A Practical Method to Improve Absolute Positioning Accuracy of Industrial Robot

Guosong Shi¹³, Shuying Zhao¹, Baolong Hu²

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. Industrial robots had been wildly used in modern factory. While the high repeatability of industrial robots can address most of need of process, their low absolute positioning ability cannot reach the demands of some high precision tasks. An example is if the action of the robot relies on the real-time sampled data rather than the simulated data. The low absolute positioning ability of the robot will cause working performance far different away from expectation. For improving the absolute positioning accuracy as long as compile with the programs of automation integration projects, a practical general compensation method has been advanced. Our method also can be a compensation frame which can absorb new better model in the future. One application of this method in a project verifies the achievement of this method.

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 highly flexible and variable projects that teach function cannot handle. For example, when the robot should act based on the real time data rather than data simulated in the CAM (Computer Aided Manufacturing) software. Unfortunately, industrial robots have high repeatability, which can only satisfy the teach function, and low absolute positioning accuracy, which is essential in real-time-data based projects.

There are several ways to enhance the absolute accuracy of a robot and the commonly ways are aiming at the geometry factors[1-5]. They believed that the positioning error is caused by the inaccurate structure parameters, and the high accuracy is coming by accurate parameters. These ways are geometry method. The effective way is to calibrate the parameters and let the robot act of calibrated parameters to perform highly accurate absolute positioning accuracy. There are also some methods aiming at non-geometry factors[6,7]. They measured the error and established the model of error and find a way to change the geometry parameter of the robot to change its positioning accuracy. There are still some model free methods which don’t use the same modelling as the robot does itself to find the relationship of the positioning error and the workspace[8]. These methods are also facing the problem that they need the robot to compensate the error itself. They change the parameters of the robot or communicate the compensation variable to the robot. As geometry methods, non-geometry
methods and model free methods are all rely on the parameters of robot. It will be convenient if capable to update the parameters in the robot. While most commercial industrial robots don’t authorize to do that, we need a way to do the offline compensation. This article addressed a compensation method which is easily be applied in both newly built projects and reformed projects without changing any inner configuration parameters of robots.

2. Kinematics of robots and IPBC & PPBC

2.1. Kinematic functions

The kinematics of robots are used to control the action of robots. Namely, kinematics functions are used to describe the relationship of control variable (commonly considered as Joint Space) and terminal pose (position and attitude, mostly represented in Cartesian Space), see figure xxx. There are two kinds of function – forward kinematic function (fk) and inversed kinematic function (ik). Forward kinematic function is the function of terminal pose of the control variable. Meanwhile the inversed kinematic function is the function in the inversed direction, see figure 1. As long as the model of robot is established, these two control functions are determined as certain functions. While the robot is a 6-DOF (degree of freedom) industrial robot – the control variables are rotation angles of the 6 joints, the kinematic functions are represented as equation (1) and (2).

$$\theta_1, \theta_2, ..., \theta_6 = ik(Terminal\_Pose)$$  \hspace{1cm} (1)

$$Terminal\_Pose = fk(\theta_1, \theta_2, ..., \theta_6)$$  \hspace{1cm} (2)

2.2. Usage of Kinematics Functions to Control a Robot

While an industrial robot had been modelled in DH/MDH model, it is possible to control the action of the robot. We advanced two popular control commands - P2P and LINE - from KUKA program to illustrate the use of kinematics. The P2P command is to let the robot move the terminal from one pose (position and attitude) to another pose without composing the robot poses in the path. The LINE command is to let the robot move the terminal from one pose to another along with the line constructed by these two points. In conclusion, the robot controller uses the fk, ik solution and the composition of them to control the movement of the robot. Figure 2 shows one working path of our robot. It is consisted of LINE command points, and the small sticks at the ends of the curve are the result of P2P commands points.

2.3. Inner parameters based computation - IPBC

When we face a certain robot, the structure is already determined and the parameters are also already stored in the inner controller of the robot. The robot controller will do the computation to control the body of the robot by these pre-stored parameters. We call this process inner-parameters based computation (IPBC). If the robot has its own real parameters which are different from the pre-stored parameters – which definitely will happen, the real pose of the terminal will be different from the controller calculated pose and the robot will perform absolute positioning error.
2.4. Precise parameters based computation - PPBC
As to control the robot more accurately, two kinematic functions can perform more accurate actions are needed. The robot will move to the precise poses produced by these new functions, and we call these two Precise Parameters Based Computation (PPBC). All the computation results of PPBC can let the robot perform the targeted terminal pose with low or 0 error. The calibration method we mentioned before is PPBC, and any kinematic functions are PPBCs only if they can perform accurate results.

3. A practical compensation method

3.1. Basic information flow of robot action
Some text. Normally, the information transferred between the control program and robot. It is easy to understand that the controller in the robot receives the command or information containing the target pose(s) and let its body act to the target pose(s) by doing IPBC in its controller. As the inner parameters doesn’t match with its solid body, the robot action contains instable error.

To lower the error and the condition that we cannot change the inner parameters of robot, we need to add a compensation step to make the robot action more accurate. The information flow with compensation step is shown in figure 3. The goal of this compensation is to let the robot do its own IPBC and act with pretty much low error. For the IPBC and PPBC are both to calculate the kinematics of the same robot we are using, it is easy to come to a conclusion that the connection of IPBC and PPBC is the control variables of the robot – which are rotation angles of 6 joints of industrial robot. Whatever the model we choose or what function we are using, they all describe the only innocent part of robot – its solid body, and that means the rotate angles are naturally the bridge of all kinematic functions and the bridge of inner controller of robot and the extern compensation methods.

![Figure 3. The information flow with compensation step](image)

3.2. Actual to inner pose compensation - Inducing Pose
The compensation process is the process to find a way to let the robot act at the right rotate angles as well as to find a proper terminal pose to let the controller to calculate out the right rotation angles from the given pose. As the fake pose doesn’t exist and is to induce the robot to the right angles, we call the fake pose inducing pose, and a path composed with a serious of fake poses are called inducing path. The inducing pose are calculated from the target rotation angles and the pre-stored parameters, so the robot controller can precisely calculate out the target rotation angles based its own IPBC. Target rotation angles are calculated out from the target terminal pose and PPBC. The information flow of inducing pose is showed in figure 4.
The first task is to find the IPBC and PPBC of the robot we used – KUKA R2700. The robot uses the MDH model to establish the kinematic functions, and after a calibration of the robot, the IPBC and PPBC are confirmed. We don’t discuss the calibration process here. The robot absolute positioning ability influence the error of the whole project in two places – sampling process and robot acts the working path. Firstly, as we need to transform inaccurate sampling robot pose(s) to accurate one, the Pose Correction step is needed. Secondly, for we need to let the robot act the target pose we need, the Inducing Pose step should be applied here. The figure 5 shows the final process of our project with two compensation steps which are coloured (or shadowed).

3.3. Inner to actual pose compensation - Pose Correction

On the other hand, if the robot carries a sensor like a laser sensor to sample data, as the robot pose data coupling with the sampling data which is need to transform the data from sensor coordinate to the robot coordinate is communicated from the robot controller, which definitely needs to be compensated. The goal is to compute the accurate robot poses of sampling moments and transform the corresponding sampling data to the robot coordinate by these poses. And the information flow is reverse from the above process.

4. Application

In our project, we use robot KUKA R2700 to carry a frame on which is installed one laser camera and the drive together. The process is to use the laser camera to scan the stand-by object, then compute the polishing path from the sampling data and finally change to drive the finish the polishing process, see figure 6(a-c). The sampling and polishing process will be all applied by the robot. While the parts are mostly unique, we cannot use the teach function which is wildly used in robot processing projects. This is the condition we mentioned before. To accomplish the function and accuracy of this polishing project, we use the two kinds of compensation based on IPBC and PPBC to low the error of the process.
The first task is to find the IPBC and PPBC of the robot we used – KUKA R2700. The robot uses the MDH model to establish the kinematic functions, and after a calibration of the robot, the IPBC and PPBC are confirmed. We don’t discuss the calibration process here. The robot absolute positioning ability influence the error of the whole project in two places – sampling process and robot acts the working path. The first one, as we need to transform inaccurate sampling robot pose to accurate one, the Pose Correction step is needed. The second one, for we need to let the robot act the target pose we need, the Inducing Pose step should be applied here. The figure 7 shows the final process of our project with two compensation steps which are colored (or shadowed).

The error was greater than 3mm before. The tool even cannot touch the working parts in the last part of the path. Data of 3 typical working paths show that the working error of the whole system had been reduced to ±0.5mm after the application of two compensation steps. See figure 8.

5. Conclusion
We discussed the concept of IPBC and PPBC and the way of coupling these two to enhance the absolute positioning accuracy of industrial robot as well as two kinds of compensation process which covers two ways of using robot pose data. Even this article picks the robot using in our project which modeled by MDH model, in fact, any other modeling method can use our concept and compensation only if the model can perform forward and inverse kinematic functions correctly. At last we show the application in our project and the final error of one of our working paths. The successful application and the data show that our concept and compensation method can conveniently be applied into the automation program of common project without disturbing the robot controller and the robot can
perform good absolute positioning ability. The level of accuracy is determined by the accuracy of PPBC and our method could use new high accuracy PPBC to improve the robot absolute positioning accuracy.

References
[1] Albert Nubiola, Ilian A Bonev. (2013) Absolute calibration of an ABB IRB 1600 robot using a laser tracker. J. Robotics and Computer-Integrated Manufacturing, 29: 236-245.
[2] Nubiola A, Bonev I A. (2014) Absolute robot calibration with a single telescoping ballbar. J. Precision Engineering, 38(3):472-480.
[3] Nguyen HN, Zhou J, Kang HJ, et al. (2015) A calibration method for enhancing robot accuracy through integration of an extended Kalman filter algorithm and an artificial neural network. J. Neurocomputing, 151: 996–1005.
[4] Bai Y, Cong M, Yang X, et al. (2015) Kinematic parameter identification for 6R serial robots based on a 6-parameter model. J. Robot; 37: 486–492.
[5] He R, Zhao Y, Yang S, et al. (2010) Kinematic-parameter identification for serial-robot calibration based on POE formula. J. IEEE Transactions on Robotics, 26: 411–423.
[6] Zhong X L, Lewis J M, N.-Nagy F L. (1996) Autonomous robot calibration using a trigger probe. J. Robotics and Autonomous Systems, 18(4):395-410.
[7] Judd RP and Knasinski AB. (1990) A technique to calibrate industrial robots with experimental verification. J. IEEE Transactions on Robotics and Automation, 6(1): 20–30.
[8] Zeng Y, Tian W and Liao W. (2016) Positional error similarity analysis for error compensation of industrial robots. J. Robotics and Computer-Integrated Manufacturing, 42: 113–120.