Robotics: Heuristics Based Singularity Free Inverse Kinematics and Jacobian Inverse

Chaman Lal Sabharwal
Missouri University of Science and Technology
Rolla, Missouri-65409, USA

Abstract **--In Robotics, inversion is encountered at two levels: Inverse Kinematics level and Inverse Jacobian level. We present a new and intuitive approach to simplify the existing analytical complex approaches for inversion: Kinematics and Jacobian. Inverse Kinematics also employs geometric approach. We present a different style to efficient geometric approach. We show that specification of end-effector position is not necessary. Robotic Jacobian is an m×n matrix with n degrees of freedom (DOF). The computation of inverse velocities is an issue when m ≠ n, for which we provide an intuitive and common sense approach to the Generalized Inverse. Thus, this paper provides intuitive exciting approaches for efficient computing of inverse kinematics and inverse Jacobian in robotics.**

Keywords: Link parameters; inverse kinematics; inverse jacobian; generalized inverse; degrees of freedom (DOF)

I. INTRODUCTION

A robot is a rigid body compose of links and joints on links for motion, see Fig. 1; Fig. 2. Kinematics is the study of motion without regard to the forces that create it. The representation of the position of the robot end effecter through the Robotics Engineering (common parameters and link) forward kinematics [1], [2], [3]. Forward kinematics yields a unique orientation and position of end-effector, but the end-effector may be reached in multiple ways. There can be several paths for a robot to reach a destination. If the link parameters for joints are specified, forward kinematics uses these joint parameter values to reach a unique destination. Inverse kinematics of robotic manipulator involves obtaining the required values manipulator joint position given the desired end point and direction. There is no unique solution and close the form of direct expression of inverse kinematics mapping [2], [3], [4]. However, if the orientation and position of end-effector is specified, then Inverse Kinematics determines possible link parameter values that may result in multiple solutions, some of which may be extraneous and inconsistent. Inverse kinematics employs a bag of heuristics using one trigonometric equation and avoiding division by zero singularities. This analysis explores the multiple solutions and exploits the valid solutions. We will devise common sense approach.

For inverse kinematics in Geometric approach, to reach intermediate frames, it is computationally efficient to start from base origin [see Fig. 5, Fig. 6] instead of instead starting from end-effector. It is also intuitive to grasp. Here we provide a heuristic approach to solve the required equations. Also, Inverse kinematics employs geometric approach to break down the 6 DOF into 3 + 3 [see Fig. 5] or 4 + 2 [see Fig. 6] etc. Motion occurs about or along the axes at the joints where frames are specified or computed.

Multiple solution issue with Jacobians is resolved by using minimum norm criteria. In practice, a robot has six DOF, articulate robot may have all revolute links. For a robot with 6 DOF, normally arm has three links and wrist has three links. Some robots may have two to any number of links. In any case, in general, the number of links may not be equal to six [5]. As in forward kinematics, application of forward velocity is also straight forward. In fact, this becomes an issue only when the Jacobian is not a square matrix. It happens when the system of equations is underdetermined or overdetermined. For this Inverse Jacobian problem, there are algebraic approaches to resolve this [2],[3]. We provide more efficient ways to grasp the concepts of inverse kinematics and inverse Jacobian.

The paper is organized as follows: Section II is overview of background, Section III describes innovative techniques for computing Inverse Kinematics, Section IV is on Geometric Inverse Kinematics, efficiency, section V is intuitive Inverse Jacobian, Section VI is Conclusion followed by References.

II. OVERVIEW OF BACKGROUND

A robot is a rigid body composed of links moving around the joint axes. The link$_k$ parameters are (θ_k, d_k, a_k, α_k) relative to link$_{k-1,}$ are used to determine a coordinate system, the link frame. The link frame$_k$ is computed by starting at the origin of frame$_{k-1}$ in four transformations (1): rotate about z_{k-1} -axis by angle θ_k , translate along z_{k-1}-axis by amount d_k , translate along x_k-axis by amount a_k, and rotate about x_k-axis by angle α_k . The composition of these four transformations called an A-matrix or coordinate frame. The A matrix is akin to transformation from one frame to next frame. The frame for transformation from frame k-1 to frame k becomes A-attribute matrix, synonymous with transformation matrix, T or the frame matrix, F. For notation: $^{k-1}A_k$ represents transformation from frame k-1 to frame k. Equivalently, it is frame k relative to frame k-1.

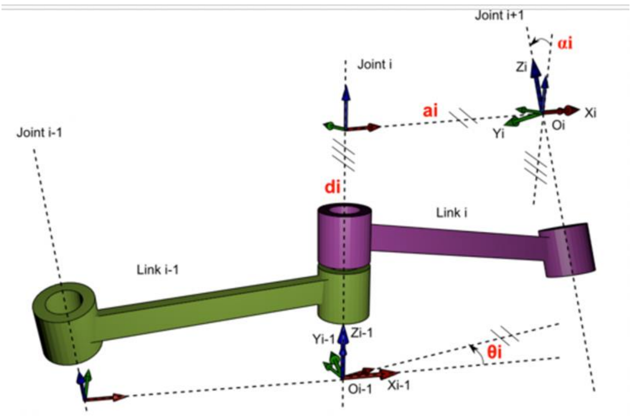

Figure 1. Description of joint coordinates and link parameters (θ_k, d_k, a_k, α_k) [2].

The link parameters (θ_k, d_k, a_k, α_k). Last two (a_k, α_k) are constant all the time, the first two (θ_k, d_k) are variable one of which is constant at the times while other is variable for motion. The equation for composition is as follows

$$ {}^{k-1}A_k \equiv {}^{k-1}F_k \equiv {}^{k-1}T_k = $$
$$ R(z_{k-1}, \theta_k)\, T(z_{k-1}, [0, 0, d_k]),\, T(x_k, [a_k, 0, 0])\, R(x_k, \alpha_k) \quad (1) $$

$$ = \begin{bmatrix} \cos\theta_k & -\sin\theta_k\cos\alpha_k & \sin\theta_k\sin\alpha_k & a_k\cos\theta_k \\ \sin\theta_k & \cos\theta_k\cos\alpha_k & -\cos\theta_k\sin\alpha_k & a_k\sin\theta_k \\ 0 & \sin\alpha_k & \cos\alpha_k & d_k \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2) $$

This is the matrix representing frame k with respect to frame k-1. All measurements are with repect to frame k-1. For frame k-1 with respect to frame k, just invert the matrix (2) [2],[3][6]. This matrix inverse is easy as inverse of rotation is transpose of rotation matrix. The end-effector with respect to base frame becomes

$$ {}^{o}A_n = {}^{0}A_1\, {}^{1}A_2\, {}^{2}A_3 \cdots {}^{n-1}A_n \quad (3) $$

The Link_k frame matrix with respect to universal base frame can be written as

$$ {}^{0}A_k = \begin{bmatrix} {}^{o}R_k & {}^{o}p_k \\ 0 & 1 \end{bmatrix} \text{ or } \begin{bmatrix} {}^{o}n_k & {}^{o}o_k & {}^{o}a_k & {}^{o}p_k \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4) $$

For notation, the symbol ${}^{0}R_k$ is the rotational part of the matrix, ${}^{o}n_k$, ${}^{o}o_k$, ${}^{o}a_k$, are unit vectors forming a right handed system of orthogonal vectors. This represents the orientation of frame_k with respect to base, and ${}^{0}P_k$ is the position of the end-effector with respect to the base frame.

Inverse Kinematics and velocity are described briefly as:

Forward kinematics is a chain of frames or A-matrices ${}^{k-1}A_k$ for k = 1,n, resulting in ${}^{o}A_n$.

In order to have complete and simple solution to *inverse kinematics* for robot manipulators, it is solved analytically to determine closed form solution. *Inverse kinematics* determines the chain of frames or A-matrices ${}^{k-1}A_k$ for k = 1,n from the end-effector, ${}^{o}A_n$. *Inverse kinematics* is based on a standard trigonometric equation $a\cos(\theta) + b\sin(\theta) = c$ with $c^2 \leq (a^2 + b^2)$. Its solution is devoid of singularities of division by zero.

For velocity,the velocity equation uses the Jacobian, J,

$$ \begin{bmatrix} \dot{X} \\ \dot{\Theta} \end{bmatrix} = J\, [\dot{q}] \quad (5) $$

Forward Jacobian: Velocities $[\dot{q}]$ at the joints are specified, the problem is to find end-effector velocity $\begin{bmatrix} \dot{X} \\ \dot{\Theta} \end{bmatrix}$.

Inverse Jacobian: The end-effector velocity $\begin{bmatrix} \dot{X} \\ \dot{\Theta} \end{bmatrix}$ is given, the problem is to find link velocities $[\dot{q}]$ at the joints.

III. TECHNIQUES FOR COMPUTING INVERSE KINEMATICS,

The inverse kinematics involves solving trigonometric equations. If such equation is not solved properly, it may lead to divide by zero singularities in the solution [3].

For example, in the case of a six link robot with first three links create the arm, we will first solve the arm before we go into full scale 6 links. For the sake of completeness, we include the robot Fig. 2, link parameter table Table 1, and the A matrices, etc. For example, Jumbo drilling robot is displayed here in Fig. 2. The Fig. 2 has clearly marked joint axes and link parameters. One may flatten the robot, see Fig. 3 to set it in the rest position for ease in creation the link parameter Table 1.

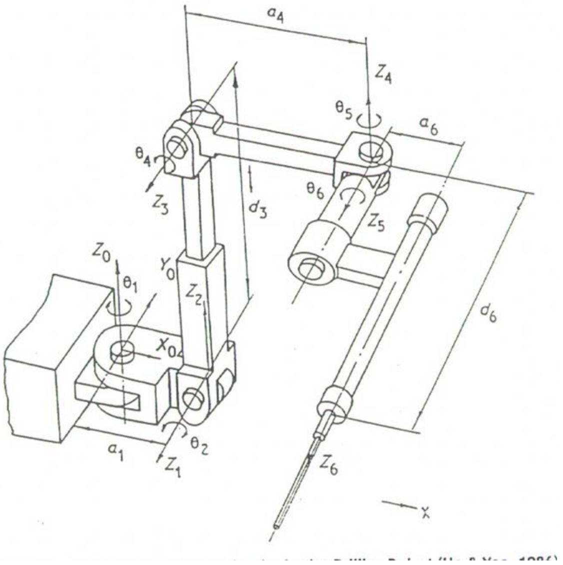

Figure 2. Jumbo Drilling Robot [7], Joint Constraint and Home position

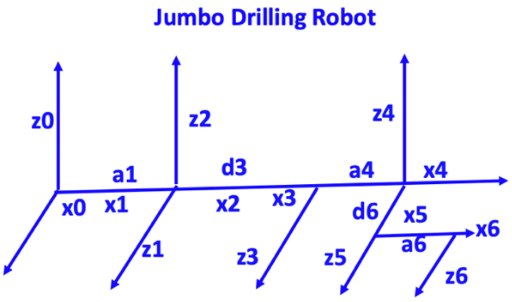

Figure 3. Rest position the Jumbo Drilling Robot.

Table 1. Link parameter table for first three links Jumbo Drilling Robot – Spherical Arm

Jumbo Drilling Robot				
	θ	d	a	α
1	_	0	a_1	90
2	_	0	0	-90
3	0	_	0	90

Using the succint notation for cosines and sines, s_k = sin(θ_k), c_k = cos(θ_k), the A matrices are

$$^0A_1 = \begin{bmatrix} c_1 & 0 & s_1 & a_1c_1 \\ s_1 & 0 & -c_1 & a_1s_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^1A_2 =$$

$$\begin{bmatrix} c_2 & 0 & -s_2 & 0 \\ s_2 & 0 & c_2 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^2A_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A. *Forward Kinematics*

In forward kinematics, we multiply these three a- matrices, use trigonometry to combine complex expressions into simpler ones. There is a unique solution for the end-effector orientation R and position P, see equation (6).

The end-effector presentation uses the simple notation for cosines and sines, s_k = sin(θ_k), c_k = cos(θ_k), and we have

$$\mathbf{^0A_3} = \begin{bmatrix} c_1c_2 & -c_1s_2 & -s_1 & -c_1(s_2d_3-a_1) \\ s_1c_2 & -s_1s_2 & -c_1 & -s_1(s_2d_3-a_1) \\ s_2 & c_2 & 0 & c_2d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (6)$$

The first three columns form a right handed system of mutually orthogonal unit vectors of the rotation matrix and the 4th column is the position vector.

B. *Detour to Trigonometry*

Inverse Kinematics depends on the ability to solve trigonometric equations. Basically, five trigonometric equations were required to solve the inverse kinematics problems [8]. It is not necessary to solve 5 different equations. Only one trigonometric equation is sufficient to resolve all cases. Any special case follows from the same equation. Consequently, in robotics, inverse kinematics amounts to solving the trigonometric equation

$$a \,\mathrm{Cos}\theta + b \,\mathrm{Sin}\theta = c. \qquad (7)$$

All other cases involving trigonometric functions are special case of this equation

$$a \,\mathrm{Cos}\theta + b \,\mathrm{Sin}\theta = c, \; c^2 \leq a^2 + b^2 \qquad (8)$$

The Joint mobility is constrained by mechanical limitations or physical stops. Each joint limits are tied to prismatic and revolute joint that are valid irrespective of the configuration of the remaining links.

There are three trigonometric functions: sin , cos, tan. The inverse functions $\sin^{-1}$(value), $\tan^{-1}$(value) determine angle between $-\frac{\pi}{2}$ and $\frac{\pi}{2}$ whereas $\cos^{-1}$(value) angle is between 0 to π. For robots, the rotation angle for the arm joints is 270 degrees and the wrist joints is 180 degrees [Fig. 4], 360 degree for drilling hole [2]. The $\sin^{-1}$(value) , $\cos^{-1}$(value) , $\tan^{-1}$(value) are insufficient to cover all quadrants for accuracy. Some authors use the inverse functions erroneously [9], [10]. For example, we explore solving the following equations for θ_1, θ_2, d_3, from the position (6) of robot arm

$$P_x = -c_1(s_2d_3-a_1) \qquad (9)$$

$$P_y = -s_1(s_2d_3-a_1) \qquad (10)$$

$$P_z = c_2d_3 \qquad (11)$$

Their [10] method would solve them as follows, their approach runs into undefined values for θ_1, θ_2 on using standard inverse atan functions

From (4,1),(4,2),(4,3)

$$P_x = -c_1(s_2d_3-a_1) \qquad (12)$$

$$P_y = -s_1(s_2d_3-a_1) \qquad (13)$$

$$P_z = c_2d_3 \qquad (14)$$

The authors from [9], [10] simplify

$$\frac{p_y}{p_x} = \frac{-s_1(s_2d_3-a_1)}{-c_1(s_2d_3-a_1)} = \frac{s_1}{c_1} \qquad (15)$$

and calculate θ_1

$$\theta_1 = \mathrm{atan}(\frac{p_y}{p_x}) \qquad (16)$$

There are several issues with this calculation 1. Cancelling negative sign creates discrepancy of 180 degrees in the angle, 2. Cancelling ($s_2d_3-a_1$) is not practical if it is zero. In that case division by zero is undefined. That is, if $p_x = 0$, then θ_1 is undefined. In fact it may be well defined. We propose to use atan2, tan inverse with two parameters, if px = 0, py ≠0, then

$$\theta_1 = \mathrm{atan2}(p_y,p_x) = \mathrm{atan2}(p_y,0) = \pm\frac{\pi}{2} \qquad (17)$$

Secondly [9] calculates

$$\theta_2 = \mathrm{atan}\,(\frac{p_z}{-(a_1-c_1p_x-s_1p_y)}) \qquad (18)$$

which is again undefined if the expression (a_1-c_1P_x-s_1P_y) = 0. It leads to both θ_1 and θ_2 as undefined as well as erroneous.

In fact, the solution can be well defined if we avert any such discrepancies involving divisions by zero that create undefined expressions. First, we use atan2 function that prevents such discrepancies of undefined terms in the calculation of angles, Second we avoid divisions by zero as follows.

In general, using atan2, the solution of

$$a \,\mathrm{Cos}\theta + b \,\mathrm{Sin}\theta = c, \; c^2 \leq a^2 + b^2 \qquad (19)$$

is

$$\theta = \mathrm{atan2}(b,a) \pm \mathrm{atan2}(\sqrt{a^2+b^2-c^2}, c) \qquad (20)$$

There is no division by zeros, or undefined expression.
Very frequently we come across the equation

$$a \operatorname{Cos}\theta + b \operatorname{Sin}\theta = 0 \quad (21)$$

with $a^2 + b^2 \neq 0$. Here too, the solution becomes

$$\theta = \operatorname{atan2}(b,a) \pm \operatorname{atan2}(\sqrt{a^2+b^2}, 0) \quad \text{or}$$
$$\theta = \operatorname{atan2}(b,a) \pm \operatorname{atan2}(1, 0) \quad \text{or}$$
$$\theta = \operatorname{atan2}(b,a) \pm \frac{\pi}{2} \quad (22)$$

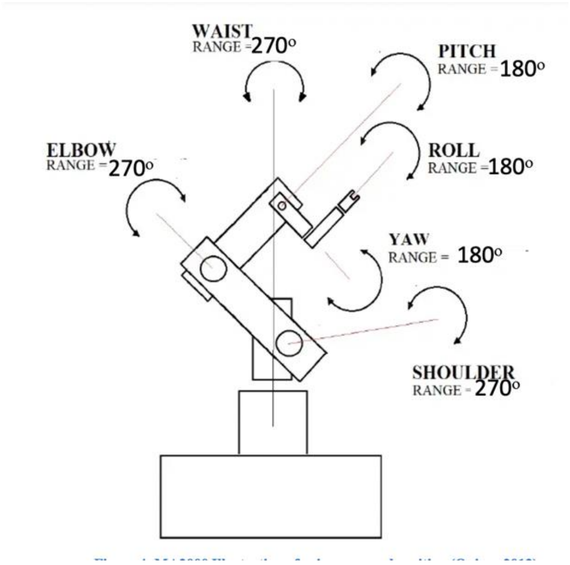

Fig. 4 Illustration of axis range and position MA2000 [9]

C. Inverse Kinematics analytical

With trigonometry at hand, we are in a position to solve the three link inverse kinematics for the end-effector

$$\mathbf{{}^0A_3} = \begin{bmatrix} c_1c_2 & -c_1s_2 & -s_1 & -c_1(s_2d_3-a_1) \\ s_1c_2 & -s_1s_2 & -c_1 & -s_1(s_2d_3-a_1) \\ s_2 & c_2 & 0 & c_2d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (23)$$

Depending on the robot, the inverse kinematics problem may be specified with just (a) Orientation of end-effector, (2) Position of end-effector or (c) both the Position and Orientation of the end-effector.

C.1. Inverse Kinematics when only end-effector position is specified

If the position alone is specified, there can be multiple solutions. Let the postion P be provided for this example.

From (4,1),(4,2)

$$P_x = -c_1(s_2d_3-a_1) \quad (24)$$
$$P_y = -s_1(s_2d_3-a_1) \quad (25)$$

Eliminate the terms that may be cause of concern.
Multiply (24) by s_1 and (25) by c_1 and subtract

$$s_1P_x-c_1P_y = 0 \quad (26)$$

There are two solutions

$$\theta_1 = \operatorname{atan2}(-P_x, P_y) \pm \frac{\pi}{2} \quad (27)$$

But [9], [10] calculate $\theta_1 = \operatorname{atan}\left(\frac{p_y}{p_x}\right)$ which will be undefined when $p_x = 0$

Reusing (4,1),(4,2) , multiply(24) by $-c_1$ and (25) by $-s_1$ and add, we get

$$-c_1P_x-s_1P_y = s_2d_3-a_1 \quad (28)$$
$$a_1-c_1P_x-s_1P_y = s_2d_3 \quad (29)$$

and from (4,3)

$$P_z = c_2d_3 \quad (30)$$

Multiply (25) $P_z = c_2d_3$ with s_2 and (29) $a_1-c_1P_x-s_1P_y = s_2d_3$ with c_2 and subtract, we get

$$s_2P_z - (a_1-c_1P_x-s_1P_y)\, c_2 = 0 \quad (31)$$
$$s_2P_z + (c_1P_x+s_1P_y- a_1)\, c_2 = 0 \quad (32)$$

There are two solutions

$$\theta_2 = \operatorname{atan2}(P_z, c_1P_x+s_1P_y- a_1) \pm \frac{\pi}{2} \quad (33)$$

Recall [9], [10] calculate θ_2 as $\operatorname{atan}\left(\frac{p_z}{-(a_1-c_1p_x-s_1p_y)}\right)$ which is undefined when $a_1-c_1P_x-s_1P_y = 0$.

Finally again from (4,1),(4,2),(4,3), for each value of θ_1 , θ_2 we get unique value for d_3. Multiply $P_z = c_2d_3$ with c_2 and $a_1-c_1P_x-s_1P_y = s_2d_3$ with s_2 and add, we get a unique

$$d_3 = (a_1-c_1P_x-s_1P_y)s_2 + P_z\, c_2 \quad (34)$$

Thus, if only arm position is specified, then there are **four possible solutions** (θ_1, θ_2, d_3)

C.2 Inverse kinematics when end-effector is completely specified.

If orientation and position are both specified, then there is a unique solution. That amounts to solving the following equations. Let the rotation matrix be R = [N, O,A] and position be P.

We use the same example and only atan2(y,x) function.

From (1,3),(2,3)

$$s_1 = - A_x,\; c_1 = - A_y \quad (35)$$

We find unique

$$\theta_1 = \operatorname{atan2}(-A_x, -A_y) \quad (36)$$

From (3,1),(3,2)

$$s_2 = N_z,\; c_2 = O_z \quad (37)$$

We find unique

$$\theta_2 = \operatorname{atan2}(N_z, O_z) \quad (38)$$

From (4,1),(4,2)

$$P_x = -c_1(s_2d_3-a_1)$$
$$P_y = -s_1(s_2d_3-a_1)$$

Now that θ_1 and θ_2 are known, we multiply the equation (24) by $-c_1$ and (25) by $-s_1$, and add. We get

$$-c_1P_x-s_1P_y = s_2d_3-a_1 \quad (39)$$
$$a_1-c_1P_x-s_1P_y = s_2d_3 \quad (40)$$

From (4,3)

$$P_z = c_2d_3 \quad (41)$$

Assembling these two equations (40),(41) we get

$$d_3 = (a_1-c_1P_x-s_1P_y)\, s_2 + P_z\, c_2 \quad (42)$$

We find d_3 is unique. Overall, by specifying both position and orientation, complete specification leads to a **unique solution** θ_1, θ_2, d_3.

IV. GEOMETRIC APPROACH TO INVERSE KINEMATICS

Here we provided an efficient way to to solve invers kinematics and inverse Jacobian proble. We determine the intermediate frame at the end of arm , link 3 or link 4 between the base frame ad end-effector of a 6 link robot. It makes it easy to solve 3 link or 4 link inverse kinemeic problem. For larger number of links, it becomes more challenging. To overcome this complexity, sometimes geometric approach is used. Six link robot inverse

Kinematics problem can be easilty solved by partitioning it into 3 + 3 [Fig. 5] or 4 + 2[Fig. 6] link inverse kinematics problems. The traditional geometric approaches require the specification, with respeect to base, of both Position and Orientation of the end-effector $\begin{bmatrix} n & o & a & p \\ 0 & 0 & 0 & 1 \end{bmatrix}$. Now with n,o,a,p geomery is used to calculate, with respect to base, $\begin{bmatrix} N & O & A & P \\ 0 & 0 & 0 & 1 \end{bmatrix}$ for the end of link3 or link4. Using $\begin{bmatrix} N & O & A & P \\ 0 & 0 & 0 & 1 \end{bmatrix}$, we calculate the parameters of first 3 or 4 links. Once it is done, we solve for parameters of the remaing links by equating frame 3 or frame 4 (product of remaing A matrices) respectively to $\begin{bmatrix} N & O & A & P \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} n & o & a & p \\ 0 & 0 & 0 & 1 \end{bmatrix}$.

We *propose to require only orientation* $[n \quad o \quad a]$ of the end-effector, that is sufficient, specification of the position of end-effector is not necessary[see Fig. 5, 6] for 3 + 3 and 4 + 2 partitioning. In either case, we need these parameters at the end of 3 or 4 links [Fig. 5, 6]. Once, the frame at the end of arm $[N \quad O \quad A]$ is determined, it can be used to solve the rest of the problem, as we saw in anaylcal solutions above. The orientation and position of link three or four can be determined without the position specification of the end-effector. All other steps are similar to previous paragraph. More details on comparison of two approaches follow.

A. *Traditional approach*

In Fig. 5 (3 + 3 case), visually the orientation of frame 3 is the same as frame 6, so the frame 3 orientation is the same as specified for the end-effector. Standard approach to get the postion of frame 3 with respect to base is as follows. Starting from P, we can go a_4 units in direction n; a_6 units along n, it gives the (a4 + a6) along n and d_6 along a. This means P + (a_4 + a_6) n + d_6 a = p, which is given. Therfore the postion of frame 3 origin is P = p - (a_4 + a_6) n- d_6 a , completely known in terms of base coordinate system.

In figure 6 (4 + 2 case), visually the orientation of frame 4 is such that N is the same as n, A is o, O is -a. Standard approach get the postion of frame 4 with respect to base frame is as follows: starting from P, we can go a_6 units along the direction n; and d_6 along a. This means P + a_6 n + d_6 a = p, which is given. Therfore the postion of frame 4 origin is P = p - a_6 n- d_6 a.

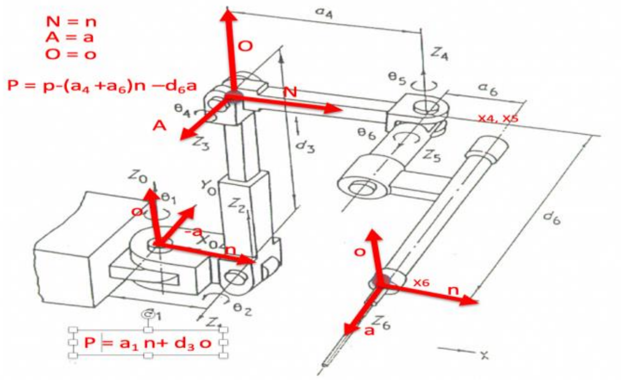

Figure 5. Partition of 6 links into 3 + 3 for inverse kinematics. Base frame expressed in terms of n, o, a.

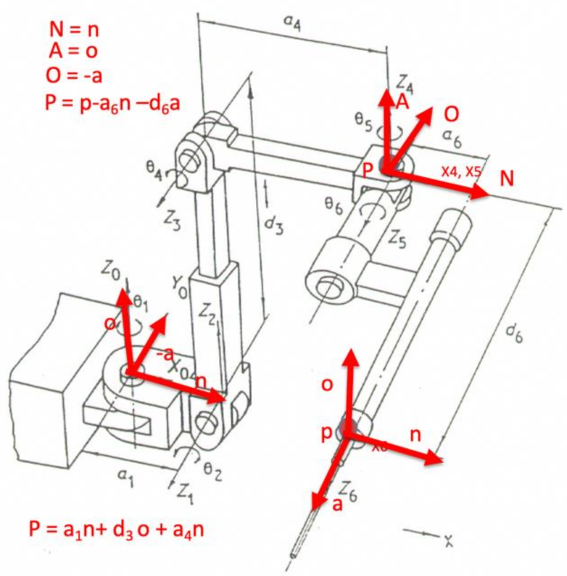

Figure 6. Partition of 6 links into 4 + 2 for inverse kinematics. Base frame expressed in terms of n, o, a.

B. More Explantion on New Approach

We propose that specification of p is not necessary. P can be directly determined without the knowledge of p. It is observed that [N,O,A] is known, base frame orientation becomes [n,-a, o], Fig. 5, Fig. 6.

In Fig. 5, we can step a_1 units along n, d_3 along o resulting in P = a_1 n + d_3 o. In Fig. 6, we can step a_1 along n, d_3 along o, a_4 along n resulting P = (a_1 + a_4) n + d_3 o. In both the cases see Figure 5 and 6, we have determined that the position is P = a_1 n + d_3 o; orientation is [N, O, A] = [n, o, a] in Fig. 5 and the position is P = (a_1 + a_4) n + d_3 o; orientation is [N, O, A] = [n, -a, o] in Fig. 6. This shows that arm position P can be constructed without specifying p.

V. HEURISTIC APPROACH AND JUSTIFICATION FOR INVERSE JACOBIAN,

The paper presents a heuristic method to solve inverse Jacobian for arbitrary number of degrees of freedom (DOF) robot. The Jacobian inverse computation takes long time, whereas first matrices must be multiplied, then inverse of the product is computed and then inverse is multiplied with the transpose of the original to derive the generalized inverse. We will show how this is simpler and will apply this approach to both under- and over-determined systems of velocity equations. We also provide a better method for solving inverse Jacobian problems that avoid singularity in the solutions.

The Jacobian equation

$$\begin{bmatrix} \dot{X} \\ \dot{\Theta} \end{bmatrix} = \mathrm{J}\,[\dot{q}] \tag{43}$$

can be mapped into simple equation

$$\mathrm{B} = \mathrm{AX} \tag{44}$$

With minimum computation effort and easily comprehensible approach, we proceed as follow. Since A is m×n, there are three possibilities

$$1.\ m < n, \quad 2.\ m = n, \quad 3.\ m > n. \tag{45}$$

For m = n, we can directly resort to inverse of A to arrive at

$$X = A^{-1}\,B.$$

For m<n or m>n we compute product of A with its transpose in such way that the product is a square matrix of smaller order. For example, if m>n , then A^TA is order n×n, and if m<n, then AA^T is smaller order m×m. Then we use the inverse of these matrices to arrive at the solution. Without going through the exercise of minimization, we can determine the solution. Vector differentiation is not trivial used for computing generalized inverses when m<n and m>n. Moreover, for m×n matrix A with m>n , since A^TA is a smaller square n×n matrix, inverse is also square n×n matrix. To derive the solution, we simply multiply n×n matrix $(A^TA)^{-1}$ and n×m matrix A^T, then n×m matrix $(A^TA)^{-1}\,A^T$ and m×1 vector B in that order resulting in solution to the equation,

$$X = (A^TA)^{-1}A^T\ B. \qquad (46)$$

For efficiently , we can multiply n×m matrix A^T and m×1 vector B first, then resulting m×1 vector A^TB, and n×n matrix $(A^TA)^{-1}$ yielding

$$X = (A^TA)^{-1}\ A^TB. \qquad (47)$$

Now we compare the computing effort in solving the equations if A^T is multiplied with $(A^TA)^{-1}$ first or when A^T is multiplied to B first in the case of m>n. Inversion is the same in both the cases, the number of multiplications is:

first case (46): $m(n^2 + (m + 1)n) > m(n^2 + (n + 1)n)$

second case(47): $m(n^2 + 2n)$

Clearly second approach is more efficient. Since we are interested in minimum error or minimum norm solution, we will create the solution directly, heuristically minimal error is achieved by using a matrix of smaller dimension for inversion. Smaller the matrix, smaller the error in computation and representation. It also leads to conceptually comprehensible and easy to grasp the solution.

A. Underdetermined system, m<n

For underdetermined systems there can be infinitely many solutions with no error of computation. However, we are interested in solution with smallest norm, in other words, a solution that is closest to the origin or a shortest distance from the origin.

We show that for m<n,

$$X = A^T(AA^T)^{-1}\,B \qquad (48)$$

is a least norm solution for equation

$$AX = B \qquad (49)$$

This X is closest to the origin. Traditionally, we minimize

$$f(X) = |X|^2 \text{ subject to } AX = B \qquad (50)$$

By method of Lagrange multipliers

$$f(X) = X^T\,X + \lambda^T\,(B\text{-}AX) \qquad (51)$$

Differentiate with respect to λ and X, and equate them to zero

$$\frac{\partial f(X)}{\partial\lambda} = 0 \text{ yields } B - AX = 0 \text{ or } B = AX \qquad (52)$$

$$\frac{\partial f(X)}{\partial X} = 0 \text{ yields } 2X\text{-}A^T\lambda = 0 \qquad (53)$$

Which is

$$2AX\text{-}AA^T\lambda = 0 \quad \text{or}$$

$$2B\text{-}AA^T\lambda = 0 \quad \text{or} \quad 2B = AA^T\lambda \qquad (54)$$

Solve for λ

$$(AA^T)^{-1}2B = \lambda \qquad (55)$$

and substitute

$$2X\text{-}A^T(AA^T)^{-1}2B = 0 \qquad (56)$$

We get

$$X = A^T(AA^T)^{-1}B \qquad (57)$$

which is the solution to the equation AX = B.

This solution can be directly and quickly determined from the equation, AX = B.

For the purpose of creating a smaller size matrix for inversion, let us write

$$X = A^TU \text{ for some U,} \qquad (58)$$

For this U, the equation becomes

$$AX = AA^TU \text{ or } B = AA^TU \qquad (59)$$

Now $A\,A^T$ is m×m square matrix, the inverse can be used to compute U.

$$U = (A\,A^T)^{-1}\,B \qquad (60)$$

From $X = A^TU$, and $U = (A\,A^T)^{-1}\,B$ we get (61)

$$X = A^T(A\,A^T)^{-1}\,B \qquad (62)$$

which proves the correctness of our intuitive solution shown above.

B. Overdetermined system, m>n

For overdetermined systems there may not be any solution. Any approximate solution may have an error of approximation. We can settle for an approximate solution that has minimum approximation error.

We show that for m>n,

$$X = (A^TA)^{-1}A^TB \qquad (63)$$

is a least square error solution for equation

$$AX = B \qquad (64)$$

We proceed to minimize $f(X) = |AX\text{-}B|^2$ subject to AX = B. The function is written as

$$f(X) = (AX\text{-}B)^T(AX\text{-}B)$$

$$= (X^TA^T\text{-}B^T)\,(AX\text{-}B) \qquad (65)$$

$$f(X) = X^TA^T\,AX\,\text{-}B^T\,AX\,\text{-}\,X^TA^TB + B^T\,B \qquad (66)$$

$$f(X) = X^TA^TAX - 2\,X^TA\,^TB + B^T\,B \qquad (67)$$

Differentiate with respect to X and set it equal to zero

$$\frac{\partial f(X)}{\partial X} = 0 \text{ yields } 2\,A^TAX\text{ - }2\,A^TB = 0 \qquad (68)$$

$$A^TAX\text{ - }A^TB = 0 \quad A^TAX = A^TB \quad \text{or} \qquad (69)$$

$$X = (A^TA)^{-1}A^TB \qquad (70)$$

Which is a solution to the equation AX = B. This can be directly and quickly determined from

$$AX = B \qquad (71)$$

Multiply by A^T

$$A^TAX = A^TB \qquad (72)$$

Now A^TA is n×n square matrix of smaller size, the inverse can be used to compute X by multiplying both sides with $(A^TA)^{-1}$.

$$X = (A^TA)^{-1}\,A^TB \quad \text{or} \qquad (73)$$

$$X = (A^TA)^{-1}A^T\,B \qquad (74)$$

which proves the correctness of our simplified intuitive solution computed above.

The matrix $(A^TA)^{-1}A^T$ or $A^T(A\,A^T)^{-1}$ is the pseudo inverse or generalized inverse of A. It is synonymously denoted with ay of several symbols such as: A^+, A^-, $A^\#$, A^I etc.

VI. CONCLUSION

Inverse Kinematics and Inverse Jacobian are two of the problems in robot motion consideration. We have shown how to resolve these two issues with an intuitive, persuasive and provable approach. We gave closed form, efficient solution for both Inverse Kinematics and Inverse Jacobians. In fact, we have proved that our approach is accurate and efficient. We hope that the designers will find it useful in understanding, and retaining the description for reference in their future implementations.

VII. REFERENCES

[1] V. Grecu, N. Dumitru, and L. Grecu,” Analysis of Human Arm Joints and Extension of the Study to Robot Manipulator”, Proceedings of the international MultiConference of Engineers and Computer Scientists, Vol 2, March 18 - 20, 2009 pp:1348-1351.

[2] Saeed B. Niku, Introduction to Robotics analysis, control, applications publisher: Wiley 2011 ISBN-978-0-470-60446-5

[3] Craig, John J, Introduction to Robotics, Mechanics and Control by. 3rd Edition, Prentice Hall 2005.

[4] C. Hua, C. and Wei-shan, “Wavelet network solution for the inverse kinematics problem in robotic manipulator”,Chen Zhejiang Univ Science , Vol. 7 ,No. 4 ,2006 PP: 525-529.

[5] V.Sanchez,R.Gutierrez, G.Valdovinos and P.Ortega, “5-DOF manipulator simulation based on MATLAB-Simulink methodology “, 2010 20th International Conference on Electronics, Communications and Computer (CONIELECOMP), 2010, pp: 295 – 300

[6] Mohammed Z. Al-Faiz, Abduladhem A. Ali3,Abbas H.Miry, HUMAN ARM SIMULATION BASED ON MATLAB WITH VIRTUAL ENVIROMENT, IJCCCE, VOL.11, NO.1, 2011, pp. 86-96.

[7] C.Y.Ho; Yao Jianchi, Design of Automated Jumbo Drilling Robot, Manipulator, IFAC Proceedings Volumes, Volume 20, Issue 8, August 1987, Pages 59-66

[8] S. Kucuk dan Z. Bingul, “Robot kinematics: Forward and inverse kinematics”, in *Industrial Robotics: Theory, Modelling and Control,* InTech, 2006.

[9] Mohammed Z. Al-Faiz ,Mohammed S.Saleh, International Journal of Robotics and Automation (IJRA), Volume (2) : Issue (4) : 2011, pp.256-264.

[10] I. Agustian, N. Daratha, R. Faurina, A. Suandi, and Sulistyaningsih, “Robot Manipulator Control with Inverse Kinematics PD-Pseudoinverse Jacobian and Forward Kinematics Denavit Hartenberg,” Jurnal Elektronika dan Telekomunikasi, vol. 21, no. 1, pp. 8–18, 2021.