Mechatronics and Automation Technology
J. Xu (Ed.)

doi:10.3233/ATDE221198

Compensation Methods for Industrial Robotics Under Varying Payloads with Deep Reinforcement Learning

Wenlei XIAO[a,b,1], Zihui SUN[a] and Qi QI[a]
[a] *School of Mechanical Engineering & Automation, Beihang University, Beijing, China*
[b] *MIIT Key Laboratory of Aeronautics Intelligent Manufacturing, Beihang University, Beijing, China*

Abstract. Due to the weak rigidity of an industrial robot, its end effector usually has poor absolute positioning accuracy, especially under varying payloads. Such situation is common in scenarios of handling, machining and tool changing. Conventional off-line calibration or compensation methods can only eliminate systematic errors, while such methods are invalid to the dynamic errors brought by varying payloads. This paper proposes a deep reinforcement learning(DRL) approach to solve the problem of dynamic errors, in consideration of external payloads changed manually. An online full closed loop system is established to verify the proposed method, which consists of a KUKA robot KR6, a Leica laser tracker, and a BECKHOFF PLC controller. The robot and the laser tracker work as the slavers of the master PLC controller, in between the communication is accomplished using EtherCAT. Logically, the robot is controlled by mxAutomation and the laser tracker is connected to an embedded EtherCAT slave card. Experiments on the robot demonstrate the effectiveness of the proposed DRL methods. The changed payloads range from 1.177Kg to 4.179 Kg, while the position accuracy of the robot can be maintained no more than 0.4mm by the DRL algorithm.

Keywords. Industrial robotics, varying payload, deep reinforcement learning, error compensation

1. Introduction

In recent years, compared with traditional application scenarios, robotics need to undertake more complex tasks, such as precision assembly and high-precision machining. But existing off-line calibration and compensation methods are not able to adapt to changing factors. Robotics can not achieve high absolute positioning accuracy.

In high-end application scenarios, it is hard to determine the movement of robotics by artificial teaching method. On the one hand, robotics need to reach thousands of point in a wide range of space, on the other hand these scenarios involve complicated curves and surfaces, so CAD software combined with robot off-line programming has become the mainstream method. Danevit and Hartenberg[1] proposed the best-known DH parameter method, which used differential kinematics to establish the identification Jacobian matrix and mapped the end error to various geometric parameters[2]. Due to

[1] Corresponding Author, Wenlei XIAO, School of Mechanical Engineering and Automation, Beihang University, 37 Xueyuan Road, Haidian District, Beijing, China; E-mail: xiaowenlei@buaa. edu. cn.

the weak rigidity of motor gears and reducer at robot joints, the position accuracy decreases obviously under the heavy payloads or high speed motion, so the modeling of robot stiffness is also one of the focuses of research. At present, mainstream methods mainly include finite element method, virtual joint method and so on. The finite element method, not only can be utilized to analyze the robot's stiffness error can also be used to analyze the temperature of the thermal expansion of error[3]. The principle of the virtual joint method[4] is to simplify the reducer gear and other parts at the joints into springs. Salisbury[5] first used this method to establish a stiffness model of the robot joint. Wang Yi[6] established a flexible model of the robot, decomposed the flexibility error into two cases of dead weight and the external load and compensated them respectively.

The modeling of complex non-geometric errors is not only complex but also has limited practical effects. In the actual work scene, loads and other factors will often change, however these off-line methods are not in a position to adapt to the changing factors. Nevertheless, online compensation can respond timely to complex time-varying factors. The main idea of robot online compensation is to integrate external measurement equipment into the robot control process and act as a sensor. In [7], c-track, a high-precision binocular camera, was used to measure the position and pose data of multiple targets in space in real time, and obtained the real-time spatial relative error of FANUC robot. However the use of camera compensation needs to paste a large number of feature points, also it is easy to be affected by lighting. So there are a large number of studies using laser tracker to compensate the errors, such as Shi[8], Qu[9] et al. has realized to the accurate compensation with laser tracker and the position of the open KUKA robot control interface RSI. Meanwhile, compensation has a large lag because of the delay of the control system. In order to achieve better real-time compensation, the robot and sensor are connected into the real-time control system.

After entering the Internet era, the application of artificial intelligence technology in the industrial field is gradually put on the agenda[10-13]. The core of reinforcement learning is to make a large amount of data getting in the process of interaction with the environment as a feedback, to guide the agent to make decisions. As outlined in [13] and [14], these characteristics are ideal for the development of intelligent manufacturing systems. Mahmood et al.[10] used UR5 to imitate the environment of OpenAI Gym and compared the effects of four algorithms TRPO, PPO, soft-Q and DDPG in the path planning task of manipulator. They found that the fewer the nodes of the manipulator, the better the control effect. In the industrial problems solved by the RL method, from [11], DDPG algorithm is combined with the trajectory planner using force feedback to solve the force control problem in the precision nail hole task. In [12], DRL methods are used to improve trajectory smoothing in CNC applications.

In this paper, an online full closed loop system is built for online compensation research, and an error compensation learning method based on DRL algorithm is proposed. The main contributions of paper are as following:

- A complete set of automatic calibration equipment and an online full closed loop compensation system are established. By comprehensively considering various technical routes, a BECKHOFF PLC controller and its supporting various special hardware modules are determined to utilize.
- A robot error compensation method based on DRL is developed, and that is evaluated experimentally on a 6-DOF industrial robot KUKA-KR6. The control goal is to follow different types of reference paths accurately, such as square or circular paths.

This paper is structured as following. Section 2 presents the proposed DRL methods. Section 3 presents the online full loop compensation system. Following that, in Section 4, the implementation of methods to control a 6-DOF robot is recommended. Finally, the paper ends in Section 5 with conclusions.

2. Reinforcement Learning Based Compensation Methods for Robotics

Industrial robotics' own dynamic calculation is complicated, meanwhile the influence of external payloads is superimposed, so the error model is inevitably large and complex. For this problem, let industrial robotics rely on large-scale data for self-learning to build a error compensation model[15-17]. Currently, the mainstream deep reinforcement learning network can be roughly divided into DQN network family based on Q-learning and Policy Gradient based on strategy gradient[18]. In order to avoid the loss of robots due to massive sampling in the stage of reinforcement learning exploration, DDPG algorithm was selected due to its high sample efficiency but relatively difficult parameter tuning.

2.1. DDPG Algorithm

DDPG is a deep deterministic strategy Gradient algorithm, which is proposed to solve the continuous action control problem. DDPG algorithm is essentially a reinforcement learning algorithm of AC framework[19]. It can predict the deterministic strategy and maximize the total reward by single-step updating policy.

The sample efficiency of policy-based methods is low because they only use the latest samples collected from the policy. In the DQN network, an important idea of experience replay is put forward. DDPG algorithm uses this idea to solve the problems of correlation and low sample efficiency by constructing an experience pool. Training labels are constructed through system information, and the commonly used format of construction data labels is (S, A, R, S'), where S is the current state of the system, A is the action finally selected by the network, R is the reward value given by the interactive environment, and S' is the system state after action A is performed. Some items are randomly extracted from the experience pool to calculate the loss function and update the network parameters in each episode.

DDPG algorithm adopts actor-critic network framework, which using two networks with different functions to realize interactive learning with the current environment. The working principle is as following:

Critic neural network is used to approximate the optimal action value function , denoted as, where ω is the parameter of the neural network. The ultimate goal of deep reinforcement learning is to maximize the cumulative reward, and the action value function Q is the conditional expectation of U_t, where γ is the discount factor.

$$U_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 r_{t+3} + \cdots \quad (1)$$

$$Q(s_t, a_t) = E[U_t | S_t = s_t, A_t = a_t] \quad (2)$$

$$Q^*(s_t, a_t) = \max_{\pi} Q(s_t, a_t) \quad (3)$$

In addition, actor neural network is used to approximate the policy function, denoted as $\mu(s|\theta)$, where θ is the parameter of network. It's input is the current state value s_t. The network outputs a deterministic action a_t to obtain the maximum Q value.

$$a_t = \mu(s_t|\theta) \quad (4)$$

The critic network is updated by the gradient descent method. As mentioned above, target network is used to ensure the convergence of parameters, and it's corresponding target network is $Q^{'}(s, a|\omega')$. The calculation formula is as follows:

$$\text{target}_t = R_{t+1} + \gamma Q^{'}(S_{t+1}, \mu(s|\mu) \mid \omega') \tag{5}$$

$$\text{Loss} = \frac{1}{N}\sum_{t=1}^{N}\left(\text{target}_t - Q(S_t, a_t|\omega)\right)^2 \tag{6}$$

The actor network updates θ through the strategy gradient algorithm to maximize the objective function J(θ). Based on the deterministic strategy gradient theorem:

$$\nabla_\theta J\left(\mu(\theta)\right) \approx \frac{1}{N}\sum_i \nabla_a Q(s, a|\theta)\,|_{s=s_i,\ a=\mu(s_i)} \nabla_\theta \mu(s|\mu)|_S \tag{7}$$

Actor neural network generates strategy according to the current environment state s_t and outputs specific action a_t to interact with the environment. Critic neural network is used to evaluate the strategic action a_t, and determine whether the situation is good or bad at this time. It is measured by a value, and the value r is returned to actor neural network for learning. Then the neural network carries out parameter optimization, so that the cost function converges to the global optimal.

2.2. DRL Compensation Method

In this section, we present a framework that uses the DPGG algorithm to compensate errors of the robot end effector under varying payload. We designed the robot to complete the target task in one round, that is, to accurately complete a whole track under the condition of varying payload. A round has n steps in total, and the number of steps is the number of the robot motion instructions. The deep reinforcement learning network takes the position and load of the robot end effector as state S, consisting of the three-dimensional coordinate position (X,Y,Z) and load η. However, since the motion deviation of the robot is generally very small, if the position of the robot motion instruction corresponding to the theoretical position is taken as the output layer, the output of the network will be extremely similar to the input, resulting in learning difficulty and failure to obtain the correct learning result.

Therefore, in order to make the input and output of the network as far away as possible, we design the robot position compensation $E(\Delta X, \Delta Y, \Delta Z)$ as the action A, then the robot motion instructions are calculated by the initial instructions and action A, and the reward R of the reinforcement learning network is calculated by the theoretical position P_L and actual position P_S of the robot, using the calculation method of the negative Mahalanobis distance.

$$D_M(P_S, P_L) = \sqrt{(P_S - P_L)^T \sum^{-1} (P_S - P_L)} \tag{8}$$

$$R = \sigma \times D_M(P_S, P_L) \tag{9}$$

Where $\sum$ is Covariance matrix of P_L and P_S, $\sigma < 0$.

3. An Online Full Closed Loop Compensation System

This section describes the establishment of the system in detail. In order to achieve real-time compensation, we designed and established an online full closed loop compensation system[20-21], which connected the robot and the laser tracker to the real-time control system to obtain the position of robot in real time, as shown in figure 1.

The system is mainly composed of three parts: the laser tracker, the robot and the BECKHOFF PLC controller. The used laser tracker is a Leica AT901-B laser tracker with TCP/IP communication port. That has high closure and low flexibility and cannot be linked with other software. In order to realize the cooperative control and systematic communication between robot and laser tracker, we designed an online measurement module of laser tracker based on STM32 chip. The used robot is KUKA-KR6 robot, and its repeated positioning accuracy is 0.05mm, absolute positioning accuracy is 2mm. The robot work as the slavers of the master PLC controller through mxAutomation (a KUKA function block), so as to control the robot movement in real time. The master PLC controller as an external controller, connects the robot system and the laser tracker system, which process all kinds of data.

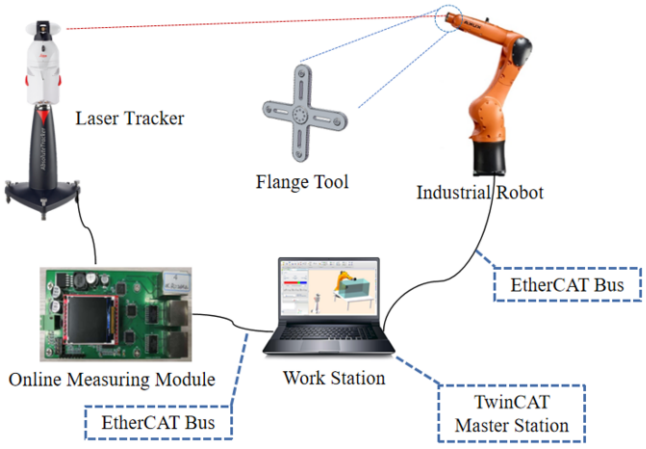

Figure 1. Schematic diagram of the online whole-close-loop compensation system.

3.1. Online Measuring Module of the Laser Tracker

In most cases, the laser tracker is connected with SA software, and data analysis is carried out in this software, but in order to improve its flexibility, it needs to be developed by secondary development API EMSCoN. We designed an embedded module based on STM32 chip which forms EtherCAT measurement slave together with module EL6021. The module is a slave terminal module from BECKHOFF, used for converting serial port into EtherCAT protocol.

Serial communication may cause some delay and instability in the system communication. So module integration processing is carried out, by using ET1100 chip to skip conversion of the external module. TCP/IP protocol data parsing and conversion of EtherCAT protocol on the embedded module are achieved, which reduces the impact

of serial communication on the module. In order to enhance the working frequency performance of the module, STM32F4 chip with a higher dominant frequency is used. That also strengthens the electrical isolation characteristics of the module, effectively isolated the electromagnetic interference that may be generated in the laser tracker controller and the robot controller, then the module works more stably.

3.2. External Control System of Robot

In the above work, the secondary development module of the laser tracker has been integrated into the PLC controller to realize online control and measurement of spatial points. In order to form an full closed loop system, the robot also needs to work as the slavers of the master PLC controller. MxAutomation is an interface for KUKA robotics to achieve external PLC control which supports external PLC programming to control robot movement in real time. The terminal module EL6695-1001 is specially developed by BECKHOFF for communication with the KUKA robot, which can establish secure communication between the external controller (TwinCAT in this paper) and KUKA KRC4 controller. The EtherCAT bridge can also be used to exchange insecure I/O data between the external controller and the KUKA-KRC4 controller if both controllers are configured as primary stations in their respective bus lines. To do that, the EtherCAT bridge must be configured as two slave stations. The EtherCAT bridge forwards the received data from one circuit to another, which allows for the exchange of large amounts of data at the bus clock rate.

The system workflow is as follows: First, the initial robot motion commands are generated by the laser tracker and off-line planning method. Then, the initial motion command is sent from PLC to the robot controller, and the laser tracker is ordered to collect position information. When the robot moves, the joint angle and confirmation signals are sent back to the PLC via the robot controller and EtherCAT, while the measured position data is sent to the PLC via the real-time measurement module, the laser tracker controller and EtherCAT in turn. In this way, the measurement data and corresponding control data can be obtained in real time, so that the error of the robot end effector can be obtained in real time, and the online error compensation can be carried out.

4. Case Studies

In order to verify the advantages of the DRL compensation method proposed in Section 2, an error compensation experimental system for industrial robot was set up, then the algorithm and system are experimentally verified and analyzed under two environments: constant payloads and varying payloads. The experimental scene is shown in figure 2, I is the KUKA-KR6 robot, and II is the robot end flange tool which is composed of three parts: target ball socket tool, adapting piece of loads and different loads. Its main function is making the target ball of laser tracker and different load installed conveniently at the end flange.

Due to the laser tracker can not measure pose of robot, so target ball socket tool is designed for that. Multi-point measurement method is used to get the pose of the measurement point. First, the target ball is placed at two points of A and B respectively, and the vector represents the X direction. Then the target ball is placed at two points of

C and D respectively, representing the Y direction. Z direction can be obtained from the cross-product of two vectors, and the origin is the position of the target ball.

One end of the adapting piece of loads is fixed on the robot flange, and the other end can be installed with different loads. Loads can be fixed and changed quickly through screws, so as to change the payloads of the robot end effector.

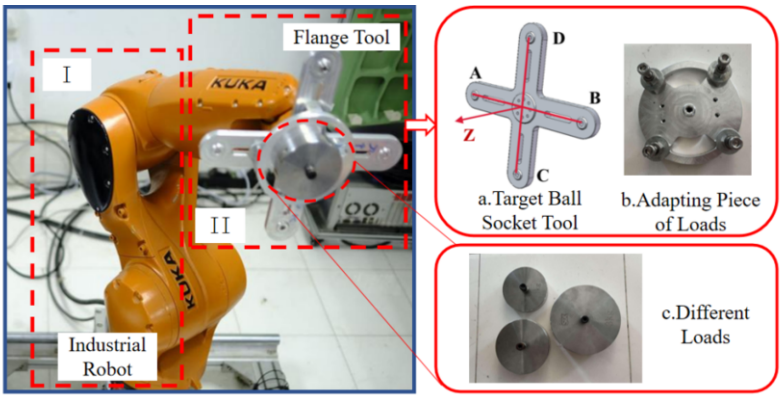

Figure 2. KUKA-KR6 robot with flange tool.

4.1. Error Compensation Analysis Under Constant Payloads

Error compensation experiments under constant payloads are carried out in the above system scenario. Robot tracks a circular trajectory in the robot workspace and the goal is to minimize the position error. First, the ideal trajectory and initial motion instructions need to be generated in Cartesian space. Circumscribed polygon is made outside the ideal circle, and the distance between the vertex of the polygon and the ideal circle is 0.1mm. The vertexes of the polygon are the initial instruction positions.

The first task is to compensate errors under constant payloads. The errors and rewards changing in the training process of the deep reinforcement learning model were recorded, as shown in figure 3. It can be observed that DDPG algorithm has achieved good results in the performance of position accuracy. And the more episodes, the greater the total reward, and the smaller the error convergence, gradually approaching 0. After 40 rounds of training, the error of the robot end effector can converge from 3mm to 0.1mm under constant payloads.

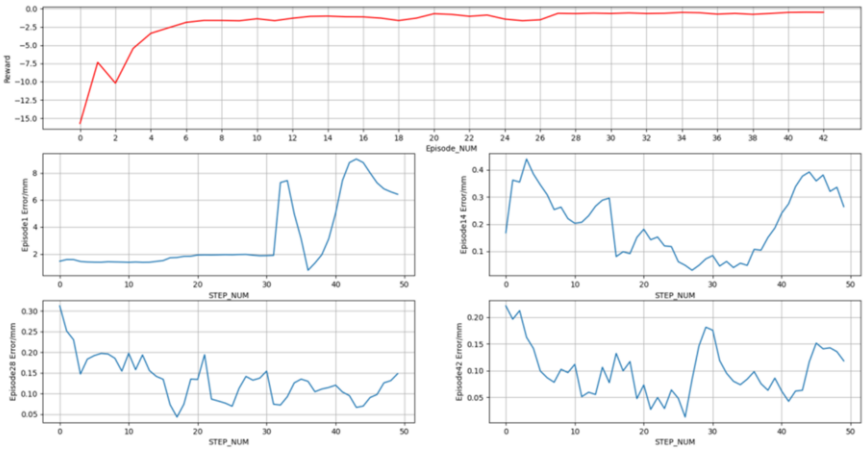

Figure 3. Training process under constant payloads.

4.2. Error Compensation Analysis under Varying Payloads

Due to the weak rigidity of robot, its accuracy will deteriorate under varying payloads, and the learning model under constant payloads cannot compensate the error, so it is necessary to carry out experiments under varying payloads of error compensation analysis of industrial robot. The maximum load-bearing capacity of the robot end flange is 6Kg, and three weights of 1Kg, 2Kg and 3Kg are used in the experiment, making the robot end load-bearing in four states: 1.177kg (no load), 2.197kg (1Kg weight), 3.201Kg (2Kg weight) and 4.197Kg (3Kg weight). First of all, as shown in figure 4, it can be seen that the influence of different payloads on error. The heavier the payloads, the greater the error of the robot. When the payload's difference is the largest, the error of 0.2mm is caused by the payload.

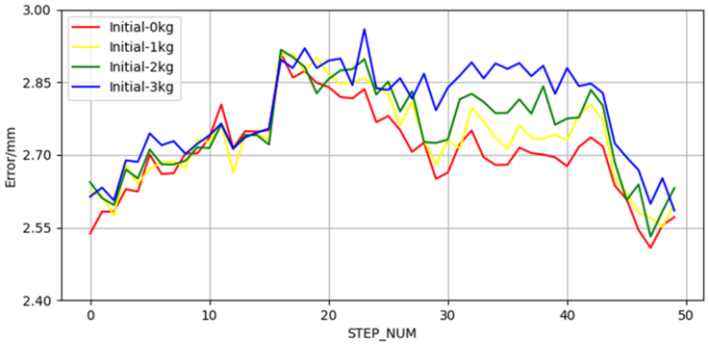

Figure 4. Errors of the robot end effector under different payloads.

DDPG algorithm is used to compensate for errors, and different payloads were changed after 5/10 rounds during network training. The algorithm training process is shown in figure 5. It can be observed that DDPG algorithm also achieves good result in the performance of position accuracy indexes under varying payloads. Moreover, it can be found that the error of the robot increases significantly after changing the payload in

the early stage of network training. At this time, the network has not learned to compensate the error caused by the load. As the number of rounds increases, the total reward of rounds generally shows an upward trend. After 45 rounds, the errors of the robot end effector under different payloads can converge from 3mm to about 0.1mm in each round. It can be verified that the algorithm has basically learned the error of the circular trajectory under different payloads.

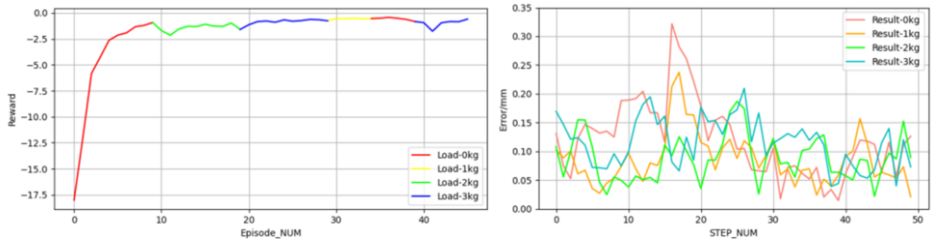

Figure 5. Training process under varying payloads. Left: Different payloads changed after 5/10 episodes. Right: Errors of the robot end effector under different payloads after training .

Next, an experiment of changing payloads within one round is carried out, and the payloads is changed every 8 steps to verify the error compensation under the condition of changing payload. The result of the experiment is shown in figure 6. It can be seen that the error of a single track under varying payloads was reduced to about 0.3mm. Although the effect was not as good as that of constant payloads. And the result of error compensation is best when the load is 1kg, which may be caused by the lack of bad data in network training. In general, this experiment verifies that reinforcement learning algorithm can compensate availably the errors of the robot end effector under varying payloads.

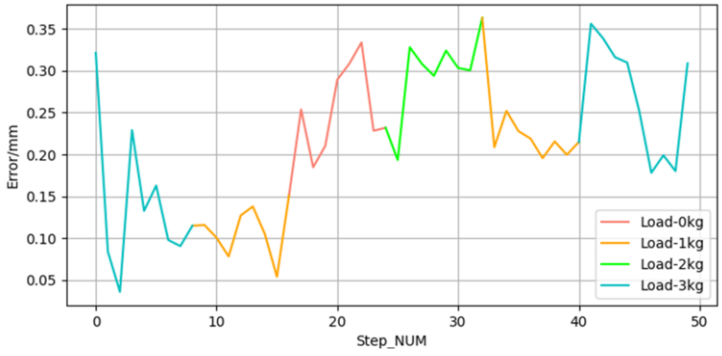

Figure 6. During one episode the errors of the robot end effector under varying payloads.

5. Conclusions

This paper introduces the online full closed loop system and error compensation method based on DRL to improve the position accuracy of the robot under varying payloads. Robot error compensation uses DDPG algorithm to realize the correction of robot motion instructions. Finally, two experiments are carried out to evaluate the developed system

and method. Results of experiments show that the proposed method can significantly reduce the position error from 3mm to 0.4mm. High accuracy can be obtained by using the online compensation strategy, and there is no need to build a complex model and collect massive data, and the adaptability to the environment is also relatively strong.

In the future, the 6-DOF laser tracker will be used to carry out more in-depth research on pose accuracy, and a variety of applications such as milling and progressive forming will be carried out on more types of robots to improve pose accuracy in different scenarios.

References

[1] Denavit J, Hartenberg R S. A Kinematic Notation for Lower-Pair Mechanisms[J]. Journal of Applied Mechanics, 1955, 22.

[2] Harb S M, Burdekin M. A systematic approach to identify the error motion of an N-degree of freedom manipulator[J]. The International Journal of Advanced Manufacturing Technology, 1994, 9(2):126-133.

[3] I. Fernández-Bustos and J. Agirrebeitia and G. Ajuria and C. Angulo. A new finite element to represent prismatic joint constraints in mechanisms[J]. Finite Elements in Analysis and Design, 2006.

[4] De blaise, Hernot X , Maurine P . A systematic analytical method for PKM stiffness matrix calculation[C]// Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on. IEEE, 2006.

[5] J. K. Salisbury, "Active stiffness control of a manipulator in cartesian coordinates," 1980 19th IEEE Conference on Decision and Control including the Symposium on Adaptive Processes, 1980, pp. 95-100, doi: 10.1109/CDC.1980.272026.

[6] Yi Wang. Study on error Model and Calibration technology of Multi-joint Kinematic Mechanism for measurement[D].Tianjin University, 2009.

[7] T. Shu, S. Gharaaty, W. Xie, A. Joubair and I. A. Bonev, "Dynamic Path Tracking of Industrial Robots With High Accuracy Using Photogrammetry Sensor," in IEEE/ASME Transactions on Mechatronics, June 2018, 23(3): 1159-1170 doi: 10.1109/TMECH.2018.2821600.

[8] Shi X, Zhang F, Qu X, et al. An online real-time path compensation system for industrial robots based on laser tracker[J]. International Journal of Advanced Robotic Systems, 2016, 13(5): 1-14.

[9] Weiwei Qu, Huiuyue Dong, Yinlin Ke. Pose accuracy compensation technology of Robot - assisted aircraft mounting in the preparation hole[J]. Chinese Journal of Aeronautics, 2011, 32(10): 1951-1960.

[10] Mahmood A R, Korenkevych D, Vasan G, et al. Benchmarking Reinforcement Learning Algorithms on Real-World Robots[J]. arXiv: Learning,2018: 561-591.

[11] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, “Asynchronous methods for deep reinforcement learning,” in Intern. Conf. on Machine Learning, 2016, pp. 1928–1937.

[12] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, “Trust region policy optimization,” in Intern. Conf. on Machine Learning, 2015, pp. 1889–1897.

[13] X. Yao, J. Zhou, J. Zhang, C.R. Boër, From intelligent manufacturing to smart manufacturing for industry 4.0 driven by next generation artificial intelligence and further on, ES, 5th Int. Conf. on Enterprise Systems (2017) 311–318.

[14] Zhong R.Y. , Xu X., E. Klotz, S.T. Newman, Intelligent manufacturing in the context of industry 4.0: A review, Engineering 3 (5) (2017) 616–630.

[15] Yudha P. Pane, Subramanya P. Nageshrao, Jens Kober, Robert Babuška, Reinforcement learning based compensation methods for robot manipulators, Engineering Applications of Artificial Intelligence, Volume 78, 2019, Pages 236-247, ISSN 0952-1976, https://doi.org/10.1016/j.engappai.2018.11.006.

[16] Y. Ansari, E. Falotico, Y. Mollard, B. Busch, M. Cianchetti and C. Laschi, "A Multiagent Reinforcement Learning approach for inverse kinematics of high dimensional manipulators with precision positioning," 2016 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob), 2016, pp. 457-463, doi: 10.1109/BIOROB.2016.7523669.

[17] Stuckelmaier P, Grotjahn M, Frager C.Iterative improvement of path accuracy of industrial robots using external measurements[C]// 2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM). IEEE, 2017.

[18] Lyu L, Shen Y and Zhang S. "The Advance of Reinforcement Learning and Deep Reinforcement Learning," 2022 IEEE International Conference on Electrical Engineering, Big Data and Algorithms (EEBDA), 2022, pp. 644-648, doi: 10.1109/EEBDA53927.2022.9744760.

[19] Y. P. Pane, S. P. Nageshrao and R. Babuška, "Actor-critic reinforcement learning for tracking control in robotics," 2016 IEEE 55th Conference on Decision and Control (CDC), 2016, pp. 5819-5826, doi: 10.1109/CDC.2016.7799164.

[20] Alici G, Shirinzadeh B. A systematic technique to estimate positioning errors for robot accuracy improvement using laser interferometry based sensing[J]. Mechanism & Machine Theory, 2005, 40(8):879-906.

[21] K. Kamali, A. Joubair, I. A. Bonev and P. Bigras, "Elasto-geometrical calibration of an industrial robot under multidirectional external loads using a laser tracker," 2016 IEEE International Conference on Robotics and Automation (ICRA), 2016, pp. 4320-4327, doi: 10.1109/ICRA.2016.7487630.