An Innovative Inverse Kinematics for Bionic Robot Arms based on Directional Tangent Matrix

Ju Yiran[1,*]
[1]Department of Electrical and Computer Engineering, University of British Columbia,
V6T 1Z4 Vancouver, Canada
Email: juyiran@alumni.ubc.ca

Meng Yu[2]
[2]College of Astronautics, Nanjing University of Aeronautics and Astronautics
Nanjing, China
Email：yuxy21@nuaa.edu.cn

***Abstract*— In this paper, we present an innovative inverse kinematics (IK) for Bionic Robot Arms using what we call Directional Tangent Matrix (DTM). We start by revisiting the concept of fixed-axis rotation, introducing the DTM as an alternative to the well-known direction cosine matrix (DCM). Then, the DTM is used for creating kinematic equations of general robot arms, simplifying these equations into a more manageable form: a second-order polynomial with four unknowns. We focus on solving these equations for robot arms with roughly parallel Joint Axes#3#4#5. The solutions we find are then applied in real-time to the actual robot arms using polynomial continuation method. Our approach offers a new way to develop and control Bionic Robot Arms, which could be more efficient and flexible compared to existing methods.**

Keywords*— *Directional Tangent Matrix, polynomial equations, Bionic Robot Arms, continuation method, Inverse Kinematics

I. INTRODUCTION

With the evolution of modern industrial and humanoid robot technologies, the role of robot arms has become indispensable. Ensuring precision in their movements necessitates intricate mathematical modeling, where Inverse Kinematics (IK) emerges as a cornerstone technology. Broadly, IK solutions can be categorized into numerical and analytical approaches. For robots equipped with up to 6 Revolute Joints (6R), the analytical approach is predominantly preferred. One reason the industry prefers decoupled robot arms with strictly-intersecting Joint Axes, specifically Axes#4#5 and #6, is their capability to generate analytical IK solutions [1,2]. However, these robot arms, when compared to human limbs, face limitations such as less practical joint forces, lower payload-weight ratios, and reduced energy efficiency. These challenges not only increase manufacturing complexity but also impact the accuracy and efficiency of the systems [3]. Addressing these issues is vital, especially in the study and development of bionic robot arms, where mimicking human arm functionality is essential [4]. Our research focuses on this area, exploring innovative IK solutions for general robot arms to enhance their practicality and efficiency in bionic applications.

For decoupled robot arms with explicit IK solutions, their position and attitude equations are fundamentally systems of 3-variable 2nd-order and 3-variable 1st-order polynomials based on tangent half-angle. Notably, the decoupling of position and orientation is achieved using a spherical wrist joint. However, the Denavit-Hartenberg (DH) method of IK within the algebraic domain is heavily influenced by the characteristics of these equations. Issues such as zero division often result in singularity problems [5-8]. This method also poses challenges when trying to generalize to standard robot arms. On examining numerical IK methods, approaches like Steepest Coordinate Descent and Modified DFP [9] are notable. While these methods can often yield a set of solutions, retrieving all possible solutions can be time-consuming [10-12].

Since manual elimination is both cumbersome and error-prone, and symbolic elimination systems often lack efficiency, there is a need to combine the advantages of the two to establish the principle of automatic elimination and solution in the specialized field of robotics. On the one hand, iterative equations are established to ensure real-time computation; On the other hand, only polynomial equations of the lowest power, number of equations, and number of independent variables can be eliminated and solved automatically and reduce the singularities and uncertainties in numerical method. Automatic modelling and solving are essential features of bionic robots.

Morgan, Alexander P.A presented a Method for computing all solutions to Systems of Polynomials Equations [13]. And they developed SYMPOL library. L.-W.Tsai reduced the 6R IK for general robot arms to that of solving a system of eight second-degree equations in eight unknowns, it is further demonstrated that is second-degree system can be routinely solved using a continuation algorithm based on SYMPOL and SYMMAN libraries[14]. He also presented the new robot arms with Joint Axes#2#3#4 strictly-parallel to each other. Charles Wampler and Alexander Morgan presented an efficient numerical method for finding all 16 solutions, based on coefficient-parameter polynomial continuation [15]. However, Dinesh Manocha and John Francis Canny[16,17] claimed to be able to get all the real solutions in about 11 milliseconds. However, the examples in the literature are not consistent with general purpose robotic arms. Other researchers have explored more improvements, but so far they have not been able to develop Robot arms structured like human arms [18-31].

6R IK is essentially the problem of solving a multivariate

polynomial system including Gröbner Basis [32,33] and Polynomial Resultants[34-38]. The Dixon matrix is dense since its entries are determinants of the coefficients of the polynomials in the original system. It has the advantage of being an order of magnitude smaller in comparison to a dialytic matrix. Therefore, applying Dixon-resultant based elimination principle to 6R inverse solutions has received increasing attention.

Our research group has observed that the 6R position and attitude equations can be equated to a multivariate vector polynomial system. This system is not only a vector system that employs basic operations like dot and cross products, but it also functions as an algebraic ring, incorporating addition, subtraction, and multiplication. Significant progress has been reported in the literature [39,40] in utilizing vector properties to solve multivariate polynomial systems.

Traditionally, expressing position and attitude equations using sine cosine and then transforming them into tangent half-angle forms has been time-consuming in terms of symbol processing and ineffective in unveiling internal laws. To address this, our paper introduces the Directional Tangent matrix (DTM) in terms of tangent half angular to establish an innovative Vector Polynomial System. We then apply this system to accurately and efficiently solve the inverse kinematics of general robot arms, particularly those with roughly-parallel Joint Axes #3, #4, and #5, in real time.

II. NOTATIONS

The specification of the structural parameters of the robot arm is shown in Fig.1. The **r**adial position coordinate vector from the **origin** of Frame#i to a point P attached to Frame#6 is noted as ${}^{i}r_{6\mathrm{P}}$. The joint axis coordinate vector i.e., the direction vectors, between Frame#5 and Frame#6 are expressed as ${}^{5}n_{6}$, which are aligned with the **d**esired position ${}^{i}_{d}r_{6\mathrm{P}}$ and desired direction ${}^{i|5}_{d}n_{6}$, respectively by controlling the joint positions. ${}^{i}\boldsymbol{R}_{6}$ indicates that the kinematic chain is composed of 6 **R**evolute joint axes.

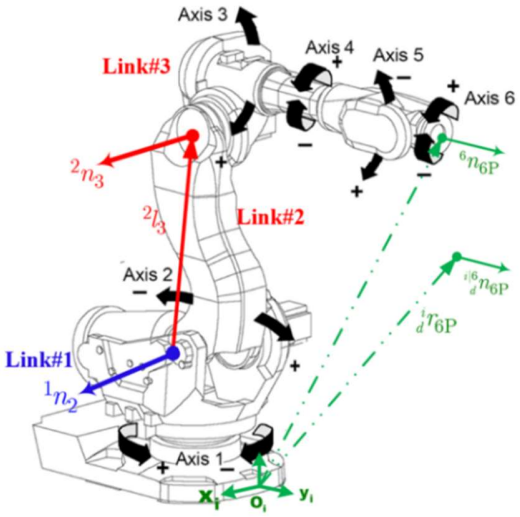

Fig. 1. Robot Arm Symbol Specification

The basic notation is as follows:

1) Lowercase letters with upper-left and lower-right corners indicate coordinate vectors, where: the upper-left indicator indicates the reference system, and for position vectors also the origin of the reference system.
2) Capital letters with upper-left and lower-right corners indicate coordinate matrices.
3) Lower case letters with upper and lower right corner labels stand for scalars.
4) Any coordinate vector defaults to a column vector.
5) The predecessor of k is written as $\overline{k}$, and predecessor of $\overline{k}$ is written as $\overline{\overline{k}}$.
6) The ${}^{\overline{k}}l_{k}$ stands for constant **l**ocation of Link#k with respect to Link#$\overline{k}$.

III. RECONCEPTUALIZING FIXED-AXIS ROTATION

Given a joint ${}^{\overline{l}}\mathrm{JT}_{l}$ between parent Link#$\overline{l}$ and child Link #l, its angular position by unit axis vector ${}^{\overline{l}}n_{l}$ is noted as $\phi_{l}^{\overline{l}}$. Clearly

$$\phi_{l}^{\overline{l}} = -\phi_{\overline{l}}^{l} \tag{1}$$

As shown in Fig. 2, the 1st-order screw of ${}^{l}r_{lS}$ is ${}^{\overline{l}}\tilde{n}_{l} \cdot {}^{l}r_{lS}$, and the 2nd-order screw of ${}^{l}r_{lS}$ is ${}^{\overline{l}}\tilde{n}_{l}^{\wedge 2} \cdot {}^{l}r_{lS}$, where $\wedge$ stands for power notation or separator.

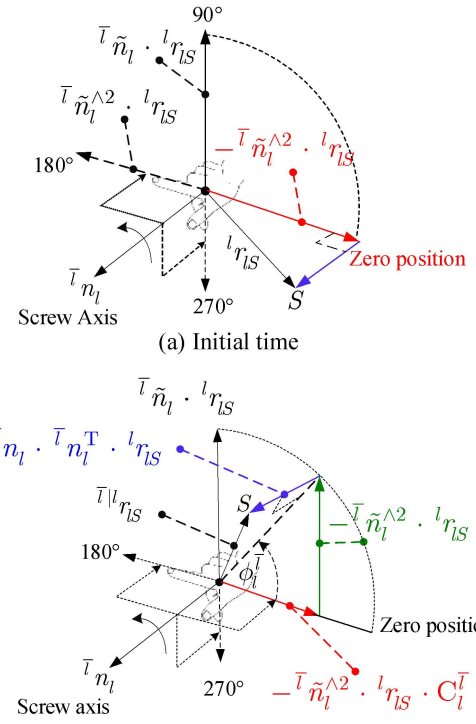

Fig. 2. Fixed axis rotation

Obviously

$$ {}^{l}r_{lS} - {}^{\overline{l}}\tilde{n}_l^{\wedge 2} \cdot {}^{l}r_{lS} = {}^{\overline{l}}n_l \cdot {}^{\overline{l}}n_l^{\mathrm{T}} \cdot {}^{l}r_{lS} \tag{2} $$

From Eq.(2)

$$ -{}^{\overline{l}}\tilde{n}_l^{\wedge 2} = \mathbf{1} - {}^{\overline{l}}n_l \cdot {}^{\overline{l}}n_l^{\mathrm{T}} \tag{3} $$

Where **1** stands for unit matrix with size 3 by 3. The axis vector and its screw matrix perform orthogonal projections. Let

$$ \mathrm{C}_l^{\overline{l}} = \cos(\phi_l^{\overline{l}}),\ \mathrm{S}_l^{\overline{l}} = \sin(\phi_l^{\overline{l}}). $$

The vector ${}^{l}r_{lS}$ after rotation by $\phi_l^{\overline{l}}$ with respect to axis vector ${}^{\overline{l}}n_l$, the ${}^{l}r_{lS}$ is **orthogonally decomposed** into its axial, 1st order radial and zero-position components. So, we get

$$ \begin{gathered} {}^{\overline{l}}Q_l \cdot {}^{l}r_{lS} = \\ \left({}^{\overline{l}}n_l \cdot {}^{\overline{l}}n_l^{\mathrm{T}} + {}^{\overline{l}}\tilde{n}_l \cdot \mathrm{S}_l^{\overline{l}} - {}^{\overline{l}}\tilde{n}_l^{\wedge 2} \cdot \mathrm{C}_l^{\overline{l}} \right) \cdot {}^{l}r_{lS} \end{gathered} \tag{4} $$

Where

$$ {}^{\overline{l}|l}r_{lS} \triangleq {}^{\overline{l}}Q_l \cdot {}^{l}r_{lS} \tag{5} $$

Where the bar | denotes orthogonal projection. The DCM ${}^{\overline{l}}Q_l$ stands for rotation from right frame to left frame, also for corresponds to orthogonal projection of left frame to right frame. Because ${}^{l}r_{lS}$ is an arbitrary vector, we obtain the following Rodrigues' rotation formula

$$ {}^{\overline{l}}Q_l = {}^{\overline{l}}n_l \cdot {}^{\overline{l}}n_l^{\mathrm{T}} + {}^{\overline{l}}\tilde{n}_l \cdot \mathrm{S}_l^{\overline{l}} - {}^{\overline{l}}\tilde{n}_l^{\wedge 2} \cdot \mathrm{C}_l^{\overline{l}} \tag{6} $$

Considering Eq.(6) and (3), obtain the DCM in terms of natural axis and angular position

$$ {}^{\overline{l}}Q_l = \mathbf{1} + {}^{\overline{l}}\tilde{n}_l \cdot \mathrm{S}_l^{\overline{l}} + {}^{\overline{l}}\tilde{n}_l^{\wedge 2} \cdot \left(1 - \mathrm{C}_l^{\overline{l}}\right) \tag{7} $$

Its inverse matrix is obtained by exchanging the upper left and lower right corner labels at the same time. Using Eq.(7), we have

$$ \begin{gathered} {}^{l}n_{\overline{l}} \triangleq {}^{l}Q_{\overline{l}} \cdot {}^{\overline{l}}n_l = \\ \left(\mathbf{1} - {}^{\overline{l}}\tilde{n}_l \cdot \mathrm{S}_l^{\overline{l}} + {}^{\overline{l}}\tilde{n}_l^{\wedge 2} \cdot \left(1 - \mathrm{C}_l^{\overline{l}}\right)\right) \cdot {}^{\overline{l}}n_l = 1 \cdot {}^{\overline{l}}n_l \end{gathered} $$

i.e.

$$ {}^{l}n_{\overline{l}} = {}^{\overline{l}}n_l \tag{8} $$

Eq.(8) says the axis vector, which has the same coordinate with respect to its adjacent frames, is an eigenvector and the scalar 1 is its eigenvalue. Returning to Fig 1, the ${}^{\overline{l}}n_l$ is the common unit axis vector between Frames# $\overline{l}$ and # l. This is the reason for the use of n to denote full-order co**n**nection. One keenly discovers that this zero-axis is none other than the real axis of quaternions. Given a rotational chain in Fig.1, its recursive position equation is shown as

$$ \begin{gathered} {}^{i}l_1 + {}^{0}Q_1 \cdot \left({}^{1}l_2 + {}^{1}Q_2 \cdot \left({}^{2}l_3 + {}^{2}Q_3 \cdot \left({}^{3}l_4 + \right.\right.\right. \\ \left.\left.\left. {}^{3}Q_4 \cdot \left({}^{4}l_5 + {}^{5}Q_6 \cdot {}^{5}l_6 \right) \right) \right) \right) = {}_{d}^{i}r_6 \end{gathered} \tag{9} $$

In Eq.(7), the motion of Link# l leads to its successors' motion. Also using Eq.(7), get

$$ {}^{\overline{l}}Q_l = \mathbf{1} \quad if \quad \phi_l^{\overline{l}} = 0 \tag{10} $$

Eq.(10) says that Frames# $\overline{l}$ and # l have the same orientation in zero position. So, a coordinate system that has the same orientation at zero position is referred to as a **Natural Coordinate System**, which is used through this paper.

IV. DTM AND SQUARE MODULUS

Considering Eq.(7), the Directional Tangent Matrices (DTM) is denoted by

$$ {}^{\overline{l}}\boldsymbol{Q}_l \triangleq \mathbf{1} + 2 \cdot \tau_l \cdot {}^{\overline{l}}\tilde{n}_l + \tau_l^{\wedge 2} \cdot {}^{\overline{l}}\mathrm{N}_l \tag{11} $$

where

$$ \tau_l \triangleq \tan\left(0.5 \cdot \phi_l^{\overline{l}}\right) \tag{12} $$

$$ \begin{aligned} \boldsymbol{\tau}_l^{\overline{l}} &\triangleq 1 + \tau_l^{\wedge 2} \geq 1 \\ \boldsymbol{\tau}_n^{i} &\triangleq \boldsymbol{\tau}_1^{i} \cdot \boldsymbol{\tau}_2^{1} \cdots \boldsymbol{\tau}_n^{\overline{n}} \end{aligned} \tag{13} $$

$$ {}^{\overline{l}}\mathrm{N}_l = \mathbf{1} + 2 \cdot {}^{\overline{l}}\tilde{n}_l^{\wedge 2} \tag{14} $$

From (7) and (12) to (14), it is obtained that

$$ \boldsymbol{\tau}_l^{\overline{l}} \cdot {}^{\overline{l}}Q_l = {}^{\overline{l}}\boldsymbol{Q}_l \tag{15} $$

$$ \begin{aligned} \boldsymbol{\tau}_n^{i} \cdot {}^{i}Q_n &= {}^{i}\boldsymbol{Q}_n \\ {}^{i}\boldsymbol{Q}_n \cdot {}^{i}\boldsymbol{Q}_n^{-1} &= {}^{i}\boldsymbol{Q}_n^{-1} \cdot {}^{i}\boldsymbol{Q}_n = \boldsymbol{\tau}_n^{i\wedge 2} \cdot \mathbf{1} \end{aligned} \tag{16} $$

From the transitivity of DCM ${}^{\overline{l}}Q_l$ and the **square modulus** $\boldsymbol{\tau}_l^{\overline{l}}$ **of DTM** ${}^{\overline{l}}\boldsymbol{Q}_l$, we have

$$ \begin{aligned} {}^{i}\boldsymbol{Q}_n &= {}^{i}\boldsymbol{Q}_1 \cdot {}^{1}\boldsymbol{Q}_2 \cdots {}^{\overline{n}}\boldsymbol{Q}_n \\ {}^{\overline{l}}\boldsymbol{Q}_l^{-1} &= \mathbf{1} - 2 \cdot \tau_l \cdot {}^{\overline{l}}\tilde{n}_l + \tau_l^{\wedge 2} \cdot {}^{\overline{l}}\mathrm{N}_l \end{aligned} \tag{17} $$

It can be seen from (16) that: ${}^{i}\boldsymbol{Q}_n$ and $\boldsymbol{\tau}_n^{i}$ are 2nd order polynomials in terms of τ_l, and from (15) that ${}^{\overline{l}}\boldsymbol{Q}_l$ and ${}^{\overline{l}}Q_l$ have similar structure. So it is called DTM. Using $\left\| {}^{\overline{l}}Q_l \right\| = 1$ and (15), obtain

$$ \boldsymbol{\tau}_l^{\overline{l}} = \left\| {}^{\overline{l}}\boldsymbol{Q}_l \right\| \tag{18} $$

The DTM and its square modulus are used to replace the DCM to build kinematic equations in terms of tangent half-angle.

V. IK THEORY OF GENERAL ROBOT ARMS

In the following, the inverse kinematics of 6R generic robot arms is expressed in Theorem 1. Then, we introduce its solving procedure.

A. Vectors polynomial equations of general robot arms

Theorem 1 Given a 6R serial chain, the system i coincides with the origin of axis #1; axes #3#4#5 are approximately parallel, axes #5#6 are neither parallel nor orthogonal; the tool central point (TCP) is at the origin of axis #6. The 4-variable 2nd-order vector polynomial system taken the desired position ${}_{d}^{i}r_6$ and desired direction ${}_{d}^{i|5}n_6$ as inputs is

$$\begin{cases} \boldsymbol{f}_v = \boldsymbol{\tau}_1^i \cdot {}^4\boldsymbol{Q}_1 \cdot {}^1l_2 + \boldsymbol{\tau}_2^i \cdot {}^4\boldsymbol{Q}_2 \cdot {}^2l_3 + \boldsymbol{\tau}_3^i \cdot {}^4\boldsymbol{Q}_3 \cdot {}^3l_4 + \\ \backslash \boldsymbol{\tau}_4^i \cdot {}^4l_5 + {}_3\mathrm{R}_3 \cdot {}_3\mathrm{N}_3 \cdot {}^4\boldsymbol{Q}_i \cdot {}_d^{i|5}n_6 - {}^4\boldsymbol{Q}_i \cdot {}_d^i r_6 = 0_3 \\ \boldsymbol{f}_s = \left({}_3\mathrm{N}_3^{[1][*]} + {}_3\mathrm{N}_3^{[3][*]} \right) \cdot {}^4\boldsymbol{Q}_i \cdot {}_d^{i|5}n_6 - \boldsymbol{\tau}_4^i = 0 \end{cases} \quad (19)$$

wherein

$${}_3\mathrm{N}_3^{-1} = \begin{bmatrix} {}^5n_6 & 2 \cdot {}^4\tilde{n}_5 \cdot {}^5n_6 & {}^4\mathrm{N}_5 \cdot {}^5n_6 \end{bmatrix} \quad (20)$$

$${}_3\mathrm{R}_3 = \begin{bmatrix} {}^5l_6 & 2 \cdot {}^4\tilde{n}_5 \cdot {}^5l_6 & {}^4\mathrm{N}_5 \cdot {}^5l_6 \end{bmatrix} \quad (21)$$

Applying Ju-Dixon multi-linear projection, the joint angles of axes #1#2#3#4 are computed sequentially. The resultant has an order of 16 by first solving for axis #1. After obtaining the IK solutions for axes #1#2#3#4, we get from

$$\left(1 + \tau_5^{\wedge 2}\right) \cdot {}_3\mathrm{N}_3^{[2][*]} \cdot {}^4\boldsymbol{Q}_i \cdot {}_d^{i|5}n_6 = \boldsymbol{\tau}_4^i \cdot \tau_5 \quad (22)$$

that axis #5 has at most 2 groups of physical solutions.

Proof. The direction alignment equation for the 6R chain is

$$\boldsymbol{\tau}_5^4 \cdot {}^4\boldsymbol{Q}_i \cdot {}_d^{i|5}n_6 = \boldsymbol{\tau}_4^i \cdot {}^4\boldsymbol{Q}_5 \cdot {}^5n_6 \quad (23)$$

We get from (11) that

$${}^4\boldsymbol{Q}_5 = \mathbf{1} + 2 \cdot \tau_5 \cdot {}^4\tilde{n}_5 + \tau_5^{\wedge 2} \cdot {}^4\mathrm{N}_5$$

and according to (20) and (23), it is obtained

$$\boldsymbol{\tau}_5^4 \cdot {}^4\boldsymbol{Q}_i \cdot {}_d^{i|5}n_6 = \boldsymbol{\tau}_4^i \cdot {}_3\mathrm{N}_3^{\wedge -1} \cdot \begin{bmatrix} 1 \\ \tau_5 \\ \tau_5^{\wedge 2} \end{bmatrix} \quad (24)$$

When the axes #4#5 are not parallel, we get from (24) that

$$\boldsymbol{\tau}_4^i \cdot \begin{bmatrix} 1 \\ \tau_5 \\ \tau_5^{\wedge 2} \end{bmatrix} = \boldsymbol{\tau}_5^4 \cdot {}_3\mathrm{N}_3 \cdot {}^4\boldsymbol{Q}_i \cdot {}_d^{i|5}n_6 \quad (25)$$

Then, the position equation for the 6R serial chain is

$$\boldsymbol{\tau}_5^4 \cdot \begin{pmatrix} \boldsymbol{\tau}_1^i \cdot {}^4\boldsymbol{Q}_1 \cdot {}^1l_2 + \boldsymbol{\tau}_2^i \cdot {}^4\boldsymbol{Q}_2 \cdot {}^2l_3 + \boldsymbol{\tau}_3^i \\ \backslash \cdot {}^4\boldsymbol{Q}_3 \cdot {}^3l_4 + \boldsymbol{\tau}_4^i \cdot {}^4l_5 - {}^4\boldsymbol{Q}_i \cdot {}_d^i r_6 \end{pmatrix} + \boldsymbol{\tau}_4^i \cdot {}^4\boldsymbol{Q}_5 \cdot {}^5l_6 = 0_3 \quad (26)$$

Obviously, Equation (26) is equivalent to (9). It is obtained from (21) and (26) that

$$\boldsymbol{\tau}_5^4 \cdot \begin{pmatrix} \boldsymbol{\tau}_1^i \cdot {}^4\boldsymbol{Q}_1 \cdot {}^1l_2 + \boldsymbol{\tau}_2^i \cdot {}^4\boldsymbol{Q}_2 \cdot {}^2l_3 + \boldsymbol{\tau}_3^i \\ \backslash \cdot {}^4\boldsymbol{Q}_3 \cdot {}^3l_4 + \boldsymbol{\tau}_4^i \cdot {}^4l_5 - {}^4\boldsymbol{Q}_i \cdot {}_d^i r_6 \end{pmatrix} + \boldsymbol{\tau}_4^i \cdot {}_3\mathrm{R}_3 \cdot \begin{bmatrix} 1 \\ \tau_5 \\ \tau_5^{\wedge 2} \end{bmatrix} = 0_3 \quad (27)$$

Since axes #5#6 are neither parallel nor orthogonal, the inverse of ${}_3\mathrm{N}_3$ matrix exists. We get from (25) and (27) that

$$\begin{aligned} &\boldsymbol{\tau}_1^i \cdot {}^4\boldsymbol{Q}_1 \cdot {}^1l_2 + \boldsymbol{\tau}_2^i \cdot {}^4\boldsymbol{Q}_2 \cdot {}^2l_3 + \boldsymbol{\tau}_3^i \cdot {}^4\boldsymbol{Q}_3 \cdot {}^3l_4 + \boldsymbol{\tau}_4^i \\ &\backslash \cdot {}^4l_5 + {}_3\mathrm{R}_3 \cdot {}_3\mathrm{N}_3 \cdot {}^4\boldsymbol{Q}_i \cdot {}_d^{i|5}n_6 - {}^4\boldsymbol{Q}_i \cdot {}_d^i r_6 = 0_3 \end{aligned} \quad (28)$$

From the first and last sub-equations of (25), we have

$$\left({}_3\mathrm{N}_3^{[1][*]} + {}_3\mathrm{N}_3^{[3][*]} \right) \cdot {}^4\boldsymbol{Q}_i \cdot {}_d^{i|5}n_6 = \boldsymbol{\tau}_4^i \quad (29)$$

Equation (19) is obtained from (28) and (29). Equation (22) is obtained from the 2nd sub-equation of (25). □

Obviously, the above process of eliminating variables is very concise, thanks to the vector property of the DTM.

B. Solving procedure of general robot arms

Since Equation (19) is a vector polynomial system, Ju-Dixon multilinear projection [39,40] is applied when axes #3#4#5 are parallel in the nominal model. Its Dixon matrix has a size of 2 by 2 and its each term is up to 8th order. So, the Dixon resultant is up to 16th order. After solving for Axes #1#2#3#4, the natural continuation method is applied to the engineering model to obtain physical solutions of Axes#1#2#3#4. The solving procedure of general robot arms is shown in Fig.3.

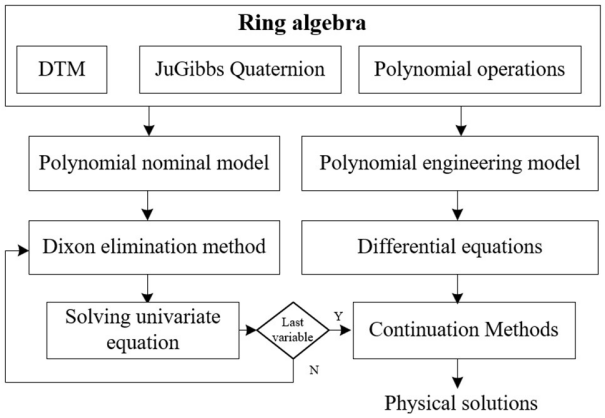

Fig. 3. Solving procedure

We implemented the described system using C++. Initially, we constructed the ring algebra system, and incorporated both the Dixon method and Continuation methods [39,40]. Users can input a Polynomial engineering model along with the desired position and direction, and the system then computes the corresponding nominal model and differential equations. Following this, the system automatically performs variable elimination and solves the nominal model. Finally, all nominal solutions are input into the continuation procedure, which generates all physical solutions.

VI. VERIFICATION OF METHOD

In the following, the validity of the above theory is demonstrated by means of several examples.

The structural parameters and IK solutions of a robot arm with axes #3#4#5 to be roughly parallel are shown in Tables Ⅰ and Ⅱ, respectively.

TABLE I. STRUCTURAL PARAMETERS FOR A 6R ROBOT ARM WITH ROUGHLY PARALLEL AXES #3#4#5

Axes	#1	#2	#3	#4	#5	#6
Axis vectors	[0.0, 0.0, 1.0]	[0.0, 1.0, 0.0]	[1.0, 0.0, 0.15]	[1.0, 0.0, 0.0]	[1.0, 0.15, 0.0]	[0.0, 1.0, 0.5]
Position vector (m)	[0.0, 0.0, 0.0]	[0.01, 0.05, 0.22]	[0.01, 0.04, 0.24]	[0.01, 0.06, 0.36]	[0.03, 0.06, 0.36]	[0.01, 0.01, 0.01]

TABLE II. IK SOLUTION EXAMPLES FOR THE ROBOT ARM WITH AXES #3#4#5 TO BE ROUGHLY PARALLEL

Desired position (m)	Desired direction	Angular positions of 6 axes (Deg)
2.43015 . [0.18946, -0.02663, 0.03534]	[-0.16842, -0.94419, -0.28305]	☆1: [-0.1782053, 101.8651081, 111.9273177, -136.8896409, -171.2033614, 7.0735059]; ☆2: [-144.6343787, 104.4909581, -118.1825457, -60.9844176, -161.4736061, 3.5746358]; ☆3: [58.3803193, 95.0873317, 150.9691157, -88.5832630, 160.5837883, -10.1537574]; ☆4: [-103.1141489, 123.6651040, -165.7925552, 103.4453048, 123.5388822, -7.6812146]; ☆5: [128.3036514, -125.7165148, 32.1125140, 106.6490848, -120.6914967, -158.2472937]; ☆6: [-100.8310658, 99.3491449, -66.2866928, -99.1399549, -133.1482108, -3.88241439]; ☆7: [157.6924600, -110.4208581, 119.56012917, -129.1894917, 1.7425093, -143.7871729]; ☆8: [-25.0097288, -41.6597497, -139.5428542, -50.5666563, 5.1052064, 144.02837312]; ☆9: [66.0000000, 77.0000000, 77.0000000, 77.0000000, 76.9999999, 0.0000000];

Our tests have demonstrated that: 1) the inverse kinematics (IK) problem has a maximum of 8 sets of solutions; 2) On a laptop computer with a main frequency of 2.8 GHz, the computation time for the IK, when using a single thread, is approximately 5.8 milliseconds with 14-digit accuracy. Furthermore, this computation time can be reduced to about 1.7 milliseconds by utilizing 4 threads.

VII. DEVELOPMENT OF BIONIC ROBOT ARMS

The real-time inverse kinematics (IK) method for robot arms with roughly parallel axes#3#4#5 discussed previously forms the cornerstone for developing the bionic robot arms shown in Fig.4.

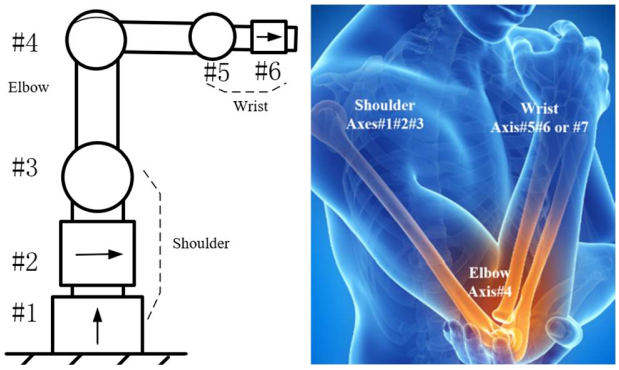

Fig. 4. Bionic Robot Arms

A key distinction between these bionic robot arms and traditional robotic arms lies in the shoulder design at the base, which incorporates three joints and the elbow which has a carrying angle about 5 to 15 degrees. This unique feature enables the bionic robot arm to maintain a lower center of gravity, thereby improving its load-to-weight ratio, energy efficiency and working space. Additionally, the design benefits absolute accuracy, as the rear three axes are not constrained by a common point. This aspect simplifies the manufacturing and assembly processes. A new model of the bionic robot arm has been developed based on these principles. Initial results confirm these advantages, and further detailed quantification is ongoing.

VIII. CONCLUSION

In this paper, we present an innovative inverse kinematics (IK) approach for Bionic Robot Arms, utilizing the Directional Tangent Matrix (DTM). Our methodology began with a redefinition of fixed-axis rotation, exploring the DTM as an alternative to the classical Direction Cosine Matrix (DCM). We then employed the DTM to construct kinematic equations for general robot arms, simplifying these equations to a second-order polynomial with four unknowns. We addressed the nominal equation involving parallel Joint Axes #3, #4, and #5, using their solutions to derive real-time physical solutions for the roughly parallel model through the polynomial continuation method. This IK approach has been successfully applied in the development of Bionic Robot Arms, demonstrating its validity and potential for advanced robotic applications.

REFERENCES

[1] Seong Hyeon Kim, Eunseok Nam, et al. Robotic Machining: A Review of Recent Progress. International Journal of Precision Engineering and Manufacturing volume 20, pages1629–1642 (2019)

[2] Jun Fujimori, Ryota Ienaka, et al. Precision-machining robot system. Kawasaki Technical Review No.172 December 2012

[3] Jadran Lenarcic and Michael Stanisic. A Humanoid Shoulder Complex and the Humeral Pointing Kinematics. IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, VOL. 19, NO. 3, JUNE 2003

[4] Jorge Angeles. Fundamentals of Robotic Mechanical Systems: Theory Methods and Algorithms[M]. Springer, 2014.

[5] F. Merat. Introduction to robotics: mechanics and control, IEEE J. Robot. Autom. 3 (2) (1987) 166 166.

[6] S. Sarabandi, A. Shabani, et al. On closed-form formulas for the 3-d nearest rotation matrix problem, IEEE Trans. Robot. 36 (4) (2020) 1333–1339.

[7] J.M. Selig. Geometric Fundamentals of Robotics, Springer Science & Business Media, 2004.

[8] S. Kucuk, Z. Bingul. Inverse kinematics solutions for industrial robot manipulators with offset wrists, Appl. Math. Model. 38 (7) (2014) 1983–1999.

[9] Saha, Ankan, Tewari, Ambuj. On the Finite Time Convergence of Cyclic Coordinate Descent Methods. https://ui.adsabs.harvard.edu/abs/2010arXiv1005.2146S

[10] Zi-wu REN, Zhen-hua WANG, et al. A hybrid biogeography-based optimization method for the inverse kinematics problem

of an 8-DOF redundant humanoid manipulator. Front Inform Technol Electron Eng 2015 16(7):607-616

[11] Seong Hyeon Kim, Eunseok Nam, et al. Robotic Machining: A Review of Recent Progress. International Journal of Precision Engineering and Manufacturing (2019) 20:1629–1642

[12] J. Demby's, Y. Gao, et al. A study on solving the inverse kinematics of serial robots using artificial neural network and fuzzy neural network, in: Proceedings of the IEEE International Conference on Fuzzy Systems, 2019, pp. 1–6.

[13] Morgan, Alexander P. A Method for Computing All Solutions to Systems of Polynomials Equations. ACM Transactions on Mathematical Software (TOMS) , Volume 9 (1) , Mar 1, 1983

[14] L.-W. Tsa, A. P Morga. Solving the Kinematics of the Most General Six and Five-Degree of Freedom Manipulators by Continuation Methods. Journal of Mechanisms, Transmissions, and Automation in Design. JUNE 1985, Vol. 107/189

[15] Charles Wampler and Alexander Morgan. solving the 6R inverse position problem using a generic-case solution methodology. Mech. Math. theol. Vol. 26, No. I. pp. 91-106, 1991

[16] Dinesh Manocha, John Francis Canny. Multipolynomial Resultants and Linear Algebra. ISSAC '92: Papers from the international symposium on Symbolic and algebraic computation

[17] Dinesh Manocha, John Francis Canny. Efficient Inverse Kinematics for General 6R Manipulators[J]. March 1999, IEEE Transactions on Robotics and Automation.

[18] M. Raghavan, B. Roth. Kinematic analysis of the 6R manipulator of general geometry. Proceedings of the International Symposium on Robotics, 1989, pp. 314–320.

[19] M. Raghavan, B. Roth. Inverse kinematics of the general 6R manipulator and related linkages, Trans. ASME J. Mech. Des. 115 (1993) 502–508.

[20] QIAO S, LIAO Q, et al. Inverse kinematic analysis of the general 6R serial manipulators based on double quaternions[J]. Mechanism & Machine Theory, 2010, 45(2):193-199.

[21] ShuguangQiao, QizhengLiao, et al. Inverse kinematic analysis of the general 6R serial manipulators based on double quaternions. Mechanism and Machine Theory, Volume 45, Issue 2, February 2010, Pages 193-199.

[22] P. Beeson, B. Ames. TRAC-IK: An open-source library for improved solving of generic inverse kinematics, in: Proceedings of the IEEE-RAS 15th International Conference on Humanoid Robots, Seoul, Korea, 2015, pp. 928–935.

[23] W. Li, T. Howison, J. Angeles. On the use of the dual Euler–Rodrigues parameters in the numerical solution of the inverse-displacement problem, Mech. Mach. Theory 125 (2018) 21–33.

[24] F. Thomas. Approaching dual quaternions from matrix algebra, IEEE Trans. Robot. 30 (5) (2014) 1037–1048.

[25] M. Shoham. A note on Clifford's derivation of Bi-Quaternions, 10th World Congress Theory on Machine Mechanics, IFToMM, 1999, pp. 43–47.

[26] A. Müller. Screw and Lie group theory in multibody kinematics, Multibody Syst. Dyn. 43 (1) (2018) 37–70.

[27] S. Qiao, Q. Liao, S. Wei, et al. Inverse kinematic analysis of the general 6R serial manipulators based on double quaternions, Mech. Mach. Theory 45 (2) (2010) 193–199.

[28] J. Xu, Z. Liu, Q. Cheng, et al. Models for three new screw-based IK sub-problems using geometric descriptions and their applications, Appl. Math. Model. 67 (2019) 399–412.

[29] D.A. Cox, J. Little, D. O'shea. Using Algebraic Geometry, Springer Verlag, 2005.

[30] B. Sturmfels. Solving Systems of Polynomial Equations, American Mathematical Society, Providence, 2002.

[31] Joseph L. Awange, Béla Paláncz. Geospatial Algebraic: Theory and Applications [M]. ISBN 978-3-319-25465-4 (eBook). Springer-Verlag Berlin Heidelberg 2016. Pages 53-66.

[32] Uchida T, Mcphee J. Triangularizing kinematic constraint equations using Gröbner bases for real-time dynamic simulation[J]. Multibody System Dynamics, 2011, 25(3): p. 335-356.

[33] Buchberger B. Gröbner Bases and Systems Theory[J]. Multidimensional Systems & Signal Processing, 2001, 12(3):223-251.

[34] Nakos G, Williams R. Elimination with the Dixon resultant[J]. Mathematica in Education and Research, 1997, 6(3): 11-21.

[35] X. Qin, L. Zhang, et al. Heuristics to sift extraneous factors in Dixon resultants, J. Symb. Comput. 112 (2022) 105–121.

[36] B. Paláncz, P. Zaletnyik, J.L, et al. Dixon resultant's solution of systems of geodetic polynomial equations, J. Geod. 82 (8) (2008) 505–511.

[37] S. Zhao. Dixon resultant research and new algorithms, Ph.D. dissertation, Chin. Acad, Sci., Chengdu, China (2005).

[38] A.D. Chtcherba, D. Kapur. Constructing Sylvester-type resultant matrices using the Dixon formulation, J. Symb. Comput. 38 (1) (2004) 777–814.

[39] Feifei Chen, Hehua Ju. Applications of an improved Dixon elimination method for the inverse kinematics of 6R manipulators. Applied Mathematical Modelling 107 (2022) 764–781

[40] Feifei Chen, Hehua Ju, Xiaohan Liu. Mechanism and Machine Theory. Mechanism and Machine Theory 179 (2023) 105-118