Modeling and Control of General Hydraulic Excavator for Human-in-the-loop Automation

Guangda Chen[1,4], Yinghao Gan[1], Jiayi Chen[2], Shuanwu Shi[3], Wei Chen[1], Yingfeng Chen[1], Rong Xiong[4] and Changjie Fan[1].

Abstract— **As labor shortages and safety regulations become more prominent, the need for human-in-the-loop automation of excavators is increasing. To meet this demand, we have developed a comprehensive modeling method for the excavator arm using nonlinear optimization approaches, including a simplified model that maps the task space to the joint space, as well as an equivalent model that maps the joint space to the actuator space. These models were then used to build a feedforward-PID joint velocity controller and a joint trajectory controller combined with position feedback, which forms the core of our proposed semi-automatic control system for the excavator arm. Our deployment scheme is simple and efficient, and has been deployed on two excavators of different makes and sizes. Experiments show that our deployment scheme performs well on both excavators, with an average error of 0.05 rad/s for the velocity controller and less than 5 cm for the trajectory controller. Using our semi-automatic system, we have completed demonstration experiments for precise digging and grading operations. A demonstration video can be found at https://youtu.be/N6I0WZGSF68.**

I. Introduction

Hydraulic excavators are extensively employed in construction, mining, and other industrial scenarios [1]. However, due to the scarcity of labor and the harsh and hazardous working conditions of excavators [2], fewer young people are opting to work as excavator operators, resulting in a shortage of skilled excavator operators [3]. Thus, there is an urgent need to enhance the level of excavator automation to reduce the threshold of excavator operation, minimize the rising labor costs, and improve the working conditions of excavator operators [4].

Excavator automation has been an area of active research for more than three decades. Early research projects such as LUCIE [5], ALS [6], and THOR [7] were representative of the field at the time. While recent research, such as AES [8] and HEAP [9], has shown rapid development in the technologies required for a fully autonomous excavator system, the complexity of the task and the uncertainty of the environment still pose significant challenges to large-scale application [10]. Consequently, teleoperation of excavators remains highly relevant and has evolved from simple remote cockpit mode [11] to semi-automatic operation with human-in-the-loop automation [12] [13]. In this paper, we present a general control system for the excavator arm, which includes kinematic modeling and calibration. This system can be used for both fully autonomous excavator operation and semi-automatic operation.

Kinematic modeling forms the foundation for implementing planning and control of the excavator arm [14]. Currently, there are two main approaches to constructing kinematic models in related works: obtaining precise 3D model information from the excavator manufacturer or directly measuring it using external equipment. For instance, the HEAP automatic excavator group has been working closely with excavator manufacturers for more than five years [9]. However, obtaining 3D model information of various excavators from different manufacturers is nearly impossible. Moreover, external measuring equipment such as total stations can be expensive and inconvenient to operate, and the measurement points may be obscured or occluded. Consequently, in the field of excavator automation, there is an urgent need for an accurate and general method to model and calibrate the excavator arm without depending on excavator manufacturers.

The motion control of the pressure-compensated mobile hydraulic valves that have a significant dead zone exhibits strong nonlinearity [15]. Velocity feedforward is used increasingly with proportional-integral-derivative (PID) controllers in heavy-duty hydraulic manipulators [16], which is based on inferring the required control values of the hydraulic valve from a steady-state actuator velocity-valve control value model based on the desired actuator velocity [17]. HEAP excavator establishes a relationship between the cylinder speed and the valve control value by directly measuring the cylinder speed with draw wire sensors. However, installing a draw wire sensor for each hydraulic cylinder is inconvenient and involves additional hardware costs. A more common approach is to use an inclination sensor to measure the joint velocity, and the relationship between the linear speed of the cylinder and the control value is still established by the look-up table method, in which the cylinder speed is calculated based on the complex structural information of the joint and the angular velocity measured by the inclination sensor [18]. A more important challenge in building velocity feed-forward models is the amount of data required for accurate system identification [19]. It is difficult to spend hours or days experimenting with each individual machine to collect enough data to learn velocity feed-forward models

[1] Fuxi Robotics in NetEase, Hangzhou, P.R. China. {chenguangda, ganyinghao, chenwei25, chenyingfeng1, fanchangjie}@corp.netease.com.
[2] College of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou, P.R. China.
[3] Department of Automation, Zhejiang University of Technology, Hangzhou, P.R. China.
[4] State Key Laboratory of Industrial Control and Technology, Zhejiang University, Hangzhou, P.R. China. rxiong@zju.edu.cn.

based on neural networks (NNs) [20], [21] and do so within safe limits. Therefore, there is an urgent need in the field for a relatively accurate, universally applicable, and rapidly deployable method to model and calibrate excavator joints in order to acquire velocity feedforward with the least amount of joint motion data.

In order to address the challenges associated with excavator automation transformation, we propose an inexpensive, easily deployable, human-in-the-loop excavator automation solution. To modify the hardware, we use economical solenoid valves instead of pilot valves to control the excavator cylinder and only four inclination sensors to obtain real-time position and velocity data of the excavator arm. We have developed a modeling approach for the excavator arm, which includes a simplified arm model from task space to joint space, and a joint equivalent model from joint space to actuator space, which serves as the velocity feedforward of the controller. By utilizing nonlinear optimization techniques, we were able to accurately calibrate the parameters of a simplified arm model without requiring collaboration from the excavator manufacturer. With minimal joint motion data, we used automatic two-stage least squares optimization to construct a joint equivalent model. Using this model, we have implemented precise joint velocity and trajectory controllers for the excavator arm by leveraging velocity feedforward models and feedback control techniques. We have demonstrated the performance and versatility of our control system by applying our proposed modeling and control system to two different excavators. Furthermore, we have designed the arm control system for semi-automatic teleoperation tasks, based on real-time trajectory control in Cartesian space and the bucket global locking controller, which significantly reduces the excavator teleoperation threshold. Ultimately, we have achieved semi-automatic digging and grading operations by using our semi-automatic system.

In summary, the contributions of this work are as follows:

- A general kinematic modeling and calibration method of excavator arm based on nonlinear optimization is proposed without relying on information from the excavator manufacturer.
- A general modeling and automatic calibration method of excavator joint based on least squares optimization is proposed to obtain the velocity feedforward with minimal joint motion data. By combining the velocity feedforward model and PID control, an accurate velocity and trajectory controller is constructed.
- A human-in-the-loop automation framework is proposed. By realizing the Cartesian-space teleoperation and the bucket locking function, the complexity and threshold of the excavator operation are effectively reduced.
- The solution was tested on two actual excavators of different models and sizes, and the performance of the controllers and semi-automatic actions were evaluated.

The rest of this paper is organized as follows. A general modeling and calibration approach for excavators is described in Section II and Section III respectively. Section IV introduces our control system of excavator arm. Section V provides experimental details and results on two excavators in the real world, followed by conclusions in Section VI.

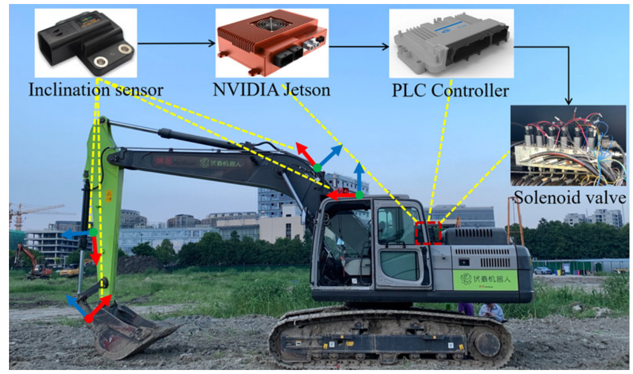


Fig. 1: The modifications made to a standard excavator's hardware and main components are highlighted. The inclination sensors' location and orientation are indicated by the coordinate systems represented by the red and blue arrows.

II. Excavator Modeling

The kinematic modeling of excavator arm is the primary prerequisite for excavator automation. This section begins by briefly introducing our hardware system. We then outline the general modeling approach of the excavator arm, which involves two parts: modeling from task space to joint space and from joint space to actuator space. As shown in Figure 1, a group of solenoid valves are installed in parallel on the pilot oil circuit of our modified excavators, enabling operation either from the cockpit or by a computer. Additionally, we've installed inclination sensors on the boom, stick, bucket, and cockpit joints, with the red and blue arrows in Figure 1 indicating their respective locations and orientations. A computing unit located inside the cockpit collects sensor data and calculates the control value of the solenoid valve. The PLC controller receives the control value from the computing unit and generates a specified pulse-width modulation (PWM) signal to control the solenoid valve.

A. *Simplified arm model*

We have proposed a simplified model of the excavator arm that can calculate the real-time attitude of the bucket by combining information from the inclination sensors. The main components of the excavator arm have been simplified as a rigid link, with the joints being represented as hinges, as shown in the bottom left of Figure 2. To obtain the angle of each joint, it is necessary to calibrate the three fixed angles (α_1, α_2, α_3) between the inclination sensor and the arm part. The joint angles of the boom, stick, and bucket are denoted as θ_1, θ_2, and θ_3, respectively. The inclination sensor can measure the angle between the sensor orientation and the horizontal plane, represented by ϕ_1, ϕ_2, ϕ_3, and ϕ_4. Then the

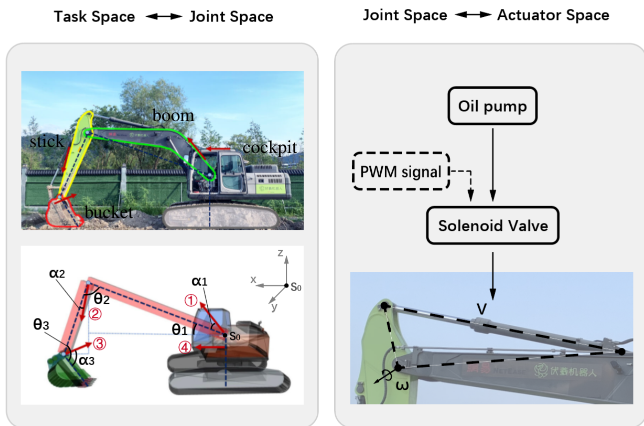


Fig. 2: Simplified arm model (left, from task space to joint space) and equivalent joint model (right, from joint space to actuator space).

angle of each joint is calculated as

$$\begin{cases} \theta_1 = \phi_1 - \phi_4 - \alpha_1 \\ \theta_2 = \pi - \phi_1 + \phi_2 + \alpha_1 + \alpha_2 \\ \theta_3 = -\phi_2 - \phi_3 - \alpha_2 + \alpha_3 \end{cases} \tag{1}$$

The position of the end point of the stick in the arm coordinate system S_0 is

$$\begin{cases} x = l_1 \cdot \cos(\theta_1) + l_2 \cdot \sin(\theta_1 + \theta_2 - \pi/2) \\ z = l_1 \cdot \sin(\theta_1) - l_2 \cdot \cos(\theta_1 + \theta_2 - \pi/2) \end{cases} \tag{2}$$

where l_1, l_2 are the lengths of the boom and stick links, respectively. Derivation of the above equation leads to the Jacobian matrix $\mathbf{J}$ in the S_0 frame as

$$\mathbf{J} = \begin{bmatrix} l_2 \sin(\theta_1 + \theta_2) - l_1 \sin(\theta_1) & l_2 \sin(\theta_1 + \theta_2) & 0 \\ l_1 \cos(\theta_1) - l_2 \cos(\theta_1 + \theta_2) & -l_2 \cos(\theta_1 + \theta_2) & 0 \\ -1 & -1 & 1 \end{bmatrix} \tag{3}$$

The Jacobian matrix $\mathbf{J}$ can convert the angular velocity of the joint space and the velocity of the task space to each other, that is

$$\begin{bmatrix} \dot{x} \\ \dot{z} \\ \dot{\Theta} \end{bmatrix} = \mathbf{J} \cdot \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \tag{4}$$

where $\dot{\Theta}$ is the angular velocity of the bucket in the arm coordinate system, $\dot{x}$ and $\dot{z}$ are the horizontal and vertical linear velocities of the end point of the stick link respectively. ω_1, ω_2 and ω_3 are the angular velocities of the boom, stick and bucket joints respectively. Note that the complete kinematic relationship of the excavator that may be required for actual operation also includes the transformation between the arm coordinate system and the chassis coordinate system. This work only focuses on the relationship under the arm coordinate system.

B. Equivalent joint model

In this paper, a two-layer modeling is carried out between the control value PWM (p) and the joint angular velocity (ω), and the first layer is the hypothetical modeling of the PWM to the cylinder linear velocity (v), and another layer is the geometric modeling of the cylinder linear velocity to the joint angular velocity. Hydraulic equipment controls the liquid flow rate and pressure inside the hydraulic cylinder by controlling the opening and closing of the valve, so the size of the valve opening corresponds to the fluid flow rate and the linear speed of the hydraulic cylinder, that is, $v = f(p)$. This simplified modeling approach is particularly suitable for load-sensitive excavators [9]. As shown in the lower right picture of Figure 2, the joint of the excavator is a transmission structure composed of multiple mechanical links. Therefore, the linear velocity of the cylinder and the joint angular velocity also have a Jacobian matrix related to the joint structure, and the shape of the joint structure is related to the joint angle θ, that is, $\omega = g(\theta) \cdot v$. And the relationship between the joint velocity and the control value PWM is:

$$\omega = g(\theta) \cdot f(p) \tag{5}$$

So far, the mapping relationship from actuator space to task space can be expressed as:

$$\begin{bmatrix} \dot{x} \\ \dot{z} \\ \dot{\Theta} \end{bmatrix} = \mathbf{J} \cdot \begin{bmatrix} g_1(\theta_1) \cdot f_1(p_1) \\ g_2(\theta_2) \cdot f_2(p_2) \\ g_3(\theta_3) \cdot f_3(p_3) \end{bmatrix} \tag{6}$$

III. Model Calibration

In this section, calibration methods are proposed to accurately obtain the model parameters mentioned in Section II. Specifically, the model parameters include the kinematic parameters of the simplified arm model (the lengths of the boom and stick links l_1, l_2 and the angles between the inclination sensors and the arm links α_1, α_2), the function $g(\theta)$ and $f(p)$ in equivalent joint model. Note that the proposed calibration method is the key to self-deployment of excavator automation without relying on excavator manufacturers.

A. Arm model calibration

The real-time position of the end point of the stick can be calculated based on the real-time inclination sensor readings and Equation (1) and (2) mentioned in Section II-A. In this subsection, we propose an automatic and accurate calibration method to obtain accurate model parameters by minimising the difference between the measured displacement of the end point of the stick and the result calculated from the simplified arm model. During the calibration implementation phase, we controlled the end point of the stick link to different positions (x_i, z_i), $i \in [1, ..., n]$. Then used a specific external device to measure the position change of the end point $(\Delta x_{ij}, \Delta z_{ij})$. Combining the inclination sensor readings at the two positions $\vartheta_i = (\theta_1^i, \theta_2^i, \theta_4^i)$ and $\vartheta_j = (\theta_1^j, \theta_2^j, \theta_4^j)$, we can construct two optimization residual terms separately

$$\begin{cases} R_{ij}^x = \left\| \Delta x_{ij} - (x(\vartheta_i) - x(\vartheta_j)) \right\|^2, \\ R_{ij}^z = \left\| \Delta z_{ij} - (z(\vartheta_i) - z(\vartheta_j)) \right\|^2. \end{cases} \tag{7}$$

where (x_i, z_i) and (x_j, z_j) are the positions containing (l_1, l_2, α_1, α_2) calculated according to the arm model. In order to

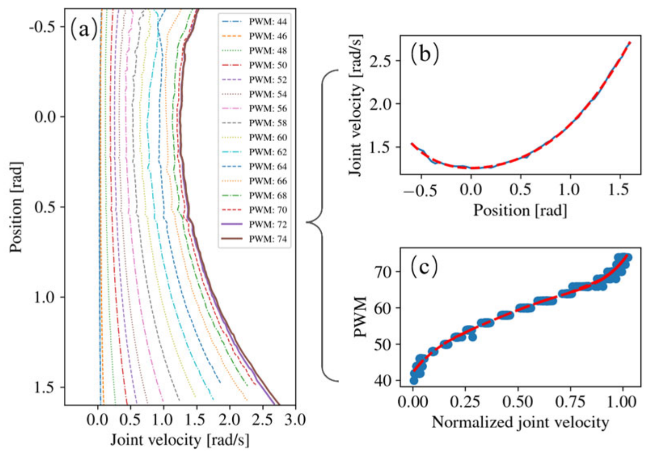


Fig. 3: (a) The relationship between velocity and angle of stick joint under different PWM values; (b) Fitted joint structure function; (c) Fitted solenoid curve.

solve for the model parameters, at least four residual terms need to be constructed, i.e. $n \geq 3$. We employ stochastic gradient descent optimization to minimize the sum of all residual terms. That is, the loss function is

$$l = \sum_{i=1}^{n-1} (R^x_{i,i+1} + R^z_{i,i+1}), \quad s.t. l_1 > 0, l_2 > 0, -\pi \leq \alpha_1, \alpha_2 \leq \pi \tag{8}$$

Note that our method only needs to measure the displacement of the stick's end point instead of each position coordinate, which simplifies the measurement process. The initial values of the parameters can be chosen arbitrarily within a reasonable interval. As the more displacement data is collected, the more residual terms are constructed, and the more accurate the optimization result is. Since the parameter α_3 only affects the attitude information of the bucket and does not affect the Jacobian matrix $\mathbf{J}$, we do not have high requirements on the accuracy of the parameter α_3. In this work, the value of α_3 is estimated by manual debugging to make the bucket attitude established by the model similar to the real bucket attitude.

B. Joint model calibration

The equivalent joint model describes the mapping relationship between the control value PWM (p) and the joint angular velocity ω. The feedforward look-up table is the common method for solenoid valve actuation modeling. However, manually identifying the feedforward model for each valve-actuator pair is often time-consuming and error-prone. For this practical reason, we propose an automatic least squares optimization method to build $f(p)$ and $g(\theta)$ in two stages, respectively.

We use multiple constant PWM values to control the joints of the excavator to move fully within the joint limits. During the process, the angle and angular velocity data of the joint at each timestamp are collected to form a containing (p, θ, ω) data set Ψ, and the result is shown in Figure 3(a). We select the data $\Psi(p = p_m)$ corresponding to one of the specific PWM (p_m), when the PWM quantity is constant, the cylinder linear velocity $v_m = f(p_m)$ is constant. As shown in Figure 3(b), we can fit the relationship $\omega_m(\theta) = v_m \cdot g(\theta)$ between angular velocity ω_m and joint angle θ using the data $\Psi(p = p_m)$. Then we process the angular velocity ω in the data set Ψ as follows

$$\frac{\omega}{\omega_m(\theta)} = \frac{f(p) \cdot g(\theta)}{v_m \cdot g(\theta)} = \frac{f(p)}{v_m}$$

Then we can get the normalized cylinder linear velocity

$$v_n(p) = \frac{f(p)}{v_m}$$

and use polynomial least squares optimization to fit these data, the result is shown in Figure 3(c). After obtaining $\omega_m(\theta)$ and $v_n(p)$, the equivalent joint model can be obtained:

$$\omega = g(\theta) \cdot f(p) = (v_m \cdot g(\theta)) \cdot \frac{f(p)}{v_m} = \omega_m(\theta) \cdot v_n(p) \tag{9}$$

Then we can predict the corresponding control value PWM (p_f) according to the expected angular velocity ω_d and the current joint angle θ,

$$p_f = v_n^{-1}\left(\frac{\omega_d}{\omega_m(\theta)}\right) \tag{10}$$

where v_n^{-1} is the inverse function of v_n. It is worth noting that our proposed two-stage modeling approach effectively reduces the dimensionality of inference while minimizing the complexity of training. Furthermore, this approach requires only a small amount of training data and does not necessitate manual data cleaning.

IV. Control System

We have established a human-in-the-loop control framework for the complex operating environment of excavators, as illustrated in Figure 4. Our control system allows for both automation and semi-automatic teleoperation of the excavator, and we have proposed two teleoperation methods to accommodate operators with varying operating habits: Cartesian-space teleoperation and joint-space teleoperation. In Cartesian-space teleoperation mode, operators can directly move the tip of the arm, and the trajectory planner will generate the corresponding joint space trajectory based on the remote control's command. For joint-space teleoperation, operators can send a teleoperation command for each joint, and the corresponding linearized PWM command is generated based on the joint command and the solenoid valve model $f(p)$ described in Section III-B. This model ensures that the remote control command is proportional to the linear speed of the cylinder.

We will now introduce the key components of the framework, including the joint velocity and trajectory controllers. In addition, we propose a practical bucket locking controller that utilizes the trajectory controller to maintain the inclination angle of the bucket relative to the horizontal plane after the operator sets the bucket locking. Our trajectory planner integrates several planners from MoveIt [22] to optimize performance.

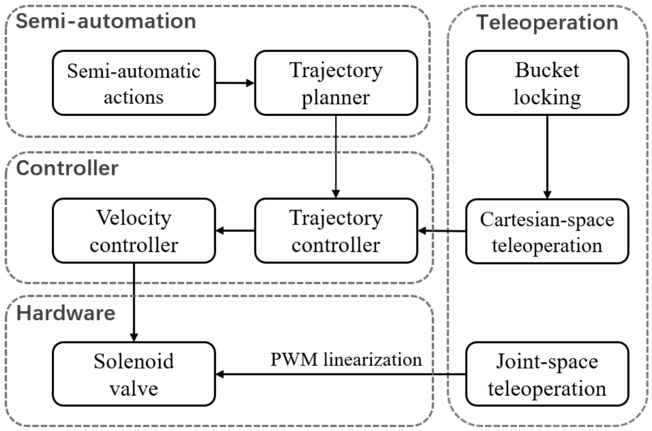


Fig. 4: Work-flow diagram of the arm control system.

A. Joint velocity controller

Our proposed feedforward-PID velocity controller includes an incremental PID term for feedback control, which is determined by the difference between the actual normalized angular velocity and the desired normalized angular velocity of the joint. This approach allows us to improve the accuracy of the controller by taking into account both the feedforward and feedback control mechanisms. The control value of PID controller at time t is

$$p_{pid} = \sum_{t=0}^{n} \Delta pid\left(\frac{\omega_d(t)}{\omega_m(\theta)}, \frac{\omega(t)}{\omega_m(\theta)}\right) \tag{11}$$

where ω_d and ω represent the expected angular velocity and the actual angular velocity of the joint, respectively. It is worth noting that using normalized velocity deviation as feedback can address the nonlinearity of angular velocity signals with respect to joint angle. The final control value p' of the velocity controller is the output of the velocity feedforward plus the output of the PID controller.

$$p' = p_f + p_{pid} \tag{12}$$

B. Joint trajectory controller

For industrial robotic manipulators, the trajectory controller only needs to control the joint velocity that needs to be executed at the current time from the planned joint trajectory, which requires precise joint velocity controllers and joint actuators. However, it is almost impossible to construct an absolutely accurate joint velocity controller according to the looseness and dynamics of excavator actuators. Therefore, we correct the historical execution deviation of the trajectory by adding a position feedback term in the trajectory controller,

$$\omega'(t) = \omega_d(t) + pid\left(\theta_d(t), \theta_t\right) \tag{13}$$

The velocity control value $\omega'(t)$ is equal to the velocity $\omega_d(t)$ that the trajectory planner expects to control at time t plus the output of PID controller according to the current joint position θ_t and the desired joint position $\theta_d(t)$. Adding a position closed loop can improve the position deviation in the process of trajectory tracking, which is very important for the excavator arm to perform precise tasks.

Unlike the trajectory generated by the trajectory planner, the trajectory of Cartesian-space teleoperation is generated in real time. The control law of the Cartesian-space teleoperation is designed as follows:

$$\begin{aligned}\omega'(t) =& \omega_d(t) \\ &+ pid\left(\theta_j + \sum_{i=j+1}^{t}\left(\omega_d(i-1)\cdot(t_i - t_{i-1})\right), \theta_t\right)\end{aligned} \tag{14}$$

Specifically, the joint angle $\theta_d(t)$ that the joint is expected to reach at time t is the expected trajectory position generated by the expected angular velocity $\omega_d(i)$ in a certain period $i \in (j,t)$. Among them, the expected angular velocity $\omega_d(t)$ of each joint at time t is calculated according to the Jacobian matrix in Equation (3) and the real-time linear velocity command in Cartesian space. Note that since the cumulative tracking error of the joint angle may lead to a large deviation between the control velocity and the velocity expected by the remote operation command, we will periodically eliminate the cumulative error of trajectory tracking (i.e. $j = t$).

C. Bucket-locking controller

Different from the above trajectory-based teleoperation method, the global absolute deviation of the bucket locking can be sensed in real time. The controller is designed as follows

$$\begin{aligned}\omega'_{bu}(t) =& \bar{\omega}_{bu}(t) \\ &+ pid\left(\theta_{bu}(i) - \Delta\theta_{bo}(t) - \Delta\theta_{st}(t), \theta_{bu}(t)\right)\end{aligned} \tag{15}$$

where $\theta_{bu}(i)$ represents the joint angle of the bucket when the bucket is locked at time i. $\Delta\theta_{bo}(t) = \theta_{bo}(t) - \theta_{bo}(i)$ and $\Delta\theta_{st}(t) = \theta_{st}(t) - \theta_{st}(i)$ represent the angle change of the boom joint and the stick joint respectively. Therefore, the term $\theta_{bu}(i) - \Delta\theta_{bo}(t) - \Delta\theta_{st}(t)$ indicates the target angle that the bucket joint should reach in order to keep the bucket locked at the current time t. While the position PID term can ensure the global bucket locking, it is laggy to keep the bucket locked during real-time teleoperation based solely on position feedback, so we added $\bar{\omega}_{bu}(t) = -\bar{\omega}_{bo}(t) - \bar{\omega}_{st}(t)$, which allows the controller to respond in advance to the new teleoperation velocity of boom and stick $(\bar{\omega}_{bo}(t), \bar{\omega}_{st}(t))$. The sum of these two items can not only realize the timely response of the bucket to the instantaneous remote control command, but also eliminate the accumulated angle error.

V. Experiments

To demonstrate the effectiveness and versatility of our excavator automation transformation method, we conducted hardware transformation and software deployment tests on a 25t Zoomlion ZE205E-10 excavator and a 7t XCMG XE75D excavator, respectively. Our experiments consisted of four parts. Firstly, we performed calibration tests on both excavators of different sizes. Secondly, we tested the joint velocity controller proposed in this method. In the third part, we evaluated the trajectory tracking performance of the

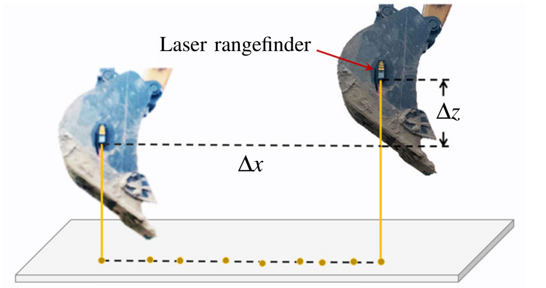


Fig. 5: Calibrating the arm kinematics of an excavator using a laser rangefinder to measure the displacement of the bucket.

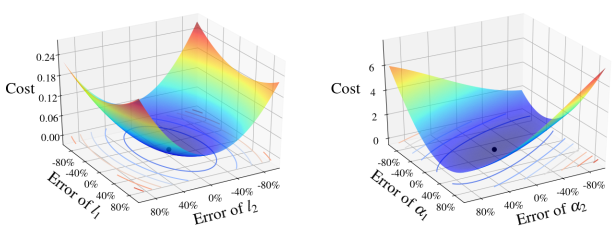


Fig. 6: The loss function surfaces represent the relationship between the error of the parameters and the resulting loss.

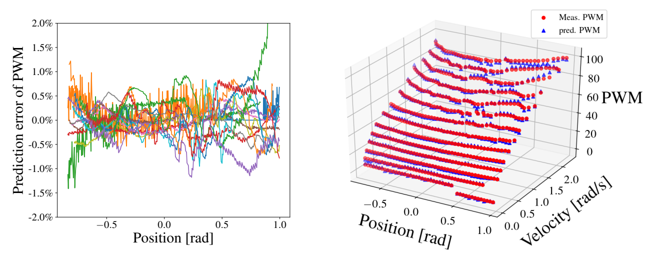


Fig. 7: Error of PWM predicted by feedforward model.

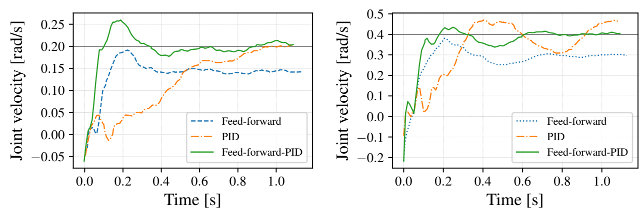


Fig. 8: Velocity control results of step signals.

proposed trajectory controller. Finally, we demonstrated the excavator's ability to perform semi-automatic operations in a real scenario. It is worth noting that a demonstration video of our method is available at https://youtu.be/N6I0WZGSF68.

A. Model calibration

A low-cost laser rangefinder was chosen to measure the displacement of the excavator bucket by taking into account the cost and calibration accuracy. However, it should be noted that our method is not limited to the use of laser rangefinders, other measurement tools such as total stations and RTK can also be used. Figure 5 illustrates the collection of displacement data from the end point of the stick link using a laser rangefinder attached rigidly to the outside of the bucket. At each measurement position, we kept the laser rangefinder perpendicular to the plane and marked the intersection with the plane to measure the distance Δx_{ij} of the intersection point of i and j. The height difference measured by the laser rangefinder represented the z-axis displacement Δz_{ij} of the end point of the stick link.

Calibration data was collected from the Zoomlion ZE205E-10 excavator using the method described above. As shown in Figure 6, the surface of the loss function plotted with respect to the parameters showed a unique minimum value that corresponded to the value of the parameters that approximated the optimal solution. The calibrated length of the boom link was found to be 5.746 m, which was 4.6 cm different from the official size of 5.7 m. The calibrated length of the stick link was 2.938 m, which was 1.3 cm different from the official size of 2.925 m. While our method used a simple and low-cost laser rangefinder scheme to collect calibration data, the displacement measurements at the tip of the arm had some error. We also calibrated the XCMG XE75D excavator and found that the difference in the length of the boom was 2 cm, the difference in the length of the stick was 2 mm, and the differences in the parameters α_1, α_2 were less than 1 degree when compared with the parameters measured manually. Our experiments have demonstrated that our method is applicable to different excavators and can accurately calibrate each kinematic parameter of the excavator arm.

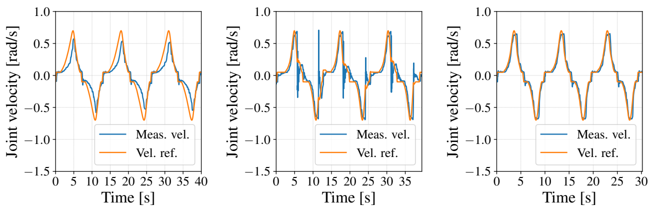
In order to build the joint model, we utilized the calibration method outlined in Section III-B. The model's predicted PWM values were then compared to the collected data, with the resulting prediction errors shown in Figure 7. We found that the prediction error was predominantly within 1%. Despite the model's strong performance on the dataset, it should be noted that the actual results of control using solely this model were not satisfactory due to factors such as joint friction and the impact of gravity in different postures.

B. Velocity controller

We utilized a combination of step signals and regularly changing signals to evaluate the velocity tracking capabilities of the feedforward-PID velocity controller (FF-PID) proposed in Section IV-A, the open-loop feedforward controller (FF) and the pure PID controller (PID) were used as the control group for comparative experiments. The control frequency of all controllers is set to 100 Hz. Figure 8 shows the tracking results of the three controllers controlling the stick joint to track step signals with different amplitudes. We count the average error in the whole process, and the average tracking error of the FF-PID controller is 0.051 rad/s, which is smaller than that of the FF controller (0.141 rad/s) and the PID controller (0.070 rad/s). We further analyze the transient and steady-state performance of several controllers using the following classical performance metrics of control system:

TABLE I: Control performance of velocity controllers

Velocity	Controller	t_r	t_s	e_{ss}
0.2	FF	0.096	**0.537**	-0.056
	PID	0.786	2.517	0.009
	FF-PID	**0.041**	0.627	$\mathbf{-0.002}$
0.4	FF	0.137	0.668	-0.103
	PID	0.244	0.926	0.065
	FF-PID	**0.05**	**0.592**	**0.003**
0.6	FF	0.245	0.701	-0.093
	PID	0.231	0.941	0.004
	FF-PID	**0.13**	**0.612**	$\mathbf{-0.002}$



(a) FF controller (b) PID controller (c) FF-PID controller

Fig. 9: The velocity tracking results of three different control methods.

- Rise time: t_r: the time it takes to rise from 10% of the final value to 90% of the final value.
- Setting time: t_s: the minimum time required to reach and remain within ±5% of final value
- Steady-state error: e_{ss}: the deviation of the steady state from the desired state.

Table I presents the performance of each controller, with the FF PID controller outperforming the others in terms of having a shorter rise time and smaller steady-state error. Its transient and steady-state performance is also significantly better compared to the other controllers. Figure 9 illustrates the tracking effect of the three controllers in tracking the period-varying signal. The FF-PID controller demonstrates superior tracking performance compared to the FF controller and PID controller. Its average tracking error of 0.052 rad/s is 53% lower than the FF controller's error of 0.111 rad/s and 44% lower than the PID controller's error of 0.093 rad/s.

C. Trajectory controller

Accurately tracking the trajectory of the excavator arm's tip is a crucial metric for evaluating the excavator's automation performance. To accomplish this, we employed Kinovea (https://www.kinovea.org), a video motion analysis software, to track the pixel coordinates of the stick link's endpoint. By utilizing calibrated camera and laser sensor parameters, we were able to project the tracking pixels onto a 3D laser coordinate system, allowing us to measure the excavator arm's actual trajectory. In this subsection, the feedforward-PID controller is used to execute the velocity command generated by the trajectory controller. To begin, we direct the endpoint of the stick link to track a horizontal line and a vertical line while maintaining the bucket posture. Our

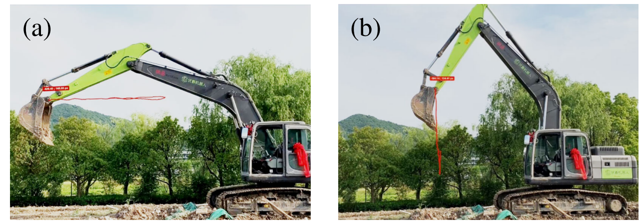


Fig. 10: The execution performance of the horizontal line and vertical line trajectory, the red lines represent the actual motion trajectory.

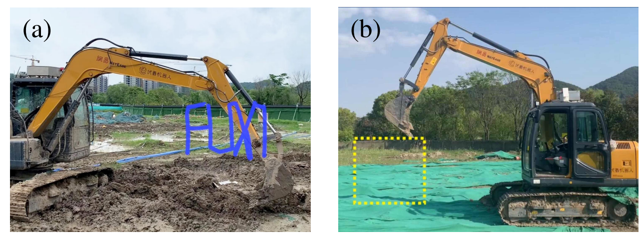


Fig. 11: Our XCMG excavator performs complex trajectory of the characters "FUXI" (a), and executes rectangular Cartesian-space teleoperation commands (b).

joint trajectory controller proves to perform exceptionally well for the line tracking task, as illustrated in Figure 10. The outcomes demonstrate that both trajectories have a maximum error of only 5 cm. Moreover, Figure 11(a) presents the results of the arm tip tracing the trajectory of the characters "FUXI", which proves that our controller excels at even complex tracking tasks. Note that the high-precision trajectory execution not only demonstrates the precision of the controllers but also serves as evidence of the accuracy of the kinematic parameter calibration.

To assess the trajectory controller's tracking performance in remote control mode, we sent a rectangular remote control command in Cartesian space, as illustrated in Figure 11(b). The average error of the trajectory during the tracking teleoperation commands was 2.73 cm, indicating excellent tracking performance of our Cartesian-space teleoperation controller. Moreover, we conducted a bucket locking experiment in remote control mode. We sent a varying Cartesian space remote control signal, as shown in Figure 12, and expected the bucket's angle to remain constant with respect to the horizontal. In the initial stage of low-speed remote control ($t \in [0, 15]$), the average angle error of bucket locking was 0.764 rad. In the mid-term remote control stage with high and drastically changing speed ($t \in [15, 70]$), the average angle error was 3.274 rad. However, the bucket could still maintain the initial attitude with a deviation of 0.24 rad.

D. Semi-automatic Tasks

Excavators are frequently tasked with digging and grading operations. To demonstrate the effectiveness and efficiency of our excavator automation solution in real-world scenarios, we designed two sets of experiments for digging and grading

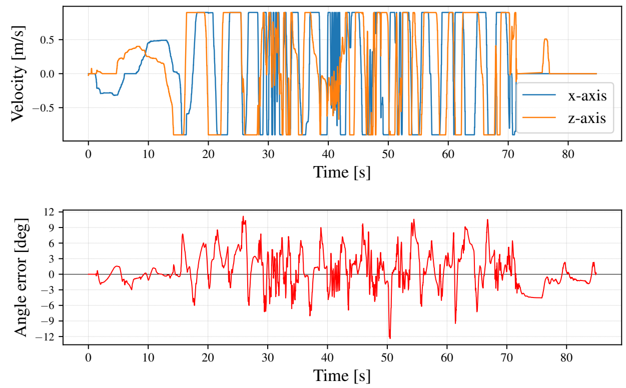


Fig. 12: The performance of the bucket locking controller (top: teleoperation commands, bottom: the corresponding angle error).

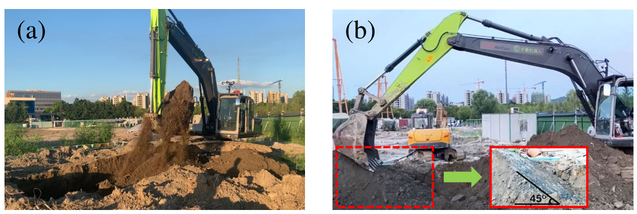


Fig. 13: Our proposed control system enables efficient and precise semi-automatic digging (a) and grading operations (b).

tasks. Figure 13(a) illustrates the semi-automatic digging process. Once the excavation point and depth are given, the trajectory planner will design a path for digging, which the excavator will follow to complete the task. The execution time averages around 12 seconds, which is faster than our statistics results for human manipulation data (15-18 seconds). When the bucket is locked, we use the trajectory planner and trajectory controller in Cartesian space to direct the bucket along the soil slope. Figure 13(b) shows the condition of the soil mound before and after the grading operation. This semi-automatic operation method requires only the starting position and slope angle to complete the task, greatly reducing operational difficulty.

VI. Conclusions

This paper presents a novel task-space arm control system for a general hydraulic excavator, which incorporates accurate kinematics modeling and a robust calibration method. The proposed calibration method for the simplified and general excavator arm model utilizes nonlinear optimization technology, which is both straightforward and manufacturer-independent. We establish a velocity feedforward model using model-free least squares optimization to relate joint velocity to control values. By integrating the velocity feed-forward model with PID feedback control technology, we achieve precise joint velocity and trajectory control for the excavator arm. Finally, we have achieved semi-automatic digging and grading operations by using our semi-automatic system. In future work, we will use hydraulic pressure sensors to address the velocity feed-forward modeling of load-insensitive excavators and to improve the robustness of the control system in the case of interaction with the environment during excavation operations.

References

[1] S. Dadhich, U. Bodin, and U. Andersson, "Key challenges in automation of earth-moving machines," *Automation in Construction*, vol. 68, pp. 212–222, 2016.

[2] K. H. Petersen, N. Napp, R. Stuart-Smith, D. Rus, and M. Kovac, "A review of collective robotic construction," *Science Robotics*, vol. 4, no. 28, p. eaau8479, 2019.

[3] J. Kim, D.-e. Lee, and J. Seo, "Task planning strategy and path similarity analysis for an autonomous excavator," *Automation in Construction*, vol. 112, p. 103108, 2020.

[4] M. Ramezani and S. Tafazoli, "Using artificial intelligence in mining excavators: Automating routine operational decisions," *IEEE Industrial Electronics Magazine*, vol. 15, no. 1, pp. 6–11, 2020.

[5] D. A. Bradley and D. W. Seward, "The development, control and operation of an autonomous robotic excavator," *Journal of Intelligent and Robotic Systems*, vol. 21, no. 1, pp. 73–97, 1998.

[6] H. N. Cannon, "Extended earthmoving with an autonomous excavator," Master's thesis, Carnegie Mellon University Pittsburgh, Pa, USA, 1999.

[7] D. Schmidt, M. Proetzsch, and K. Berns, "Simulation and control of an autonomous bucket excavator for landscaping tasks," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2010, pp. 5108–5113.

[8] L. Zhang, J. Zhao, P. Long, L. Wang, L. Qian, F. Lu, X. Song, and D. Manocha, "An autonomous excavator system for material loading tasks," *Science Robotics*, vol. 6, no. 55, p. eabc3164, 2021.

[9] D. Jud, S. Kerscher, M. Wermelinger, E. Jelavic, P. Egli, P. Leemann, G. Hottiger, and M. Hutter, "Heap-the autonomous walking excavator," *Automation in Construction*, vol. 129, p. 103783, 2021.

[10] J. S. Lee, Y. Ham, H. Park, and J. Kim, "Challenges, tasks, and opportunities in teleoperation of excavator toward human-in-the-loop construction automation," *Automation in Construction*, vol. 135, p. 104119, 2022.

[11] H. Sulaiman, M. N. A. Saadun, and A. A. Yusof, "Modern manned, unmanned and teleoperated excavator system," *Journal of Mechanical Engineering and Technology (JMET)*, vol. 7, no. 1, 2015.

[12] J. S. Lee and Y. Ham, "Exploring human-machine interfaces for teleoperation of excavator," in *Proceedings of the Construction Research Congress*, 2022, pp. 757–765.

[13] S. Okishiba, R. Fukui, M. Takagi, H. Azumi, S. Warisawa, R. Togashi, H. Kitaoka, and T. Ooi, "Tablet interface for direct vision teleoperation of an excavator for urban construction work," *Automation in Construction*, vol. 102, pp. 17–26, 2019.

[14] J. Xu and H.-S. Yoon, "A review on mechanical and hydraulic system modeling of excavator manipulator system," *Journal of construction engineering*, vol. 2016, 2016.

[15] J. Nurmi and J. Mattila, "Automated feed-forward learning for pressure-compensated mobile hydraulic valves with significant dead-zone," in *Fluid Power Systems Technology*, vol. 58332. American Society of Mechanical Engineers, 2017, p. V001T01A027.

[16] D. Ortiz Morales, S. Westerberg, P. X. La Hera, U. Mettin, L. Freidovich, and A. S. Shiriaev, "Increasing the level of automation in the forestry logging process with crane trajectory planning and control," *Journal of Field Robotics*, vol. 31, no. 3, pp. 343–363, 2014.

[17] J. Mattila, J. Koivumäki, D. G. Caldwell, and C. Semini, "A survey on control of hydraulic robotic manipulators with projection to future trends," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 2, pp. 669–680, 2017.

[18] P. Egli and M. Hutter, "A general approach for the automation of hydraulic excavator arms using reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5679–5686, 2022.

[19] A. Taheri, P. Gustafsson, M. Rösth, R. Ghabcheloo, and J. Pajarinen, "Nonlinear model learning for compensation and feedforward control of real-world hydraulic actuators using gaussian processes," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9525–9532, 2022.

[20] M. Lee, H. Choi, C. Kim, J. Moon, D. Kim, and D. Lee, "Precision motion control of robotized industrial hydraulic excavators via data-driven model inversion," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1912–1919, 2022.

[21] E. Halbach, J. Kämäräinen, and R. Ghabcheloo, "Neural network pile loading controller trained by demonstration," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 980–986.

[22] D. Coleman, I. Sucan, S. Chitta, and N. Correll, "Reducing the barrier to entry of complex robotic software: a moveit! case study," *arXiv preprint arXiv:1404.3785*, 2014.