

Article

Mining Trajectory Planning of Unmanned Excavator Based on Machine Learning

Zhong Jin ^{1,2}, Mingde Gong ^{1,*}, Dingxuan Zhao ¹, Shaomeng Luo ², Guowang Li ², Jiaheng Li ², Yue Zhang ¹ and Wenbin Liu ¹

¹ School of Mechanical Engineering, Yanshan University, Qinhuangdao 066004, China; jinzhong@liugong.com (Z.J.); zdx@ysu.edu.cn (D.Z.); zhangyue123@stumail.ysu.edu.cn (Y.Z.); ysliuwb@163.com (W.L.)

² Guangxi Liugong Machinery Co., Ltd., Liuzhou 545007, China; luoshaomeng@liugong.com (S.L.); liguowang@liugong.com (G.L.); lijiaheng@liugong.com (J.L.)

* Correspondence: gmd@ysu.edu.cn

Abstract: Trajectory planning plays a crucial role in achieving unmanned excavator operations. The quality of trajectory planning results heavily relies on the level of rules extracted from objects such as scenes and optimization objectives, using traditional theoretical methods. To address this issue, this study focuses on professional operators and employs machine learning methods for job trajectory planning, thereby obtaining planned trajectories which exhibit excellent characteristics similar to those of professional operators. Under typical working conditions, data collection and analysis are conducted on the job trajectories of professional operators, with key points being extracted. Machine learning is then utilized to train models under different parameters in order to obtain the optimal model. To ensure sufficient samples for machine learning training, the bootstrap method is employed to adequately expand the sample size. Compared with the traditional spline curve method, the trajectories generated by machine learning models reduce the maximum speeds of excavator boom arm, dipper stick, bucket, and swing joint by 8.64 deg/s, 10.24 deg/s, 18.33 deg/s, and 1.6 deg/s, respectively; moreover, the error does not exceed 2.99 deg when compared with curves drawn by professional operators; and, finally, the trajectories generated by this model are continuously differentiable without position or velocity discontinuities, and their overall performance surpasses that of those generated by the traditional spline curve method. This paper proposes a trajectory generation method that combines excellent operators with machine learning and establishes a machine learning-based trajectory-planning model that eliminates the need for manually establishing complex rules. It is applicable to motion path planning in various working conditions of unmanned excavators.

Keywords: unmanned excavator; spline curve; bootstrap; machine learning; trajectory planning

MSC: 00A06; 68T07



Citation: Jin, Z.; Gong, M.; Zhao, D.; Luo, S.; Li, G.; Li, J.; Zhang, Y.; Liu, W. Mining Trajectory Planning of Unmanned Excavator Based on Machine Learning. *Mathematics* **2024**, *12*, 1298. <https://doi.org/10.3390/math12091298>

Academic Editor: Ivan Lorencin

Received: 23 March 2024

Revised: 19 April 2024

Accepted: 20 April 2024

Published: 25 April 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

Trajectory planning is a crucial technology in unmanned excavators, and a well-designed motion trajectory significantly influences the control effectiveness of these machines. Current research on trajectory generation primarily focuses on three aspects: (1) generating trajectories using high-order curves based on physical constraint points and other constraints; (2) fitting artificial operation curves, with a particular emphasis on bucket tip position trajectory fitting; and (3) generating trajectories based on force and torque feedback, with an emphasis on excavation action trajectory planning. YOO et al. [1] proposed generating motion trajectories using B-spline curves constrained by the maximum pump flow rate. ZOU et al. [2] also utilized B-spline curves combined with constraint conditions to generate three types of excavation trajectories: shortest time, minimum energy, and minimum operating force. JUD et al. [3] achieved results similar to those of operators through end-effector

force-torque path planning and later adopted cubic Hermite spline curves for continuous excavation path planning [4]. KIM et al. [5] designed two different operational paths based on the shortest operating time and the minimum torque, respectively. GROLL et al. [6] divided the operational motion into segments and established sub-modules for each stage to form a complete trenching process. LEE et al. [7] divided tasks into three stages, completed path planning under global planner, and utilized model-predictive control (MPC) to update local path information in real-time. ZHANG et al. [8] added acceleration jerk constraints and used a sequential quadratic programming algorithm to calculate optimal decision-making solutions; ZHAO et al. [9] established topologically equivalent paths to ensure smooth excavation processes; and FENG H. et al. [10] decomposed trenching operations.

Due to its exceptional adaptability to nonlinear processes, machine learning has been extensively employed in the domain of unmanned excavators. PARK et al. [11] utilized echo-state networks for generating excavator models. ZHOU et al. [12] and OGUMA et al. [13] employed neural networks for diagnosing excavator faults, and the mechanical performance of excavators was predicted using a similar approach by Seker et al. [14]. KURINOV et al. [15] accomplished the automatic loading of excavators through reinforcement learning within a simulation environment, and the study conducted by Samtani et al. [16] also employed reinforcement learning techniques to develop Dueling Double Deep-Q Networks and extensively investigated the fracturing actions of excavators. Through data-driven approaches, task planning research was conducted by ZHAO et al. [9], and inverse motion control of excavator models was conducted by Lee et al. [17]. EGLI et al. [18] trained models using reinforcement learning techniques to accurately identify soil hardness, while YU et al. [19] integrated physical and data-driven models for real-time soil resistance prediction. Li et al. [20] also employed a hybrid approach that integrates kinematic modeling and machine learning techniques to accurately predict resistance. GUO et al. [21], on the other hand, simulated operator's operating patterns using deep neural networks. By leveraging the well-established applications of image analysis in deep learning, OSA T et al. [22] investigated mining tasks through deep image learning techniques, while Kim et al. [23] developed a visual-based model for recognizing mining actions in job statistics. On the other hand, YAO et team [24] generated continuous excavation trajectories employing PINN. Furthermore, ZHAO et al. [25] also discovered that, under optimally planned trajectories, the fluctuation of excavation angles in unmanned excavators is greater than that observed during manual operations.

Based on the comprehensive research mentioned above, the current challenge in trajectory planning for work devices is as follows: the effectiveness of trajectory planning relies solely on the quality of established planning rules, often limited to fulfilling specific optimal objectives without considering holistic outcomes [26]. Presently, most studies concentrate on singular objectives [27] (e.g., minimizing energy consumption) for job trajectory planning, posing difficulties in balancing multiple objectives (such as minimizing time, reducing energy consumption, and mitigating impact). Trajectory planning that prioritizes specific objectives typically compromises other parameter characteristics and encounters practical implementation challenges.

In response to the aforementioned issues, this study proposes a trajectory planning method that integrates the exceptional operator's operational trajectory with machine learning. During long-term practical operations, skilled operators have developed an operating style intuitive and challenging to precisely describe, which is continuously fine-tuned to achieve a balance among multiple objectives. Therefore, it is imperative to employ machine learning techniques for analyzing the operator's operational trajectory and establishing a model that generates optimal operation trajectories aligning with the overall characteristics while considering multiple objectives. This paper initially describes and models traditional trajectory generation methods, followed by collecting and analyzing professional operator's operational data while expanding the sample size using a bootstrap method [28]. Subsequently, machine learning is applied for model training in order to obtain the final trajectory-planning model. Finally, a comparative analysis of differences

in the trajectory characteristics between machine learning model trajectories, professional operator trajectories, and traditional spline curve-based trajectory planning is conducted. The results demonstrate that, in terms of consistency and continuity, among other aspects, the machine learning model can generate high-quality operating trajectories similar to those of exceptional operators while surpassing conventional trajectory planning methods.

2. Sample Methods

2.1. Traditional Trajectory Generation

Excavator trajectory planning is a crucial part of motion planning for unmanned excavators, and its core lies in accurately planning the movement trajectory of the excavator arm from the starting position to the target position, as well as intermediate positions, based on the task requirements. Typically, we use spline curves for representation. Spline curves possess characteristics such as simplicity, ease of use, continuity, and controllability of start and endpoints, making them highly suitable for generating excavation trajectories [29]. By treating n key points as the segment points and adding appropriate control points to achieve the segmented fitting of joint movements [30,31], corresponding excavation trajectories can be generated.

As an illustration of trajectory generation for dynamic boom joint angles, the break-points are presented below.

$$\begin{aligned}
 T_{boom} &= (t_1, t_2, \dots, t_{n-1}, t_n)^T \\
 \Theta_{boom} &= (\theta_{boom(1)}, \theta_{boom(2)}, \dots, \theta_{boom(n-1)}, \theta_{boom(n)})^T
 \end{aligned}
 \tag{1}$$

Among them, T_{boom} represents the time coordinate points of the trajectory, while Θ_{boom} denotes the angular coordinate points of the trajectory.

The calculation of trajectory control points can be formulated according to the following equation:

$$\begin{aligned}
 T_{boom-c} &= A_{boom} T_{boom} \\
 \Theta_{boom-c} &= B_{boom} T_{boom}
 \end{aligned}
 \tag{2}$$

Among them, A_{boom} and B_{boom} are control point matrix coefficients, with the specific form shown below.

$$A_{boom} = \left(\begin{array}{cccc} \begin{pmatrix} a_{11} & 0 \\ a_{21} & a_{22} \\ a_{31} & a_{32} \\ 0 & a_{42} \end{pmatrix} & (0_{4 \times 2}) & (0_{4 \times 2}) & \dots & (0_{4 \times 2}) \\ (0_{4 \times 2}) & (\dots) & (0_{4 \times 2}) & \dots & (0_{4 \times 2}) \\ (0_{4 \times 2}) & (0_{4 \times 2}) & (\dots) & \dots & (0_{4 \times 2}) \\ \vdots & \vdots & \vdots & (\dots) & \vdots \\ (0_{4 \times 2}) & (0_{4 \times 2}) & (0_{4 \times 2}) & \dots & \begin{pmatrix} a_{(4n-3)(n-1)} & 0 \\ a_{(4n-2)(n-1)} & a_{(4n-2)(n)} \\ a_{(4n-1)(n-1)} & a_{(4n-1)(n)} \\ 0 & a_{(4n)(n)} \end{pmatrix} \end{array} \right)
 \tag{3}$$

$$B_{boom} = \begin{pmatrix} \begin{pmatrix} b_{11} & 0 \\ b_{21} & b_{22} \\ b_{31} & b_{32} \\ 0 & b_{42} \end{pmatrix} & (0_{4 \times 2}) & (0_{4 \times 2}) & \dots & (0_{4 \times 2}) \\ (0_{4 \times 2}) & (\dots) & (0_{4 \times 2}) & \dots & (0_{4 \times 2}) \\ (0_{4 \times 2}) & (0_{4 \times 2}) & (\dots) & \dots & (0_{4 \times 2}) \\ \vdots & \vdots & \vdots & (\dots) & \vdots \\ (0_{4 \times 2}) & (0_{4 \times 2}) & (0_{4 \times 2}) & \dots & \begin{pmatrix} b_{(4n-3)(n-1)} & 0 \\ b_{(4n-2)(n-1)} & b_{(4n-2)(n)} \\ b_{(4n-1)(n-1)} & b_{(4n-1)(n)} \\ 0 & b_{(4n)(n)} \end{pmatrix} \end{pmatrix} \quad (4)$$

Similarly, the control points for trajectory generation of the arm joint, bucket joint, and swing joint can be obtained as illustrated below.

$$\begin{aligned} T_{arm-c} &= C_{arm}(t_1, t_2, \dots, t_n, t_{n-1})^T \\ \Theta_{arm-c} &= D_{arm}(\theta_{arm1}, \theta_{arm2}, \dots, \theta_{arm(n-1)}, \theta_{arm(n)})^T \\ T_{bkt-c} &= E_{bkt}(t_1, t_2, \dots, t_n, t_{n-1})^T \\ \Theta_{bkt-c} &= F_{bkt}(\theta_{bkt1}, \theta_{bkt2}, \dots, \theta_{bkt(n-1)}, \theta_{bkt(n)})^T \\ T_{swg-c} &= G_{swg}(t_1, t_2, \dots, t_n, t_{n-1})^T \\ \Theta_{swg-c} &= H_{swg}(\theta_{swg1}, \theta_{swg2}, \dots, \theta_{swg(n-1)}, \theta_{swg(n)})^T \end{aligned} \quad (5)$$

The coefficient matrix exhibits the same structure as the dynamic arm matrix, and its specific values should be suitably adjusted in accordance with practical circumstances. However, further elaboration on this matter is beyond the scope of this discussion.

The excavation trajectory of each joint can be precisely fitted by employing a segmented cubic Bézier spline curve. The operator’s operation trajectory can be effectively segmented, enabling a seamless fit of all curve types, including straight lines, through the utilization of cubic Bézier curves. Consequently, the expressions for each joint’s trajectories can be standardized as cubic Bézier curves.

The mathematical representation of a third-order Bézier curve is as follows:

$$B(\alpha) = \sum_{i=0}^3 P_i \frac{3!}{i!(3-i)!} (1-\alpha)^{3-i} \alpha^i, 0 \leq \alpha \leq 1 \quad (6)$$

In addition to the two known endpoints, each segment of the trajectory that requires fitting necessitates the inclusion of two additional intermediate points as the control points. As these newly added points do not lie on the trajectory itself, it becomes imperative to adjust the aforementioned control matrix coefficients based on both the trend of trajectory changes and the characteristics inherent to Bézier curves. Once the key points are determined for a given operator excavation trajectory, the corresponding joint trajectories can be generated using the aforementioned control point formula.

2.2. Professional Operator Data Collection and Analysis

The generation of excavation trajectories must consider multiple factors, including power characteristics, workspace limitations, motion time, and path smoothness of the excavators. However, commonly used point-fitting methods currently struggle to comprehensively account for these diverse factors. In practical operations, professional operators often exhibit optimized and efficient characteristics when operating excavators, making them suitable templates for unmanned excavators. In this section, we employ the bootstrap method to expand their excavation trajectories as samples based on data from professional

operators and utilize them as inputs for machine learning algorithms to generate excavation trajectories.

In order to ensure the standardization and reproducibility of professional operators' actions, this article presents comprehensive specifications for the operators, work scenarios, and objects involved in the operations.

Two individuals engaged in the occupation of excavator-testing operators, with five years of experience in operating excavators. They were 38 years old and male, reaching an expert level in terms of operational skills.

Excavators are standardized batch products, devoid of any modifications or customizations. Only when executing trajectory tracking control, additional computational controller hardware is incorporated and integrated into the primary machine controller. The parameters for each joint can be found in the table provided (Table 1).

Table 1. Relevant parameters of excavator joints.

	θ_{min} (deg)	θ_{max} /deg	θ_{range} /deg	Note
Rotating	0	360	360	Rotate continuously
Boom	−48	65	113	Relative to the water level
Arm	−158	−35	123	Relative to boom
Bucket	−150	49	199	Relative to arm

The task scenario involves excavation and loading on a high platform, with the angles of excavation and unloading being approximately 90 degrees.

The target for the excavation is fine sand characterized by a low water content, absence of compaction, and a uniform particle size. As depicted in Figure 1, the fine sand exhibits a distinct morphology that is closely associated with the positioning of vehicles.



Figure 1. Professional operator job scenario illustration.

To ensure the consistency of the operator data collected during the operations, it is required that the operators refrain from moving the excavator tracks and only perform rotation, excavation, and unloading actions. Operator 1 conducted a series of 20 consecutive excavation and unloading operations before site restoration, followed by Operator 2 performing excavation operations under identical conditions. The real-time collection of operational trajectory data was carried out for both operators. After eliminating significant differences in the data curves, a final set of 19 valid excavation data points were obtained for Operator 1, while Operator 2 yielded 12 valid sets.

2.3. Sample Expansion

A substantial volume of raw data is imperative for attaining optimal training outcomes in machine learning. However, amassing an extensive dataset of operator operation curves proves to be neither cost-effective nor practical. Hence, a viable approach to tackle this issue lies in sample augmentation. Currently, there exist several methods for augmenting small samples, including the random sequence method, the recursive interpolation method, the offset method, the SMOTE method, the Monte Carlo method, and the bootstrap resampling method. Among these techniques, except for bootstrap resampling, other sample augmentation methods have certain limitations, such as the requirement to determine the distribution characteristics of the samples. However, unlike other approaches, bootstrap resampling demonstrates a broader applicability as it does not necessitate specific distribution characteristics of the samples.

The principle of bootstrap capacity expansion is briefly outlined as follows [32].

Firstly, arrange the initial samples in a sorted manner and subsequently generate regenerated samples using the following formula:

$$p_i = p_i^* + ((n-1)\delta - \lfloor (n-1)\delta \rfloor)(p_{i+1}^* - p_i^*) \quad (7)$$

Among them,

δ is uniformly distributed random numbers in $[0, 1]$;

p_i^* is the i -th original sample value.

After undergoing standardization processing using Matlab, this method has been encapsulated into the `bootstrp()` function for direct implementation. The function exclusively supports the expansion of one-dimensional samples, necessitating the conversion of the operator's two-dimensional data into a one-dimensional format. After analyzing the operator's operation curve, we have determined that utilizing two parameters, namely, task cycle and joint angle range, would effectively reduce the data dimensions. Utilizing the reduced dataset, we independently expand both the task cycle and joint angle range. Subsequently, new sample curves are generated by applying the expanded data to the original curve.

The dataset of job cycle samples and joint angle amplitude after expansion is shown below (Figure 2).

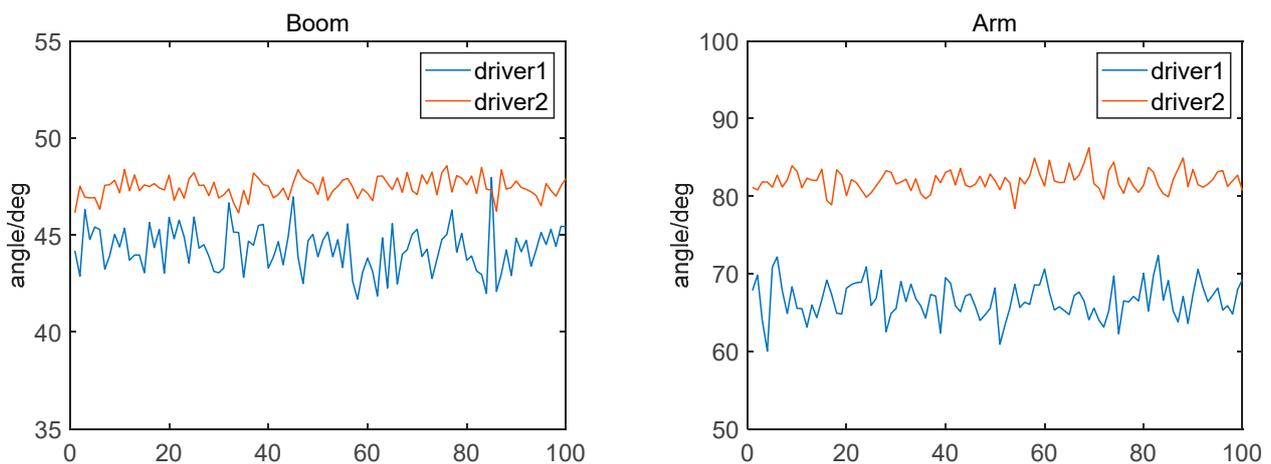


Figure 2. Cont.

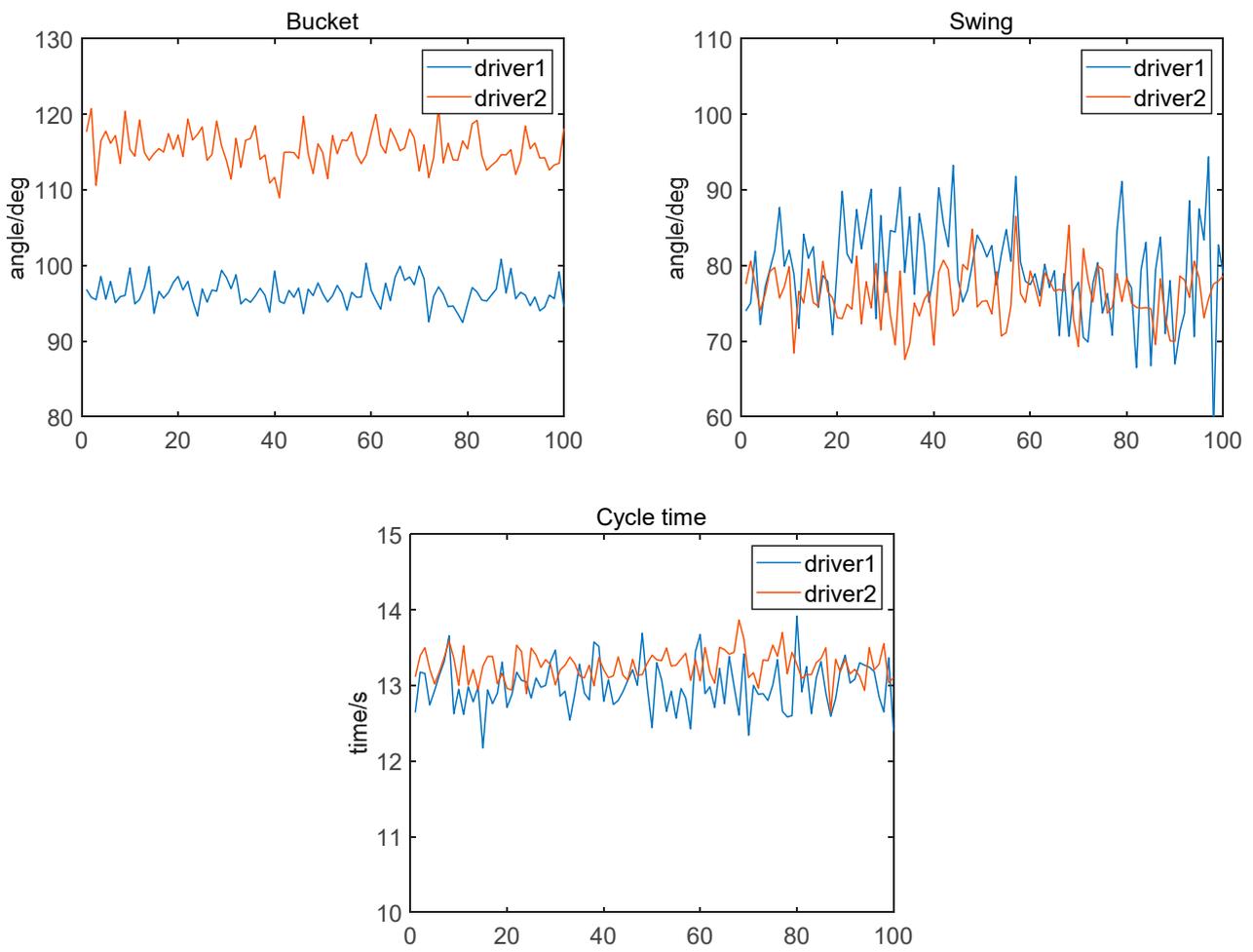


Figure 2. Post-expansion job cycle samples and joint angle amplitude samples.

The new sample curve, depicted in the figure below, was obtained by combining expanded samples with the original samples (Figure 3).

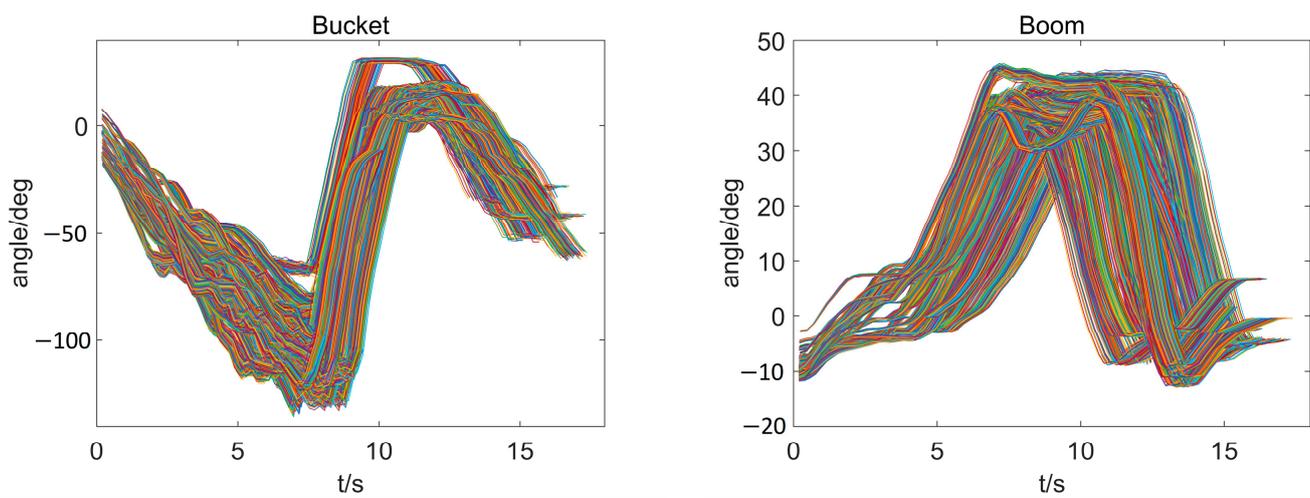


Figure 3. Cont.

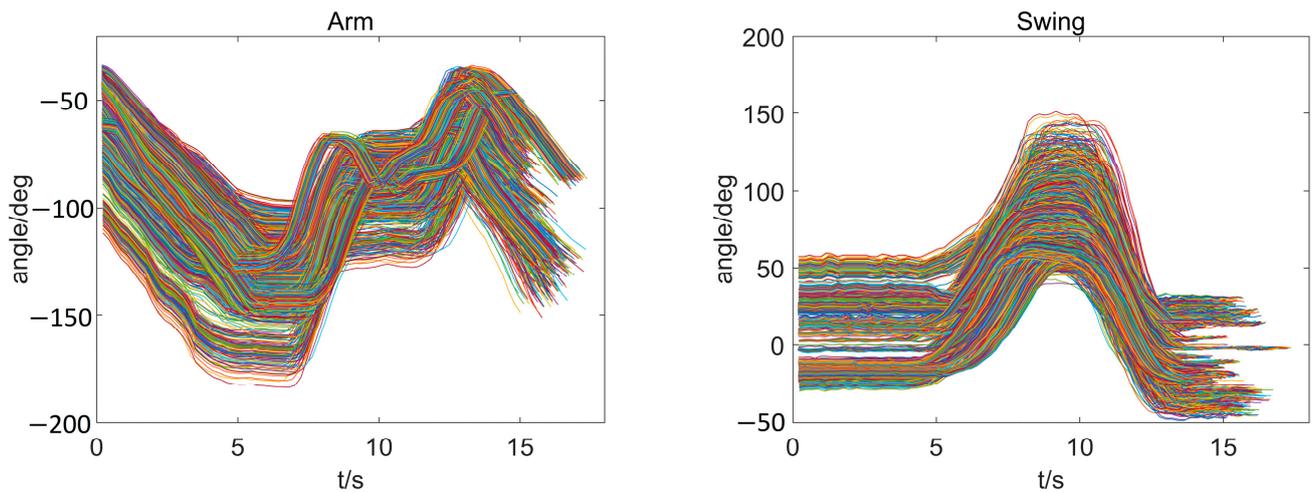


Figure 3. Four expanded joint trajectory curves (The colors of the 3100 lines were randomly assigned).

2.4. Generate Mining Trajectories

Machine learning is currently an increasingly mature solution. Several scholars have already conducted research on motion trajectory generation using machine learning techniques [15,16,20,23,33,34]. However, this paper distinguishes itself by employing bootstrap resampling [32] to augment the operator’s limited sample data, resulting in relatively conservative yet diverse samples. Subsequently, training is carried out based on key points to generate the final trajectory model.

2.4.1. Key Point Selection

The coordinate system of an excavator is defined as illustrated in the diagram. The machine coordinate system (x_0, y_0, z_0) has its origin located at the pivot center and at the same height as the lower hinge point of the boom. Within this machine coordinate system, distinct coordinate systems are established for the boom (x_1, y_1, z_1) , arm (x_2, y_2, z_2) , and bucket (x_3, y_3, z_3) . The angles of the boom, arm, and bucket are all relative values that increase in a counterclockwise direction. The swing angle is determined based on the current posture, with positive increments considered when rotating clockwise (Figure 4).

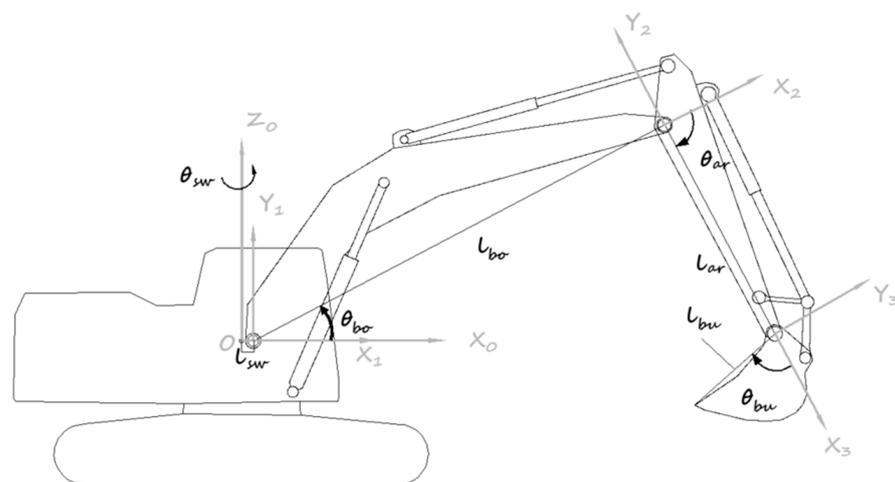


Figure 4. Definition of excavator coordinate system.

By analyzing the spatial motion trajectory of the excavator bucket tip in the machine coordinate system, as well as the individual motion trajectories of its four joints, we have chosen to focus our research on the most representative curves of these four joints. The

accompanying diagram presents contour plots depicting these joint trajectories during operations conducted by two skilled operators. From these motion trajectory curves, it can be observed that both operators' work paths exhibit overall similarity while also displaying distinct personal characteristics. For instance, in terms of arm trajectory, Operator 1 demonstrates a comparatively slower lifting speed compared to Operator 2; however, their boom lowering speed is relatively faster. In contrast, Operator 2 exhibits a higher level of consistency when reaching maximum angles—a quality noticeably absent in Operator 1's curve (Figure 5).

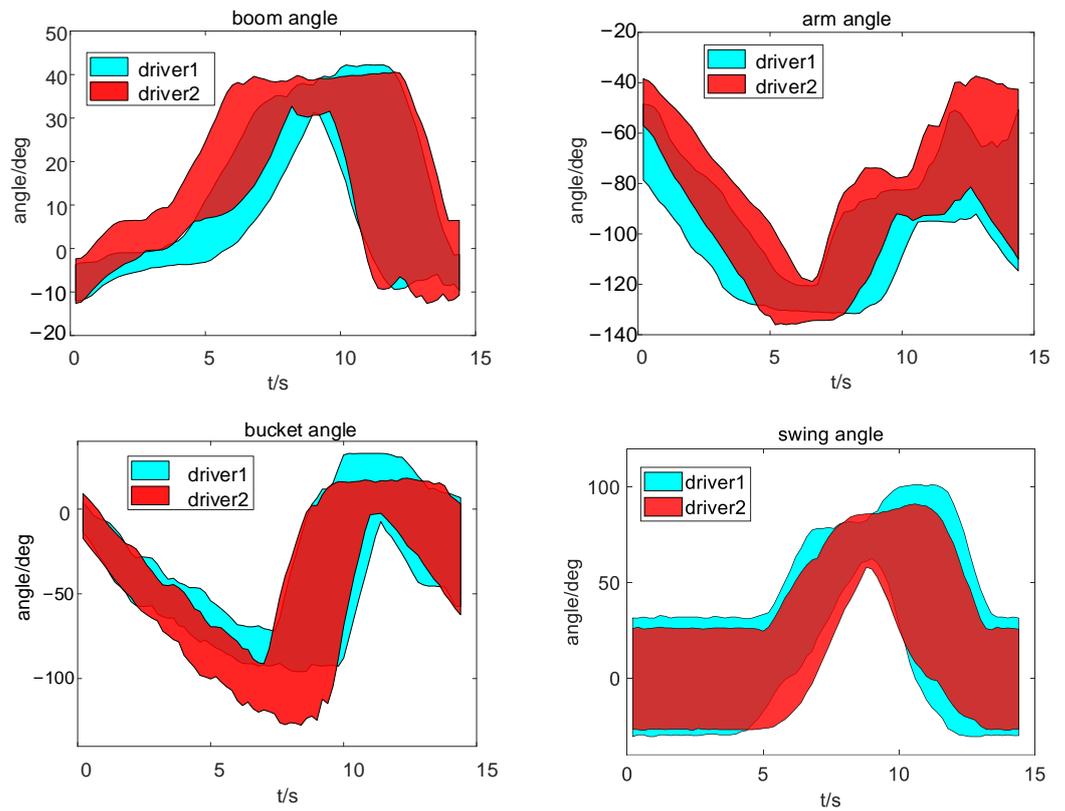


Figure 5. Proficient operator maneuvering curve contours.

During excavation operations, the operator must precisely execute six key points while allowing for flexible adjustments in intermediate steps as needed. The sequence of these fixed key points includes the starting and ending excavation points, two obstacle avoidance points, an unloading point, and the next excavation point. Although minor differences may occur due to safety space considerations made by the operator during each operation cycle, these six key points are relatively consistent. The diagram below illustrates their positions (Figure 6).



Figure 6. Cont.



Figure 6. Diagram of six key point positions.

Based on the curves of the four joint angles, the positions of the six key points can be determined, as illustrated in the diagram provided below (Figure 7).

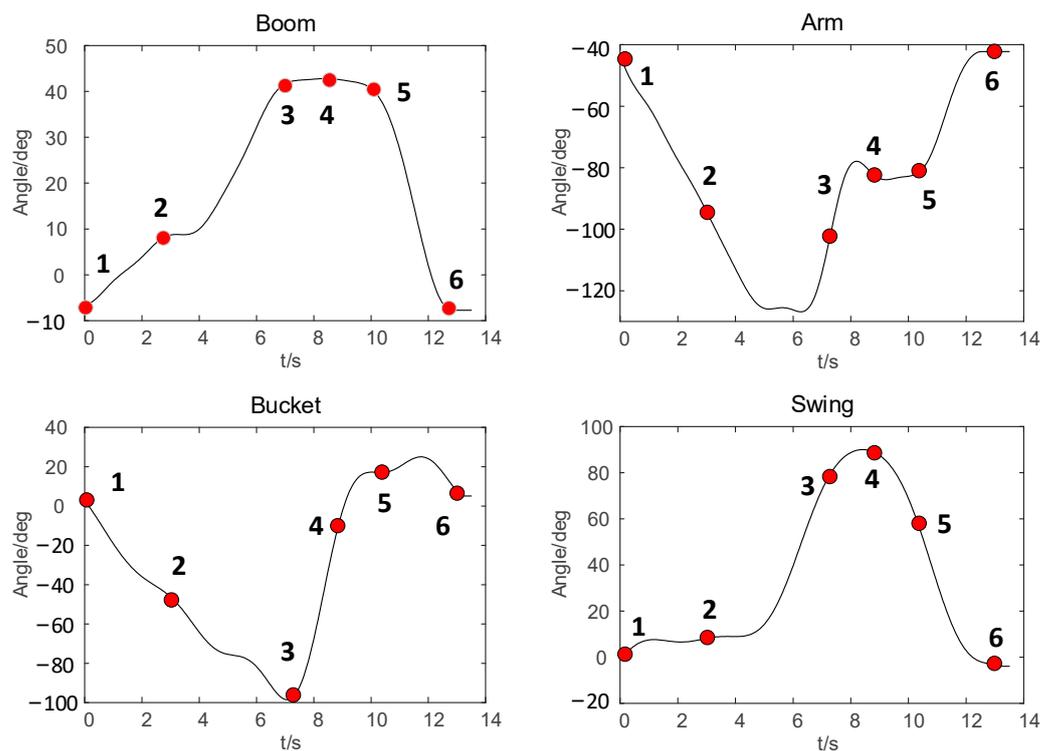


Figure 7. The key point lies in the position diagram of joint trajectories.

The spatial coordinates of each key point can be acquired through perception or pre-determined methods and are presented as known conditions in this paper. The four joint angle values corresponding to the coordinates of each key point can be obtained through robot inverse kinematics solving, which will not be further expounded upon within the scope of this paper.

The temporal coordinates of the key points are also crucial factors in acquiring a reasonable trajectory; however, these temporal coordinates cannot be directly obtained.

By observing the operational curves of two operators, certain principles can be inferred to serve as the foundation for calculating the corresponding temporal coordinates of each key point.

Guideline 1: The excavator arm should retract at a consistent speed from the excavation point to the endpoint.

Guideline 2: Upon reaching the endpoint, the boom should briefly pause before lifting uniformly towards obstacle avoidance point 1.

Guideline 3: The bucket should open evenly from obstacle avoidance point 1 to the unloading point.

Guideline 4: From the unloading point to obstacle avoidance point 2, the excavator should rotate at a relatively balanced pace.

Guideline 5: The boom should descend at a fixed rate from the obstacle avoidance point to the next digging location.

The time coordinates of six key points can be obtained by setting appropriate parameters for the joint motion speed, based on the aforementioned five rules. The calculation formula is presented below.

$$\begin{aligned}
 t_1 &= 0 \\
 t_2 &= t_1 + \frac{\theta_{arm2} - \theta_{arm1}}{v_{armset}} \\
 t_3 &= t_2 + \frac{\theta_{boom3} - \theta_{boom2}}{v_{boomset1}} + t_{boomhalt} \\
 t_4 &= t_3 + \frac{\theta_{bkt4} - \theta_{bkt3}}{v_{bktset}} \\
 t_5 &= t_4 + \frac{\theta_{swg5} - \theta_{swg4}}{v_{swgset}} \\
 t_6 &= t_5 + \frac{\theta_{boom6} - \theta_{boom5}}{v_{boomset2}}
 \end{aligned} \tag{8}$$

2.4.2. Dataset Construction

By utilizing time, arm, boom, bucket, and swing angle values corresponding to six key points as the feature inputs for machine learning, we represent them as a one-dimensional vector of size 1×30 . To establish the model labels, we uniformly sample 72 points from each joint curve of the operator and represent it as a one-dimensional vector of size 1×360 . Following model prediction, the output is presented as a one-dimensional vector of size 1×360 according to Table 2.

Table 2. Machine learning model input, model label, and model output.

	Symbol	Variable	Description	Unit
Model Input	$x_1 \sim x_6$	$t_{x1} \sim t_{x6}$	Time of key point	s
	$x_7 \sim x_{12}$	$\theta_{xbo1} \sim \theta_{xbo6}$	Angle values of the pivotal boom	deg
	$x_{13} \sim x_{18}$	$\theta_{xar1} \sim \theta_{xar6}$	Angle values of the pivotal arm	deg
	$x_{19} \sim x_{24}$	$\theta_{xbu1} \sim \theta_{xbu6}$	Angle values of the pivotal bucket	deg
	$x_{25} \sim x_{30}$	$\theta_{xsw1} \sim \theta_{xsw6}$	Angle values of the pivotal swing	deg
Model Label	$y_1 \sim y_{72}$	$t_1 \sim t_{72}$	Time of the operator curve	s
	$y_{73} \sim y_{144}$	$\theta_{bo1} \sim \theta_{bo72}$	Boom angle of the operator curve	deg
	$y_{145} \sim y_{216}$	$\theta_{ar1} \sim \theta_{ar72}$	Arm angle of the operator curve	deg
	$y_{217} \sim y_{288}$	$\theta_{bu1} \sim \theta_{bu72}$	Bucket angle of the operator curve	deg
	$y_{289} \sim y_{360}$	$\theta_{sw1} \sim \theta_{sw72}$	Swing angle of the operator curve	deg
Model Output	$\hat{y}_1 \sim \hat{y}_{72}$	$\hat{t}_1 \sim \hat{t}_{72}$	Time of the deep learn model	s
	$\hat{y}_{73} \sim \hat{y}_{144}$	$\hat{\theta}_{bo1} \sim \hat{\theta}_{bo72}$	Boom angle of the deep learn model	deg
	$\hat{y}_{145} \sim \hat{y}_{216}$	$\hat{\theta}_{ar1} \sim \hat{\theta}_{ar72}$	Arm angle of the deep learn model	deg
	$\hat{y}_{217} \sim \hat{y}_{288}$	$\hat{\theta}_{bu1} \sim \hat{\theta}_{bu72}$	Bucket angle of the deep learn model	deg
	$\hat{y}_{289} \sim \hat{y}_{360}$	$\hat{\theta}_{sw1} \sim \hat{\theta}_{sw72}$	Swing angle of the deep learn model	deg

Here are the formats of the input vector and label vector.

$$\begin{aligned} \text{Input} &= [t_{x1} \cdots t_{x6} \quad \theta_{xbo1} \sim \theta_{xbo6} \quad \theta_{xar1} \sim \theta_{xar6} \quad \theta_{xbu1} \sim \theta_{xbu6} \quad \theta_{xsw1} \sim \theta_{xsw6}] \\ \text{Label} &= [t_1 \cdots t_{72} \quad \theta_{bo1} \sim \theta_{bo72} \quad \theta_{ar1} \sim \theta_{ar72} \quad \theta_{bu1} \sim \theta_{bu72} \quad \theta_{sw1} \sim \theta_{sw72}] \end{aligned} \tag{9}$$

The 3100 expanded samples will be randomly partitioned into a training set and a test set in an 8:2 ratio for the purposes of model training and performance evaluation.

2.4.3. Model Construction and Evaluation Metrics

The present study employs a fully connected neural network as the fundamental model for trajectory generation. This model comprises an input layer, hidden layers, and an output layer. There are three key parameters that impact the performance of the fully connected model: the total number of layers (i.e., model depth), the number of neurons in each individual fully connected layer (i.e., network width), and activation functions that enhance the nonlinear expression capability. In theory, higher values for network width and model depth yield a more intricate model with an enhanced learning ability. However, this may also lead to overfitting, training challenges, and increased computational burden. Therefore, when determining an appropriate network structure, it is crucial to thoroughly consider the characteristics of the research object. Given the absence of a unified and effective method for determining the optimal network structure presently available, this study will obtain a relatively suitable trajectory generation network model through experimentation.

The present study employs a set of 30 feature values as the inputs, comprising a combination of 5 elements and 6 key points. The input feature vector utilized by the model is depicted in Equation (9).

The output of the hidden layer is illustrated below.

$$\alpha^{(j)} = f(W^{(j)} * \alpha^{(j-1)} + b^{(j)}) \tag{10}$$

Among them, f represents the rectified linear unit (ReLU) activation function, as illustrated below.

$$f(z) = \begin{cases} z & , z > 0 \\ 0 & , z \leq 0 \end{cases} \tag{11}$$

The weight matrix ($W^{(j)}$) and bias vector ($b^{(j)}$) represent the parameters of the j th hidden layer.

By iteratively processing through neural networks, we can derive a control sequence vector comprising 360 data points encompassing time, arm movement, boom angle, bucket position, and rotation angle. The specific details are outlined as follows:

$$Y = [\hat{t}_1 \cdots \hat{t}_{72} \quad \hat{\theta}_{bo1} \sim \hat{\theta}_{bo72} \quad \hat{\theta}_{ar1} \sim \hat{\theta}_{ar72} \quad \hat{\theta}_{bu1} \sim \hat{\theta}_{bu72} \quad \hat{\theta}_{sw1} \sim \hat{\theta}_{sw72}]^T \tag{12}$$

The complete training process of the trajectory generation neural network is shown in the following diagram (Figure 8).

The loss function plays a pivotal role in evaluating the performance of machine learning models as it quantifies the disparity between the predicted and actual outcomes. Commonly employed loss functions encompass the mean squared error, the cross-entropy loss, the log-likelihood loss, and the absolute error. The selection of an appropriate loss function should be guided by specific problem domains and requirements. Consequently, the choice of an adequate loss function is crucial for assessing the efficacy of machine learning models.

$$L_{MSE} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \tag{13}$$

Among them, N represents the number of samples; y_i represents the true value label for the i -th sample; and \hat{y}_i represents the predicted value of the model for the i -th sample.

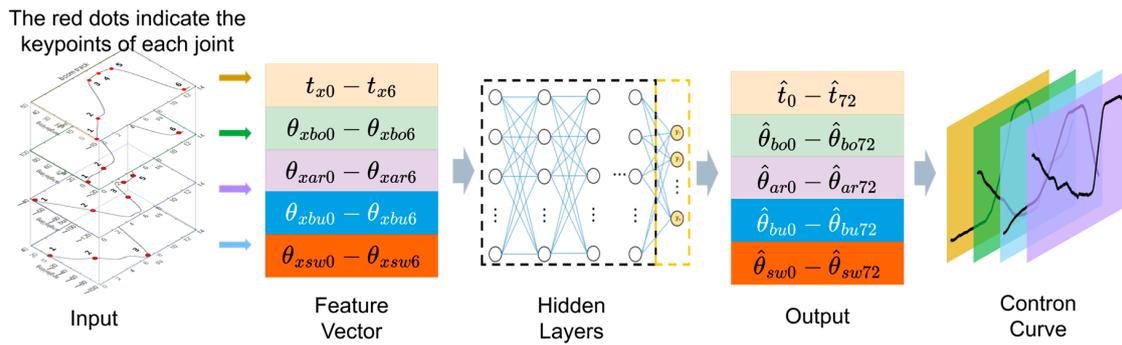


Figure 8. Machine learning for mining trajectory generation.

3. Results and Discussion

The models utilized in this article were tested under a standardized hardware and software environment. The hardware configuration comprised an Intel(R) Xeon(R) CPU E5-2678 v3@2.5GHz and NVIDIA RTX A5000 with a graphics memory capacity of 24 G and a running memory of 16 G. The software platform was based on the Windows 10 operating system, employing the Pytorch 1.2 framework.

3.1. Results

To achieve an optimal model structure, this study trained networks with varying numbers of layers and neurons while keeping the training parameters consistent. The specific outcomes are presented in Table 3.

Table 3. Comparison of neural network models with different parameters.

No.	Network Layers	Number of Neurons per Layer	Parameter Quantity	Training Parameters
1	5	256	297,832	Training epochs: 15,000 Optimizer: Adam Learning rate: 0.001 Learning rate strategy: Exponential decay Loss function adjustment: Mean squared error
2	10	256	626,792	
3	15	256	955,752	
4	5	512	988,520	
5	10	512	2,301,800	
6	15	512	3,615,080	

As depicted in the figure below, the training loss curves exhibit variations across different network architectures. Upon careful examination of the chart, it becomes evident that a model comprising 5 layers and 256 neurons per layer demonstrates rapid convergence and showcases a performance consistency comparable to more intricate models on the test set (Figure 9).

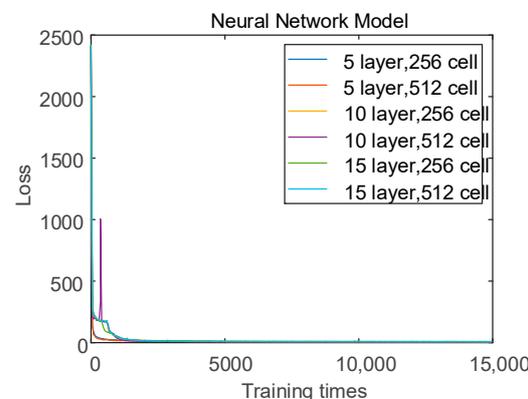


Figure 9. Machine learning loss function curve.

The set of mining curves generated using this model is shown in the following figure (Figure 10).

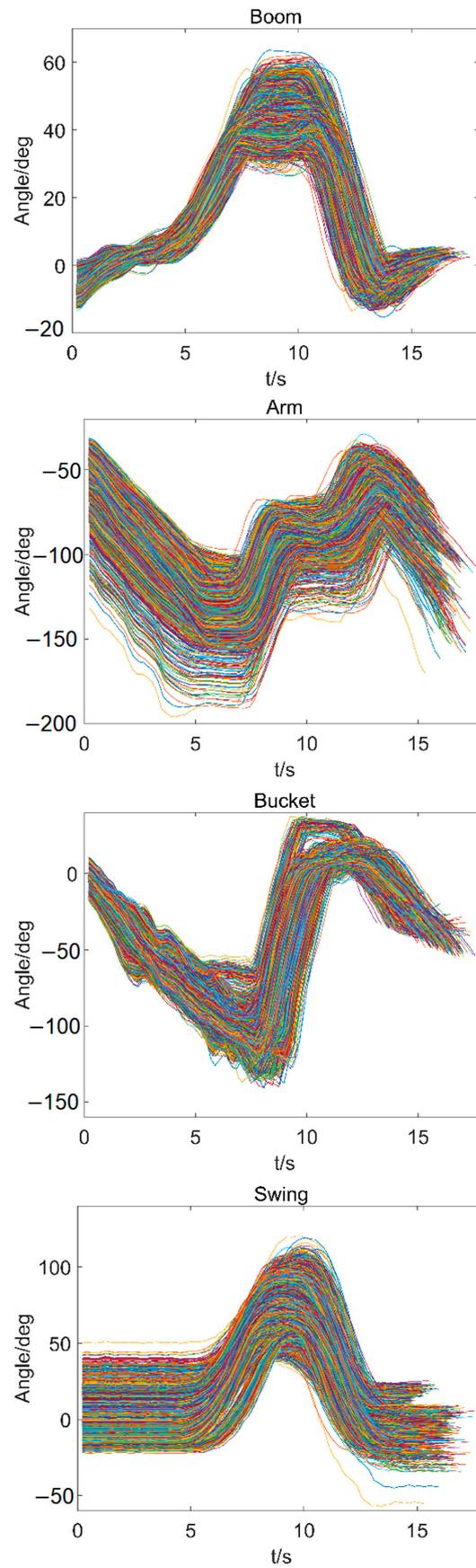


Figure 10. 3100 trajectory curves generated by machine learning models (The colors of the 3100 lines were randomly assigned).

3.2. Discussion

We conducted a comparative analysis by utilizing pre-established reference test curves. We randomly selected one of the curves and extracted its key points, followed by employing a machine learning model (abbreviated as MLM in the image) and a spline curve model to generate mining trajectories. The resulting trajectory curves are depicted in the figure below. Utilizing the operator’s operational curve as a baseline, we calculated the error values between the spline curve model and the machine learning model for comparative purposes (Figure 11).

Both the spline curve model and the machine learning model can generate mining trajectories that are in good overall agreement with the operator’s operating curves. The maximum deviation between the two methods is presented in the table below, indicating that the machine learning model exhibit a superior performance compared to the spline curves (Table 4).

Table 4. The maximum deviation of the joint angles between the spline curve model and the machine learning model.

	θ_{boom}/deg	θ_{arm}/deg	θ_{bucket}/deg	θ_{swing}/deg
Spline curve model	4.22	4.97	−3.2	−4.98
Machine learning model	1.74	−2.99	−2.4	−2.6

The speed comparison between the operator-operated trajectory, the machine learning model-generated trajectory, and the spline curve model-generated trajectory is illustrated in the subsequent figure (Figure 12).

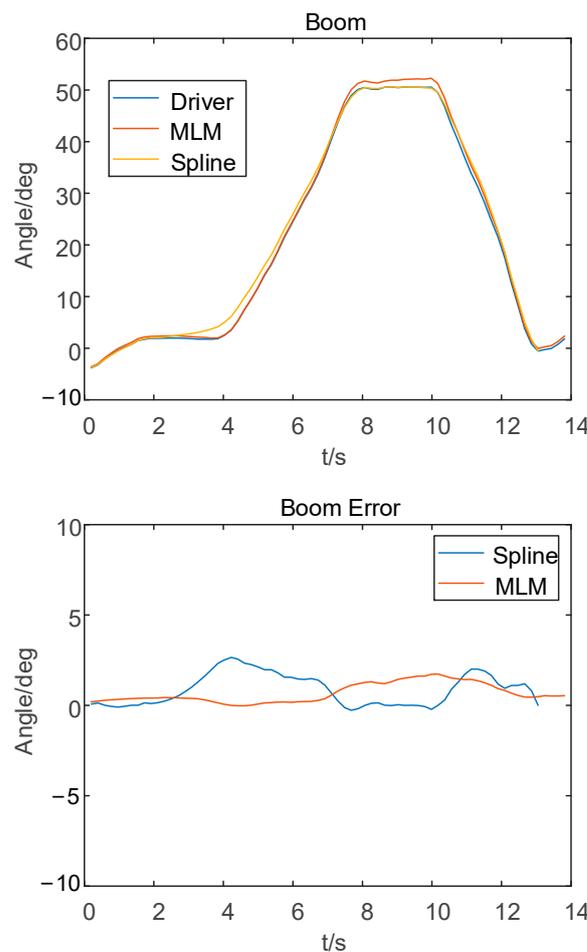


Figure 11. Cont.

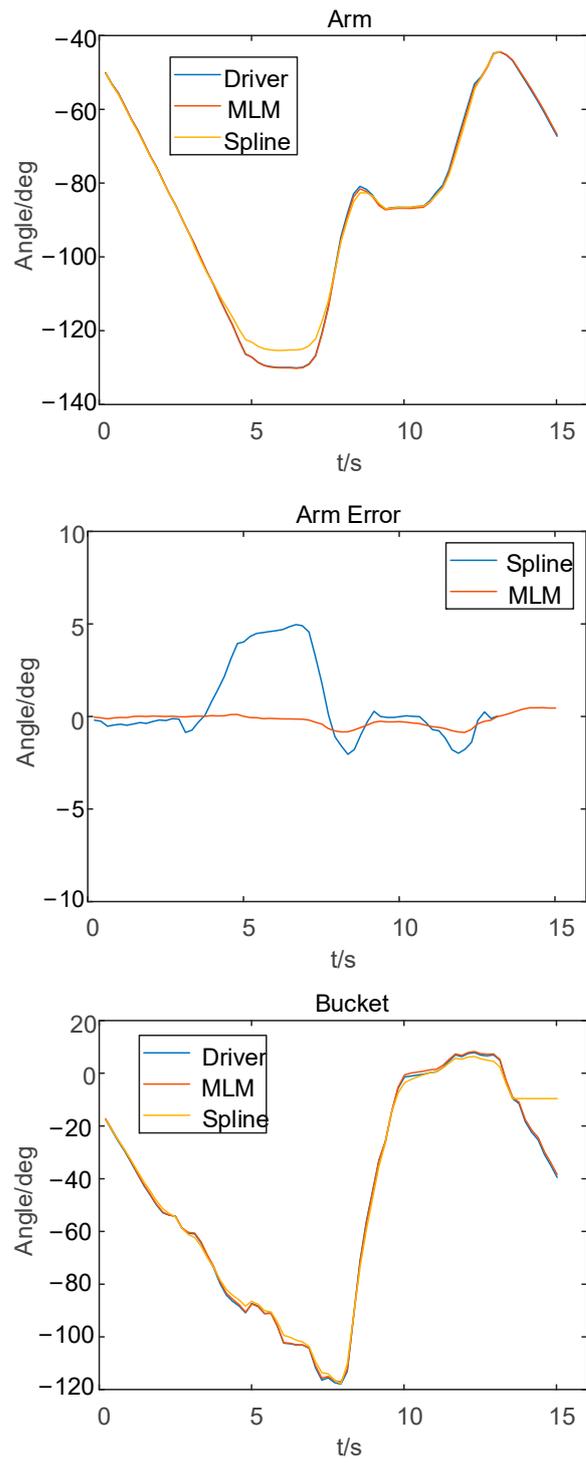


Figure 11. Cont.

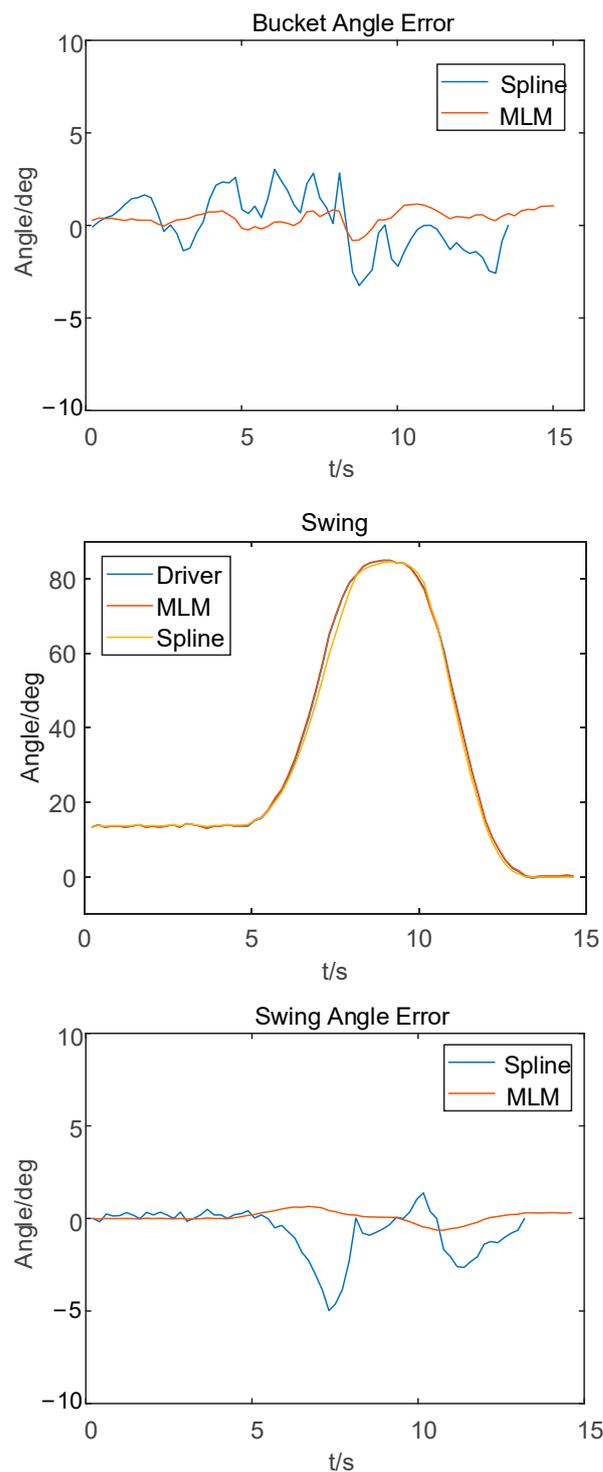


Figure 11. Comparison between operator trajectory, spline curve model trajectory, and machine learning model trajectory.

Based on the aforementioned comparison, it is evident that all three models exhibit a consistent trend in terms of changes in their movement speeds. In contrast to the operator-controlled trajectories, both the machine learning model and the spline curve model demonstrate a decrease in the maximum speed; however, the machine learning model experiences a more pronounced decline. This suggests that the control system enables target trajectory tracking for the machine learning model with reduced energy consumption, thereby highlighting its superiority over the control system (Table 5).

Table 5. Comparison of maximum angular velocities for three trajectories.

	$\omega_{\max \text{ boom}}$ deg/s	$\omega_{\max \text{ arm}}$ deg/s	$\omega_{\max \text{ bucket}}$ deg/s	$\omega_{\max \text{ swing}}$ deg/s
Operator’s trajectory	−33.89	47.58	90.58	−41.38
Spline curve model	−23.51	39.80	90.23	−43.27
Machine learning model	−25.25	37.34	72.25	−39.78

Based on the table presented above, it can be inferred that the mining trajectory generated by the machine learning model demonstrates a decrease in the maximum angular velocities of the boom, arm, bucket, and swing joint by 8.64 deg/s, 10.24 deg/s, 18.33 deg/s, and 1.6 deg/s, respectively. This outcome surpasses that of the spline curve model.

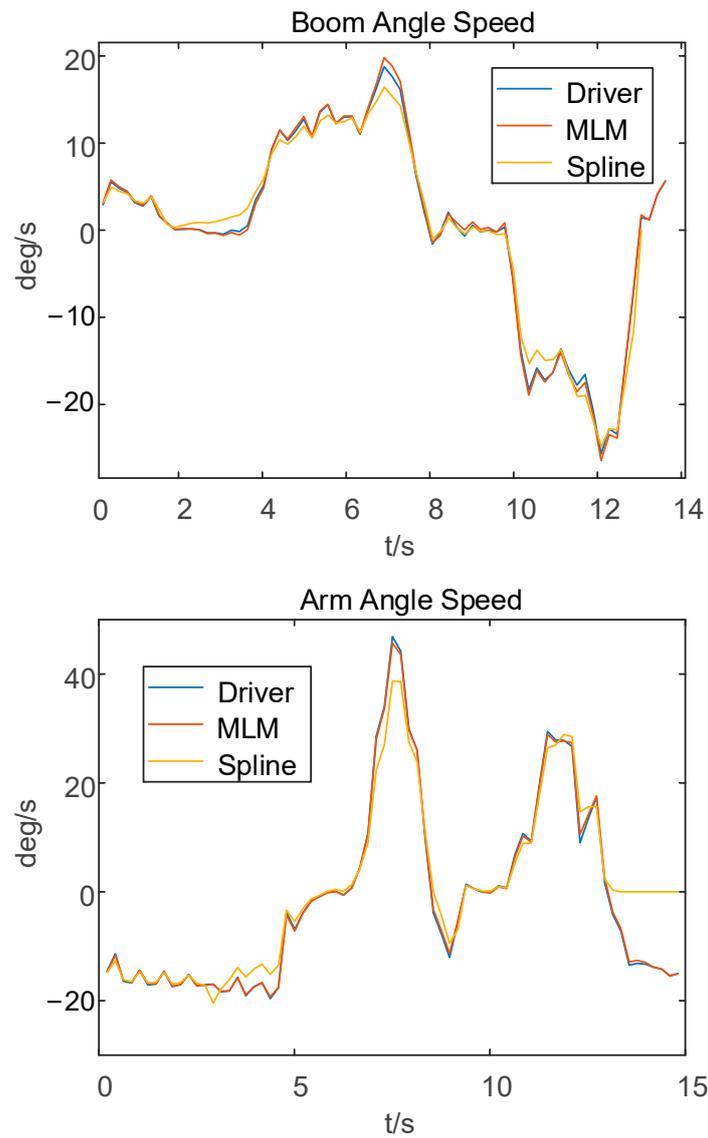


Figure 12. Cont.

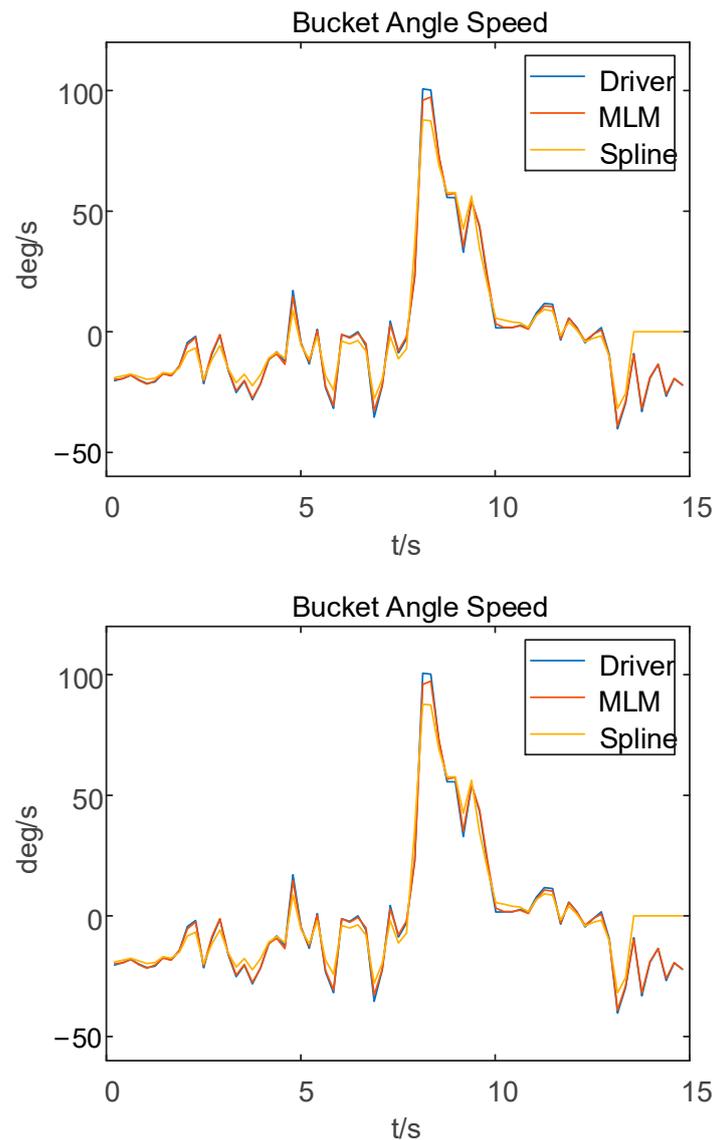


Figure 12. Comparison of angular velocities between manual operation trajectory, spline curve model trajectory, and machine learning model trajectory.

When employing conventional approaches for trajectory generation, it is imperative to meticulously examine and formulate appropriate rules for trajectory planning. The computational resources required for trajectory generation calculations are generally minimal and do not necessitate a machine learning training phase. Hence, it becomes essential to compare the cost of machine learning with traditional trajectory generation methods. We generated 3100 planned trajectories on the same computer and conducted a comparative analysis between the two methods in terms of memory usage, computation time, and training time to assess their computational costs. The traditional method utilized MATLAB tools for generating the trajectories. The specific outcomes are presented in the table below (Table 6).

Table 6. Comparison of costs between traditional methods and ML models.

	Memory Usage	Computation Time	Training Time
Spline curve model	768 Kb	1.5 s	0 s
Machine learning model	1014 Kb	0.14 s	0.231 s

After conducting a thorough comparison, it was observed that traditional methods exhibit advantages in terms of memory utilization and training duration. However, when it comes to generating a substantial number of trajectories, their computational time experiences a significant prolongation. Conversely, machine learning models demonstrate an evident superiority in generating an extensive volume of trajectories, thereby implying acceptable levels of memory usage and training time.

4. Conclusions

After collecting data from professional operators conducting high-platform excavation operations with excavators, we constructed a preliminary sample of operator operation curves and expanded the sample using bootstrap resampling techniques. Key features were extracted as the inputs, and machine learning methods were employed to train the samples. Consequently, we successfully established an excavation trajectory generation model and validated it as an exceptional method for generating trajectories. By comparing the trajectories generated by the machine learning model with those produced by spline curve models and the actual operator trajectories in terms of angles and angular velocities, we observed that the accuracy of the machine learning-generated trajectories significantly surpassed that of the spline curve models. Furthermore, when examining speed variations among the machine learning model trajectories, operator trajectories, and spline curve trajectories, our findings revealed that the speed changes generated by the machine learning model were smoother and exhibited superior adaptability within the control systems. In the future, we will further investigate two aspects. Firstly, we aim to eliminate the need for manually specifying key points in the trajectory and instead analyze professional operators' operational trajectories directly using deep learning methods. This approach aims to establish a trajectory planning method that is independent of subjective factors introduced by researchers, thereby forming a rule-free trajectory planning approach. Secondly, we will conduct machine learning research on additional working conditions based on professional operators' data and develop machine learning-based trajectory planning methods for the automatic recognition and extraction of typical working conditions such as excavation, ground leveling, slope repair, and trenching.

Author Contributions: Methodology, Z.J., M.G. and J.L.; Software, S.L.; Validation, G.L.; Formal analysis, Z.J.; Resources, D.Z.; Data curation, Z.J.; Writing—original draft, Z.J.; Writing—review & editing, Y.Z. and W.L.; Supervision, M.G. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Natural Science Foundation of Hebei Province, China, under Grant No. E2021203145, the National Natural Science Foundation of China under Grant No. 52175063 for the research on active vibration control theory and vehicle attitude control strategy of emergency rescue vehicles with active suspension systems, and the Joint Fund for Regional Innovation Development of National Natural Science Foundation of China under Grant No. U20A20332 for the study on the dynamic control of highly maneuverable emergency rescue vehicles based on multi-information fusion. This work was supported in part by the Special Carrier Equipment Research Center of Yanshan University.

Data Availability Statement: Data are contained within this article.

Acknowledgments: The authors would like to express their thanks to the Natural Science Foundation of Hebei Province, China.

Conflicts of Interest: Authors Zhong Jin, Shaomeng Luo, Guowang Li and Jiaheng Li were employed by the company Guangxi Liugong Machinery Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest. The [company Guangxi Liugong Machinery Co., Ltd.-companies in affiliation and funding] had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Yoo, S.; Park, C.G.; You, S.H. A Dynamics-Based Optimal Trajectory Generation for Controlling an Automated Excavator. *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.* **2010**, *224*, 2109–2119. [\[CrossRef\]](#)
2. Zou, Z.; Chen, J.; Pang, X. Task space-based dynamic trajectory planning for digging process of a hydraulic excavator with the integration of soil–bucket interaction. *Proc. Inst. Mech. Eng. Part K J. Multi-Body Dyn.* **2019**, *233*, 598–616. [\[CrossRef\]](#)
3. Jud, D.; Hottiger, G.; Leemann, P.; Hutter, M. Planning and Control for Autonomous Excavation. *IEEE Robot. Autom. Lett.* **2017**, *2*, 2151–2158. [\[CrossRef\]](#)
4. Jud, D.; Leemann, P.; Kerscher, S.; Hutter, M. Autonomous Free-Form Trenching Using a Walking Excavator. *IEEE Robot. Autom. Lett.* **2019**, *4*, 3208–3215. [\[CrossRef\]](#)
5. Kim, Y.B.; Ha, J.; Kang, H.; Kim, P.Y.; Park, J.; Park, F. Dynamically optimal trajectories for earthmoving excavators. *Autom. Constr.* **2013**, *35*, 568–578. [\[CrossRef\]](#)
6. Groll, T.; Hemer, S.; Ropertz, T.; Berns, K. Autonomous trenching with hierarchically organized primitives. *Autom. Constr.* **2019**, *98*, 214–224. [\[CrossRef\]](#)
7. Lee, D.; Jang, I.; Byun, J.; Seo, H.; Kim, H.J. Real-Time Motion Planning of a Hydraulic Excavator using Trajectory Optimization and Model Predictive Control. *arXiv* **2021**. [\[CrossRef\]](#)
8. Zhang, Y.; Sun, Z.; Sun, Q.; Wang, Y.; Li, X.; Yang, J. Time-jerk optimal trajectory planning of hydraulic robotic excavator. *Adv. Mech. Eng.* **2021**, *13*, 168781402110346. [\[CrossRef\]](#)
9. Zhao, J.; Zhang, L. TaskNet: A Neural Task Planner for Autonomous Excavator. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 2220–2226. [\[CrossRef\]](#)
10. Feng, H.; Jiang, J.; Ding, N.; Shen, F.; Yin, C.; Cao, D.; Li, C.; Liu, T.; Xie, J. Multi-objective time-energy-impact optimization for robotic excavator trajectory planning. *Autom. Constr.* **2023**, *156*, 105094. [\[CrossRef\]](#)
11. Park, J.; Lee, B.; Kang, S.; Kim, P.Y.; Kim, H.J. Online Learning Control of Hydraulic Excavators Based on Echo-State Networks. *IEEE Trans. Autom. Sci. Eng.* **2017**, *14*, 249–259. [\[CrossRef\]](#)
12. Zhou, Q.; Chen, G.; Jiang, W.; Li, K.; Li, K. Automatically Detecting Excavator Anomalies Based on Machine Learning. *Symmetry* **2019**, *11*, 957. [\[CrossRef\]](#)
13. Oguma, S.; Omatsu, S.; Ohno, S.; Iwasaki, K.; Shishido, Y. Failure Prediction Based on Operational Data of Hydraulic Excavator with Machine Learning. *IEEE Trans. Electr. Electron. Eng.* **2021**, *16*, 1438–1440. [\[CrossRef\]](#)
14. Seker, S.E.; Ibrahim, O. Performance Prediction of Roadheaders Using Ensemble Machine Learning Techniques. *Neural Comput. Appl.* **2019**, *31*, 1103–1116. [\[CrossRef\]](#)
15. Kurinov, I.; Orzechowski, G.; Hamalainen, P.; Mikkola, A. Automated Excavator Based on Reinforcement Learning and Multibody System Dynamics. *IEEE Access* **2020**, *8*, 213998–214006. [\[CrossRef\]](#)
16. Samtani, P.; Leiva, F.; Ruiz-Del-Solar, J. Learning to Break Rocks With Deep Reinforcement Learning. *IEEE Robot. Autom. Lett.* **2023**, *8*, 1077–1084. [\[CrossRef\]](#)
17. Lee, M.; Choi, H.; Kim, C.; Moon, J.; Kim, D.; Lee, D. Precision Motion Control of Robotized Industrial Hydraulic Excavators via Data-Driven Model Inversion. *IEEE Robot. Autom. Lett.* **2022**, *7*, 1912–1919. [\[CrossRef\]](#)
18. Egli, P.; Gaschen, D.; Kerscher, S.; Jud, D.; Hutter, M. Soil-Adaptive Excavation Using Reinforcement Learning. *IEEE Robot. Autom. Lett.* **2022**, *7*, 9778–9785. [\[CrossRef\]](#)
19. Yu, S.; Song, X.; Sun, Z. On-Line Prediction of Resistant Force During Soil–Tool Interaction. *Dyn. Syst. Meas. Control.* **2023**, *145*, 081004. [\[CrossRef\]](#)
20. Li, S.; Wang, S.; Chen, X.; Zhou, G.; Wu, B.; Hou, L. Application of Physics-Informed Machine Learning for Excavator Working Resistance Modeling. *Mech. Syst. Signal Process.* **2024**, *209*, 111117. [\[CrossRef\]](#)
21. Guo, Q.; Ye, Z.; Wang, L.; Zhang, L. Imitation Learning and Model Integrated Excavator Trajectory Planning. In Proceedings of the 2022 IEEE RSJ International Conference on Intelligent Robots and Systems (IROS), Kyoto, Japan, 23–27 October 2022; pp. 5737–5743. [\[CrossRef\]](#)
22. Osa, T.; Aizawa, M. Deep Reinforcement Learning With Adversarial Training for Automated Excavation Using Depth Images. *IEEE Access* **2022**, *10*, 4523–4535. [\[CrossRef\]](#)
23. Kim, J.; Seokho, C. Action Recognition of Earthmoving Excavators Based on Sequential Pattern Analysis of Visual Features and Operation Cycles. *Autom. Constr.* **2019**, *104*, 255–264. [\[CrossRef\]](#)
24. Yao, Z.; Zhao, S.; Tan, X.; Wei, W.; Wang, Y. Real-time task-oriented continuous digging trajectory planning for excavator arms. *Autom. Constr.* **2023**, *152*, 104916. [\[CrossRef\]](#)
25. Zhao, J.; Tan, P.; Hu, Y.; Tian, M.; Xia, X. Autonomous Excavation Trajectory Generation for Trenching Tasks Based on Skills of Skillful Operator. In Proceedings of the 2022 International Conference on Mechanical and Electronics Engineering (ICMEE), Xi'an, China, 21–23 November 2022; pp. 134–138. [\[CrossRef\]](#)
26. Yang, Y.; Zhang, L.; Cheng, X.; Pan, J.; Yang, R. Compact Reachability Map for Excavator Motion Planning. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; IEEE: Macau, China, 2019; pp. 2308–2313. [\[CrossRef\]](#)
27. Tremblay, L.F.Y.; Arsenaault, M.; Zeinali, M. Development of a trajectory planning algorithm for a 4-DoF rockbreaker based on hydraulic flow rate limits. *Trans. Can. Soc. Mech. Eng.* **2020**, *44*, 501–510. [\[CrossRef\]](#)

28. Picheny, V.; Kim, N.H.; Haftka, R.T. Application of bootstrap method in conservative estimation of reliability with limited samples. *Struct. Multidiscip. Optim.* **2010**, *41*, 205–217. [[CrossRef](#)]
29. Yang, Y.; Long, P.; Song, X.; Pan, J.; Zhang, L. Optimization-Based Framework for Excavation Trajectory Generation. *IEEE Robot. Autom. Lett.* **2021**, *6*, 1479–1486. [[CrossRef](#)]
30. Zhang, B.; Wang, S.; Liu, Y.; Yang, H. Research on Trajectory Planning and Autodig of Hydraulic Excavator. *Math. Probl. Eng.* **2017**, *2017*, 7139858. [[CrossRef](#)]
31. Yang, J.; Gao, Y.; Guo, R.; Gao, Q.; Zhao, J. Research on Excavator Trajectory Control Based on Hybrid Interpolation. *Sustainability* **2023**, *15*, 6761. [[CrossRef](#)]
32. Ding, H.; Sang, Z.; Li, Z.; Shi, J.; Wang, Y.; Meng, D. Trajectory planning and control of large robotic excavators based on inclination-displacement mapping. *Autom. Constr.* **2024**, *158*, 105209. [[CrossRef](#)]
33. Huh, J.; Bae, J.; Lee, D.; Kwak, J.; Moon, C.; Im, C.; Ko, Y.; Kang, T.K.; Hong, D. Deep Learning-Based Autonomous Excavation: A Bucket-Trajectory Planning Algorithm. *IEEE Access* **2023**, *11*, 38047–38060. [[CrossRef](#)]
34. Egli, P.; Marco, H. A General Approach for the Automation of Hydraulic Excavator Arms Using Reinforcement Learning. *IEEE Robot. Autom. Lett.* **2022**, *7*, 5679–5686. [[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.