

## Research Article

# Path Planning for Excavator Arm: Fuzzy Logic Control Approach

**Nga Thi-Thuy Vu**  and **Nam Phuong Tran**

*School of Electrical Engineering, Hanoi University of Science and Technology, Hanoi, Vietnam*

Correspondence should be addressed to Nga Thi-Thuy Vu; [nga.vuthithuy@hust.edu.vn](mailto:nga.vuthithuy@hust.edu.vn)

Received 8 September 2020; Revised 28 October 2020; Accepted 13 November 2020; Published 21 November 2020

Academic Editor: Changsheng Li

Copyright © 2020 Nga Thi-Thuy Vu and Nam Phuong Tran. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper proposes an algorithm to design the path for automated excavator arm. The path designing is divided into two phases. Firstly, the fuzzy logic technique is used to determine the change of the path to adapt with the variation of the material after each digging period. Secondly, the B-spline algorithm is used to build the optimal trajectory. By this, the excavator can maximize the dug weight although the shape of the material is variation. The effectiveness of the algorithm is verified through computer simulation and real-time experiment. The simulation and experimental results show that the generated optimal path keeps the dug weight around the expected value regardless of the change of the environment.

## 1. Introduction

Excavator is one of the popular machines in the field of construction and transportation. The working environment of excavator is usually dirty and dangerous; so, automating the operation of the excavator is the trend recently. In the automation working mode of each machine, the reference trajectory is one of the important factors which guarantee the efficiency of the system. For this reason, the trajectory of the unmanual system should be predesigned to satisfy some specific requirements.

For the problems of optimal trajectory, the B-spline technique is widely used in many works. In [1, 2], the B-spline is used to build the smooth, minimum-time, and jerk trajectories for manipulator robots. The parameters of the B-spline curves are obtained from solving the cost function with the mechanical constraints. The effectiveness of the method is verified through computer simulation and practical robots. In [3], the cubic B-spline is combined with the particle swarm optimization algorithm to generate the smooth optimal trajectory for the serial and parallel manipulators. The cubic spline interpolation is changed in the first and last knots of the optimized trajectory for having zero jerk at the beginning and end points of the trajectory. In [4], the strategy which combines the multidegree splines in the Cartesian space and multidegree B-spline in the joint

space is designed for manipulator robot. The generated trajectory guarantees the controllability, bound, and continuity of the velocity and the acceleration at the initial and ending moments for the end effector.

Unlike the spline methods which are very helpful in solving optimal problems, artificial intelligent including fuzzy logic and neural network is known as a powerful tool to solve the obstacle avoidance problem. In [5–7], a fuzzy-based path planning technique is introduced for multi-mobile robot with unknown dynamic environment and obstacles. There are two fuzzy logic systems are used in these works, one is used for obstacle avoidance, and the other for target reaching [5, 6] or planning speed of robots [7]. The unknown environment is detected by sensors which are located on the robots [5]. Meanwhile, the optimal motion path of the robot arm is implemented in [8] based on the fuzzy logic technique. The algorithm can shorten the time of motion and make the robots reach the target smoothly. Moreover, the robots in this work can avoid safety accidents, reduce casualties, and improve industrial productivity. The problem of the collision avoidance for 2 manipulators sharing a workspace is presented in [9]. In this context, one manipulator is considered as moving obstacle with the other; then, based on the output of the artificial potential field and the error between current position of the end effector and the goal, the fuzzy system infers the correction in

displacement for each joint of each individual robot's arm. Some other applications of fuzzy logic and neural network in path planning are studied in [10, 11].

For the excavator robots, recently there are some research studies which focus on this topic. In [12], the optimal trajectory for the excavator arm is built based on the velocity and acceleration of the bucket; however, measuring velocity and acceleration is difficult in practical cases. Likewise, designing the working time and torque optimal trajectory for the excavator with limitation in the actuator's force is shown in [13]. Otherwise, to create a suitable working path with surrounding environment, in [14], the laser scanner is used to obtain the information of the pile, and then the pile model is analysed and divided into a specific layer. Based on this information, the local path and then the global path are designed. While in [15], the neural network is employed in order to identify the soil model. Based on the interacting force between the bucket and pile, the optimal trajectory is designed by combining this result with reaction exerted force of the bucket. In [16], the adaptive neuron-fuzzy inference system (ANFIS) is combined with B-spline technique to generate the optimal time and jerk for the excavator arm. The scheme gives the good results with a smooth and optimal generated trajectory; however, the working condition in this case is in the static environment.

In this paper, an algorithm based on the B-spline and fuzzy logic techniques is proposed to design the path for the automated excavator arm. The optimal time and jerk for each digging period is supported by the B-spline algorithm while the fuzzy logic scheme is used to determine the change of the path after each digging period. By this way, the reference path for the excavator is adjusted after each digging period, so it can maximize the dug weight despite the variation in the shape of the material. The effectiveness of the scheme is verified through the computer simulation and experimentation. The simulation and experimental results show that the built reference path keeps the dug weight around the expected value regardless of the change of the environment. Also, the time and jerk in each digging period is minimum.

## 2. Path Planning for Excavator Arm

*2.1. Problem Description.* Let us consider the excavator arm with a block diagram as shown in Figure 1. The excavator arm includes three parts: the boom, the stick, and the bucket.

One of the requirements for excavator during working process is that the weight of material after each excavating period should be within acceptable range to optimize the efficiency. In practical cases, however, working environment is dynamic and always changes through excavating process. Hence, if the trajectory of the excavator is fixed, it cannot ensure that dug weight is maintained during working process. In the manual excavator, the operator must observe constantly the working environment and digging process of the excavator in order to adjust the excavator trajectory suitably. To automate the excavating operation, a trajectory planning module is designed to substitute the role of the operator.

In this paper, the trajectory for the arm is designed to satisfy two requirements:

- (i) Optimize the time and jerk in each digging period
- (ii) Change the trajectory after some periods to keep the dug weight in the acceptable range as the working environment changed

Before going into the detail work, the following assumptions are used:

- (i) The material lies under the ground
- (ii) The material is considered as uniform density

It is noted that these assumptions are only used to support for simulation. In the simulation, the change of the material and the dug weight after each digging period are unknown. In order to get the feedback information of the dug weight, the virtual environment is built in Matlab/SIMULINK with using these assumptions to calculate the weight and to express the change of the material. In the real world, this problem is solved easily by using a load cell, so the assumptions are not necessary.

In each period, the trajectory of the arm is divided into four stages as follows:

- (i) Stage 1: firstly, the bucket's angle should be opened largely to create a suitable angle with material's surface. Simultaneously, the boom's angle and the stick's angle change to move the end effect to the desired spot.
- (ii) Stage 2: the bucket's angle is kept constant and the bucket is forced inside the material by moving the other angles.
- (iii) Stage 3: the bucket angle is closed to get material inside the bucket. In this time, the boom's angle and the stick's angle are kept unchanged.
- (iv) Stage 4: in the last stage, by remaining the bucket's angle nearly zero, the material is kept inside the bucket. The boom's angle and the stick's angle change to bring the material out of the ground.

Figure 2 shows the four stages of the trajectory in the joint coordinate, and Figure 3 shows the shape of the trajectory in the working coordinate. It is noted that the point  $D$  in Figure 3 is the deepest spot that the excavator's bucket reaches in one digging period, which is corresponding to the end of the second stage.

From the predesigned trajectory as shown in Figure 3, five via-points are obtained corresponding with the beginning and the end of all stages. Based on these via-points, a smooth and differentiable trajectory is designed to optimize the time and jerk in each period of the working. In the next section, a time and jerk optimization algorithm based on the fifth order B-spline curve is employed to improve the smoothness and time-saving properties of the trajectory.

*2.2. Time and Jerk Optimization Trajectory Based on 5<sup>th</sup> Order B-Spline.* The problem of time and jerk optimization is formulated as follows [17]:

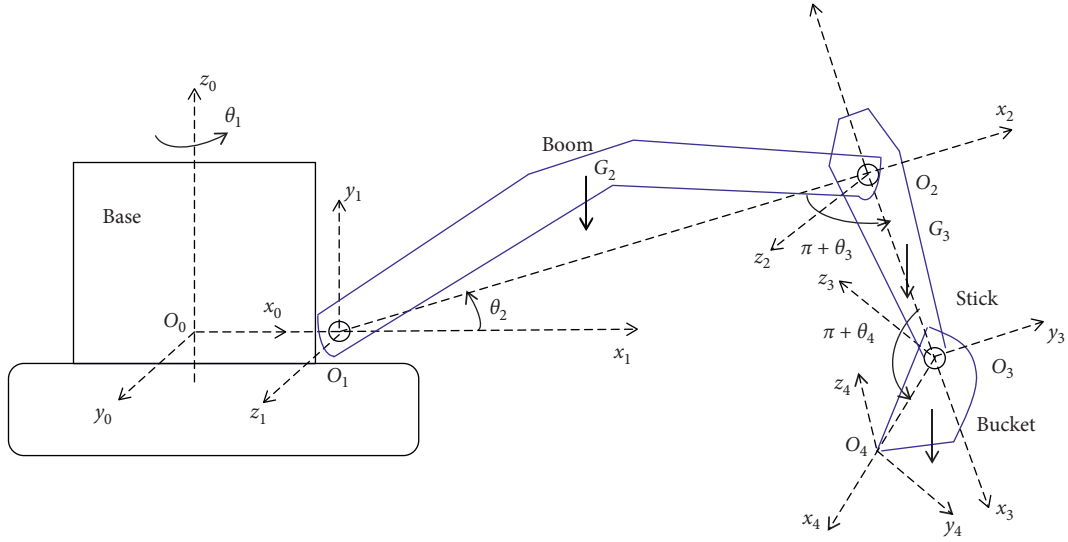


FIGURE 1: Block diagram of the excavator [16].

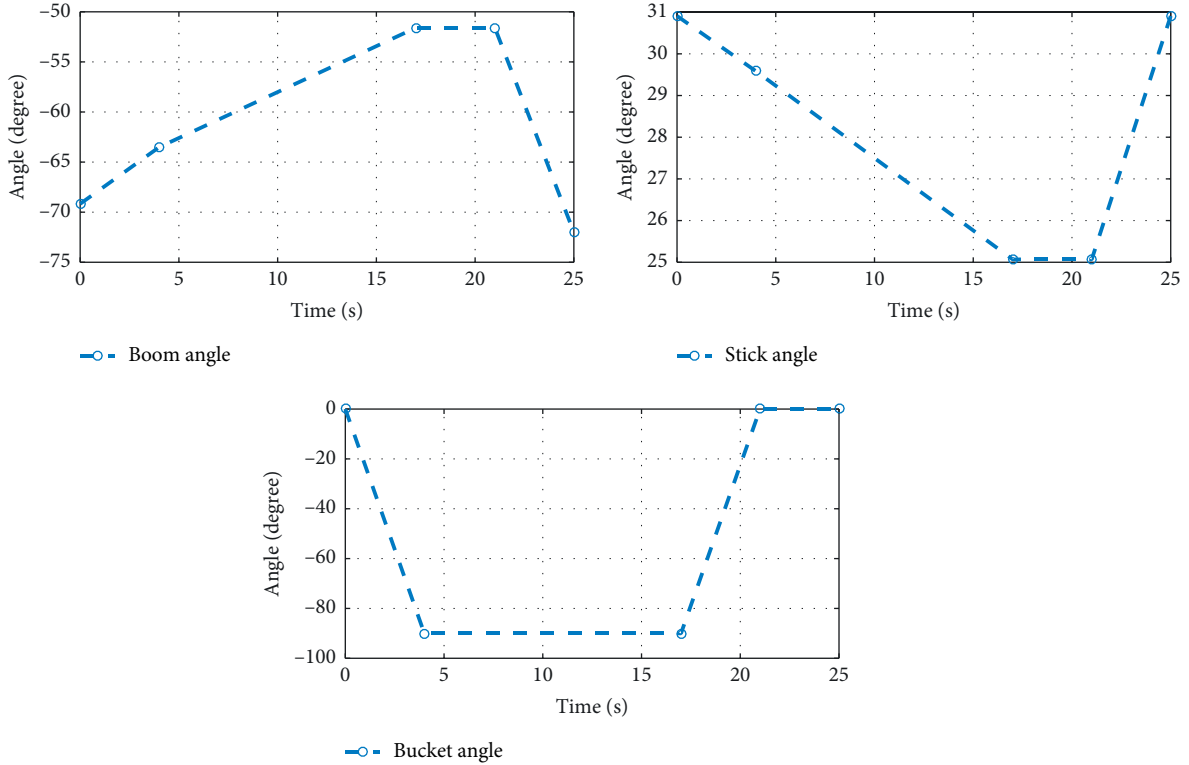


FIGURE 2: The stages of the joint angle.

$$\begin{cases} \text{find } \min k_T N \sum_{i=1}^{vp-1} h_i + k_J \sum_{j=1}^N \int_0^{t_f} (\ddot{q}(t))^2 dt, \\ \text{Subject to:} \\ |\dot{q}_j(t)| \leq VC_j, \quad j = 1, \dots, N, \\ |\ddot{q}_j(t)| \leq AC_j, \quad j = 1, \dots, N, \\ |\ddot{q}_j(t)| \leq JC_j, \quad j = 1, \dots, N. \end{cases} \quad (1)$$

In which,  $k_T$  and  $k_J$  are scalars;  $vp$  is the number of via-points;  $h_i$  is time interval that the trajectory moves from  $i^{th}$  point to  $(i+1)^{th}$  point;  $\dot{q}(t)$ ,  $\ddot{q}(t)$ , and  $\ddot{q}(t)$  are the velocity, acceleration, and jerk of the  $j^{th}$  joint, respectively;  $VC_j$ ,  $AC_j$ , and  $JC_j$  are the bound of velocity, acceleration, and jerk for the  $j^{th}$  joint, respectively; and  $N$  is the number of joints.

In each period of working time, the trajectory of the arm is desired to have the form as shown in Figure 4. From this desired shape, some via-points are defined; then, the optimization problem is solved by using sequential quadratic

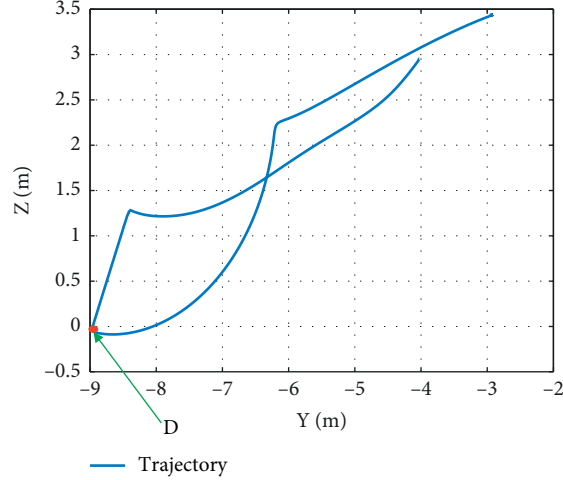


FIGURE 3: The shape of the trajectory in the working coordinate.

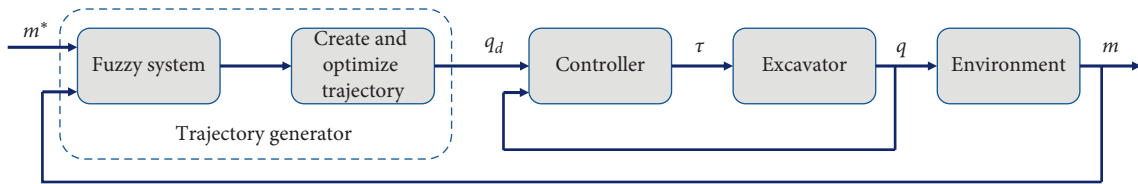


FIGURE 4: Whole system's structure.

programming. After solving this problem, the B-spline technique is used to build the smooth curve for each joint in one period of digging by using the following equation:

$$B_p(t) = \sum_{i=1}^n Q_i \cdot N_{i,p}(t), \quad (2)$$

where  $n$  is the number of control points,  $Q_i$  is the control point of the B-spline, and  $N_{i,p}(t)$  is the basic function of degree  $p$ . The basic function is given in the form of regression as follows:

$$\begin{cases} N_{i,p}(t) = \frac{t-t_i}{t_{i+p}-t_i} \cdot N_{i,p-1}(t) + \frac{t_{i+p+1}-t}{t_{i+p+1}-t_{i+1}} \cdot N_{i+1,p-1}(t), \\ N_{i,0} = \begin{cases} 1, & \text{for } t_i \leq t < t_{i+1}, \\ 0, & \text{elsewhere.} \end{cases} \end{cases} \quad (3)$$

In (3), the values of  $t_i$ ,  $t_{i+1}$ ,  $t_{i+p+1}$  are determined from solving optimal problem (1). Finally, the trajectory for the arm has the following form:

$$q(t) = \sum_{i=1}^n Q_i \cdot N_{i,p}(t), \quad (4)$$

where the values of  $Q_i$  are calculated from the condition that the trajectory goes through the via-points which are pre-designed.

*2.3. Automatically Adjusting Trajectory's Parameter to Adapt with the Variation of the Working Environment.* During the operation of the excavator, the shape of material changes after each digging period. Therefore, adjusting appropriately parameters of the trajectory is necessary to guarantee work efficiency. According to the shape of the trajectory that is described in Section 2.1, one should analyse several potential changes in the trajectory's parameters that can affect the amount of the excavated material in each period. Particularly, it can be seen easily that by shifting the deepest point of the trajectory,  $D(y_D, z_D)$  in Figure 4, the dug weight can be changed.

The structure of the whole system is demonstrated in Figure 4. The fuzzy blocks in Figure 4 are used to estimate the change in amount of  $y_D$  and  $z_D$  according to the error between the dug weight in the previous period and the desired value. Once the values of  $y_D$  and  $z_D$  are calculated, the complete trajectory is formulated based on the optimization algorithm which is developed in Section 2.2. The algorithm for trajectory inference is shown in Figure 5.

In Figure 5,  $\Delta z_{D_0}$  is the first chosen value as default decrease amount for  $z_D$ . Since after each period, the material is taken, so  $z_D$  must be declined in order that the excavator can move more deeply in the next period. In detail, if the value of  $\Delta z_{D_0}$  does not help excavating enough material as expectation, one should increase  $\Delta z_{D_0}$  by adding a small number  $\Delta z_D$  and vice versa. This process can be formulated as follows:

$$\begin{cases} \Delta z_D(k+1) = \Delta z_D(k) + \varepsilon(\Delta m_k), \\ z_D(k+1) = z_D(k) - \Delta z_D(k+1), \end{cases} \quad (5)$$

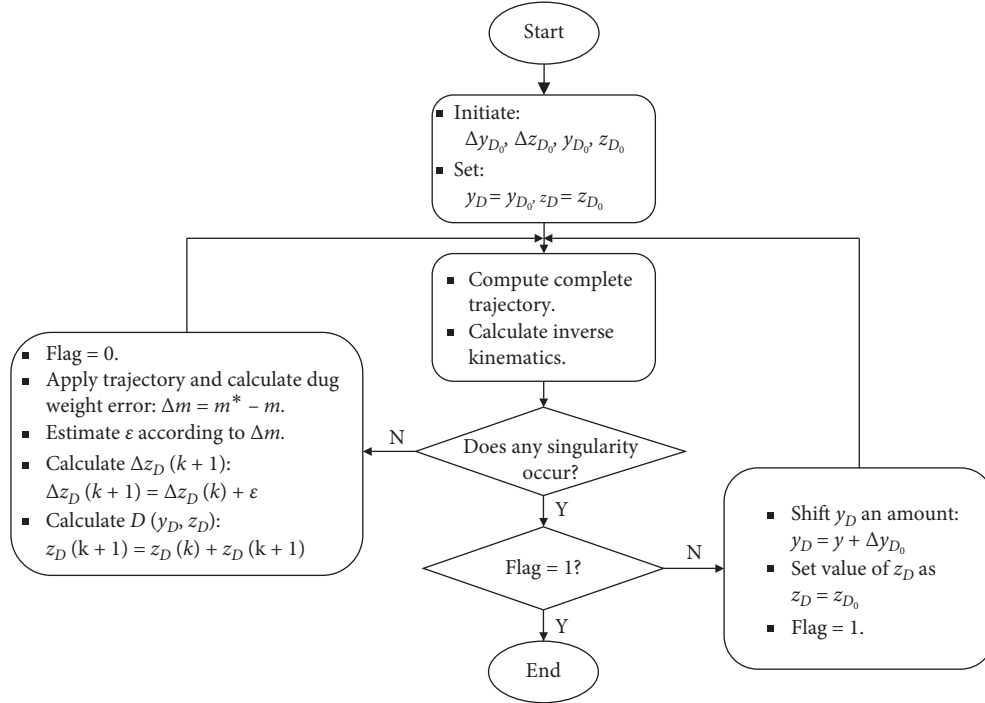


FIGURE 5: Algorithm of trajectory generation.

where the initial values are set as  $z_D(0) = z_{D0}$ ,  $\Delta z_D(0) = \Delta z_{D0}$ , and  $\Delta m_k = m^* - m_k$ , with  $m^*$  is the desired amount of the dug weight and  $m_k$  is the dug weight in the  $k^{\text{th}}$  period. The first equation of (5) implies that after each period, the displacement of  $z_D$ , denoted by  $\Delta z_D$ , should be adjusted based on the weight error  $\Delta m_k$ . For instance, if  $\Delta m_k$  is positive, it means that the amount of displacement in the previous period is not large enough; hence,  $\Delta z_D$  in the next step should be increased. Generally, the change in the displacement rate  $\Delta z_D$  depends on the function of  $\Delta m_k$  that is denoted by  $\varepsilon(\Delta m_k)$  as in the first equation of (5). The second equation of (5) describes the shifting of  $z_D$  based on the position displacement in the first equation. Note that the initial state  $z_D(0)$  and  $\Delta z_D(0)$  are considered as default value of position and shifted position, and  $z_D$  and  $\Delta z_D$  are reset to the default value if the certain criteria are met.

From (5), it is obviously that the probable adjustment in  $\Delta z_D$  calculated by  $\varepsilon(\Delta m_k)$  determines the work efficiency. Therefore, the heart of the trajectory inference algorithm is the function  $\varepsilon(\Delta m_k)$ . In this work, the  $\varepsilon(\Delta m_k)$  is constructed by utilizing fuzzy logic. Structure of the fuzzy logic system representing for  $\varepsilon(\Delta m_k)$  is illustrated in Figure 6.

The input and output of the system are dug weight error  $\Delta m$  and adjusting amount  $\varepsilon(\Delta m_k)$ , respectively. Membership function of input is the triangular function:

$$\mu(x) = \begin{cases} 0, & \text{if } x \leq a, \\ \frac{x-a}{b-a}, & \text{if } a \leq x \leq b, \\ \frac{c-x}{c-b}, & \text{if } b \leq x \leq c, \\ 0, & \text{if } x \geq c. \end{cases} \quad (6)$$

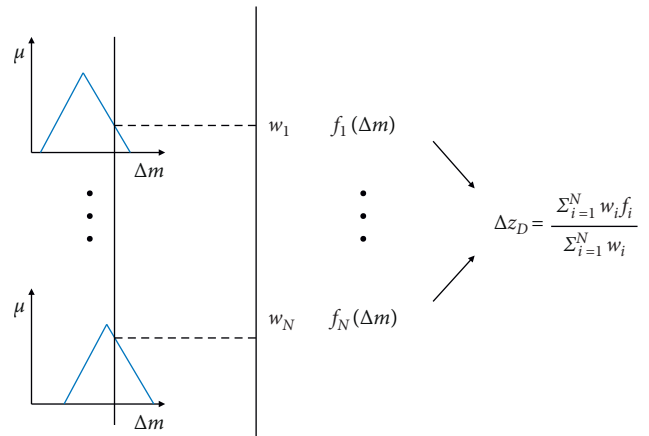


FIGURE 6: Sugeno fuzzy logic inference system.

In this paper, seven membership functions of input are used corresponding with linguistic variables as follows: very negative big (vnb), negative big (nb), negative small (ns), zero, positive small (pm), positive big (pb), and very positive big (vpb). The rule base of the fuzzy logic system is expressed as follows:

- If  $\Delta m$  is vnb, then  $\varepsilon$  is  $r_1$ . If  $\Delta m$  is zero, then  $\varepsilon$  is  $r_4$ .
  - If  $\Delta m$  is nb, then  $\varepsilon$  is  $r_2$ . If  $\Delta m$  is ps, then  $\varepsilon$  is  $r_5$ .
  - If  $\Delta m$  is ns, then  $\varepsilon$  is  $r_3$ . If  $\Delta m$  is pb, then  $\varepsilon$  is  $r_6$ .
  - If  $\Delta m$  is vpb, then  $\varepsilon$  is  $r_7$ ,
- where  $[r_1, r_2, r_3, r_4, r_5, r_6, r_7] = [-0.4, -0.25, -0.1, 0, 0.1, 0.25, 0.4]$ .

Note that if some positions on the trajectory are out of reach of the manipulator, singularities in inverse kinematics

will occur. In this case, the value of  $z_D$  should be reset and the value of  $y_D$  should be changed an amount  $\Delta y_{D_0}$ . In the case that singularity still occurs after changing  $y_D$  and reset value of  $z_D$ , it means that the manipulator has already reached its limit in workspace, and the algorithm then is terminated. The variable flag in the algorithm indicates that  $y_D$  is already changed.

### 3. Results and Discussion

*3.1. Optimal Trajectory for Digging Process.* In each period, after getting the deepest point from the fuzzy module, the via-points for each joint are calculated based on the desired shape. Then, optimal problem (1) is solved with the constraint as shown in Table 1.

Velocity, acceleration, and jerk in both starting and ending point are set at 0. By adjust the value of  $k_T$ ,  $k_J$  in (1), the trajectory can be created either more time-saving or smoother. In this work, two cases ( $k_T = 1$ ,  $k_J = 0$  and  $k_T = 0.1$ ,  $k_J = 0.9$ ) are implemented to show the effect of  $k_T$ ,  $k_J$  to the properties of the trajectory.

*Case 1.*  $k_T=1$  and  $k_J=0$

By setting  $k_T=1$  and  $k_J=0$ , the designer's intention is to reduce the time of the trajectory without considering the jerk. The joint angle and jerk responses are shown in Figure 7. It can be seen easily that the jerk of the bucket's joint reaches the limit of jerk (60 deg/s<sup>3</sup>).

*Case 2.*  $k_T=0.1$  and  $k_J=0.9$

In this case, the smoothness in working is concerned much more, joint angles and jerks are shown in Figure 8. Total time of the trajectory slightly increases compared to Case 1, from 16.17 s in the first case to 18.38 s in the second case. However, the jerk of all three joints is reduced significantly, and maximum jerk in the second case is only about 40 (deg/s<sup>3</sup>).

*3.2. Calculating the Change of Trajectory between Periods to Maximum the Dug Weight as the Environment Changing.* Secondly, the fuzzy logic system that is proposed in Section 2.3 is utilized to calculate the suitable trajectory based on observing dug weight error. At the first step,  $D$  is set at default value  $D = (y_D, z_D) = (-9, -0.1)$ , and desired dug weight is  $m^* = 30$  kg. After applying the default trajectory, excavated weight is 8.5 kg. The dug weight error in 20 periods is shown in Figure 9. It can be seen that by applying fuzzy logic to change the trajectory, the dug weight after the first step is improved significantly and the maximum error is about 5 kg, but this error occurs only in one period. The trajectories in 1<sup>st</sup>, 3<sup>rd</sup>, 5<sup>th</sup>, 7<sup>th</sup>, 11<sup>th</sup>, 13<sup>th</sup>, 15<sup>th</sup>, and 17<sup>th</sup> are shown in Figure 10.

Note that, in this example, after the 10<sup>th</sup> digging period, the trajectory makes the system reach the kinematic limitation of the excavator, so it is reset to the highest point and the  $y_D$  should be changed to adjust the horizon of the

TABLE 1: Mechanical constraint of each joint.

	Velocity (deg/s)	Acceleration (deg/s <sup>2</sup> )	Jerk (deg/s <sup>3</sup> )
1	100	60	60
2	100	60	60
3	100a	60	60

trajectory. The shape of the material after the 10<sup>th</sup> period and the 20<sup>th</sup> period is shown in Figure 11.

*3.3. Experimental Results.* To evaluate the performances of the proposed algorithm, the experimental system has been built and illustrated in Figure 12. The parameters of the system are as follows:

- Type: hydraulic excavator
- Volume of oil type: 100 ml
- Depth of digging: 414 mm
- Height: 832 mm
- Width: 240 mm
- Length: 1.15 m
- Maximum load: 10 kg
- Maximum oil pressure: 20 Bar
- Engine of oil pumps: BLDC

The information about the dug weight is feedback to the computer from the load cell after each digging period. This signal then is fed to the fuzzy system to determine the change of the  $y_D$  and  $z_D$ . From these results, the optimization toolbox will solve optimal problem (1) and then the B-spline technique will be used to build the reference trajectory for the excavator arm. The experimental results are shown in Figures 13–15.

Figure 13 illustrates the trajectory of the excavator arm in the joint space in the 2<sup>nd</sup> period. In the first stage, the bucket opens to reach a suitable angle (about  $-82$  degree), and the boom and stick angles are unchanged. Next, the bucket angle is remained, and the bucket is forced inside the material by increasing the boom angle from initial value (about  $-72$  degree) to the value of  $-35$  degree. In the 3<sup>rd</sup> stage, the bucket angle increases to close the bucket and the material is kept inside the bucket during the unload process. Finally, the boom angle changes to move the material out of the ground. During the cycle of digging, the stick angle is kept constant. Figure 14 depicts the angles of the boom, stick, and bucket at the 4<sup>th</sup> period. At this cycle, the angle of the stick is reduced to  $-25$  degree to adapt with the change of the material.

Figure 15 presents the dug weight error in each period of digging. In the first cycle, the dug weight error is quite high (about 3.5 kg) because the trajectory for this cycle is set randomly. In the 2<sup>nd</sup> cycle, the trajectory of the arm is adjusted and then the error reduces to 0.3 kg. This trajectory is also applied to the 3<sup>rd</sup> cycle, and the weight error increases to 0.7 kg. This big error leads to the change of the trajectory in the 4<sup>th</sup> cycle to maximize the dug weight. The experiments are repeated for other cycles.

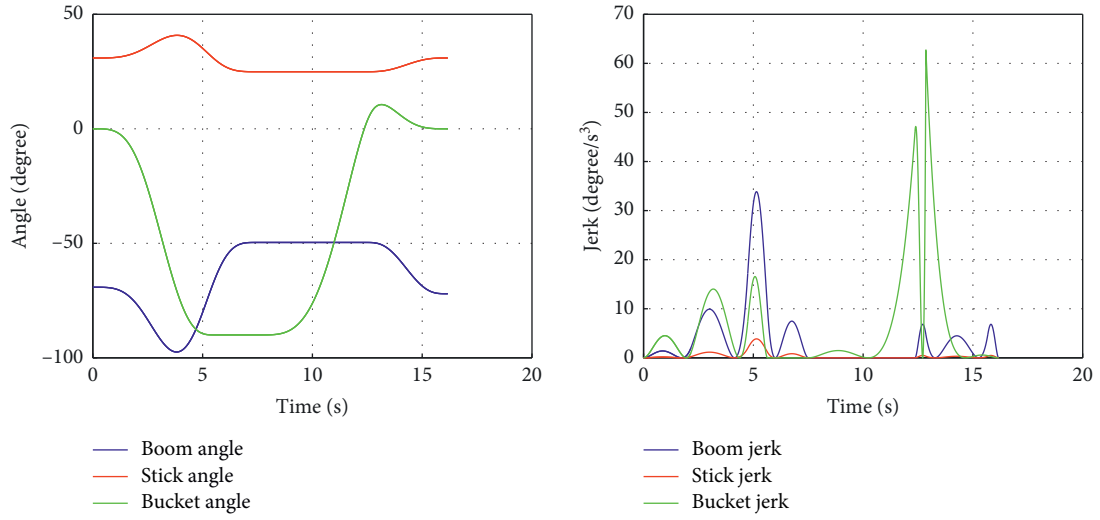


FIGURE 7: Joint angle and jerk with  $k_T=1$  and  $k_J=0$ .

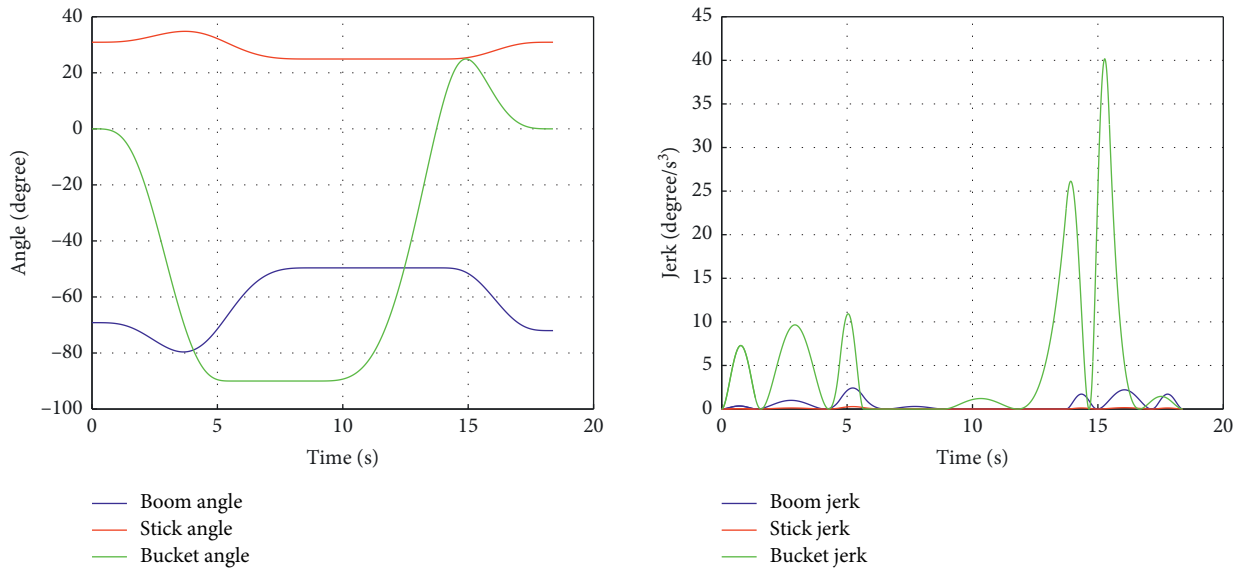


FIGURE 8: Joint angle and jerk with  $k_T=0.1$  and  $k_J=0.9$ .

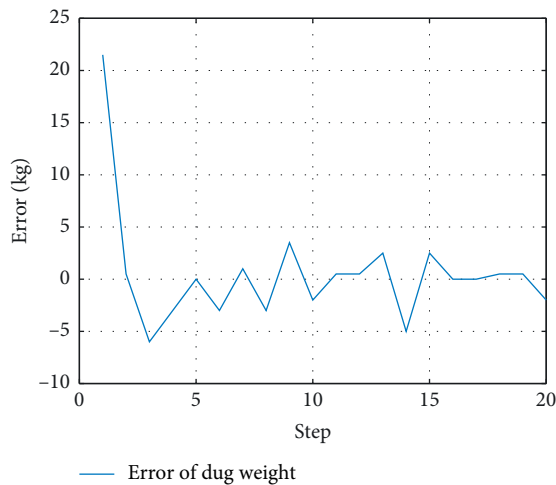


FIGURE 9: Dug weight error through each period.

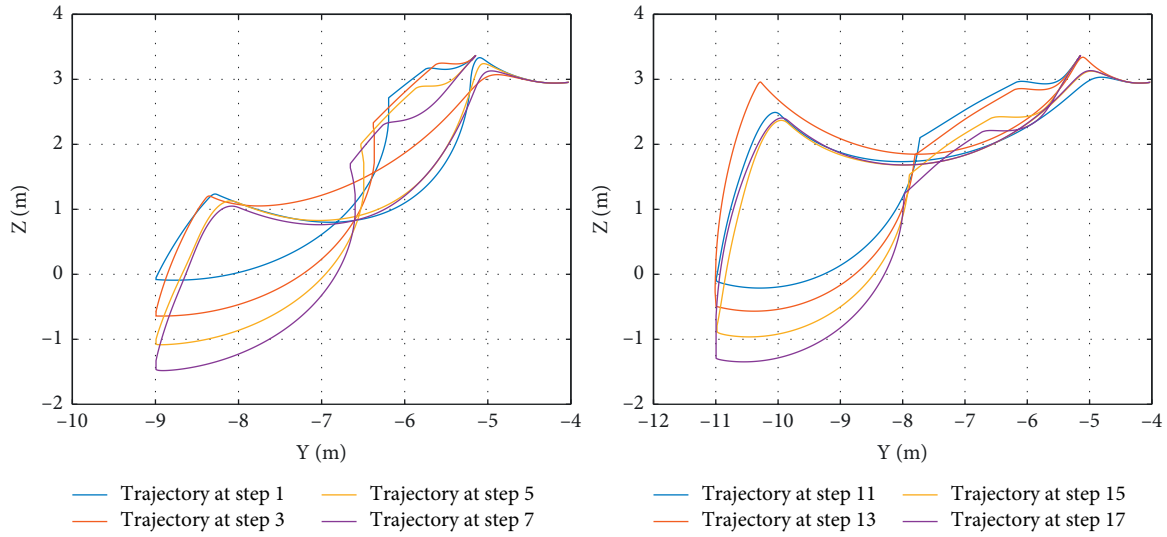


FIGURE 10: Trajectory in several excavating periods.

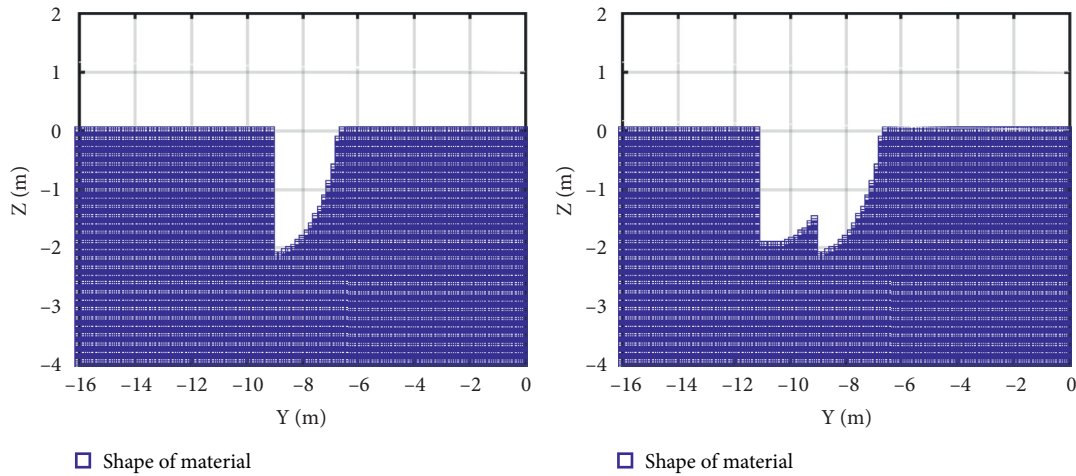


FIGURE 11: Shape of pile after 10<sup>th</sup> and 20<sup>th</sup> periods.

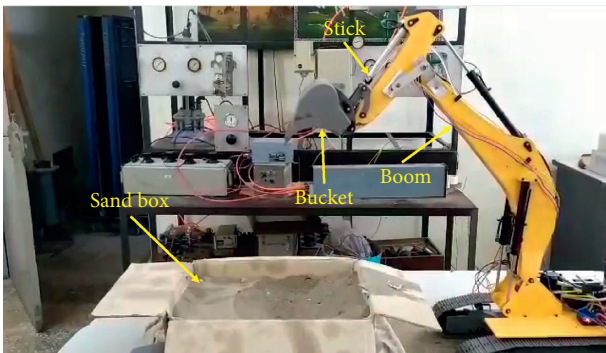


FIGURE 12: The laboratory size excavator.

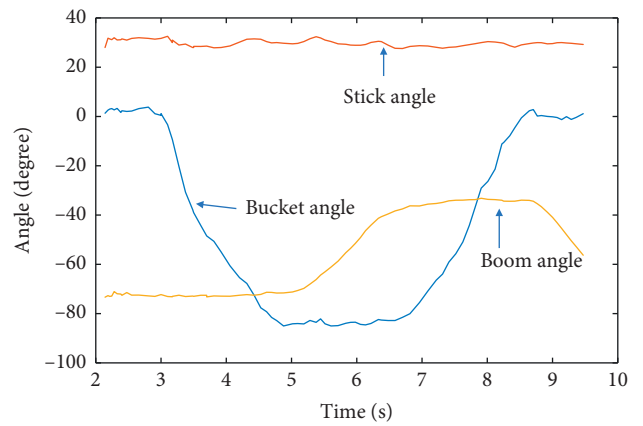


FIGURE 13: The trajectory of the excavator arm in the joint space in the 2<sup>nd</sup> period.

The simulation and experimental results illustrated in Figure 9 through 16 all indicate that the proposed algorithm has good performances in both computer simulation and

real-time experiment. The trajectory of the excavator arm is adjusted as the dug weight is out of the acceptable range; so, the weight error is kept small.



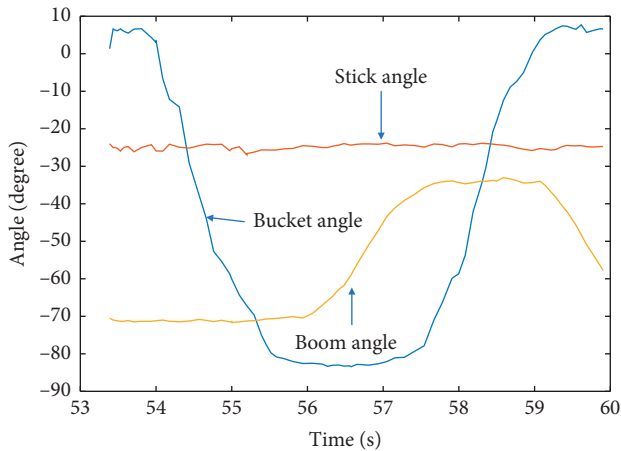


FIGURE 14: The trajectory of the excavator arm in the joint space in the 4<sup>th</sup> period.

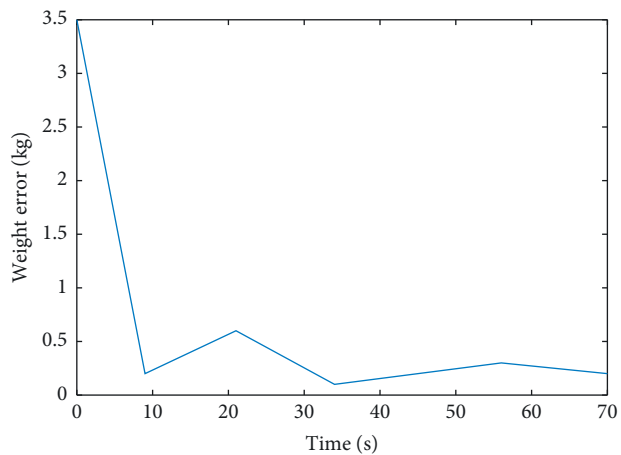


FIGURE 15: The weight error during working time.

## 4. Conclusions

The combination of B-spline technique and the fuzzy logic algorithm is proposed in this paper to generate the reference trajectory for the excavator arm robot. The design procedure includes two steps: firstly, the change of trajectory in comparison with the previous period is calculated to adapt with the change of the material during digging process; secondly, the reference trajectory for the current period is determined based on the result from the first step. The contributions of the proposed algorithm can be concluded as follows:

The trajectory of the excavator arm is changed automatically depending on the reduction of the material during digging process.

The weight of material which is dug in each period is kept in an acceptable range despite of reduction of the material (the shape of the material pile is changed).

The algorithm uses only the information of the trajectories in the previous periods and the error between dug weight and desired value to calculate the reference

trajectory. Some previous works solve this problem by using cameras [14] which may have the low reliability in the dust or dark environment.

The working of the robot arm in each cycle can reach the time and jerk optimization conditions.

The effectiveness of the scheme is verified through the computer simulation and laboratory size test-bed. The simulation and experimental results show that in each period of digging, the system can reach the minimum time and jerk by suitable selected parameters for the optimal function. Also, the dug weight is kept around the expected value regardless of the change of the environment.

## Data Availability

The datasets used in the present study are available in the article and the references.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## References

- [1] A. Gasparetto and V. Zanotto, "A technique for time-jerk optimal planning of robot trajectories," *Robotics and Computer-Integrated Manufacturing*, vol. 24, no. 3, pp. 415–426, 2008.
- [2] J. Huang, P. Hu, K. Wu, and M. Zeng, "Optimal time-jerk trajectory planning for industrial robots," *Mechanism and Machine Theory*, vol. 121, pp. 530–544, 2018.
- [3] S. Kucuk, "Optimal trajectory generation algorithm for serial and parallel manipulators," *Robotics and Computer-Integrated Manufacturing*, vol. 48, pp. 219–232, 2017.
- [4] H. Liu, X. Lai, and W. Wu, "Time-optimal and jerk-continuous trajectory planning for robot manipulators with kinematic constraints," *Robotics and Computer-Integrated Manufacturing*, vol. 29, no. 2, pp. 309–317, 2013.
- [5] R. Zhao and H.-K. Lee, "Fuzzy-based path planning for multiple mobile robots in unknown dynamic environment," *Journal of Electrical Engineering and Technology*, vol. 12, no. 2, pp. 918–925, 2017.
- [6] A. Maoudj, A. Hentout, B. Bouzouia, and R. Toumi, "On-line fault-tolerant fuzzy-based path planning and obstacles avoidance approach for manipulator robots," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 26, no. 05, pp. 809–838, 2018.
- [7] Z. Tao, L. Haodong, and D. Songyi, "Multi-robot path planning based on improved artificial potential field and fuzzy inference system," *Journal of Intelligent & Fuzzy Systems*, pp. 1–17, 2020.
- [8] G. Huanbing, L. Shouyin, and W. Tao, "Motion path planning of 6-DOF industrial robot based on fuzzy control algorithm," *Journal of Intelligent & Fuzzy Systems*, vol. 38, no. 3, pp. 1–10, 2020.
- [9] E. A. Merchán-Cruz and A. S. Morris, "Fuzzy-GA-based trajectory planner for robot manipulators sharing a common workspace," *IEEE Transactions on Robotics*, vol. 22, no. 4, pp. 613–624, 2006.
- [10] S. Wen, X. Hu, X. Lv, Z. Wang, and Y. Peng, "Q-learning trajectory planning based on Takagi-Sugeno fuzzy parallel

- distributed compensation structure of humanoid manipulator,” *International Journal of Advanced Robotic Systems*, vol. 16, no. 1, 2019.
- [11] A. Bakdi, A. Hentout, H. Boutami et al., “Optimal path planning and execution for mobile robots using genetic algorithm and adaptive fuzzy-logic control,” *Robotics and Autonomous Systems*, vol. 89, pp. 95–109, 2017.
- [12] Z. Maoudj, X. Li, S. Liu, and L. Jin, “A study on Trajectory planning of hydraulic robotic excavator based on movement stability,” in *Proceedings of International Conference on Ubiquitous Robots and Ambient Intelligence*, Xian, China, August 2016.
- [13] Y. B. Kim, J. Ha, H. Kang et al., “Dynamically optimal trajectories for earthmoving excavators,” *Automation in Construction*, vol. 35, pp. 568–578, 2013.
- [14] R. Kim, J. Knowles, and G. Danko, “Bucket trajectory classification of mining excavators,” *Automation in Construction*, vol. 31, pp. 128–139, 2013.
- [15] S. Lee, D. Hong, H. Park, and J. Bae, “Optimal path generation for excavator with neural networks based,” in *Proceedings of International Conference on Multisensor Fusion and integration for Intelligent Systems*, Seoul, Korea, August 2008.
- [16] N. T.-T. Vu, N. P. Tran, and N. H. Nguyen, “Adaptive neuro-fuzzy inference system based path planning for excavator arm,” *Journal of Robotics*, vol. 2018, Article ID 2571243, 7 pages, 2018.
- [17] A. Gasparetto and V. Zanotto, “A new method for smooth trajectory planning of robot manipulators,” *Mechanism and Machine Theory*, vol. 42, no. 4, pp. 455–471, 2007.