Journal of Physics: Conference Series **1449** (2020) 012128 doi:10.1088/1742-6596/1449/1/012128

# Practical Inverse Kinematic Solution Based on Calibrated Parameters Using in Engineering

**Guosong Shi[1,3], Shuying Zhao[1], Tianyu Lin[2]**

[1]College of Information Science and Engineering, Northeastern University, Shenyang 110819, P.R. China.

[2]Quality and R & D Center, Yuanda Intelligent Robot Co.,Ltd. No.27, street 16, Kaifa Road, Economic and Technological Development Zone, Shenyang 110144, P.R. China.

[3]Corresponding author's e-mail address: shiguosong2007@126.com

**Abstract**. Inverse kinematic (IK) problem is an essential problem in industrial robot control. In some high accuracy projects, the solution needs the robot act based on online sampled data rather than the pre-teached actions. It is necessary to reduce the absolute positioning error to complete these challenging projects. The IK solution is on the way there. This article simply introduces the standard IK solution based on robot KUKA R2700, points out the shortage of standard IK solution in enhancing the absolute positioning accuracy and develops an practical IK solution, PIKFCP, which is necessary in some measures.

## 1. Introduction

The advantages of robot and the development of robot technology make robot wildly used in modern industry. While the teach function of robot had covered most of the needs of industrial projects, there still exist some high flexible and variable projects that teach function can not handle. When the solution of project doesn't rely on the quantity but the quality, the robot should act in high stable accurate performance. Namely, the robot should act based on the real time data rather than data simulated in the CAM (Computer Aided Manufacturing) software. As the projects rely on real time data rather than off-line design, we call this kind of projects real time projects.

Unfortunately, the industrial robots have high repeatability, which can only satisfy the teach function, and low absolute positioning accuracy, which is essential in real time projects. The absolute positioning error can be assured for many reasons, and should be solved to be part of real time - high accuracy projects.

There are several ways to enhance the absolute accuracy of a robot and the commonly used way is the calibration of the geometry parameters. Some methods aiming at the DH model which robot used [1-3], others aiming at the modified model and finally compensate the error back to the DH parameters [4,5]. It is believed that the positioning error is caused by the inaccurate structure parameters, and the high accuracy is coming by accurate parameters. The effective way is to calibrate the parameters and let the robot act in calibrated parameters to perform high accurate absolute positioning accuracy. It will be convenient if capable to update the parameters in the robot, or otherwise we can only apply the offline compensation.

## 2. Kinematic Functions of Robot

The forward and inversed kinematic function are used to control the movement of robot. The forward kinematic function uses the joint variables to compute the terminal pose of robot, and the inversed uses compute the joint variables from the terminal pose. These two functions are determined by the model of the robot. A popular robot structure model is DH model which simplify the number of parameters to describe the 6DOF transformation from 6 to 4 which is easily applied to a robot with elaborately designed structure. In robot KUKA R2700 (see figure 1) the joints are all rotated joint and the joint variables are all angles. KUKA robot system use ZY'X'' Euler angles to represent attitude in degree. It uses modified DH model to control the movement of robot, which is a little different from standard DH model in serial linked robot the forward kinematic function is easy to compute, to the contrary, the inversed kinematic is hard to solve and the solitary of solution is no longer exists. There exists closed form IK solution if the structure of a robot satisfies the Pipper Criterion [6] and fortunately the KUKA R2700 does.

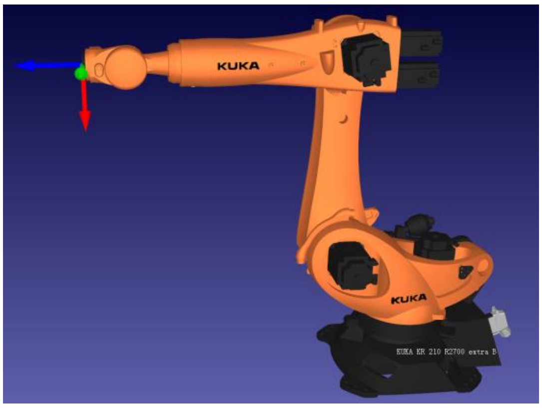


Figure 1. Industrial Robot KUKA KR210 R2700.

### *2.1. Standard DH model IK Solution of R2700*

The DH model uses 4 parameters – θ, d, a, α- to describe one 6-DOF Euler Transformation. Specific in R2700, the θ is the variable to move the body and the other 3 parameters are fixed. So the θi(I = 1,2,…6) are the only variable of kinematic functions of R2700. The IK solution of R2700 can be solved with some intuition of mathematics and structure. We call the IK solution based on the standard model standard IK solution as it is solved based on standard parameters which some length are 0, some angles are 0 or PI/2, see table 1. The wrist point pose of robot can be described as equation (1) in which the Ai means the transformation matrix of the i-th joint and the wrist point is at the end of the 4-th joint. But error occurs while the standard IK solution is applied on calibrated parameters even though there is only a little difference away from 0 or PI/2, the kinematic function becomes very complicated and impossible to get the closed form solution under low computational resource. The equation (2) and equation (3) show the different expressions of x data of wrist point under standard parameters and calibrated parameters. To be noticed, in equation (2), the ci and si mean cos($\theta$i) and sin($\theta$i), while all the values of sin($\alpha$i) or cos($\alpha$i) are 0 or 1 here, and the equation (3) only shows a small part of the complicated expression.

$$T_{wrist} = A_1 * A_2 * A_3 * A_4 = \begin{bmatrix} Rw & \begin{matrix} Pwx \\ Pwy \\ Pwz \end{matrix} \\ 0 & 1 \end{bmatrix} \tag{1}$$

$$\mathrm{Pwx_{standard}} = \mathrm{d4} * (\mathrm{c1c2s3} + \mathrm{c1s2c3}) + \mathrm{a1c1} + \mathrm{a2c2c2} + \mathrm{a3c1c2c4} - \mathrm{a3c2s2s3} \tag{2}$$

$$\mathrm{Pwx}_{\mathrm{original}} = a1 * \cos(\theta1) + d4 * (\cos(\alpha3) * (\cos(\alpha2) * \sin(\alpha1) * \sin(\theta1) + \sin(\alpha2) * \cos(\theta1) * \sin(\theta2) + \cos(\alpha1) * \sin(\alpha2) * \cos(\theta2) * \sin(\theta1)) + \sin(\alpha3) * \sin(\theta3) * (\cos(\theta1) * \cos(\theta2) - \cos(\alpha1) * \sin(\theta1) * \sin(\theta2)) + \sin(\alpha3) * \cos(\theta3) * (\cos(\alpha2) * \cos(\theta1) * \sin(\theta2) - \sin(\alpha1) * \sin(\alpha2) * \sin(\theta1) + \mathit{another}\ 15\ \mathit{lines} \quad (3)$$

Table 1. Standard MDH Parameters of Robot KUKA R2700.

| | $\alpha$(deg) | $a$(mm) | $\theta$(deg) | $d$(mm) |
|---|---|---|---|---|
| Joint 1 | 0.000 | 0.000 | 0.000 | 675.000 |
| Joint 2 | -90.000 | 350.000 | 0.000 | 0.000 |
| Joint 3 | 0.000 | 1150.000 | -90.000 | 0.000 |
| Joint 4 | -90.000 | -41.000 | -0.000 | 1200 |
| Joint 5 | 90.000 | 0.000 | 0.000 | -0.000 |
| Joint 6 | -90.000 | 0.000 | 180.000 | 215.000 |

### *2.2. The Standard Closed-form Solution of Inversed Kinematic subsection*

There exists closed-form solution of the IK problem while the structure of the robot satisfied the Pipper Criterion. Many solutions had been addressed and some of them are solved based on DH or Modified DH model. The closed-form solution had been addressed by some researchers [7-8]. We just simply introduce the solution here.

DH model simplified the structure description from 6 parameters to 4 parameters when the structure of robot subjected to certain limitation. These 4 parameters are marked as alpha, a, d, theta, and they represent one transformation from a joint to its next joint. While the industrial robot is a 6-DOF serial robot that the design structure satisfied the Pipper Criterion, we take KUKA KR2700 robot as example. It has 6 rotation joints and the axis of joint 4-6 are cross at one point which is called wrist point. Then the process of IK problem solution can be simply descripted as follows.

The robot is divided into 2 parts at the wrist point and each part includes 3 joints. The first part determines the position of wrist, and the last part determines the attitude of robot, finally the two parts determines the position and attitude of robot together. We can simply compute the wrist point from the target pose (position and attitude) as it is located at the original point of the coordinate of the 6th joint. Table 2 shows the algorithm of standard IK solution using in our KUKA R2700. Commonly 8 solutions can be addressed from this process, and 2 or 4 solution can be applied to robot as they haven't outranged the limit angle of joint. How to choose the final solution is up to the control algorithm.

Table 2. Basic Description to Algorithm of standard IK solution of KUKA R2700.

| | |
|---|---|
| Step 1 | Given the target pose of robot |
| Step 2 | Compute the wrist point position from the given target pose. |
| Step 3 | Solve the first 3 joint angles from wrist point position, 4 solutions can be addressed. |
| Step 4 | Solve the last 3 joint angles by computing the hand attitude from the attitude of first three joint and target pose respectively, each solution of the first part can address 2 solutions. |
| Step 5 | Compose the results of STEP3 and the corresponding results STEP4 to get the final solution. |

## 3. Practical IK problem solution using Full Calibrated Parameters (PIKFCP)

### *3.1. Introduction of PIKFCP*

Nowadays, as the projects need higher absolute positioning precision, and the fact that the manufacturers of robot haven't take measures the assure the accuracy, it is the project engineers' responsibility to reduce the absolute positioning error. Calibration of the DH (or MDH) parameter using some high accuracy instrument and computation algorithms is a good option. We can get a new table of DH parameters which most ones are a little different from the factory designed parameters, see table 3.

Table 3. An Example of Calibrated MDH Parameters of our Robot KUKA R2700.

| | $\alpha$(deg) | $a$(mm) | $\theta$(deg) | $d$(mm) |
|---|---|---|---|---|
| Joint 1 | 0.000 | 0.000 | 0.000 | 675.000 |
| Joint 2 | -89.990 | 351.316 | 0.028 | 0.000 |
| Joint 3 | 0.020 | 1150.626 | -89.984 | -0.190 |
| Joint 4 | -90.003 | -40.627 | -0.037 | 1198.725 |
| Joint 5 | 89.981 | 0.250 | 0.020 | -0.016 |
| Joint 6 | -90.007 | -0.053 | 180.000 | 215.000 |

Another problem occurred after the calibration as this causes the impossibility to solve the IK problem using the method we mentioned before. As the essential condition the closed-form IK solution is the most parameters of DH model is 0 or PI/2, which can simplify the function of the forward kinematics. Even the parameters variate a little, the new parameters cause the function becomes too complicated to solve. Furthermore, using the previous mentioned standard IK solution directly may cause huge error. Under these calibrated parameters, a solution is needed to find a high accuracy solution using limited computational resource to avoid increasing the burden of the industrial projects. Even some research had applied the optimization based or neural network based inverse kinematic solution to solve the problem [9-10], they all cost too much resources - big data or long time.

In this circumstance, a Practical IK problem solution using Full Calibrated Parameters (PIKFCP) had been developed. This method can get high accuracy IK solution using limited computational resource which can satisfied both the precision and the time cost requests of industrial projects.

### *3.2. Basic theory of PIKFCP*

Even the function of forward kinematic is a nonlinear function, it can be linearized while the value of variables varies in a small range. We assume the wrong solution using factory parameters is in the neighborhood of real solution and use the backward compensation method to get the solution. The backward compensation method is getting an iterated solution by adding a compensation vector to the target position. Each iteration has narrowed the distance of iteration solution and real solution.

It is mainly illustrated as two parts of the process of PIKFCP as it processes the compensation individually in the two parts of the standard IK solution. The algorithm takes the standard IK solution as the raw result and applies the compensation to get the high accuracy solution.

Each single solution of the raw result is used to find the wrist points respectively, while the wrist point won't be the same point any more. The wrist point of each single solution is computed from the target pose with the process inversed from the forward kinematics of DH model. Specifically applying the inversed transform of 6th joint, 5th joint and 4th joint to the target pose, the corresponding wrist pose is ready.

### *3.3. Iteration of the first part of robot*

In the solution of first part of robot before the wrist point, after we get the first three values of the raw result, a new shadow wrist point (SWP) can be computed using forward kinematic formula. As we expected, the SWP is different from the targeted wrist position (TWP), then a temporary compensation vector (TCV) can be constructed from these two points. Another new compensated wrist point (CWP) is available by adding the TCV to the target wrist point. After solving the first part using CWP, one round iteration has finished. The result of this round is closer to the real result than the raw result, namely, the norm of TCV in the next round will be much smaller.

The high accuracy solution of first part of robot can be addressed after enough rounds of iteration.

### *3.4. Iteration of last part of robot*

Different from the position solution, accurate attitude solution cannot be compensated by vector. It is because the vector represents the attitude is not a Cartesian coordinate vector, that means there is no meaningful addition operation in attitude vector. Fortunately, we can use isometry matrix or rotation matrix to represent the attitude and the multiplication of inverse of these kinds of matrix and another attitude matrix can equally act as the subtraction.

As to compensate the error of solution caused by the difference of standard DH parameters and calibrated DH parameters, our method computes a temporary compensation matrix (TCM) to compute the compensated attitude matrix (CAM) which is used to compute the accurate attitude solution.

As we discussed in the process of the solution of wrist point, the result will inch towards the real solution after each round of iteration.

### *3.5. Algorithm of Practical IK solution using Full Calibrated Parameters (PIKFCP*

The whole algorithm of PIKFCP can be addressed with the combine of 3.3 and 3.4 and 2.2, and is showed in table 4.

Table 4. Algorithm of PIKFCP.

| | |
|---|---|
| Step 1 | Apply the standard IK solution. |
| Step 2 | Pick up the reasonable solution by the limits of joint angles (optional) and save into TIK(or save the result of step1 into TIK directly). |
| Step 3 | For i = 1 to n (n is the needed iteration round): |
| Step 4 |   For each single solution (tik) in TIK: |
| Step 5 |     Compute the wrist point position under tik. |
| Step 6 |     Compute the compensated first 3 angles. |
| Step 7 |     Compute the compensated last 3 angles and pick the one near tik. |
| Step 8 |     Compose the results of step6 and step7 to get the solution of this round. |
| Step 9 |   End For |
| Step 10 |   Save the iterated result into TIK. |
| Step 11 | End For |
| Step 12 | TIK is the final solution. |

## 4. Conclusion

We picked 36 poses in robot workspace. Two ways of choice had been taken, one is that the points can cover the workspace, namely some are near the outer edge of workspace, some are near the inner edge, others are in the middle where is most likely the working area of this robot – in fact, they truly are part of a simulated working path of previous mentioned robot KUKA R2700 in a project. We can find out that all the positioning error of PIKFCP are less than 0.01mm in figure 2 and that means the IK solution barely influence the final error of project uses the robot.

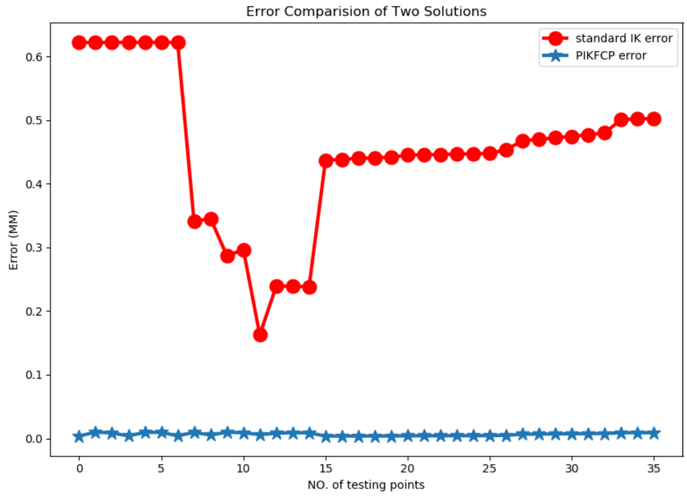


Figure 2. Error Comparison of Two Different Solutions.

We can see from the data and curve that using our PIKFCP, the errors are all below 0.01MM which can satisfy industrial projects after two rounds of iteration. If necessary, we believe the accuracy can be improved easily, which will cost more iteration and more time.

## References

[1] Albert Nubiola and Ilian A Bonev J 2013 Absolute calibrationof an ABB IRB 1600 robot using a laser tracker Robotics and Computer-Integrated Manufacturing 29: 236-245

[2] A Nubiola and I A Bonev J 2014 Absolute robot calibration with a singletelescoping ballbar. Precision Engineering 38(3):472-480

[3] Joubair A, Zhao L F and Bigras P et al. J 2015 Absolute accuracy analysis and improvement of a hybrid 6-DOF medical robot. Industrial Robot: An International Journal 42(1):44-53

[4] Bai Y, Cong M and Yang X, et al. J 2015 Kinematic parameter identification for 6R serial robots based on a 6-parameter model. Robot 37(4):486-492

[5] He R, Zhao Y and Yang S, et al. J 2010 Kinematic-parameter identification for serial-robot calibration based on POE formula. IEEE Transactions on Robotics 26: 411–423

[6] Bruno Siciliano and Lorenzo Sciavicco, et al. M 2009 Robotics Modelling, Planning and Control Londn:Springer-Verlag London Limited.95-95, 90-99

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

[8] J. Sun, G. Cao, W. Li, Y. Liang and S. Huang, "Analytical inverse kinematic solution using the D-H method for a 6-DOF robot," *2017 14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, Jeju, 2017, pp. 714-716.

[9] Köker, Raşit. A genetic algorithm approach to a neural-network-based inverse kinematics solution of robotic manipulators based on error minimization[M]. Elsevier Science Inc. 2013.

[10] Duka, Adrian-Vasile. Neural Network based Inverse Kinematics Solution for Trajectory Tracking of a Robotic Arm[J]. Procedia Technology, 2014, 12:20-27.