IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 22 February 2023, accepted 23 March 2023, date of publication 14 April 2023, date of current version 20 April 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3267120

RESEARCH ARTICLE

Deep Learning-Based Autonomous Excavation: A Bucket-Trajectory Planning Algorithm

JAEMYUNG HUH[1], JANGHO BAE[2], DONGHYEON LEE[1], JAEWON KWAK[1], CHANYOUNG MOON[1], CHULWHAN IM[1], YEONHO KO[1], TAE KOO KANG[3], AND DAEHIE HONG[1]

[1]School of Mechanical Engineering, Korea University, Seoul 02841, Republic of Korea
[2]Department of Mechanical and Automotive Engineering, Kyungsung University, Busan 48434, Republic of Korea
[3]Department of Human Intelligence and Robot Engineering, Sangmyung University, Cheonan 31066, Republic of Korea

Corresponding authors: Daehie Hong (dhhong@korea.ac.kr) and Tae Koo Kang (tkkang@smu.ac.kr)

This work was supported by the Technology Innovation Program (or Industrial Strategic Technology Development Program) (20010776, Development of Collaborated Self Control System and Unmanned Excavator Technology for an Automated Earthworks) funded By the Ministry of Trade, Industry and Energy(MOTIE, Korea).

ABSTRACT Increased safety risks and the difficulties of training excavator operators, combined with manpower shortages, have led to an increased demand for machine automation. This study applies a long short-term memory algorithm to automate a bucket tip-trajectory planning artificial intelligence (AI) system. Unlike other autonomous excavation techniques, our approach performs bucket-trajectory planning without prior knowledge of the nonlinear bucket–soil interaction dynamics during excavation, which require the parameters to be precisely adjusted via heuristic analysis of their correlations. Using data acquired from the excavations of excavator experts, this method uses three-dimensional point cloud of terrain and bucket motion data of excavation process to train and apply the AI modules. Especially, we transform the point cloud, which comprises a massive number of points and entails a considerable computation complexity, into much fewer values, which are enough to represent the three-dimensional shape of the target terrain. To ensure prevention of collisions with underground obstacles along a given excavation path, a collision avoidance algorithm, based on continuous pressure monitoring in the excavator's hydraulic cylinder, is applied. Comparison experiments reveal that the bucket-tip trajectory planning AI system with collision avoidance algorithm generates a traceable trajectory for the machine controller, equipped in an excavator, and yields the desired excavation volume and lead time without collision, regardless of the topographic changes caused by successive excavations.

INDEX TERMS Automated excavation, bucket-tip trajectory planning, collision avoidance algorithm, field robots, long short-term memory, point cloud data.

I. INTRODUCTION

The global demand for excavators has soared owing to their operational versatility, which allows them to be applied to excavation, loading, and drilling tasks. Excavators are essential heavy machinery on construction sites. However, the number of new workers applying to operate such construction equipment has stagnated, owing to the hazardous nature of the job. In particular, the time required to train new personnel up to proficiency on the machine exceeds 5 years. Moreover, the population of skilled operators is declining because of ageing (i.e., retirement). Therefore, the construction industry is suffering from a shortage of workers.

Various studies (from estimation of the soil–bucket interaction force to trajectory tracking and autonomous excavation) have been conducted to address the shortage of personnel. In the field of excavation automation, several studies have investigated the interaction dynamics between the soil and the tool or bucket. For example, some researchers have estimated soil information from the energy dissipated during soil–tool

The associate editor coordinating the review of this manuscript and approving it for publication was Kumaradevan Punithakumar.

interactions, by measuring the force and displacement of the bucket [1]. Dynamic models based on the Newton–Euler equations [2] and soil–tool interactions [3] have been used to predict the resistive forces exerted upon the bucket during excavation. These predicted values were fed to a controller to plan the trajectory of the bucket [4], [5]. Additionally, other studies have attempted to apply fuzzy methods to identify the soil property parameters and fundamental earthmoving equations, to thereby predict the soil resistance forces during excavation [6].

Despite these efforts to mathematically represent soil–tool interactions, the non-linearity of soil dynamics still complicates the bucket's motion control. Moreover, because the interaction dynamics depend on the bucket size and soil type, the controller must be manually tuned to the work site and excavator model. However, such customization is time-consuming. Therefore, recent research into autonomous excavation has focused on the application of machine and deep learning approaches that do not require information regarding the complex dynamics between the soil and bucket motion. Some researchers [7] have attempted to replace existing control methods [e.g., proportional–integral–derivative (PID) control] with a reinforcement learning (RL)-based trajectory tracking controller; this generates valve commands to supply hydraulic power and operate the excavator arm. In [8] and [9], for a given excavator link trajectory, impedance and sliding mode controllers were used to calculate the desired cylinder forces. Subsequently, echo state networks took these forces and cylinder pressure values as inputs to control the servo valves and force the bucket along a set course.

These artificial-intelligence (AI)-based approaches have focused upon the trajectory-tracking performance when setting a trajectory, instead of generating a desired path. However, incorporating expert manipulation for existing tracking methods is a difficult task, which depends on the expert's intuitions regarding the excavation state (rather than calculations of the excavation forces). To automate an excavator, we primarily need a bucket-tip trajectory that incorporates human expertise. Later studies developed AI systems that could imitate the manipulation of experts or generate the total path of the bucket by referring to in-situ data. Reference [10] used neural networks to determine the soil type and design an appropriate path for the bucket. Moreover, some researchers [11] have developed an automatic bucket-filling algorithm for a wheel loader, based on the time-delayed neural network; this uses encoder and pressure transducer measurements to imitate the manipulation of experts in the ''bucket filling'' stage of the excavation process. In [12], taking the topology of the soil surface as input, a heteroscedastic Gaussian process was developed and applied to an electric manipulator for autonomous excavation. The method accurately determined the switching time from the drag to the scoop phase, to obtain the desired excavation volume. In [13], a multi-layer perceptron was used to generate a dataset of shaping forces; these were transformed into the point set of bucket trajectory using the dynamic movement primitives. Furthermore, in [14], deep reinforcement learning (DRL) was used to develop the bucket tip trajectory. Data for training the DRL were obtained from a robotic simulation program. After learning, the DRL of the excavator could output each control action for excavation.

However, the control actions determined by the AI systems in [10], [11], and [12] were extremely restrictive in certain stages of the excavation process, which limits their applicability across different terrain shapes. In the force-data-based trajectory planning in [13], the force configuration during excavation differed between fields with different soil hardness, even when the paths were identical. Moreover, the configuration step requires different tuning to be set (according to the force set) for the transformation into the point's coordinate value. The DRL has been widely developed and applied in robotics, because the computing performance has progressed considerably in terms of memory and computing speed, as reported in [7], [14], [15], [16], and [17]. Generally, DRL involves numerous trial and error operations when training the control policy to output the Q value for each action. Because repetitive trial and error is difficult to perform in practice owing to safety risks, several studies have performed multiple trial and error simulations and stored the data for each excavation in a replay buffer [14], [15]. However, in virtual simulations, the integration of soil dynamics, including the soil–tool interaction and configuration of soil piles and particles, not only requires knowledge of physical soil characteristics but also considerable computational power, depending on the size and number of soil particles. Furthermore, in DRL, the reward function, which yields a reward to the agent depending on the agent's action based on states, requires a heuristic approach for its configuration and must thoroughly consider the relation between the control action and state of excavation.

To address the limitations and difficulties of prior studies, we developed a trajectory planning system for the bucket-tip position and orientation; to this end, we used expert excavation data (composed of the point cloud for the target terrain before excavation and the position values of the bucket tip during excavation). The bucket-tip trajectory planning system consists of two AI approaches. The first generates the region of excavation, considering safety against collisions; the second locates waypoints inside that region. These two AI modules include long short-term memory (LSTM) algorithms as the main layer. LSTM is an advanced recurrent neural network (RNN) and is suitable for training with one-dimensional (1D) long-sequence datasets. For configuration of the 1D input dataset, we transformed the point cloud into three values (representing the terrain features). In addition to the bucket-tip trajectory planning AI system, we developed a collision avoidance algorithm to prevent the bucket from crashing into underground obstacles, and we adjusted its excavation range to ensure safety. Using these algorithms, this study proposes an excavation procedure that is robust

against topographic changes (produced via successive excavations) and safe to operate when underground obstacles are present; this approach does not require knowledge of soil dynamics. To validate the method's robustness, we compare its performance against those of experts, changing the terrain in each trial. The collision avoidance algorithm is validated in a simulated environment.

TABLE 1. Specifications of experimental excavator.

Description	Specification
Scale	1/8
Size (mm)	952 × 295 × 272
Hydraulic pressure of pump	4 MPa
Power (Boom/Arm/Bucket)	Hydraulic
Power (Swing)	Electric

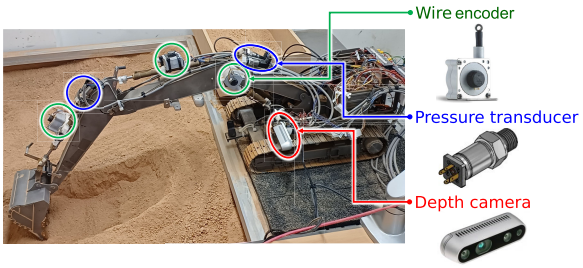


FIGURE 1. Excavator and sensors used for experimental verification.

II. METHOD

A. EXPERIMENTAL EXCAVATOR HARDWARE SETUP

The proposed trajectory planning algorithm was applied to a 1/8-scale excavator (Table 1, Fig. 1). The boom, arm, and bucket were rotated via the linear actuation of hydraulic cylinders, which were fed hydraulic pressure from the hydraulic pump. The rotation of the pump and swing actuator was controlled via an electronic speed controller, using pulse width modulation (PWM) signals from the controller PC. Servo valves for operating the hydraulic cylinders were also controlled by PWM signals. The velocity of the cylinder was controlled using the open/close ratio of the servo valve, which was proportionate to the duration of the PWM "ON" signal (with each period spanning 20 ms).

To establish the automatic excavation system, we installed wire encoders, pressure transducers, and a depth camera on the excavator (Fig. 1). Wire encoders can measure the stroke of the hydraulic cylinders; this was transformed into the rotation angles of the boom, arm, and bucket joints. Two pressure transducers were installed at the ends of the hydraulic cylinders, and the difference between the two pressure values for each cylinder was proportional to the load on that cylinder. The depth camera was installed on the cabin to obtain the point cloud for the terrain.

The excavator's workspace extended 1 m from the boom joint; however, in this study, we limited its range to 0.95 m from the boom joint, to prevent tipping of the excavator under heavy loads.

B. EXPERIMENTAL CONTROLLER SETUP

The control system of the excavator comprised three computers, including the AI computer, machine controller, and host computer (Fig. 2).

The AI computer was equipped with an NVIDIA RTX 3090 GPU for rapid training and application of bucket trajectory planning, based on deep learning. Moreover, it collected the point cloud data via a direct connection with the depth camera, to minimize the data losses during transfer.

For the machine controller of the excavator, we used the CompactRIO made by National Instruments (NI); this gathers sensor data from the excavator and transmits the control input to manipulate machine links in real time. These input and output data flow through input/output (I/O) modules installed on the CompactRIO. The operation of CompactRIO was controlled and managed through virtual instruments from NI's LABVIEW, which were operated in a real-time (RT) module and field-programmable gate array (FPGA).

The host computer was connected to the AI computer via the user datagram protocol communication method. The cable transmitted the measured position and angle information for the bucket tip and received the subsequent position and angle data, which were calculated by the AI computer. Virtual instrument panels for the RT and FPGA modules were operated on the host computer. These panels employed the subsequent position and angle values to calculate PWM "ON" duration signals and thereby control three links of the excavator, and they transmitted these control signals to the three servo valves via the I/O modules of the compact RIO. Two joysticks were also plugged into the host computer, to manipulate the excavator.

C. TEST SIMULATOR

To obtain the training dataset for expert operation, we constructed a personnel test simulator (Fig. 3), using the cockpit seat of an excavator. The soil in this test was decomposed granite. The excavator was installed on a workbench, and experts manipulated it using joysticks. The simulator required the same joystick manipulation method as the real excavator. Experts used the joystick buttons to initiate and terminate data saving. After each round of excavation, the experts performed swing and soil loading operations on a box located on a weighing scale, to measure the weights of the excavated soils. After loading, the test manager recorded the lead time for excavation and the excavated soil weight. In emergencies (e.g., unexpected manipulation and motion of the excavator), the test manager was able to push the kill switch, to cut off electricity to the actuation system and sensors of the excavator.

D. POSITION CONTROL FOR EXCAVATION MOTION

The AI system generated waypoints along the bucket tip trajectory. After transmitting the point's position data from the

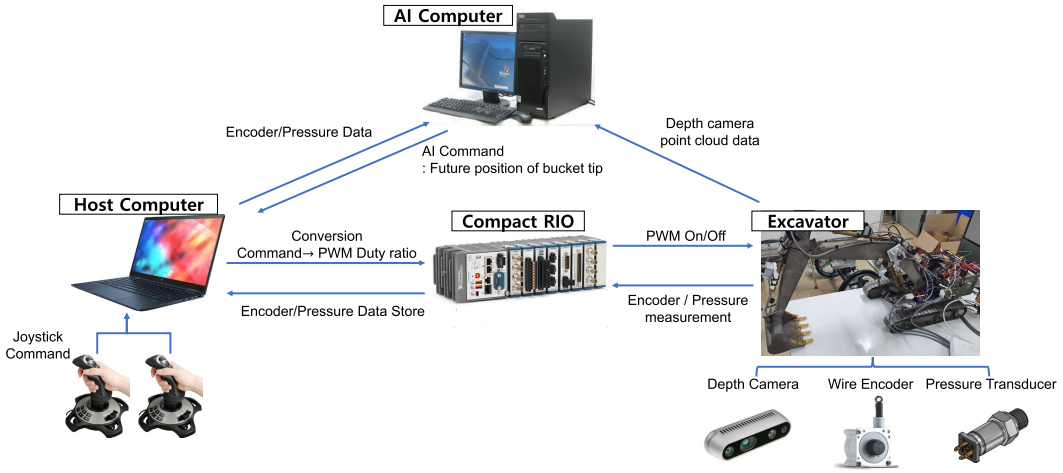


FIGURE 2. Excavator controllers: Host computer, AI computer, and CompactRio.

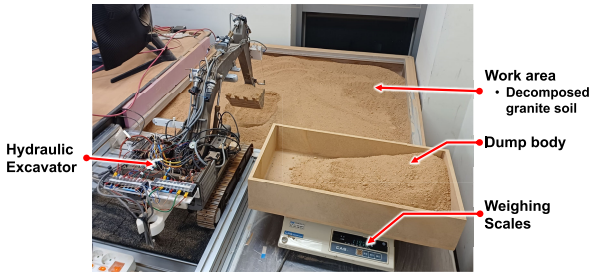


FIGURE 3. Expert test bed.

TABLE 2. Denavit-Hartenberg parameters for excavator.

Link	Link length	Link twist	Link offset	Joint angle
1	a_1	0	0	θ_1
2	a_2	0	0	θ_2
3	a_3	0	0	θ_3

AI computer, the machine controller implemented position control via the point-to-point method. The information for each point included the distance, depth, and angle of attack of the bucket tip; these were calculated and measured with respect to the revolute joint of the excavator's boom as origin. The machine controller transformed these values into variables of the excavator's joints, using the Denavit–Hartenberg (DH) configuration (shown in Table. 2 and Fig. 4) and inverse kinematics. The excavator's manipulator arm, which is composed of three revolute joints except for the swing joint fixed during excavation in this study, had three degrees of freedom (DOF). First, as (1), using the angle of the bucket tip, the sum of the three joint angles is obtained. For a simple representation of θ_1 and θ_2, we calculated the position of the bucket joint using the angle and length between the bucket joint and tip, as expressed in (2). Subsequently, in accordance with the law of cosines expressed in (3) and (4), θ_1 and θ_2 were calculated using the length of the arm and boom and the position of the bucket joint. Thus, θ_3 was calculated using (1) with θ_1 and θ_2.

$$\theta_{123} = \theta_1 + \theta_2 + \theta_3 = \pi - \theta_{bucket} \tag{1}$$

$$p'_x = p_x - a_3 c_{123}, \quad p'_y = p_y - a_3 s_{123} \tag{2}$$

$$\theta_1 = \pi - \arctan\frac{p'_y}{p'_x} - \arccos\frac{{a_1}^2 + (p'_x)^2 + (p'_y)^2 - {a_2}^2}{2a_1\sqrt{(p'_x)^2 + (p'_y)^2}} \tag{3}$$

$$\theta_2 = \pi - \arccos\frac{{a_1}^2 + {a_2}^2 - (p'_x)^2 - (p'_y)^2}{2a_1 a_2} \tag{4}$$

After applying inverse kinematics, the three joint variables were transformed into the hydraulic cylinder lengths of the boom, arm, and bucket. In Figs. 5 and 6, the boom and arm hydraulic cylinders are located facing the angles of the revolute joints with the exception of the bucket joint which was equipped with a 4-bar linkage, as shown in Fig. 7. L_{cyl1} and L_{cyl2}, which represent the cylinder lengths of the arm and boom, were calculated using the law of cosines and angles calculated previously for the inverse kinematics, as shown in (5) and (6).

$$\pi - \theta_1 + \theta_{c_1} + \theta_{c_2} = \arccos\frac{{L_{O_0a}}^2 + {L_{O_0b}}^2 - {L_{cyl1}}^2}{2L_{O_0a}L_{O_0b}} \tag{5}$$

$$\begin{aligned}\theta_{c_4} &= \pi - \theta_{c_3} - (\theta_{c_5} - \theta_2) \\ &= \arccos\frac{{L_{O_1c}}^2 + {L_{O_1d}}^2 - {L_{cyl2}}^2}{2L_{O_1c}L_{O_1d}}\end{aligned} \tag{6}$$

To determine the length of the bucket cylinder L_{cyl3} from θ_3, which was also calculated via inverse kinematics, we initially calculated the sum of θ_{c_9} and $\theta_{c_{10}}$, as shown in (7). Thereafter, the 4-bar linkage near the bucket joint was divided into two triangular sections: $\triangle O_2gf$ and $\triangle O_2hf$. Thus, θ_{c_9} and $\theta_{c_{10}}$ were determined from the law of sines and cosines in these two triangles, using (8) and (9). Moreover, by calculating

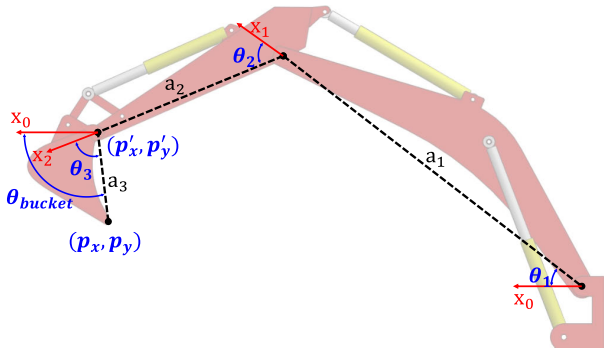


FIGURE 4. 3-DOF robotic system for excavator.

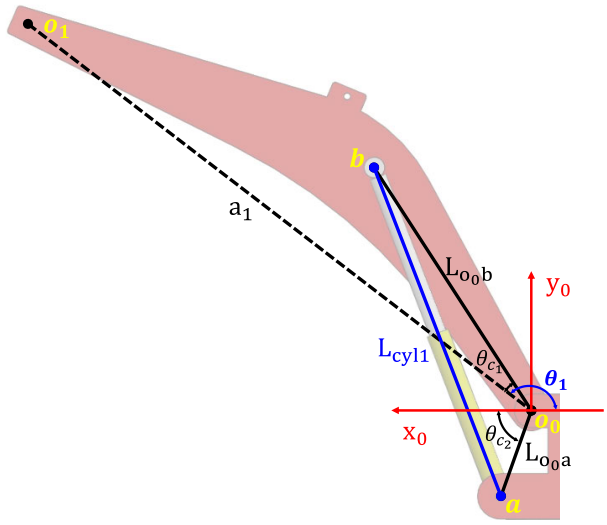


FIGURE 5. Configuration of boom joint and link.

these two angles, the length of L_{O_2f} was determined using the length of bucket cylinder (L_{cyl3}) in this process, as shown in (10). By implementing the inverse sequence, L_{cyl3} was obtained from θ_3.

$$\pi - \theta_{c_8} - \theta_{c_{11}} - \theta_3 = \theta_{c_{10}} + \theta_{c_9} \tag{7}$$

$$\theta_{c_9} = \arcsin \frac{L_{gf} \sin \theta_{c_7}}{L_{O_2f}} \tag{8}$$

$$\theta_{c_{10}} = \arccos \frac{{L_{O_2h}}^2 + {L_{O_2f}}^2 - {L_{fh}}^2}{2L_{O_2h}L_{O_2f}} \tag{9}$$

$$L_{O_2f} = \sqrt{{L_{O_2g}}^2 + {L_{gf}}^2 - 2L_{O_2g}L_{gf}\cos\theta_{c_7}}$$

$$\theta_{c_7} = \pi - \arccos \frac{{L_{gf}}^2 + {L_{eg}}^2 - {L_{cyl3}}^2}{2L_{gf}L_{eg}} \tag{10}$$

As mentioned earlier, the PWM "ON" duration signal was based upon the difference between the target and current length of the cylinder. This difference in each cylinder was used as an input to the PID controllers, which outputted the PWM signals. To summarize this section, the control system (including the AI system) is displayed in Fig. 8.

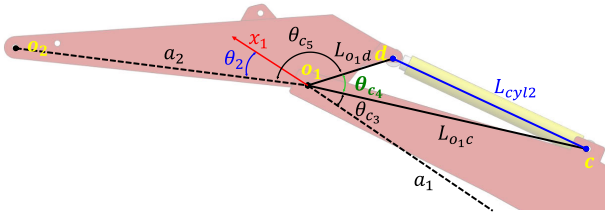


FIGURE 6. Configuration of arm joint and link.

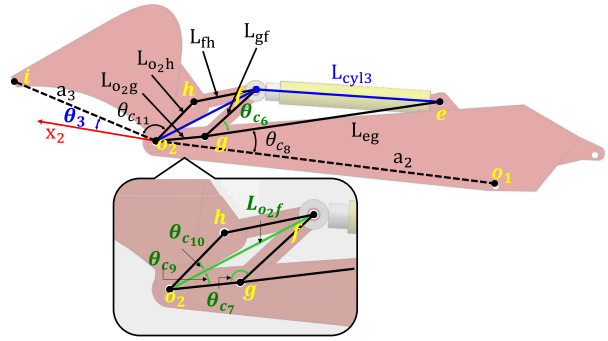


FIGURE 7. Configuration of bucket joint and link.

E. LONG SHORT-TERM MEMORY

RNNs are suitable for processing 1D sequence datasets. They focus on the relations between elements of the input vector and their sequence, using directional connections between RNN cells in the hidden layer. In each step of the RNN, the hidden state of the current cell receives the current input and the hidden state of the previous cell and is calculated as the output. In this process, parameters (including the weights and biases for calculation of the hidden state) are shared across the sequence. Thus, the RNN is advantageous for obtaining features from a sequence with a low computational complexity.

However, the RNN's learning result is considerably influenced by the weight between cells, because the gradient of the loss function is proportional to the iterative multiplication of weights between two cells, as reported in [18]. For example, if the weight value is larger than 1, the gradient will be exponentially large during back-propagation through time (BPTT). By contrast, if the weight is smaller than 1 and the length of the input sequence is large, the gradient can vanish during BPTT.

Owing to the instability of the RNN learning process, we employed the LSTM (an improved RNN) as the foundation of the AI system for bucket-trajectory planning. The LSTM's memory cell replaces the RNN cell and generates the hidden state h_t (called the short-term memory) and cell state C_t (called the long-term memory) from the current input, cell state, and hidden state of the previous step as input, as depicted in Fig. 9. When updating states, the memory cell employs sigmoid gates and a tangent layer. The sigmoid gates include forget, input, and output gates. The forget gate

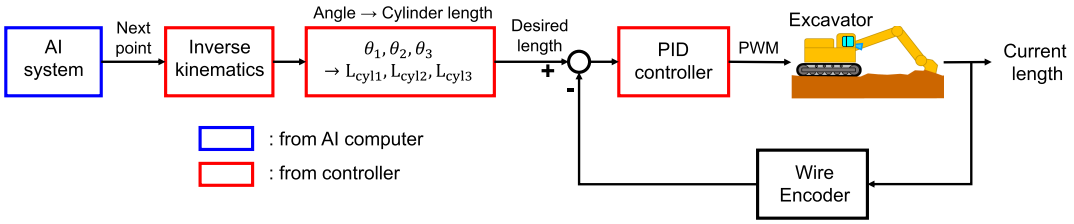


FIGURE 8. Block diagram of excavator control system.

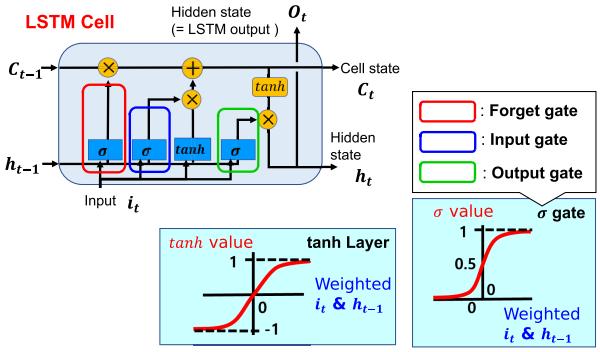


FIGURE 9. Structure of long short-term memory (LSTM).

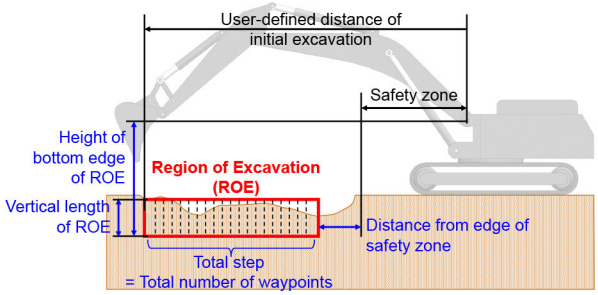


FIGURE 10. Concept of ROE generation AI.

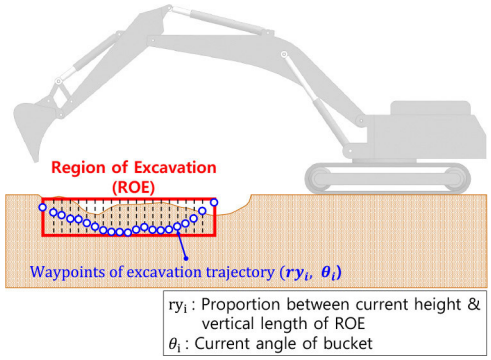


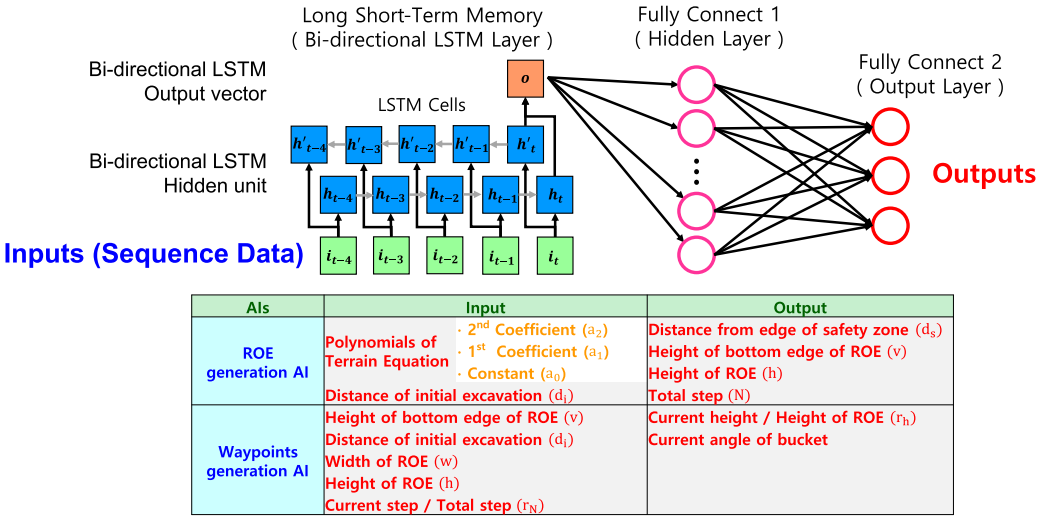
FIGURE 11. Concept of waypoint generation AI.

determines values to be forgotten or transmitted from the previous cell state. The input gate identifies which of the calculation results of the *tanh* layer are to be selected for the cell state update; it takes a combination of the current input and prior hidden state as inputs. The output gate determines which values will become the current hidden state after calculating the *tanh* function using the current (updated) cell state as its input. During this process, the prior cell state is transmitted in its original form. This strategy ensures that the LSTM is not affected by vanishing or explosion of the gradient. The parameter sharing and stable gradients of the LSTM make it an appropriate algorithm to generate an output vector from the 1D long-input dataset.

F. BUCKET-TIP TRAJECTORY-PLANNING AI SYSTEM

The bucket-tip trajectory planning AI system was used as a framework to generate the base trajectory of the bucket as well as the angle of the bucket along this generated trajectory. The system was inspired by the instance segmentation method, which includes algorithms for generating the region-of-interest (ROI) box near the target object and for drawing a segmentation mask to separate it from the background image. Similarly, the first AI module [the region-of-excavation (ROE) generation AI] generates the ROE, which is a minimum excavation boundary (from penetration to bucket curl) that includes all waypoints of the bucket tip, to penetrate the ground without crashing into the excavator cabin (as presented in Fig. 10). It also defines the number of waypoints in the ROE. The second AI (called the waypoint generation AI) generates the trajectory of the bucket tip in the workspace specified by the ROE generation AI, and it calculates the angle of the bucket in each step (as illustrated in Fig. 11). These AIs possess the same structure and comprise multiple layers, including a bidirectional LSTM that considers the effects of forward and backward sequences of input data and two fully-connected layers (as depicted in Fig. 12). However, the inputs and outputs of these AIs differ.

In the data used for the ROE generation AI in Fig. 12, polynomials of the terrain equation are the most important parameters for the ROE creation of bucket tip. We initially obtained point cloud data for the soil surface using a depth camera before the start of each excavation round. The data in this measurement included 1,270 × 840 points, owing to the large dataset size, terrain features could not be feasibly obtained for generating the bucket-tip trajectory. To address this problem, we set the point cloud ROI to reduce the size of the point cloud, according to the width of the bucket and the starting point of excavation. Subsequently, we projected



AIs	Input	Output
ROE generation AI	Polynomials of Terrain Equation · 2nd Coefficient (a_2) · 1st Coefficient (a_1) · Constant (a_0) Distance of initial excavation (d_i)	Distance from edge of safety zone (d_s) Height of bottom edge of ROE (v) Height of ROE (h) Total step (N)
Waypoints generation AI	Height of bottom edge of ROE (v) Distance of initial excavation (d_i) Width of ROE (w) Height of ROE (h) Current step / Total step (r_N)	Current height / Height of ROE (r_h) Current angle of bucket

FIGURE 12. Structure of AI systems in bucket-tip trajectory-planning AI system.

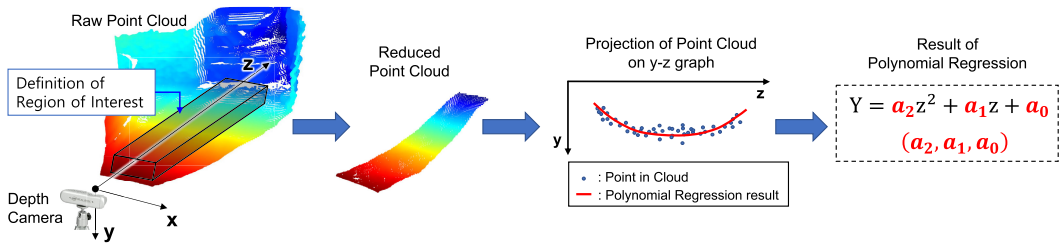


FIGURE 13. Transformation from point cloud to polynomials.

points in the defined ROI onto the y–z graph and performed polynomial regression of the y–z scatter diagram, to ensure three coefficients of the 2^{nd} order equation, which represents a slope or a curvature of the terrain, after polynomial regression, as depicted in Fig. 13. After polynomial regression, the ROE generation AI used these polynomials (a_2, a_1, a_0) and the distance from the boom joint to the user-defined initial excavation point d_i as inputs. The ROE generation AI outputted features of the ROE, including the distance between the edge of the ROE and the edge of the safety zone d_s, height of the ROE's bottom edge v, vertical length of ROE h, and number of excavation waypoints N in the ROE. After operation of ROE generation AI, the width of ROE w was determined using (11).

$$w = d_i - d_s - s \tag{11}$$

where s represents the range of the safety zone from the boom joint, which is determined from crash safety considerations.

Subsequently, the waypoint generation AI used the geometric information of the ROE and the distance of the initial excavation point as inputs (including the height of the ROE's bottom edge v, distance of the initial excavation point d_i, ROE's vertical length h, and ROE's width w), and the current step number divided by the total number of waypoints r_N, which was updated in every loop of the bucket-tip-trajectory generation process.Using these values, the waypoint generation AI outputted the proportion r_h between the current height of the bucket tip and the vertical length of the ROE, as well as the angle of the bucket in each step. After application of the waypoint generation AI, the bucket tip position was calculated as (12) and (13).

$$p_x = d_i - wr_N \tag{12}$$

$$p_y = v + hr_h \tag{13}$$

respectively, where p_x and p_y denote the positions of the bucket tip, as shown in Fig. 4. These values and the angle of the bucket tip were transmitted to the controller, to operate the excavator.

G. COLLISION AVOIDANCE ALGORITHM

The bucket-tip trajectory planning AI system primarily generated the excavation trajectory after training using the human expert trajectory data. However, when the bucket sweeps through the ground, any unexpected variances (including underground rocks or buried pipes) must be considered. This variability under the soil surface cannot be identified using the depth camera or trajectory planning AI and leads to unsatisfactory results, including machine stalling or rollover

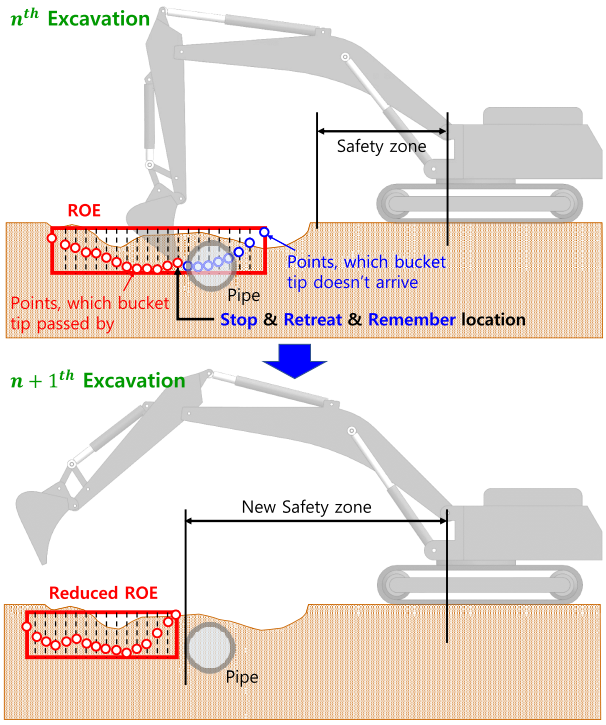


FIGURE 14. Process of collision avoidance algorithm.

of the excavator during excavation (due to overloading of the bucket).

To address these problems, we applied the collision avoidance algorithm by monitoring the cylinder pressure in each loop, as illustrated in Fig. 14. When the bucket tip approached an underground obstacle, the pressure of the cylinder increased and exceeded the defined pressure threshold. Thereafter, the collision avoidance algorithm suspended excavation and retracted the bucket in the opposite direction along the path obtained by the trajectory planning AI system. This strategy minimized the likelihood of the bucket getting trapped. After a few reverse moves, the collision avoidance algorithm pulled up the bucket and finished the excavation. Before the bucket was retracted, the collision avoidance algorithm saved the location information of the place at which the cylinder pressure exceeded the threshold. This saved value was applied to determine the width of the ROE in the subsequent excavation, by updating the safety zone range, to thereby protect the bucket from crashing. If the ROE width after ROE generation is too large to prevent crashes, the ROE width is recalculated for the new safety zone to be established in front of the underground obstacles. If this is not the case, the ROE configuration of the ROE generation AI module will be transmitted into the waypoint generation AI without modification.

H. ACQUISITION OF TRAINING DATA

We obtained the data for the AI systems from experts' excavations, following the sequence illustrated in Fig. 15.

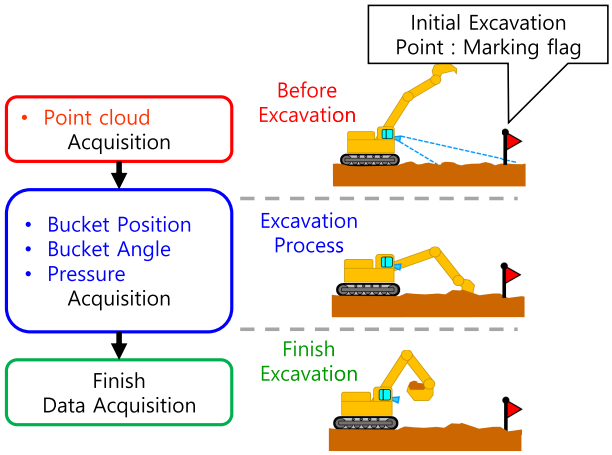


FIGURE 15. Process of expert data acquisition.

TABLE 3. Hyperparameters of bucket trajectory-planning AI systems.

Hyperparmeters	Value and Info.
Units of LSTM layer	2(bidirectional) × 128
Units of 1^{st} FC layer	64
Units of 2^{nd} FC layer	4 (waypoint generation AI: 2)
Activation of FC	Linear
Loss	MSE
Optimizer	ADAM
Metric	MSE
Epoch	30
Batch size	10 (Waypoint generation AI: 100)

During expert excavation, four successive excavations were performed in each cycle, to produce AI approaches that are robust against changes in the terrain slope. In this study, we set the distance from the initial excavation point to the boom joint as 0.95 m and we placed a marking flag there to allow the experts to identify where to initiate the penetration. Expert excavation was performed under the same conditions and heights as those applied to our AI system, to compare performances. When the bucket approached the marking flag during expert excavations (from penetration to bucket curl), the data (including the point cloud, bucket position and angle, and cylinder pressure) were saved. Among the expert data, we selected 180 trials as qualified training data; the soil weight varied from 2.8 kg to 3.15 kg and the lead time varied from 0.5 sec to 1.1 sec. To supplement these data, we performed data augmentation by inserting random noise into the dataset. We limited the magnitude of random noise to avoid the results from being affected.

To obtain training data for ROE generation, point clouds were pre-processed via size reduction, projection onto a 2D graph, and polynomial regression (to extract polynomials), following the process depicted in Fig. 13. The distance of initial excavation d_i was determined from the first position of the bucket tip in each trial. The ROE configuration was

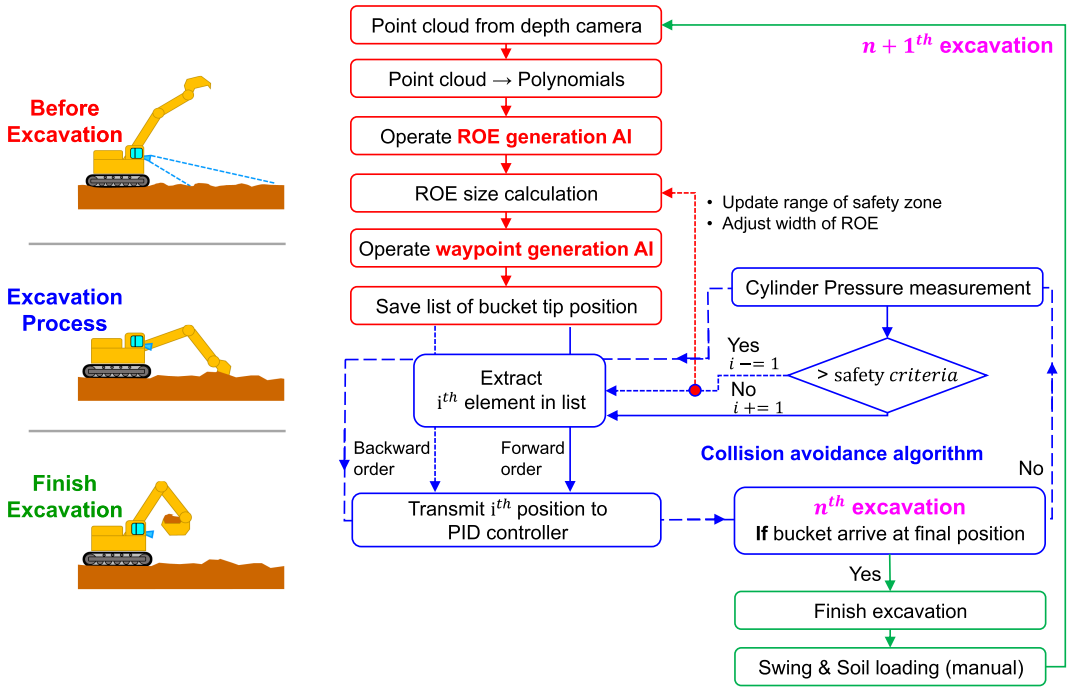


FIGURE 16. Process of bucket tip-trajectory planning algorithm.

Algorithm 1 Bucket Tip-Trajectory Planning Algorithm

Input: Pressure, Point cloud
Output: Future position, Future angle
Initialisation:
Obtain point cloud in defined ROI
Calculate polynomials of terrain equation
Generate ROE box by AI, considering safety zone
Enlist bucket-tip trajectories by AI
Loop process
if State1 **then**
  if $i ==$ Size of trajectory list **then**
    State1 = False
  end if
  if (Current pressure > Pressure threshold) **then**
    State1, State2 = False, True
    Save current location as new safety zone
  end if
  Transmit i^{th} vector set of list to controller
  $i+ = 1$
end if
if State2 **then**
  Transmit i^{th} vector set of list to controller
  $i- = 1$
  if (i == Threshold of retreat) **then**
    State2 = False
  end if
end if
return *Futureposition*, *FutureAngle*

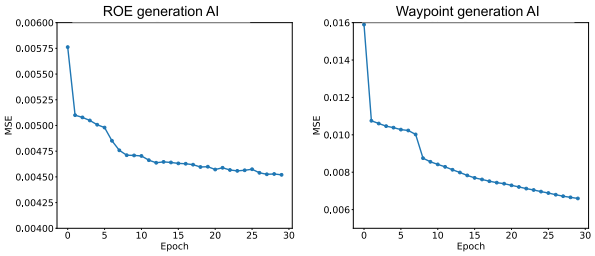


FIGURE 17. Training losses of two AI systems.

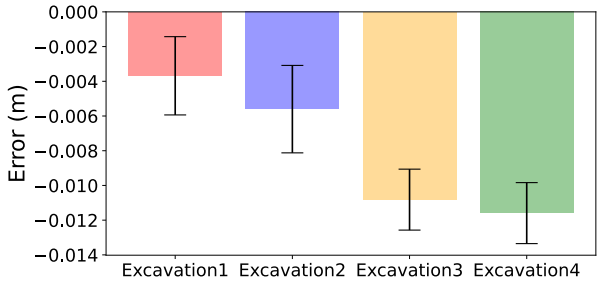


FIGURE 18. Tracking performance of each excavation: tracking error.

determined by calculating the smallest region including all waypoints for each excavation.

The training data for the waypoint generation AI included the ROE configuration (calculated during the data generation process for the ROE generation AI) at the distance of

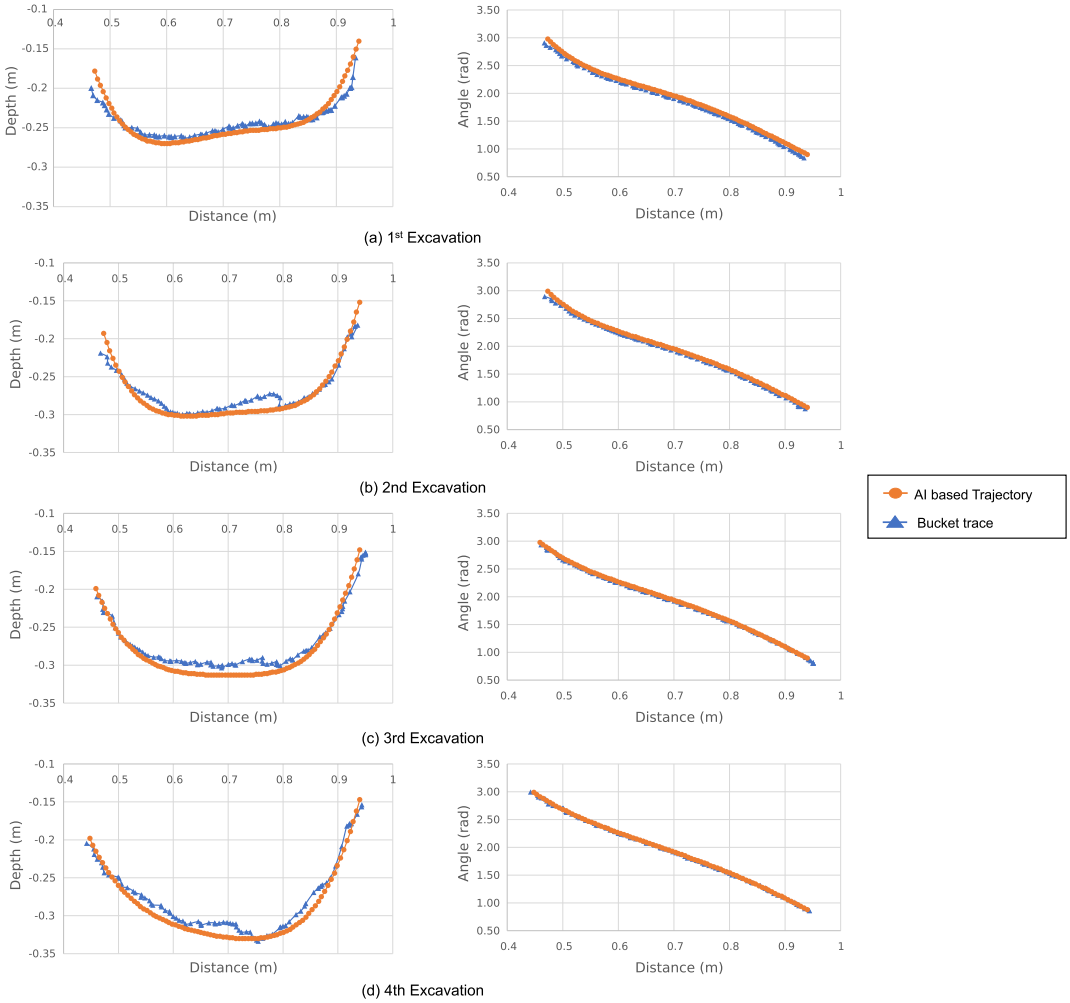


FIGURE 19. Comparison between AI-based trajectory and bucket trace.

initial excavation. Furthermore, the ratio between the bucket displacement and ROE width was calculated using (14).

$$r_N = \frac{d_i - p_x}{w} \tag{14}$$

where r_N denotes the proportion value and corresponds to the current step divided by the total steps in Fig. 12, w indicates the ROE width, and p_x expresses the distance of the bucket tip in each excavation step. Among the outputs of the training data set, the ratio between the vertical displacement of the bucket and the height of the ROE was calculated as (15).

$$r_h = \frac{p_y - v}{h} \tag{15}$$

where r_h corresponds to the current height of the bucket divided by the vertical length of the ROE (see Fig. 12); v and h represent the height of the ROE's bottom edge and the vertical length of the ROE, and p_y denotes the current depth of the bucket tip. Additionally, the scaling factor was applied to the angle of the bucket tip, to adjust the magnitude to match other values.

TABLE 4. Confidence interval range of tracking error.

Excavation No.	99% Confidence interval range (m)
Excavation1	-0.006 ∼ -0.0014
Excavation2	-0.0082 ∼ -0.003
Excavation3	-0.0126 ∼ -0.009
Excavation4	-0.0134 ∼ -0.0098

I. HYPERPARAMETERS

The hyperparameter set for the two AI modules included the cost function, size of the mini-batch, epochs, and node size of each layer. These hyperparameters are listed in Table 3.

J. ALGORITHM STRUCTURE

The algorithm included three main component algorithms (as shown in Fig. 16 and Algorithm 1): the ROE generation AI, waypoint generation AI, and collision avoidance algorithm.

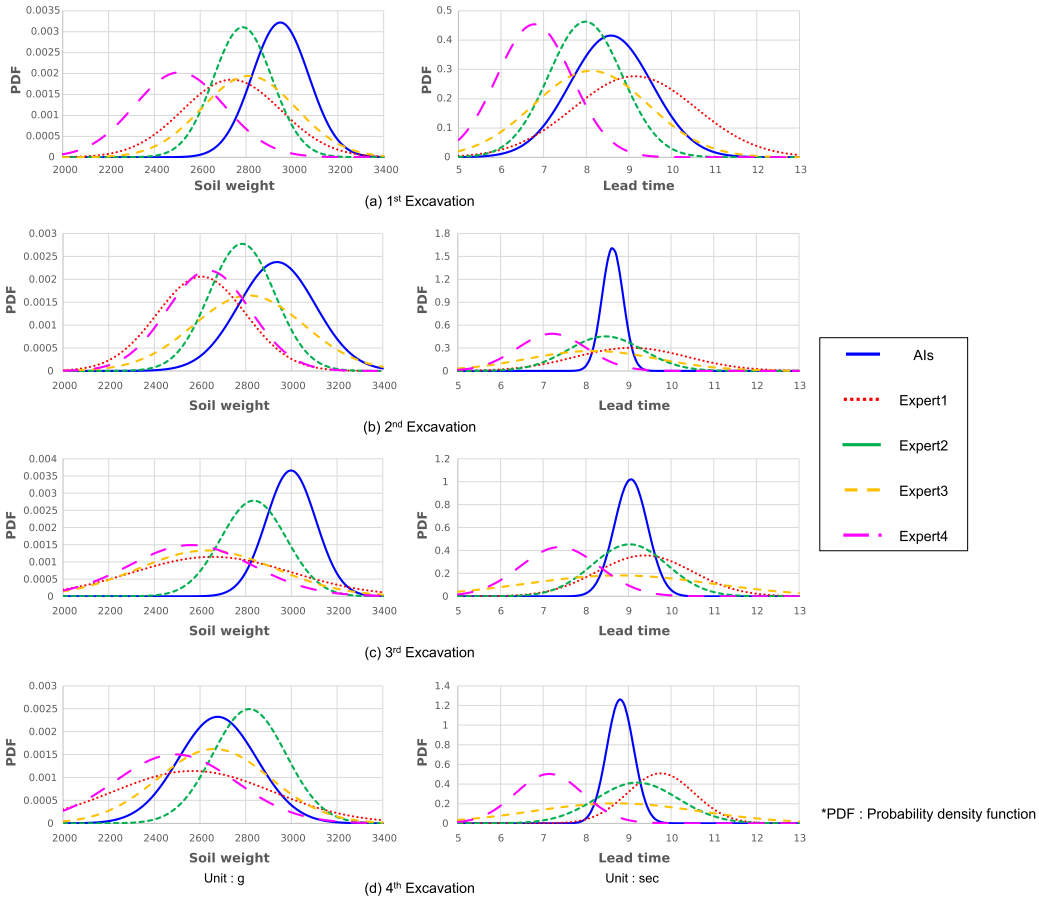


FIGURE 20. Normal distribution of excavation performance for excavation by experts and AI system.

TABLE 5. Average and standard deviation of excavated soil weight.

unit : g		AI	Expert1	Expert2	Expert3	Expert4
1^{st}	avg	2950	2737	2786	2813	2509
	std	124	215	128	205	197
2^{nd}	avg	2937	2605	2784	2814	2638
	std	168	194	144	242	182
3^{rd}	avg	2998	2654	2835	2630	2560
	std	109	348	143	299	268
4^{th}	avg	2677	2569	2816	2663	2491
	std	172	349	160	246	266

TABLE 6. Average and standard deviation of excavation lead time.

unit : sec		AI	Expert1	Expert2	Expert3	Expert4
1^{st}	avg	8.6	9.1	8	8.1	6.8
	std	0.96	1.4	0.86	1.4	0.88
2^{nd}	avg	8.6	9.1	8.5	8.2	7.2
	std	0.25	1.3	0.88	1.5	0.82
3^{rd}	avg	9.1	9.4	9	8.8	7.4
	std	0.39	1.1	0.88	2.2	0.93
4^{th}	avg	8.8	9.8	9.2	8.7	7.1
	std	0.32	0.78	0.96	2	0.79

Before excavation, the depth camera obtained the point cloud within the defined ROI; subsequently, polynomial regression was applied to secure polynomial values to be inputted to the ROE generation AI. The ROE generation AI outputted the ROE configuration for the number of waypoints, and the waypoint generation AI generated the depth and bucket tip angle as location data for the waypoint. Incorporating the distance calculated in (12), the waypoint's location was saved in a list variable. When the waypoint of the final step was saved, the excavation started. During excavation, the pressure values for the boom, arm, and bucket cylinder were continuously monitored by the collision avoidance algorithm, to determine whether they exceeded the safety criteria. Depending on the results, the excavator continued the current excavation or cancelled it and adjusted the ROE width for the next excavation, whilst identifying the locations of underground obstacles.

III. EXPERIMENTAL RESULTS

The primary purpose of the bucket-tip trajectory planning AI system is to achieve robust excavation of various terrains.

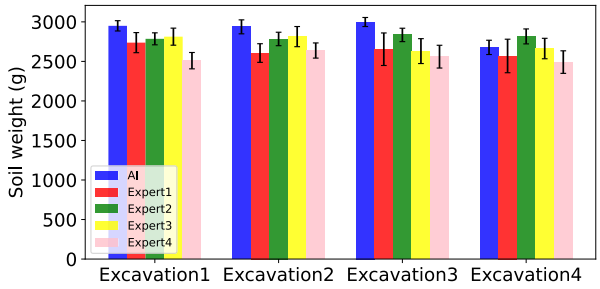


FIGURE 21. 99% Confidence interval range: Soil weight.

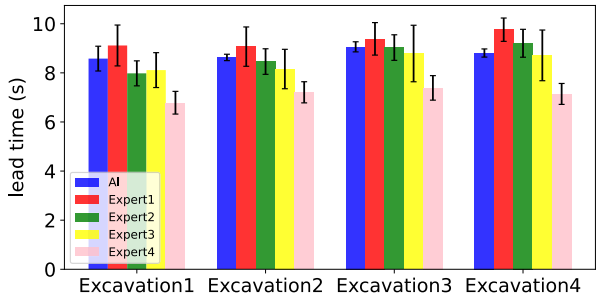


FIGURE 22. 99% Confidence interval range: Lead time.

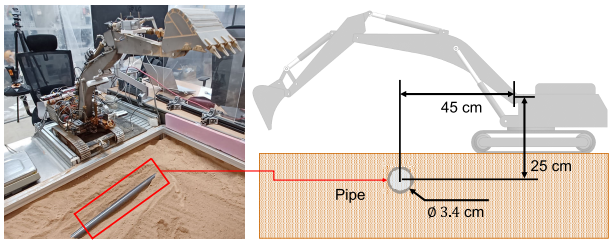


FIGURE 23. Underground obstacle(Pipe), installed in front of excavator.

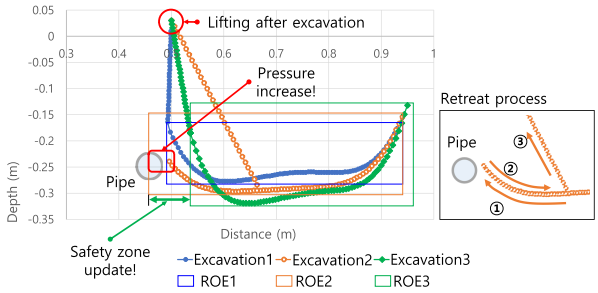


FIGURE 24. Excavation under activation of collision avoidance algorithm.

To validate our AI system, we performed four successive excavations. Subsequently, we performed comparison experiments against expert's trials, to verify that the AI-directed excavation matched those of experts in terms of efficiency. During the excavation, we constrained the distance of the excavation starting point, which is approximately 0.95 m from the boom joint for safety. We also focused on the excavation volume and excavation lead time as performance factors.

A. PERFORMANCE VALIDATION OF TRAINED AI

The performance of the two AIs was validated during training. The training loss and accuracy were the key performance parameters used in the validation. The loss and metric used to optimize and evaluate the AIs were calculated using the mean squared error (MSE) expressed in (16).

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2, \tag{16}$$

where $\hat{Y}_i$ is the estimated value of the ROE configuration in the ROE generation AI and the estimated rate for the height and bucket tip angle in the waypoint generation AI. Y_i denotes the truth value (i.e., the experts' results) for the input data. The MSE of each AI is illustrated in Fig. 17, and the performance stabilized well at ∼30 epochs for both AI algorithms.

B. TRAJECTORY TRACKING PERFORMANCE DURING SUCCESSIVE EXCAVATIONS

This experiment was performed to verify the quality of the bucket-tip trajectories generated by the AI system, by comparing the bucket courses and AI-obtained trajectories (see Figs. 18 and 19). We verified that the bucket-tip trajectory developed by the AI system could be followed by the PID controller, in spite of undesirable factors (e.g., delays between the opening of the electronic valve and the transmission of hydraulic power and during the continuous update of new target points from the AI system with a 100 ms period). Moreover, more effort was required to optimize the gains of PID controllers during excavation. Comparing this against the tracking performance of an excavator considered in a prior study [15], whose average position tracking error was 0.078 m, we find that the limit for the tracking error in this study was less than 0.015 m. When the tracking error between the measured track and AI-based path (shown in Fig. 19) satisfied this criterion, the bucket-tip trajectory planning AI system could develop a traceable track for the bucket tip. The results of the tracking experiment are illustrated in Fig. 18 and Table 4 for the 99% confidence interval range; they show that the tracking errors of the 1^{st}, 2^{nd}, 3^{rd}, and 4^{th} excavations were under 0.015 m on average. Thus, our algorithm can develop a traceable path.

C. EFFICIENCY OF THE EXPERTS AND THE AI

The excavation performance of the bucket-tip trajectory planning AI system was compared against those of experts, to assess efficiency. We used the weight of excavated soil and the lead time (from the starting location to the location at which the bucket was lifted) as indicators of performance. Over 80 excavation trials were performed by experts and the AI system, and the mean and standard deviation of each were calculated. In Fig. 20, and Tables. 5 and 6, the AI system and experts can be seen to excavate more than 2 kg

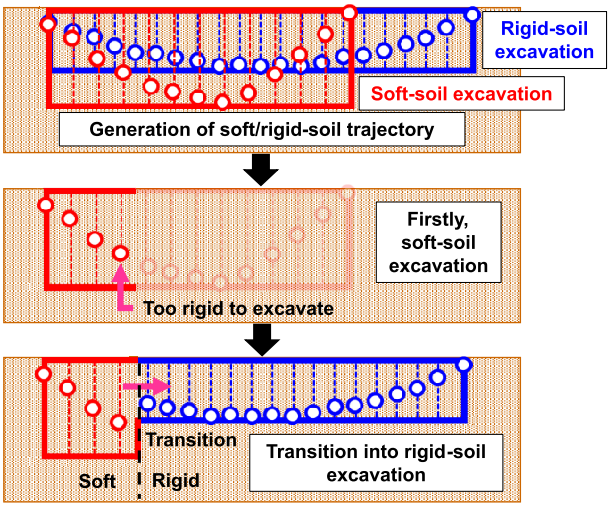


FIGURE 25. Concept of trajectory transition.

of soil and to require 8 ∼ 9 sec for excavation. However, the bucket-tip trajectory planning AI system exhibited a steadier work efficiency compared to the experts, as validated by the standard deviations of soil weight and lead time (presented in the 99% confidence interval ranges in Figs. 21 and 22).

D. APPLICATION OF COLLISION AVOIDANCE ALGORITHM

We also verified the applicability of the collision avoidance algorithm, via pressure monitoring. We installed a pipe to serve as an underground obstacle (as shown in Fig. 23) and performed successive excavations. When the pressure values of the three cylinders exceeded the safety criteria, the collision avoidance algorithm halted excavation and retracted the bucket (similar to the 2^{nd} excavation illustrated in Fig. 24). The ROE width decreased after we updated the range of the safety zone to cover the location of the detected underground pipe. After reduction of the ROE width, the trajectory of the 3^{rd} excavation (Fig. 24) was also shortened, without crashing into the underground pipe.

IV. CONCLUSION AND FUTURE WORKS

The autonomous excavation operated using our newly proposed bucket-tip trajectory planning AI system and a collision avoidance algorithm based on pressure monitoring yielded results comparable to those of experts (in terms of the excavation volume and lead time). Although the geography of the targeted terrain changed under successive excavations, the LSTM-based AI system robustly generated a traceable trajectory without knowledge of the soil dynamics. The collision avoidance algorithm adjusted the ROE width for the waypoint generation AI by monitoring the pressure, thereby preventing the bucket from crashing into underground obstacles.

However, certain challenges must be overcome when applying the bucket-tip trajectory-planning AI system in real-world excavations. The initial penetration into frozen or wet soil is not smooth. Consequently, pitch occurs in the cabin during the penetration process. In future works, to ensure safety, we will consider the excavation of soils of different hardness, by applying a transition algorithm from soft- to rigid-soil excavation (as shown in Fig. 25), where the trajectories are developed by the proposed AI system following additional training under rigid soil. The transition function will use the excavator's pose and position data to determine the transition time during excavation. Furthermore, we will extend the application of our AI system from one section to a field including several sections. To this end, we will investigate excavation task planning according to topographical data for the specified field, by referencing [19], and we will integrate this with the AI system proposed in this study. We hope that this work will contribute to the expansion of the excavation industry and improve worker safety.

REFERENCES

[1] S. M. Vahed, X. Song, J. S. Dai, H. K. Lam, L. D. Seneviratne, and K. Althoefer, "Soil estimation based on dissipation energy during autonomous excavation," *IFAC Proc. Volumes*, vol. 41, no. 2, pp. 13821–13826, 2008.
[2] A. J. Koivo, M. Thoma, E. Kocaoglan, and J. Andrade-Cetto, "Modeling and control of excavator dynamics during digging operation," *J. Aerosp. Eng.*, vol. 9, no. 1, pp. 10–18, Jan. 1996.
[3] B. P. Patel and J. M. Prajapati, "Soil-tool interaction as a review for digging operation of mini hydraulic excavator," *Int. J. Eng. Sci. Technol.*, vol. 3, no. 2, pp. 894–901, 2011.
[4] O. Luengo, S. Singh, and H. Cannon, "Modeling and identification of soil-tool interaction in automated excavation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. Innov. Theory, Pract. Appl.*, vol. 3, Oct. 1998, pp. 1900–1906.
[5] P. Corke, J. Trevelyan, H. Cannon, and S. Singh, "Models for automated earthmoving," in *Experimental Robotics VI*. Springer, 2000, pp. 163–172, doi: 10.1007/BFb0119395.
[6] Y. Zhao, J. Wang, Y. Zhang, and C. Luo, "A novel method of soil parameter identification and force prediction for automatic excavation," *IEEE Access*, vol. 8, pp. 11197–11207, 2020.
[7] P. Egli and M. Hutter, "A general approach for the automation of hydraulic excavator arms using reinforcement learning," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 5679–5686, Apr. 2022.
[8] J. Park, D. Cho, S. Kim, Y. B. Kim, P. Y. Kim, and H. J. Kim, "Utilizing online learning based on echo-state networks for the control of a hydraulic excavator," *Mechatronics*, vol. 24, no. 8, pp. 986–1000, Dec. 2014.
[9] J. Park, B. Lee, S. Kang, P. Y. Kim, and H. J. Kim, "Online learning control of hydraulic excavators based on echo-state networks," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 1, pp. 249–259, Jan. 2017.
[10] S. Lee, D. Hong, H. Park, and J. Bae, "Optimal path generation for excavator with neural networks based soil models," in *Proc. IEEE Int. Conf. Multisensor Fusion Integr. Intell. Syst.*, Aug. 2008, pp. 632–637.
[11] S. Dadhich, F. Sandin, U. Bodin, U. Andersson, and T. Martinsson, "Field test of neural-network based automatic bucket-filling algorithm for wheel-loaders," *Autom. Construct.*, vol. 97, pp. 1–12, Jan. 2019.
[12] R. J. Sandzimier and H. H. Asada, "A data-driven approach to prediction and optimal bucket-filling control for autonomous excavators," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 2682–2689, Apr. 2020.
[13] B. Son, C. Kim, C. Kim, and D. Lee, "Expert-emulating excavation trajectory planning for autonomous robotic industrial excavator," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 2656–2662.
[14] P. Egli, D. Gaschen, S. Kerscher, D. Jud, and M. Hutter, "Soil-adaptive excavation using reinforcement learning," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 9778–9785, Oct. 2022.
[15] P. Egli and M. Hutter, "Towards RL-based hydraulic excavator automation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 2692–2697.
[16] L. Tai, J. Zhang, M. Liu, J. Boedecker, and W. Burgard, "A survey of deep network solutions for learning control in robotics: From reinforcement to imitation," 2016, *arXiv:1612.07139*.

[17] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *Int. J. Robot. Res.*, vol. 37, nos. 4–5, pp. 421–436, 2018.
[18] R. Grosse, "Lecture 15: Exploding and vanishing gradients," Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, Tech. Rep., 2017.
[19] T. Osa and M. Aizawa, "Deep reinforcement learning with adversarial training for automated excavation using depth images," *IEEE Access*, vol. 10, pp. 4523–4535, 2022.

JAEMYUNG HUH received the B.S. degree from the School of Mechanical Engineering, Korea University, Seoul, South Korea, in 2012, where he is currently pursuing the Ph.D. degree in mechanical engineering.

His research interests include algorithm design for automated excavation, machine learning, and mechanism design for automated guided vehicles and gripper systems.

JANGHO BAE received the B.S., M.S., and Ph.D. degrees from the Department of Mechanical Engineering, Korea University, Seoul, South Korea.

He is currently a Professor with the Department of Mechanical and Automotive Engineering, Kyungsung University, Busan, South Korea. His research interests include the design and control of hydraulic robotic systems, physical human–robot interactions, human–robot collaboration systems, the application of artificial intelligence in field robotics, and autonomous car systems.

DONGHYEON LEE received the B.S. degree from the School of Mechanical Engineering, Korea University, Seoul, South Korea, in 2021, where he is currently pursuing the M.S. degree in mechanical engineering.

His research interests include human–robot interaction, field robotics, and machine learning.

JAEWON KWAK received the B.S. degree from the School of Mechanical Engineering, Korea University, Seoul, South Korea, in 2021, where he is currently pursuing the M.S. degree in mechanical engineering.

His research interests include hydraulic excavator control and computer vision.

CHANYOUNG MOON received the B.S. degree in mechanical engineering from Korea University, Seoul, South Korea, in 2019, where he is currently pursuing the M.S. degree in mechanical engineering.

His research interests include automation in excavation, field robotics, and medical robots.

CHULWHAN IM received the B.S. degree from the School of Mechanical Engineering, Korea University, Seoul, South Korea, in 2022, where he is currently pursuing the M.S. degree in mechanical engineering.

His research interests include field robotics, computer vision, and autonomous excavators.

YEONHO KO received the B.S. degree in mechanical design engineering from Chungnam National University, Daejeon, South Korea, in 2015. He is currently pursuing the Ph.D. degree in mechanical engineering with Korea University, Seoul, South Korea.

He is also a researcher. His research interests include the design and control of wearable robotic systems, physical human–robot interactions and collaboration systems, and construction field robotics.

TAE KOO KANG received the B.S. degree in applied electrical engineering, the M.S. degree in visual image processing, and the Ph.D. degree in electrical engineering from Korea University, Seoul, Republic of Korea, in 2001, 2004, and 2012, respectively.

He was a Research Professor with Korea University, from 2012 to 2014, and an Assistant Professor in information and telecommunication engineering in Cheonan, Republic of Korea, in 2015 and 2016. He is currently an Assistant Professor with the Department of Human Intelligence and Robot Engineering, Sangmyung University, Cheonan. His research interests include computer vision, robotics, artificial intelligence, and machine learning.

DAEHIE HONG received the B.S. and M.S. degrees from the School of Mechanical Engineering, Korea University, Seoul, South Korea, in 1985 and 1987, respectively, and the Ph.D. degree from the Department of Mechanical and Aerospace Engineering, University of California at Davis, CA, USA, in 1994.

He is currently a Professor with the School of Mechanical Engineering, Korea University. His research interests include manufacturing automation, precision machine design and control, field robotics, and autonomous car systems.