

## Article

# Kinematics Analysis and Trajectory Planning of 6-DOF Hydraulic Robotic Arm in Driving Side Pile

Mingjie Feng <sup>1,2</sup>, Jianbo Dai <sup>1,2,\*</sup>, Wenbo Zhou <sup>3</sup>, Haozhi Xu <sup>1,2</sup> and Zhongbin Wang <sup>1,2</sup>

<sup>1</sup> College of Mechanical and Electrical Engineering, China University of Mining and Technology, Xuzhou 221116, China; fengmj629@outlook.com (M.F.); ts22050187p31@cumt.edu.cn (H.X.); wangzbpaper@126.com (Z.W.)

<sup>2</sup> Jiangsu Collaborative Innovation Center of Intelligent Mining Equipment, China University of Mining and Technology, Xuzhou 221008, China

<sup>3</sup> System Engineering Institute of Sichuan Aerospace, Chengdu 610100, China; zbw84809137@163.com

\* Correspondence: jianbodai@cumt.edu.cn; Tel.: +86-25-84892901

**Abstract:** Given the difficulty in manually adjusting the position and posture of the pile body during the pile driving process, the improved Denavit-Hartenberg (D-H) parameter method is used to establish the kinematics equation of the mechanical arm, based on the motion characteristics of each mechanism of the mechanical arm of the pile driver, and forward and inverse kinematics analysis is carried out to solve the equation. The mechanical arm of the pile driver is modeled and simulated using the Robotics Toolbox of MATLAB to verify the proposed kinematics model of the mechanical arm of the pile driver. The Monte Carlo method is used to investigate the working space of the mechanical arm of the pile driver, revealing that the arm can extend from the nearest point by 900 mm to the furthest extension of 1800 mm. The actuator's lowest point allows for a descent of 1000 mm and an ascent of up to 1500 mm. A novel multi-strategy grey wolf optimizer (GWO) algorithm is proposed for robotic arm three-dimensional (3D) path planning, successfully outperforming the basic GWO, ant colony algorithm (ACA), genetic algorithm (GA), and artificial fish swarm algorithm (AFSA) in simulation experiments. Comparative results show that the proposed algorithm efficiently searches for optimal paths, avoiding obstacles with shorter lengths. In robotic arm simulations, the multi-strategy GWO reduces path length by 16.575% and running time by 9.452% compared to the basic GWO algorithm.

**Keywords:** kinematics analysis; robotic arm; side clip pile driving; workspace; trajectory planning



**Citation:** Feng, M.; Dai, J.; Zhou, W.; Xu, H.; Wang, Z. Kinematics Analysis and Trajectory Planning of 6-DOF Hydraulic Robotic Arm in Driving Side Pile. *Machines* **2024**, *12*, 191. <https://doi.org/10.3390/machines12030191>

Academic Editor: Hermes Giberti

Received: 12 January 2024

Revised: 16 February 2024

Accepted: 24 February 2024

Published: 15 March 2024



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

## 1. Introduction

In construction, pile driving is the preferred foundation method for sturdy structures, offering cost-effective and environmentally friendly alternatives to impact driving, especially in sensitive areas [1,2]. However, challenges such as blurred marking positions and manual inspection during piling lead to low efficiency and safety risks. Addressing these issues is crucial for enhancing efficiency, accuracy, and safety in heavy-duty construction projects. Currently, manual operation is relied upon for vibration pile drivers in pile foundation construction, requiring repeated adjustments and resulting in both high personnel skill demands and low construction efficiency [3]. The size of large equipment, the distance between pile holes, and blind spots in the cab lead to communication issues between operators and commanders. In addition, the use of a total station to measure the verticality of piles lacks real-time monitoring, so correcting any deviation is time-consuming and labor-intensive, seriously affecting construction efficiency. Numerous researchers have effectively addressed the accuracy challenge in pile driving by integrating navigation systems into pile drivers [4–7]. These systems enable the precise positioning and real-time guidance of piles. However, the constraints of manual control in complex real-time processes remain a concern. Simultaneously, the widespread use of robotic arms

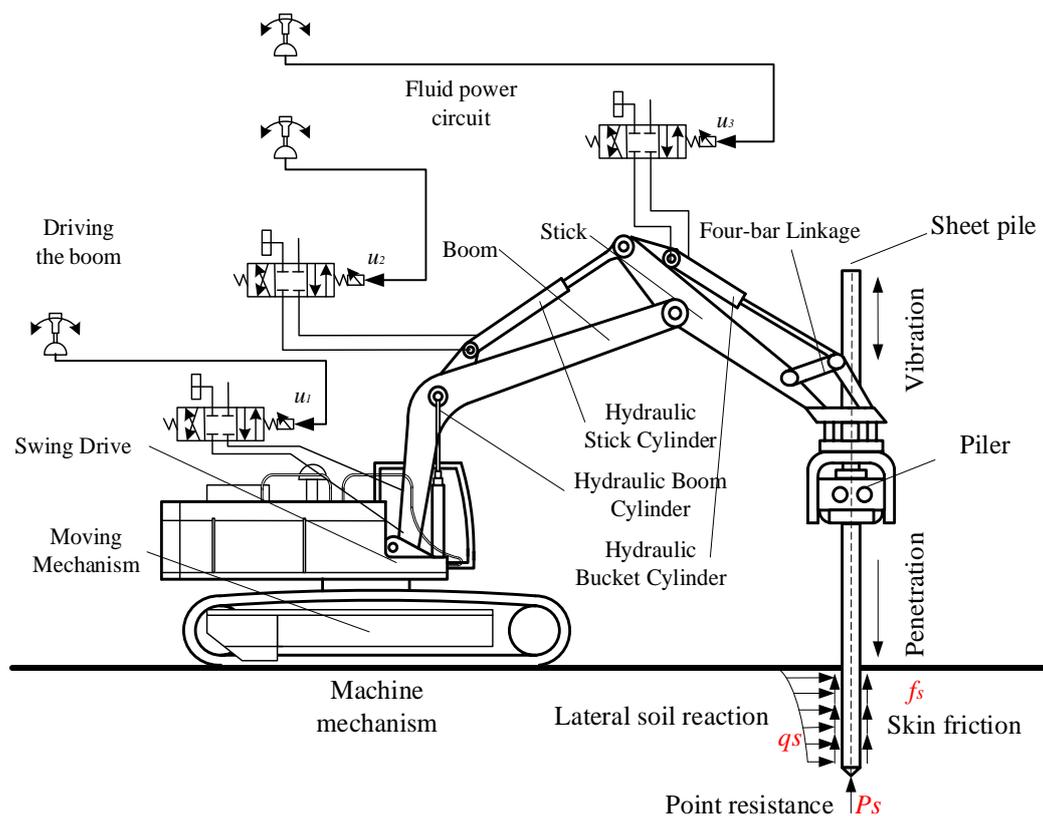
in geotechnical engineering [8–11] makes motion control of the vibration pile driver mechanism a viable solution for real-time monitoring and adaptive control of pile position and direction. Guan et al. [12] developed forward kinematics and simulation models for robotic excavators and established constraint models based on excavation operations to prevent contact with pile bodies, providing important insights for the design and control of robotic excavators in pile construction. Typically, the D-H parameter method is employed for mathematical modeling, especially for series robot mechanisms adhering to the Pieper criterion. This method systematically represents the geometric shape and joint relationships of robots. In inverse kinematics, the commonly used matrix inverse transformation method [13–16] determines joint angles, ensuring accurate positioning and direction control of the pile. These sophisticated kinematic modeling and solving techniques prove instrumental in enhancing the accuracy and efficiency of pile adjustments during pile driver operations. For robotic arms with low degrees of freedom, solving inverse kinematics problems typically relies on geometric methods [17–21]. It is noteworthy that there is limited research on the motion challenges of the mechanical arm in side-clamp pile drivers. Therefore, this article delves into the kinematic issues associated with robotic arms in the context of pile driver applications.

In addition, the performance of trajectory planning algorithms significantly improves the motion efficiency of robotic arms [22]. In the last few decades, a plethora of techniques have emerged for autonomous robot path planning. These include artificial potential fields [23,24], sampling-based algorithms [25], graph-based models [26–28], as well as swarm intelligence (SI) algorithms [29–32]. SI algorithms aim to replicate the adaptability, learning, and planning capabilities observed in organisms within their environments. In contrast to traditional path planning algorithms, these approaches demonstrate robust adaptability and effective search capabilities, particularly in complex and dynamic environments. Among these SI algorithms, the GWO algorithm has good global planning ability and great potential in path planning. Although applying the basic GWO algorithm for path planning can improve the path planning ability of robotic arms to a certain extent, there are still problems such as low operating accuracy, slow convergence speed, and easy falling into local optima. Much of the literature has also proposed corresponding improvement methods for the above issues [33–36]. Although these methods can play a certain role in addressing specific problems, there are still problems such as poor robustness, poor universality, and susceptibility to local optima when applied to the path planning of robotic arms.

This study addresses the challenges posed by manual operation and the difficulty in adjusting the pile position during the pile driving process of the side-clamp pile driver. Based on the motion characteristics of each mechanism, kinematic analysis was conducted on the mechanical arm of the pile driver, and an improved GWO path planning algorithm is proposed to improve the efficiency of the mechanical arm. In Section 2, the composition and characteristics of each mechanism of the side-clamp vibration pile driver are introduced. In Section 3, the enhanced D-H parameter method was utilized to analyze the kinematics of the mechanical arm of the hydraulic vibration pile driver. An inverse kinematics solution, employing matrix inverse transformation, was then employed to establish the correlation between the joint angle and hydraulic cylinder stroke, including Monte Carlo random sampling for a reachable workspace calculation, providing a theoretical foundation for accurate positioning of piles. In Section 4, the working principle of the GWO algorithm is described. A new multi-strategy GWO algorithm for three-dimensional (3D) path planning is proposed, which outperforms the basic GWO, ant colony algorithm (ACA), genetic algorithm (GA), and artificial fish swarm algorithm in simulation experiments. In the Section 5, the study summarizes its contributions, highlighting the achievement of precise execution of the planned trajectory of the side-clamp pile driver and automatic dynamic control of pile movement.

## 2. Side-Grip Hydraulic Vibration Pile Driver Mechanism

Currently, hydraulic pile drivers have emerged as the predominant choice in the pile driver market, owing to their robust power and efficient construction capabilities. The vibration pile driver excels in dual functionalities, seamlessly handling both pile sinking and pulling operations. Its construction principle revolves around harnessing the high-frequency vibration generated by the exciter to propel the adjacent soil into forced vibrations through the pile body. This induces alterations in the soil structure, subsequently diminishing the friction between the pile body and the surrounding soil. Consequently, the resistance encountered during both pile sinking and pulling operations is significantly minimized [37]. The work process consists of phases, where the excavator–piler combination unloads the piles from the truck and transports them to the workplace to drive them into the ground. After adjusting the drive line to the vertical, the vibratory unit works to compensate for the combined effects of point and skin friction of the soil, while the large translatory motion of the pile is fed by the excavator boom [3], as shown in Figure 1. The figure illustrates the schematic diagram of the side-grip hydraulic vibration pile driver mechanism. The operational components predominantly include the moving mechanism, swing drive, boom, stick, four-bar linkage, and swinging mechanism. These components are characterized by rotational joints, collectively constituting a six-degree-of-freedom (6-DOF) serial mechanism.



**Figure 1.** Working conditions of the vibratory pile driver.

### 2.1. Swing Drive

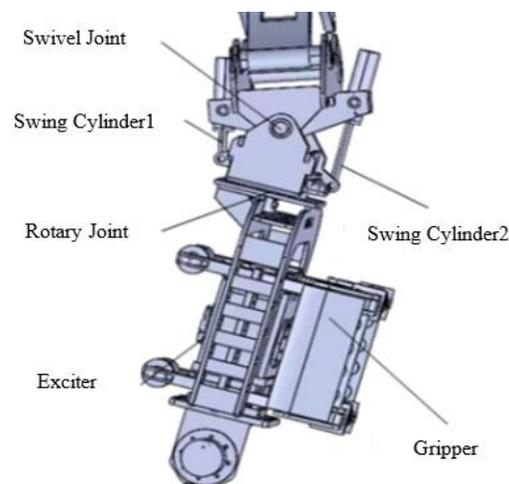
The swing drive comprises a slewing motor, a set of interlocking large and small gears, and a slewing support platform. The complete robotic arm system is securely mounted on the swinging support platform. In operation, a gear pump channels high-pressure oil to propel the slewing motor, facilitating the meshing transmission of the large and small gears. This synchronized movement achieves the rotation of the swing support platform. Throughout the operational sequence, the mechanical arm system undergoes controlled rotation by manipulating the swing drive.

## 2.2. Boom, Stick and the Four-Bar Linkage

The boom, affixed to the slewing support platform, undergoes elevation and descent through the action of the boom cylinder, thereby influencing the entire robotic arm's vertical movement. Extending forward and backward is accomplished by the bucket rod, securely attached at the boom's end, and propelled by the bucket rod cylinder. Additionally, the four-bar linkage mechanism, connected to the extremity of the bucket rod, experiences movement through the four-bar linkage cylinder, facilitating the motion of the end deflection mechanism. The collaborative operation of the boom, bucket rod, four-bar linkage mechanism, and slewing device orchestrates the dynamic repositioning of the pile body within the defined working range.

## 2.3. Swinging Mechanism and Gripper

The deflection mechanism of the pile driver is illustrated in Figure 2, featuring the swinging mechanism comprised of essential components such as the swing cylinder, exciter, and gripper. The gripper, functioning as an end effector and a sub-mechanism of the swinging mechanism, is responsible for securely clamping objects. This swinging mechanism is intricately linked to the four-bar linkage, possessing two degrees of freedom.



**Figure 2.** Swinging mechanism.

The first degree of freedom involves a set of swing cylinders strategically positioned on the left and right sides. By manipulating these cylinders, precise control is exerted, allowing the clamped pile body to rotate left and right around the swinging joint. The second degree of freedom is realized through a rotary joint situated beneath the swinging joint. This rotary joint facilitates the rotation of the clamped pile body around its axis, contributing to the mechanism's overall flexibility and adaptability during operation.

## 3. Kinematic Description of Working Mechanism System

### 3.1. Establishment of the Coordinate System

The side-clamp pile driver is a sequential robotic arm with six joints, involving rotating or moving elements and connecting rods. These connecting rods are linked to the base of the robotic arm. The structure of the robotic arm follows the D-H coordinate system, with an improved version based on its configuration [38]. The base coordinate system denoted as  $X_0Y_0Z_0$ , aligns with the coordinate system  $X_1Y_1Z_1$  at the turning device. Sequentially, coordinate systems are established at each joint. The coordinate system  $X_6Y_6Z_6$  is defined as the gripper coordinate system at the end-clamping mechanism of the pile driver. In the D-H coordinate system, each connecting rod comprises four independent parameters. These parameters are defined as follows: link  $a_{i-1}$  = distance from  $z_{i-1}$  to  $z_i$  along  $x_{i-1}$ , rotation  $\alpha_{i-1}$  = rotating from  $z_i$  to  $z_{i-1}$  around  $x_{i-1}$ , distance  $d_i$  = distance from  $x_{i-1}$  to  $x_i$  along  $z_i$ , joint rotation  $\theta_i$  = rotating from  $x_{i-1}$  to  $x_i$  around  $z_i$ . The D-H parameters of the

robotic arm can be determined based on the coordinate system established in Figure 3 and the dimensions of the robotic arm link, as listed in Table 1.

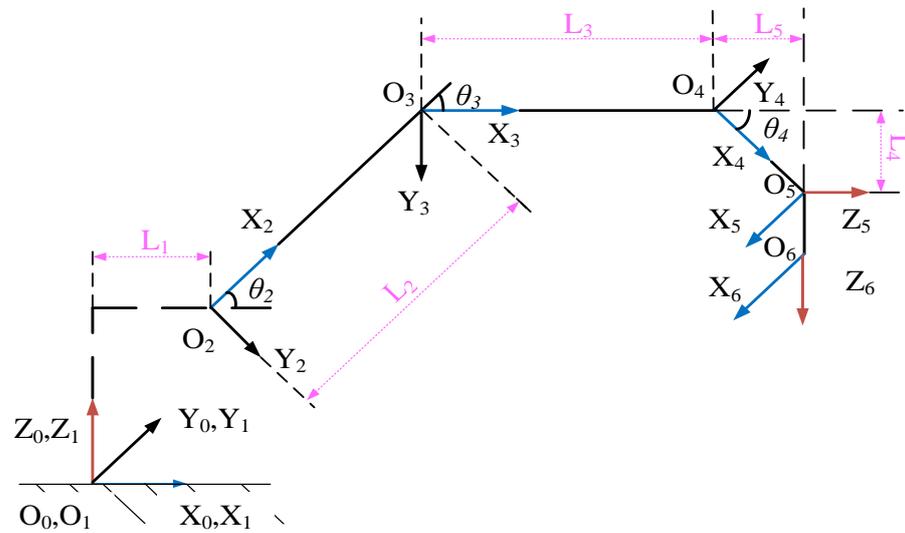


Figure 3. Coordinate system of each joint of the robotic arm.

Table 1. Parameters of the side-clamp pile driver D-H.

Link	$a_{i-1}$	$\alpha_{i-1}$ (rad)	$d_i$	$\theta_i$	Range $\theta_i$ (rad)
1	0	0	0	$\theta_1$	$(-\pi, \pi)$
2	$L_1$	$\pi/2$	0	$\theta_2$	$(0, \pi/3)$
3	$L_2$	0	0	$\theta_3$	$(-\pi/3, 0)$
4	$L_3$	0	0	$\theta_4$	$(-\pi/3, 0)$
5	$L_4$	$-\pi/2$	$L_5$	$\theta_5$	$(-\pi/6, \pi/6)$
6	0	$-\pi/2$	0	$\theta_6$	$(-\pi, \pi)$

### 3.2. Forward Kinematics Analysis

Forward kinematics is to solve the pose of the end effector of the robot arm relative to the robot base coordinate system by knowing the joint angle of each link and the structural size of the link. According to the homogeneous coordinate transformation method, the general expression of the transformation matrix between two adjacent joint coordinate systems is as follows:

$$\begin{aligned}
 {}^i T_i &= Rot(x, \alpha_{i-1}) Trans(x, a_{i-1}) Trans(z, d_i) Rot(z, \theta_i) \\
 &= \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & a_{i-1} \\ \sin(\theta_i) \cos(\alpha_{i-1}) & \cos(\theta_i) \cos(\alpha_{i-1}) & -\sin(\alpha_{i-1}) & -d_i \sin(\alpha_{i-1}) \\ \sin(\theta_i) \sin(\alpha_{i-1}) & \cos(\theta_i) \sin(\alpha_{i-1}) & \cos(\alpha_{i-1}) & d_i \cos(\alpha_{i-1}) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)
 \end{aligned}$$

Putting the connecting rod parameters in Table 1 into Formula (1), and multiplying the transformation matrices in sequence, the transformation matrix of the robot arm end coordinate system {6} under the base coordinate system {0} can be obtained as follows:

$${}^0 T_6 = {}^0 T_1 \cdot {}^1 T_2 \cdot {}^2 T_3 \cdot {}^3 T_4 \cdot {}^4 T_5 \cdot {}^5 T_6 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

The values of each element of Formula (2) are obtained as follows:

$$\begin{aligned}
 n_x &= c_6(s_1c_5 + s_5c_1s_{234}) - s_6c_1c_{234} \\
 n_y &= -s_6s_1c_{234} - c_6(c_1c_5 - s_5s_1s_{234}) \\
 n_z &= -s_6s_{234} - c_6s_5c_{234} \\
 o_x &= -s_6(s_5s_1 + s_5c_1s_{234}) - c_6c_1c_{234} \\
 o_y &= s_6(c_1c_5 - s_5s_1s_{234}) - c_6s_1c_{234} \\
 o_z &= s_5s_6c_{234} - c_6s_{234}, a_x = c_5c_1s_{234} - s_1s_5 \\
 a_y &= c_5s_1s_{234} + c_1s_5, a_z = -c_5c_{234} \\
 p_x &= L_1c_1 + L_3c_1c_{23} + L_4c_1s_{234} + L_5c_1c_{234} + L_2c_1c_2 \\
 p_y &= L_1s_1 + L_3s_1c_{23} + L_2c_2s_1 + L_4s_1s_{234} + L_5s_1c_{234} \\
 p_z &= L_2s_2 + L_3s_{23} - L_4c_{234} + L_5s_{234}
 \end{aligned}$$

and  $c_i = \cos(\theta_i)$ ,  $s_i = \sin(\theta_i)$ .

### 3.3. Inverse Kinematics Model Analysis

To manipulate the orientation of the pile body, solving the inverse kinematics of the robotic arm becomes imperative. In simpler terms, this entails determining the angles of each joint in the robotic arm when provided with the desired orientation of the end effector coordinate system concerning the base coordinate system.

The mechanical arm characteristics of the side-clamp pile driver under investigation adhere to the Pieper criterion, implying the presence of an analytical solution. The proposed solution method involves employing the inverse matrix transformation method. The procedural steps are outlined below.

Given that the end effector pose is  ${}^0_6T$ , the transformation matrix of the end coordinate system in the base coordinate system can be obtained as follows:

$$\begin{aligned}
 {}^1_6T &= {}^0_1T^{-1} \cdot {}^0_6T = \begin{bmatrix} c_1 & s_1 & 0 & 0 \\ -s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} c_1n_x + n_ys_1 & c_1o_x + o_ys_1 & a_xc_1 + a_ys_1 & c_1p_x + p_ys_1 \\ c_1n_y - n_xs_1 & c_1o_y - o_xs_1 & a_yc_1 - a_xs_1 & c_1p_y - p_xs_1 \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)
 \end{aligned}$$

where

$${}^1_6T = {}^1_2T \cdot {}^2_3T \cdot {}^3_4T \cdot {}^4_5T \cdot {}^5_6T \quad (4)$$

Let the (2, 4) elements in the matrices of Formulas (3) and (4) be equal, and we can obtain the following:

$$\theta_1 = \arctan\left(\frac{p_y}{p_x}\right) \quad (5)$$

Let the elements on both sides of the matrices (1, 4) and (3, 4) of Formulas (3) and (4) be equal, and we can obtain the following:

$$\theta_3 = \arctan\left(\frac{s_3}{c_3}\right) \quad (6)$$

Among them, we can obtain the following:

$$\begin{aligned}
 m_1 &= (c_1p_x + s_1p_y - L_5c_{234} - L_1 - L_4s_{234})^2 \\
 m_2 &= (p_z + L_4c_{234} - L_5s_{234})^2 - L_2^2 - L_3^2 \\
 c_3 &= \frac{(m_1 + m_2)}{2} \times L_2 \times L_3 \\
 s_3 &= \pm \sqrt{1 - c_3^2}
 \end{aligned} \quad (7)$$

After sorting out Formulas (3) and (4), we can obtain the following:

$${}^4_6T = {}^3_4T^{-1} \cdot {}^2_3T^{-1} \cdot {}^1_2T^{-1} \cdot {}^0_1T^{-1} \cdot {}^0_6T = {}^4_5T \cdot {}^5_6T \quad (8)$$

Let the elements (2, 3) on both sides of the matrix of Formula (8) be equal, and we can obtain  $\theta_{234}$  as follows:

$$\theta_{234} = \arctan\left(\frac{s_{234}}{c_{234}}\right) \quad (9)$$

Here,

$$\begin{aligned} s_{234} &= (p_y^2 - p_x^2)a_x - 2a_y p_x p_y \\ c_{234} &= (p_y^2 + p_x^2)a_z \end{aligned} \quad (10)$$

Let the elements of matrices (1, 3) and (3, 3) of Formula (8) be equal and we can obtain the following:

$$\theta_5 = \arctan\left(\frac{s_5}{c_5}\right) \quad (11)$$

Here,

$$\begin{aligned} s_5 &= c_1(a_y c_1 - a_x s_1) - s_1(a_x c_1 + a_y s_1) \\ c_5 &= s_{234}((c_1^2 - s_1^2)a_x + 2s_1 c_1 a_y) - a_z c_{234} \end{aligned} \quad (12)$$

Let the elements on both sides of Formulas (3) and (4) matrices (1, 4) (3, 4) be equal to obtain the following:

$$\theta_2 = \arctan\left(\frac{s_2}{c_2}\right) \quad (13)$$

Here,

$$\begin{aligned} c_2 &= \frac{k_2 L_3 s_3 + k_1 (L_2 + L_3 c_3)}{(L_2 + L_3 c_3)^2 + (L_3 s_3)^2} \\ s_2 &= \frac{k_2 L_3 s_3 (L_2 + L_3 c_3) + k_1 (L_2 + L_3 c_3)^2}{((L_2 + L_3 c_3)^2 + (L_3 s_3)^2) L_3 s_3} - \frac{k_1}{L_3 s_3} \\ k_1 &= c_1 p_x + p_y s_1 - L_5 c_{234} - L_1 - L_4 s_{234} \\ k_2 &= p_z + L_4 c_{234} - L_5 s_{234} \end{aligned} \quad (14)$$

Let the elements (2, 1) and (2, 2) on both sides of Formula (8) be equal, and we can obtain the following:

$$\theta_6 = \arctan\left(\frac{s_6}{c_6}\right) \quad (15)$$

Here,

$$\begin{aligned} s_6 &= n_z s_{234} + c_{234}(n_x(c_1^2 - s_1^2) + 2c_1 s_1 n_y) \\ c_6 &= o_z s_{234} + c_{234}(o_x(c_1^2 - s_1^2) + 2c_1 s_1 o_y) \end{aligned} \quad (16)$$

According to the above formula, the values of  $\theta_3$ ,  $\theta_{234}$ , and  $\theta_2$  can be obtained as follows:

$$\theta_4 = \theta_{234} - \theta_2 - \theta_3 \quad (17)$$

### 3.4. Relationship between Pile Body Posture and Hydraulic Cylinder Stroke

The pile body posture adjustment is finally completed through the joint adjustment of the hydraulic boom cylinder, hydraulic stick cylinder, four-bar linkage drive cylinder, and swinging mechanism. To realize the adjustment of the clamped object according to the desired posture, the joint angle  $\theta_i$  must first be solved through the inverse kinematics of the manipulator, and then the mapping relationship between the joint rotation angle  $\theta_i$  and the joint cylinder stroke  $\lambda_i$  must be established. Finally, the pile position can be achieved by controlling the cylinder stroke and posture adjustment. In this article, the corresponding relationship between the joint angles of the pile driver's mechanical arm and its cylinder length is described below.

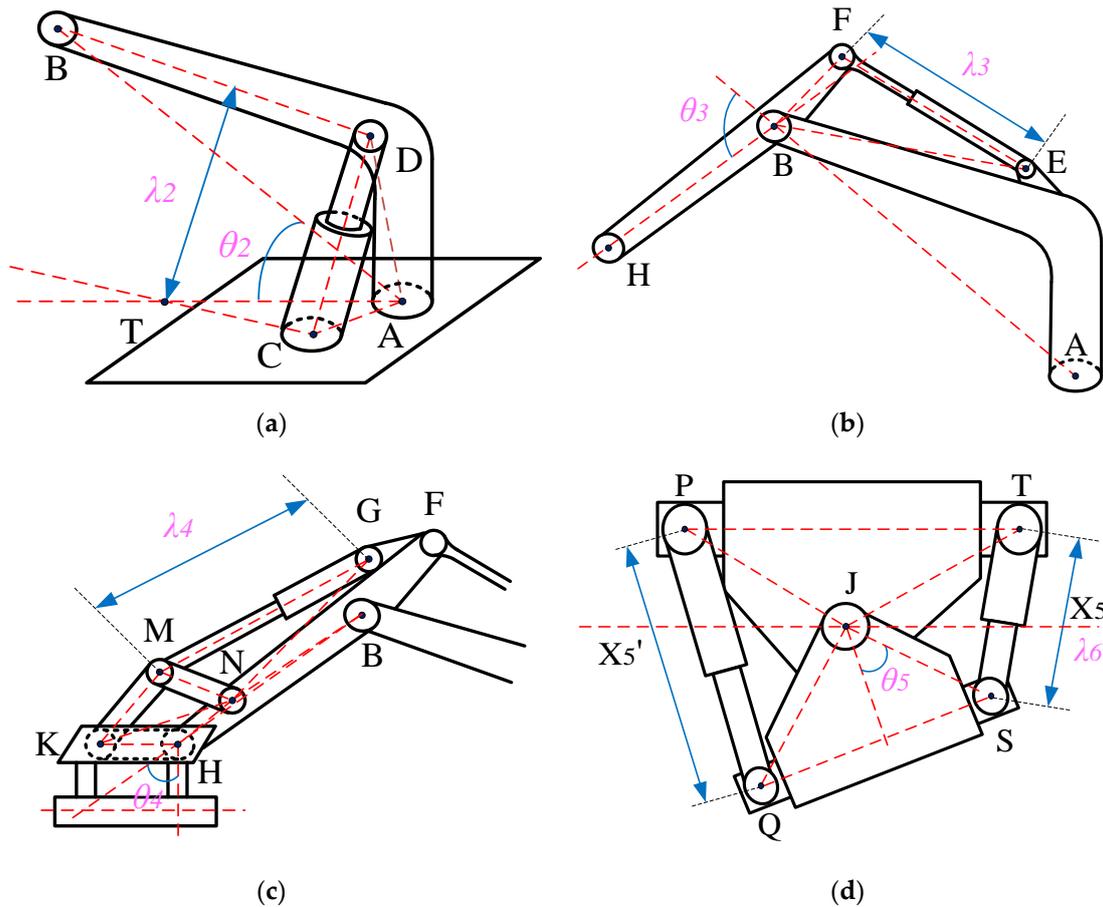
### 3.4.1. The Relationship between the Boom Joint Angle and Its Cylinder Length

The schematic diagram of the vibration pile driver arm mechanism is shown in Figure 4a. In the figure,  $L_{DC}$  is the telescopic length of the boom cylinder to be found. In the triangle CAD it is obtained as follows:

$$\lambda_2 = L_{DC} = \sqrt{L_{AC}^2 + L_{AD}^2 - 2 \cdot L_{AC} \cdot L_{AD} \cdot \cos(\angle CAD)} \quad (18)$$

Here,

$$\angle CAD = \angle CAT + \theta_2 + \angle BAD$$



**Figure 4.** Relationship between pile body posture and hydraulic cylinder stroke. (a) Schematic diagram of the boom mechanism; (b) schematic diagram of the stick mechanism; (c) schematic diagram of the four-bar linkage mechanism; (d) schematic diagram of the deflection mechanism.

### 3.4.2. The Relationship between Stick Joint Angle and Cylinder Length

The schematic diagram of the arm mechanism of the vibration pile driver is shown in Figure 4b. In the figure,  $L_{EF}$  is the length of the arm cylinder to be found. According to the triangle EBF, it can be obtained as follows:

$$\lambda_3 = L_{EF} = \sqrt{L_{BE}^2 + L_{BF}^2 - 2 \cdot L_{BE} \cdot L_{BF} \cdot \cos(\angle EBF)} \quad (19)$$

Here,

$$\angle EBF = 180^\circ + \theta_3 - \angle FBH - \angle ABE$$

### 3.4.3. The Relationship between the Four-Link Joint Angle and Its Cylinder Length

The schematic diagram of the four-bar linkage mechanism of the pile driver is shown in Figure 4c.

As can be seen from Figure 4c, in the triangle MNG,

$$\lambda_4 = L_{MG} = \sqrt{L_{MN}^2 + L_{NG}^2 - 2 \cdot L_{MN} \cdot L_{NG} \cdot \cos(\angle MNG)} \quad (20)$$

Taking joint point H as the center of the circle, we can obtain the following:

$$L_{KN} = \sqrt{L_{KH}^2 + L_{HN}^2 - 2L_{KH} \cdot L_{HN} \cdot \cos(\angle KHN)} \quad (21)$$

Taking joint point N as the center of the circle, we can obtain the following:

$$\angle KNH = \arccos\left(\frac{L_{HN}^2 + L_{KN}^2 - L_{KH}^2}{2 \cdot L_{HN} \cdot L_{KN}}\right) \quad (22)$$

$$\angle KNM = \arccos\left(\frac{L_{MN}^2 + L_{KN}^2 - L_{KM}^2}{2 \cdot L_{MN} \cdot L_{KN}}\right) \quad (23)$$

Therefore, we can obtain the following:

$$\angle MNG = 360^\circ - \angle GNB - \angle BNH - \angle HNK - \angle KNM$$

### 3.4.4. The Relationship between the Swinging Joint Angle and the Length of the Swinging Cylinder

The joint angle of the swinging mechanism and the geometric structure of the drive cylinder of the swinging mechanism are shown in Figure 4d. P, Q, T, and S are the installation positions of the swinging cylinder, and point J is the joint point of the swinging mechanism. The lengths of  $L_{PT}$ ,  $L_{PJ}$ ,  $L_{TJ}$ ,  $L_{QJ}$ ,  $L_{QS}$ , and  $L_{SJ}$  in the figure are all fixed values. According to triangles PJQ and SJT, we can know the following:

$$\lambda_5 = L_{PQ} = \sqrt{L_{PJ}^2 + L_{QJ}^2 - 2 \cdot L_{PJ} \cdot L_{QJ} \cdot \cos(\angle PJQ)} \quad (24)$$

$$\lambda_6 = L_{TS} = \sqrt{L_{TJ}^2 + L_{SJ}^2 - 2 \cdot L_{TJ} \cdot L_{SJ} \cdot \cos(\angle SJT)} \quad (25)$$

Here,

$$\angle PJQ = 180^\circ - \theta_5 - \frac{\angle QJS}{2} + \angle X'_5JP$$

$$\angle SJT = \theta_5 - \frac{\angle X_4JS}{2} + \angle X_5JP$$

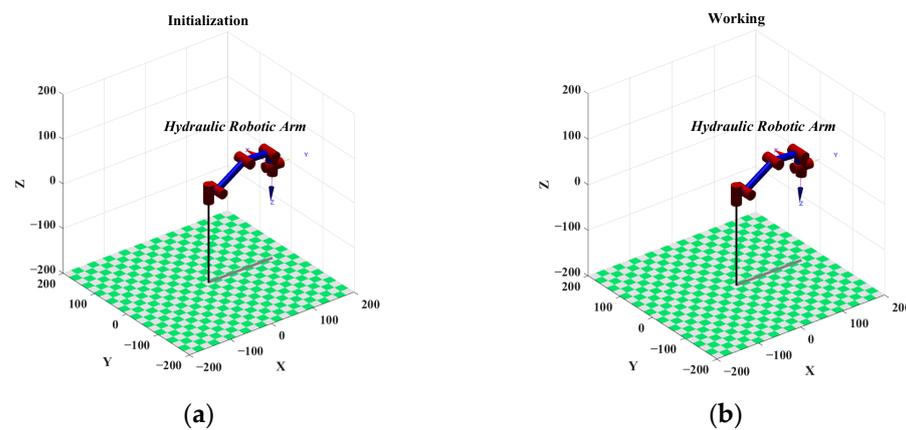
### 3.5. Forward Kinematics Simulation

To verify the correctness of the kinematics model of the pile driver's manipulator, MATLAB's robotics toolbox was used to perform kinematic modeling and simulation. First, the MATLAB environment was used to import the D-H parameters in Table 1 through the Link function, and the Serial Link function was used to build the robotic arm model. Then, given a set of initial joint angles  $q_1 = [0,0,0,0,0,0]$ , at this time, the pile driver mechanical arms are in a horizontal state relative to the zero point posture. If the pile driver is in working condition to keep the pile body sinking vertically, the boom, stick, and yaw joint angles will change. Within the working range,  $q_2 = [-60, 30, -20, -10, 0, 0]$ . The length of each connecting rod of the pile driver is  $L_1 = 20$  cm,  $L_2 = 80$  cm,  $L_3 = 60$  cm,  $L_4 = 30$  cm,

and  $L_5 = 10$  cm. Bringing the vectors  $q_1$  and  $q_2$  into the kinematics forward solution model solved above, the pose matrix of the end effector is obtained as follows:

$${}^0_6T^1 = \begin{bmatrix} 0 & -1 & 0 & 190 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -30 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^0_6T^2 = \begin{bmatrix} -0.866 & -0.5 & 0 & 89.19 \\ -0.5 & 0.866 & 0 & -154.5 \\ 0 & 0 & -1 & -20.42 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The result of the kinematics forward solution model is  ${}^0_6T^1, {}^0_6T^2$  matrices which are exactly the same as the matrices solved by the Fkine function in the MATLAB robotics toolbox, which verifies the correctness of the forward solution model. The simulation model of its robotic arm is shown in Figure 5.



**Figure 5.** Simulation model. (a) Initialization; (b) the working range of the pile driver.

### 3.6. Inverse Kinematics Simulation

Inverse kinematics simulation gives the pose matrix of the actuator coordinates at the end of the robot arm, uses the matrix as a known quantity to calculate the angles corresponding to each joint of the robot arm, and inputs this angle as the forward kinematics solution and brings it into the robot tool. If the solved matrix is consistent with the given pose matrix, the correctness of the inverse kinematic solution model is verified.

Randomly select a set of joint angles  $S_1 = [50, 30, -20, -10, 5, 20]$  within the working range of the pile driver, and use the Fkine function of the robot toolbox to solve the corresponding matrix:

$${}^0_6T^3 = \begin{bmatrix} 0.4973 & -0.865 & -0.0667 & 114.7 \\ -0.8637 & -0.5 & 0.056 & 136.6 \\ -0.0819 & 0.0298 & -0.99 & 40.42 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A 6-DOF manipulator may have eight sets of joint angle combinations in the same posture. However, in the actual operation of the pile driver manipulator, there will only be one set of solutions due to the stroke limitations of each joint mechanism and the drive cylinder. Bring the matrix  ${}^0_6T^3$  as the input value into the kinematics inverse solution model, and solve a set of solutions  $S_2 = [50.007, 30.002, -20.001, -10.0, 5.00, 20.001]$ . Bring the solved joint angle  $S_2$  into the kinematics forward solution model, and compare the solved matrix  ${}^0_6T^4$  with the original  ${}^0_6T^3$  matrix.

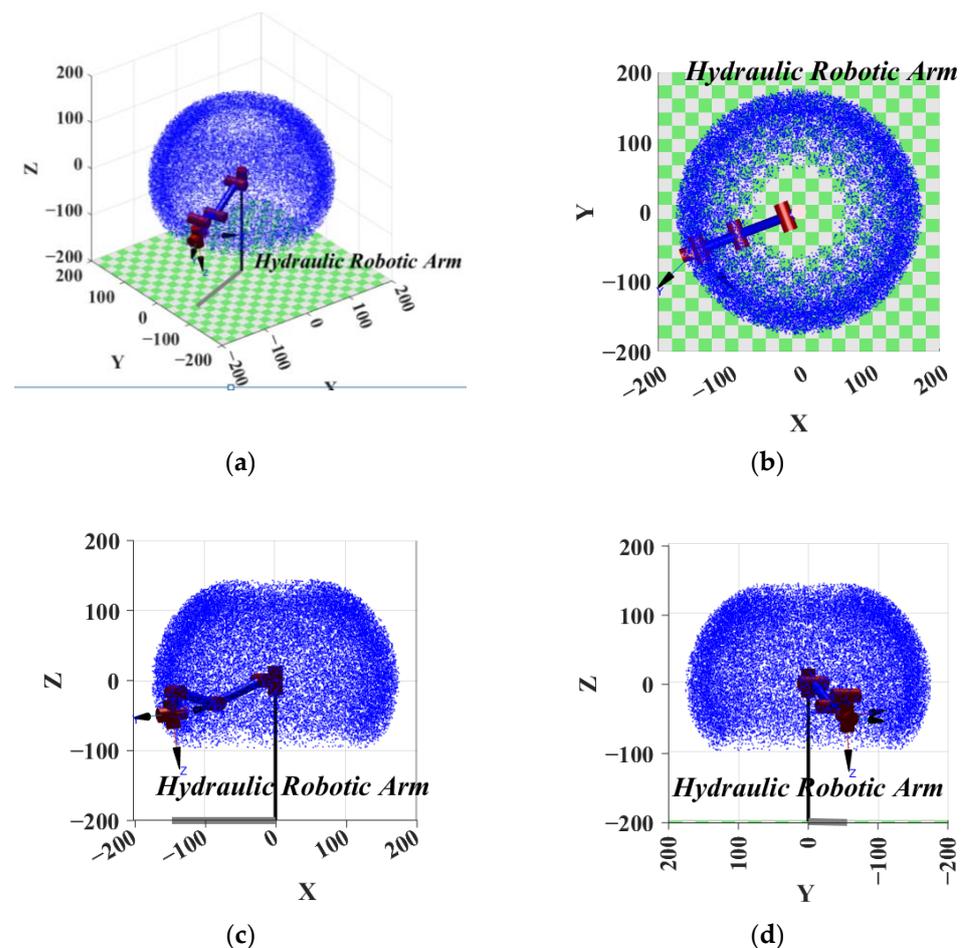
$${}^0_6T^4 = \begin{bmatrix} 0.4973 & -0.865 & -0.0667 & 114.7 \\ -0.8637 & -0.5 & 0.056 & 136.6 \\ -0.0819 & 0.0298 & -0.99 & 40.42 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The results are the same, which verifies the correctness of the inverse solution model.

### 3.7. Workspace Analysis

The working space of the pile driver's robotic arm is a collection of points that the end effector can reach in the space. The working space reflects the structural characteristics of the robotic arm. Analysis of its characteristics is of great significance for optimizing the structural research of the robotic arm and the performance of the robotic arm.

This article uses the Monte Carlo method [14–16] to conduct workspace simulation analysis. The values of the six joint variables are randomly selected within the value range of the six joint variables. The position of the end actuator constitutes a set of points that constitutes the robot arm workspace. In this paper,  $N = 30,000$  random values are selected for each joint variable, and a forward kinematics solution is performed on  $N$  groups of joint vectors to obtain the position set of the end actuator in space. The results are shown in Figure 6a. To clearly and intuitively observe the simulation represented, the graphics are projected on the XOY, XOZ, and YOZ planes, and the results are shown in Figure 6b–d. From the working space point cloud diagram in Figure 6a, it can be seen that the working space of the pile driver is approximately a sphere, with a large working range and a uniform distribution of points in the working space, indicating that the mechanical arm structure is reasonably arranged. According to Figure 6b, the range of  $X$  is between  $-1800 \sim -900$  mm and  $900 \sim 1800$  mm and the range of  $Y$  is between  $-1800 \sim -900$  mm and  $900 \sim 1800$  mm. As shown in Figure 6c,d, the range of  $Z$  is from  $-1000$  to  $1500$  mm.



**Figure 6.** Pile driver robotic arm working space. (a) Three-dimensional view of the work space; (b) projected on the XOY plane; (c) projected on the XOZ plane; (d) projected on the YOZ plane.

#### 4. Trajectory Planning

Piling machines often operate unmanned in hazardous environments, like geological disaster areas, repairing damaged infrastructure. Unmanned pile drivers are used for underwater bridge maintenance and repair, stabilizing piers. In offshore platform construction, they drive pile foundations to ensure stability, particularly in deep-sea areas, reducing risks to personnel from underwater environments and high-pressure gases. The system must execute tasks remotely or through preset programs. In dynamic pile driving operations, studying obstacle avoidance motion planning strategies is crucial. Traditional robotic arm methods rely heavily on manual programming, resulting in inefficiency and adaptability issues for obstacle avoidance. To address this, robotic arms need autonomous obstacle-free trajectory planning, involving processing information from the starting point to the target, exploring collision-free joint angle configurations, and completing pile driving tasks. Advanced path-planning algorithms are essential for efficient, safe, and compliant robotic arm operations.

##### 4.1. Gray Wolf Algorithm

The SI algorithm often finds inspiration from the collective behavior observed in various animal species [39], which typically simulates the hunting process of prey, searching for food based on local interactions between group members or based on the environment (granular) [40]. Therefore, grouped groups are self-organizing. It is typically initiated by a set of candidate solutions. Usually, groups can be divided into two main groups, leaders and followers, where interaction occurs by attracting followers to the leader. Therefore, convergence towards the optimal solution can take place. GWO is the fastest growing SI algorithm introduced by Mirjalili et al. [41] to mimic the hunting behavior of gray packets in nature. GWO is a powerful optimizer because it has impressive features compared to other models, such as easy adaptation, no parameters, no derivatives, no memory, no computation, flexibility, and complete sound. In the initial search phase, GWO starts with high intensity in the exploration phase, and in the final run, GWO gradually shifts its focus to the development phase through the positions of the three best leaders. The social division of labor among wolves includes alpha wolf, beta wolf, delta wolf, and omega wolf.

The hunting model of wolf packs is showed in Figure 7. During the iteration process, the objective function value of the optimal wolf after each iteration is compared with the value of the alpha wolf in the previous generation. If it is better, the position of the alpha wolf is updated. If there are multiple horses at this time, randomly select one to become the alpha wolf. The alpha wolf does not perform three intelligent behaviors and directly enters the next iteration until it is replaced by other stronger omega wolves. The behaviors are described below.

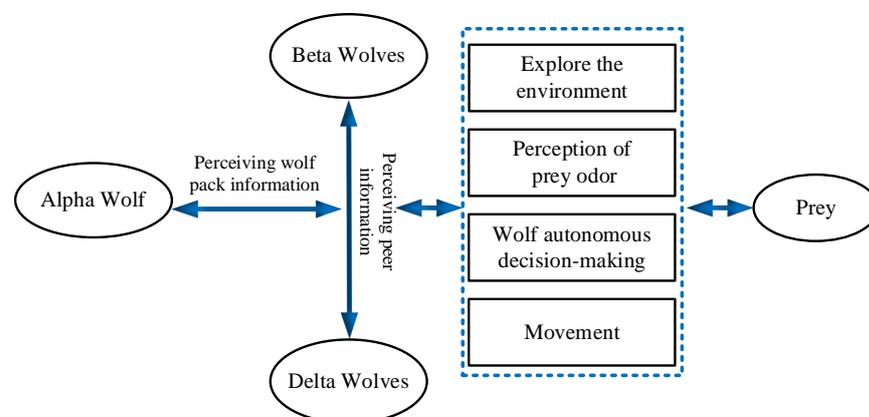


Figure 7. The hunting model of wolf packs.

- Wandering behavior: In the solution space, the best  $S$  omega wolves other than the alpha wolf are considered beta wolves, and the fitness of the prey searched by each beta

wolf is calculated separately. If it is greater than the fitness of the alpha wolf, the beta wolf becomes the alpha wolf and initiates a new summoning behavior. Otherwise, the beta wolf will choose the direction with the strongest odor and the odor concentration  $Y_i$  greater than the current position in the direction  $p$  and move forward one step as follows:

$$x_{id}^p = x_{id} + \sin(2\pi \times \frac{p}{h}) \times step_s^d \quad (26)$$

where  $step_s^d$  is the walking step length,  $h$  is the walking direction,  $p = 1, 2, \dots$ . Repeat the above wandering behavior until there is a scent concentration  $Y_i > Y_{lead}$  of a certain beta wolf, then beta wolf  $i$  replaces the alpha wolf and initiates a summoning behavior, or reaches the maximum number of wandering times  $T_{max}$ .

- **Summoning behavior:** After the wandering behavior ends, an alpha wolf will be generated. The alpha wolf uses a howling method to initiate a summoning behavior, quickly summoning the surrounding  $M$  delta wolves towards their position. The delta wolves quickly approach the alpha wolf with a running stride and search for prey. During the delta wolf attack, if the prey has a higher adaptability, let the delta wolf replace the alpha wolf. When the distance between the delta wolf and the alpha wolf is less than the threshold, it switches to besieging behavior. When the delta wolf  $j$  undergoes the  $k + 1$  iteration, its position in the  $d$ -dimensional space can be expressed as follows:

$$x_{jd}^{k+1} = x_{jd}^k + step_b^d \times \frac{g_d^k - x_{jd}^k}{|g_d^k - x_{jd}^k|} \quad (27)$$

where  $g_d^k$  is the position of the alpha wolf of the  $k$  generation group in the  $d$ -dimensional space,  $x_{jd}^k$  is the current position of the delta wolf  $j$ , and  $step_b^d \times \frac{g_d^k - x_{jd}^k}{|g_d^k - x_{jd}^k|}$  is the gradual gathering of the delta wolf  $j$  towards the position of the alpha wolf. On the way, if the odor concentration of the delta wolf is  $Y > Y_{lead}$ , then the delta wolf  $i$  replaces the alpha wolf and initiates a summoning behavior. If  $Y < Y_{lead}$ , the delta wolf continues to attack until  $d_{is} \leq d_{near}$ , joining the attack on the prey, that is, entering a siege behavior as follows:

$$d_{near} = \frac{1}{D\omega} \sum_{d=1}^D |\max_d - \min_d| \quad (28)$$

where  $\omega$  is for the distance determination factor, and the range of values for the  $d$ -dimensional variable to be optimized is  $[\min_d, \max_d]$ .

- **Siege behavior:** A combination of delta wolves and beta wolves to encircle and capture prey. The delta wolf sensed the call of the alpha wolf and immediately ran towards the position of the alpha wolf. During the running process, if it found that the prey had higher adaptability, it immediately replaced the original alpha wolf and commanded other wolves to take action.

$$x_{id}^{k+1} = x_{id}^k + \lambda \times step_w^d \times |G_d^k - x_{id}^k| \quad (29)$$

here  $\lambda$  is a random number within the interval of  $[-1,1]$ , uniformly distributed,  $step_w^d$  is the attack step size of omega wolf  $i$  (including beta and delta wolves) when engaging in besieging behavior in the  $d$ -dimension space. After adopting a siege behavior, if the odor concentration perceived by omega wolf  $i$  is greater than the concentration perceived at its original position, the position of omega wolf  $i$  will be updated; otherwise, the position of

omega wolf  $i$  will remain unchanged. The relationship between the walking stride  $step_s^d$ , attacking stride  $step_b^d$ , and siege stride  $step_w^d$  of omega wolves is as follows:

$$step_s^d = \frac{step_b^d}{2} = 2 \times step_w^d \times \frac{|M_d - m_d|}{C} \tag{30}$$

The path planning process diagram based on GWO is shown in Figure 8.

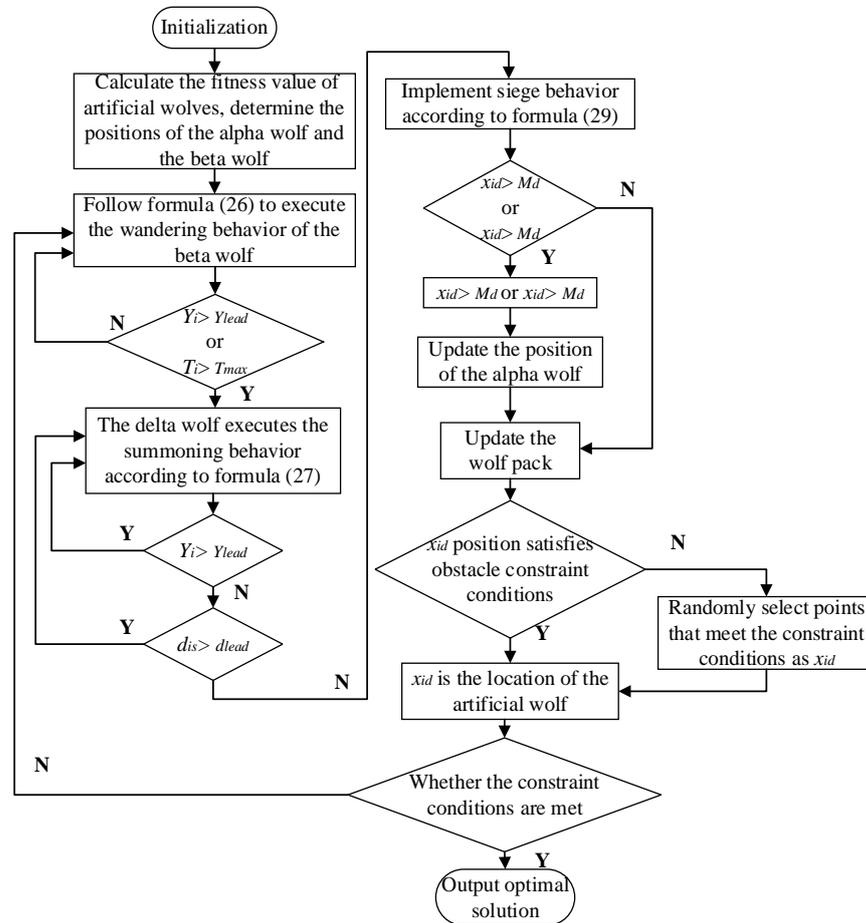


Figure 8. GWO flowchart.

#### 4.2. Improved GWO

##### 4.2.1. Adaptive Wandering and Sieging Strategy

In the basic GWO algorithm, the update step size of the wandering and sieging behavior positions of each generation of wolf packs is fixed and invariant, and both are related to the walking step size. This fixed step size approach can cause a slow convergence speed in the initial stage of the algorithm and a decrease in search precision in the later stage, leading to the algorithm falling into the local optima. Therefore, this article adopts an adaptive step size approach to optimize the walking step size and running step size in the basic Gray Wolf Algorithm, and improves the search accuracy and convergence speed of the algorithm by reasonably changing the adaptive step size.

The article adopts an adaptive step size of

$$step = rand \times \|x_i - X_{lead}\|_2$$

where  $rand$  represents a random number between  $[0, 1]$ ;  $x_i$  represents the current position of the wolf  $i$ ;  $X_{lead}$  represents the position of the alpha wolf.

#### 4.2.2. Adaptive Walk Behavior Based on Levy Flight Strategy

Levy flight is a random search method that follows the Levy distribution and has good search randomness and global optimization ability. Using the Levy flight search strategy for a global random search can make the distribution of wolf scouts more extensive in the search space, improve the algorithm's global search ability, and easily jump out of local optimal solutions.

The formula for updating the position of Levy flight is as follows:

$$x_i^{t+1} = x_i^t + l \oplus Levy(\lambda)$$

where  $l$  represents step weight, and  $l = 0.01 \times (x_i^t - x_{ibest})$ ,  $\oplus$  represents point-to-point multiplication;  $Levy$  represents a random path that follows the Levy distribution, satisfying the following conditions:

$$Levy \sim u = t^{-\lambda}, 1 < \lambda \leq 3$$

The formula for updating the position of the adaptive walking behavior of the beta wolf with the addition of Levy flight strategy is as follows:

$$x_{id}^t = x_{id} + \sin\left(2\pi \times \frac{p}{h}\right) \times rand \times \|x_i - X_{lead}\|_2 + l \oplus Levy(\lambda)$$

where  $x_{id}^t$  is the position of the beta wolf  $i$  in the  $d$ -dimensional space in the  $t$  iteration, and  $X_{lead}$  is the position of the alpha wolf (i.e., prey).

Based on the above formula for updating the position of the wandering behavior, the algorithm has the excellent global search ability of Levy flight, while dynamically adjusting the forward step size of the beta wolf to compensate for the slow convergence speed of Levy flight strategy.

#### 4.2.3. Adaptive Summoning Behavior

During the summoning behavior, delta wolves rush toward the position of their prey (i.e., the position of the alpha wolf). If the target fitness value  $Y_j$  perceived by the delta wolf during the attack is smaller than the target fitness value  $Y_{lead}$  perceived by the alpha wolf, then  $Y_{lead} = Y_j$ , and the delta wolf replaces the alpha wolf to initiate the summoning behavior again. If the target fitness value  $Y_j$  perceived by the delta wolf is greater than the target function value  $Y_{lead}$  perceived by the alpha wolf, the delta wolf will continue to perform the rushing behavior until the distance between the delta wolf and the alpha wolf is less than  $d_{near}$ .

The formula for updating the position of delta wolf  $j$  in the  $t + 1$  iteration is as follows:

$$x_{jd}^{t+1} = x_{jd}^t + rand \times \left\| x_{jd}^t - x_{lead}^t \right\|_2 \times \frac{x_{lead}^t - x_{jd}^t}{\left| x_{lead}^t - x_{jd}^t \right|}$$

where  $x_{jd}^t$  is the position of the delta wolf  $j$  in the  $d$ -dimensional space in the  $t$  iteration.

### 4.3. Simulation Environment and Experimental Data Preparation

In order to verify the effectiveness of the multi-strategy improved GWO algorithm in 3D path planning of robotic arms, the basic GWO algorithm and the proposed multi-strategy improved GWO algorithm were used for 3D path planning simulation under the same conditions. The simulation experimental environment is a computer with a Win10 64-bit operating system, featuring an AMD Ryzen 7 3750H 2.30 GHz processor, and 8 GB memory configuration, manufactured by AMD and assembled by Asus in China. The simulation experiment software used is MATLAB R2021b. At the same time, in order to further demonstrate the effectiveness and superiority of the improved GWO algorithm, this paper added comparative experiments of the improved GWO algorithm,

basic GWO algorithm, basic ACA, basic GA, and basic AFSA in the 3D environment simulation experiment.

The task space for simulating the environment and preparing experimental data for path planning is  $100 \text{ mm} \times 100 \text{ mm} \times \text{Within } 100 \text{ mm}$ ; five obstacles are set up in the space, all of which are spherical. The relevant parameter settings are as follows: the starting point coordinate is  $Start = [1, 1, 1]$ , and the target point coordinate is  $Goal = [100, 100, 100]$ . The GWO parameters are shown in Table 2.

**Table 2.** Parameter settings for Gray Wolf Algorithm.

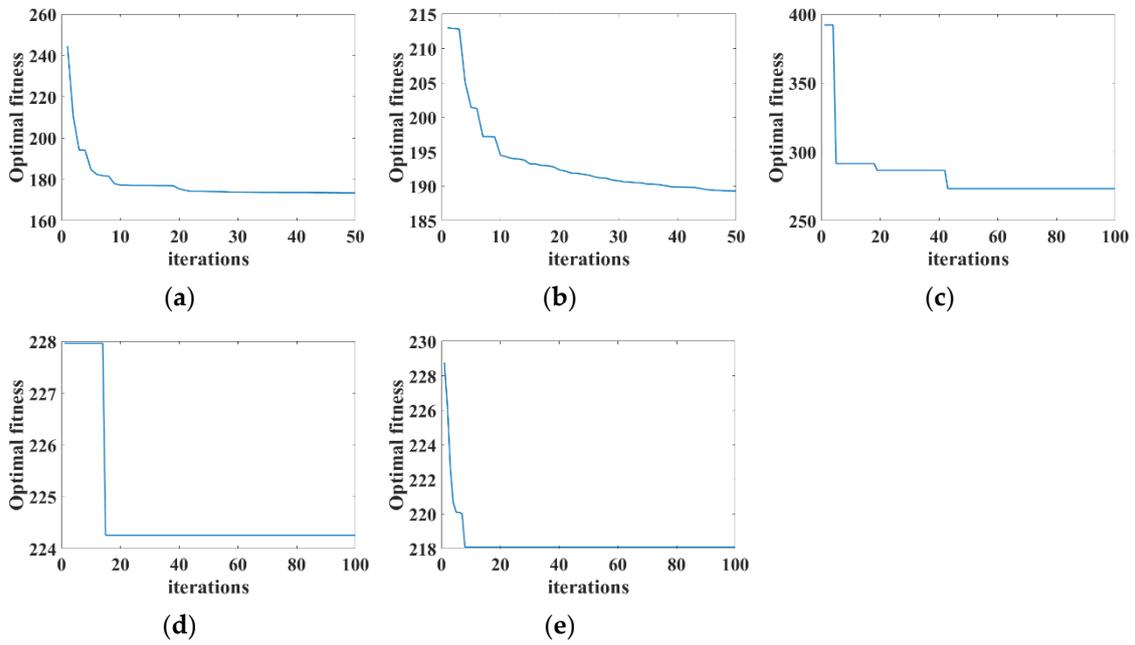
Parameter	Meaning	Value
$N$	Number of wolf packs	100
$\alpha$	Scale Factor of Beta Wolves	0.5
$S$	Step factor	100
$K_{max}$	Maximum number of walks	10
$\omega$	Distance determination factor	3
$\beta$	Wolf pack update ratio factor	3
$h$	Number of walking directions	20
$T_{max}$	Maximum number of iterations	50
$PC$	Selection probability	0.6

The parameter settings for basic AFSA are as follows: number of fish schools is 50, perceived distance of artificial fish is 50, the maximum step size of artificial fish movement is 3, crowding factor is 10, the maximum number of iterations for a single fish is 50, iterations is 100, and dimension of solution is 3. The parameter settings for basic GA are as follows: population size is 5, chromosome length is 5, cross probability is 0.8, selection probability is 0.5, and mutation probability is 0.2. The parameter settings for basic ACA are as follows: ant number is 100, pheromone importance factor is 10, heuristic function importance factor is 1; pheromone volatilization factor is 0.1, constant is 1, maximum number of iterations is 100, and number of sliced structures is 9.

#### 4.4. Simulation Results and Analysis

Figure 9 shows the convergence curve of the GWO for the entire path planning. It should be noted that the optimization problem studied in this article is a minimization problem, that is, the smaller the fitness value, the better the solution. From Figure 9, it can be seen that the GWO converges quickly to a very small fitness value. At around 20 iterations, the GWO falls into the local optima; however, as the number of iterations increases, the optimizer breaks through local optimal stagnation and continues to converge towards reducing the fitness value. The random dispersion strategy in GWO can effectively overcome premature convergence and enhance the algorithm's global search ability. The convergence process of the optimizer at other points on the target trajectory is similar to that point, but due to space limitations, it will not be presented in this article. It gradually converges to 174.1 in the 25th generation, with stronger global optimization ability. While converging quickly, the improved GWO also has such superior search accuracy in the later stage, resulting in a shorter path length and better quality. Compared with other basic SI algorithms, the improved GWO can effectively improve the problem of getting stuck in local optima. From Table 3 and Figure 10, it can be seen that due to the limitations of GWO's own algorithm characteristics, the fitness values of the current wolf and the head wolf need to be calculated and judged in each iteration of the three intelligent behaviors. Therefore, compared to ACA, GA, and AFSA, GWO has a longer running time. However, compared to the other three algorithms, the improved GWO algorithm plans a shorter path length, converges faster, and has a shorter average path. From Figure 10, it can be seen that the paths planned by the improved GWO algorithm and the other three algorithms can avoid obstacles, allowing the robotic arm to reach the target point smoothly. However, compared

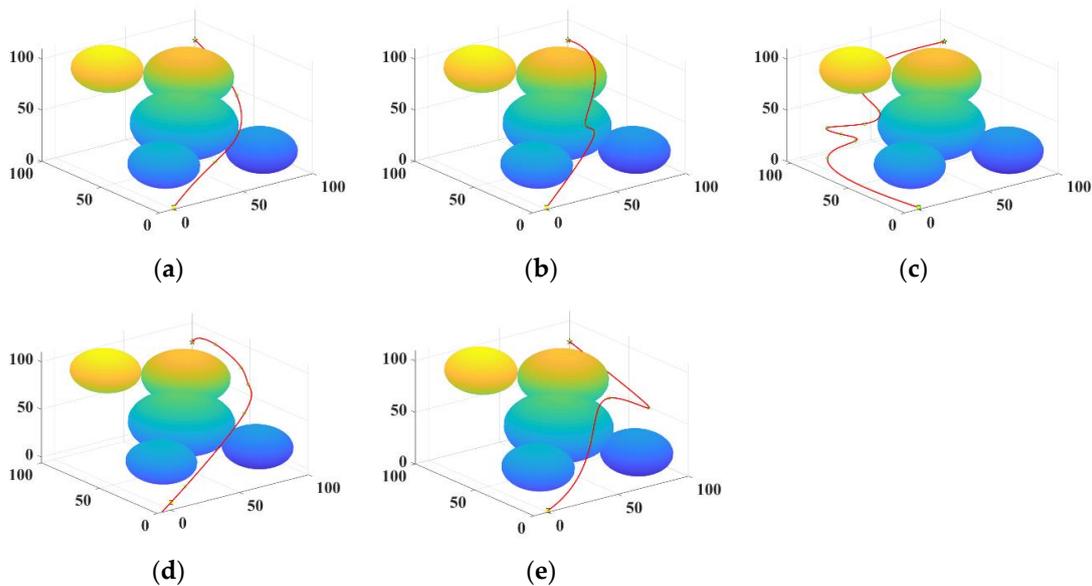
to the other three algorithms, the multi-strategy improved GWO algorithm produces a relatively shorter and better quality 3D path length.



**Figure 9.** Fitness value variation curve. (a) Improved GWO; (b) basic GWO; (c) basic ACA; (d) basic GA; (e) basic AFSA.

**Table 3.** Comparison results of four algorithms.

Algorithm	Path Length/m	Average Path/m	Running Time/s
Improved GWO	66.731	66.863	156.251
Basic GWO	68.315	69.561	162.561
Basic ACA	104.516	106.256	42.591
Basic GA	75.261	76.163	39.739
Basic AFSA	86.631	89.901	38.517



**Figure 10.** 3D path front view. (a) Improved GWO; (b) basic GWO; (c) basic ACA; (d) basic GA; (e) basic AFSA.

From this, the following conclusion can be drawn: the improved GWO algorithm has a faster convergence speed and a shorter planned path length. Improving the GWO algorithm can effectively improve the slow convergence speed and the tendency to fall into local optima caused by fixed step size in basic GWO algorithms, and the improved GWO algorithm has a stronger global optimization ability.

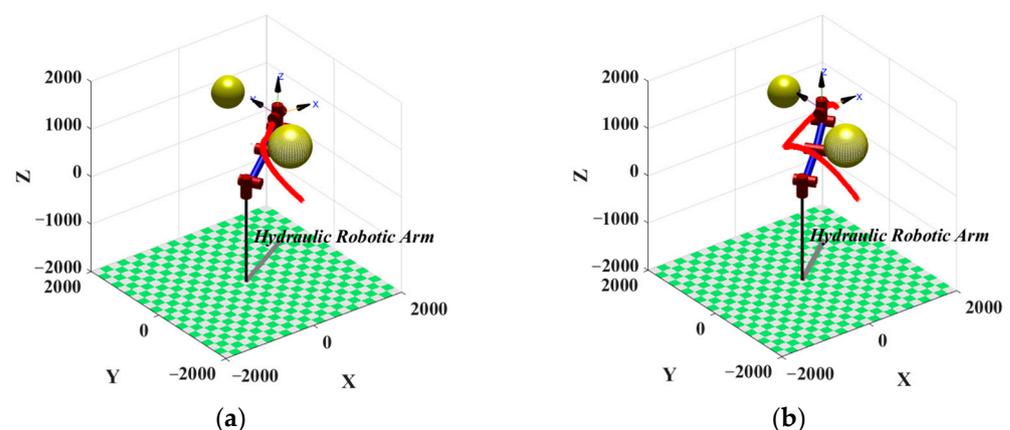
#### 4.5. Simulation of Robotic Arm Path Planning

The core idea of adjusting the pile body posture is to use the kinematic inverse solution knowledge introduced in Section 3 to solve the corresponding expected rotation angles of each joint based on the given coordinate values of the end effector of the vibrating pile driver. Then, the expected rotation angles of each joint are converted into the expected displacement of its corresponding oil cylinder.

Through the robotics toolbox in MATLAB, the multi-strategy improved Gray Wolf Algorithm and basic Gray Wolf Algorithm were applied to a six-degree-of-freedom pile driving robotic arm, and the performance of the two algorithms was tested simultaneously in a 3D environment. The starting position of the path [500, −1000, 0] and the ending position [1800, 1800, 800] were set, and two obstacles were set in the environment, both of which were enveloped with spherical shapes. The simulation results are shown in Table 4 and Figure 11, with the red line indicating the planned path. From Figure 11, it can be seen that both the improved GWO and the path planned by GWO can avoid obstacles and reach the target point, but the path planned by the improved GWO is shorter. From Table 4, it can be seen that compared to the basic GWO algorithm, the improved GWO algorithm reduces the 3D path length by 16.575% and the running time by 9.452%. In summary, it can be seen that compared to basic GWO algorithms, applying improved GWO algorithms to path planning on robotic arms yields better paths. Therefore, the improved algorithm can effectively achieve path planning and convert the expected displacement based on the given end effector's expected pose coordinate value, in order to achieve the adjustment of the pile's pose.

**Table 4.** Comparison results of two algorithms.

Algorithm	Path Length/mm	Average Path/mm	Running Time/s
Improved GWO	3751.04	3832.23	156.251
Basic GWO	4567.32	4593.62	172.561



**Figure 11.** Schematic diagram of robotic arm path planning. (a) Improved GWO; (b) basic GWO 5.

In the subsequent work, it is necessary to use sensors installed on the boom cylinder, stick cylinder, and four-bar drive cylinder to detect the real-time displacement of each group of cylinders in real time, and compare the detected real-time displacement of the cylinders with the expected displacement of the cylinders as input values for the controller.

Once there is a deviation between the detected cylinder displacement and the expected cylinder displacement, GWO will play an adjusting role to eliminate the deviation between the two and achieve a real-time attitude adjustment of the pile towards the desired attitude. Thus, monitoring the pose of the end effector of the working device and adjusting the pose of the end effector of the working device are carried out for experimental verification.

## 5. Conclusions

Addressing the imperative need for adjusting the pile body posture in the context of the side-clamp vibrating pile driver, this study focused on the mechanical arm mechanism. The research outcomes are summarized as follows:

- (1) Taking the side-clamp vibration pile driver as the research object, an improved D-H parameter method was used to establish the kinematic model of the working device of the pile driver, and the kinematic equations were solved in forward and inverse kinematics. A mathematical model was established based on inverse kinematics to solve the joint angles and their corresponding hydraulic cylinder propulsion stroke. The forward and inverse kinematic models of the working device of the pile driver using the MATLAB robotics toolbox were simulated to verify the correctness of the kinematic models. And based on the Monte Carlo method, the motion space simulation of the working device was carried out to solve the working range of the mechanical arm of the pile driver.
- (2) A multi-strategy improved GWO path planning algorithm is proposed to address the issue of operators being unable to directly obtain the vertical posture of the pile due to limited field of view, making it difficult to operate. The improved Grey Wolf algorithm was successfully applied to the three-dimensional path planning problem of robotic arms. Compared with the basic GWO algorithm, the improved GWO algorithm reduced the three-dimensional path length by 16.575% and the running time by 9.452%. The expected rotation angle of each joint was efficiently converted into the expected displacement of its corresponding oil cylinder, in order to achieve real-time pose adjustment of the pile to the desired pose.

**Author Contributions:** Conceptualization, J.D.; methodology, J.D. and H.X.; software, M.F.; validation, W.Z.; formal analysis, M.F.; investigation, M.F.; data curation, H.X.; writing—original draft preparation, J.D.; writing—review and editing, W.Z.; supervision, J.D. and Z.W.; project administration, J.D. and Z.W.; funding acquisition, J.D. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by Jiangsu Province Natural Science Fund, grant number BK20210495, Chinese Postdoctoral Science Foundation, grant number 2020M681761, National Natural Science Foundation of China, grant number 52174152.

**Data Availability Statement:** The original contributions presented in the study are included in the article; further inquiries can be directed to the corresponding author.

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

## References

1. Massarsch, K.R.; Fellenius, B.H. Engineering assessment of ground vibrations caused by impact pile driving. *Geotech. Eng. J. SEAGS AGSSEA* **2015**, *46*, 54–63.
2. Holeyman, A.; Whenham, V. Critical Review of the Hypervib1 Model to Assess Pile Vibro-Drivability. *Geotech. Geol. Eng.* **2017**, *35*, 1933–1951. [[CrossRef](#)]
3. Keskinen, E.; Launis, S.; Cotsaftis, M.; Raunisto, Y. Performance Analysis of Drive-Line Steering Methods in Excavator-Mounted Sheet-Piling Systems. *Comput. Aided Civ. Infrastruct. Eng.* **2001**, *16*, 229–238. [[CrossRef](#)]
4. Sasaki, T.; Kawahara, H.; Inoue, F.; Hashimoto, H. Circle fitting based pile positioning and machine pose estimation from range data for pile driver navigation. *IFAC Proc. Vol.* **2012**, *45*, 848–853. [[CrossRef](#)]
5. Huang, X.; Sasaki, T.; Hashimoto, H.; Inoue, F.; Zheng, B.; Masuda, T.; Ikeuchi, K. An accurate and efficient pile driver positioning system using laser range finder. In *Advances in Depth Image Analysis and Applications: International Workshop, WDIA 2012, Tsukuba, Japan, November 11, 2012, Revised Selected and Invited Papers*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 158–167.

6. Xie, Y.; Wang, Q.; Yao, L.; Meng, X.; Yang, Y. Integrated multi-sensor real time pile positioning model and its application for sea piling. *Remote Sens.* **2020**, *12*, 3227. [[CrossRef](#)]
7. Inoue, F.; Sasaki, T.; Huang, X.; Hashimoto, H. Development of position measurement system for construction pile using laser range finder. In Proceedings of the 28th International Symposium on Automation and Robotics in Construction, Seoul, Republic of Korea, 29 June–2 July 2011; pp. 794–800.
8. Sharma, S.; Ahmed, S.; Naseem, M.; Alnumay, W.S.; Singh, S.; Cho, G.H. A survey on applications of artificial intelligence for pre-parametric project cost and soil shear-strength estimation in construction and geotechnical engineering. *Sensors* **2021**, *21*, 463. [[CrossRef](#)] [[PubMed](#)]
9. Han, G.; Seo, D.; Ryu, J.H.; Kwon, T.H. RootBot: Root-inspired soft-growing robot for high-curvature directional excavation. *Acta Geotech.* **2023**, 1–13. [[CrossRef](#)]
10. Berner, M.; Sifferlinger, N.A. Analysis of excavation methods for a small-scale mining robot. In *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*; IAARC Publications: Kitakyushu, Japan, 2020; Volume 37, pp. 481–490.
11. Isaka, K.; Tsumura, K.; Watanabe, T.; Toyama, W.; Sugawara, M.; Yamada, Y.; Yoshida, H.; Nakamura, T. Development of underwater drilling robot based on earthworm locomotion. *IEEE Access* **2019**, *7*, 103127–103141. [[CrossRef](#)]
12. Guan, D.; Yang, N.; Lai, J.; Siu, M.F.F.; Jing, X.; Lau, C.K. Kinematic modeling and constraint analysis for robotic excavator operations in piling construction. *Autom. Constr.* **2021**, *126*, 103666. [[CrossRef](#)]
13. Sun, J.D.; Cao, G.Z.; Li, W.B.; Liang, Y.X.; Huang, S.D. Analytical inverse kinematic solution using the DH method for a 6-DOF robot. In Proceedings of the 2017 14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), Jeju, Republic of Korea, 28 June–1 July 2017; pp. 714–716.
14. Shrey, S.; Patil, S.; Pawar, N.; Lokhande, C.; Dandage, A.; Ghorpade, R.R. Forward kinematic analysis of 5-DOF LYNX6 robotic arm used in robot-assisted surgery. *Mater. Today Proc.* **2023**, *72*, 858–863. [[CrossRef](#)]
15. Ge, W.; Li, L.; Xing, E.; Lei, M.; Yang, S.; Zielinska, T. Design Study of 6-DOF Grinding Robot. In Proceedings of the 2019 IEEE International Conference on Mechatronics and Automation (ICMA), Tianjin, China, 4–7 August 2019; pp. 285–290.
16. Dou, R.; Yu, S.; Li, W.; Chen, P.; Xia, P.; Zhai, F.; Yokoi, H.; Jiang, Y. Inverse kinematics for a 7-DOF humanoid robotic arm with joint limit and end pose coupling. *Mech. Mach. Theory* **2022**, *169*, 104637. [[CrossRef](#)]
17. Chen, P.; Liu, L.; Yu, F.; Li, H.; Yang, F.; Wang, X. A geometrical method for inverse kinematics of a kind of humanoid manipulator. *Robot* **2012**, *34*, 211–216. [[CrossRef](#)]
18. Karakaya, S.; Kucukyildiz, G.; Ocak, H. A new mobile robot toolbox for MATLAB. *J. Intell. Robot. Syst.* **2017**, *87*, 125–140. [[CrossRef](#)]
19. Luo, J.; Wen, Q.; He, J.; Ye, B. Workspace Analysis of 7-DOF Humanoid Robotic Arm Based on Monte Carlo Method. In Proceedings of the 2015 International Conference on Intelligent Systems Research and Mechatronics Engineering, Zhengzhou, China, 11–13 April 2015; pp. 48–51.
20. Peidró, A.; Reinoso, Ó.; Gil, A.; Marín, J.M.; Payá, L. An improved Monte Carlo method based on Gaussian growth to calculate the workspace of robots. *Eng. Appl. Artif. Intell.* **2017**, *64*, 197–207. [[CrossRef](#)]
21. Wang, C.; Han, Q.S. Kinematics of Six-degree-of-freedom Series Manipulator and Its Working Space. *Modul. Mach. Tool Autom. Manuf. Tech.* **2020**, *6*, 32–36.
22. Han, T.; Li, J.; Huang, Y.; Xu, S.; Xu, J. Research on trajectory planning algorithm of manipulator arm of coal mine rescue robot. *Ind. Mine Autom.* **2021**, *47*, 45–52.
23. Szczepanski, R.; Bereit, A.; Tarczewski, T. Efficient local path planning algorithm using artificial potential field supported by augmented reality. *Energies* **2021**, *14*, 6642. [[CrossRef](#)]
24. Szczepanski, R.; Tarczewski, T.; Erwinski, K. Energy efficient local path planning algorithm based on predictive artificial potential field. *IEEE Access* **2022**, *10*, 39729–39742. [[CrossRef](#)]
25. Li, C.; Meng, F.; Ma, H.; Wang, J.; Meng, M.Q.H. Relevant region sampling strategy with adaptive heuristic for asymptotically optimal path planning. *Biomim. Intell. Robot.* **2023**, *3*, 100113. [[CrossRef](#)]
26. Lei, T.; Chintam, P.; Carruth, D.W.; Jan, G.E.; Luo, C. Human-autonomy teaming-based robot informative path planning and mapping algorithms with tree search mechanism. In Proceedings of the 2022 IEEE 3rd International Conference on Human-Machine Systems (ICHMS), Orlando, FL, USA, 17–19 November 2022; pp. 1–6.
27. Lei, T.; Chintam, P.; Luo, C.; Rahimi, S. Multi-Robot Directed Coverage Path Planning in Row-based Environments. In Proceedings of the 2022 IEEE Fifth International Conference on Artificial Intelligence and Knowledge Engineering (AIKE), Laguna Hills, CA, USA, 19–21 September 2022; pp. 114–121.
28. Li, X.; Gao, X.; Zhang, W.; Hao, L. Smooth and collision-free trajectory generation in cluttered environments using cubic B-spline form. *Mech. Mach. Theory* **2022**, *169*, 104606. [[CrossRef](#)]
29. Short, D.; Lei, T.; Luo, C.; Carruth, D.W.; Bi, Z. A bio-inspired algorithm in image-based path planning and localization using visual features and maps. *Intell. Robot.* **2023**, *3*, 222–241. [[CrossRef](#)]
30. Wang, J.; Chen, W.; Xiao, X.; Xu, Y.; Li, C.; Jia, X.; Meng, M.Q.H. A survey of the development of biomimetic intelligence and robotics. *Biomim. Intell. Robot.* **2021**, *1*, 100001. [[CrossRef](#)]
31. Rezoug, A.; Iqbal, J.; Tadjine, M. Extended grey wolf optimization-based adaptive fast nonsingular terminal sliding mode control of a robotic manipulator. *Proc. Inst. Mech. Eng. Part I J. Syst. Control Eng.* **2022**, *236*, 1738–1754. [[CrossRef](#)]

32. Pehlivanoglu, Y.V.; Pehlivanoglu, P. An enhanced genetic algorithm for path planning of autonomous UAV in target coverage problems. *Appl. Soft Comput.* **2021**, *112*, 107796. [[CrossRef](#)]
33. Shi, J.; Tan, L.; Zhang, H.; Lian, X.; Xu, T. Adaptive multi-UAV path planning method based on improved gray wolf algorithm. *Comput. Electr. Eng.* **2022**, *104*, 108377.
34. Cheng, X.; Li, J.; Zheng, C.; Zhang, J.; Zhao, M. An improved PSO-GWO algorithm with chaos and adaptive inertial weight for robot path planning. *Front. Neurobotics* **2021**, *15*, 770361. [[CrossRef](#)] [[PubMed](#)]
35. Zhang, C.; Liu, Y.; Hu, C. Path planning with time windows for multiple UAVs based on gray wolf algorithm. *Biomimetics* **2022**, *7*, 225. [[CrossRef](#)]
36. Qazani, M.R.C.; Asadi, H.; Arogbonlo, A.; Rahimzadeh, G.; Mohamed, S.; Pedrammehr, S.; Lim, C.P.; Nahavandi, S. Whale optimization algorithm for weight tuning of a model predictive control-based motion cueing algorithm. In Proceedings of the 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Melbourne, Australia, 17–20 October 2021; pp. 1042–1048.
37. Deckner, F.; Viking, K. Wave patterns in the ground: Case studies related to vibratory sheet pile driving. *Geotech. Geol. Eng.* **2017**, *35*, 2863–2878. [[CrossRef](#)]
38. Khan, M.F.; ul Islam, R.; Iqbal, J. Control strategies for robotic manipulators. In Proceedings of the 2012 International Conference of Robotics and Artificial Intelligence, Rawalpindi, Pakistan, 22–23 October 2012; pp. 26–33.
39. Fausto, F.; Reyna-Orta, A.; Cuevas, E.; Andrade, Á.G.; Perez-Cisneros, M. From ants to whales: Metaheuristics for all tastes. *Artif. Intell. Rev.* **2020**, *53*, 753–810. [[CrossRef](#)]
40. Bansal, J.C.; Singh, P.K.; Pal, N.R. (Eds.) *Evolutionary and Swarm Intelligence Algorithms*; Springer: Cham, Switzerland, 2019.
41. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [[CrossRef](#)]

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