A Multiple Solutions Approach to the Inverse Kinematics Problem of a General Serial Manipulator

Yanping Fan\textsuperscript{1}, Pin Li\textsuperscript{1,2*}, Xijin Zhen\textsuperscript{2} and Zongbao Shen\textsuperscript{1}

\textsuperscript{1} School of Mechanical Engineering, Jiangsu University, Zhenjiang 212013, China
\textsuperscript{2} Shipbuilding Technology Research Institute, Shanghai 200000, China
Email: lip@ujs.edu.cn

Abstract. The inverse kinematics problem (IKP) is at the center of computer control algorithms for general serial manipulator. The exact number of feasible solutions of the IKP is highly dependent on the manipulator structure as well as the desired end-effector pose, and the existing inverse kinematics methods have difficulties to solve all feasible solutions of a general serial manipulator. In this work, a multiple solutions approach is applied to the IKP. Then, three numerical examples are used to validate the effectiveness of the multiple solutions approach, and the numerical results demonstrate that the multiple solutions approach is effective and practicable. Finally, an infinite solutions example is presented to show that the proposed approach could also provide limited number of numerical solutions in singularity.

Keywords: General serial manipulator; Inverse kinematics problem; Multiple solutions.

1. Introduction
The inverse kinematics problem (IKP) of a general serial manipulator is to find a set of joint variables to place the end-effector into a desired pose. It is obvious that the IKP of a serial manipulator requires the solution of nonlinear kinematics equations [1]. However, the exact number of feasible solutions is highly dependent on the manipulator structure as well as the desired end-effector pose [2]. In a case of six degrees of freedom (DOFs) serial manipulator, there are at most sixteen feasible solutions depending on the desired end-effector pose [3]. The well-known methods include closed-form [4], symbolic elimination [5, 6], continuation methods and iterative methods.

Closed-form solutions are faster and quickly identify all possible solutions. The disadvantage is that they are not general, but robot dependent [1]. Piper has shown a sufficient condition for a serial manipulator to have a closed-form solution, including that 3 consecutive revolute joint axes intersect [4]. Most industrial manipulators have such 3 consecutive revolute joint axes, which is due to the fact that it permits more efficient coordination software. Symbolic elimination methods involve eliminating of variables from nonlinear kinematics equations [5]. Lee and Liang [7] proposed a full theoretically correct solution, and reduced the degree of the polynomial to 16 in the half tangent of one joint variable. The disadvantage is that these methods involve symbolic preprocessing and symbolic matrix operations to avoid severe numerical singularity [8]. Continuation methods involve tracking a homotopy path from a beginning system with known solutions to a goal system whose solutions are sought as the beginning system is transformed into the goal system [9]. The disadvantage is that number of homotopy paths is usually lead to heavy computation burden and some solutions may be missed [10]. Iterative methods are common methods to solve the nonlinear kinematics equations [11]. The disadvantage of iterative methods is that Newton-like algorithms are considered as “local method” and the computational efficiency is hindered by the inverse Jacobian matrix calculation [12].
In a word, it is difficult to solve all feasible solutions of a general serial manipulator by using the existing inverse kinematic methods. A kind of multiple solutions approach is applied to the IPK of a general manipulator based on the iterative methods in this work. The effectiveness of the proposed method is validated by some numerical examples, and the compute time is improved by parallel computing.

2. Multiple Solutions Approach
Iterative methods can obtain a single solution based on a “good” original guess and an appropriate step size control. The modified method has quick convergence and self-correction. In this section, we found that there are some features focusing on the initial used when using differential kinematics as step size control. Then, a one dimensional search for the set of initial joint variables is used to find all possible feasible solution and the proposed multiple solutions procedure is presented.

2.1. Iterative Methods
Iterative methods convert the IKP to a differential equation, and the computational procedure can be expressed as follow:

\[ x_k = f(\theta_k) \]
\[ \Delta \theta_k = f^{-1}(\theta_k) \Delta x_k \]
\[ \theta_{k+1} = \theta_k + \Delta \theta_k \]
\[ x_{k+1} = f(\theta_{k+1}), \quad k = 0, 1, 2, \ldots \]

where \( x_k \) is the end-effector pose, \( \theta_k \) is the joint variables, \( f \) represents the nonlinear kinematics equations, \( f(\theta_k) = \partial f / \partial \theta_k \) is Jacobian matrix of the joint displacement \( \theta_k \), \( \theta_0 \) represents the initial variables. \( \Delta x \) can be expressed using differential kinematics as follows:

\[ \Delta x = \begin{pmatrix} \Delta x_p \\ \Delta x_R \end{pmatrix} = \begin{pmatrix} P_t - P_a \\ \text{ve}x(R_tR_a^T - I) \end{pmatrix} \]

where the actual end-effector pose \( x_a \) and the target end-effector pose \( x_t \), respectively; \( \text{ve}x() \) means converting a skew-symmetric matrix to a vector, e.g. given a vector \( \nu = (v_x, v_y, v_z)^T \).

Nowadays, a “good” initial guess was necessary by employing iterative methods to obtain a higher precision solution. Nevertheless, most of the initial can converge quickly when using differential kinematics as step size control. The difference is that it is not strictly monotonically decreased in the iterative process as shown in figure 1, where the orientation of the end-effector is illustrated by Euler angle and the position of the end-effector is illustrated by unit Cartesian coordinates.

![Figure 1](image_url)

Figure 1. Not strictly monotonically decreased.
2.2. Multiple Solutions Procedure

It is found that different results of feasible solutions are mainly dependent on the initial used. The one is a singular initial used; the other is the endless iteration. First, a set of joint variables $\theta_r^{(1)}$ can be randomly generated to act as the initial $\theta_0$. Then, a singular initial can be easily seen by checking the range and null spaces of the Jacobian matrix. And a set of joint variables $\theta_r^{(2)}$ can be randomly re-generated to replace. Third, the endless iteration can be aborted when the number of iterations exceeds a user-defined number max_iter.

To solve all feasible solutions, a one dimensional search for the set of joint variables $\theta_r$ is used in this paper. max_search is a user-defined maximum search time. It is found that the value of max_search more than 100 is enough to obtain all solutions. Formally, we have the following multiple solutions procedure:

Procedure: Multiple_solutions_Procedure ($x_t$)
Step 0: max_search:=100, max_iter:=30, i:=1;
Step 1: if i=max_search then goto Step 5;
                 else Randomly generate the joint variables $\theta_r$, $\theta(i)$: = $\theta_r$, k:=1;
Step 2: Compute the Jacobian matrix $J(\theta(i))$;
         Compute the actual end-effector pose $x_a = f(\theta(i))$;
Step 3: if $x_a = x_t$ then goto Step 4;
             else if Jacobian matrix $J(\theta(i))$ is singular then
                        Randomly re-generate the joint variables $\theta_r$, $\theta(i)$: = $\theta_r$, goto Step 2;
             else if k=max_iter then
                        Randomly generate the joint variables $\theta_r$, $\theta(i)$: = $\theta_r$, k:=1, goto Step 2;
             else Compute $\Delta x$ using the equation (2)~(6) as described in Section II,
                        $\Delta \theta = f^{-1}(\theta(i)) \Delta x$, $\theta(i) := \theta(i) + \Delta \theta$, k:=k+1, goto Step 2;
Step 4: i:=i+1, goto Step 1;
Step 5: return $\theta_f$: = $\theta$

3. Numerical Examples

In this section, three numerical examples have been used to validate the effectiveness of the multiple solutions approach. And the linkage parameters of the three previous examples are given in table 1.

Example 1: “Special” Robot, Last Three Axes Intersecting. The system partially decouples and has 8 solutions as realized the desired pose in Tsai and Morgan;

Example 2: Symmetrical Robot. The system is symmetric about the fourth joint axis and has 6 solutions as realized the desired pose in Tsai and Morgan;

Example 3: General Robot. The system has 12 solutions as realized the desired pose in Tsai and Morgan;

Table 1. Linkage parameters of three previous examples, (a) example 1, (b) example 2, (c) example 3.

| $i$ | $a_i$ (unit) | $d_i$ (unit) | $\alpha_i$ (deg) |
|-----|-------------|-------------|-----------------|
| 1   | 0.0         | 0.0         | -90.00          |
| 2   | 1.000000    | 0.345000    | 0.0             |
| 3   | -0.04700    | 0.0         | 90.00           |
| 4   | 0.0         | 1.000000    | -90.00          |
| 5   | 0.0         | 0.0         | 90.00           |
| 6   | 0.0         | 0.130000    | 0.0             |
The iterative processes of the three examples by using the proposed approach are presented in figure 2 - figure 4. All feasible solutions of the examples have been obtained quickly, and no solutions are missed. The results demonstrate the effectiveness of the proposed method to the IKP of a general serial manipulator.

| i  | α_i (unit) | d_i (unit) | α_i (deg) |
|----|------------|------------|-----------|
| 1  | 0.45000    | 0.50000    | 80.00     |
| 2  | 0.55000    | 0.60000    | 93.00     |
| 3  | 0.75000    | 0.40000    | 120.00    |
| 4  | 0.75000    | 1.00000    | 120.00    |
| 5  | 0.55000    | 0.40000    | 93.00     |
| 6  | 0.45000    | 0.60000    | 80.00     |

| i  | α_i (unit) | d_i (unit) | α_i (deg) |
|----|------------|------------|-----------|
| 1  | 0.50000    | 0.187500   | 80.00     |
| 2  | 1.00000    | 0.375000   | 15.0      |
| 3  | 0.12500    | 0.250000   | 120.00    |
| 4  | 0.62500    | 0.875000   | 75.00     |
| 5  | 0.31250    | 0.500000   | 100.00    |
| 6  | 0.25000    | 0.125000   | 60.0      |

Figure 2. Example 1 Last Three Axes intersect with 8 solutions.
In addition, compared with symbolic elimination methods which involving symbolic handing, both continuation methods and iterative methods are numerical and faster. The compute time of the two numerical methods is discussed as following.

The computational complexity of continuation methods depends on the number of homotopy paths. Examples 1~3 were solved in Tsai and Morgan [10] using two continuation algorithms SYMPOL and SYMMAN, running on the IBM 370-3033 processor. SYMPOL is the generic continuation method determined by theory. It is not customized to the IKP, backed up by much computational experience. SYMMAN is a special, faster and simplified continuation method but some solutions may be missed. Meanwhile, the proposed approach is improved by adopting multithread technology to implement parallel computing in this paper, running on a PC with Intel® Core i5-5200U CPU. The compute time of the proposed approach and the two continuation methods is given in table 2. The results show that the proposed approach has a better practicable.
Table 2. Compute time of the solution algorithms (s).

| Example  | Continuous method | Proposed approach |
|----------|-------------------|-------------------|
|          | SYMPOL            | SYMMAN            | Non-parallel | Parallel |
| Example 1| 212.363           | 2.012             | 4.6644      | 0.043    |
| Example 2| 266.513           | 8.028             | 6.5988      | 0.036    |
| Example 3| 250.868           | 10.618            | 5.3664      | 0.045    |

4. Infinite Solutions Example
In this section, a “special” robot PUMA560 and its singular position is used to show that the proposed approach can provide limited number of numerical solutions in singularity. The singular position would be skipped in the middle of the path in the computer control algorithms which might make the motion becomes not smooth. Figure 5 shows the kinematics structure.

Figure 5. Kinematic structure of “special” robot PUMA560.

The axes of 3 revolute joints intersect at the same point. The special structure enables the decomposition of the IKP into inverse position kinematics and inverse orientation kinematics. Generally, there are four solutions for the inverse position kinematics, and the inverse orientation kinematics problem has two solutions. So the total number of the solutions is eight. However, there would be infinite solutions when toward to a position where \( \theta_5 = 0 \). The revolute joint axes of joint 4 and joint 6 would be same and the position is singular. Table 3 gives some numerical solutions of the “special” robot as towards a singular position \( \theta_t = [30 \ -45 \ 15 \ 30 \ 0 \ 0] \).

Table 3. Infinite solutions example.

| No. | \( \theta_1 \) (deg) | \( \theta_2 \) (deg) | \( \theta_3 \) (deg) | \( \theta_4 \) (deg) | \( \theta_5 \) (deg) | \( \theta_6 \) (deg) | \( (\theta_4 + \theta_6) \) (deg) |
|-----|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
| 1   | 178.877             | -135.000            | 170.383             | -59.457             | -17.462             | -64.134             | -123.591            |
| 2   | 178.877             | -135.000            | 170.383             | 120.543             | 17.462              | 115.866             | 236.409             |
| 3   | 178.877             | 122.613             | 15.000              | 163.978             | 110.555             | 51.848              | 215.825             |
| 4   | 178.877             | 122.613             | 15.000              | -16.022             | -110.555            | -128.152            | -144.175            |
| 5   | 30.000              | 57.387              | 170.383             | -180.000            | -102.230            | -150.000            | -330.000            |
| 6   | 30.000              | -45.000             | 15.000              | 66.607              | 0.000               | -36.607             | 30.000              |
| 7   | 30.000              | -45.000             | 15.000              | 77.456              | 0.000               | -47.456             | 30.000              |
| 8   | 30.000              | -45.000             | 15.000              | -46.812             | 0.000               | 76.812              | 30.000              |

5. Conclusions
This paper presents a multiple solutions approach to solve all feasible solutions for the IKP of a general serial manipulator. A one dimensional search for the set of joint variables is used to find all possible feasible solution.
Three numerical examples have been used to validate the effectiveness of the multiple solutions approach. The compute time of the proposed approach is improved by parallel computing. The results demonstrate that the multiple solutions approach is effective and practicable. Meanwhile, an infinite solutions example of a “special” robot is presented, and the results show that the proposed approach can also provide limited number of numerical solutions in singularity.

Reference
[1] Siciliano B and Khatib O 2007 Springer Handbook of Robotics (New York: Springer-Verlag): 27-29.
[2] Manseur R and Doty K L 1989 Int. J. Robot. Res. 8(8): 75-78.
[3] Liu K, Yu J J and Kong X W 2021 Mech. Mach. Theory 155: 104110.
[4] Song, G L, Su S and Li Y L 2021 Robotica 39: 572-581.
[5] Raghaven M and Roth B 1991 The 5th Int. Symp. on Robotics Research: 263-269.
[6] Duffy J and Crane C 1980 Mech. Mach. Theory 15(3): 153-169.
[7] Lee H Y and Liang C G 1988 Mech. Mach. Theory 23(3): 219-226.
[8] Raghavan M and Roth B 1993 J. Mech. Des. 115(3): 502-508.
[9] Manocha D and Canny J F 1992 Proc. - IEEE Int. Conf. on Robotics and Automation: 383-389.
[10] Tsai L W and Morgan A P 1985 Journal of Mechanisms, Transmissions, and Automation in Design 107(2): 189-200.
[11] Morgan A P 1983 ACM Trans. Math. Softw. 9(1): 1-17.
[12] Manseur R and Doty K I L 1988 Int. J. Robot. Res. 7(3): 52-63.