Proceedings of the 2015
IEEE Conference on Robotics and Biomimetics
Zhuhai, China, December 6-9, 2015

A Novel Kinematics Analysis for a 5-DOF Manipulator Based on KUKA youBot*

Yaolun ZHANG[1], Yangmin LI[1,2], *Senior Member, IEEE* and Xiao XIAO[1]

***Abstract*— Inverse kinematics solutions can provide useful data for trajectory planning, motion control, etc. In this paper, a hybrid algebraic and analytical geometric solution with reservation method is proposed to solve the inverse kinematics, and the existence of all the possible solutions are specified. Therefore, without the selection and matching of multiple solutions for each joint, we can directly obtain the uniqueness of multiple solutions, according to the desired configuration in industrial application. Hence, for the method herein, it is not necessary to calculate all the possible solutions, therefore, compared with other methods, the computational efficiency is greatly enhanced. Finally, the proposed method is verified through simulation and experiment based on a KUKA youBot manipulator.**

***Index Terms*— KUKA youBot manipulator, Inverse kinematics, Hybrid algebraic and analytical geometric, Reservation method, Serial manipulator.**

I. Introduction

Currently “Industry 4.0" has been a hot topic, which was first used at the Hanover Fair in 2011, while in China, “Made in China 2025" was also debuted in May this year. In industrial applications, the specific tasks are usually planned in Cartesian space, so it is inevitable to transform it into the corresponding joint space to control each joint of the manipulator to execute the desired tasks. Inverse kinematics is of fundamental importance [1] [2], and in order to obtain the inverse kinematics, forward kinematics should be acquired firstly [3] [4]; The latter can be computed in a unique manner and only one solution will be obtained, while the former is quite complicated. In industrial applications, manipulators with 5-DOF or 6-DOF usually are manufactured with structures that closed-form solutions exist, namely, three consecutive joint axes intersecting at a point and three consecutive parallel axes [5]. As we know, a minimum of 6-DOF is required to bring its end-effector into any orientation and any position in space. While kinematically deficient manipulator can not approach the desired path in any arbitrary orientation, and it is more difficult to follow a given path with a various orientation and position in the workspace, because the unreachable poses exist, and result in no analytical solutions for inverse kinematics. Although it is impossible to use 5-DOF manipulator to completely satisfy the desired position and orientation that a manipulator with 6-DOF can achieve, when used for welding, grinding, assembling and other applications, the 5-DOF manipulator has some advantages over the 6-DOF manipulator, such as, simpler control, better stiffness of the whole arm, less expensive, etc.

The inverse kinematics for serial manipulators is formed between Cartesian space and joint space, which involves nonlinear and transcendental equations, so it is usually difficult and complicated to solve them. In addition, the inverse kinematics for serial manipulator usually has multiple solutions for each joint, and even for some particular manipulators, it is not always possible to obtain the closed-form solutions. The popular methods are still based on look-up tables, which are designed in a manual manner, and through teach pendant to move manipulator to the desired taught points [6]. Meanwhile, other methods are mainly numerical methods, such as the Newton-Raphson method, neural networks [7] [8], genetic algorithm [9], evolutionary approach [10], optimal search [11] and hybrid search [12], etc., and they often encounter some severe problems, so that sometimes satisfactory and desired solution cannot be obtained. In classic robotic books, there mainly exist two strategies: closed-form solution and numerical solution. The former one consists of algebraic and geometric, and is faster than the latter, moreover, all possible solutions can be identified; the latter one usually adopts the general iterative method, which can just find one solution for one set of starting values; it is much slower than the corresponding closed-form solution, and even fails to converge, so that it can’t be determined whether the inverse kinematics may have a solution or not. Therefore, closed-form solution method is preferred. Liu et al. presented a novel approach for manipulators with Pieper-criterion based geometry based on orthogonal matrix, dot production operation and block matrix [13]. Gan et al. proposed a complete analytical solution to Pioneer-arm(5-DOF) manipulator, but it cannot decide which one is the desired solution [14]. He presented geometric-analytical algorithm to solve the inverse kinematics of PUMA-560, which greatly reduced the computation of transcendental function [15]. Sariyildiz et al.

This research was sponsored by National Natural Science Foundation of China (51575544, 51275353), Macao Science and Technology Development Fund (108/2012/A3, 110/2013/A3), Research Committee of University of Macau (MYRG2015-00194-FST, MYRG203(Y1-L4)-FST11-LYM).

[1] Y.L. ZHANG, Y.M. LI and X. XIAO are with Department of Electromechanical Engineering, Faculty of Science and Technology, University of Macau, Taipa, Macao SAR, China.

[2] Y.M. LI is also a visiting Chair Professor with Tianjin Key Laboratory for Advanced Mechatronic System Design and Intelligent Control, Tianjin University of Technology, Tianjin 300384, China. Corresponding author e-mail: ymli@umac.mo.

978-1-4673-9675-2/15/$31.00 © 2015 IEEE

compared three inverse kinematic formulation methods for serial manipulator, namely, quaternion algebra, dual-quaternion algebra, and exponential mapping method (matrix algebra) [16]. Manseur solved the inverse kinematics of all 5-DOF robot manipulators by using an one-dimensional iterative technique, which is similar to Newton-Raphson [17].

This paper is organized as follows: Section 2 presents some details of KUKA youBot platform and coordinate frame assignation. Section 3 introduces the forward kinematics analysis, then the inverse kinematics will be settled in Section 4. Simulation and experimental results are provided in Section 5. Finally, the paper is concluded in Section 6.

II. Coordinate frames for KUKA youBot

The KUKA youBot is a “desktop" mobile manipulator (`http://www.youbot-store.com/`), which is a partly completed machinery without any preinstalled software, and a great deal of software can be available on KUKA youBot store, so it offers us maximum experimental freedom with industry-based research and development departments [18]. The KUKA youBot consists two parts: one is omni-directional mobile platform, and the other one is a 5-DOF arm with two-finger gripper (Fig.1).

Fig. 1. KUKA youBot

In order to clearly describe the kinematical model of the 5-DOF manipulator, Denavit-Hartenberg method is popular in robotic field, which uses four quantities for each link. A modified D-H method is adopted in this paper, and the base frame is affixed to the base of the manipulator. Frame $\{0\}$ coincides with frame $\{1\}$ to simplify the matters involved in the following analysis. Figure 2 shows the details about the KUKA youBot manipulator's coordinate frames, and also denotes the initial configuration. Frame $\{5\}$ is an auxiliary frame attached to the end-effector, which also denotes the orientation and position of end-effector, and is consistent with each discrete point's desired pose of the planned path.

III. Kinematic Analysis

A. Forward Kinematics

Firstly, the forward kinematics (direct kinematics) is considered, which is used to compute the position and orientation of end-effector relative to the base when the joint angles of each joint are given.

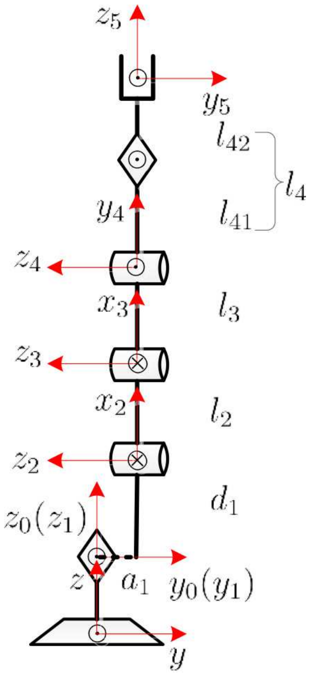

Fig. 2. KUKA youBot arm frames

TABLE I
KUKA youBot D-H Parameters

link	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	$\pi/2$	a_1	d_1	$\pi/2+\theta_2$
3	0	l_2	0	θ_3
4	0	l_3	0	$-\pi/2+\theta_4$
5	$-\pi/2$	0	l_4	θ_5

Based on the D-H convention, the transformation matrix from joint i to joint $i+1$ is given by

$$
{}^{i-1}_{\ \ i}T = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1}d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1}d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}
$$

where $c\alpha$ is shorthand for $\cos\alpha$, $s\alpha$ for $\sin\alpha$, and so on. Using the link parameters shown in Table (I), and substituting the parameters into equation (1), then the individual link transformation for each link from the base to end-effector can be computed.

Therefore, the kinematics from joint space to Cartesian space can be calculated by matrix multiplication of the individual link matrices as follows: ${}^B_5T = {}^B_0T\,{}^0_1T\,{}^1_2T\,{}^2_3T\,{}^3_4T\,{}^4_5T$.

while ${}^B_5T$ represents the orientation and position of end-effector relative to the base of the robot, ${}^0_5T$ represents the pose of end-effector relative to frame $\{0\}$, which will be used as the forward kinematics for the following inverse kinematics in this paper, in addition, ${}^B_5T$ can be obtained through multiplying ${}^0_5T$ by ${}^B_0T$, which is constant.

$$
{}^0_5T = {}^0_1T\,{}^1_2T\,{}^2_3T\,{}^3_4T\,{}^4_5T \tag{2}
$$

then we can obtain the product of all the five-link transformations.

$$
{}^0_5T = \begin{bmatrix} c\theta_{234}c\theta_1 c\theta_5 - s\theta_1 s\theta_5 & -c\theta_5 s\theta_1 - c\theta_{234}c\theta_1 s\theta_5 & -s\theta_{234}c\theta_1 & -c\theta_1(l_3 s\theta_{23} + l_2 s\theta_2 + l_4 s\theta_{234} - a_1) \\ c\theta_1 s\theta_5 + c\theta_{234}c\theta_5 s\theta_1 & c\theta_1 c\theta_5 - c\theta_{234}s\theta_1 s\theta_5 & s\theta_{234}s\theta_1 & -s\theta_1(l_3 s\theta_{23} + l_2 s\theta_2 + l_4 s\theta_{234} - a_1) \\ s\theta_{234}c\theta_5 & -s\theta_{234}s\theta_5 & c\theta_{234} & d_1 + l_3 c\theta_{23} + l_2 c\theta_2 + l_4 c\theta_{234} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)
$$

where $c\theta_{234}$ is shorthand for $\cos(\theta_2 + \theta_3 + \theta_4)$, $s\theta_{234}$ for $\sin(\theta_2 + \theta_3 + \theta_4)$, and so on.

If the orientation and position of end-effector relative to frame$\{0\}$ are given, then we use $\boldsymbol{n} = [n_x, n_y, n_z]^T$, $\boldsymbol{o} = [o_x, o_y, o_z]^T$, and $\boldsymbol{a} = [a_x, a_y, a_z]^T$ to represent the orientation in terms of D-H convention, while $\boldsymbol{P} = [P_x, P_y, P_z]^T$ for the position.

B. Inverse Kinematics

Here is a completely closed-form solution, and it establishes all the possible solutions, what's more, without calculating all the possible solutions, we can select the optimal solution among them.

For specific tasks, the orientation and position of end-effector is given or has been planned during trajectory planning, and the pose of end-effector is relative to the goal pose, therefore we will solve the following equation

$$
OP_{ee} = \begin{bmatrix} \boldsymbol{n} & \boldsymbol{o} & \boldsymbol{a} & \boldsymbol{P} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)
$$

$$
OP_{ee} = {}^0_5T = {}^0_1T {}^1_2T {}^2_3T {}^3_4T {}^4_5T \quad (5)
$$

We solve the inverse kinematics as follows: firstly, we settle the controlling joint, i.e. joint 1 and middle joint, i.e. joint 3 through analytical geometric method, then other joints will be settled through algebraic method, and for last joint for KUKA youBot, the method is proposed in this paper with two methods, namely, it is calculated based on the goal pose for grasping operation.

1) Solution for θ_1: The position of joint 4 can be obtained through the position of end-effector and the approach vector $\boldsymbol{a}$, so we have

$$
P_4 = P_5 - l_4\boldsymbol{a} \quad (6)
$$

meanwhile, we establish a virtual joint above the controlling joint (joint 1), for the position of the virtual joint about joint 2 for KUKA youBot (note that the virtual joint is not a necessary one here, however in order to meet the proposed method, so in this paper, we can also use the position of joint 1 to settle the corresponding problem), we can have

$$
P_{2_{vir}} = [0, 0, d_1]^T \quad (7)
$$

then, we can obtain a vector $\boldsymbol{P}_{24_{vir}}$ through the position of joint 4 and the virtual joint about joint 2.

$$
\boldsymbol{P}_{24_{vir}} = P_4 - P_{2_{vir}} \quad (8)
$$

Here we adopt the function $Atan2$, which is a two-argument arc tangent function, and not only can compute $tan^{-1}(\frac{y}{x})$, but also can use the signs of both x and y to identify the quadrant in which the resulting angle lies with a full 360°, so it is very convenient to settle the vector projection in the XOY plane.

Through the vector projection method, we can obtain the projection of $\boldsymbol{P}_{24_{vir}}$ onto XOY plane, so the solution for θ_1 can be calculated as

$$
\theta_{1f} = Atan2(\boldsymbol{P}_{24_{vir}}(2), \boldsymbol{P}_{24_{vir}}(1)) \quad (9)
$$

Note that there exist two possible solutions for θ_1, and the other one can be obtained through phase transformation, which is central symmetrical with respect to the origin of XOY, namely,

$$
\theta_{1b} = \theta_{1f} - sgn(\theta_{1f})\pi \quad (10)
$$

where θ_{1f} denotes the calculated solution and $sgn(\theta_{1f})$ denotes the sign ($+$ or $-$) of the calculated solution for θ_1. Here, the former solution of joint 1 (θ_{1f}) is named the projection solution, while the latter (θ_{1b}) the phase solution. In addition, if just for simple application, you can also use the formula $\theta_{1f} = Atan2(P_y, P_x)$ to calculate the solution for joint 1.

2) Solution for θ_3: From the previous section, we have obtained the value of joint 1, then we can calculate the real position of joint 2, namely,

$$
P_{2f} = [a_1 c\theta_{1f}, a_1 s\theta_{1f}, d_1]^T \quad (11)
$$

then, the vector $\boldsymbol{P}_{24f}$ can be calculated as follows:

$$
\boldsymbol{P}_{24f} = P_4 - P_{2f} \quad (12)
$$

therefore, the distance between joint 2 and joint 4 is the vector norm for $\boldsymbol{P}_{24f}$, namely, $lP_{24f} = |\boldsymbol{P}_{24f}|$.

Meanwhile, corresponding to the phase solution of joint 1, we can obtain another position of joint 2, which is different from the previous one.

$$
P_{2b} = [a_1 c\theta_{1b}, a_1 s\theta_{1b}, d_1]^T \quad (13)
$$

Corresponding to the phase solution of joint 1, the vector from joint 2 to joint 4 is calculated as follows:

$$
\boldsymbol{P}_{24b} = P_4 - P_{2b} \quad (14)
$$

and the distance between joint 2 and joint 4 is to calculate the vector norm for $\boldsymbol{P}_{24b}$, namely, $lP_{24b} = |\boldsymbol{P}_{24b}|$.

Based on the position of joint 2, joint 4 and joint 3, we can get a spatial triangle (Fig.3). From the law of cosines, we obtain the supplementary angle of θ_3.

$$
\cos\theta_{3_{supp}} = \frac{l_2^2 + l_3^2 - lp_{24f}^2}{2l_2 l_3} \quad (15)
$$

then, the joint value for joint 3 is calculated as

$$
\theta_{3fu} = -(\pi - \theta_{3_{supp}}) \quad (16)
$$

At the same time, there exists another configuration to execute the tasks as shown in Fig.3 using the dashed lines.

$$
\theta_{3fd} = \pi - \theta_{3_{supp}} \quad (17)
$$

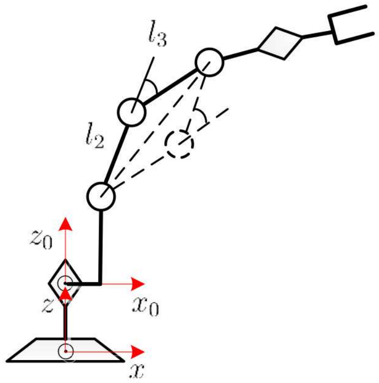

Fig. 3. youBot middle configuration

Corresponding to the phase solution of joint 1, the other two solutions of joint 3 can be obtained, namely, θ_{3bu} and θ_{3bd}.

3) Soluton for θ_2: A restatement of equation (5) that puts the dependence on θ_1 on the left-hand side of the equation is

$$({}^0_1T)^{-1}OP_{ee} = {}^1_2T{}^2_3T{}^3_4T{}^4_5T \tag{18}$$

inverting ${}^0_1T$,

equating the $(1,3)$ and $(3,3)$ elements from both sides of (18), we obtain

$$\begin{aligned} s\theta_{234} &= -(c\theta_{1f}a_x + s\theta_{1_{call}}a_y) \\ c\theta_{234} &= a_z \end{aligned} \tag{19}$$

From equation (5) and (3), namely, $OP_{ee} = {}^0_5T$, equating the $(1,4)$, $(2,4)$ and $(3,4)$, we obtain

$$\begin{aligned} -P_x/c\theta_1 - l_4s\theta_{234} + a_1 &= l_3s\theta_{23} + l_2s\theta_2 \\ -P_y/s\theta_1 - l_4s\theta_{234} + a_1 &= l_3s\theta_{23} + l_2s\theta_2 \\ P_z - d_1 - l_4c\theta_{234} &= l_3c\theta_{23} + l_2c\theta_2 \end{aligned} \tag{20}$$

Note that the elimination method is usually adopted to solve the above functions, then $\theta_2 = \theta_{23} - \theta_3$. Compared with it, the reservation method is proposed to settle it here, which will help to avoid the following selection and matching from multiple solutions.

Here we make the following substitutions

$$\begin{aligned} var_{2_x} &= -P_x/c\theta_{1f} - l_4s\theta_{234} + a_1 \\ var_{2_y} &= -P_y/s\theta_{1f} - l_4s\theta_{234} + a_1 \\ var_{2_z} &= P_z - d_1 - l_4c_{234} \\ c_{aux1} &= l_2 + l_3c\theta_{3_{cal1}} \\ s_{aux1} &= l_3s\theta_{3_{cal1}} \end{aligned} \tag{21}$$

Let

$$c^2_{aux1} + s^2_{aux1} = l_2^2 + l_3^2 + 2l_2l_3c\theta_3 \tag{22}$$

as for industrial manipulator, generally the length of upper arm differs from the length of the forearm, and due to the stiffness of the whole arm, the former one is longer than the latter one, so $c^2_{aux1} + s^2_{aux1} \geqq 0$; if and only if $\theta_3 = \pi$ and $l_2 = l_3$, $c^2_{aux1} + s^2_{aux1} = 0$.

hence

$$l_2^2 + l_3^2 - 2l_2l_3 = (l_2 - l_3)^2 \not\equiv 0 \tag{23}$$

Therefore

$$\begin{aligned} var_{2_x} &= (l_3c\theta_3 + l_2)s\theta_2 + l_3s\theta_3c\theta_2 \\ var_{2_y} &= (l_3c\theta_3 + l_2)s\theta_2 + l_3s\theta_3c\theta_2 \\ var_{2_z} &= (l_3c\theta_3 + l_2)c\theta_2 - l_3s\theta_3s\theta_2 \end{aligned} \tag{24}$$

a) Solution with trigonometric function: Simplifying

$$\begin{aligned} s(\theta_{2_1} + \varphi_1) &= \frac{var_{2_x}}{\sqrt{c^2_{aux1} + s^2_{aux1}}} \\ c(\theta_{2_1} + \varphi_1) &= \frac{var_{2_z}}{\sqrt{c^2_{aux1} + s^2_{aux1}}} \end{aligned} \tag{25}$$

then

$$\begin{aligned} \varphi_1 &= Atan2(s_{aux1}, c_{aux1}) \\ \theta_{2_1} + \varphi_1 &= Atan2(s(\theta_{2_1} + \varphi_1), c(\theta_{2_1} + \varphi_1)) \end{aligned} \tag{26}$$

we obtain

$$\begin{aligned} \theta_{2fu} &= \theta_{2_1} + \varphi_1 - \varphi_1 \\ \theta_{2fu} &= \theta_{2fu} + sgn(\theta_{2fu})(1 - sgn(\frac{3}{2}\pi - |\theta_{2fu}|))\pi \end{aligned} \tag{27}$$

Similarly, we can obtain another solution, namely, θ_{2fd} corresponding another configuration under the projection solution of joint 1.

In the same manner, corresponding to the phase solution of joint 1, we can also obtain two solutions for θ_2, namely, θ_{2bu} and θ_{2bd}, but note that sometimes these two solutions may not always exist as well as θ_{3bu} and θ_{3bd}.

b) Solution with equation set: The above-mentioned method can successfully settle the problem about θ_2, and the range of it is $[-\frac{\pi}{2}, \frac{\pi}{2}]$. Although it meets most of industrial manipulators, and can calculate the joint variables, which will not exceed its joint range, however there still exist some manipulators, and their joint range exceeds the range of $[-\frac{\pi}{2}, \frac{\pi}{2}]$, so another method is proposed to satisfy the requirement.

Based on the equation (24), we can obtain

$$\begin{aligned} c\theta_2 &= \frac{l_3s\theta_3var_{2x} + (l_3c\theta_3 + l_2)var_{2z}}{l_3^2 + l_2^2 + 2l_2l_3c\theta_3} \\ s\theta_2 &= \frac{l_3c\theta_3 + l_2c\theta_2 - var_{2z}}{l_3s\theta_3} \end{aligned} \tag{28}$$

$$\theta_2 = Atan2(s\theta_2, c\theta_2) \tag{29}$$

Similarly, we can calculate the other joint values.

4) Solution for $\theta 4$: A restatement of equation (5) that puts the dependence on θ_1, θ_2 and θ_3 on the left-hand side of the equation is

$$({}^2_3T)^{-1}({}^1_2T)^{-1}({}^0_1T)^{-1}{}^0_5T = {}^3_4T{}^4_5T \tag{30}$$

Inverting ${}^0_1T$, ${}^1_2T$ and ${}^2_3T$, equating the $(1,3)$ and $(2,3)$ elements from both side of (30), we can obtain

$$\begin{aligned} s\theta_{4fu} = &- c(\theta_{2fu} + \theta_{3_{cal1}})c\theta_{1_{cal1}}a_x - c(\theta_{2fu} + \theta_{3_{cal1}}) \\ &s\theta_{1_{cal1}}a_y - s(\theta_{2fu} + \theta_{3_{cal1}})a_z \\ c\theta_{4fu} = &- s(\theta_{2fu} + \theta_{3_{cal1}})c\theta_{1_{cal1}}a_x - s(\theta_{2fu} + \theta_{3_{cal1}}) \\ &s\theta_{1_{cal1}}a_y + c(\theta_{2fu} + \theta_{3_{cal1}})a_z \end{aligned} \tag{31}$$

$$\theta_{4fu} = Atan2(s\theta_{4fu}, c\theta_{4fu}) \tag{32}$$

Corresponding to another configuration, the other solution for θ_4 is calculated, namely, θ_{4fd}.

Similarly, the other two solutions corresponding to the phase solution of joint 1 can be calculated, namely, θ_{4bu} and θ_{4bd}.

5) Solution for θ_5: Usually, the joint 5 is calculated through a restatement of equation (5) that puts the dependence on θ_1 on the left-hand side of the equation, namely, equation (18). We can find the details about joint 5.

$$\begin{aligned} s\theta_5 &= -s\theta_1 n_x + c\theta_1 n_y \\ c\theta_5 &= -s\theta_1 o_x + c\theta_1 o_y \end{aligned} \tag{33}$$

$$\theta_5 = Atan2(s\theta_5, c\theta_5) \tag{34}$$

Similarly, other solutions can be obtained.

Here, the self-optimizing method is proposed to settle this problem, namely, the joint solution for the self-optimizing joint is calculated based on the goal pose, especially when the goal is a moving object, so the pose of it might be time-varying, and the robot can be more efficient to grasp it with as less adjustment as possible, because it is calculated to meet the grasp requirement.

From equation (3), we can obtain $\boldsymbol{o}$; for specific task, the goal pose should be given or planned.

Let

$$OP_d = \begin{bmatrix} \boldsymbol{n}_d & \boldsymbol{o}_d & \boldsymbol{a}_d & \boldsymbol{P}_d \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{35}$$

For grasping operation, two methods are proposed here. The approach vector $\boldsymbol{a}$ will be related to the Z direction of the goal frame, then the orientation vector $\boldsymbol{o}$ will be consistent with the Y direction of the goal frame, which can be the same as the direction of $\boldsymbol{o}$ or the opposite direction of it, and both of them can achieve the desired tasks.

a) : The sine and cosine values of joint 5 can be calculated as follows:

$$\begin{aligned} s\theta_5 &= \frac{o_x c\theta_1 + o_y s\theta_1}{-c\theta_{234}} \\ s\theta_5 &= \frac{o_z}{-s\theta_{234}} \\ c\theta_5 &= \frac{o_y - s\theta_1(o_x c\theta_1 + o_y s\theta_1)}{c\theta_1} \\ c\theta_5 &= \frac{c\theta_1(o_x c\theta_1 + o_y s\theta_1) - o_x}{s\theta_1} \end{aligned} \tag{36}$$

According to the planned tasks, the angle of joint 1 is determined, so we can obtain the desired sine and cosine values of joint 5, so that it avoids that denominator is zero. Therefore the solution for joint 5 is

$$\theta_{5so} = Atan2(s\theta_5, c\theta_5) \tag{37}$$

b) : Another method will be presented in the following, which is easier to use. According to the calculated joint angles θ_1, θ_2, θ_3, and θ_4, and through forward kinematics, the pose of end-effector can be calculated as follows:

$${}^0_4T = {}^0_1T\,{}^1_2T\,{}^2_3T\,{}^3_4T \tag{38}$$

The orientation vector of end-effector through four joint values can be obtained by

$$\boldsymbol{o}_{int} = [-s\theta_{234}c\theta_1, -s\theta_{234}s\theta_1, c\theta_{234}]^T \tag{39}$$

while the desired orientation is $\boldsymbol{o}_d = [o_{x_d}, o_{y_d}, o_{z_d}]^T$, so we can calculate the angle between $\boldsymbol{o}_{int}$ and $\boldsymbol{o}_d$

$$\cos\theta_5 = \frac{\boldsymbol{o}_{int}\boldsymbol{o}_d}{|\boldsymbol{o}_{int}||\boldsymbol{o}_d|} \tag{40}$$

$$\theta_{5so} = \arccos(\cos\theta_5) \tag{41}$$

In summary, without the selection and matching of multiple solutions, we can specify four solutions for KUKA youBot individually, and choose that which one is the optimal solution. In industrial application, due to the mechanical constraints, we usually adopt the elbow-up configuration, hence under this condition, θ_{3fu} related with the projection solution of joint 1 is the optimal solution. Compared with "closest solution" to choose, the method proposed here is more efficient and less time-consuming. The whole optimal solution in industrial application is as follows: equation (9) denotes the joint angle for joint 1, θ_{1f}; equation (27) or (29) for joint 2, θ_{2fu}; equation (16) for joint 3, θ_{3fu}; equation (32) for joint 4, θ_{4fu}; equation (37) or (41) for joint 5, θ_{5so}. To verify the above-mentioned solutions, simulations and experiments have been carried out, and both of them validate the proposed method.

IV. Simulations and Experiments

To verify the feasibility of the proposed method, experiments are conducted in both simulation and real KUKA youBot. Due to the limited equipment, the real position of the end-effector can not be measured directly and precisely, so for the real experimental result, we will present them with picture and video.

Here is the desired path equation (note that just the desired position is given here, but we carried out both simulation and real experiment with time-varying orientation. Due to the limitation of this paper, the details will be presented in future paper).

$$\begin{aligned} \gamma &= \frac{pi}{50}t \\ x_d &= 380 \\ y_d &= 100\cos(\gamma) \\ z_d &= 260 + 100\sin(\gamma) \end{aligned} \tag{42}$$

As for the figure of simulation and experiment, we just make 10^2 discrete points in the simulation, while in real experiments, to achieve better performance, 10^4 is used to obtain the experimental results and the video is recorded. According to the experiment, if the more discrete points are adopted, the better performance will be obtained, however, with the same preset value, it will be time-consuming correspondingly. The simulated result is shown in Fig.4, while Fig. 5 shows the real experimental results. The experiment carried out in this paper is just to control the robot manipulator to draw a circle on a plane, and we set the joint angle for joint 5 unchanged, so the Fig. 4 shows that the joint angle of

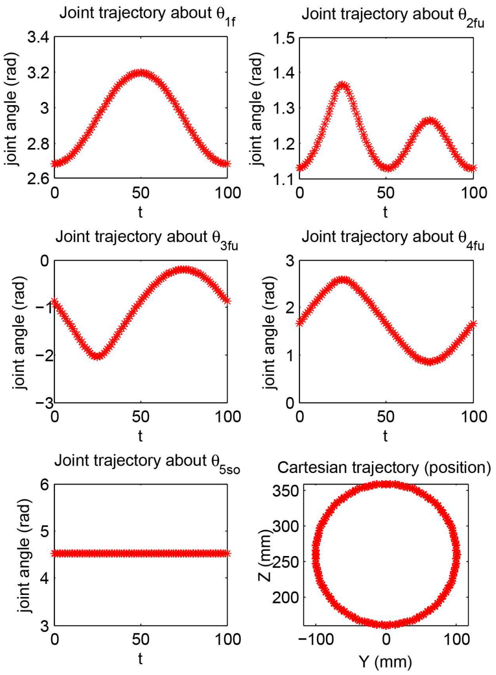

Fig. 4. youBot individual joint angle of simulation

joint 5 remains constant. Both simulation and experimental results validate the proposed method in this paper, which is beneficial for solving the inverse kinematics of 5-DOF manipulators.

V. CONCLUSIONS

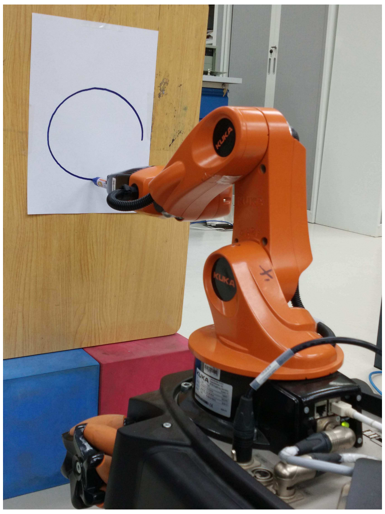

Fig. 5. youBot experimental result

In this paper, a strategy based on a hybrid algebraic and analytical geometric method is proposed to solve the inverse kinematics of a 5-DOF manipulator, in addition, the reservation method is also specified, which helps to simplify the complexity of inverse kinematics, hence, a complete closed-form solution has been derived. Compared with other methods, the one proposed in this paper does not need to check the calculated solutions, and select and match the optimal solution, and just according to the desired configuration, we can directly obtain the optimal solution without calculating all the possible solutions. Due to the limitation of this paper, we also propose the trajectory planning method to deal with the above-mentioned problems in simulation and experiment, and some remained issues will be given in our future work.

REFERENCES

[1] J. -G. Wang, Y.M. Li , and X. Zhao, "Inverse kinematics and control of a 7-DOF redundant manipulator based on the closed-loop algorithm", *Int. J. Adv. Rob. Syst.*, vol. 7, no.4, pp. 1-9, 2010.

[2] J.-G.Wang and Y.M. Li, "Comparative analysis for the inverse kinematics of redundant manipulators based on repetitive tracking tasks", *IEEE Int. Conf. Autom. Log.(ICAL)*, Shenyang, China, August 2009, pp.164-169.

[3] Y.M. Li and Y. G. Liu, "Kinematics and tip-over stability analysis for the mobile modular manipulator" *Proc. Instit. Mech. Eng., J. Mech. Eng. Sci., Part C*, vol. 219, no. 3, pp. 331-342, 2005.

[4] X. Xiao, Y.M. Li and H. Tang, "Kinematics and interactive simulation system modeling for katana 450 robot," *IEEE Int. Conf. Inf. Autom.(ICIA)* , August 26-28, 2013, Yinchuan, Ningxia, China, pp.1177-1182.

[5] D.L. Pieper, “The kinematics of manipulators under computer control,” Ph.D. dissertation, Dept. Computer Science, Stanford Univ., Stanford, CA USA, 1968.

[6] J.J. Craig, “*Introduction to robotics: mechanics and control*," New Jersey: Pearson Prentice Hall, 2005.

[7] E. Oyama, N.Y. Chong, A. Agah, and T. Maeda, “Inverse kinematics learning by modular architecture neural networks with performance prediction networks,” *IEEE Int. Conf. Robot. Autom. (ICRA)*, Seoul, Krea, vol. 1, pp. 1006-1012, 2001.

[8] Y.M. Li and S. H. Leong, "Kinematics control of redundant manipulators using CMAC neural network combined with genetic algorithm", *Robotica*,vol. 22, no. 6, pp.611-621, 2004.

[9] A.C. Nearchou, “Solving the inverse kinematics problem of redundant robots operating in complex environments via a modified genetic algorithm,” *Mech. Mach. Theory*, vol. 33, no. 3, pp. 273-292, 1998.

[10] P. Kalra, P.B. Mahapatra, and D.K. Aggarwal, “An evolutionary approach for solving the multimodal inverse kinematics problem of industrial robots,” *Mech. Mach. Theory*, vol. 41, no. 10, pp. 1213-1229, 2006.

[11] Z.W. Ren, Z.H. Wang, and L.N. Sun, “A hybrid biogeography-based optimization method for the inverse kinematics problem of an 8-dof redundant humanoid manipulator,” *Front. Inf. Technol. Electron. Eng.*, vol. 16, no. 7, pp. 607-616, 2015.

[12] D. Xu, C.A.A. Calderon, J.Q. Gan, H. Hu, “An analysis of the inverse kinematics for a 5-dof manipulator,” *Int. J. Autom. Comput.*, vol. 2, no. 2, pp. 114-124, 2005.

[13] H.S. Liu, W.N. Zhou, X.B. Lai, S. Q. Zhu, “An efficient inverse kinematic algorithm for a PUMA 560-structured robot manipulator," *Int. J. Adv. Robot. Syst.*, vol. 10, pp. 236-240, 2013.

[14] J.Q. Gan, E. Oyama, E.M. Rosales, H. Hu, “A complete analytical solution to the inverse kinematics of the pioneer2 robotic arm,” *Robotica*, vol. 23, pp. 123-129, 2005.

[15] Y.Y. He, “New efficient algorithm for inverse kinematics of robot puma 560,” *Robot*, vol. 11, no. 3, pp. 19-26, 1989.

[16] E. Sariyildiz, E. Cakiray, and H. Temeltas, “A comparative study of three inverse kinematic methods of serial industrial robot manipulators in the screw theory framework,” *Int. J. Adv. Robot. Syst.*, vol. 8, pp. 9-24, 2011.

[17] R. Manseur and K.L. Doty, “Fast inverse kinematics of five-revolute-axis robot manipulators,” *Mech. Mach. Theory*, vol. 27, no. 5, pp. 587-597, 1992.

[18] R. Bischoff, U. Huggenberger, and E. Prassler, “Kuka youbot-a mobile manipulator for research and education,” *IEEE Int. Conf. Robot. Autom. (ICRA)*, Shanghai, China, pp. 1-4, 2011.