4, 6	Implements a driver-in-the-loop model. Analyzes actual driver data from the V-pattern work cycle. From this, MPC and LQR are used to determine the throttle, brake, and steering inputs. The trajectory is predicted and analyzed regarding the optimal path, energy flow, and loss.	Analyzes real operator data to create the driver model. The given regulators correctly track the trajectory by controlling the steering, throttle, and brake. The energy flow identifies acceleration as a large contributor to fuel usage.	Only compares the model with real data and does not test the control setup on an actual system. Lift and tilt are considered during modeling, but they seem to not have been considered during control.
[6]	Full	Demonstrates the full short-loading cycle autonomously on a real vehicle. The solution consists of three subsystems: measuring and modeling the environment, task planning, and motion control.	Demonstrates the full short-loading cycle on a real vehicle under specific circumstances. Fully explainable solution.	Low productivity and efficiency as the solution takes around 60 s to perform a single cycle.
[57]	Full	Examines the fuel efficiency and cycle time for the short-loading cycle. Formulates the short-loading cycle as an optimal control problem. Shows that the optimal path is unique and identical for the minimum fuel usage and minimum cycle time. Small perturbations to the boundary conditions, such as load receiver orientation angle, can remove this uniqueness.	Gives good insight into the impact of changes in the setup for the short-loading cycle and its effect on fuel efficiency. Furthermore, the work sets a good baseline of a cycle time of 25 s for any autonomous system.	It is unclear how close the model used is to a real vehicle as the solution has not been tested on a real machine.
[54]	Full	Formulates gravel scooping during the short loading cycle as an optimal control problem. Simulates the gravel pile using a discrete element simulation. The optimal path is assumed to be known. Dynamic programming is used to find the optimum concerning fuel efficiency.	Compares the proposed optimal control to real operators, showing the optimal control to have 15% higher fuel efficiency. Considers all the main actuators throughout the entire cycle.	As the author states, further validation is needed for the proposed control, as the result might change due to the machine and environment modeling.

Table 2. Cont.

Publ.	Steps	Summary	Advantages	Disadvantages
[52]	Full	Proposes a deep LSTM network for brake aperture during the short-loading cycle. The network is trained on time-series data from different operators in different environments. The proposed network can correctly predict the braking aperture throughout the cycle.	Solution based on data from multiple expert operators in different types of environments. The solution shows that an LSTM-based solution can handle long-term dependencies, which are important during the short-loading cycle. Performance of the solution compared to real data.	It is unclear how this type of single-action system can work together with a set of other systems for automation. The amount of braking differs depending on skill level, where the absolute highest skilled operators rarely use the break, as the necessity to break signifies excessive acceleration, wasting fuel.
[53]	Full	Trains an LSTM and neural network on real operator data for the short-loading cycle to predict the throttle and state values of the wheel loader. The state consists of tilt pressure, lift pressure, engine speed, and velocity. Qualitatively, the network can correctly predict both the state and throttle value.	Data are collected from skilled drivers performing the short-loading cycle on a real vehicle scooping either small or large gravel. The results indicate that an LSTM can correctly predict the next state. This can be very helpful if correctly modeling parts of the wheel loader interaction with the pile.	Due to the black-box nature of deep learning methods, this type of solution has low explainability and there might be edge cases that lead to unexpected behavior. If some autonomous solution relies on this type of state prediction, the long tail of prediction (the last 1% accuracy is very difficult to reach), but it is very important for state prediction.

### 3.3. Qualitative Analysis

From the results of the survey, it is clear that there have been attempts to automate different parts of the short-loading cycle using a wide range of techniques. None of these solutions have resulted in a generally applicable solution capable of performing the short-loading cycle in any environment. However, the solutions presented above have different levels of maturity, where some are more ready than others.

Technology readiness level (TRL) [59] is a systematic measurement that assesses the maturity of a given technology. To assess maturity, TRL consists of nine levels, where a higher level corresponds to a higher maturity of the technology. It is important to note that TRL begins with applied research, hence one could consider there existing an additional level of zero that encompasses fundamental research.

The nine levels are defined as follows: (Level 1) Basic principles are observed, where scientific research is conducted and begins to be translated into applied research but with no experimental proof of concept. None of the surveyed articles fall under this category as most works include types of practical applications. (Level 2) The technology concept is formulated, where basic principles and concepts are established with some feasibility assessment such as initial experiments. For example, simulations examining path generation without testing on miniature vehicles or high-fidelity simulations [38]. (Level 3) Experimental proof of concept is introduced, where separate elements of the system are validated within a lab, for example, performing Steps 1 and 2 in a high-fidelity simulation using reinforcement learning [33]. (Level 4) Technology is validated in the lab, where the basic technology is validated in a relatively low-fidelity environment to verify different components working together, for example, when training in a simulation and then performing scooping on a miniature wheel loader with no additional training [31]. (Level 5) Technology is validated in a relevant setting, where the basic technology is validated in a high-fidelity environment to confirm the integration of components. (Level 6) Technology is demonstrated in a relevant setting, where a prototype is showcased in an environment that surpasses TRL 5, for example, when training an agent through imitation learning to scoop material with a real vehicle or performing the full cycle on a real vehicle using rule-based systems. (Level 7) System prototype demonstration is carried out in the operational environment, where the prototype is demonstrated in an actual operational environment. (Level 8) The system is complete and qualified, where the system is proven



to work under specific conditions and is in some way certified by regulators to be deployed in said environment. (Level 9) The actual system is proven in an operational environment, where the system is proven correct through tests and evaluations. The system at this point can potentially keep improving throughout the years. None of the surveyed articles reach Levels 7–9; thus, all articles are contained within Levels 2–6. The qualitative TRL for each work is summarized in Table 3.

From Table 3, we can see that the two data-driven approaches have a TRL of six, where both of these works study the bucket-filling phase matching operator performance. Two rule-based solutions have a TRL of six; however, both were only tested in a single environment and only the scooping work was capable of good performance. This means that it is unclear whether or not they can match the reported performance when some variables change. It is clear to us that any future MVP should be capable of scooping material from a pile of small to medium-sized material at a very similar level to an expert operator in terms of fill factor.

When it comes to locating and navigating to the scooping point, some work has shown potential in simulation but it is currently unclear what the best approach is. This also holds for the navigation towards the dump truck. However, there is a requirement for some type of system that allows the system to perceive the world to locate these positions. Furthermore, there might be a need for the system to translate the position in, for example, a camera feed to some other position, such as GPS, to facilitate navigation. Here, more research is needed, both in terms of rule-based and data-driven approaches to reach the required level for an MVP.

**Table 3.** The technology readiness of each surveyed work. The individual numbers coincide with a single citation. The blue-marked citations are rule-based, the red citations are data-driven, and the green citations use a mix of both.

		Technology Readiness Level								
		1	2	3	4	5	6	7	8	9
Publication year	2001				37		22			
	2004				23					
	2005				24					
	2006		38		26					
	2008						6			
	2011			27						
	2013			41	39					
	2014			57						
	2015			49	43					
	2016			55						
	2017		46	19						
	2018		47	54						
	2019					48		7		
	2020		42	52		29		28		
2021			33, 34, 53, 44		31, 30					
2022			45, 36							
2023		35	50, 51		56					

From Figure 3, only a single data-driven approach [51] has been applied to the navigation problem; hence, further exploration of such solutions for navigation and observing their performance compared to rule-based systems could be interesting. This also includes Step 6, since in order to fill a tipping body of a dump truck, the cycle has to be performed 3–4 times; therefore, the system needs to navigate to the start position after finishing dumping material. There should be inspiration that can be taken from similar domains due to



the nature of the navigation task. However, it is important to note the actuated nature of the wheel loader together with the wheel loader mechanics.

From our survey, no identified work has examined Step 5, dumping, which is not surprising when only considering an empty tipping body. However, when the tipping body has had at least one bucket of material dumped into it, there is a need for long-term memory as the system should not dump material in the same spot. Furthermore, when dumping material in the tipping body, the material needs to be spread both horizontally and vertically, making the process potentially complex. This means that when examining multiple rounds, which is a requirement for an MVP for the automation of the short-loading cycle, this seems to be non-trivial. Hence, this is an area where more research is needed.

Figure 3 shows what steps of the short-loading cycle have been examined throughout the years, together with the type of technology used. The vast majority of the work has been rule-based, especially for Steps 3 and 4. Steps 3 and 4 seem to mostly be examined together, which is probably a result of the highly coupled nature of these steps. This is due to many of the variables such as the reversal point, orientation towards, and ending point being important for both steps, making them difficult to optimize separately. Furthermore, very little amount of work has used data-driven solutions to examine these steps.

Data-driven solutions have almost exclusively examined scooping or scooping together with the approach towards the pile while showing good results. Pile data are often difficult to model and the material can often act in a stochastic way, adding to the difficulty of writing a rule-based system which can maintain high productivity.

Only a single rule-based work has demonstrated the full short-loading cycle in a relevant environment. Rather than demonstrating the full cycle, the rest of the rule-based systems examine the optimal control from some modeling. The data-driven solutions attempted to predict a single or a few actions from real data such as brake aptitude.

#### 4. Plausible MVP

From the discussion above, it is clear that there is still a large set of challenges that need to be overcome to realize a fully autonomous solution which is ready for deployment. However, the continuation of the presented approaches is furthering the domain progressively. To continue this positive trend, we attempt to present a reasonable MVP that manages to capture the most important metrics, which are productivity and energy efficiency, while also showing consideration for the challenges outlined in Section 2. Furthermore, this MVP attempts to capture what is required to see some adoption in the industry while considering the current status of the domain together with what appears to be possible to achieve in a shorter time horizon.

**Prescooping:** Prescooping entails both the task of identifying the scooping position and navigating towards it while positioning the wheel loader and bucket in the correct scooping position. From previously presented work, it can be difficult to obtain the ground truth for the optimal scooping point, as different expert operators choose different scooping points on the same pile structure. However, it is clear that operators choose the scooping point at a convex point of the pile to simplify the correct loading of the bucket.

The MVP should thus be capable of determining a good scooping point through some vision system, such as LIDAR or cameras, based on where the pile is convex. Furthermore, the MVP should be able to navigate towards the identified scooping point while positioning the bucket and making the rest of the vehicle ready for scooping under some assumptions.

**Scooping:** As previously mentioned, the step that has been given the most amount of attention is the scooping step, where multiple different approaches have been able to match expert operator performance of small and medium grain-sized materials. The majority of these high-performing solutions leveraged deep learning-based solutions to deal with the difficulties of modeling the pile–bucket interactions during scooping.

Due to this, we consider that a scooping system for the MVP is capable of matching expert operators in terms of both fill factor and fill time on any small- and medium-sized material.

**Navigation under load:** After the scooping step is completed, the operator reverses from the pile while moving the bucket upwards until the bucket is sufficiently high, at which point the operator changes direction to approach the tipping body of the dump truck. During navigation, the operator has to understand how much steering actuation is proper depending on the load in the bucket, ground material, and conditions. As discussed in Section 2, these interactions can be difficult to model.

Many of the previously proposed solutions abstract away many of the difficult variables of the navigation steps, such as lift timing. However, understanding and dealing with at least some of these variables is a requirement of any MVP to see adoption. Furthermore, the MVP should be capable of autonomously deciding both the dumping and the reversal point depending on the relevant variables. Once these points are established, the MVP has to be capable of navigating between the identified points.

**Dumping:** While dumping, the operator spreads the scooped material in the tipping body to balance the load for the operator of the dump truck. This is achieved by leveraging forward and backward motion while emptying the bucket which allows better dispersion of the material in the tipping body. Furthermore, the chosen dumping point depends on where the operator dumped in the previous rounds of the cycle.

Very little work has been carried out in terms of optimizing the dumping phase, especially over time. This step is important as a correctly loaded dump truck is able to safely carry more material at a time. An important consequence of an underloaded dump truck is that the overall efficiency of the entire process decreases considerably.

Because of the importance of this step, we consider that the MVP is required to be capable of performing the dumping in an efficient way to be able to safely fit the same amount of material in the tipping body as an expert operator. Furthermore, this solution should focus on minimizing spillage rather than maximizing productivity or fuel efficiency to reduce the amount of cleaning required between cycles.

**Repeating:** Lastly, the MVP needs to be capable of performing many rounds of the cycle autonomously to completely fill the tipping body. This requires the solution to be able to navigate back after dumping, determining multiple scooping points and multiple dumping points. Furthermore, as mentioned, the dumping points depend on the previous dumping points, meaning that the MVP is required to understand and deal with some long-term dependencies.

**Limitations:** This MVP is not a generally applicable solution, meaning that there are a set of limitations that have to be addressed after the initial feedback.

First, because of the safety concerns outlined in Section 2, this system is treated as a conventional autonomous system where any safety concerns or uncertainties lead to a complete halt of the system. This is to minimize the risk of damage to the vehicle, infrastructure, and other agents.

Second, the MVPs only need to work with a set of different materials where larger types of materials, such as boulders, are not considered. This is because scooping these types of materials is more complex compared to smaller, more homogenous materials. In addition, from this survey, no literature was identified which has examined this type of scooping task.

Third, the MVP does not have to be capable of dealing with very loose ground while navigating under load, as this introduces very complex navigation, especially under load. Furthermore, because only hard ground is considered and the MVP should minimize spillage, the MVP does not have to be capable of cleaning the work area between cycles. This allows for us to make the assumption that the MVP only considers the current cycle and does not have to change the behavior in order to maximize efficiency over a very long time horizon.

Fourth, the MVPs only have to consider very high-level wear variables, such as wheel spin, and not very small ones, such as internal wear variables. These include but are not limited to brake wear due to hard braking habits, transmission wear due to shifting, and bucket wear due to scooping habits.

Lastly, the MVP does not perform the cycle with optimal productivity and fuel efficiency; however, the MVP should be capable of improving over time in terms of productivity and fuel efficiency. Furthermore, when it comes to the steps where very good performance is observed, such as scooping, the MVP should be able to match these systems.

The limitations outlined above are made to be able to obtain, as quickly as possible, a reasonable solution in real environments to collect valuable feedback. However, all these limitations have to be addressed at some point over time to realize a generally applicable solution.

## 5. Framework for Automation

### 5.1. Background

As previously discussed, the short-loading cycle can be divided into a series of distinct steps to perform in succession. Each of these steps need to have a set of different possible solutions leveraging various technologies, where it is unclear which is the best-performing one. One solution might be the best-performing solution for single-step validation; however, it might not perform as well when combined with other solutions to carry out the full cycle.

The ability to test new novel solutions that only address one or a few steps of the short-loading cycle in the context of the entire cycle is preferred as it allows quantification of how well it performs in terms of full-cycle efficiency. For example, there might be one solution that seems to perform the navigation between the pile and the tipping body without any load in the bucket while performing poorly with load in the bucket.

Furthermore, productivity and energy efficiency are defined in terms of the amount of either material moved or fuel used over a longer period, which requires either full cycle experiments or assumptions concerning the other steps.

Some previous work has explicitly leveraged FSMs within their solution; alternatively, the solution can be described as an FSM, for example, decomposing the scooping task into fuzzy behaviors using finite-state machines [22], flow charts of modeling and action [34], or step-based solution [6].

Thus, the features that we want the finished system to include are heavily associated to easily changing and comparing different solutions for each substep, the ability to perform such a comparison in terms of the full task, facilitation of describing the given task as a set of an arbitrary number of complex finite state machines, and the possibility for each state to be defined as any function.

Furthermore, the system should allow for any data to be accessible between states, and each function should be able to read information from, for example, the sensors.

From this, we propose and present a proof-on-concept finite-state machine-based (FSM) framework to aid in the research of automation of the short-loading cycle and, in addition, to be a tool in the pursuit of an MVP.

### 5.2. Framework Design

The proposed framework (the implementation of the given framework can be found at <https://github.com/Felmrt-0/PyFSMwDB> (accessed on 23 May 2024)) is an FSM-based solution where each state in the FSM is defined as any arbitrary function or a nested FSM. An arbitrary function is any function that can be described within the language. In the case of automating the short loading cycle, this could, for example, be a neural network that performs scooping, a rule-based system that identifies the reversal point, or an agent trained through reinforcement learning that navigates towards the dump truck. Furthermore, if it is instead easier to describe a sub-step as an FSM, it is possible to nest an FSM within a state rather than having a function. The goal of this is the ability to define each sub-step in terms of a single arbitrary function or as a nested FSM in case a sub-step needs to be decomposed further.

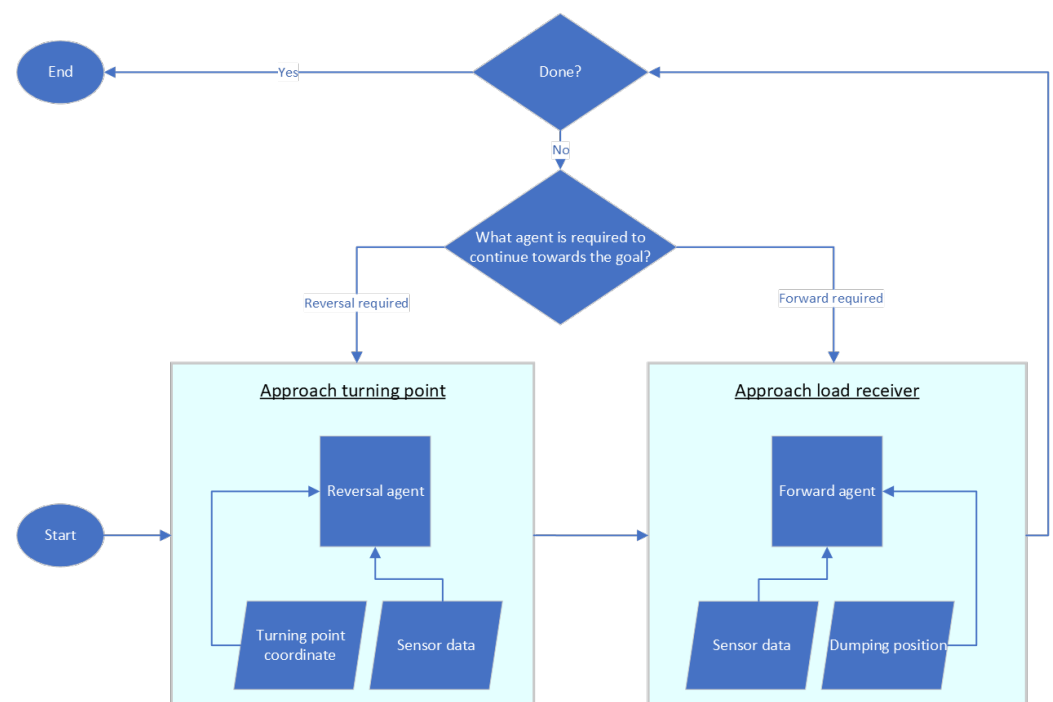
To transition between states, conditions or a set of conditions are defined within the arbitrary function that allows for transition from the current state to the corresponding connecting states. This means that a single state can have multiple transitions, allowing for

some terminal state that, for example, can act as a “kill switch”, where, if something goes wrong, the system is completely halted and the physical machine stops.

Sometimes it is needed to propagate collected information through the transitions or unnesting as information collected in a previous step is needed in a later state, for example, if in some state the scooping point is selected from a sensor, such as a camera, and then this identified scooping point is propagated to the next state where the defined function uses that to navigate towards the pile. Because of this, any amount of information can be sent through the transitions if it is only required during runtime. The information that is required to be propagated between states is provided by the sub-processes at termination and received by the main process. It is then the main process’s responsibility to propagate this to the next state and process. If the data needs to be stored persistently, such as new training data, this is possible to achieve either through database integration or through just persistent saving.

### 5.3. Example Use Case of the Automation Framework

To explain the idea of the automation framework further, Figure 4 shows a small use case. In this use case, the task is to reverse from the pile after scooping material to the reversal point and navigate to the dump truck, stopping before the material is to be dumped into the tipping body. This means that when in the initial state, the vehicle has just finished scooping, and the terminal state is when the loader is in a position ready to dump material.



**Figure 4.** Example of the proposed framework. This example consists of two sequential agents performing the navigation between the pile and dump truck (Steps 3 and 4). In this case, the nodes are defined by two functions which are both given predefined data and sensor data used to complete the navigation task.

As mentioned, each state within the FSM is defined by an arbitrary function that executes until one of the swapping conditions is met. We assume that, for this use case, the two arbitrary functions defining the two states (navigation) are two sequential reinforcement learning agents [51]. Furthermore, we also assume, for this example, that the sequential agents function correctly in reality. Here, there is no requirement to send information between the two states as they have been developed in isolation and the second state

is not dependent on any of the information from the first. However, both require reading from a set of different sensors to correctly perform the task, and a swapping condition is required.

There exists a large amount of different possible swap conditions that could be used to transition between states. One possible set of transitions is to leverage the vehicle's vicinity to the reversal point together with the vehicle's current speed. Once the vehicle is close enough to the reversal point with low speed, we consider the function in the first state to be completed. To transition from navigation to the end state, one possible condition is to use lift height together with the distance from the dump truck, as once the bucket is over the tipping body, the material can be dumped into the dump truck.

For this example, the termination criterion is simply that the process contained within the approach load receiver state terminates, and once it does, we assume that the full process is finished. However, the same holds for the termination condition as for the swap conditions, where any arbitrary function can be used to determine whether to terminate or not.

## 6. Open Issues and Gaps

As the research area of automation of the short-loading cycle continues to evolve, several challenges lie ahead to realize the MVP. These challenges include but are not limited to abstraction level due to rule-based interfaces, delimitation of the solution to enforce a set of behaviors, understanding the effect of different assumptions, and balancing of rule-based and data-driven solutions. Overcoming these challenges is crucial for realizing the full potential of automation of the short-loading cycle and facilitating the integration into daily operations.

### 6.1. Abstraction Level Due to Rule-Based Interfaces

There exists a wide set of abstraction levels that can be enforced using rule-based interfaces to limit the behavior of the solution. These abstraction levels can simplify the learning of neural networks due to decreasing the relevant search space. However, these types of abstractions can also interfere with the proximity with which the solution can reach a global optimum.

Some possible abstraction levels include high-level commands, trajectory planning, and environmental abstractions.

1. High-level commands allow for a rule-based system to provide high-level commands such as "turn left" or "turn right" which can act as abstractions to guide the behavior without explicitly defining low-level motor control.
2. Trajectory planning leverages some heuristics to generate abstract trajectories that the network could follow. This allows for the network to focus on following the trajectory rather than dealing with complexities such as reversal point identification or slip angle minimization.
3. Environmental abstraction can provide a higher abstraction level for the network when perceiving the environment. For example, rather than feeding raw sensor data to the network, we could instead pre-process the data to extract objects and important features. This would allow the network to not have to first learn low-level perception, and the network could instead only focus on learning higher-level decision-making.

Providing this type of high-level directive can facilitate the learning possibility of a learning agent; however, it comes with the trade-off of not being able to reach the theoretical global optimum. Currently, it is unclear what abstraction level offers the best trade-off in terms of learning and important properties. Is an abstraction hiding the lower-level control signals the best approach, or could an end-to-end solution still be a superior option? Could data-driven approaches be used for modeling the system when, for example, trying to use a PID regulator, or could the PID be completely tuned using some type of a data-driven approach?

In addition, the ways in which feedback is collected and the assumptions are made can play a big role especially when performing development by some proxy such as simulation or miniature wheel loaders. Sensors are not perfect, leading to small differences in the outputted values. The ways in which they are attached differ between machines, and the way in which a machine ultimately works is unique from machine to machine. This is important to consider when defining the abstraction level due to rule-based interfaces together with well-known issues such as time delays, noise, and sensor failure.

### *6.2. Enforce Safe Behavior*

It does not seem reasonable that we will see autonomous wheel loaders freely roam around during day-to-day operations in the near future. Furthermore, even driver assistance systems based fully on data-driven approaches seem quite far away. A significant reason for this is that there is currently no possible way to ensure safe operations for the human inside the machine, the humans around the machine, and machine-to-machine safety. However, safety is relevant throughout the full-time axis of automation of the short-loading cycle.

Due to data-driven approaches currently suffering from low explainability, we currently require some other system to enforce the safe behavior of the system. It is not enough to attempt to embed safe behavior into the data as learning edge cases is difficult due to the statistical nature of data-driven approaches.

Some options to enforce safe behavior are formal verification, which is highly complex and time-consuming, and hard limits or prohibited actions, both of which might prevent us from reaching the desired efficiency. However, currently, it is unclear how this should be enforced and how that will affect the final efficiency of the solution.

### *6.3. Effect of Assumptions*

When looking to create an autonomous system, it is important to understand the set of assumptions that are made and the effect they will have on the system, especially when these assumptions fail. Solutions often make assumptions about the possible sensor data available and the accuracy of said sensor. For example, often, some type of positioning sensors are used where the assumptions are that there is 100% availability and that the value is very accurate. This does not always hold due to failure in the sensor system or some type of drift in the measurement. Furthermore, it is not always possible to even use global positioning; for example, in deep mines where the signal cannot pass through the ground and/or walls.

It might be possible to use some other sensor to extract positioning, such as vision, but that would lead to another set of assumptions to be considered.

These assumptions need to be evaluated and validated towards real-world operation iteratively, making sure they align with the conditions and situations that the system will encounter. Failing to do so might lead to a poor system, while correct assumptions can refine the system, making it more robust and efficient.

### *6.4. Consideration Using Data-Driven Approaches Together with Classical Control Theory*

When examining different types of solutions, it becomes clear that they all have strengths and weaknesses which have to be considered. This holds when using data-driven approaches together with classical control theory solutions. It seems improbable to completely use deep learning methods to solve different control problems as there are always trade-offs to consider. Furthermore, control theory methods have quite a few properties that are preferred when automating this type of task.

These methods are particularly well-suited for tasks that are safety-critical, such as the automation of the short-loading cycle, where system stability, explainability, robustness, and analytical design are crucial. However, some challenges that exist for these types of methods include modeling complexity, model error sensitivity, and design complexity and tuning.



This means that some type of cooperation between the two types of solutions is probably preferable; however, it is currently unclear to what extent data-driven approaches should be used in the domain of automation of the short-loading cycle.

## 7. Conclusions and Future Work

One common task in the construction industry is the short-loading cycle, which involves using a wheel loader to move material from a pile to the tipping body of a nearby dump truck. The short-loading cycle is a representative and repetitive task performed in large volumes, making it a great candidate for automation.

In this paper, a survey and an analysis of the literature regarding the automation of the short-loading cycle are presented. The examination delves into the challenges that are making this type of automation difficult to realize. A comprehensive literature review assesses the current state of short-loading cycle automation, providing an overview of the advantages and disadvantages associated with each solution. From the survey and subsequent analysis, the requirements for a minimal viable product capable of performing the short-loading cycle autonomously are defined. An automation tool based on finite-state machines is presented to facilitate the automation of the short-loading cycle and allow for the validation of different solutions for each substep. Lastly, future challenges are identified which relate to topics such as abstraction levels, enforcing safe behavior, the effect of assumptions, and considerations using data-driven approaches together with classical control theory.

**Author Contributions:** Conceptualization, C.B. and U.B.; methodology, C.B.; validation, C.B.; investigation, C.B.; writing—original draft preparation, C.B.; writing—review and editing, C.B., U.B., H.A. and F.S.; visualization, C.B.; supervision, U.B., H.A. and F.S.; project administration, U.B.; funding acquisition, U.B., H.A. and F.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by Sweden's Innovation Agency under grant number 2021-05035.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data is contained within the article.

**Acknowledgments:** The authors thank Volvo Construction Equipment AB for their contribution and insights. The authors also thank Andreas Söderman, Arian Asghari, Felix Mårtensson, and Mattias Öz for the implementation of the framework.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Bodin, U.; Andersson, U.; Dadhich, S.; Uhlin, E.; Marklund, U.; Häggströmff, D. Remote controlled short-cycle loading of bulk material in mining applications. *IFAC-PapersOnLine* **2015**, *48*, 54–59. [CrossRef]
2. Frank, B.; Skogh, L.; Filla, R.; Froberg, A.; Alaküla, M. On Increasing Fuel Efficiency by Operator Assistance Systems in a Wheel Loader. In Proceedings of the International Conference on Advanced Vehicle Technologies and Integration (VTI), Changchun, China, 16–19 July 2012; pp. 1–7. [CrossRef]
3. Volvo Construction Equipment. Volvo LX03. 2021. Available online: <https://www.volvoce.com/global/en/about-us/what-we-believe-in/innovation-at-our-core/concept-lab/lx03/> (accessed on 23 May 2024).
4. Dadhich, S.; Bodin, U.; Andersson, U. Key challenges in automation of earth-moving machines. *Autom. Constr.* **2016**, *68*, 212–222. [CrossRef]
5. Filla, R. Evaluating the efficiency of wheel loader bucket designs and bucket filling strategies with non-coupled DEM simulations and simple performance indicators. In Proceedings of the Fachtagung Baumaschinentechnik, Dresden, Germany, 17–18 September 2015; pp. 1–20. [CrossRef]
6. Sarata, S.; Koyachi, N.; Sugawara, K. Field test of autonomous loading operation by wheel loader. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Nice, France, 22–26 September 2008; pp. 2661–2666. [CrossRef]

7. Dadhich, S.; Sandin, F.; Bodin, U.; Andersson, U.; Martinsson, T. Field test of neural-network based automatic bucket-filling algorithm for wheel-loaders. *Autom. Constr.* **2019**, *97*, 1–12. [[CrossRef](#)]
8. Borngrund, C.; Sandin, F.; Bodin, U. Deep-learning-based vision for earth-moving automation. *Autom. Constr.* **2022**, *133*, 104013. [[CrossRef](#)]
9. Fang, W.; Ding, L.; Love, P.E.; Luo, H.; Li, H.; Peña-Mora, F.; Zhong, B.; Zhou, C. Computer vision applications in construction safety assurance. *Autom. Constr.* **2020**, *110*, 103013. [[CrossRef](#)]
10. You, K.; Zhou, C.; Ding, L. Deep learning technology for construction machinery and robotics. *Autom. Constr.* **2023**, *150*, 104852. [[CrossRef](#)]
11. Parascho, S. Construction Robotics: From Automation to Collaboration. *Annu. Rev. Control Robot. Auton. Syst.* **2023**, *6*, 183–204. [[CrossRef](#)]
12. Lenarduzzi, V.; Taibi, D. MVP Explained: A Systematic Mapping Study on the Definitions of Minimal Viable Product. In Proceedings of the 42th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), Limassol, Cyprus, 31 August–2 September 2016; pp. 112–119. [[CrossRef](#)]
13. Metz, L.D. *Dynamics of Four-Wheel-Steer Off-Highway Vehicles*; SAE Technical Paper; SAE: Warrendale, PA, USA, 1993.
14. Troyanovskaya, I.P.; Shepelev, S.D. Design of Agricultural Loader with Articulated Frame. In Proceedings of the 6th International Conference on Industrial Engineering (ICIE 2020), Sochi, Russia, 18–22 May 2020; Radionov, A.A., Gasiyarov, V.R., Eds.; Springer: Cham, Switzerland, 2021; pp. 142–150.
15. Kim, H.; Oh, K.; Ko, K.; Kim, P.; Yi, K. Modeling, validation and energy flow analysis of a wheel loader. *J. Mech. Sci. Technol.* **2016**, *30*, 603–610. [[CrossRef](#)]
16. Muro, T. Estimation of wear life of wheel loader tyre. *J. Terramech.* **1992**, *29*, 137–147. [[CrossRef](#)]
17. Rahimdel, M.J. Residual lifetime estimation for the mining truck tires. *Proc. Inst. Mech. Eng. Part D J. Automob. Eng.* **2022**, *237*, 3232–3244. [[CrossRef](#)]
18. Frank, B.; Skogh, L.; Alaküla, M. On wheel loader fuel efficiency difference due to operator behavior distribution. In Proceedings of the Commercial Vehicle Technology 2012, Kaiserslautern, Germany, 13–15 March 2020; Shaker: Herzogenrath, Germany, 2012; pp. 329–346. Available online: [http://www.iea.lth.se/publications/Papers/Frank\\_2012.pdf](http://www.iea.lth.se/publications/Papers/Frank_2012.pdf) (accessed on 28 February 2023).
19. Filla, R.; Frank, B. Towards Finding the Optimal Bucket Filling Strategy through Simulation. In Proceedings of the 15th Scandinavian International Conference on Fluid Power, Linköping, Sweden, 7–9 June 2017; Linköping University Electronic Press: Linköping, Sweden, 2017; pp. 402–417. [[CrossRef](#)]
20. Xiao, Y.; Watson, M. Guidance on Conducting a Systematic Literature Review. *J. Plan. Educ. Res.* **2019**, *39*, 93–112. [[CrossRef](#)]
21. Borngrund, C.; Bodin, U.; Sandin, F. Machine Vision for Construction Equipment by Transfer Learning with Scale Models. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July 2020; pp. 1–8. [[CrossRef](#)]
22. Lever, P.J.A. An automated digging control for a wheel loader. *Robotica* **2001**, *19*, 497–511. [[CrossRef](#)]
23. Sarata, S.; Osumi, H.; Kawai, Y.; Tomita, F. Trajectory arrangement based on resistance force and shape of pile at scooping motion. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), New Orleans, LA, USA, 26 April–1 May 2004; Volume 4, pp. 3488–3493. [[CrossRef](#)]
24. Sarata, S.; Weeramhaeng, Y.; Tsubouchi, T. Approach Path Generation to Scooping Position for Wheel Loader. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Barcelona, Spain, 18–22 April 2005; pp. 1809–1814. [[CrossRef](#)]
25. Konishi, Y.; Takahashi, H. Autonomous Loading of Rocks by Use of Intelligent Loaders with a Vision System-A Concept of Autonomous Loading and Path Generation. In Proceedings of the 5th Asia-Pacific Regional Conference of International Society for Terrain Vehicle Systems (ISTVS), Seoul, Republic of Korea, 19–23 October 1998; Volume 1, pp. 413–420.
26. Sarata, S.; Weeramhaeng, Y.; Horiguchi, A.; Tsubouchi, T. V Shape Path Generation for Loading Operation by Wheel Loader. In *Field and Service Robotics*; Corke, P., Sukkariah, S., Eds.; Springer: Berlin/Heidelberg, Germany, 2006; pp. 591–602.
27. Magnusson, M.; Almqvist, H. Consistent Pile-shape quantification for autonomous wheel loaders. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), San Francisco, CA, USA, 25–30 September 2011; pp. 4078–4083. [[CrossRef](#)]
28. Dadhich, S.; Sandin, F.; Bodin, U.; Andersson, U.; Martinsson, T. Adaptation of a wheel loader automatic bucket filling neural network using reinforcement learning. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July 2020; pp. 1–9. [[CrossRef](#)]
29. Yang, W.; Strokina, N.; Serbenyuk, N.; Ghabchelloo, R.; Kämäräinen, J. Learning a Pile Loading Controller from Demonstrations. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 4427–4433. [[CrossRef](#)]
30. Yang, W.; Strokina, N.; Serbenyuk, N.; Pajarinen, J.; Ghabchelloo, R.; Vihonen, J.; Aref, M.M.; Kämäräinen, J.K. Neural Network Controller for Autonomous Pile Loading Revised. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 2198–2204. [[CrossRef](#)]

31. Azulay, O.; Shapiro, A. Wheel Loader Scooping Controller Using Deep Reinforcement Learning. *IEEE Access* **2021**, *9*, 24145–24154. [[CrossRef](#)]
32. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. *arXiv* **2019**, arXiv:1509.02971. Available online: <https://arxiv.org/abs/1509.02971> (accessed on 9 September 2023).
33. Backman, S.; Lindmark, D.; Bodin, K.; Servin, M.; Mörk, J.; Löfgren, H. Continuous Control of an Underground Loader Using Deep Reinforcement Learning. *Machines* **2021**, *9*, 216. [[CrossRef](#)]
34. Huang, J.; Kong, D.; Gao, G.; Cheng, X.; Chen, J. Data-Driven Reinforcement-Learning-Based Automatic Bucket-Filling for Wheel Loaders. *Appl. Sci.* **2021**, *11*, 9191. [[CrossRef](#)]
35. Yao, J.; Edson, C.P.; Yu, S.; Zhao, G.; Sun, Z.; Song, X.; Stelson, K.A. Bucket Loading Trajectory Optimization for the Automated Wheel Loader. *IEEE Trans. Veh. Technol.* **2023**, *72*, 6948–6958. [[CrossRef](#)]
36. Chen, Y.; Jiang, H.; Shi, G.; Zheng, T. Research on the Trajectory and Operational Performance of Wheel Loader Automatic Shoveling. *Appl. Sci.* **2022**, *12*, 12919. [[CrossRef](#)]
37. Takahashi, H.; Konishi, Y. Path Generation for Autonomous Locomotion of Articulated Steering Wheel Loader. *Comput.-Aided Civ. Infrastruct. Eng.* **2001**, *16*, 159–168. [[CrossRef](#)]
38. Yossawee, W.; Tsubouchi, T.; Kurisu, M.; Sarata, S. A semi-optimal path generation scheme for a frame articulated steering-type vehicle. *Adv. Robot.* **2006**, *20*, 867–896. [[CrossRef](#)]
39. Takei, T.; Ichikawa, K.; Okawa, K.; Sarata, S.; Tsubouchi, T.; Torige, A. Path planning of wheel loader type robot for scooping and loading operation by genetic algorithm. In Proceedings of the 13th International Conference on Control, Automation and Systems (ICCAS), Gwangju, Republic of Korea, 20–23 October 2013; pp. 1494–1499. [[CrossRef](#)]
40. Reeds, J.; Shepp, L. Optimal paths for a car that goes both forwards and backwards. *Pac. J. Math.* **1990**, *145*, 367–393. [[CrossRef](#)]
41. Alshaer, B.; Darabseh, T.; Alhanouti, M. Path planning, modeling and simulation of an autonomous articulated heavy construction machine performing a loading cycle. *Appl. Math. Model.* **2013**, *37*, 5315–5325. [[CrossRef](#)]
42. Shi, J.; Sun, D.; Qin, D.; Hu, M.; Kan, Y.; Ma, K.; Chen, R. Planning the trajectory of an autonomous wheel loader and tracking its trajectory via adaptive model predictive control. *Robot. Auton. Syst.* **2020**, *131*, 103570. [[CrossRef](#)]
43. Takei, T.; Hoshi, T.; Sarata, S.; Tsubouchi, T. Simultaneous determination of an optimal unloading point and paths between scooping points and the unloading point for a wheel loader. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 5923–5929. [[CrossRef](#)]
44. Kawabe, T.; Takei, T.; Imanishi, E. Path planning to expedite the complete transfer of distributed gravel piles with an automated wheel loader. *Adv. Robot.* **2021**, *35*, 1418–1437. [[CrossRef](#)]
45. Song, R.; Ye, Z.; Wang, L.; He, T.; Zhang, L. Autonomous Wheel Loader Trajectory Tracking Control Using LPV-MPC. In Proceedings of the 2022 American Control Conference (ACC), Atlanta, GA, USA, 8–10 June 2022; pp. 2063–2069. [[CrossRef](#)]
46. Hong, B.; Ma, X. Path planning for wheel loaders: A discrete optimization approach. In Proceedings of the IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, Japan, 16–19 October 2017; pp. 1–6. [[CrossRef](#)]
47. Hong, B.; Ma, X. Path optimization for a wheel loader considering construction site terrain. In Proceedings of the IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 26–30 June 2018; pp. 2098–2103. [[CrossRef](#)]
48. Sardarmehni, T.; Song, X. Sub-Optimal Control of Autonomous Wheel Loader with Approximate Dynamic Programming. In Proceedings of the Dynamic Systems and Control Conference, Park City, UT, USA, 8–11 October 2019; pp. 1–8. [[CrossRef](#)]
49. Oh, K.; Kim, H.; Ko, K.; Kim, P.; Yi, K. Integrated wheel loader simulation model for improving performance and energy flow. *Autom. Constr.* **2015**, *58*, 129–143. [[CrossRef](#)]
50. Sardarmehni, T.; Song, X. Path Planning and Energy Optimization in Optimal Control of Autonomous Wheel Loaders using Reinforcement Learning. *IEEE Trans. Veh. Technol.* **2023**, *72*, 9821–9834. [[CrossRef](#)]
51. Borngrund, C.; Bodin, U.; Sandin, F.; Andreasson, H. Autonomous Navigation of Wheel Loaders using Task Decomposition and Reinforcement Learning. In Proceedings of the 2023 IEEE 19th International Conference on Automation Science and Engineering (CASE), Auckland, New Zealand, 26–30 August 2023; pp. 1–8. [[CrossRef](#)]
52. Shi, J.; Sun, D.; Hu, M.; Liu, S.; Kan, Y.; Chen, R.; Ma, K. Prediction of brake pedal aperture for automatic wheel loader based on deep learning. *Autom. Constr.* **2020**, *119*, 103313. [[CrossRef](#)]
53. Huang, J.; Cheng, X.; Shen, Y.; Kong, D.; Wang, J. Deep Learning-Based Prediction of Throttle Value and State for Wheel Loaders. *Energies* **2021**, *14*, 7202. [[CrossRef](#)]
54. Frank, B.; Kleinert, J.; Filla, R. Optimal control of wheel loader actuators in gravel applications. *Autom. Constr.* **2018**, *91*, 1–14. [[CrossRef](#)]
55. Nezhadali, V.; Frank, B.; Eriksson, L. Wheel loader operation—Optimal control compared to real drive experience. *Control Eng. Pract.* **2016**, *48*, 1–9. [[CrossRef](#)]
56. Edson, C.P.; Yao, J.; Zhao, G.; Sun, Z. Joint Optimization for Transport and Bucket Loading Phases of Automated Wheel Loaders. *IEEE/ASME Trans. Mechatronics* **2023**, *28*, 2047–2055. [[CrossRef](#)]
57. Nezhadali, V.; Eriksson, L. Wheel loader optimal transients in the short loading cycle. *Int. Fed. Autom. Control Proc. Vol. (IFAC)* **2014**, *47*, 7917–7922. [[CrossRef](#)]

58. Haarnoja, T.; Zhou, A.; Abbeel, P.; Levine, S. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. *arXiv* **2018**, arXiv:1801.01290. Available online: <http://arxiv.org/abs/1801.01290> (accessed on 9 September 2023).
59. Commission, E.; Centre, J.R.; Martínez-Plumed, F.; Caballero, F.; Castellano-Falcón, D.; Fernández-Llorca, D.; Gómez, E.; Hupont-Torres, I.; Merino, L.; Monserrat, C.; et al. *AI Watch, Revisiting Technology Readiness Levels for Relevant Artificial Intelligence Technologies*; Publications Office of the European Union: Brussels, Belgium, 2022; ISBN 978-92-76-52328-4. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.