Geometric Search-Based Inverse Kinematics of 7-DoF Redundant Manipulator with Multiple Joint Offsets

Anirban Sinha[1] and Nilanjan Chakraborty[2]

***Abstract*— We propose a geometric method to solve inverse kinematics (IK) problems of 7-DoF manipulators with joint offsets at shoulder, elbow, and wrist. Traditionally, inverse position kinematics for redundant manipulators are solved by using an iterative method based on the pseudo-inverse of the manipulator Jacobian. This provides a single solution among the infinitely many possible solutions for the IK problem of redundant manipulators. There are no closed-form IK solutions for redundant manipulators with multiple joint offsets. Using our method we can compute multiple IK solutions using two-parameter search by exploiting geometry of the structure of a redundant manipulator. Our proposed IK algorithm can handle multiple joint offsets and is mathematically simple to implement in a few lines of code. We apply our algorithm to compute IK solutions for 7-DoF redundant Baxter robot (that has joint offsets at shoulder, wrist, and elbow joints) for end-effector configurations where existing geometry-based IK solvers fail to find solutions. We also demonstrate the use of our algorithm in an application where we want to compute an IK solution (among the infinitely many possible solutions) that has minimum error bound in end-effector position, in the presence of random joint actuation and sensing uncertainties.**

I. Introduction

Inverse kinematics (IK) is a fundamental problem in robot motion and trajectory planning. Given a desired end effector position and orientation (i.e., pose), the IK problem is to find the set of configurations or joint angles of the manipulator such that if the robot goes to any configuration in this set, then its end effector is at the desired pose. In other words, IK gives a mapping from the end-effector space or *task space* to *joint space* of a robot. The IK mapping between the task space and joint space is usually a one-to-many map. Depending on the degree of redundancy of a manipulator one can have different number of inverse kinematics solutions for the same given end-effector configuration. For instance a 6 degree of freedom (DoF) spatial manipulator has at most 16 IK solutions. On the other hand a 7 DoF manipulator will have infinitely many inverse kinematics solutions.

There are many applications in which we need to search over multiple IK solutions to compute an IK solution that is relevant to the task at hand. For redundant robots this is known as redundancy resolution [1]. Motivated by applications in assembly with close tolerance, we are interested in computing the IK solution, where we can bound the worst case positioning error of the end-effector of a redundant manipulator. Errors in control/sensing implies that although all IK solutions for reaching an end-effector configuration are theoretically equivalent, the effect of errors in joint angles may lead to different errors in the end-effector space, since the forward kinematics mapping is nonlinear. In this paper, we are motivated by a redundancy resolution application, where we want to compute an IK solution that minimizes the maximum positioning error in the task space. Therefore, we need a method for generating all IK solutions within a numerical tolerance.

[1]Anirban Sinha and [2]Nilanjan Chakraborty are affiliated with Department of Mechanical Engineering at Stony Brook University. Email: anirban.sinha@stonybrook.edu; nilanjan.chakraborty@stonybrook.edu

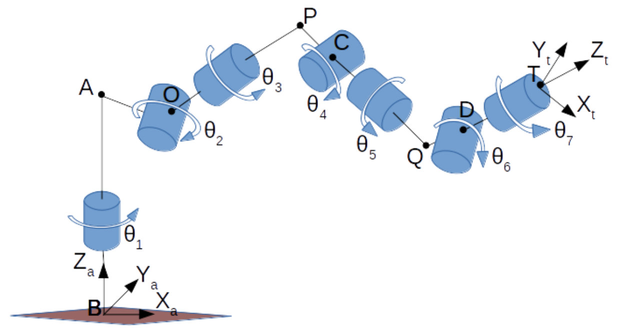

Fig. 1. Schematic of the robot arm in accordance to Table I with non zero joint angles. The points of interest are also shown along with the link lengths. Robot base is located at point B

There are two classes of methods used for solving IK problems, namely, Jacobian-based methods and geometric methods. However, all inverse kinematics methods available in the literature are not suitable to find multiple IK solutions. For instance, pseudo-inverse of Jacobian based methods (also known as velocity-based methods [2]) can only produce one IK solution for a given end-effector configuration. The analytical methods, which use geometry of manipulator links and joints, can produce multiple IK solutions. Although there are quite a few geometric methods for solving IK of redundant manipulators [3], [4], [5], [6], there are currently available commercial manipulators (e.g., Baxter from Rethink Robotics) for which no geometric methods are available. The goal of this paper is to develop geometric solution methods for the IK problem of a class of 7 DoF redundant manipulators (like Baxter arm) with joint offsets.

Figure 1 gives a schematic sketch of the class of manipulators that are considered in this paper. Many of the 7 DoF manipulators that are commercially available (e.g., WAM arm from Barrett Technologies, Baxter arm from Rethink Robotics) are anthropomorphically inspired and variants of the spherical-revolute-spherical (SRS) arm. The SRS arm is a 7-DoF arm with a spherical joint at the shoulder, a revolute joint at the elbow and a spherical joint at the wrist. Practical manufacturing and actuation constraints make designers introduce offsets between the joint axes at the shoulder and wrist joint. In particular, for the arm shown in Figure 1, there is an offset between joint 2 and joint 1, joint 3 and joint 4, and joint 5 and joint 6. These offsets make the computation of the IK solutions challenging and current available methods [3], [4], [5], [7] are inapplicable.

By exploiting the geometric structure of the manipulator, we develop a novel two-parameter search method to find multiple IK solutions. The first parameter is first joint variable, namely θ_1, which parametrizes this angle between its upper and lower limits. The second parameter parametrizes a circle upon which the joint 4 will always stay. Our approach incorporates the joint limits of the manipulator and outputs only feasible solutions that do not violate joint limits. We also show application of our IK method for computing the IK solution for minimizing the maximum positioning error.

II. Related Work

The inverse kinematics problem is one of the fundamental problems in robotics and has a rich history (see [8], [9], [10], [11], [12], [13], [14] and references therein). In this work, we are concerned with IK of redundant manipulators [15], [16], [17], [18], [19], [20]. Therefore, we will restrict ourselves to this literature. The IK problem of a redundant manipulator can have an infinite number of solutions and hence it is mathematically ill-posed. Major approaches to solution of IK problems for redundant manipulators are (a) Differential Kinematics based approach [21], [22] (b) Optimization-based approach [23] and (c) Geometric approaches [3], [4], [5]. Differential kinematics and optimization-based approaches are designed to obtain a path for the manipulator to reach one solution of the IK problem (among the infinitely many possible), by often using additional criterion like joint limit avoidance, obstacle avoidance etc. to make the problem well posed. This is known as redundancy resolution in the literature (see Chapter 11 of [8] and references therein). Such algorithms are capable of producing solutions to the IK problem (by ignoring the path), but they can produce only one solution. Geometric methods on the other hand are intended to solve the IK problem only. Geometric methods provide parametric solutions to the IK problem (with low number of parameters).Thus, they provide the possibility of computing multiple solutions (of course, within a numerical resolution). Geometric methods can be also used for redundancy resolution [4].

Geometric solutions for the IK problem of 7-DoF redundant manipulators without joint offsets are proposed in [4], [6]. This corresponds to the zero lengths of the links AO, PC and QD of the robot model presented in Fig. 1. The method proposed in [4], [6] produces a one parameter solution and the solutions satisfy joint limits. In [24] IK problem is solved for a 7-Dof manipulator with shoulder offset. Their method requires an offset angle and end effector configuration to compute 16 joint solutions corresponding to the given input. The IK problem of an anthropomorphic robot arm without joint offset has been solved in [25]. Analytical IK solution for a 7-DoF manipulator with shoulder and wrist offset is given in [26]. In their robot model, $|PC|= 0$, and therefore, they do not address the problem with elbow offset. One parameter IK solution for the WAM arm (whole arm manipulator) with elbow and wrist offset is proposed in [3]. In this manipulator, PC and QD has non zero lengths. However for WAM all its main and offset links are arranged in such a manner that they are bound to lie in a single plane. Their work does not discuss about the problem when offset links are allowed to lie in different planes while attaining a posture. Further, their work does not state the case when shoulder joint has offset.

In [27], the author presents a closed form IK method (also known as IKFast) for two special classes of manipulators, where either first three or last three joint axes of the manipulator intersect at a point. Although there is a custom closed form IKFast available for Baxter robot, it is unclear how the joint offsets are taken into account. To the best of our knowledge there is no documentation of extending the closed form methods in [27] to manipulators with joint offsets (like the ones we consider in this paper). Experimentally, we have found that there are configurations where the IKFast code does not provide a solution. Therefore, for our redundancy resolution application for finding robust joint space configuration, we develop a method to compute multiple IK solutions, for manipulators with joint offsets.

Our proposed method can solve IK for 7-DoF redundant manipulator with shoulder, elbow, and wrist offsets. A schematic of 7-DoF manipulator with shoulder, elbow and wrist offsets are shown in Figure 1. This manipulator is a general case of the 7-DoF manipulators for which geometric solutions are not available in the literature. We apply our IK method to determine the joint angles of a Baxter robot arm.

III. Inverse Kinematics Problem

We consider 7-DoF serial chain manipulator with revolute joints with joint limits. Let $\mathbb{R}^n$ be the real Euclidean space of dimension n, $\mathbb{R}^{m\times n}$ be the set of all $m \times n$ matrices with real entries, $SO(3)$ be the Special Orthogonal group of dimension 3, and $SE(3)$ be the Special Euclidean group of dimension 3. $SO(3)$ and $SE(3)$ are defined as [28]: $SO(3) = \{\mathbf{R} \subset \mathbb{R}^{3\times 3} | \mathbf{R}^T\mathbf{R} = \mathbf{R}\mathbf{R}^T = \mathbf{I}\}$ $SE(3) = \{(\mathbf{R}, \mathbf{p}) | \mathbf{R} \in SO(3), \mathbf{p} \in \mathbb{R}^3\}$ where $\boldsymbol{I}$ is a 3×3 identity matrix. The set of all joint angles, $\mathcal{J}$, is called the *joint space* of the robot. In our case $\mathcal{J} \subset \mathbb{R}^7$. The set of all end-effector configurations is the *end-effector space* or *task space* of the robot and is a subset of $SE(3)$. An element $\boldsymbol{T} \in SE(3)$ can be represented by a 4×4 transformation matrix,

$$\boldsymbol{T} = \left[\begin{array}{ccc|c} & \boldsymbol{R} & & \boldsymbol{p} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{array}\right] \tag{1}$$

Problem Statement: Given a desired end effector configuration ${}^0\boldsymbol{T}_{des} \in SE(3)$ determine the joint angle vector $\boldsymbol{\Theta} = [\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7]$, such that $\boldsymbol{F}(\boldsymbol{\Theta}) = {}^0\boldsymbol{T}_{des}$ where $\boldsymbol{F}(\cdot) : \mathbb{R}^n \to SE(3)$ is *forward kinematics* map.

Since the DoF of the manipulator is greater than 6 (the DoF of a rigid body moving in $SE(3)$), there are infinitely

TABLE I

DH (STANDARD) PARAMETER FOR KINDS OF MANIPULATORS WHERE OUR METHOD IS USEFUL

sl no.	$\theta(rad.)$	$a_k(m.)$	$\alpha_k(rad.)$	$d_k(m.)$
1	θ_1	a_1	$-\pi/2$	d_1
2	θ_2	0	$\pi/2$	0
3	θ_3	a_3	$-\pi/2$	d_3
4	θ_4	0	$\pi/2$	0
5	θ_5	a_5	$-\pi/2$	d_5
6	θ_6	0	$\pi/2$	0
7	θ_7	0	0	d_7

many possible values of $\boldsymbol{\Theta}$ for the same end effector configuration. The DH parameters of the type of manipulator that we are considering are given in Table I. A schematic diagram of the robot is depicted in Figure 1. The difference between the DH parameters in Table I here and in [4] is that the later has every entry in the a_i column as 0. In other words, there are no joint offsets in [4]. Our manipulator is made of a shoulder (joints 1, 2, 3), elbow (joints 4) and wrist (joints 5, 6, 7). The manipulator has 3 joint offsets, first one is between joints 1 and 2, the second offset is between joints 3 and 4, the third offset is between joints 5 and 6. On the other hand, the joints 2 and 3, joints 4 and 5, and joints 6 and 7 have consecutive intersecting axes. The correspondence between Table I and links of the robot in Figure 1 are:

$\overline{BA} \rightarrow d_1$, $\overline{OP} \rightarrow d_3$, $\overline{CQ} \rightarrow d_5$, $\overline{DT} \rightarrow d_7$, $\overline{AO} \rightarrow a_1$, $\overline{PC} \rightarrow a_2$, $\overline{QD} \rightarrow a_3$.

Each row of the DH Table I assigns one frame to describe the robot. Location of these frames pertaining to robots in Figure 1 are, frame 1 is located at point B. Frames 2 and 3 are concentrated at point O. Frames 4 and 5 are located at point C. Frames 6 and 7 are located at point D. From now on we will use the terms joints and frames interchangeably.

Given DH parameters for i^{th} link, the transformation matrix of i^{th} frame with respect to the $(i-1)^{th}$ frame is

$$^{i-1}\boldsymbol{T}_i = \begin{bmatrix} C\theta_i & -S\theta_i C\alpha_i & S\theta_i S\alpha_i & a_i C\theta_i \\ S\theta_i & C\theta_i C\alpha_i & -C\theta_i S\alpha_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (2)$$

The transformation matrix for n^{th} frame with respect to the base frame can be obtained by [29],

$$^{0}\boldsymbol{T}_n = ^{0}\boldsymbol{T}_1\, ^{1}\boldsymbol{T}_2(\theta_1)\, ^{2}\boldsymbol{T}_3(\theta_2) \cdots\, ^{n-1}\boldsymbol{T}_n(\theta_{n-1}) \quad (3)$$

Note that the transformation matrix ($^{0}\boldsymbol{T}_1$) in Equation (3) represents transformation of robot's base frame (frame 1) with respect to (w.r.t.) world frame (frame 0). If one considers robot base frame as world frame then simply assume $^{0}\boldsymbol{T}_1 = \boldsymbol{I}_{4\times4}$. However there are commercially available robots (Baxter robot [30]) whose world frame is different from first DH frame.

IV. Solution Approach

We now present a two parameter search method to solve the IK problem described in the previous section. Our proposed algorithm relies on four main facts revealed by the geometry of the robot described in Figure 1 and Table I.

1) Lengths of OC, CD, and DT are invariant irrespective of configuration of end effector (please see Figure 1 and system of equations (9)).
2) For a given end effector configuration of a redundant manipulator, although there exists multiple IKs for each of them, position of the wrist (i.e. point D, origin of frames 6 and 7) is same for all IKs.
3) Point C (location of the origins of frames 4 and 5) will always lie on a circle (we call it redundancy circle) if an IK solution exists. This circle is defined by the intersection of the two spheres with radii OC and CD (more details are in section IV-A).
4) Rotation axis of frame 6 (which we denote as $\boldsymbol{z}_6 \in \mathbb{R}^3$) is perpendicular to the tool frame's z axis (we denote as $\boldsymbol{t}_{R_z} \in \mathbb{R}^3$ which is same as z axis of 7^{th} frame) or mathematically $\boldsymbol{z}_6 \perp \boldsymbol{t}_{R_z}$.

Basic idea behind our IK algorithm is as following Given an end effector configuration we first compute position of the wrist using fact (2). Using this information and for a chosen value of θ_1 (within its joint limits), using fact (1) and (3), we obtain redundancy circle and discretize it at n points. For each of those discrete points solve for θ_2, θ_3 as described in section IV-B and use them to find θ_4, θ_5 as in section IV-C. Using computed $\theta_2 - \theta_5$ for a chosen θ_1 if we find fact (4) is satisfied then we compute θ_6, θ_7 as described in section IV-D. We check if all computed θ_i's, $i \in 1, \ldots 7$ are well within their respective joint limits. Next we describe all these steps in detail.

Computing the Wrist Position: Given desired configuration of the end effector (or tool), we can find position of wrist with respect to frame 0 by using d_7 in Table I

$$^{0}\boldsymbol{w}_{pos} = \boldsymbol{t}_{pos} - d_7 * \boldsymbol{t}_{\mathrm{R}_z} \quad (4)$$

where ($^{0}\boldsymbol{w}_{pos} \in \mathbb{R}^3$) denotes coordinates of point D (as in Figure 3) or wrist position with respect to global frame. Note that $\boldsymbol{t}_{pos}$ denotes desired tool position and $\boldsymbol{t}_{R_z}$ is desired direction of z axis of tool frame and can be obtained by extracting 4^{th} and 3^{rd} columns of $^{0}\boldsymbol{T}_{des}$ respectively.

A. Finding Redundancy Circle in Parametrized form

Since $\boldsymbol{w}_{pos}$ or point D in Figure 1 will be same for all IK solutions (as stated in fact (2)), if we fix point O (by choosing a particular value of θ_1), point C has to be on the intersection of the two spheres, one centered at O and radius OC and other centered at D and radius CD. If the spheres do not intersect, it means for that particular choice of θ_1 no IK solution exists. Intersection of the two spheres could be a circle (or a point if links OC and CD are in full stretch configuration), which from now on will be referred to as *redundancy circle*. Next we will discuss a procedure to obtain a parametric representation of the *redundancy circle* assuming that θ_1 is given (*i.e.,* point O is fixed).

For ease of mathematical manipulation we want $^{0}\boldsymbol{w}_{pos}$ to be represented with respect to a suitable frame which we call *artificial frame*. The origin of the *artificial frame* is at

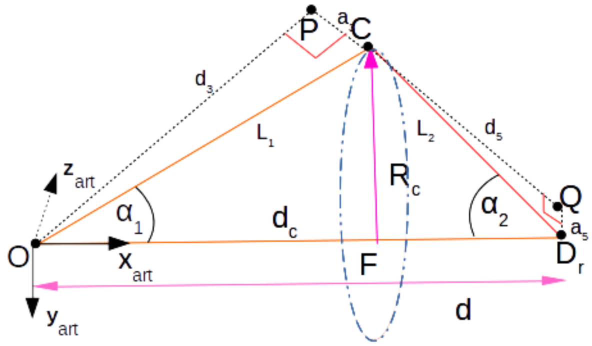

Fig. 2. An instance when desired wrist point D of figure1 lies on x_{art}. The redundancy circle is shown in blue dashed line, centered at point F and radius R_c. d_c and d are distances of points F and D respectively from point O. d determines existence of IK whereas d_c and R_c are useful to discretize redundancy circle as in equation (7)

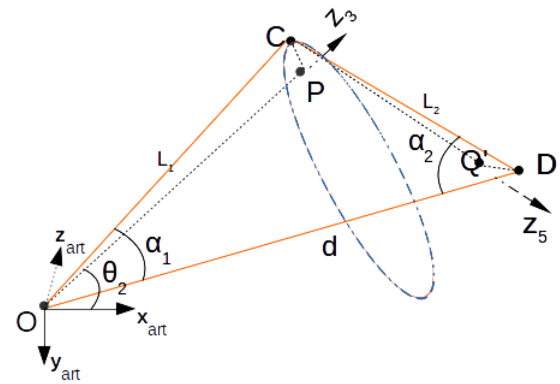

Fig. 3. An instance when wrist point D does not lie on x_{art}. All the names denote similar meanings as given in Figure 2. The redundancy circle (blue dashed line) and triangle OCD (orange line) are rotated as compared to the same in Figure 2.

point O and its x-axis (denoted by x_{art}) is along AO (refer to Figure 1). The transformation of the artificial frame with respect to global frame, ${}^0\boldsymbol{T}_{art}$, is obtained by computing forward kinematics up to joint 2 by choosing $\theta_1 = \phi_i$ where $\theta_1^{min} \leq \phi_i \leq \theta_1^{max}$ as given in equation (5).

$$ {}^0\boldsymbol{T}_{art} = {}^0\boldsymbol{T}_1\, {}^1\boldsymbol{T}_2(\theta_1 = \phi_i) \tag{5}$$

Then position of wrist with respect to *a*rtificial frame,

$$ {}^{art}\boldsymbol{w}_{pos} = {}^0\boldsymbol{R}_{art}^T \left({}^0\boldsymbol{w}_{pos} - {}^0\boldsymbol{p}_{art}\right) \tag{6}$$

If the point D or desired wrist position is on $\boldsymbol{x}_{art}$ (see Figure 2), parametric description of redundancy circle with respect to artificial frame would be

$$ C_i = [d_c,\ -R_c \cos(\psi_i),\ R_c \sin(\psi)], \quad 0 \leq \psi_i \leq 2\pi \tag{7}$$

However in general D will not lie on $\boldsymbol{x}_{art}$ (see Figure 3). In that case parametric representation of redundancy circle is obtained by pre-multiplying $\tilde{\boldsymbol{R}}$ (rotation matrix that aligns vector OD to x_{art} in cases where point D does not lie on x_{art}) with equation (7).

$$ C_i = \tilde{\boldsymbol{R}}^T [d_c,\ -R_c \cos(\psi_i),\ R_c \sin(\psi)], \quad 0 \leq \psi_i \leq 2\pi \tag{8}$$

The term d_c is the distance of the center of redundancy circle along OD whereas R_c is the radius of redundancy circle and one can easily compute them using equation (9).

$$\begin{aligned} d &= \|{}^{art}\boldsymbol{w}_{pos}\|_2,\ L_1 = \sqrt{d_3^2 + a_3^2},\ L_2 = \sqrt{d_5^2 + a_5^2} \\ \alpha_2 &= \arccos(d^2 + L_2^2 - L_1^2/2dL_2), \\ \alpha_1 &= \arcsin(L_2 \sin\alpha_2 / L_1) \\ d_c &= L_1 \cos\alpha_1,\ R_c = L_1 \sin\alpha_1 \end{aligned} \tag{9}$$

Using above parameters one can immediately find a necessary (not sufficient) condition of existence of a valid IK solution as in equation (10) by *cosine law*. If that condition is not satisfied then the desired end-effector configuration has no IK for chosen $\theta_1 = \phi$.

$$ |(d^2 + L_2^2 - L_1^2)/2dL_2| \leq 1 \tag{10}$$

B. *Finding θ_2 and θ_3*

This section assumes that given a desired end-effector configuration one has computed all the parameters until equation (9) and equation (10) is satisfied. Here we describe the method to compute θ_2 and θ_3 for a particular point on the redundancy circle (say i^{th} point, denoted as $C_i \in \mathbb{R}^3$). We need to find θ_2 and θ_3 in such a manner that the position vector of the transformation ${}^2\boldsymbol{T}_3(\theta_2){}^3\boldsymbol{T}_4(\theta_3)$ equals the coordinate of C_i. This would result in a system of three equations with two unknowns, namely θ_2 and θ_3 and is presented in equations (11)-(13). Recall that we parametrized redundancy circle with respect to artificial frame, hence we need to discard pre-multiplication of ${}^0\boldsymbol{T}_1^1\boldsymbol{T}_2(\theta_1)$ with ${}^2\boldsymbol{T}_3(\theta_2){}^3\boldsymbol{T}_4(\theta_3)$ while equating position vector to coordinates of C_i. In the equations below terms $C_{i_x}, C_{i_y}, C_{i_z}$ denote x, y, z components of $C_i \in \mathbb{R}^3$.

$$\begin{aligned} a_3 \sin\theta_3 + d_2 &= C_{i_z} && (11) \\ -(a_3 \cos\theta_3 + a_2)\sin\theta_2 + d_3 \cos\theta_2 &= C_{i_x} && (12) \\ (a_3 \cos\theta_3 + a_2)\cos\theta_2 + d_3 \sin\theta_2 &= C_{i_y} && (13) \end{aligned}$$

The system of equations above is a redundant system of equations. Thus, one can always consider two of them to obtain unique solutions for θ_2 and θ_3. However doing so does not automatically guarantee that the third equation is satisfied. Since our requirement is to find θ_2 and θ_3 that will satisfy all three equations simultaneously, we first solve for θ_3 using equation (11). Then with that θ_3 we solve for $\cos\theta_2$ and $\sin\theta_2$ using equations (12) and (13) respectively. Then we can obtain θ_2 by $\arctan(\sin\theta_2/\cos\theta_2)$ as in equation (14). One can always check for existence of θ_3 by checking $\left|\frac{C_{i_z} - d_2}{a_3}\right| \leq 1$. To check existence of θ_2, one needs to individually check the absolute values of right hand side expressions in equations (15) and (16) respectively whether they are ≤ 1 or not.

$$\theta_3 = \arcsin\left(\frac{C_{i_z} - d_2}{a_3}\right) = \pi - \arcsin\left(\frac{C_{i_z} - d_2}{a_3}\right)$$
$$\theta_2 = \text{atan2}(\sin\theta_2, \cos\theta_2), \quad \text{where,} \tag{14}$$

$$\sin\theta_2 = \frac{d_3 c_y - (a_3\cos\theta_3 + a_2)c_x}{(a_3\cos\theta_3 + a_2)^2 + d_3^2} \tag{15}$$
$$\cos\theta_2 = \frac{d_3 c_x + (a_3\cos\theta_3 + a_2)c_y}{(a_3\cos\theta_3 + a_2)^2 + d_3^2} \tag{16}$$

C. Finding θ_4 and θ_5

The approach for computing θ_4 and θ_5 is similar to that of θ_2 and θ_3. We represent coordinates of point D with respect to frame 4 and equate it with the position vector of the homogeneous transform obtained by ${}^2\boldsymbol{T}_3(\theta_2){}^3\boldsymbol{T}_4(\theta_3)$. After computing ${}^0\boldsymbol{T}_4 = {}^0\boldsymbol{T}_1^1\boldsymbol{T}_2(\theta_1){}^2\boldsymbol{T}_3(\theta_2){}^3\boldsymbol{T}_4(\theta_3)$ we get wrist position D with respect to frame 4 as,

$${}^4\boldsymbol{w}_{pos} = {}^0\boldsymbol{R}_4^T\left({}^0\boldsymbol{w}_{pos} - {}^0\boldsymbol{p}_4\right) \tag{17}$$

We use D'_x, D'_y and D'_z to represent x, y, z coordinates of wrist position in frame 4. Then equating position vector of ${}^2\boldsymbol{T}_3(\theta_2){}^3\boldsymbol{T}_4(\theta_3)$ with ${}^4\boldsymbol{w}_{pos}$ we get,

$$a_5\sin\theta_5 + d_4 = D'_z \tag{18}$$
$$(a_5\cos\theta_5 + a_4)\cos\theta_4 + d_5\sin\theta_4 = D'_x \tag{19}$$
$$(a_5\cos\theta_5 + a_4)\sin\theta_4 - d_5\cos\theta_4 = D'_y \tag{20}$$

Solving the above system of equations for θ_4 and θ_5 we get,

$$\theta_5 = \arcsin\left(\frac{D'_z - d_4}{a_5}\right) = \pi - \arcsin\left(\frac{D'_z - d_4}{a_5}\right)$$
$$\theta_4 = \text{atan2}(\sin\theta_4, \cos\theta_4) \tag{21}$$

We can compute values $\sin\theta_4$ and $\cos\theta_4$ in similar fashion as in equations (15)and (16). We can check for existence of θ_3 using the condition $\left|\frac{D'_z - d_4}{a_5}\right| \leq 1$. The check for existence of θ_4 is analogous to θ_2. So far we have only checked existence of $\theta_1, \ldots, \theta_5$ and computed them if existed. Validity of computed θ_1 to θ_5 is ensured by checking if fact (4) described in section IV, i.e., $\boldsymbol{z}_6 \perp \boldsymbol{t}_{R_z}$ is satisfied. Recall that $\boldsymbol{z}_6 \in \mathbb{R}^3$ and $\boldsymbol{t}_{R_z} \in \mathbb{R}^3$ are first three rows of third columns of ${}^0\boldsymbol{T}_6$ and ${}^0\boldsymbol{T}_{des}$ respectively. Expression of ${}^0\boldsymbol{T}_6$ is given in equation (22).

$${}^0\boldsymbol{T}_6 = {}^0\boldsymbol{T}_1^1\boldsymbol{T}_2(\theta_1){}^2\boldsymbol{T}_3(\theta_2){}^3\boldsymbol{T}_4(\theta_3){}^4\boldsymbol{T}_5(\theta_4){}^5\boldsymbol{T}_6(\theta_5) \tag{22}$$

D. Finding θ_6 and θ_7

This part assumes that one has already computed a valid set of $\theta_1, \ldots, \theta_5$ following the validity check described in section IV-C. A valid set of θ_1 through θ_5 implies that using them, the robot wrist can be positioned at ${}^0\boldsymbol{w}_{pos}$ and can be oriented to achieve desired tool frame orientation. We need to find θ_6 in such a manner that would result in third column of ${}^0\boldsymbol{T}_7$ same as $\boldsymbol{t}_{R_z}$. The idea here is to define a temporary frame named ${}^0\boldsymbol{T}_7^{temp}$ and defined as,

$${}^0\boldsymbol{T}_7^{temp} = {}^0\boldsymbol{T}_1^1\boldsymbol{T}_2(\theta_1){}^2\boldsymbol{T}_3(\theta_2){}^3\boldsymbol{T}_4(\theta_3){}^4\boldsymbol{T}_5(\theta_4){}^5\boldsymbol{T}_6(\theta_5){}^6T_7(0)$$

Then we can compute θ_6 as,

$$\theta_6 = \arccos\left(\boldsymbol{z}_7^{temp} \cdot \boldsymbol{t}_{R_z}\right) \tag{23}$$

where $\boldsymbol{z}_7^{temp}$ and $\boldsymbol{t}_{R_z}$ are 3×1 column vectors obtained by extracting first three rows of third column of ${}^0\boldsymbol{T}_7^{temp}$ and ${}^0\boldsymbol{T}_{des}$ respectively. As θ_6 aligns $\boldsymbol{z}_7$ along $\boldsymbol{t}_{R_z}$, the angle θ_7 is responsible for orienting x and y axis of tool frame along $\boldsymbol{t}_{R_x}$ and $\boldsymbol{t}_{R_y}$ respectively. We can compute θ_7 in similar manner to θ_6 using:

$$\theta_7 = \arccos\left(\boldsymbol{x}_7 \cdot \boldsymbol{t}_{R_x}\right) \tag{24}$$

One can find reduced range of θ_1 instead of searching over all its range from $(w_x - a_1\cos\theta_1)^2 + (w_y - a_1\sin\theta_1)^2 + (w_z - d_1) < (L_1 + L_2)^2$ where w's are components of desired wrist point with respect to *base* frame and a_1 and d_1 are from first row of Table I. Also $L_1 = \sqrt{d_3^2 + a_3^2}$, $L_2 = \sqrt{d_5^2 + a_5^2}$ can be obtained from Table I. The detailed derivation is omitted due to lack of space.

V. ALGORITHM

Algorithm 1 : Input: ${}^0\boldsymbol{T}_{des}$, tol **Output**: Θ

1: Given ${}^0\boldsymbol{T}_{des}$ extract $\boldsymbol{t}_{pos}$ and $\boldsymbol{t}_{R_z}$
2: $\quad\boldsymbol{w}_{pos} \leftarrow \boldsymbol{t}_{pos} - d_7 * \boldsymbol{t}_{R_z}$
3: Discretize θ_1 to n points
4: **for** each θ_1 **do**
5: $\quad$ Compute parameters in equation (9)
6: $\quad$ **if** $|(d^2 + L_2^2 - L_1^2)/2dL_2| \leq 1$ **then**
7: $\quad\quad \boldsymbol{T}_{art} \leftarrow^0 \boldsymbol{T}_1\ {}^1\boldsymbol{T}_2(\theta_1 = \phi_i)$
8: $\quad\quad \boldsymbol{w}_{pos\_art} \leftarrow \boldsymbol{T}_{art}^{-1}\,\boldsymbol{w}_{pos}$
9: $\quad\quad$ Generate C_is using equation (8)
10: $\quad\quad$ **for** Each C_i **do**
11: $\quad\quad\quad$ **if** $\left|\frac{C_{i_z} - d_2}{a_3}\right| \leq 1$ **then**
12: $\quad\quad\quad\quad$ Find θ_2, θ_3 using equations (11)-(13)
13: $\quad\quad\quad\quad$ **if** $\left|\frac{D'_z - d_4}{a_5}\right| \leq 1$ **then**
14: $\quad\quad\quad\quad\quad$ Find θ_4, θ_5 using equations (18)-(20)
15: $\quad\quad\quad\quad\quad$ **if** $z_6 \cdot t_{R_z} \leq tol$ **then**
16: $\quad\quad\quad\quad\quad\quad$ Find θ_6, θ_7 by equations (23)-(24)

Algorithm 1 takes in the desired tool frame configuration ${}^0\boldsymbol{T}_{des}$ and a user defined *tolerance* value *tol* and returns one or more IK solutions. In line 1 we extract the tool position, $\boldsymbol{t}_{pos}$ and tool direction $\boldsymbol{t}_{R_z}$ from ${}^0\boldsymbol{T}_{des}$. Line 2 computes position of wrist, $\boldsymbol{w}_{pos}$ which is same for all joint solutions for a given ${}^0\boldsymbol{T}_{des}$. Next we discretize range of θ_1 at n points in line 3. From line 4 onwards the algorithm finds out possible joint solutions for each value of θ_1. Line 5 computes relevant parameters as in Equation set (9). Now using computed parameters L_1, L_2 and d we check reachability of the desired tool configuration and if that returns true we move to line 7 where $\boldsymbol{T}_{art}$ is computed. Line 8 computes desired wrist position in *artificial frame* and we call this variable ${}^{art}\boldsymbol{w}_{pos}$. Line 9 generates m discretized points (C_i's) on the *redundancy circle*. Line 10 onwards for each C_i, we compute joint angles θ_2 - θ_7 if exists. Existence of θ_2, θ_3 is checked in line 11 whereas existence of θ_4, θ_5 is checked in line 13. If a valid set of θ_1 to θ_5 can be found and condition in line 15 is satisfied, θ_6 and θ_7 is computed.

VI. Comparison with other IK Solvers

In this Section, we apply our Algorithm 1 for computing IK solutions for Baxter robot. We show that we can compute multiple IK solutions for different ${}^0\boldsymbol{T}_{des}$ where existing modules fail to get any solution. Although IKFast is a commonly used IK solver, it fails to compute IK solution for some end-effector configurations of Baxter robot. We have tested our algorithm on such points where IKFast fails and computed multiple IK solutions using Algorithm 1. One such ${}^0\boldsymbol{T}_{des}$ is presented in Equation (25) (configuration of *left_gripper* frame) and shown in Figure 4. In Table II, we provide some sample solutions. Our IK solver (implemented in C++) took $0.05 - 0.07$ seconds to compute one IK solution with accuracy of $1e-5$ on a machine with Intel core $i5$ processor and 8GB memory. We extracted kinematic parameters of Baxter robot from the URDF to perform numerical computations required in Algorithm 1. We have also successfully computed multiple IK solutions using our IK method at points where IK-Service module of Rethink robotics [30] fails.

$$
{}^0\boldsymbol{T}_{des} = \begin{bmatrix} -1 & 0 & 0 & 0.6750 \\ 0 & 1 & 0 & 0.2250 \\ 0 & 0 & -1 & 0.130 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{25}
$$

Fig. 4. Configuration of the end-effector as in equation (25) where IKFast fails to provide any solution whereas our IK produces multiple solutions.

TABLE II
SOME OF THE MULTIPLE IK SOLUTIONS COMPUTED USING ALGORITHM1 FOR ${}^0\boldsymbol{T}_{des}$ IN EQUATIONS (25)

no.	sol1	sol2	sol3	sol4	sol5	sol6
θ_1	-1.4896	-1.3893	-1.2890	-1.1886	-1.0883	-0.5867
θ_2	-0.8570	-0.9577	-1.0229	-1.0669	-1.0962	-1.0945
θ_3	0.9992	0.8138	0.6495	0.4962	0.3494	-0.3595
θ_4	1.5691	1.5843	1.5978	1.6089	1.6174	1.6169
θ_5	-0.6291	-0.4754	-0.3579	-0.2615	-0.1783	0.1837
θ_6	1.2102	1.1525	1.1177	1.0954	1.0810	1.0818
θ_7	0.4130	0.3158	0.2338	0.1614	0.0951	-0.2107

VII. Robust Positioning Using Our IK Method

In this section we show an application where computing multiple IK solutions for a given ${}^0\boldsymbol{T}_{des} \in SE(3)$ is necessary. Suppose we want to place the end-effector in $SE(3)$ at ${}^0\boldsymbol{T}_{des}$. No matter how accurately we compute IK solution (we denote it as $\Theta \in \mathbb{R}^n$, where n is DoF of manipulator), actuation uncertainty will always introduce error in joint-space (we call it $\boldsymbol{\delta\Theta} \in \mathbb{R}^n$). Because of joint-space error, end-effector will end up at a slightly different position (and orientation) after execution of the IK solution. If we can define a bound on joint-space error, position error in end-effector space can be computed as,

$$
\boldsymbol{\delta X} = \boldsymbol{F}_p(\boldsymbol{\Theta} + \boldsymbol{\delta\Theta}) - \boldsymbol{F}_p(\boldsymbol{\Theta}) \approx \boldsymbol{J}_p(\boldsymbol{\Theta})\boldsymbol{\delta\Theta} \tag{26}
$$

where $\boldsymbol{F}_p$ and $\boldsymbol{J}_p$ are *position forward kinematics* map and *position Jacobian* of a manipulator and depends on joint angle (or IK solution) $\boldsymbol{\Theta}$. To bound joint space error we use an n-dimensional ball of radius c as $\boldsymbol{\delta\Theta}^T\boldsymbol{\delta\Theta} \leq c^2$. By assuming individual joint errors, $\delta\theta_i$ where $i \in 1, \ldots, n$, as zero-mean Gaussian and $c = k\sigma$ we can bound different confidence level of joint space error where σ is standard deviation of joint space error and k is number of standard deviation to bound joint space error. Using joint space error and Equation (26), we can get the expression for task space error set as, $\boldsymbol{\delta X}^T \left[\boldsymbol{J}_p(\boldsymbol{\Theta})\boldsymbol{J}_p(\boldsymbol{\Theta})^T\right]^{-1} \boldsymbol{\delta X} \leq c^2$. The error map in task space is an ellipsoid and a function of joint angle $\boldsymbol{\Theta}$. By computing square-root of maximum eigenvalue of $c\boldsymbol{J}_p(\boldsymbol{\Theta})\boldsymbol{J}_p(\boldsymbol{\Theta})^T$, we can get the bound of maximum possible error for a particular $\boldsymbol{\Theta}$. Hence one would be interested in obtaining a particular IK solution that corresponds to minimum task space error bound and we define this as robust-IK ($\boldsymbol{\Theta}^*$). Thus, finding robust-IK for a given ${}^0\boldsymbol{T}_{des}$ is equivalent to solving the following optimization problem,

$$
\boldsymbol{\Theta}^* = \underset{\boldsymbol{\Theta}^* \in \boldsymbol{\Theta}}{\operatorname{argmin}} \; \max_{\lambda} \; \operatorname{eig}\left(c\boldsymbol{J}_p(\boldsymbol{\Theta})\boldsymbol{J}_p(\boldsymbol{\Theta})^T\right) \tag{27}
$$

We find $\boldsymbol{\Theta}^*$ for Baxter robot for ${}^0\boldsymbol{T}_{des}$ in equation (25) by solving min-max problem in equation (27) by searching over multiple IKs computed using Algorithm 1. For $k = 2$, and $\sigma = 0.01$rad, the maximum possible errors for robust and worst case are $0.017m$ and $0.018m$ respectively. This difference of 1 mm can be significant in tight tolerance assembly tasks. We found robust-IK as $\boldsymbol{\Theta}^* = [-1.621, -0.571, 1.348, 1.549, -0.984, 1.396, 0.594]$ (solution truncated after three decimal places).

VIII. Conclusion

We have developed a search based IK by exploiting the geometry of redundant manipulators with several joint offsets. Unlike existing IK methods, our method is guaranteed to find solution if exist while taking care of joint limits. We showed an application which utilizes multiple IK solutions generated by Algorithm 1 to obtain robust-IK resulting in minimum error bound in task space. In future we want to discretize redundancy circle in an adaptive manner to compute solutions faster. We also plan to use our IK method with inverse kinematics based motion planners for robot trajectory planning.

References

[1] Y. Nakamura and H. Hanafusa, "Inverse kinematic solutions with singularity robustness for robot manipulator control," *Journal of Dynamic Systems, Measurement, and Control*, vol. 108, no. 3, 1986.

[2] J. Wang, Y. Li, and X. Zhao, "Inverse kinematics and control of a 7-dof redundant manipulator based on the closed-loop algorithm," *Int. Journal of Adv. Robotics System*, vol. 7, no. 4, pp. 1–10, 2010.

[3] G. K. Singh and J. Classens, "An analytical solution for the inverse kinematics of redundant 7dof manipulator with link offsets," in *IEEE/RSJ International Conference on Intelligent Robot and Systems*, 2010, pp. 2979–2982.

[4] M. Shimizu, H. Kakuya, W. Yoon, K. Kitagaki, and K. Kosuge, "Analytical inverse kinematics computation for 7-dof redundant manipulators with joint limits and its application to redundancy resolution," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1131–1142, October 2008.

[5] D. Zhou, L. Ji, Q. Zhang, and X. Wei, "Practical analytical inverse kinematic approach for 7-dof space manipulators with joint and attitude limits," *Intelligent Service Robotics*, vol. 8, pp. 215–224, 2015.

[6] Y. Wang and P. Artemiadis, "Closed-form inverse kinematics solution for anthropomorphic motion in redundant robot arms," *Advances in Robotics and Automation*, vol. 2, no. 3, 2013.

[7] P. Dahm and F. Joublin, "Closed form solution for the inverse kinematics of a redundant robot arm," *Ruhr University, Bochum, Internal Report IRINI 97-08*, 1997.

[8] B. Siciliano and O. Khatib, *Handbook of Robotics*. Springer.

[9] C. Wampler, "Manipulator inverse kinematics solutions based on vector formulations and damped least-squares methods," *IEEE Transactions on Systems and Cybernetics*, vol. 16, no. 1, pp. 93–101, 1986.

[10] M. Raghavan and B. Roth, "Kinematic analysis of the 6R manipulator of general geometry," *5th Int. Symp. Robot. Res.*, pp. 263–269, 1990.

[11] D. Manocha and J. Canny, "Real time inverse kinematics for general 6r manipulators," *IEEE International Conference on Robotics and Automation*, 1992.

[12] B. Buchberger, "Applications of grbner bases in non-linear computational geometry," *In: Trends in Computer Algebra, Lect. Notes Comput. Sci.*, vol. 296, pp. 52–80, 1989.

[13] M. Tarokh, "Fast inverse kinematics computation by decomposition, classification and approximation," *Research Report No. CSRR-07*, October 2006.

[14] M. Tarokh and M. Kim, "Inverse kinematics of 7-dof robots and limbs by decomposition and approximation," *IEEE Transactions on Robotics*, vol. 23, no. 3, pp. 595–600, October 2007.

[15] D. Baker and C. Wampler, "On the inverse kinematics of redundant manipulators," *Int. J. Robot. Res.*, vol. 7, no. 2, pp. 3–21, 1988.

[16] J. Burdick, "On the inverse kinematics of redundant manipulators: characterization of the self-motion manifolds," *IEEE Int. Conf. Robot. Autom.*, pp. 264–270, 1989.

[17] S. Chiaverini, O. Egeland, and R. Kanestrm, "Weighted damped least-squares in kinematic control of robotic manipulators," *Adv. Robot.*, vol. 7, pp. 201–218, 1993.

[18] P. Chang, "A closed-form solution for inverse kinematics of robot manipulators with redundancy," *IEEE J. Robot. Autom.*, vol. 3, pp. 393–403, 1987.

[19] K. Hauser, "Continuous pseudoinversion of a multivariate function: Application to global redundancy resolution," *12th International Workshop on the Algorithmic Foundations of Robotics*, 2016.

[20] D. Rakita, B. Mutlu, and M. Gleicher, "Relaxedik: Real-time synthesis of accurate and feasible robot arm motion," *Robotics: Science and Systems*, June 26-30, 2018 2018.

[21] C. Klein and C. Huang, "Review of pseudoinverse control for use with kinematically redundant manipulators," *IEEE Trans. Syst.*, vol. 13, pp. 245–250, 1983.

[22] G. Cheng and K. Gupta, "A study of robot inverse kinematics based upon the solution of differential equations," *J. Robot. Syst.*, vol. 8, no. 2, pp. 115–175, 1991.

[23] K. Shankar, J. W. Burdick, and N. H. Hudson, "A quadratic programming approach to quasi-static whole-body manipulation," *Algorithmic Foundations of Robotics XI*, vol. 107, pp. 553–570, 2015.

[24] T. Nammoto and K. Kosuge, "An analytical solution for a redundant manipulator with seven degrees of freedom," *International journal of Automation and Smart Technology*, vol. 2, no. 4, pp. 339–346, October 2012.

[25] T. Asfour and R. Dillmann, "Human-like motion of a humanoid robot arm based on a closed-form solution of the inverse kinematics problem," in *Intl. Conference on intelligent Robots and Systems*, Las Vegas, Nevada, October 2003.

[26] R. C. Luo, T. wei, and Y. Tsai, "Analytical inverse kinematics solution for modularized 7-dof redundant manipulators with offsets at shoulder and wrist," in *IEEE/RSJ international Conference on Intelligent Robots and Systems*, 2014, pp. 516–521.

[27] R. Diankov, "Automated construction of robotic manipulation programs," Ph.D. dissertation, Carnegie Mellon University, 2010.

[28] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to robotic Manipulation*. CRC Press, 1994.

[29] J. J. Craig, *Introduction to Robotics: Mechanics and Control*, 2005.

[30] "Baxter research robot." [Online]. Available: http://www.rethinkrobotics.com/baxter