

Received August 7, 2018, accepted September 10, 2018, date of publication September 17, 2018, date of current version October 12, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2870332

Inverse Kinematics for 6-DOF Serial Manipulators With Offset or Reduced Wrists via a Hierarchical Iterative Algorithm

JING XU¹, KECHEN SONG¹, YU HE¹, ZHIPENG DONG¹, AND YUNHUI YAN¹

School of Mechanical Engineering and Automation, Northeastern University, Shenyang 110819, China

Corresponding author: Yunhui Yan (yanyh@mail.neu.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFB0304200, in part by the National Natural Science Foundation of China under Grants 51374063 and 51505069, and in part by the Fundamental Research Funds for the Central Universities under Grant N150308001 and Grant N170304014.

ABSTRACT It is still a challenge to accurately and efficiently solve inverse kinematics (IK) using six-degrees-of-freedom (DOF) serial manipulators with offset or reduced wrists, which is mainly due to the complex, highly nonlinear, and coupled IK equations. In order to solve this problem, an efficient IK algorithm called hierarchical iterative inverse kinematic algorithm (HIIKA) is proposed in this paper. This novel IK solution algorithm consists of two-level iterations. The first-level iteration can achieve accurate initial estimates through an extended heuristic iterative method that proposed in this paper. The second-level iteration calculates all joint angles based on the results of the first-level iteration. To evaluate the performance of the HIIKA, experiments were carried out with a six-DOF manipulator with a two-DOF reduced wrist. The results obtained demonstrate the advancement and excellence of the HIIKA in terms of its convergence, accuracy, computational efficiency, and multiple solutions. The HIIKA provides a new tool for the analysis and application of manipulators with offset or reduced wrists.

INDEX TERMS Hierarchical iterative algorithm, inverse kinematics, offset or reduced wrists, 6-DOF serial manipulators.

I. INTRODUCTION

6-Degrees-of-Freedom (DOF) serial manipulators with offset or reduced wrists, whose three consecutive joint axes of the end do not intersect at a common point, have a higher load capacity and flexibility than those with Euler wrists, whose three axes intersect at a common point [1]. IK plays a very important role in the structural design, motion planning, and dynamic analysis of manipulators. However, because IK equations are complex, highly nonlinear, and coupled, it is difficult for 6-DOF manipulators with offset or reduced wrists to accurately and efficiently solve Inverse Kinematics (IK) [2], [3]. In general, the main methods of solving IK are analytical methods and numerical methods [4]. The analytical methods cannot be applied to most of the complicated 6-DOF manipulators with offset or reduced wrists because of the strong coupling of IK transcendental equations. Although IK solutions can always be obtained with the numerical methods, it is hard to have a tradeoff between accuracy and computational efficiency due to the complex

and highly nonlinear IK equations. Multiple solutions and additional constraints may also increase the complexity of solving the IK equations.

So far, there have been many works for solving IK of manipulators with offset or reduced wrists. Some methods are based on the inverse Jacobian matrix, such as the Newton-Raphson, the predictor-corrector, and the damped least-squares (DLS) methods [5]–[8]. For these methods, the singular points at which the algorithm can fail to converge must be avoided. This is critical for accurate initial estimations in order to improve the convergence and computational efficiency of the methods. Some methods employ the gradient-based nonlinear method for solving an equivalent minimization problem [9]. These methods do not consider singular points, however their computational cost increases significantly due to the high nonlinearity IK equations. Multiple solutions and additional constraints will further reduce the convergence and computational efficiency of the numerical methods, making them more complex. Other methods

such as neural network methods [10]–[12], geometric methods [13]–[16], and reconfiguration methods [17], [18] have been developed for manipulators with offset or reduced wrists to solve IK. These methods are however so complex that establishing equations can be time-consuming or can cause computational inefficiency from a real-time control point of view.

Besides, there are alternative methods for manipulators with offset or reduced wrists to solve IK. Raghavan and Roth [19] simplified the problem so as to search the roots of a 16-degree polynomial concerning one joint angle. Pashkevich [20] proposed a real-time IK algorithm that did not use the Jacobian matrix. This reduced the IK equations into an efficient one-dimensional search problem. Kucuk and Bingul [21] developed a similar algorithm called a New Inverse Kinematics Algorithm (NIKA), which improved the iterative process of the algorithm [20]. However, they did not investigate the multiple solutions to the IK problem and did not discuss the End Effector (EE) orientation errors for NIKA. Although these algorithms reduced the complexity of the IK problem, computational efficiency was still not stable due to random initial estimations.

In order to ensure both computational efficiency and accuracy for IK of 6-DOF manipulators with offset or reduced wrists, a novel and simple Hierarchical Iterative Inverse Kinematic Algorithm (HIIKA) is presented. The algorithm consists of two-level iterations. The first-level iteration achieves accurate initial estimations by an extended heuristic iterative method with joint constraints we hereby propose in this paper. In the second-level iteration, all joint angles were calculated by an iterative method, based on an analytical calculation. The convergence, accuracy, and computational efficiency of HIIKA can be improved by giving accurate initial estimations, and reducing the dimension of search space. Multiple solutions may also be calculated quickly by the analytical calculation in the second-level iteration. Moreover, the singularity due to the Jacobian matrix does not need to be considered.

Our work makes the following contributions:

- 1) A novel and simple hierarchical iterative inverse kinematic algorithm are proposed for accurately and efficiently solving IK problem of 6-DOF manipulators with offset or reduced wrists.

- 2) To achieve accurate initial estimations, the extended heuristic iterative method with joint constraints which can be applied to gain IK initial estimations for general serial manipulators was also proposed.

- 3) A detailed experiments was carried out in Section V to evaluate the performance of HIIKA. The results show that HIIKA offers improved performance on the accuracy, computational efficiency, multiple solutions, and convergence compared to other algorithm.

The paper is organized in the following manner. HIIKA framework is explained in Section II, while forward kinematics as an example of 6-DOF manipulator with a 2-DOF reduced wrist is investigated in Section III.

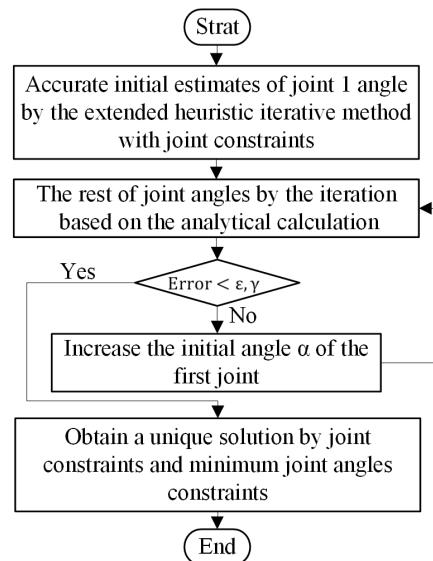


FIGURE 1. Algorithm framework of HIIKA.

Section IV explains the two-level iterations of HIIKA and in Section V, the results of the experiments are discussed in detail by making comparisons between NIKA and HIIKA. Section VI gives a conclusion of the entire work.

II. ALGORITHM FRAMEWORK

HIIKA involves two-level iterations to achieve accurate initial estimations and calculate all joint angles. The algorithm framework is depicted in Fig. 1. In the first-level iteration, two accurate initial estimations of joint 1 angle are gained by the extended heuristic iterative method. In the second-level iteration, the IK problem is reduced to 1-DOF search problem. The rest of the joint angles are calculated by the accurate initial estimations of joint 1 angle and the iteration, based on analytical calculation. Using forward kinematics, the allowed EE position error ϵ , and the allowed EE orientation error γ was used to verify the correctness of the IK solutions. If IK solutions are not correct, the solutions of joint 1 angle are searched near the initial estimations by increasing angle α , then the rest of the joint angles are calculated. The second-level iteration is repeated until the errors of IK solutions satisfied ϵ and γ . Finally, joint constraints and minimum joint angle constraints are introduced to select a set of IK solutions from multiple solutions. Thus, HIIKA can plan a continuous and smooth trajectory path. As an example, the kinematics of the 6-DOF manipulator with a 2-DOF reduced wrist is studied in this paper.

III. FORWARD KINEMATICS

Fig. 2 shows the structure and coordinate frames for the manipulator with a 2-DOF reduced wrist. The link parameters follow the Denavit-Hartenberg (D-H) convention, and D-H parameters of the manipulator are illustrated in TABLE I. a_{i-1} and α_{i-1} are the translation and rotation about x_{i-1} -axis,

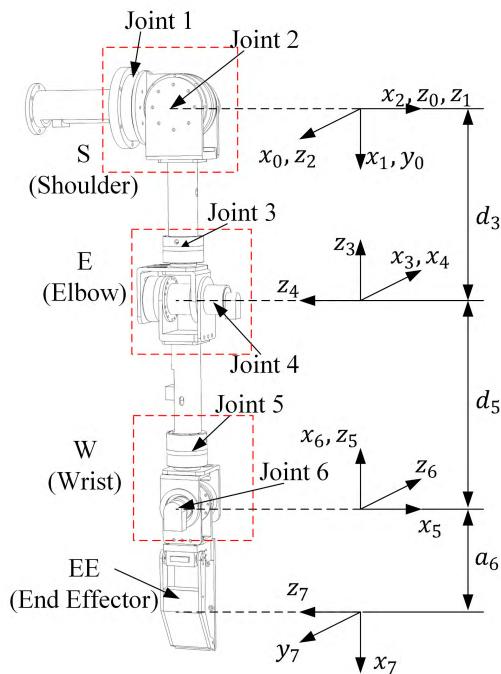


FIGURE 2. The configuration of the 6-DOF manipulator with a 2-DOF reduced wrist.

TABLE 1. D-H Parameters of the manipulator with a 2-dof reduced wrist.

i	$a_{i-1}(m)$	$\alpha_{i-1}(\text{deg})$	$d_i(m)$	$\theta_i(\text{deg})$
1	0	0	0	θ_1
2	0	90	0	θ_2
3	0	-90	d_3	θ_3
4	0	-90	0	θ_4
5	0	90	d_5	θ_5
6	0	-90	0	θ_6
7	a_6	90	0	180

respectively. d_i and θ_i are the translation and rotation about z_i -axis, respectively.

A homogenous transformation matrix ${}^{i-1}T$ for each joint gives the position and orientation of i th coordinate frame about $(i-1)$ th coordinate frame.

$${}^{i-1}T = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -d_i s\alpha_{i-1} \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & d_i c\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

where $c\theta_i$ denotes $\cos\theta_i$, $s\theta_i$ denotes $\sin\theta_i$. Transformation matrices for the manipulator with a reduced wrist can be obtained by applying D-H parameters in TABLE 1 to (1).

The forward kinematics from joint 1 to joint 6 frame is calculated by multiplying ${}^{i-1}T(i = 1, 2, \dots, 6)$ matrices.

6T represents the fixed position and orientation from joint 6 coordinate frame to EE coordinate frame.

$${}^6T = {}^0T_1 T_2^1 T_3^2 T_4^3 T_5^4 T_6^5 T = \begin{bmatrix} R_{3 \times 3} & P_{3 \times 1} \\ 0 & 1 \end{bmatrix} \quad (2)$$

where $R_{3 \times 3}$ represents the rotation matrix and $P_{3 \times 1}$ is the position vector of the wrist. The terms $R_{3 \times 3}$, $P_{3 \times 1}$ and ${}^{i-1}T(i = 1, 2, \dots, 7)$ are given in APPENDIX A.

IV. INVERSE KINEMATICS

A. THE EXTENDED HEURISTIC ITERATIVE METHOD

The iterative method which is based on analytical calculation, gives IK solutions of the manipulator by operating initial estimations of joint 1 angle. Because accurate initial estimations can seriously reduce the computational efficiency, the initial estimations cannot be any random values. Thus, it is essential to get accurate initial estimations in order to reduce the number of iterations. For this purpose, we present an extended heuristic iterative method with joint constraints based on [22] achieving accurate initial estimations. The extended heuristic iterative method includes the heuristic configuration estimation for the manipulator and initial estimations for joint 1 angle.

1) ESTIMATING THE CONFIGURATION FOR THE MANIPULATOR

In our case, the heuristic configuration estimation is used for the manipulator from the shoulder to the EE. Joint 1 is treated as the base joint. The given position and orientation are respectively treated as the desired EE position and orientation. The initial configuration of the manipulator is shown in Fig. 2 and Fig. 3 (a). The procedure for the heuristic configuration estimation is as follows:

1. The EE position and orientation are made to coincide with the target. Position W' is determined at a distance a_6 from EE' along the direction vector of the EE' orientation. The shaded ellipsoidal shape shown in Fig. 3(b) is the allowed range of motion defined in Fig. 4. β_i , ($i = 1, \dots, 4$), is the allowed range of joint angle. Determine the distance L from the joint position W' to the point O and calculate the distances $q_i = L \tan(\beta_i)$, for $i = 1, \dots, 4$, which can define the shaded ellipsoidal shape.

2. Position E^c is the nearest point to the joint position E on the cone surface to ensure that the new joint position E' satisfies joint constraints. Position E' is determined at a distance d_5 from the position W' along the line E^c-W' shown in Fig. 3 (C) and Fig. 3 (D).

3. Steps 1 and 2 are repeated for all the remaining joints to achieve the process of forward reaching with joint constraints for the manipulator, as shown in Fig. 3 (e).

4. Position S' is moved to position S'' overlapping with the base point. position E'' is determined at a distance d_3 from position S'' along the line $S''-E'$, as shown in Fig. 3 (f).

5. Step 4 is repeated for the rest of the joints to achieve the process of backward reaching without joint constraints for the manipulator, as shown in Fig. 3 (g).

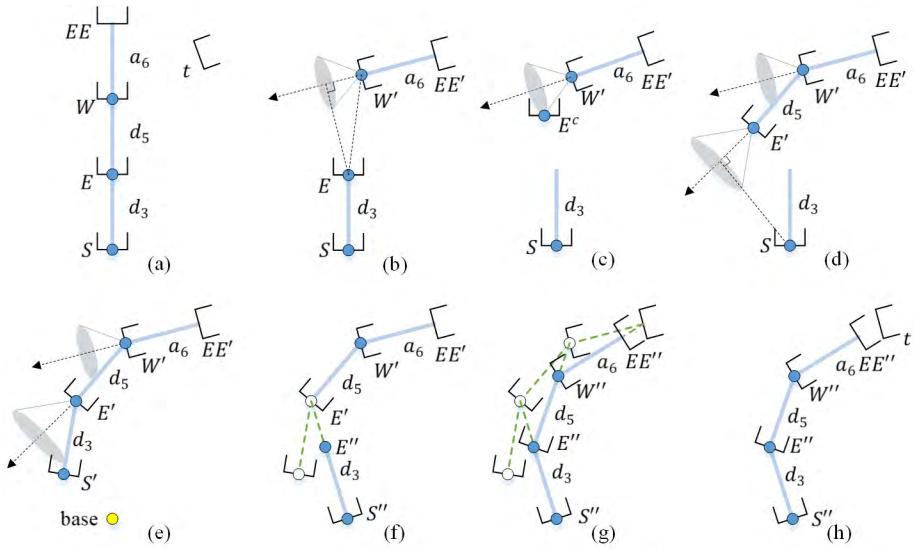


FIGURE 3. One full iteration of the heuristic configuration estimation. (a), (b), (c), (d), (e) – forward reaching with joint constraints. (f), (g), (h) – backward reaching without joint constraints.

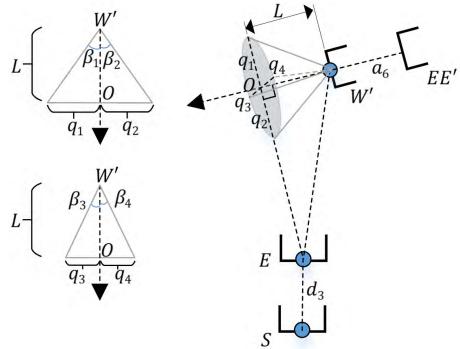


FIGURE 4. The definition of the allowed range of motion.

6. The heuristic configuration estimation is executed once from steps 1 to 5 shown in Fig. 3 (h). The procedure is repeated until the difference between the position of the EE and the target is within the allowed position error δ or the number of iterations reaches the maximum $\tau \cdot \delta$ and τ are parameters for the extended heuristic iterative method.

In Fig. 3, the shoulder, elbow, wrist, and end effector are expressed as S, E, W, and EE, respectively. a_6 , d_3 , and d_5 represent the length of links. And the t represents the position and orientation of the target. It is worth noting that joint constraints are added in the process of forward reaching, but joint constraints are not added during the process of back reaching. Because the orientation of the base joint is unknown, joint constraints cannot be added. Therefore, each process of forward reaching with joint constraints eliminates the error that is produced due to the last process of back reaching without joint constraints. The last iteration will produce a slight error because of the last process of back reaching without joint constraints.

2) OBTAINING INITIAL ESTIMATIONS

The extended heuristic iterative method with joint constraints can calculate the final manipulator configuration close to the initial configuration, and a final manipulator configuration gives one accurate initial estimations of joint 1 angle. Hence, two different initial configurations of the manipulator are set in Fig. 5 (a) and (c). To get two initial estimations of joint 1 angle, initial configurations specify joint 1 angle as the minimum and maximum values, respectively. The final configurations of the manipulator, which are obtained by the heuristic configuration estimation in Fig. 5 (b) and (d), give two initial estimations of joint 1 angle.

In Fig. 6, the configuration S-E-W-EE of the manipulator is obtained via the heuristic configuration estimation. Plane Φ is perpendicular to the axis of joint 1. The vector v_0 is a vertical downward vector in the plane Φ . The vector Φ . The vector v_1 is a vector which is point to E from S. and the vector v_2 is the projection vector of the vector v_1 in the plane Φ . The initial estimation of joint 1 angle θ_1 can be calculated by the vector v_0 and v_2 .

B. SOLVING FOR ALL DOF

A set of IK solutions $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$ of the manipulator with a reduced wrist are calculated by the iterative method based on the analytical calculation, using the forward kinematics and stopping criteria to verify the IK solutions. If the IK solutions are incorrect, the solution of joint 1 angle is searched within $\pm\sigma^\circ$ until they are correctly found. Therefore, multiple solutions for the IK problem can be obtained by HIIKA. The algorithm is shown in algorithm 1.

In algorithm 1, α is the incremental angle for the initial estimation of joint 1 angle. ε_c and γ_c are errors of position and orientation between the EE and the target, respectively. ε and γ are the allowed EE position error and the allowed EE orientation error, respectively.

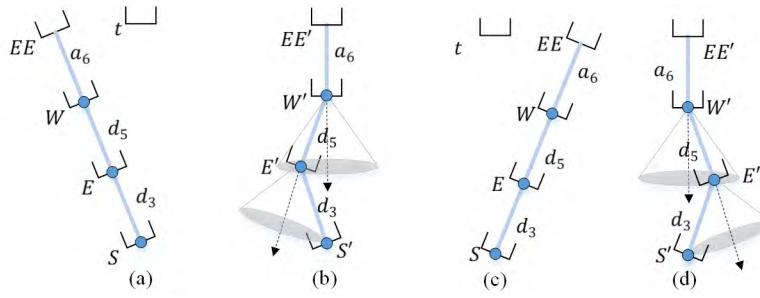


FIGURE 5. The different configurations of the manipulator. (a), (c) – the initial configurations that specify the joint 1 angle as the minimum and maximum values respectively. (b), (d) – the final configurations are obtained based on (a) and (b) by heuristic configuration estimation.

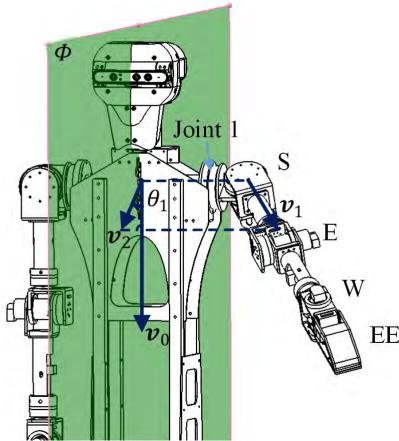


FIGURE 6. The initial estimation of joint 1 angle θ_1 .

The analytical calculation for IK may gain joint angles $\theta_2, \theta_3, \theta_4, \theta_5, \theta_6$ in the case of the given homogenous transformation matrix T_{EE} and joint 1 angle θ_1 . In order to get θ_2 , (3) is derived from (2) and triangulation substitution.

$$\theta_2 = \phi + \text{Atan}2(k/\rho, \pm\sqrt{1 - (k/\rho)^2}) \quad (3)$$

where ρ, ϕ and k are given in APPENDIX B. θ_3 and θ_4 can be calculated by (4) and (5) derived from (2), (8).

$$\theta_3 = \text{Atan}2(p_x s \theta_1 - p_y c \theta_1, -p_z s \theta_2 - p_x c \theta_1 c \theta_2 - p_y c \theta_2 s \theta_1) \quad (4)$$

$$\theta_4 = \text{Atan}2(\pm\sqrt{1 - c^2 \theta_4}, \chi) \quad (5)$$

where χ is given in APPENDIX B. Giving θ_5 and θ_6 , (6) and (7) are derived from the calculated $\theta_2, \theta_3, \theta_4$ and (2), respectively.

$$\theta_5 = \text{Atan}2(-\zeta, \eta) \quad (6)$$

$$\theta_6 = \text{Atan}2(\lambda, \tau) \quad (7)$$

where ζ, η, λ and τ are given in APPENDIX B.

Two solutions exist for joint 1, joint 2, and joint 4, respectively. There is one solution for each of the remaining

joints. Thus, HIIKA can provide up to eight sets of possible IK solutions for the manipulator with a reduced wrist.

Not all IK solutions of the manipulator are within the limits of the joint angles. Feasible IK solutions are filtered out by joint constraints. Only one set of reasonable IK solution is used for trajectory planning. Therefore, the continuous and smooth manipulator trajectory can be planned by minimum joint angle constraints.

C. SINGULARITY

In fact, (8) derived from (2), the singularity could also be analyzed by (8).

$$\begin{cases} p_z s \theta_2 + p_x c \theta_1 c \theta_2 + \\ p_y c \theta_2 s \theta_1 = -d_5 c \theta_3 s \theta_4 \\ p_x s \theta_1 - p_y c \theta_1 = d_5 s \theta_3 s \theta_4 \end{cases} \quad (8)$$

Two kinds of cases should be concerned in (8). If $s \theta_4 \neq 0$, (4) can be got by (8). If $s \theta_4 = 0$, θ_4 is 0° or 180° , the manipulator is in singular configuration that the axis of joint 3 coincides with the axis of joint 5 and (4) cannot be derived. In this case, the EE of manipulator can only move along the tangent, therefore the DOF of manipulator will be reduced in the workspace. $\theta_4 = 180^\circ$ is out of the range of joint 4, thence the manipulator is in singular configuration when θ_4 is 0° . In singular configuration, set an arbitrary angle θ_3 within the range of joint 3, then calculate θ_5 based on it.

V. SIMULATIONS AND DISCUSSION

HIIKA is verified and the application of HIIKA for motion planning is shown in this section. The convergence, accuracy, efficiency, and multiple solutions of the algorithm are explained by the experiments.

To evaluate the performance of HIIKA for the manipulator with a 2-D reduced wrist, the geometric model of a robot was constructed, which will not influence the analysis of the results. The manipulator at the initial time is illustrated in Fig. 7. The red, green, and blue lines respectively indicate x, y, and z-axes of the coordinate frame. TABLE 2 shows the simulation parameters in Fig. 7.

Algorithm 1: The Proposed Algorithm for IK

Input Homogenous transformation matrix T_{EE} of EE
Output The IK solutions $(\theta_1^i, \theta_2^i, \theta_3^i, \theta_4^i, \theta_5^i, \theta_6^i)$,
 $i = 1, 2 \dots 8$

- 1 % The two initial estimations of the joint 1
- 2 % θ_1, θ'_1 calculated by the extended heuristic
- 3 % iterative method
- 4 $\psi = [\theta_1, \theta'_1]$
- 5 % Solving for all DOF
- 6 **for** $j = 1:2$
- 7 $\theta_1^c = \psi(j)$
- 8 dataset_up = $\theta_1^c : \alpha : \theta_1^c + \sigma^\circ$
- 9 dataset_down = $\theta_1^c : -\alpha : \theta_1^c - \sigma^\circ$
- 10 n = size(dataset_up)
- 11 **for** $k = 1:n$
- 12 $\theta_1^s = \text{dataset_up}(k)$
- 13 % IK solutions by the iterative method
- 14 % based on analytical calculation
- 15 Res(j) $\leftarrow (\theta_1^i, \theta_2^i, \theta_3^i, \theta_4^i, \theta_5^i, \theta_6^i)$,
- 16 *i* = 1, 2 ... 4
- 17 $\varepsilon_c, \gamma_c \leftarrow$ Errors of EE position and orientation
- 18 **if** $\varepsilon_c \leq \varepsilon$ & $\gamma_c \leq \gamma == 1$
- 19 **Return** Res(j)
- 20 **break**
- 21 **end**
- 22 **end**
- 23 **end**
- 24 **if** Res(j) is not empty
- 25 **for** $h = 1:n$
- 26 $\theta_1^s = \text{dataset_down}(h)$
- 27 Res(j) $\leftarrow (\theta_1^i, \theta_2^i, \theta_3^i, \theta_4^i, \theta_5^i, \theta_6^i)$,
- 28 *i* = 1, 2 ... 4
- 29 $\varepsilon_c, \gamma_c \leftarrow$ Errors of EE position and orientation
- 30 **if** $\varepsilon_c \leq \varepsilon$ & $\gamma_c \leq \gamma == 1$
- 31 **Return** Res(j)
- 32 **break**
- 33 **end**
- 34 **end**
- 35 **end**
- 36 **end**

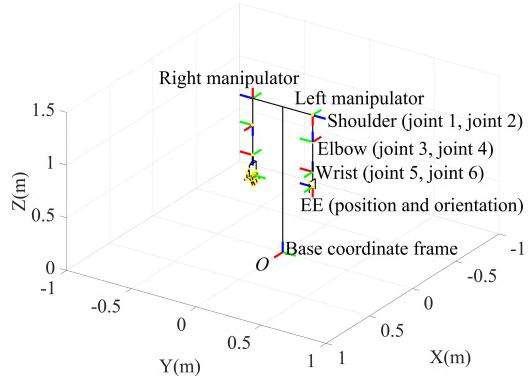


FIGURE 7. The manipulator with a reduced wrist at the initial time. The red, green, and blue lines indicate x, y, and z-axis of the coordinate frame, respectively.

TABLE 2. Simulation parameters of the manipulator

Parameters	Values
Each link from shoulder	0.2551m/ 0.2828m/ 0.1415m
Initial angle from joint 1 to joint 6	90°/90°/90°/0°/-90°/-90°
Each joint limits from joint 1 to 6	0°-135°/90°-180°/0°-180°/-45°-90°-180°-90°/-180°-0°

TABLE 3. EE poses in terms of the base coordinate frame.

EEP	EE Position		EE Orientation	
	<i>p</i>	<i>n</i>	<i>o</i>	<i>a</i>
Initial	0	0	1	0
	0.2310	0	0	-1
	0.7006	-1	0	0
Pose-1	0.5518	1	0	0
	0.2310	0	0.5	0.8660
	1.1591	0	0.8660	0.5
Pose-2	0.2593	0.4571	-0.833	-0.3108
	0.5889	0.3295	0.4833	-0.8111
	1.1291	0.8262	0.2683	0.4954

terms of the base coordinate frame are given in TABLE 3 where EEP denotes end-effector poses. In TABLE 4, the first two items provide parameters for the extended heuristic iterative method and the last three items provide parameters for both HIIKA and NIKA. The EE position errors, EE orientation errors, and IK are illustrated in TABLE 5.

In TABLE 5, for pose-1, 6 and 4 sets of IK solutions can be obtained by using HIIKA and NIKA, respectively. And for pose-2, HIIKA and NIKA give 4 and 2 sets of IK solutions, respectively. Because of the analytical calculation, both two algorithms give low EE position errors close to zero. EE orientation errors were between 0.5° and 1° for both HIIKA and NIKA, due to the allowed EE orientation error set for the

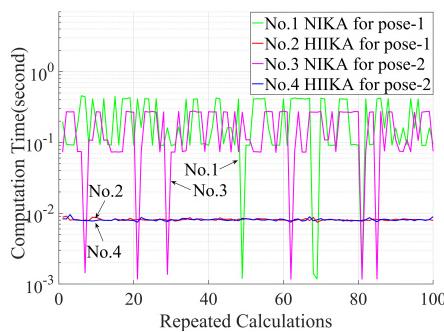
The comparative experiments are carried out by using NIKA to solve IK. NIKA is a numerical method based on an analytic method. Since the initial value is randomly selected, NIKA may not be all the time efficient. Both HIIKA and NIKA were performed on a CPU of Intel Core i7-8700k 3.70GHz with 16GB RAM using MATLAB 2016b software program.

A. ACCURACY, EFFICIENCY AND MULTIPLE SOLUTIONS

In order to show the accuracy, efficiency and multiple solutions for HIIKA, experiments of IK solutions for the manipulator with a reduced wrist were executed. Two EE poses in

TABLE 4. SIMULATION parameters of HIIKA AND NIKA.

Parameters	Values
The allowed position error for the extended heuristic iterative method	$\delta = 1 \times 10^{-7}$ m
Max iterations for the extended heuristic iterative method	$\tau = 5$
The search range of joint 1 angle	$\sigma = \pm 5^\circ$
The allowed EE position error	$\varepsilon = 1 \times 10^{-5}$ m
The allowed EE orientation error	$\gamma = 1^\circ$
Incremental angle	$\alpha = 0.001^\circ$

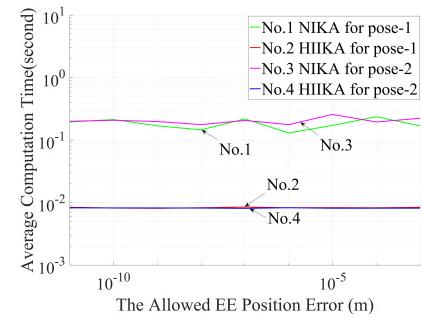
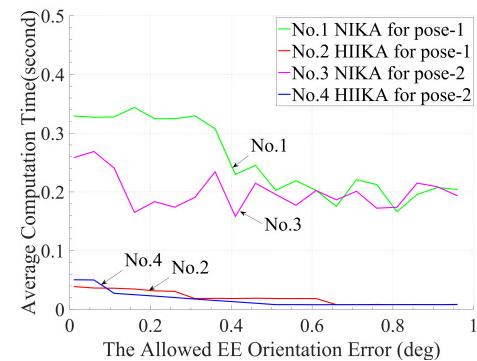
**FIGURE 8.** Computation time for 100 repeated calculations, using HIIKA and NIKA for pose-1 and pose-2 respectively.

stopping criteria. For pose-1, average EE orientation errors are 0.7119° and 0.9570° for HIIKA and NIKA, respectively. For pose-2, average EE orientation errors are 0.7430° and 0.9923° for HIIKA and NIKA, respectively. HIIKA was slightly better than NIKA for EE orientation error on the same allowed EE orientation error.

To verify the computational efficiency, IK solutions for pose-1 and pose-2 for 100 times were calculated using two algorithms. The time of each computation is illustrated in Fig. 8. HIIKA took much less computational time than NIKA. The computational time curves obtained by HIIKA were stable. However, because of the randomness of the given initial value, there was a large fluctuation for the computational time curves obtained by NIKA.

For joint 1 angle, TABLE 6 shows the accurate initial estimates and the errors between the estimates and the solutions. The percentages of errors between the initial estimates and the solutions were 0.0478%, 0.0652%, 0.0470%, and 1.824%, which indicate that the extended heuristic iterative method in the first-level iteration can achieve accurate initial estimates for joint 1 angle.

The statistics for the computation time after 100 times of repeated calculations are shown in TABLE 7. According to the average computation time, computational efficiency of HIIKA was 95.03% higher than NIKA for pose-1 and 95.43% higher than NIKA for pose-2. The average computation time for HIIKA was about 8ms, which satisfies real-time control

**FIGURE 9.** The influence of the different allowed EE position error on the computation time.**FIGURE 10.** The influence of the different allowed EE orientation error on the computation time.

for the manipulator. HIIKA had higher computational efficiency and faster convergence than NIKA due to accurate initial estimations.

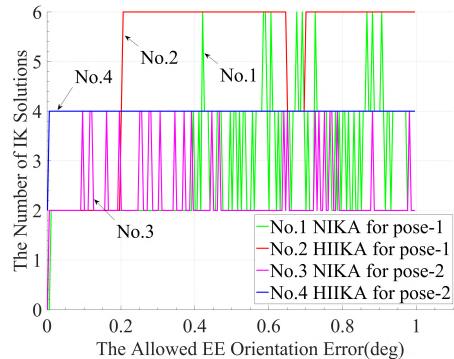
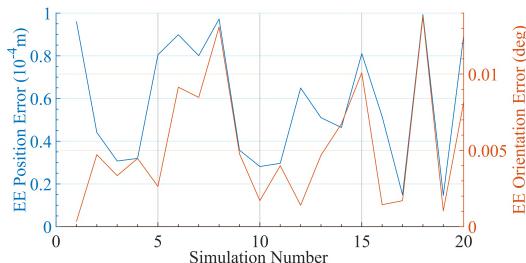
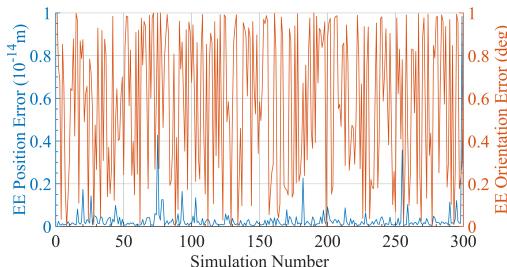
Different allowed EE position error and orientation error have different effect on the computation time of each algorithm (Fig.9 and 10). The average computation time was obtained after 20 repeated calculations for each algorithm at each allowed EE position error and orientation error. Changes in the average computation time curves were not sensitive to the allowed EE position error for both two algorithms. due to low position error for each algorithm. The average computation time of NIKA changes significantly with different allowed EE orientation error. Thus, regardless of the change of the allowed EE position error or orientation error, HIIKA has a much higher computational efficiency and stronger convergence than NIKA.

Fig. 11 shows IK multiple solutions for two algorithms. The allowed EE orientation error is from 0.001° to 1° . In most cases, 4 or 6 sets of IK solutions were obtained by HIIKA. HIIKA gave only 2 sets of IK solutions when the allowed EE orientation error is 0.001° , while NIKA gave 2 or 4 sets of IK solutions, sometimes even 6 sets of IK solutions can be obtained with NIKA. But IK solutions could not be solved when the allowed EE orientation error was 0.001° . Obviously, HIIKA is shown to have better multiple solutions than NIKA.

To further show the ability of HIIKA for solving IK, the proposed algorithm solved IK for 300 groups of random

TABLE 5. The position errors, orientation errors, and IK solutions

EEP	Algorithm	Position error (m)	Orientation error (deg)	IK solution					
				θ_1 (rad)	θ_2 (rad)	θ_3 (rad)	θ_4 (rad)	θ_5 (rad)	θ_6 (rad)
Pose-1	HIIKA	9.467×10^{-17}	0.6470	1.0466	1.5448	1.6129	1.0472	-1.0909	-1.5502
		9.231×10^{-17}	0.6470	1.0466	1.5448	4.7545	-1.0472	2.0507	-1.5502
		8.654×10^{-17}	0.6978	1.0466	1.5968	1.5286	1.0472	-1.0037	-1.5909
		8.846×10^{-17}	0.6978	1.0466	1.5968	4.6702	-1.0472	2.1379	-1.5909
		3.184×10^{-17}	0.7910	0.7672	1.0835	2.6021	1.0472	-2.0404	-1.0704
	NIKA	4.859×10^{-17}	0.7910	0.7672	1.0835	5.7436	-1.0472	1.1012	-1.0704
		8.443×10^{-17}	0.9764	0.7750	1.0877	2.5854	1.0472	-2.0268	-1.0768
		6.871×10^{-17}	0.9764	0.7750	1.0877	5.7270	-1.0472	1.1148	-1.0768
		6.676×10^{-17}	0.9375	1.0460	1.5324	1.6331	1.0472	-1.1119	-1.5402
		5.003×10^{-17}	0.9375	1.0460	1.5324	4.7747	-1.0472	2.0297	-1.5402
Pose-2	HIIKA	4.449×10^{-16}	0.5075	1.0637	1.9380	-3.0814	0.5236	-0.9810	0.0830
		4.236×10^{-16}	0.5075	1.0637	1.9380	0.0602	-0.5236	2.1606	0.0830
		2.775×10^{-17}	0.9784	0.7674	2.1250	-1.9823	0.5236	-2.2164	0.3749
		2.775×10^{-17}	0.9784	0.7674	2.1250	1.1593	-0.5236	0.9251	0.3749
	NIKA	8.236×10^{-17}	0.9923	1.0810	1.9374	-3.1324	0.5236	-0.9236	0.0711
		8.236×10^{-17}	0.9923	1.0810	1.9374	0.0092	-0.5236	2.2180	0.0711

**FIGURE 11.** The influence of the different allowed EE orientation error on IK multiple solutions.**FIGURE 13.** The EE position and orientation error for 20 groups of random singular poses.**FIGURE 12.** The EE position and orientation error for 300 groups of random poses.

poses without singular poses and 20 groups of random singular poses in the usable workspace shown in TABLE 2. The parameters shown in TABLE 4 were used in the experiment, but the allowed EE position error ε changed as 1×10^{-4} m,

not 1×10^{-5} m, since random singular poses did not always converge when $\varepsilon = 1 \times 10^{-5}$ m.

TABLE 8 and Fig.12 show the experimental results for 300 groups of random poses. The average position error, average orientation error and average computation time are 2.6615×10^{-16} m, 0.5006° , and 18.9 ms, respectively, even the max values are good enough for real-time control, which illustrate the accuracy and efficiency in the usable workspace without singular poses.

TABLE 9 and Fig.13 show the experimental results for 20 groups of random singular poses. It is notable that the position error and computation time are influenced by the

TABLE 7. The statistics for the computation time in 100 repeated calculations.

EEP Algorithm	Pose-1		Pose-2	
	HIIKA	NIKA	HIIKA	NIKA
Average computation time(ms)	8.8	17.72	8.5	186
Max computation time(ms)	28.9	46.24	14.9	339
Min computation time(ms)	7.6	1.2	7.6	1.2

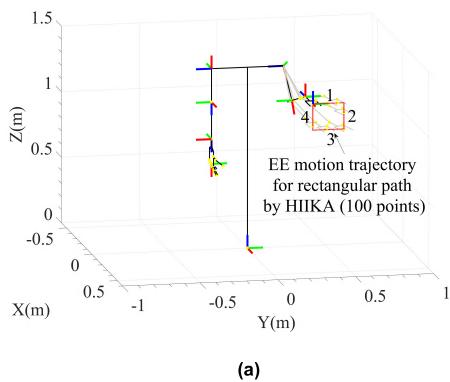


FIGURE 14. Actual tracking of the rectangular path. (a) – The rectangular path by HIIKA. (b) – The rectangular path by NIKA.

singularity. However, the average position error and average computation time are less than 0.1mm and 0.065s, respectively, which illustrate that the proposed algorithm is both accurate and efficient even for singular poses.

B. MOTION PLANNING

This section discusses the performance of HIIKA with regards to motion planning. The trajectory simulated and implemented is a rectangle on a plane using a total of 100 path points. Each side of the rectangle was 0.2m. The z-axis of the EE coordinate frame was always consistent with the direction vector (0, 0, 1) during the simulation. The motion planning was carried out by using HIIKA and NIKA, respectively. The simulation parameters of the two algorithms are shown in TABLE 9.

Fig.14 (a) shows the actual tracking of the rectangular path. By using HIIKA, red points are the actual EE position.

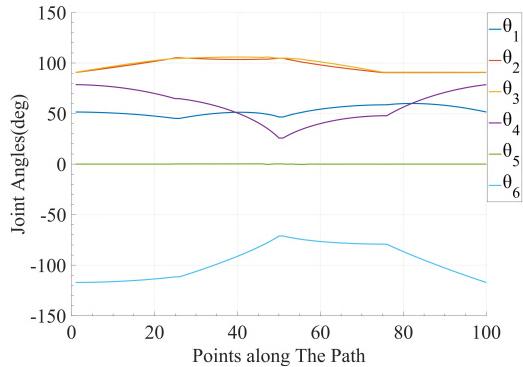


FIGURE 15. Joint trajectories for the rectangular path.

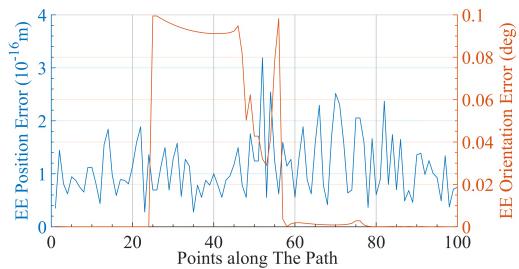


FIGURE 16. The EE position and orientation error for the rectangular path.

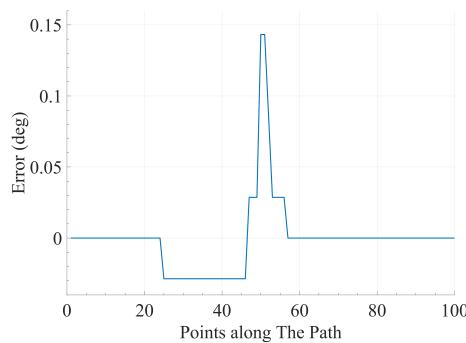
The motion planning sequence is marked by 1, 2, 3, and 4. The 100 path points were all solved by HIIKA. The total motion planning time was 1.7687s, and the average computational time for each path point was 17.7ms. The maximum single computation time was 25.4ms, while the minimum was 13.1ms. Obviously, HIIKA can accomplish real-time computation with low allowed EE position error and orientation error. In Fig.14 (b), the motion planning could not be completed by NIKA using simulation parameters ε , γ , α in TABLE 8, because of the randomness of the initial value and the larger incremental angle α . HIIKA can therefore be efficient with stable convergent.

Fig.15 shows all the joint trajectories for the rectangular paths. The minimum joint angle constraints make the joint trajectories continuous and smooth. EE position errors were between 10^{-16} and 10^{-15} , and EE orientation errors did not exceed 0.1° (Figs.16). The reason for the EE position and orientation errors was that the backward reaching of the extended heuristic iterative method did not add joint constraints in the last iteration. In Fig. 17, the errors between the initial estimations and the solutions for joint 1 angle re from -0.04° to 0.16° , which indicates that the extended heuristic iterative method may give accurate initial estimates for joint 1 angle in motion planning.

To illustrate the effectiveness of the proposed algorithm, some other techniques from the literature used to solve IK are compared for the computation time and EE pose accuracy. Kucuk and Bingul [21] carried out a comparative study between NIKA and Newton-Raphson method for a 6-DOF manipulator with offset wrists. The former took

TABLE 8. The solution errors and computation time for random poses.

Poses	Random poses	Random singular poses
Average position error(m)	2.6615×10^{-16}	5.786×10^{-5}
Max position error(m)	6.4259×10^{-15}	9.931×10^{-5}
Average orientation error(deg)	0.5006	0.0053
Max orientation error(deg)	0.9992	0.0138
Average computation time(ms)	18.9	62.2
Max computation time(ms)	36.6	139.5

**FIGURE 17.** The errors between the initial estimations and the solutions for the joint 1 angle.**TABLE 9.** Simulation parameters of HIIKA and NIKA for motion planning.

Parameters	Values
The allowed position error for the extended heuristic iterative method	$\delta = 1 \times 10^{-8}$ m
Max iterations for the extended heuristic iterative method	$\tau = 15$
The search range of joint 1 angle	$\sigma = \pm 0.5^\circ$
The allowed EE position error	$\varepsilon = 1 \times 10^{-5}$ m
The allowed EE orientation error	$\gamma = 0.1^\circ$
Incremental angle	$\alpha = 0.001^\circ$

time for solving IK from 0.035685s to 0.358804s with errors up to 0.0685 unit, and the latter's computation time was from 0.248976s to 0.406556s with errors up to 7.8209 unit. Wang *et al.* [13] proposed a geometrical approach, which took an average of 0.061s to solve IK for a 6-DOF robot with non-spherical wrist. The CCD method in [23] solving the IK for a 6-DOF manipulator took a total of 8s for a path with 72 points. The proposed algorithm in this paper took an average of 18.9ms with position errors up to 6.4259×10^{-15} m, even for singular poses the average computation time was 62.2ms. And a rectangle path with 100 points took a total of 1.7687s using HIIKA.

VI. CONCLUSIONS

This paper presents HIIKA for the IK problem of 6-DOF manipulators with offset or reduced wrists. By reason of

coupled orientation and position, as well as the highly nonlinear multivariate transcendental equations, the natural choice for IK is numerical methods. However, numerical methods are not always economical due to a huge search space dimension, inaccurate initial estimations, singularities, multiple solutions and additional constraints. The significance of HIIKA is that, it improves the deficiencies of these traditional numerical methods. HIIKA can also produce continuous and smooth motion trajectories for each joint. Finally, the experimental results show that HIIKA is advanced and excellent in convergence, accuracy, computational efficiency, and multiple solutions for IK solutions.

APPENDIX A

$$\left\{ \begin{array}{l} {}^0_1 T = \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & 0 \\ s\theta_1 & c\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\ {}^1_2 T = \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s\theta_2 & c\theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ {}^2_3 T = \begin{bmatrix} c\theta_3 & -s\theta_3 & 0 & 0 \\ 0 & 0 & 1 & -d_3 \\ -s\theta_3 & -c\theta_3 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\ {}^3_4 T = \begin{bmatrix} c\theta_4 & -s\theta_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s\theta_4 & -c\theta_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ {}^4_5 T = \begin{bmatrix} c\theta_5 & -s\theta_5 & 0 & 0 \\ 0 & 0 & -1 & d_5 \\ s\theta_5 & c\theta_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\ {}^5_6 T = \begin{bmatrix} c\theta_6 & -s\theta_6 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s\theta_6 & -c\theta_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array} \right.$$

$$\begin{aligned}
{}^6T &= \begin{bmatrix} -1 & 0 & 0 & a_6 \\ 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&\quad \begin{bmatrix} R_{3 \times 3} & P_{3 \times 1} \\ 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} \\
n_x &= -c\theta_6\{c\theta_5[c\theta_4(s\theta_1s\theta_3 - c\theta_1c\theta_2c\theta_3) - c\theta_1s\theta_2s\theta_4] \\
&\quad + s\theta_5(c\theta_3s\theta_1 + c\theta_1c\theta_2s\theta_3)\} + s\theta_6[s\theta_4(s\theta_1s\theta_3 \\
&\quad - c\theta_1c\theta_2c\theta_3) + c\theta_1c\theta_4s\theta_2] \\
n_y &= c\theta_6\{c\theta_5[c\theta_4(c\theta_1s\theta_3 + s\theta_1c\theta_2c\theta_3) + s\theta_1s\theta_2s\theta_4] \\
&\quad + s\theta_5(c\theta_3c\theta_1 - s\theta_1c\theta_2s\theta_3)\} - s\theta_6[s\theta_4(c\theta_1s\theta_3 \\
&\quad + s\theta_1c\theta_2c\theta_3) - s\theta_1c\theta_4s\theta_2] \\
n_z &= -c\theta_6[c\theta_5(c\theta_2s\theta_4 - c\theta_3c\theta_4s\theta_2) + s\theta_2s\theta_3s\theta_5] \\
&\quad - s\theta_6(c\theta_2c\theta_4 + c\theta_3s\theta_2s\theta_4) \\
o_x &= s\theta_6\{c\theta_5[c\theta_4(s\theta_1s\theta_3 - c\theta_1c\theta_2c\theta_3) - c\theta_1s\theta_2s\theta_4] \\
&\quad + s\theta_5(c\theta_3s\theta_1 + c\theta_1c\theta_2s\theta_3)\} + c\theta_6[s\theta_4(s\theta_1s\theta_3 \\
&\quad - c\theta_1c\theta_2c\theta_3) + c\theta_1c\theta_4s\theta_2] \\
o_y &= -s\theta_6\{c\theta_5[c\theta_4(c\theta_1s\theta_3 + s\theta_1c\theta_2c\theta_3)] \\
&\quad + s\theta_1s\theta_2s\theta_4\} + s\theta_5(c\theta_3c\theta_1 - s\theta_1c\theta_2s\theta_3) \\
&\quad - c\theta_6[s\theta_4(c\theta_1s\theta_3 + s\theta_1c\theta_2c\theta_3) - s\theta_1c\theta_4s\theta_2] \\
o_z &= s\theta_6[c\theta_5(c\theta_2s\theta_4 - c\theta_3c\theta_4s\theta_2) + s\theta_2s\theta_3s\theta_5] \\
&\quad - c\theta_6(c\theta_2c\theta_4 + c\theta_3s\theta_2s\theta_4) \\
a_x &= s\theta_5[c\theta_4(s\theta_1s\theta_3 - c\theta_1c\theta_2c\theta_3) - c\theta_1s\theta_2s\theta_4] \\
&\quad - c\theta_5(c\theta_3s\theta_1 + c\theta_1c\theta_2s\theta_3) \\
a_y &= -s\theta_5[c\theta_4(c\theta_1s\theta_3 + s\theta_1c\theta_2c\theta_3) + s\theta_1s\theta_2s\theta_4] \\
&\quad + c\theta_5(c\theta_3c\theta_1 - s\theta_1c\theta_2s\theta_3) \\
a_z &= s\theta_5(c\theta_2s\theta_4 - c\theta_3c\theta_4s\theta_2) - s\theta_2s\theta_3c\theta_5 \\
p_x &= d_5[s\theta_4(s\theta_1s\theta_3 - c\theta_1c\theta_2c\theta_3) + c\theta_1c\theta_4s\theta_2] + d_3c\theta_1s\theta_2 \\
p_y &= -d_5[s\theta_4(c\theta_1s\theta_3 + s\theta_1c\theta_2c\theta_3) - s\theta_1c\theta_4s\theta_2] + d_3s\theta_1s\theta_2 \\
p_z &= -d_5(c\theta_2c\theta_4 + c\theta_3s\theta_4s\theta_2) - d_3c\theta_2
\end{aligned}$$

APPENDIX B

$$\begin{aligned}
\rho &= [(p_x c\theta_1 + p_y s\theta_1)^2 + p_z^2]^{\frac{1}{2}} \\
\phi &= \text{Atan2}(p_z, p_x c\theta_1 + p_y s\theta_1) \\
k &= (p_x^2 + p_y^2 + p_z^2 + d_3^2 - d_5^2)/2d_3 \\
\chi &= (p_x c\theta_1 s\theta_2 + p_y s\theta_1 s\theta_2 - p_z c\theta_2 - d_3)/d_3 \\
\xi &= a_x (c\theta_1 s\theta_2 s\theta_4 - c\theta_4 s\theta_1 s\theta_3 + c\theta_1 c\theta_2 c\theta_3 c\theta_4) \\
&\quad - a_z (c\theta_2 s\theta_1 - c\theta_3 c\theta_4 s\theta_2) \\
\eta &= a_y (c\theta_1 c\theta_3 - c\theta_2 s\theta_1 s\theta_3) \\
&\quad - a_x (c\theta_3 s\theta_1 + c\theta_1 c\theta_2 s\theta_3) - a_z s\theta_2 s\theta_3 \\
\lambda &= n_x (s\theta_1 s\theta_3 s\theta_4 + c\theta_1 c\theta_4 s\theta_2 - c\theta_1 c\theta_2 c\theta_3 s\theta_4) \\
&\quad - n_z (c\theta_2 c\theta_4 + c\theta_3 s\theta_2 s\theta_4) - n_y (c\theta_1 s\theta_3 s\theta_4 \\
&\quad - c\theta_4 s\theta_1 s\theta_2 + c\theta_2 c\theta_3 s\theta_1 s\theta_4)
\end{aligned}$$

$$\begin{aligned}
\tau &= o_x (s\theta_1 s\theta_3 s\theta_4 + c\theta_1 c\theta_4 s\theta_2 - c\theta_1 c\theta_2 c\theta_3 s\theta_4) \\
&\quad - o_z (c\theta_2 c\theta_4 + c\theta_3 s\theta_2 s\theta_4) - o_y (c\theta_1 s\theta_3 s\theta_4 \\
&\quad - c\theta_4 s\theta_1 s\theta_2 + c\theta_2 c\theta_3 s\theta_1 s\theta_4)
\end{aligned}$$

REFERENCES

- [1] D. L. Pieper, "The kinematics of manipulators under computer control," Ph.D. dissertation, Dept. Mech. Eng., Stanford Univ., Stanford, CA, USA, 1968.
- [2] Y. Wei, S. Jian, S. He, and Z. Wang, "General approach for inverse kinematics of nR robots," *Mechanism Mach. Theory*, vol. 75, pp. 97–106, May 2014.
- [3] S. Qiao, Q. Liao, S. Wei, and H.-J. Su, "Inverse kinematic analysis of the general 6R serial manipulators based on double quaternions," *Mechanism Mach. Theory*, vol. 45, no. 2, pp. 193–199, Feb. 2010.
- [4] H. Ananthanarayanan and R. Ordóñez, "Real-time inverse kinematics of (2n+1) DOF hyper-redundant manipulator arm via a combined numerical and analytical approach," *Mechanism Mach. Theory*, vol. 91, pp. 209–226, Sep. 2015.
- [5] J. Angeles, "On the numerical solution of the inverse kinematic problem," *Int. J. Robot. Res.*, vol. 4, no. 2, pp. 21–37, 1985.
- [6] A. A. Goldenberg, J. A. Apkarian, and H. W. Smith, "A new approach to kinematic control of robot manipulators," *J. Dyn. Syst. Meas. Control*, vol. 109, no. 2, pp. 97–103, 1987.
- [7] S. Chiaverini, B. Siciliano, and O. Egeland, "Review of the damped least-squares inverse kinematics with experiments on an industrial robot manipulator," *IEEE Trans. Control Syst. Technol.*, vol. 2, no. 2, pp. 123–134, Jun. 1994.
- [8] W. Xu, J. Zhang, B. Liang, and B. Li, "Singularity analysis and avoidance for robot manipulators with nonspherical wrists," *IEEE Trans. Ind. Electron.*, vol. 63, no. 1, pp. 277–290, Jan. 2015.
- [9] F. Yin, Y.-N. Wang, Y. Yang, and Y.-M. Yang, "Inverse kinematics solution for robot manipulator based on neural network under joint subspace," *Int. J. Comput. Commun. Control*, vol. 7, no. 3, pp. 459–472, Mar. 2012.
- [10] Q. Shi and J. Xie, "A research on inverse kinematics solution of 6-DOF robot with offset-wrist based on Adaboost neural network," in *Proc. IEEE Int. Conf. CIS/IEEE Conf. RAM*, Nov. 2017, pp. 370–375.
- [11] Z. Bingul, H. M. Ertunc, and C. Oysu, "Comparison of inverse kinematics solutions using neural network for 6R robot manipulator with offset," in *Proc. IEEE ICSC Congr. Comput. Intell. Methods Appl.*, Dec. 2005, p. 5.
- [12] Z. Bingul, H. M. Ertunc, and C. Oysu, "Applying neural network to inverse kinematic problem for 6R robot manipulator with offset wrist," in *Adaptive and Natural Computing Algorithms*. Vienna, Austria: Springer, 2005, pp. 112–115.
- [13] X. Wang, D. Zhang, and C. Zhao, "The inverse kinematics of a 7R 6-degree-of-freedom robot with non-spherical wrist," *Adv. Mech. Eng.*, vol. 9, no. 8, pp. 1–11, Aug. 2017.
- [14] D. C. Trinh, D. Zlatanov, M. Zoppi, and R. M. Molfino, "A geometrical approach to the inverse kinematics of 6R serial robots with offset wrists," in *Proc. ASME Int. Design Eng. Tech. Conf. Comput. Inf. Eng. Conf.*, Aug. 2015, pp. 1–10.
- [15] Y. Lin and H. Min, "Inverse kinematics of modular manipulator robot with shoulder offset based on geometric method mixed with analytical method algorithm," in *Proc. IEEE Int. Conf. Cyber Technol. Automat., Control, Intell. Syst.*, Jun. 2015, pp. 1198–1203.
- [16] Z. Fu, W. Yang, and Z. Yang, "Solution of inverse kinematics for 6R robot manipulators with offset wrist based on geometric algebra," *J. Mech. Robot.*, vol. 5, no. 3, p. 031010, 2013.
- [17] X. Huo, Y. Liu, J. Li, and H. Liu, "Optimal kinematic control of humanoid arms with offset wrist," in *Proc. IEEE Int. Conf. Mechatronics Automat.*, Aug. 2014, pp. 1011–1016.
- [18] Z.-Z. Liu, H.-Y. Liu, Z. Luo, and F. Wang, "Inverse kinematics algorithm for 6-DOF robots with offset wrist based on offset compensation," *J. Northeastern Univ. (Natural Sci.)*, vol. 33, no. 6, pp. 870–874, 2012.
- [19] M. Raghavan and B. Roth, "Inverse kinematics of the general 6R manipulator and related linkages," *J. Mech. Des.*, vol. 115, no. 3, pp. 502–508, Sep. 1993.
- [20] A. Pashevich, "Real-time inverse kinematics for robots with offset and reduced wrist," *Control Eng. Pract.*, vol. 5, no. 10, pp. 1443–1450, 1997.
- [21] S. Kucuk and Z. Bingul, "Inverse kinematics solutions for industrial robot manipulators with offset wrists," *Appl. Math. Model.*, vol. 38, nos. 7–8, pp. 1983–1999, 2014.

- [22] A. Aristidou and J. Lasenby, "FABRIK: A fast, iterative solver for the Inverse Kinematics problem," *Graph. Models.*, vol. 73, no. 5, pp. 243–260, Sep. 2011.
- [23] L.-C. T. Wang and C. C. Chen, "A combined optimization method for solving the inverse kinematics problems of mechanical manipulators," *IEEE Trans. Robot. Autom.*, vol. 7, no. 4, pp. 489–499, Aug. 1991.



JING XU received the B.S. and M.S. degrees from the School of Mechanical Engineering, Liaoning Shihua University, Fushun, China, in 2013 and 2016, respectively. He is currently pursuing the Ph.D. degree with the School of Mechanical Engineering and Automation, Northeastern University, Shenyang, China. His research interests include robot motion planning and robot control.



YU HE received the B.S. degree from the School of Mechanical Engineering and Automation, Liaoning Technical University, Fuxin, China, in 2014, and the M.S. degree from the School of Mechanical Engineering and Automation, Northeastern University, Shenyang, China, in 2016, where he is currently pursuing the Ph.D. degree. His research interests include deep learning, pattern recognition, and intelligent inspection.



ZHIPENG DONG received the B.S. degree in mechanical design manufacture and automation from the Harbin Institute of Technology, Harbin, Heilongjiang, China, in 2013, and the M.S. degree in mechanical design and theory from Northeastern University, Shenyang, Liaoning, China, in 2015, where he is currently pursuing the Ph.D. degree in mechanical engineering. His research interest includes robotic vision, manipulation, and machine learning techniques.



KECHEN SONG received the B.S., M.S., and Ph.D. degrees from the School of Mechanical Engineering and Automation, Northeastern University, Shenyang, China, in 2009, 2011, and 2014, respectively. He has been a Teacher with the Northeastern University of China since 2014. His research interest covers vision-based inspection system for steel surface defects, surface topography, image processing, and pattern recognition.



YUNHUI YAN received the B.S., M.S., and Ph.D. degrees from the School of Mechanical Engineering and Automation, Northeastern University, Shenyang, China, in 1981, 1985, and 1997, respectively. He has been a Teacher with the Northeastern University of China since 1982 and became a professor in 1997. From 1993 to 1994, he was a Visiting Scholar with the Tohoku National Industrial Research Institute. His research interest covers intelligent inspection, image processing, and pattern recognition.

• • •