

Research Article

Extending FABRIK with Obstacle Avoidance for Solving the Inverse Kinematics Problem

Songqiao Tao ¹, Huajin Tao,² and Yumeng Yang³

¹School of Intelligent Manufacturing, Wuhan Technical College of Communications, Wuhan 430065, China

²Faculty of Physics and Electronic Science, Hubei University, Wuhan 430062, China

³Jiangsu Midea Cleaning Appliance Co., Ltd, Suzhou 215131, China

Correspondence should be addressed to Songqiao Tao; taosongqiao@163.com

Received 2 March 2021; Revised 3 April 2021; Accepted 20 April 2021; Published 27 April 2021

Academic Editor: Yaoyao Wang

Copyright © 2021 Songqiao Tao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Inverse kinematics (IK) has been extensively applied in the areas of robotics, computer animation, ergonomics, and gaming. Typically, IK determines the joint configurations of a robot model and achieves a desired end-effector position in robotics. Since forward and backward teaching inverse kinematics (FABRIK) is a forward and backward iterative method that finds updated joint positions by locating a point on a line instead of using angle rotations or matrices, it has the advantages of fast convergence, low computational cost, and visualizing realistic poses. However, the manipulators usually work in a complex environment. So, the kinematic chains are easy to produce the interference with their surrounding scenarios. To resolve the above mentioned problem, a two-step obstacle avoidance technology is proposed to extend the basic FABRIK in this study. The first step is a heuristic method that locates the updated linkage bar, the root joint, and the target position in a line, so that the interference can be eliminated in most cases. In the second step, multiple random rotation strategies are adopted to eliminate the interference that has not been eliminated in the first step. Experimental results have shown that the extending FABRIK has the obstacle avoidance ability.

1. Introduction

In recent years, IK has been extensively applied in the areas of robotics, computer animation, ergonomics, and gaming [1]. More generally, IK is the use of kinematic equations to determine the joint parameters of a manipulator, so that the end effector is moved to a desired position [2]. For a serial manipulator, for instance, IK is to find the angles of all joints in the manipulator desiring position of the end effector.

Aristidou et al. [1] presented a forward and backward iterative technology called FABRIK to seek out updated joint positions by locating a point on a line instead of using angle rotations or matrices. The FABRIK has the advantages of converge in few iterations, low computational complexity, and visualizing realistic poses. Compared with the CCD algorithm, the FABRIK produces visually smooth movements without oscillations and discontinuities and handles multiple chains with multiple end effectors. However, the work environment of a manipulator is usually complex. So,

the joint chain is easy to interfere with its surrounding scenarios. To resolve the above mentioned problem, we introduce a two-step obstacle avoidance technology to extend the basic FABRIK in this study.

The remainder of this study is organized as follows. First, a brief review of the related work is presented in Section 2. Second, the basic FABRIK is introduced in Section 3. In Section 4, an extending FABRIK with obstacle avoidance is presented in detail. Then, Section 5 presents some results of experiments that apply the proposed approach to solve IK problems. Finally, the study ends up with some conclusions in Section 6.

2. Related Work

Since, recently, IK problems have become an active research topic, a lot of works have been performed in disciplines such as use of robotics, computer animation, ergonomics, and gaming. Instead of giving a complete overview to IK

problems, which can be found in [2–4], here we focus on the literature in analytical methods and numerical methods.

2.1. Analytical Methods. Analytical methods usually find all possible solutions as a function of the mechanism’s lengths, its initial position, and the rotation constraints. To just calculate a single solution, analytical solutions are usually based on some assumptions [2]. Typical analytical IK solutions for dealing with general 6R manipulators early appeared in [5, 6]. Diankov [7] has developed a tool called IKFast to solve robot IK equations in an analytic form and deployed motion planning algorithms in real-world robotics applications. More recently, Liu et al. [8] presented an analytical IK solver that realizes human-like configurations by means of selecting a redundancy resolution. Zhao et al. [9] proposed a new IK method to obtain anthropomorphic arm postures for a given end-effector position and orientation. The position and orientation of the end-effector determines the wrist position. Then, the elbow twist angle is adopted to solve the redundancy resolution of human arm. Tian et al. [10] proposed an analytical IK method for 7-DOF anthropomorphic manipulators. They realized the combined control of global arm configuration manifolds and local self-motion without offset by defining a new arm reference plane for 7-DOF anthropomorphic manipulators. Tong et al. [11] presented an analytical IK parameterized method for redundant sliding manipulators. They solved the IK analytically and used the Newton–Raphson algorithm for secondary adjustment. Analytical IK solutions are usually global, nonsingular, fast, simple, and reliable. However, the nonlinear characteristics of the kinematic equations make analytical IK methods unsuitable for redundant systems with more than 7-DoF.

2.2. Numerical Methods. Numerical methods are usually the iterative approaches that formulate IK problems using a cost function to be minimized. Typical numerical IK solutions have Jacobian, Newton, and Heuristic methods.

Jacobian approaches usually use Jacobian matrix to express the linear approximation of nonlinear functions, which repeatedly change the configuration of a complete chain, such that it gradually makes the end-effector approach the target position at each step. Some different ways have been proposed for calculating or approximating the inverse of the Jacobian matrix, such as the Jacobian transpose [12, 13], singular value decomposition [14, 15], damped least squares [16, 17], and incorporating constraints [18, 19]. Jacobian methods usually have smooth posture. However, they have the disadvantages of high computational cost, complex matrix calculations, and singularity problems [20]. Recently, Libretto et al. [21] presented singularity-free solutions for inverse kinematics of degenerate mobile robots. They solved the IK problem using degenerate wheel structure that does not suffer from singularity problems.

Newton approaches usually find target configurations that are posed as solutions to a minimization problem. Therefore, they return smooth motion without erratic discontinuities [20]. The most famous approaches are the

Broyden method, Powell method, and the Broyden, Fletcher, Goldfarb, and Shanno method [22]. To resolve the inverse problems that the manipulator link lengths, joint angles, and end-effector uncertainty bounds are given, Kumar et al. [23] adopted an interval Newton method to solve the IK and redundancy resolution of a serial redundant manipulator. To resolve the problem that the 6-DOF joint robot with offset wrist cannot satisfy the Pieper rule, Lei et al. [24] presented a modified Newton iteration method to solve IK problem for 6-DOF joint robot with offset wrist. Duleba and Karcz-Duleba [25] presented accelerating Newton algorithms of inverse kinematics for robot manipulators. Erleben and Andrews [26] proposed a numerical approach for calculating the exact Hessian of an IK system with prismatic, revolute, and spherical joints. The advantage of Newton methods is that they have nonsingular problems. However, their disadvantages are complex, difficult to implement, and have high computational cost at each iteration.

Heuristic approaches usually use simple ways for solving the IK problem without using complex equations and calculations. Typical heuristic methods for solving IK problems have cyclic coordinate descent (CCD) [27–30] and triangulation IK [31, 32]. CCD is an iterative heuristic technique that is suitable for interactive control of an articulated body, which is early proposed by Wang and Chen [27]. Kenwright has given a comprehensive review of CCD in [33]. CCD provides a numerically stable solution and has the advantage of low computational cost in an iteration. However, CCD may generate unrealistic postures and has difficulty in solving IK problems with multiple end effectors. As triangulation IK calculates the joint angle by using the cosine law, which usually finds the desired position in few iterations and has low computational cost. However, using triangulation IK to solve IK problems with multiple end effectors is an intractable task. FABRIK is another heuristic method for solving the IK problem, which finds the joint positions by locating a point on a line instead of using angle rotations [1]. Some researchers have developed extending FABRIK methods because of its efficiency and simplicity. Jing and Pelachaud [34] presented an efficient energy transfer IK solution using a variation of FABRIK. A mass-spring model with minimizing the force energy is adopted to adjust the joint positions. Agrawalet and Michiel [35] introduced a data-driven prior to warm start FABRIK for the production of high-quality motions. Tenneti and Sarkar [36] proposed a modified FABRIK method to solve the IK problem. They adopted Denavit and Hartenberg parameters to define the geometry of a serial robot manipulator. Santos et al. [37] presented a new IK method called M-FABRIK for mobile manipulator robots. They may use various criteria (such as decreasing convergence time, avoiding contact with obstacles, and avoiding joint angle limits) to locate the robot.

3. The Basic FABRIK

The basic FABRIK can be found in section 3 of [1]. Here, we use a manipulator with a single target T and four joints P_1, P_2, P_3, P_4 to explain a full iteration. In Figure 1, P_1 and P_4 , respectively, represent the root joint and the end effector,

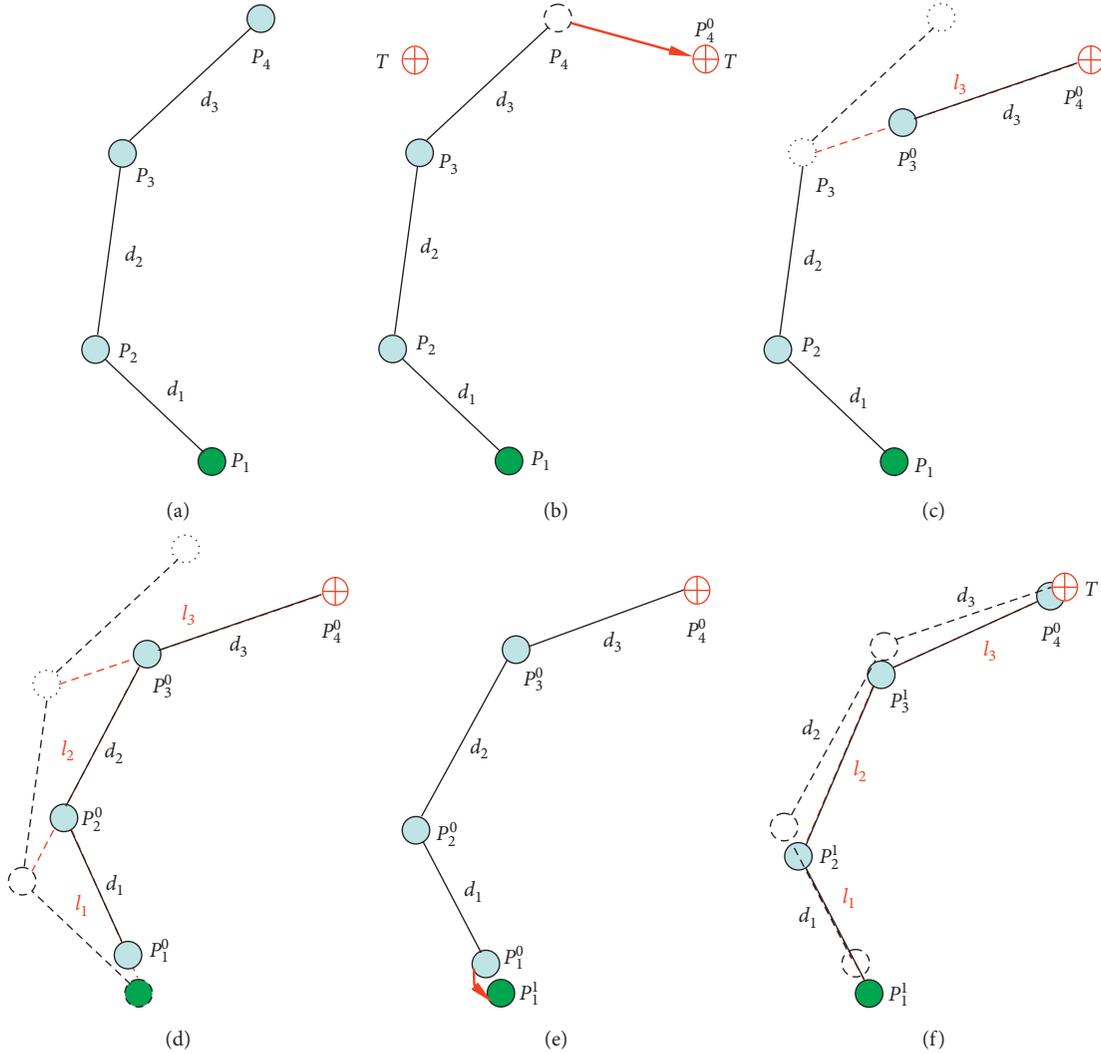


FIGURE 1: An example of a full iteration for a manipulator with a single target and four joints. (a) The initial state of the manipulator and the target position. (b) The end effector P_4 moved to the target position (T). (c) The middle joint P_3 's updated position P_3^0 . (d) The updated positions P_2^0 and P_1^0 . (e) P_1^0 moved to the initial position P_1 (P_1^1 is the updated position of P_1^0). (f) Updated positions P_2^1 , P_3^1 , and P_4^1 .

as well as P_2 and P_3 , respectively, express the middle joints, and $d_j = |P_{j+1} - P_j|$ ($j = 1, 2, 3$) is the distance between two adjacent joints P_{j+1} and P_j .

First, calculate the distances $d_t = |T - P_1|$ and $d_j = |P_{j+1} - P_j|$ ($j = 1, 2, 3$). Then, check the target position T whether it is reachable or not. If $d_t < |d_1 + d_2 + d_3|$, the target position T is reachable, and a full iteration will be implemented in two stages; else, T is unreachable, and the FABRIK is terminated. In the first stage, Figures 1(b)–1(d) represent the forward search. Figure 1(b) shows that the end effector P_4 is moved to the target position T . In Figure 1(c), the updated position P_3^0 of the middle joint P_3 is found in the line l_3 , which has the distance $|P_4^0 - P_3^0| = |P_4 - P_3| = d_3$. Similarly, the updated positions P_2^0 and P_1^0 for the middle joint P_2 and the root joint P_1 are, respectively, shown in Figure 1(d). In the second stage, Figures 1(e)–1(f) express the backward search. Figure 1(e) shows that P_1^0 is moved to the initial position P_1 (P_1^1 is the updated position of P_1^0). Similar to Figures 1(b) and 1(c), the updated positions P_2^1 , P_3^1 , and P_4^1 are found in Figure 1(f).

After a full iteration is performed, the updated position P_4^1 of the end effector P_4 is very close to the target point T . Repeat above iteration procedures until the distance between the end effector P_4 and the target position T is less than the convergence precision β .

The limitation of the FABRIK (a singularity problem) is explained and resolved in section 3.2 of [20]. Besides, the proof of FABRIK converges can be found in section 3.3 of [20].

4. Extending FABRIK with Obstacle Avoidance

Since a manipulator usually works in a complex environment, the joint chain may have the interference with its surroundings. Thus, the surrounding scenarios between the desired position T and the initial position P_1 should be considered for extending the basic FABRIK. So, a two-step obstacle avoidance technique is adopted to ensure that the extending FABRIK has the obstacle avoidance ability.

4.1. The Two-Step Obstacle Avoidance Technique. The first step is a heuristic way that locates the updated linkage bar, the root joint, and the target position in a line, so that the interference can be eliminated in most cases. If a linkage bar $P_{j-1}^v P_j^0$ ($j = 3, 4, \dots, n$) ($P_{j-1}^v P_j^0$ ($j = 2, 3, \dots, n-1$)) has the interference with its surroundings in the forward (backward) search, it will rotate along a local optimization direction to an updated position $P_{j-1}^l P_j^0$ ($P_{j-1}^l P_j^0$) (Figures 2(b) and 2(e)). The local optimization direction is that the root joint P_1 , the updated linkage bar $P_{j-1}^l P_j^0$ ($P_{j-1}^l P_j^0$), and the target position T are located in a line, and the updated position P_{j-1}^l (P_j^0) is close to the root joint P_1 (the target position T). The reasons are that if P_1 , $P_{j-1}^l P_j^0$ ($P_{j-1}^l P_j^0$) and T are collinear, the interference between $P_{j-1}^l P_j^0$ ($P_{j-1}^l P_j^0$) and the obstacles can be eliminated in most cases, and when P_{j-1}^l (P_j^0) is as close to the root joint P_1 (the target position T) as possible, the joint chains may have slight changes.

In the second step, multiple random rotation strategies are adopted to eliminate the interference that has not been eliminated in the first step (Figure 2(e)). Since there is the interference between the linkage bar $P_{j-1}^l P_j^0$ ($P_{j-1}^l P_j^0$) and the obstacle in Figure 2(e), $P_{j-1}^l P_j^0$ ($P_{j-1}^l P_j^0$) randomly rotates many times around P_j^0 (P_{j-1}^l) in the forward (backward) search so that the interference is eliminated. And the optimal solution $P_{j-1}^k P_j^0$ is selected from k updated positions of $P_{j-1}^k P_j^0$ ($P_{j-1}^k P_j^0$), which corresponds to the position with the minimum rotation angle (Figure 2(f)). As only the linkage bar $P_{j-1}^v P_j^0$ ($P_{j-1}^v P_j^0$) is optimized in an iteration, the rotation technique is just a local optimization. However, the FABRIK is a heuristic iterative approach, subsequent iterations will gradually improve previous results, and the final result is very close to the optimum solution.

4.2. An Example for Extending FABRIK with Obstacle Avoidance. In the following, we use Figure 2 to explain the extending FABRIK with obstacle avoidance. The circles with shadow express the obstacles in Figure 2. Two cases are that the obstacles with various positions are listed in Figures 2(a)–2(c) and Figures 2(d)–2(f). Figures 2(a)–2(c) represent the case that P_1 , $P_3 P_4^0$, and T are collinear, and there is no interference between $P_3 P_4^0$ and the obstacle. Thus, $P_3 P_4^0$ does not need to rotate. Conversely, Figures 2(d)–2(f) express the case that there is the interference between $P_3 P_4^0$ and the obstacle.

First, the initial state of the manipulator, the target position, and the obstacle are listed in Figures 2(a) and 2(d). Then, Figures 2(b) and 2(e) are that the end effector P_4 is moved to the target position T (P_4^0 is the updated position of P_4), and the basic FABRIK is adopted to find the updated position P_3^v of P_3 . However, there is the interference between the updated linkage bar $P_3^v P_4^0$ and the obstacle. So, $P_3^v P_4^0$ took a rotation along the local optimization direction to its updated position $P_3^l P_4^0$, which has P_1 , $P_3^l P_4^0$, and T in a line. Since there is no interference between the updated linkage bar $P_3^l P_4^0$ and the obstacle in Figure 2(b), the local optimization rotation is terminated. However, $P_3^l P_4^0$ is still interferential with the obstacle in Figure 2(e). To resolve above mentioned problem, multiple random rotations are used for finding k updated positions of $P_3^k P_4^0$. And the optimization position $P_3^k P_4^0$ is selected from

$P_3^k P_4^0$, which has the smallest rotation angle. Finally, the basic FABRIK is adopted to find the updated positions P_2^0 and P_1^0 in Figures 2(c) and 2(f).

4.3. The Procedure of Extending FABRIK with Obstacle Avoidance. In the following, we give the procedures of extending FABRIK with obstacle avoidance in detail.

Input. The target position T , the root joint P_1 , the end effector P_n , middle joint P_{j+1} ($j = 1, 2, \dots, n-2$), the convergent precision β , and the maximum number of iterations n_{\max}

Output. Updated joint positions P_j ($j = 1, 2, \dots, n$)

Step 1. Calculate the distances $d_j = |P_{j+1} - P_j|$ ($j = 1, 2, \dots, n-1$) and $d_t = |T - P_1|$

Step 2. Checks whether the target position T is within reach

If $d_t < \sum_{j=1}^{n-1} d_j$, the target position T is reachable, go to the next step; else, go to Step 6.

Step 3. Forward search with obstacle avoidance

Step 3.1. Move the end effector P_n to the target position T (P_n^0 is the updated position of P_n), and find the updated position P_{n-1}^v of the middle joint P_{n-1} in the line $P_{n-1} P_n^0$, which has the distance $|P_n^0 - P_{n-1}^v| = |P_n - P_{n-1}| = d_{n-1}$;

Step 3.2. Check whether there is the interference between the linkage bar $P_{n-1}^v P_n^0$ and the obstacle. If the interference exists, go to the next step; else, go to Step 3.6;

Step 3.3. The linkage bar $P_{n-1}^v P_n^0$ takes a rotation around P_n^0 , until P_1 , $P_{n-1}^l P_n^0$ are located in a line; then, P_{n-1}^v has its updated position P_{n-1}^l , and P_{n-1}^l is close to the root joint P_1 ;

Step 3.4. Check whether there is the interference between the linkage bar $P_{n-1}^l P_n^0$ and the obstacle. If the interference exists, go to the next step; else, go to Step 3.6;

Step 3.5. The linkage bar $P_{n-1}^l P_n^0$ rotates many times around P_n^0 at random, until there is no interference between $P_{n-1}^l P_n^0$ and the obstacle; then, select the optimization position P_{n-1}^k from k updated positions of P_{n-1}^k ;

Step 3.6. Repeat Step 3.1–Step 3.5, and find updated positions $P_{n-2}^0, \dots, P_2^0, P_1^0$.

Step 4. Backward search with obstacle avoidance

Step 4.1. Move P_1^0 to the root joint P_1 (P_1^1 is the updated position of P_1^0), and find the updated position P_2^v of the middle joint P_2 in the line $P_2 P_1^1$, which has the distance $|P_2^v - P_1^1| = |P_2 - P_1| = d_1$;

Step 4.2. Check whether there is the interference between the linkage bar $P_1^1 P_2^v$ and the obstacle. If the interference exists, go to the next step; else, go to Step 4.6;

Step 4.3. The linkage bar $P_1^1 P_2^v$ takes a rotation around P_1^1 , until T , $P_1^1 P_2^l$ are located in a line; then, P_2^v has its

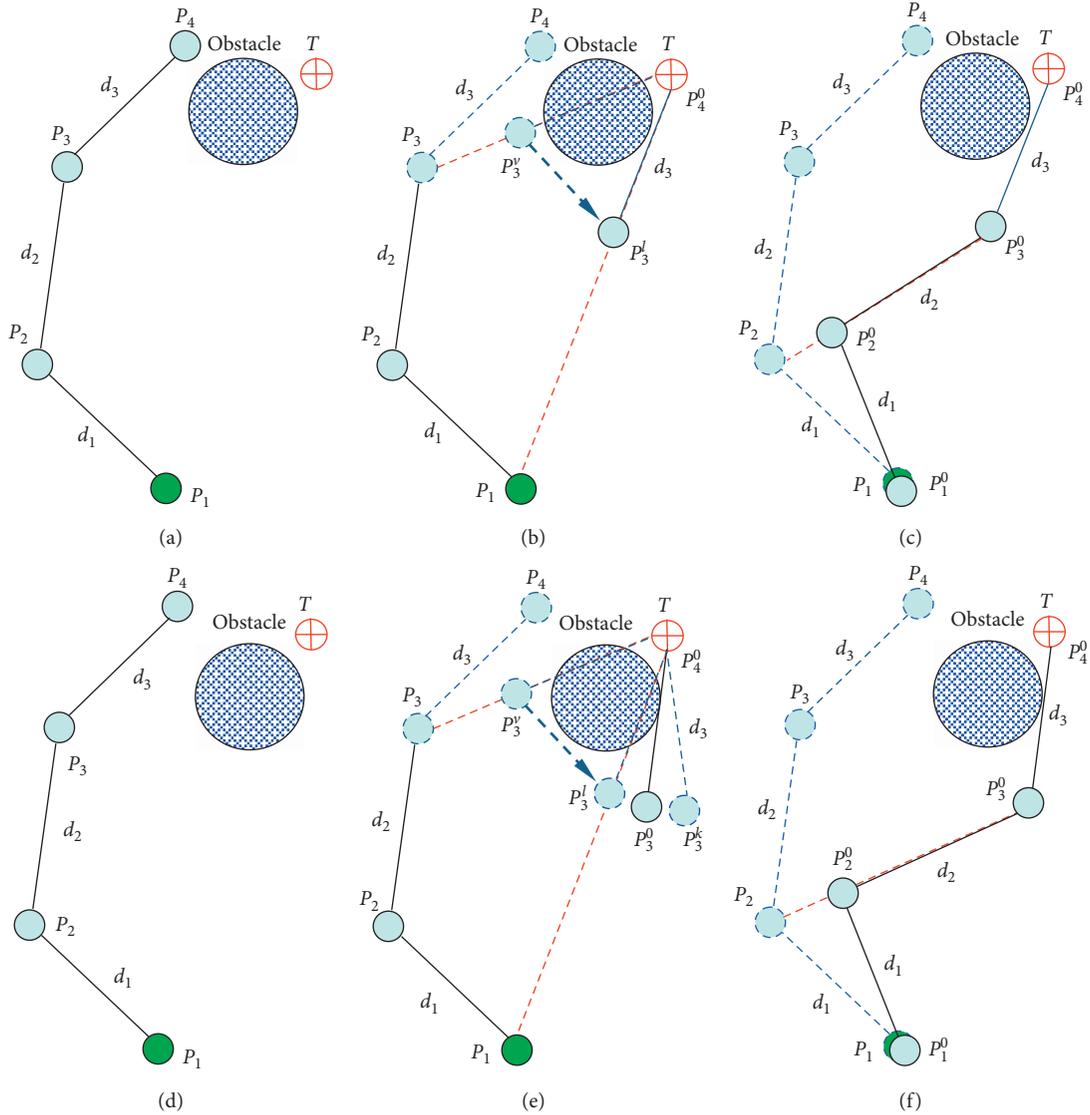


FIGURE 2: Extending FABRIK with obstacle avoidance. (a), (d) The initial state of the manipulator, the target position, and the obstacle. (b), (e) P_4 moved to the target position (T) and the optimization P_3^0 found. (c), (f) The basic FABRIK used to find the updated positions P_2^0 and P_1^0 .

updated position P_2^l , and P_2^l is close to the target position T ;

Step 4.4. Check whether there is the interference between the linkage bar $P_1^l P_2^l$ and the obstacle. If the interference exists, go to the next step; else, go to Step 4.6;

Step 4.5. The linkage bar $P_1^l P_2^l$ rotates many times around P_1^l at random, until there is no interference between $P_1^l P_2^l$ and the obstacle; then, select the optimization position P_2^l from k updated positions of P_2^k ;

Step 4.6. Repeat Step 4.1–Step 4.5, and updated positions $P_3^l, P_4^l, \dots, P_n^l$.

Step 5. Check whether the updated linkage bars $P_1^l P_2^l, P_2^l P_3^l, \dots, P_{n-1}^l P_n^l$ have interference with the obstacles and if the distance $d = |T - P_n^l| < \beta$:

If the two constraint conditions are true, the target position is reachable, and the updated joint positions $P_j (j = 1, 2, \dots, n)$ are output; otherwise, go to Step 2. When the number of iterations $n \geq n_{\max}$, two constraint conditions are false, and the target position is unreachable.

Step 6. End.

5. Experimental Results

To validate the proposed method, we use Matlab program to test if the extending FABRIK has the obstacle avoidance ability. The test subject is a four-joint manipulator with the coordinates $(0, 0)$, $(0, 1)$, $(0, 2)$, $(0, 3)$, and $(0, 4)$. Here, three groups of target positions with different distances are used to

TABLE 1: The experiment results of extending FABRIK with obstacle avoidance in test 1.

Whether the target is within reach or not?		The number of iterations	The distance between the target position and the end effector (mm)	Run time (s)	The results
1	Yes	77	9.8638e-04	0.2659	Figure 3(a)
2	Yes	55	9.8044e-04	0.2206	Figure 3(b)
3	Yes	9	8.5390e-04	0.1097	Figure 3(c)
4	Yes	30	8.7228e-04	0.2102	Figure 3(d)
5	Yes	10	9.2472e-04	0.1272	Figure 3(e)
6	Yes	23	8.5583e-04	0.1408	Figure 3(f)
7	No	100	0.1195	0.8193	Figure 3(g)
8	No	100	0.3136	0.9215	Figure 3(h)
9	No	100	0.2938	1.1800	Figure 3(i)

TABLE 2: The experiment results of extending FABRIK with obstacle avoidance in test 2.

Whether the target is within reach or not?		The number of iterations	The distance between the target position and the end effector (mm)	Run time (s)	The results
1	Yes	7	8.6819e-04	0.0880	Figure 4(a)
2	Yes	4	6.1996e-04	0.0898	Figure 4(b)
3	Yes	3	4.2464e-04	0.0853	Figure 4(c)
4	Yes	2	5.1837e-04	0.1160	Figure 4(d)
5	Yes	5	3.0560e-04	0.1012	Figure 4(e)
6	Yes	5	2.5677e-06	0.0984	Figure 4(f)
7	Yes	3	4.9393e-06	0.1119	Figure 4(g)
8	Yes	3	6.0699e-04	0.1326	Figure 4(h)
9	Yes	11	4.6694e-04	0.2120	Figure 4(i)

TABLE 3: The experiment results of extending FABRIK with obstacle avoidance in test 3.

Whether the target is within reach or not?		The number of iterations	The distance between the target position and the end effector (mm)	Run time (s)	The results
1	Yes	3	4.9565e-04	0.0748	Figure 5(a)
2	Yes	2	3.7932e-05	0.0826	Figure 5(b)
3	Yes	2	1.0127e-05	0.0917	Figure 5(c)
4	Yes	1	5.0544e-04	0.0884	Figure 5(d)
5	Yes	2	3.1292e-05	0.0918	Figure 5(e)
6	Yes	2	2.7737e-04	0.0922	Figure 5(f)
7	Yes	4	5.5242e-04	0.1394	Figure 5(g)
8	Yes	6	2.7428e-05	0.1448	Figure 5(h)
9	Yes	3	3.7580e-05	0.1350	Figure 5(i)

explain the influence of distance on the experimental results. The coordinates of target positions in test 1–test 3 are, respectively, set at (2.8, 2.8), (2.4, 2.4), and (2.0, 2.0). Besides, five round obstacles with various centers and radius are selected to test the obstacle avoidance adaptability of the proposed method. The centers of five round obstacles in test 1–test 3 are, respectively, located in three rectangles with diagonal vertices $((0.3, 0.3), (2.5, 2.5))$, $((0.3, 0.3), (2.1, 2.1))$, and $((0.3, 0.3), (1.7, 1.7))$, and their radius are randomly generated between 0.1 and 0.3. The random rotating number k , the maximum number of iterations n_{\max} , and the convergence precision β are, respectively, set at 10, 100, and 0.001. The tests are carried out on a computer with an AMD 3.6 GHz CPU and 8.0 GB RAM.

In three groups of the experiments, the extending FABRIK is run nine times in succession to test if it has the ability to avoid various obstacles and reach different target positions. The statistical results in test 1–test 3 are, respectively, given in Tables 1–3, and visualization experiment

results are, respectively, shown in Figures 3–5 and. Tables 1–3 present the number of iterations needed to reach the target, run time, and the distance between the target position and the end effector. It has shown that the extending FABRIK has lower computational cost and faster convergence speed since it treats finding the joint locations as a problem of finding a point on a line. And the distance between the target position and the root joint is closer, and the number of iterations and run time are less.

In Figures 3–5, the joint chains are labeled as blue, as well as the target positions and obstacles are, respectively, represented by red circles and pink circles. Meanwhile, vertical joint chains express the initial state, and other joint chains are the final postures that the end effectors reach the desired positions. It can be seen from Figure 3 that the extending FABRIK successfully avoids obstacles six times to reach their desired positions (Figures 3(a)–3(f)), and the target position is without reach thrice (Figures 3(g)–3(i)). In Figures 4 and 5, the extending FABRIK successfully avoids obstacles nine

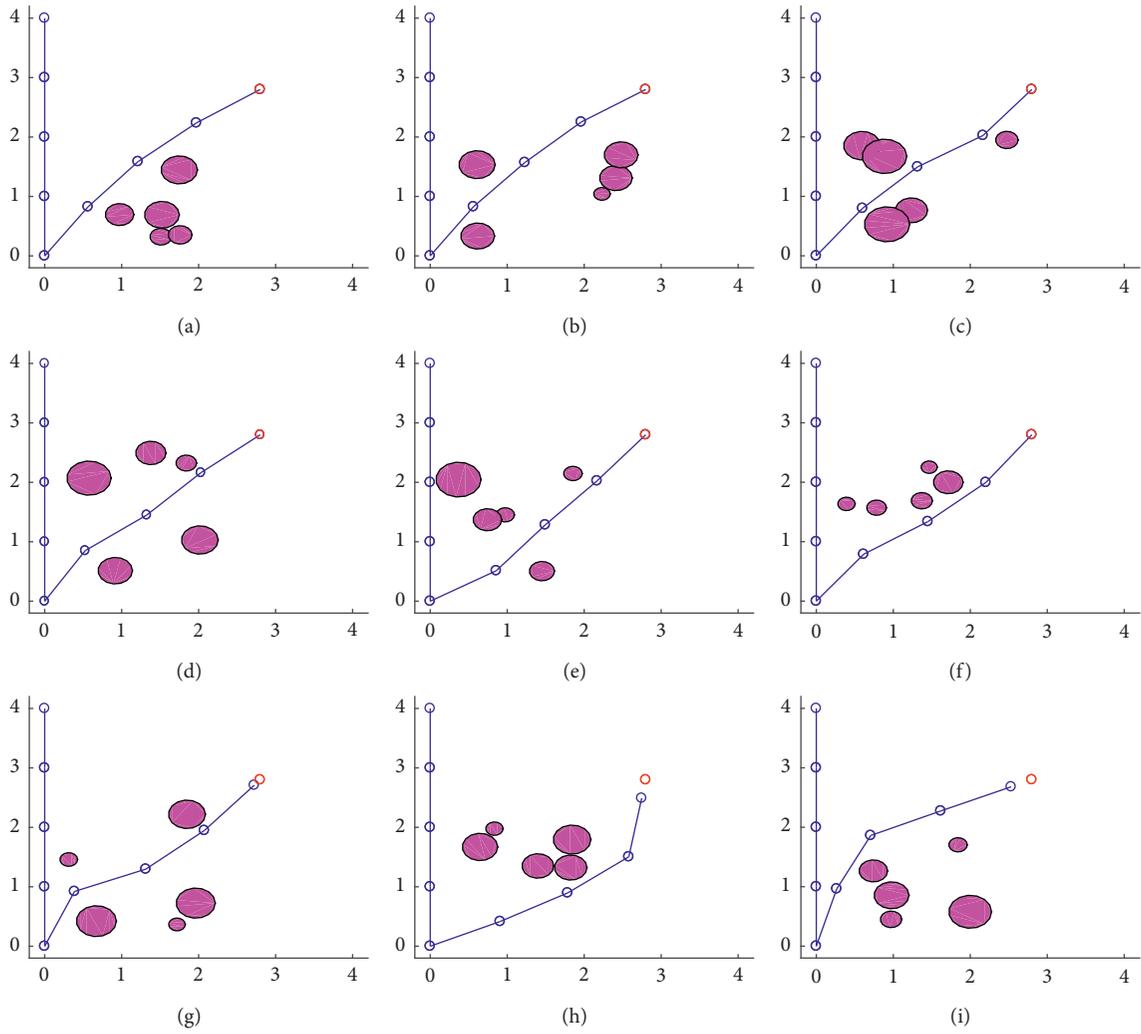


FIGURE 3: Visualization experiment results of extending FABRIK with obstacle avoidance in test 1.

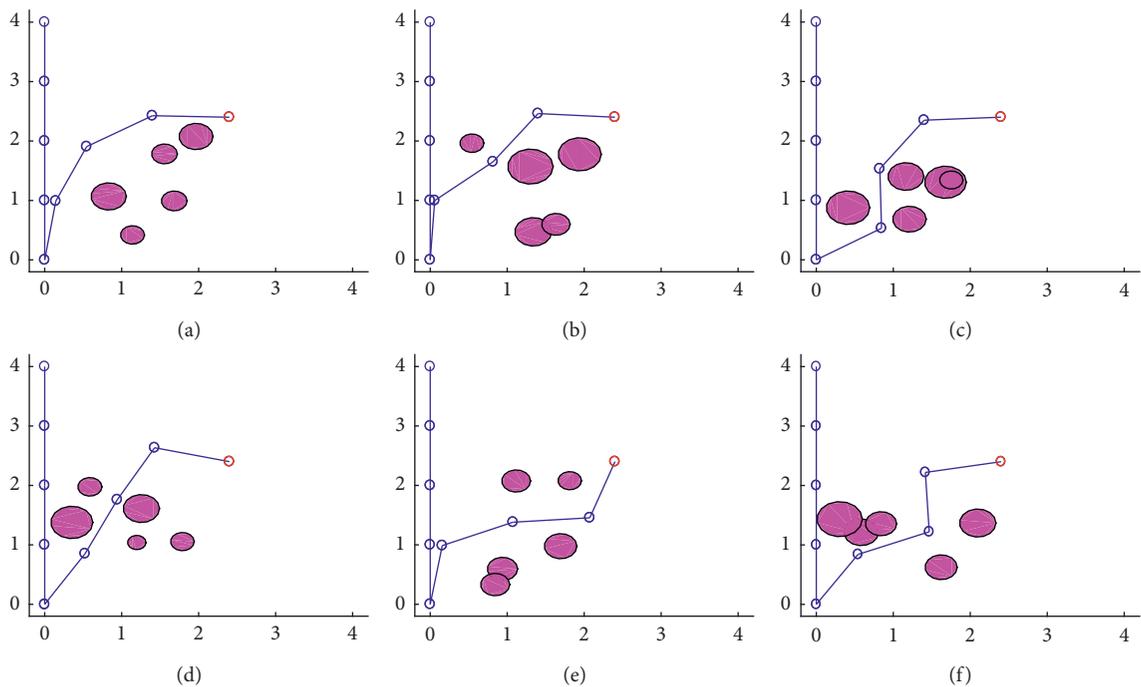


FIGURE 4: Continued.

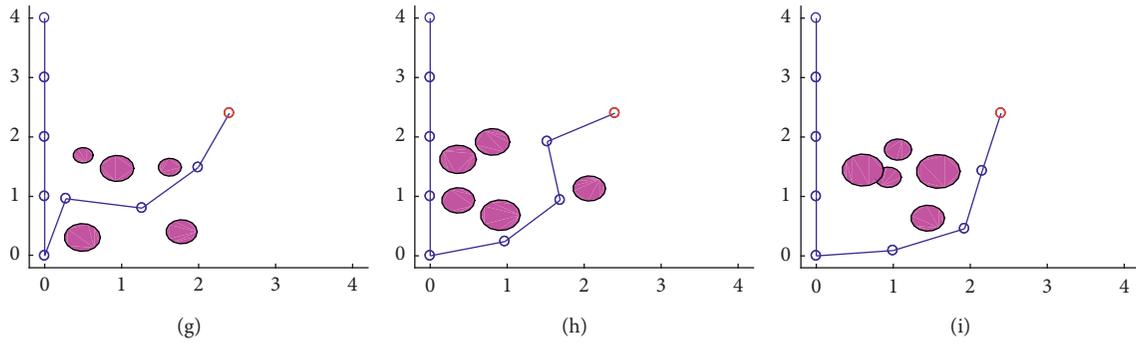


FIGURE 4: Visualization experiment results of extending FABRIK with obstacle avoidance in test 2.

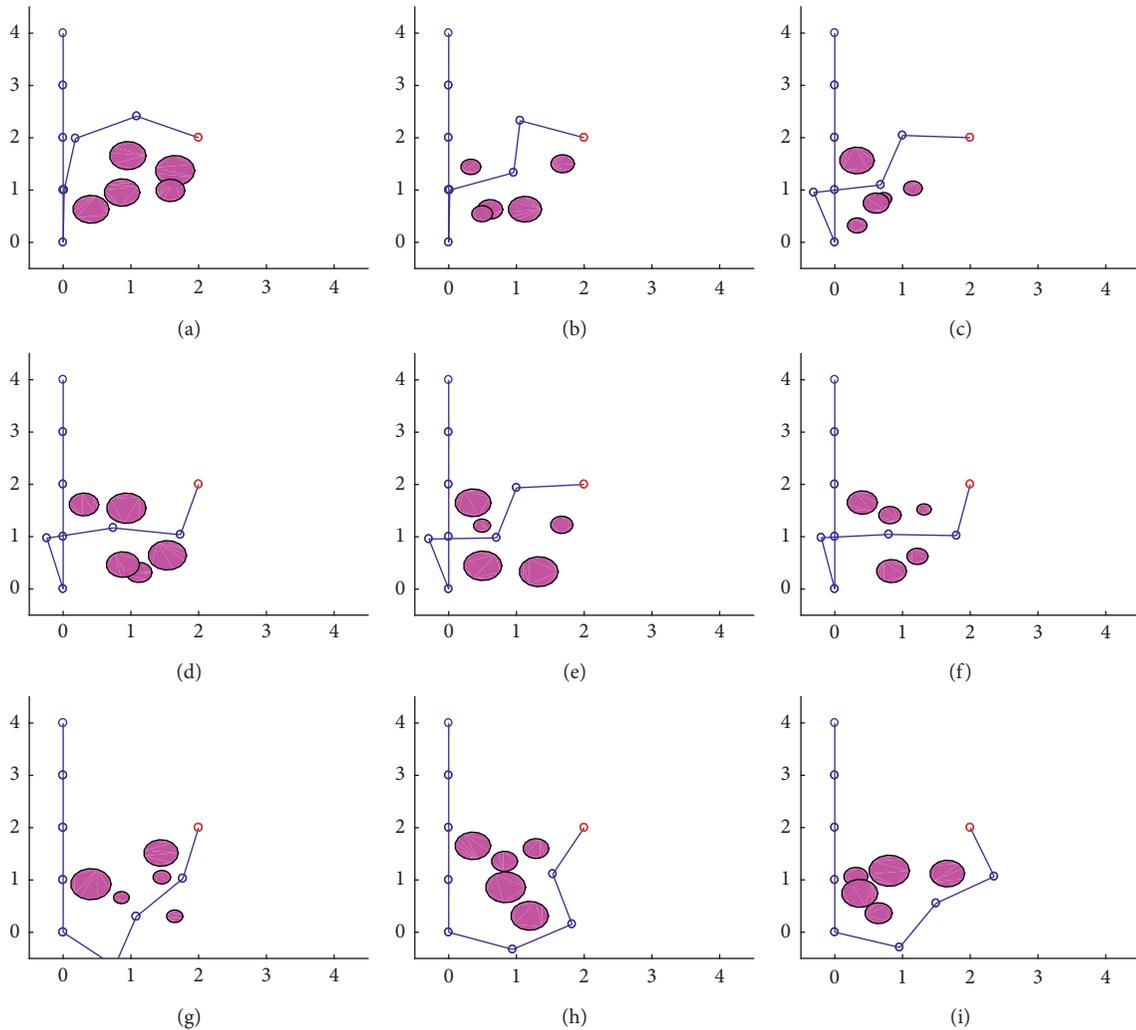


FIGURE 5: Visualization experiment results of extending FABRIK with obstacle avoidance in test 3.

times to make the end effectors reach the desired positions. The experimental results have shown that the extending FABRIK has a stronger adaptability for avoiding various obstacles with different sizes and positions. Even if the desired positions are unreachable in Figures 3(g)–3(i), the

extending FABRIK ensures that the end effector is close to the target point. Clearly, the experimental results of test 2 and test 3 are better than test 1. The reason is that the distance between the target position T and the initial position P_1 is closer in Figures 4 and 5.

6. Conclusions

Since the manipulator usually works in a complex environment, the joint chain may produce an interference with its surrounding scenarios. So, a two-step obstacle avoidance technique is proposed to extend the basic FABRIK in this study. In the first step, a heuristic method is adopted to ensure the updated linkage bar, the root joint, and the target position in a line, so that the interference can be eliminated in most cases. In the second step, multiple random rotation strategies are applied for eliminating the interference that has not been eliminated in the first step. Experimental results have shown that the proposed method has the obstacle avoidance ability. However, the extending FABRIK still confronts with the problem that the experimental results may vary for different tests. The reason is that a random rotation is adopted to avoid the obstacle (Step 3.5 in Section 4.3). The next step is to improve the stability and optimize the test results for the proposed method in the future.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This work was supported in part by the Natural Science Foundation of Hubei Province, China (2019CFB693).

References

- [1] A. Aristidou and J. Lasenby, "FABRIK: a fast, iterative solver for the Inverse Kinematics problem," *Graphical Models*, vol. 73, no. 5, pp. 243–260, 2011.
- [2] A. Aristidou, J. Lasenby, Y. Chrysanthou, and A. Shamir, "Inverse kinematics techniques in computer graphics: a survey," *Computer Graphics Forum*, vol. 37, no. 6, pp. 35–58, 2018.
- [3] J. Craig, *Introduction to Robotics: Mechanics and Control*, Addison-Wesley Longman Publishing Co., Reading, MA, USA, 3rd edition, 2003.
- [4] A. Aristidou and J. Lasenby, *Inverse Kinematics: A Review of Existing Techniques and Introduction of a New Fast Iterative Solver*, Cambridge University Engineering Department, Cambridge, UK, 2009.
- [5] M. Raghavan and B. Roth, "Inverse kinematics of the general 6R manipulator and related linkages," *Journal of Mechanical Design*, vol. 115, no. 3, pp. 502–508, 1993.
- [6] D. Manocha and J. F. Canny, "Efficient inverse kinematics for general 6R manipulators," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 5, pp. 648–657, 1994.
- [7] P. R. Diankov, "Automated construction of robotic manipulation programs," Ph.D. thesis, Carnegie Mellon University, Robotics Institute, Pittsburgh, PA, USA, 2010.
- [8] W. Liu, D. Chen, and J. Steil, "Analytical inverse kinematics solver for anthropomorphic 7-DOF redundant manipulators with human-like configuration constraints," *Journal of Intelligent & Robotic Systems*, vol. 86, no. 1, pp. 63–79, 2017.
- [9] J. Zhao, "Analytical inverse kinematics of anthropomorphic movements for 7-DOF humanoid manipulators," *Journal of Mechanical Engineering*, vol. 54, no. 21, p. 25, 2018.
- [10] X. Tian, Q. Xu, and Q. Zhan, "An analytical inverse kinematics solution with joint limits avoidance of 7-DOF anthropomorphic manipulators without offset," *Journal of the Franklin Institute*, vol. 358, no. 2, pp. 1252–1272, 2021.
- [11] Y. Tong, J. Liu, Y. Liu, and Y. Yuan, "Analytical inverse kinematic computation for 7-DOF redundant sliding manipulators," *Mechanism and Machine Theory*, vol. 155, Article ID 104006, 2021.
- [12] G. Schreiber and G. Hirzinger, "Singularity consistent inverse kinematics by enhancing the jacobian transpose," in *Advances in Robot Kinematics: Analysis and Control*, Springer, Berlin, Germany, 1998.
- [13] Z. Bhatti, A. Shah, F. Shahidi, and M. Karbisi, "Forward and inverse kinematics seamless matching using Jacobian," *Sindh University Research Journal (SURJ)*, vol. 45, no. 2, pp. 387–392, 2014.
- [14] Y. Nakamura and H. Hanafusa, "Inverse kinematic solutions with singularity robustness for robot manipulator control," *Journal of Dynamic Systems, Measurement, and Control*, vol. 108, no. 3, pp. 163–171, 1986.
- [15] L. Cui and J. S. Dai, "Reciprocity-based singular value decomposition for inverse kinematic analysis of the metamorphic multifingered hand," *Journal of Mechanisms & Robotics*, vol. 4, no. 3, pp. 34502–34506, 2012.
- [16] C. Wampler, "Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 16, no. 1, pp. 93–101, 1986.
- [17] L. M. Phuoc, P. Martinet, S. Lee, and H. Kim, "Damped least square based genetic algorithm with Gaussian distribution of damping factor for singularity-robust inverse kinematics," *Journal of Mechanical Science and Technology*, vol. 22, no. 7, pp. 1330–1338, 2008.
- [18] C. Welman, *Inverse Kinematics and Geometric Constraints for Articulated Figure Manipulation*, M.S. thesis, Simon Fraser University, Department of Computer Science, Burnaby, Canada, 1993.
- [19] M. Meredith and S. Maddock, "Adapting motion capture data using weighted real-time inverse kinematics," *Computers in Entertainment*, vol. 3, no. 1, pp. 1–15, 2005.
- [20] A. Andreas, C. Yiorgos, and L. Joan, "Extending FABRIK with model constraints," *Computer Animation & Virtual Worlds*, vol. 27, no. 1, pp. 35–57, 2016.
- [21] M. Libretto, Y. Qiu, E. Kim, K. Pluckter, N. S. Yuk, and J. Ueda, "Singularity-free solutions for inverse kinematics of degenerate mobile robots," *Mechanism and Machine Theory*, vol. 153, no. 6, Article ID 103988, 2020.
- [22] R. Fletcher, *Practical Methods of Optimization*, Wiley-Interscience, New York, NY, USA, 2nd edition, 1987.
- [23] V. Kumar, S. Sen, S. Sen, S. S. Roy, S. K. Das, and S. N. Shome, "Inverse kinematics of redundant manipulator using interval Newton method," *International Journal of Engineering and Manufacturing*, vol. 5, no. 2, pp. 19–29, 2015.
- [24] H. Lei, D. Yan, X. Zhang et al., "Inverse kinematics algorithm for 6-DOF joint robot with offset wrist based on modified newton iteration method," *Journal of Mechanical Transmission*, 2017.
- [25] I. Duleba and I. Karcz-Duleba, "Accelerating Newton algorithms of inverse kinematics for robot manipulators," in

- Proceedings of the 2018 23rd international conference on methods & models in automation & robotics (MMAR)*, pp. 50–372, Faculty of Electronics, Wrocław University of Science and Technology, Wrocław, Poland, August 2018.
- [26] K. Erleben and S. Andrews, “Solving inverse kinematics using exact Hessian matrices,” *Computers & Graphics*, vol. 78, pp. 1–11, 2019.
- [27] L.-C. T. Wang and C. C. Chen, “A combined optimization method for solving the inverse kinematics problems of mechanical manipulators,” *IEEE Transactions on Robotics and Automation*, vol. 7, no. 4, pp. 489–499, 1991.
- [28] A. L. Olsen and H. G. Petersen, “Inverse kinematics by numerical and analytical cyclic coordinate descent,” *Robotica*, vol. 29, no. 4, pp. 619–626, 2011.
- [29] A. N. Kamal and H. Jing, “Constrained cyclic coordinate descent for cryo-EM images at medium resolutions: beyond the protein loop closure problem,” *Robotica*, vol. 34, no. 8, pp. 1777–1790, 2016.
- [30] D. Bao, “Exploration of virtual human running based on CCD algorithm,” *Concurrency and Computation Practice and Experience*, vol. 31, no. 10, Article ID e4901, 2019.
- [31] R. Muller-Cajar and R. Mukundan, *Triangulation—A New Algorithm for Inverse Kinematics*, University of Canterbury Computer Science & Software Engineering, Christchurch, New Zealand, 2007.
- [32] R. Mukundan, “A robust inverse kinematics algorithm for animating a joint chain,” *International Journal of Computer Applications in Technology*, vol. 34, no. 4, pp. 303–308, 2009.
- [33] B. Kenwright, “Inverse kinematics—cyclic coordinate descent (CCD),” *Journal of Graphics Tools*, vol. 16, no. 4, pp. 177–217, 2012.
- [34] H. Jing and C. Pelachaud, “An efficient energy transfer inverse kinematics solution,” in *Motion in Games*, Springer, Berlin, Germany, 2012.
- [35] S. Agrawal and V. Michiel, “Task-based locomotion,” *ACM Transactions on Graphics*, vol. 35, no. 4, p. 82, 2016.
- [36] R. A. Tenneti and A. Sarkar, “Implementation of modified FABRIK for robot manipulators,” in *Proceedings of the Advances in Robotics 2019*, July 2019.
- [37] P. C. Santos, R. C. S. Freire, E. A. N. Carvalho, L. Molina, and E. O. Freire, “M-FABRIK: a new inverse kinematics approach to mobile manipulator robots based on FABRIK,” *IEEE Access*, vol. 8, pp. 208836–208849, 2020.