

Article

# Solving the Time-Varying Inverse Kinematics Problem for the Da Vinci Surgical Robot

Long Bai <sup>\*</sup>, Jianxing Yang, Xiaohong Chen, Pei Jiang , Fuqiang Liu, Fan Zheng and Yuanxi Sun

State Key Laboratory of Mechanical Transmission, Chongqing University, Chongqing 400044, China; 20132295@cqu.edu.cn (J.Y.); chenxh@cqu.edu.cn (X.C.); peijiang@cqu.edu.cn (P.J.); liufq@cqu.edu.cn (F.L.); 20160702066@cqu.edu.cn (F.Z.); sunyuanxi@cqu.edu.cn (Y.S.)

\* Correspondence: bailong@cqu.edu.cn

Received: 15 November 2018; Accepted: 3 February 2019; Published: 6 February 2019



**Featured Application:** The work of this paper is mainly applied for the inverse kinematics solving of Da Vinci surgical robot and its similar robotic systems that do not satisfy the Pieper principle.

**Abstract:** A dialytic-elimination and Newton-iteration based quasi-analytic inverse kinematics approach is proposed for the 6 degree of freedom (DOF) active slave manipulator in the Da Vinci surgical robot and other similar systems. First, the transformation matrix-based inverse kinematics model is derived; then, its high-dimensional nonlinear equations are transformed to a high-order nonlinear equation with only one unknown variable by using the dialytic elimination with a unitary matrix. Finally, the quasi-analytic solution is eventually obtained by the Newton iteration method. Simulations are conducted, and the result show that the proposed quasi-analytic approach has advantages in terms of accuracy (error < 0.00004 degree (or mm)), solution speed (<20 ms) and is barely affected by the singularity during intermediate calculations, which proves that the approach meets the real-time and high-accuracy requirements of master–slave mapping control for the Da Vinci surgical robots and other similar systems. In addition, the proposed approach can also serve as a design reference for other types of robotic arms that do not satisfy the Pieper principle.

**Keywords:** minimally invasive surgery robot; inverse kinematics; dialytic elimination; Newton iteration

## 1. Introduction

In recent years, medical robots have increasingly been used as fundamental surgical instrument/equipment [1], among which the Da Vinci system developed by Intuitive Surgical Company is the most successful [2]. The Da Vinci robotic system consists of three components: the surgeon console (master manipulators), the patient trolley (slave manipulators), and the imaging system [3]. The surgeon manipulates two master handles at the master remote console, which can acquire high-resolution, binocular, three-dimensional, magnified views of the operative field as compared with open surgery [4]. However, this kind of master–slave operation mode poses a great challenge to the real-time and accurate performance of the master–slave mapping control, among which the inverse kinematics problem is the first obstacle. The Da Vinci system and similar surgical robot systems satisfy the following two features:

- (1) contain mechanical structures to generate Remote Center of Motion [5] (RCM, a fixed point on the active arm, there are two methods to generate RCM based motion: (a) virtual RCM, (b) physically constrained RCM [6]), and the structure can be simplified;
- (2) do not satisfy the Pieper principle.

At present, the main methods of robotic inverse kinematics include the analytic method and the numerical method. The analytic approach uses vector, spinor or Lie algebra to obtain results [7–9], which can only obtain satisfied solutions in limited robotic configurations that should satisfy the Pieper principle [10,11], etc. However, to imitate the surgeon operation, the manipulators of most thoracic and abdominal surgery robots, including the Da Vinci system, contain rotational/translational kinematic pairs, tilt angle joints, and offset joints at the tip (artificial wrist part) that are designed for additional flexibility, which is beyond the solution capability of the analytic approach.

One feasible strategy is to simplify the kinematic model to acquire analytical results. For example, Ai et al. and Podsedkowski et al. [12,13] obtained an approximate solution by separating the position and orientation of the slave manipulators, the solution deviation of which needs to be manually eliminated by experienced surgeons via visual inspection. Zhang et al. [14] solved the inverse kinematics of a 4-DOF surgical robot using the elimination method, and Mezouar et al. [15] used unit dual quaternions to model the kinematics, however, outcomes of their studies benefited strongly from the simplicity of the arm configuration. In addition, Fu et al. [16,17] obtained an approximate solution based on the differential transformation that increases error feedbacks to compensate the cumulative error, the final precision of which can reach 0.18 mm.

Another way to solve the inverse kinematics problem of the surgical robot that does not satisfy the Pieper principle is the Jacobian matrix-based numerical method [18]. However, this method involves a Jacobian matrix, which may lose versatility due to its singularity problem as well as the low convergence speed of the iterative solutions. To improve the numerical solution performance, a numerical method based on the position increment of robotic joints was proposed [18]. Specifically, to make this method suitable for the inverse kinematics analysis of a robotic manipulator with an offset wrist, Bu et al. [19] used the method of decoupling DOFs based on a cut-off point that is limited by the manipulator configuration. In addition, modern methods such as the neural network method [20], the genetic algorithm [21], the electromagnetism-like mechanism [22,23], the hybrid electromagnetism-like mechanism [24], etc., have been gradually explored for possible solutions of the inverse kinematics of surgical robots. However, these methods either require a large amount of sample data or suffer from poor convergence speed; therefore, these methods are not suitable for the rapid development of a master–slave manipulator in surgical robots.

To address the lack of efficient and accurate inverse kinematics solutions for the Da Vinci system, we propose a quasi-analytic solution method that involves dialytic elimination and the Newton iteration method. The dialytic elimination method [25] is used to transform the high-dimensional equations into a nonlinear equation containing only one unknown variable, which is then solved by the Newton iteration method. In applying this method to the inverse kinematics solution of slave manipulator, we show dual superiorities: (a) barely affected by the singularity problem and (b) maintain rapidity.

The steps of our proposed method as follows:

- (1) simplify the structure of RCM, using an equivalent mechanism to take the place of the RCM mechanism.
- (2) establish the coordinate system, this step can be realized by the D–H (Denavit–Hartenberg) method or combine the characteristics of the structure to make sure there are more origins of the coordinate system can be set up at the same point.
- (3) obtain the mathematical model of forward and inverse kinematics via the matrix transformation method. Then, transform the high-dimensional equations into a nonlinear equation containing only one unknown parameter using the dialytic elimination method, and solve it by the Newton iteration method.

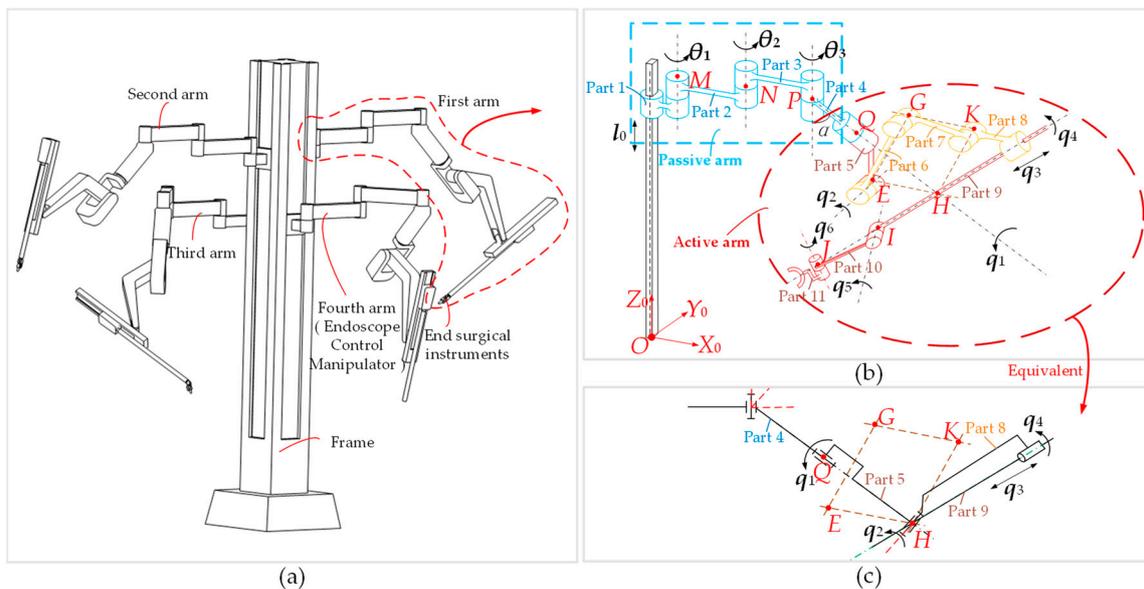
The rest of this paper is organized as follows: Section 2 presented the kinematics analysis of the Da Vinci system; Section 3 solves the inverse kinematics by the Newton iteration method; Section 4

analyzes our solution and compared it with other methods via a simulation; Section 5 concludes the paper.

## 2. Kinematics Analysis

### 2.1. Forward Kinematics Analysis

During surgery using the Da Vinci medical robot, dual articulated arms (slave manipulators) work as the right and left hand of the surgeon, respectively. The coordinate frame of the master/slave manipulator is set the same as the coordinate frame of human sight, i.e., the world coordinate frame. Therefore, in Figure 1, the origin of the base coordinate frame of the slave manipulator is located at the bottom of the robot frame. The  $Z_0$ -axis is upward, along the robot frame.



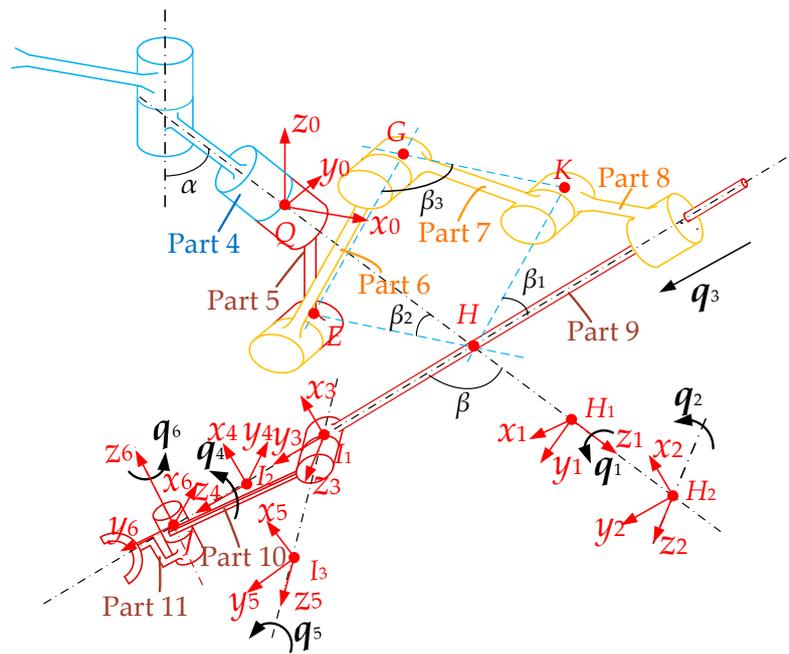
**Figure 1.** Structure schematic of the Da Vinci patient trolley. (a) Composition of the patient trolley, including four arms (first arm, second arm, third arm and fourth arm) and a frame. (b) Structure schematic of the first arm (slave manipulator), including the passive arm (dashed rectangle area) and active arm (dashed ellipse area), and point  $H$ , which is the RCM. (c) Equivalent diagram of the parallelogram mechanism (yellow part).

Before surgery, the angles of passive joints  $M$ ,  $N$ , and  $P$  and the position of the  $M$  joint on the frame are adjusted manually and then remain fixed during surgery. The joints that mainly assist the surgeon are the active joints of the slave manipulator:  $Q$ ,  $E$ ,  $G$ ,  $K$ ,  $H$ ,  $I$ , and  $J$  in Figure 1b.

The DOFs of the slave manipulator includes four DOFs:  $l_0$ ,  $\theta_1$ ,  $\theta_2$  and  $\theta_3$ . The parts 6, 7, and 8 in the active arm belong to the parallelogram mechanism (i.e., these three parts cannot move freely but under the motion constraint of the parallelogram mechanism in Figure 1c. Then only parts 4, 5, 8 and 9 can move freely in the slave manipulator, i.e., part 5 rotates around part 4, there is rotation between part 6 and part 5, and there is rotation between part 9 and part 8. Therefore, the entire slave manipulator will have 6 DOFs when the additional 3 DOFs of the  $H$  joint are considered.

### 2.2. Mathematical Model of Forward Kinematics

According to the previous analysis, an active arm coordinate system is established in Figure 2. The origin of the base coordinate  $x_0y_0z_0$  (0-coordinate frame) of the active arm is established at the joint  $Q$  between the passive and active arms, the direction of which is the same as that of the frame coordinate  $X_0Y_0Z_0$ .



**Figure 2.** Structure diagram of the active arm of the slave manipulator. To more clearly, we draw  $H$  and  $I$  as  $H, H_1, H_2$  and  $I_1, I_2, I_3$  respectively. 1-coordinate frame fixed on part 5; 2-coordinate frame fixed on part 8; 3-coordinate frame fixed on part 9; 4-coordinate frame fixed on part 9; 5-coordinate frame fixed on part 10; 6-coordinate frame fixed on part 11.

Figure 2 shows the six DOFs of the active arm in the slave manipulator, namely five rotating joints:  $q_1, q_2, q_4, q_5, q_6$ , and a translational joint:  $q_3$ .

Six transform matrixes frame from  $x_1y_1z_1$  (1-coordinate frame) to  $x_6y_6z_6$  (6-coordinate frame) can be obtained as follows:

$$\left\{ \begin{array}{l} T = \{-(\pi - \alpha)\}_x \{q_1\}_z \{l_{QH}\}_z \\ {}^1_2T = \{-(\beta - \frac{\pi}{2})\}_x \{\frac{\pi}{2}\}_y \{q_2\}_z \\ {}^2_3T = \{q_3\}_y \\ {}^3_4T = \{-\frac{\pi}{2}\}_x \{q_4\}_z \\ {}^4_5T = \{\frac{\pi}{2}\}_x \{q_5\}_z \\ {}^5_6T = \{l_{IJ}\}_y \{\frac{\pi}{2}\}_y \{q_6\}_z \end{array} \right. \quad (1)$$

where  ${}^i_jT$  means transform matrix from the  $i$ -coordinate frame to the  $j$ -coordinate frame;  $\{ \}_x, \{ \}_y$  and  $\{ \}_z$  represent the rotation angle around the  $x$ -axis,  $y$ -axis and  $z$ -axis, respectively;  $\{ \}_x, \{ \}_y$  and  $\{ \}_z$  denotes the translational movement along the  $x$ -axis,  $y$ -axis and  $z$ -axis, respectively.

Then, the pose matrix of the active arm at the endpoint of the slave manipulator can be obtained as follows:

$${}^0_6T = {}^0_1T {}^1_2T {}^2_3T {}^3_4T {}^4_5T {}^5_6T = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R & P \\ \mathbf{0} & 1 \end{bmatrix}, \quad (2)$$

where  $R$  and  $P$  are the orientation and position of the slave manipulator endpoint, respectively. Substituting Equation (1) into Equation (2) yields Equation (3):

$${}^0_6T = \{-(\pi - \alpha)\}_x \{q_1\}_z \{l_{QH}\}_z \{-(\beta - \frac{\pi}{2})\}_x \{\frac{\pi}{2}\}_y \{q_2\}_z \{q_3\}_y \{-\frac{\pi}{2}\}_x \{q_4\}_z \{\frac{\pi}{2}\}_x \{q_5\}_z \{l_{IJ}\}_y \{\frac{\pi}{2}\}_y \{q_6\}_z. \quad (3)$$

### 2.3. Mathematical Model of Inverse Kinematics

In the case of inverse kinematics, the known pose matrix is  ${}^0_6T$  in Equation (3), and the unknowns are  $q_1, q_2, q_3, q_4, q_5$  and  $q_6$ .

Left-multiplying the matrices  $T = \{q_3\}^y \{q_2\}_z^{-1} \{\frac{\pi}{2}\}_y^{-1} \{-(\beta - \frac{\pi}{2})\}_x^{-1} \{l_{QH}\}^z \{q_1\}_z^{-1} \{-(\pi - \alpha)\}_x^{-1}$  on both sides of Equation (3) yields Equation (4):

$$T^0_6T = \left\{ -\frac{\pi}{2} \right\}_x \{q_4\}_z \left\{ \frac{\pi}{2} \right\}_x \{q_5\}_z \{l_{IJ}\}_y \left\{ \frac{\pi}{2} \right\}_y \{q_6\}_z. \tag{4}$$

To simplify the complexity of the formula, we construct a unitary matrix [26]  $U$  as follows. After introducing the matrix  $U$ , the variables in the  ${}^0_6T$  can be replaced by the Euler formula.

$$U = \begin{bmatrix} \frac{1}{\sqrt{2}}i & -\frac{1}{\sqrt{2}}i & 0 & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5}$$

In addition, we know that  $E = U^{-1}U$ . Left-multiply the inverse of unitary matrix  $U^{-1}$  and right-multiply the matrix  $U$  on both sides of Equation (4). Then, multiply the  $E$  matrix between these two matrices to transform Equation (4) into the following:

$$Y_3X_2C_1X_1C_2C_3 = C_4X_4C_5X_5C_6X_6. \tag{6}$$

Here,  $C_1 = U^{-1}[\{l_{QH}\}^z\{\pi/2 - \beta\}_z]^{-1}U$ ,  $C_2 = U^{-1}[\{\alpha - \pi\}_x]^{-1}U$ ,  $C_3 = U^{-1}{}^0_6TU$ ,  $C_4 = U^{-1}\{-\pi/2\}_zU$ ,  $C_5 = U^{-1}\{\pi/2\}_zU$ , and  $C_6 = U^{-1}\{l_{IJ}\}_y\{\pi/2\}_yU$ . Clearly,  $C_1, C_2, C_3, C_4, C_5$  and  $C_6$  are constant matrices or parameter matrices related to the structure of the robot. In addition,  $X_j$  ( $j = 1, 2, 4, 5, 6$ ) is as follows:

$$X_j = \begin{cases} T_{Xj}^{-1} & j = 1, 2 \\ T_{Xj} & j = 4, 5, 6 \end{cases}, \tag{7}$$

where  $T_{Xj}$  ( $j = 1, 2, 4, 5, 6$ ) represents the results of the transform matrix of  $q_j$  rotating around the z-axis, which undergoes the left-multiplication of  $U^{-1}$  and right-multiplication of  $U$ . Let  $x_j = \cos(q_j) + isin(q_j)$ ,  $x_j^{-1} = \cos(q_j) - isin(q_j)$ ; thus,  $T_{Xj}$  can be expressed as follows:

$$T_{Xj} = U^{-1}\{q_i\}_zU = \begin{bmatrix} x_j & 0 & 0 & 0 \\ 0 & x_j^{-1} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{8}$$

Let  $Y_j$  represents the result of the transform matrix  $q_j$  along the y-axis, which undergoes the left-multiplication of  $U^{-1}$  and right-multiplication of  $U$ , and  $x_j = q_j$ ; then,  $Y_j$  can be expressed as follows:

$$Y_j = U^{-1}\{q_i\}_yU = \begin{bmatrix} 1 & 0 & 0 & -\frac{\sqrt{2}}{2}x_j \\ 0 & 1 & 0 & -\frac{\sqrt{2}}{2}x_j \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{9}$$

Take  $A = Y_3X_2C_1X_1C_2C_3$ ,  $B = C_4X_4C_5X_5C_6X_6$ . Then the dimensions of matrix  $A$  and matrix  $B$  are both  $4 \times 4$ . Each element in matrix  $A$  is a function containing  $x_1, x_2$ , and  $x_3$ , while the elements in matrix  $B$  are functions containing  $x_4, x_5$ , and  $x_6$ . In addition, corresponding elements in matrix  $A$  and matrix  $B$  are equal, i.e.,  $A_{ij} = B_{ij}$ . The specific contents of the elements are detailed in Appendix A.

To realize element elimination, we evaluated elements in matrix  $A$  and  $B$  that do not contain  $x_6$  and  $x_6^{-1}$ :  $B_{13}, B_{14}, B_{23}, B_{24}, B_{33}$  and  $B_{34}$  to establish Equation (10):

$$\left. \begin{aligned} A_{13} &= B_{13} \\ A_{14} &= B_{14} \\ A_{23} &= B_{23} \\ A_{24} &= B_{24} \\ A_{33} &= B_{33} \\ A_{34} &= B_{34} \end{aligned} \right\} \quad (10)$$

To further eliminate  $x_4$  and  $x_5$ , the linear combination of  $x_4$  and  $x_5$ , and their reciprocals are treated as new variables, i.e.,  $X = [x_4x_5, x_4x_5^{-1}, x_4^{-1}x_5, x_4^{-1}x_5^{-1}, x_4^{-1}, x_5^{-1}, x_4, x_5, x_4^0x_5^0]^T$ , which contains nine variables. Because there are only six equations in Equation (10), three more equations are required, which are attainable by conducting nonlinear computations on the elements in Equation (10), as shown in Equation (11).

$$\left. \begin{aligned} A^{L1} &= -A_{34}A_{23} + A_{24}A_{33} = -B_{34}B_{23} + B_{24}B_{33} = B^{L1} \\ A^{L2} &= -A_{24}A_{13} + A_{14}A_{23} = -B_{24}B_{13} + B_{14}B_{23} = B^{L2} \\ A^{L3} &= -A_{13}A_{34} + A_{14}A_{33} = -B_{13}B_{34} + B_{14}B_{33} = B^{L3} \end{aligned} \right\} \quad (11)$$

Then, by combining Equations (10) and (11):

$$[D(x_1, x_2, x_3)]_{9 \times 9} X = 0. \quad (12)$$

The matrix  $[D(x_1, x_2, x_3)]_{9 \times 9}$  contains only the variables  $x_1, x_2$ , and  $x_3$  since  $X$  is nonzero. The condition that Equation (12) has a solution is that the determinant of  $[D(x_1, x_2, x_3)]_{9 \times 9}$  equals zero. The following equation, which contains  $x_1, x_2$ , and  $x_3$ , can be obtained provided that  $|D(x_1, x_2, x_3)| = 0$ :

$$(C'_1x_1 + C'_2 + C'_3x_1^{-1})x_2 = -(C'_4x_1 + C'_5 + C'_6x_1^{-1})x_2^{-1}, \quad (13)$$

where  $C'_j$  is a constant related to the robot's parameters. Considering that  $x_1, x_2, x_4, x_5$ , and  $x_6$  can be defined as  $x_j = \cos(q_j) + i\sin(q_j)$  and  $x_j^{-1} = \cos(q_j) - i\sin(q_j)$ , then  $x_1, x_2, x_4, x_5$ , and  $x_6$  are all non-zero. Therefore,  $x_2$  can be expressed by  $x_1$  according to Equation (13):

$$x_2^2 = \frac{-(C'_1x_1 + C'_2 + C'_3x_1^{-1})}{[(C'_4x_1 + C'_5 + C'_6x_1^{-1})]} = f_1(x_1). \quad (14)$$

To represent  $x_3$  by  $x_1$  and  $x_2$ , we construct the equations  $A^{L4} = A_{23}A_{14} + A_{13}A_{24} + A_{33}A_{34}$  and  $B^{L4} = B_{23}B_{14} + B_{13}B_{24} + B_{33}B_{34}$ . Let  $A^{L4} = B^{L4}$ , then

$$(C'_7x_1 + C'_9 + C'_{12}x_1^{-1})x_3x_2 + (C'_8x_1 + C'_{11} + C'_{13}x_1^{-1})x_3x_2^{-1} + C'_{10} = 0. \quad (15)$$

Equation (15) contains only three variables,  $x_1, x_2$ , and  $x_3$ , therefore,  $x_3$  can be expressed by  $x_1$  and  $x_2$  as:

$$x_3 = f_2(x_1, x_2). \quad (16)$$

Because both  $A^{L2}$  and  $A^{L3}$  are functions of  $x_1, x_2$  and  $x_3$ , by combining  $A^{L2} = B^{L2}$  with  $A^{L3} = B^{L3}$ , the following can be obtained:

$$\begin{bmatrix} x_4 \\ x_4^{-1} \end{bmatrix} = \begin{bmatrix} -\frac{i}{l_{ij}} & -\frac{\sqrt{2}i}{l_{ij}} \\ -\frac{i}{l_{ij}} & \frac{\sqrt{2}i}{l_{ij}} \end{bmatrix} \begin{bmatrix} A^{L2} \\ A^{L3} \end{bmatrix}. \quad (17)$$

Therefore,

$$x_4 = -\frac{i}{l_{IJ}}A^{L2} - \frac{\sqrt{2}i}{l_{IJ}}A^{L3} = f_3(x_1, x_2, x_3). \tag{18}$$

Similarly, by combining  $A_{13} = B_{13}$  and  $A_{14} = B_{14}$  with Equation (16), the following can be obtained:

$$\begin{bmatrix} x_5 \\ x_5^{-1} \end{bmatrix} = \begin{bmatrix} -\frac{\sqrt{2}}{(A^{L2}i - l_{IJ})} & -\frac{\sqrt{2}il_{IJ}}{(A^{L2}i - l_{IJ})} \\ -\frac{\sqrt{2}i}{(A^{L2} - l_{IJ}i)} & -\frac{\sqrt{2}l_{IJ}}{(A^{L2} - l_{IJ}i)} \end{bmatrix} \begin{bmatrix} A_{14} \\ A_{13} \end{bmatrix}. \tag{19}$$

Therefore,  $x_5$  can be represented by  $x_1, x_2,$  and  $x_3$ :

$$x_5 = -\frac{\sqrt{2}A_{14}}{(A^{L2}i - l_{IJ})} - \frac{\sqrt{2}il_{IJ}A_{13}}{(A^{L2}i - l_{IJ})} = f_4(x_1, x_2, x_3). \tag{20}$$

Combined with Equations (16) to (20) and  $A_{11} = B_{11}$ ,  $x_6$  can finally be represented by  $x_1$ :

$$x_6 = \frac{(b_1i + b_2)x_1x_2^{-1} + (b_3i + b_4)x_2^{-1} + (b_5i + b_6)x_1^{-1}x_2^{-1}}{\frac{1}{8}x_5x_4 + \frac{i}{4}x_4 - \frac{1}{8}x_5^{-1}x_4 + \frac{1}{4}x_5 + \frac{1}{4}x_5^{-1} + \frac{1}{8}x_5x_4^{-1} - \frac{i}{4}x_4^{-1} - \frac{1}{8}x_5^{-1}x_4^{-1}} = f_5(x_1), \tag{21}$$

where  $b_j$  is a constant related to the robot's parameters.

From Equations (16) to (21), the variables  $x_2, x_3, x_4, x_5,$  and  $x_6$  can be represented with the variable  $x_1$ . Let  $A_{24}$  and  $B_{24}$  be equal, then:

$$F(x_1) = A_{24} - B_{24} = 0. \tag{22}$$

By solving Equation (22),  $x_1$  can be solved, then  $x_2, x_3, x_4, x_5,$  and  $x_6$  can be subsequently solved by Equation (16) to (21). Specifically, we know from Equation (14) that  $x_2 = \pm\sqrt{f_1(x_1)}$ , therefore,  $F(x_1)$  will have two expressions:

$$F(\xi_1) = \begin{cases} F_1(x_1)x_2 = \sqrt{f_1(x_1)} \\ F_2(x_1)x_2 = -\sqrt{f_1(x_1)} \end{cases}. \tag{23}$$

Equation (23) will be electively solved according to the minimum norm of the distance the joint moved.

### 3. Inverse Kinematics Solution

From the derivation of the forward and inverse kinematics of the Da Vinci slave manipulator, it is known that the key to solving inverse kinematics is to solve the nonlinear Equation (22).

#### 3.1. Inverse Kinematics Solution Method

The Newton–Raphson method is an efficient numerical iterative approach to solve nonlinear equations in real and complex domains. In this paper, we construct the following iterative formula:

$$x_{n+1} = x_n - \frac{F(x_n)}{F'(x_n)}. \tag{24}$$

#### 3.2. Initial Value of the Inverse Kinematics Solution

The motion ranges of each joint in the Da Vinci slave manipulator are as follows:  $\theta_1$  from  $-90^\circ$  to  $90^\circ$ ,  $\theta_2$  from  $-135^\circ$  to  $0^\circ$ ,  $l_{IH}$  from 0 to 315 mm,  $\theta_3$  from  $-180^\circ$  to  $180^\circ$ ,  $\theta_4$  from  $-90^\circ$  to  $90^\circ$ , and  $\theta_5$  from  $-90^\circ$  to  $90^\circ$ . In the previously defined  $x_1 = \cos(q_1) + isin(q_1)$ ,  $q_1$  is the rotation angle of the first joint with a range of  $[-\pi/2, \pi/2]$ . A finite number of values are obtained by extracting values within these available ranges at a certain angle interval (our subsequent calculation results show that a less than  $5^\circ$  angle interval is acceptable) to solve the forward kinematics, the results of which are stored as the initial value database for the inverse kinematics solution. By selecting the data from the initial

value database that is the least squares approximate to the required inverse kinematics as the initial values, the Newton iteration method will successfully converge.

### 3.3. Inverse Kinematics Solution Process

According to the inverse kinematics of the Da Vinci slave manipulator and the derived solution process, a detailed solution flow chart is constructed in Figure 3.

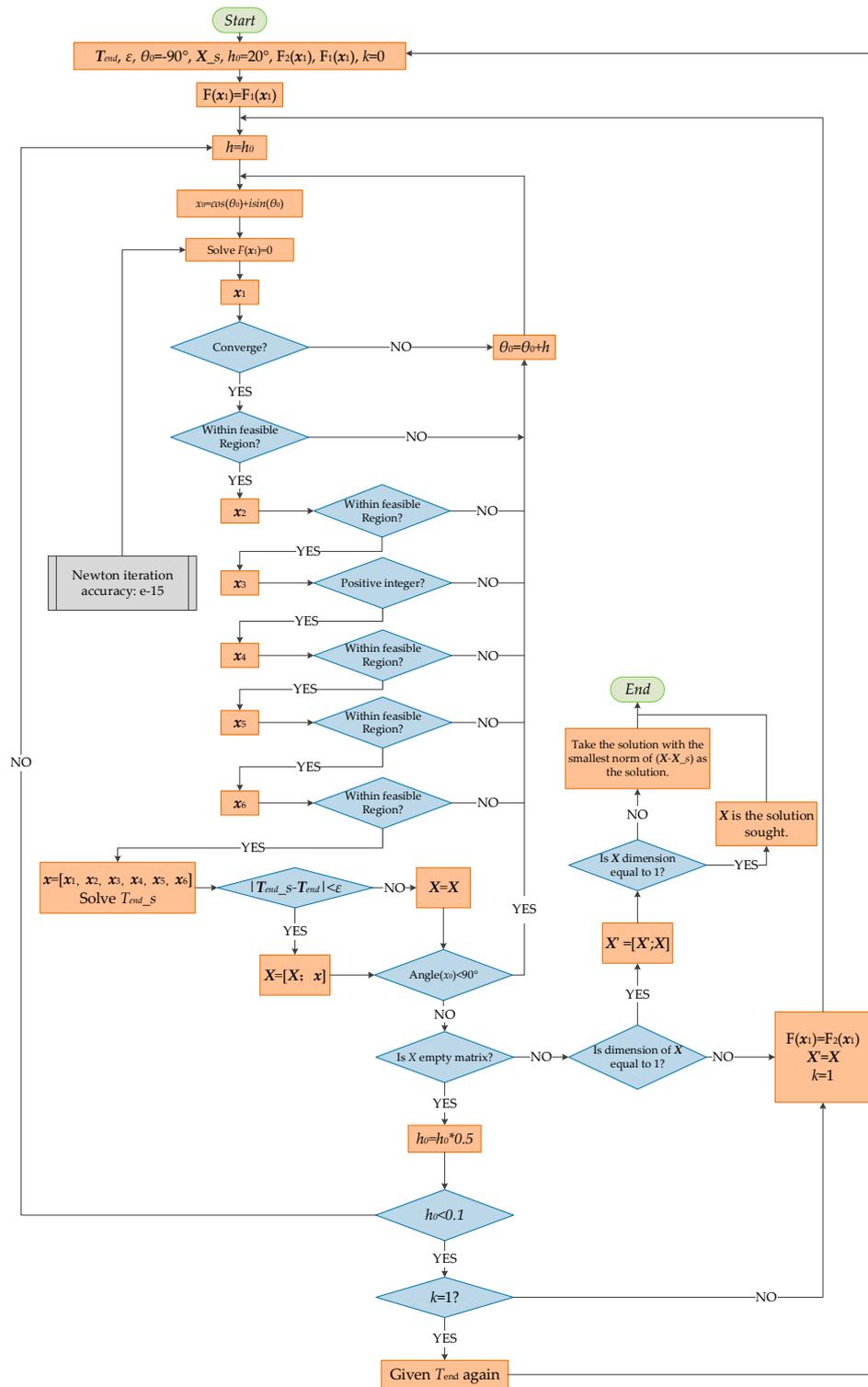


Figure 3. Inverse kinematics solution process of the Da Vinci slave manipulator.

In Figure 3,  $T_{end}$  is the pose matrix at the endpoint of the robot (i.e.,  ${}^0_6T$ ),  $\varepsilon$  is the precision of the solution,  $\theta_0$  is the initial value,  $h$  is the initial step value, and  $X_s$  represents the values of each joint at previous positions. The value  $k$  is used to determine whether both forms of  $F(x_1)$  have been solved.

#### 4. Example Analysis and Discussion

##### 4.1. Parameters of the Da Vinci Slave Manipulator

According to Figure 4, the size parameters of the Da Vinci slave manipulator are shown in Table 1.

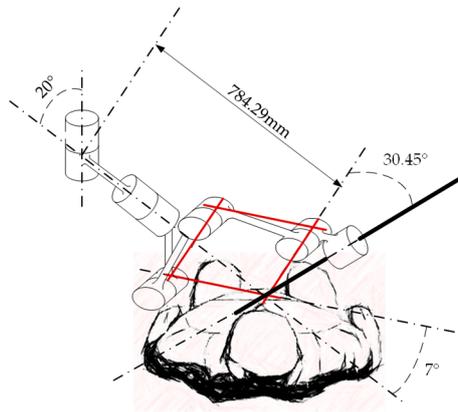


Figure 4. Range of motion of the slave manipulator.

Table 1. Structure parameters of the Da Vinci slave manipulator.

$\beta_1$ (°)	$\beta_2$ (°)	$l_{QH}$ (mm)	$\beta_3$ (°)	$\alpha$ (°)	$l_{IJ}$ (mm)
30.45	7	784.29	0	20	10

##### 4.2. Solution of Proposed Method

As shown in Table 2 (the form is referenced by the book: An Introduction to Error Analysis [27]), the input values are a set of arbitrarily selected joint parameters within the effective range of joint space. The outputs in Table 2 are the calculated values by our proposed method.  $h$  represents the step size of the solution, and the error is calculated as follows:

$$\varepsilon = |\text{Output} - \text{Input}|. \tag{25}$$

In this paper, the solution precision of the Newton–Raphson method is  $1E-10$   $\varepsilon = 1E-7$  for  $T_{end}$ ,  $\theta_0$  takes the minimum value of  $-90^\circ$  within the feasible region, and the initial step size is  $h_0 = 20^\circ$ . The test carried on CodeBlocks 16.01 based on the hardware that a PC platform running with an Intel (R) Core (TM) i5-4460 CPU, clocked at 3.2 GHz (memory: 4 GB).

We also use the same input to run the procedure on the “R12CCPU-V” MELSEC iQ-R series C language controller from Mitsubishi Electric and VxWorks 6.9 operating system, the results are list in Table 3.

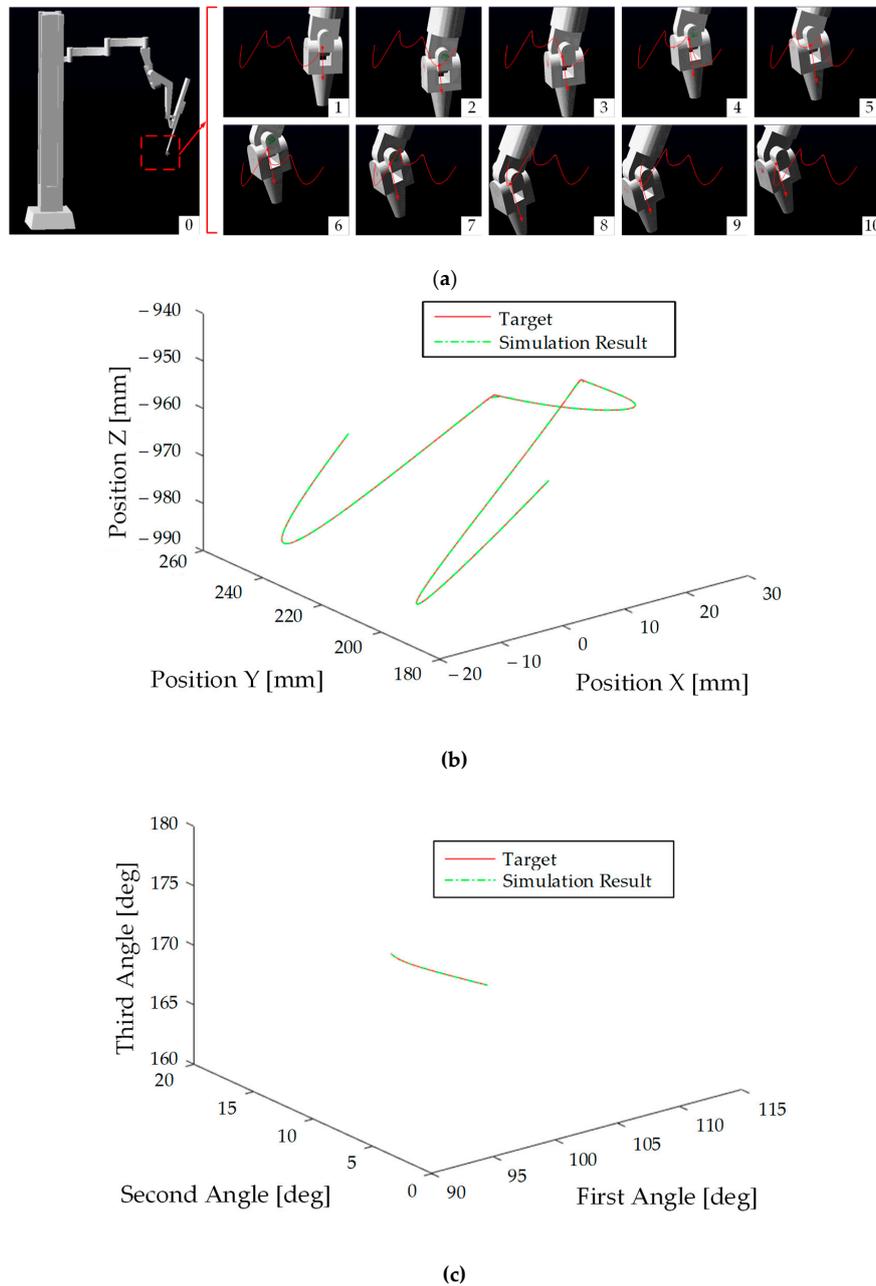
**Table 2.** Inverse kinematics solution of the Da Vinci slave manipulator.

Joint \ Step (°)	Output h = 20	Error	Input	Joint \ Step (°)	Output h = 20	Error	Input
$\theta_1$ (°)	10.00000000000004	0.00000000000004	10	$\theta_1$ (°)	15.39999999999996	0.00000000000003	15.4
$\theta_2$ (°)	-124.999999995	0.000000005	-125	$\theta_2$ (°)	-116.499999995	0.000000005	-116.5
$l_{IH}$ (mm)	100.0000000000001	0.0000000000001	100	$l_{IH}$ (mm)	101.3000000000001	0.0000000000001	101.3
$\theta_3$ (°)	99.999999995	0.000000005	100	$\theta_3$ (°)	86.3999999999996	0.0000000000004	86.4
$\theta_4$ (°)	-14.0000000000008	0.0000000000008	-14	$\theta_4$ (°)	13.6999999999998	0.0000000000002	13.7
$\theta_5$ (°)	18.000000002	0.000000002	18	$\theta_5$ (°)	-15.899999996	0.000000004	-15.9
t (s)	0.421			t (s)	0.406		
Joint \ Step (°)	Output h = 10	Error	Input	Joint \ Step (°)	Output h = 20	Error	Input
$\theta_1$ (°)	-50.24000000000009	0.00000000000009	-50.24	$\theta_1$ (°)	48.3689999999998	0.0000000000002	48.369
$\theta_2$ (°)	-60.75999999999998	0.00000000000002	-60.76	$\theta_2$ (°)	-80.59700000000002	0.0000000000002	-80.597
$l_{IH}$ (mm)	126.9400000000001	0.0000000000001	126.94	$l_{IH}$ (mm)	125.4680000000002	0.0000000000002	125.468
$\theta_3$ (°)	101.379999995	0.000000005	101.38	$\theta_3$ (°)	-80.5939999999998	0.0000000000002	-80.594
$\theta_4$ (°)	-15.9400000000002	0.0000000000002	-15.94	$\theta_4$ (°)	-68.5920000000003	0.0000000000003	-68.592
$\theta_5$ (°)	24.789999999	0.000000001	24.79	$\theta_5$ (°)	15.237999998966513		15.238
t (s)	0.484			t (s)	0.390		
Joint \ Step (°)	Output h = 20	Error	Input	Joint \ Step (°)	Output h = 20	Error	Input
$\theta_1$ (°)	-56.54835000000004	0.00000000000004	-56.54835	$\theta_1$ (°)	43.36587495200008	0.00000000000008	43.365874952
$\theta_2$ (°)	-53.895759999999999	0.00000000000001	-53.89576	$\theta_2$ (°)	-125.687459795	0.000000005	-125.6874598
$l_{IH}$ (mm)	168.85324	0.00004	168.8532	$l_{IH}$ (mm)	136.8452159699998	0.0000000000002	136.84521597
$\theta_3$ (°)	-96.358469995	0.000000005	-96.35847	$\theta_3$ (°)	75.312548960004	0.0000000000004	75.31254896
$\theta_4$ (°)	-50.268430000001	0.000000000001	-50.26843	$\theta_4$ (°)	25.984572680002	0.0000000000002	25.98457268
$\theta_5$ (°)	50.468509999	0.000000001	50.46851	$\theta_5$ (°)	-42.58794151	0.000000001	-42.587941523
t (s)	0.437			t (s)	0.437		

**Table 3.** Inverse kinematics solution of the Da Vinci slave manipulator (based on better hardware and software).

Step (°)	Output	Error	Input	Step (°)	Output	Error	Input
<b>Joint</b>	<b>h = 20</b>			<b>Joint</b>	<b>h = 20</b>		
$\theta_1$ (°)	9.99999999999998	0.000000000000002	10	$\theta_1$ (°)	15.39999999999995	0.000000000000005	15.4
$\theta_2$ (°)	-124.999999995	0.000000005	-125	$\theta_2$ (°)	-116.499999995	0.000000005	-116.5
$l_{IH}$ (mm)	99.9999999999999	0.00000000000001	100	$l_{IH}$ (mm)	101.3000000000002	0.0000000000002	101.3
$\theta_3$ (°)	99.999999995	0.000000005	100	$\theta_3$ (°)	86.3999999999998	0.0000000000002	86.4
$\theta_4$ (°)	-13.9999999999995	0.00000000000005	-14	$\theta_4$ (°)	13.69999999999991	0.00000000000009	13.7
$\theta_5$ (°)	18.000000002	0.000000002	18	$\theta_5$ (°)	-15.899999996	0.000000004	-15.9
$t$ (s)	0.016			$t$ (s)	0.016		
Step (°)	Output	Error	Input	Step (°)	Output	Error	Input
<b>Joint</b>	<b>h = 20</b>			<b>Joint</b>	<b>h = 20</b>		
$\theta_1$ (°)	-50.240000000000002	0.000000000000002	-50.24	$\theta_1$ (°)	48.36899999999998	0.00000000000002	48.369
$\theta_2$ (°)	-60.759999999999998	0.000000000000002	-60.76	$\theta_2$ (°)	-80.59699999999998	0.00000000000002	-80.597
$l_{IH}$ (mm)	126.94000000000001	0.00000000000001	126.94	$l_{IH}$ (mm)	125.4680000000002	0.0000000000002	125.468
$\theta_3$ (°)	101.379999995	0.000000005	101.38	$\theta_3$ (°)	-80.5939999999998	0.0000000000002	-80.594
$\theta_4$ (°)	-15.94000000000008	0.00000000000008	-15.94	$\theta_4$ (°)	-68.59200000000003	0.00000000000003	-68.592
$\theta_5$ (°)	24.789999999	0.000000001	24.79	$\theta_5$ (°)	15.237999999	0.000000001	15.238
$t$ (s)	0.016			$t$ (s)	0.016		
Step (°)	Output	Error	Input	Step (°)	Output	Error	Input
<b>Joint</b>	<b>h = 20</b>			<b>Joint</b>	<b>h = 20</b>		
$\theta_1$ (°)	-56.54835000000004	0.000000000000004	-56.54835	$\theta_1$ (°)	43.36587495200008	0.000000000000008	43.365874952
$\theta_2$ (°)	-53.895759999999999	0.000000000000001	-53.89576	$\theta_2$ (°)	-125.687459795	0.000000005	-125.6874598
$l_{IH}$ (mm)	168.853200000000007	0.000000000000007	168.8532	$l_{IH}$ (mm)	136.8452159699999	0.00000000000001	136.84521597
$\theta_3$ (°)	-96.358469995	0.000000005	-96.35847	$\theta_3$ (°)	75.3125489599998	0.0000000000002	75.31254896
$\theta_4$ (°)	-50.2684300000001	0.0000000000001	-50.26843	$\theta_4$ (°)	25.9845726799996	0.0000000000004	25.98457268
$\theta_5$ (°)	50.468509999	0.000000001	50.46851	$\theta_5$ (°)	-42.58794151	0.000000001	-42.587941523
$t$ (s)	0.016			$t$ (s)	0.016		

In order to verify the proposed method, we build a simulation model in Adams. Firstly, we gave a random movement for the simulation model, then we obtained the matrix of the end pose from the simulation result (Figure 5a), which included the position and the orientation. Secondly, the matrixes were imported to the algorithm we proposed as input. Thirdly, we used our solution to drive the robot model. In the final results, depicted in Figure 5b, we see that the curve obtained by our algorithm is very close to the target curve. This means the proposed method is of high accuracy.



**Figure 5.** Simulation results: (a) simulation in Adams, (b) end position comparison between target and our result, (c) the end orientation (Euler angles) comparison between target and our result.

### 4.3. Discussion of the Solution

As noted in the previous section, the main focus of the solution includes the solution accuracy, real-time performance (solution speed) and the singularity problem (based on the joint position solution or not).

(1) Solution accuracy

Table 2 shows that for any input joint parameter, whether an integer or a decimal value accurate to nine decimal places, the error between the calculated values (output) and the given values (input) of six joint parameters of the slave manipulator is within 0.00004 degree (or mm), which demonstrates that the proposed approach for the Da Vinci surgical robot inverse kinematics is capable of finding effective solutions with high accuracy.

(2) Solution speed

The number of calculation steps of the algorithm/program is one of the influential factors that affect the solution speed. As schematically shown in Figure 3, the number of steps of the inverse kinematics approach proposed in this paper depends mainly on two factors:

① The number of Newton iterations in the best-case scenario, the numerical solution can be obtained by one iteration when the initial value is perfectly selected. In general, if the given initial value ensures convergence, then the solution of  $F(x_1) = 0$  is expected to take approximate 20 iterations; in cases where the convergence fails, the program is designed to iterate at most 50 times and then exit to avoid a dead loop.

② The step size. when the step size  $h$  is large, the iteration times are reduced, thereby shorten the calculation time. However, the selection of initial values will be relatively imprecise, resulting in higher possibility of solution failure due to unreasonable initial values. In cases where the step size is small, the initial values tend to be more precise, resulting in a greater possibility of obtaining a solution. However, the resulting shortcoming is the demand for more calculation steps and longer time.

An improved calculation process is proposed by taking both factors into account: first, start the procedure by following the flow chart shown in Figure 3 with a relatively large step size; second, reduce the step size gradually for more precise initial values if the equation has no solution. This approach boosts the speed of calculation but also reduces the possibility of solution failure.

(3) Singularity influences

Our method abandons the use of Jacobian matrix inversion. Therefore, there will be barely no singular solution in the condition of a good initial value from a mathematical perspective. On the other hand, because our algorithm is based on the position increment of the manipulator’s joints, Then, the motion joint position trajectory is generated by interpolation based on the initial joint position and the end joint position. The joint position interpolation is not affected by the singularity.

4.4. Comparison with Other Methods

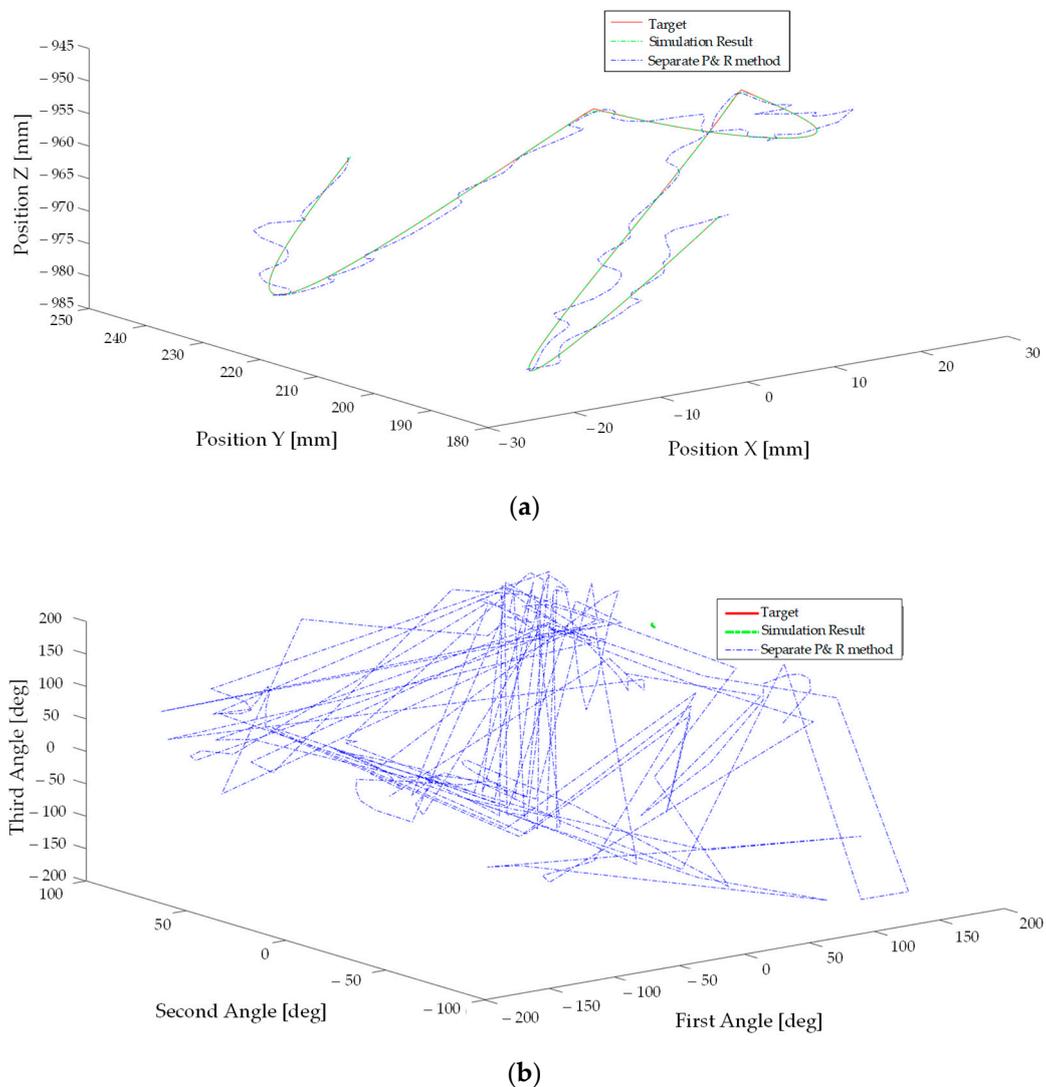
Table 4 compared our method with another two typical methods in applicability, complexity of calculation, solution precision, solution speed and singularity influences.

Table 4. Comparison between the proposed method and others.

Method	Applicability	Complexity of Calculation	Solution Precision	Solution Speed	Singularity Influences
separates the position and orientation [10,11]	especially for the structure satisfied Pieper principle	low	high	quick	need to consider about
method based on the Jacobian matrix [16,28]	all	high	middle	not quick	need to consider about
The proposed method	especially for the structure not satisfied Pieper principle	middle	high	quick	barely considering about

Particularly, we choose separates the position and orientation method (name as “separate P&R method” below) to applied on the Da Vinci slave manipulator, and the comparison with the proposed method within the same simulation model is shown in Figure 6. The position curve of “separate P&R method” is following the target with some fluctuation, while the orientation result is far away from the correct solution, which means that the Separate P&R method is not suitable for the slave manipulator

that dissatisfied the Pieper principle. Even though increasing the error feedback to compensate for the results may increase the solution precision, the compensation operation will take a large amount of time, which proves the superiority of our proposed method in terms of solution precision and speed.



**Figure 6.** The endpoint comparison between results of our proposed method and the separate P&R method. (a) The position result comparison; (b) the orientation result comparison.

## 5. Conclusions

(1) In this paper, an inverse kinematics mathematical model of the 6-DOF active arm of the Da Vinci surgical robot established by using a coordinate transformation matrix method according to its mechanical motion diagram and working characteristics. By introducing a unitary matrix, the mathematical model on the real plane is transformed to a complex plane, consequently avoiding the relatively complex trigonometric calculations associated with the conventional solution procedure. Following the steps proposed, this approach can apply to other similar robot kinematic problems.

(2) The dialytic elimination method serves to individually eliminate the joint variables and ultimately obtains a high-order nonlinear equation with only one unknown variable of the first joint. Subsequently, the Newton iteration method is introduced to solve the nonlinear equation on the basis of selecting multiple initial values reasonably in the feasible region. The test shows that carefully changing the step size and selecting initial values in the feasible region for the iteration solution of the nonlinear equation will significantly reduce the possibility of solution failure.

(3) In theory, the accuracy of the approach can reach an error of  $< 0.00004$  degree (or mm) and achieve a solution time of 0.5 s (Intel (R) Core i5-4460 CPU, clocked at 3.2 GHz (memory: 4 GB)). Running the procedure on better hardware and software (based on the “R12CCPU-V” MELSEC iQ-R series C language controller from Mitsubishi Electric and VxWorks 6.9 operating system) decreases the solution time to 20 ms, which fully satisfies the real-time performance requirement of surgical robots.

(4) A method that combines dialytic elimination and the Newton iteration method is applied to solve the inverse kinematics problem of the slave manipulators of the master–slave surgical robot. The proposed approach achieves higher accuracy than other solutions based on position and orientation separation while the real-time performance is satisfied. Moreover, in contrast to other numerical solutions, the solution method is based on the position of the joint and is therefore barely not affected by any singularity. Further, the application of our approach removes the restriction of the design of the master manipulator configuration.

**Author Contributions:** Conceptualization, L.B. and J.Y.; methodology, X.C.; software, F.Z.; validation, J.Y., L.B., Y.S. and X.C.; formal analysis, J.Y.; investigation, P.J.; resources, F.L.; data curation, L.B.; writing—original draft preparation, J.Y.; writing—review and editing, L.B., X.C. and Y.S.; visualization, X.C.; supervision, L.B.; project administration, L.B.; funding acquisition, X.C.

**Funding:** This research was funded by the Special Cooperation Program for Higher Education Institutions Collaborative Innovation (grant no. KH2016006), the Foundation for Sci & Tech Research Project of Chongqing Science & Technology Commission (grant nos. cstc2016jcyjA0472, and cstc2017zdcy-zdzzX0007), the National Natural Science Foundation of China (grant nos. 51705050, and 51709023), and Fundamental Research Funds for the Central Universities (grant no. 2018CDGFJX0022, 2018CDQYHK0029).

**Acknowledgments:** The authors gratefully acknowledge Zixiang Zhang for his valuable support with language editing.

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

## Appendix A

$$\begin{aligned}
 A_{11} &= (c_1i + c_2)x_1x_2^{-1} + (c_3i + c_4)x_2^{-1} + (c_5i + c_6)x_1^{-1}x_2^{-1}; \\
 A_{12} &= (c_7i + c_8)x_1x_2^{-1} + (c_9i + c_{10})x_2^{-1} + (c_{11}i + c_{12})x_1^{-1}x_2^{-1}; \\
 A_{13} &= (c_{13}i + c_{14})x_1x_2^{-1} + (c_{15}i + c_{16})x_2^{-1} + (c_{17}i + c_{18})x_1^{-1}x_2^{-1}; \\
 A_{14} &= (c_{19}i + c_{20})x_1x_2^{-1} + (c_{21}i + c_{22})x_2^{-1} - \frac{\sqrt{2}}{2}x_3 + (c_{23}i + c_{24})x_1^{-1}x_2^{-1}; \\
 A_{21} &= (c_{25}i + c_{26})x_1x_2 + (c_{27}i + c_{28})x_2 + (c_{29}i + c_{30})x_2x_1^{-1}; \\
 A_{22} &= (c_{31}i + c_{32})x_1x_2 + (c_{33}i + c_{34})x_2 + (c_{35}i + c_{36})x_2x_1^{-1}; \\
 A_{23} &= (c_{37}i + c_{38})x_1x_2 + (c_{39}i + c_{40})x_2 + (c_{41}i + c_{42})x_2x_1^{-1}; \\
 A_{24} &= (c_{43}i + c_{44})x_1x_2 + (c_{45}i + c_{46})x_2 - \frac{\sqrt{2}}{2}x_3 + (c_{47}i + c_{48})x_2x_1^{-1}; \\
 A_{31} &= (c_{49}i + c_{50})x_1 + (c_{51}i + c_{52})x_1^{-1}; \\
 A_{32} &= (c_{53}i + c_{54})x_1 + (c_{55}i + c_{56})x_1^{-1}; \\
 A_{33} &= (c_{57}i + c_{58})x_1 + (c_{58} - c_{57}i)x_1^{-1}; \\
 A_{34} &= (c_{59}i + c_{60})x_1 + (c_{60} - c_{59}i)x_1^{-1}; \\
 B_{11} &= \frac{1}{8}x_6x_5x_4 + \frac{1}{4}x_6x_4 - \frac{1}{8}x_6x_5^{-1}x_4 + \frac{1}{4}x_6x_5 + \frac{1}{4}x_6x_5^{-1} + \frac{1}{8}x_6x_5x_4^{-1} - \frac{1}{4}x_6x_4^{-1} - \frac{1}{8}x_6x_5^{-1}x_4^{-1}; \\
 B_{12} &= \frac{1}{8}x_5x_4x_6^{-1} - \frac{1}{4}x_4x_6^{-1} - \frac{1}{8}x_5^{-1}x_4x_6^{-1} + \frac{1}{4}x_5x_6^{-1} + \frac{1}{4}x_5^{-1}x_6^{-1} + \frac{1}{8}x_5x_4^{-1}x_6^{-1} + \frac{1}{4}x_4^{-1}x_6^{-1} - \frac{1}{8}x_5^{-1}x_4^{-1}x_6^{-1}; \\
 B_{13} &= \frac{-i\sqrt{2}}{8}x_5x_4 - \frac{i\sqrt{2}}{8}x_5^{-1}x_4 - \frac{i\sqrt{2}}{4}x_5 + \frac{i\sqrt{2}}{4}x_5^{-1} - \frac{i\sqrt{2}}{8}x_5x_4^{-1} - \frac{i\sqrt{2}}{8}x_5^{-1}x_4^{-1}; \\
 B_{14} &= \left( \frac{\sqrt{2}}{8}x_5x_4 - \frac{\sqrt{2}}{8}x_5^{-1}x_4 + \frac{\sqrt{2}}{4}x_5 + \frac{\sqrt{2}}{4}x_5^{-1} + \frac{\sqrt{2}}{8}x_5x_4^{-1} - \frac{\sqrt{2}}{8}x_5^{-1}x_4^{-1} \right) l_{IJ}; \\
 B_{21} &= -\frac{1}{8}x_6x_5x_4 - \frac{1}{4}x_6x_4 + \frac{1}{8}x_6x_5^{-1}x_4 + \frac{1}{4}x_6x_5 + \frac{1}{4}x_6x_5^{-1} - \frac{1}{8}x_6x_5x_4^{-1} + \frac{1}{4}x_6x_4^{-1} + \frac{1}{8}x_6x_5^{-1}x_4^{-1}; \\
 B_{22} &= -\frac{1}{8}x_6^{-1}x_5x_4 + \frac{1}{4}x_6^{-1}x_4 + \frac{1}{8}x_6^{-1}x_5^{-1}x_4 + \frac{1}{4}x_6^{-1}x_5 + \frac{1}{4}x_6^{-1}x_5^{-1} - \frac{1}{8}x_6^{-1}x_5x_4^{-1} + \frac{1}{4}x_6^{-1}x_4^{-1} + \frac{1}{8}x_6^{-1}x_5^{-1}x_4^{-1}; \\
 B_{23} &= \frac{i\sqrt{2}}{8}x_5x_4 + \frac{i\sqrt{2}}{8}x_5^{-1}x_4 - \frac{i\sqrt{2}}{4}x_5 + \frac{i\sqrt{2}}{4}x_5^{-1} + \frac{i\sqrt{2}}{8}x_5x_4^{-1} + \frac{i\sqrt{2}}{8}x_5^{-1}x_4^{-1}; \\
 B_{24} &= \left( -\frac{\sqrt{2}}{8}x_5x_4 + \frac{\sqrt{2}}{8}x_5^{-1}x_4 + \frac{\sqrt{2}}{4}x_5 + \frac{\sqrt{2}}{4}x_5^{-1} - \frac{\sqrt{2}}{8}x_5x_4^{-1} + \frac{\sqrt{2}}{8}x_5^{-1}x_4^{-1} \right) l_{IJ}; \\
 B_{31} &= -\frac{\sqrt{2}}{8}x_6x_5x_4 - \frac{i\sqrt{2}}{4}x_6x_4 + \frac{\sqrt{2}}{8}x_6x_5^{-1}x_4 + \frac{\sqrt{2}}{8}x_6x_5x_4^{-1} - \frac{i\sqrt{2}}{4}x_6x_4^{-1} - \frac{\sqrt{2}}{8}x_6x_5^{-1}x_4^{-1}; \\
 B_{32} &= -\frac{\sqrt{2}}{8}x_6^{-1}x_5x_4 + \frac{i\sqrt{2}}{4}x_6^{-1}x_4 + \frac{\sqrt{2}}{8}x_6^{-1}x_5^{-1}x_4 + \frac{\sqrt{2}}{8}x_6^{-1}x_5x_4^{-1} + \frac{i\sqrt{2}}{4}x_6^{-1}x_4^{-1} - \frac{\sqrt{2}}{8}x_6^{-1}x_5^{-1}x_4^{-1}; \\
 B_{33} &= \frac{1}{4}x_5x_4 + \frac{1}{4}x_5^{-1}x_4 - \frac{1}{4}x_5x_4^{-1} - \frac{1}{4}x_5^{-1}x_4^{-1};
 \end{aligned}$$

$$\mathbf{B}_{34} = \left( -\frac{1}{4}x_5x_4 + \frac{1}{4}x_5^{-1}x_4 + \frac{1}{4}x_5x_4^{-1} - \frac{1}{4}x_5^{-1}x_4^{-1} \right) l_{II},$$

where  $c_1$  is a function related to the structural parameters of the robot, which is known and can be regarded as a constant.

## References

1. Taylor, R.H. A Perspective on Medical Robotics. *Proc. IEEE* **2006**, *94*, 1652–1664. [[CrossRef](#)]
2. Turchetti, G.; Palla, I.; Pierotti, F.; Cuschieri, A. Economic evaluation of Da Vinci-assisted robotic surgery: A systematic review. *Surg. Endosc.* **2012**, *26*, 598. [[CrossRef](#)] [[PubMed](#)]
3. Pugin, F.; Bucher, P.; Morel, P.; Bucher, P.; Morel, P. History of robotic surgery: From AESOP and ZEUS to Da Vinci. *J. Visc. Surg.* **2011**, *148*, e3–e8. [[CrossRef](#)] [[PubMed](#)]
4. Ishikawa, N.; Watanabe, G. Robot-assisted cardiac surgery. *Ann. Thorac. Cardiovasc. Surg.* **2015**, *21*, 322. [[CrossRef](#)] [[PubMed](#)]
5. Taylor, R.H.; Funda, J.; Grossman, D.D.; Karidis, J.P.; LaRose, D.A. Remote center-of-motion robot for surgery. U.S. Patent 5397323 A, 14 March 1995.
6. Zhou, X.; Zhang, H.; Feng, M.; Zhao, J.; Fu, Y. New remote centre of motion mechanism for robot-assisted minimally invasive surgery. *Biomed. Eng. Online* **2018**, *17*, 170. [[CrossRef](#)] [[PubMed](#)]
7. Ozgoren, M.K. Topological analysis of 6-joint serial manipulators and their inverse kinematic solutions. *Mech. Mach. Theory* **2002**, *37*, 511–547. [[CrossRef](#)]
8. Balkan, T.; Ozgoren, M.K.; Arikan, M.A.S.; Baykurt, H.M. A kinematic structure-based classification and compact kinematic equations for six-dof industrial robotic manipulators. *Mech. Mach. Theory* **2001**, *36*, 817–832. [[CrossRef](#)]
9. Balkan, T.; Ozgoren, M.K.; Arikan, M.A.S.; Baykurt, H.M. A method of inverse kinematics solution including singular and multiple configurations for a class of robotic manipulators. *Mech. Mach. Theory* **2000**, *35*, 1221–1237. [[CrossRef](#)]
10. Vol, N. Structural kinematics of 6-revolute-axis robot manipulators. *Mech. Mach. Theory* **1996**, *31*, 647–658.
11. Pieper, D.L. *The Kinematics of Manipulators under Computer Control*; Stanford University: Stanford, CA, USA, 1968.
12. Ai, Y.; Pan, B.; Niu, G.; Fu, Y.; Wang, S. Master-slave control technology of isomeric surgical robot for minimally invasive surgery. In Proceedings of the IEEE International Conference on Robotics and Biomimetics, Qingdao, China, 3–7 December 2017; pp. 2134–2139.
13. Podsedkowski, L. Forward and Inverse Kinematics of the Cardio-Surgical Robot with Non-Coincident Axis of the Wrist. *IFAC Proc. Vol.* **2003**, *36*, 437–442. [[CrossRef](#)]
14. Zhang, X.; Nelson, C.A. Kinematic Analysis and Optimization of a Novel Robot for Surgical Tool Manipulation. *J. Med. Devices* **2008**, *2*, 737–745. [[CrossRef](#)]
15. Mezouar, Y. Kinematic modeling and control of a robot arm using unit dual quaternions. *Robot. Auton. Syst.* **2016**, *77*, 66–73.
16. Xie, Q.; Pan, B.; Fu, Y.; Wang, S. Master-slave control technology research based on abdominal minimally invasive surgery robot. *Robot* **2011**, *33*, 53–58. [[CrossRef](#)]
17. Fu, Y.; Li, H.; Xie, Q. Master-slave control technology research for abdominal minimally invasive surgery robot. In *ASME 2010 International Mechanical Engineering Congress and Exposition*; American Society of Mechanical Engineers: New York, NY, USA, 2010; pp. 53–58.
18. Kajita, S. *Humanoid Robots*; Tsinghua University Press: Beijing, China, 2007.
19. Bu, W.; Liu, Z.; Tan, J. Inverse displacement analysis of 6R robots with offset wrists based on decoupling degrees of freedom at the cutoff points. *J. Mech. Eng.* **2010**, *46*, 1–5. [[CrossRef](#)]
20. Koker, R.; Oz, C.; Cakar, T.; Ekiz, H. A study of neural network based inverse kinematics solution for a three-joint robot. *Robot. Auton. Syst.* **2004**, *49*, 227–234. [[CrossRef](#)]
21. Lin, Y.; Zhao, H.; Ding, H. Solution of inverse kinematics for general robot manipulators based on multiple population genetic algorithm. *J. Mech. Eng.* **2017**, *53*, 1. [[CrossRef](#)]
22. Birbil, S.I.; Fang, S.C. An Electromagnetism-like Mechanism for Global Optimization. *J. Glob. Optim.* **2003**, *25*, 263–282. [[CrossRef](#)]
23. Birbil, S.I.; Fang, S.C.; Sheu, R.L. On the Convergence of a Population-Based Global Optimization Algorithm. *J. Glob. Optim.* **2004**, *30*, 301–318. [[CrossRef](#)]

24. Ren, Z.; Wang, Z.; Li, J.; Sun, L. Inverse Kinematics Solution for Robot Manipulator Based on Hybrid Electromagnetism-like Mechanism Algorithm. *J. Mech. Eng.* **2012**, *48*, 21–28. [[CrossRef](#)]
25. Yin, B.; Yan, B.; Shuai, J.; Ren, W.-B. Solution for direct kinematics of 3-PRS parallel manipulator using sylvester dialytic elimination method. In Proceedings of the International Conference on Mechanics and Civil Engineering, Orlando, FL, USA, 23–25 June 2014.
26. Roger, A.; Horn, R. *Matrix Analysis: English*; People Post Press: Beijing, China, 2015.
27. Taylor, R.J. *An Introduction to Error Analysis*; University Science Books: New York, NY, USA, 1997.
28. Hayashibe, M.; Suzuki, N.; Hashizume, M.; Konishi, K.; Hattori, A. Robotic surgery setup simulation with the integration of inverse-kinematics computation and medical imaging. *Comput. Methods Programs Biomed.* **2006**, *83*, 63–72. [[CrossRef](#)] [[PubMed](#)]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).