Research and Development of High Precision Robot Automatic Docking System Based on mxAutomation

Dechao Zheng¹,², Rui Guo¹,², Kai Chang¹,² and Jian Yin¹,²

¹Intelligent Detection and Equipment Laboratory, Shenyang Institute Of Automation, Chinese Academy Of Sciences, No.114Nanta Street, Shenhe District, Shenyang, Liaoning Province, P.R.China
²Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences, Shenyang 110016, China
zhengdechao@sia.cn

Abstract. In this paper we designs an automatic docking system to simulate the precision docking of the key components in aerospace assembly line. By using the robot to make the actual measurement of the experimental workpiece, we figure out the spatial coordinate system of the workpiece which correspond to the robot world coordinate system. This coordinate system is used to automatically control the robot to calculate the motion trajectory and complete the precision docking work. This system is characterized by the use of Siemens PLC to control the KUKA robot dynamically through the KUKA-TIA-Library interface, which can meet the requirements of flexibility and customization in aerospace industry.

1. Introduction
In aerospace manufacturing, objects such as rockets and aircraft are very large in size and require a lot of assembly space. Usually, the main body is not the whole processing, but needs to be manufactured in sections, and then in the final assembly link to assemble the large parts together [1]. The traditional docking methods mainly relies on tooling, standard templates, samples, gauges and process compensation to ensure the accurate relative position of large size components. This kind of docking method is difficult to locate the benchmark, difficult to adjust by manual, low accuracy, and greatly affected by human factors, resulting in long manufacturing cycle, poor reliability, so the conventional docking method is difficult to meet its requirements. In this paper, a set of high-precision dynamic and automatic robot auto-docking system based on mxAutomation control is designed to improve and optimize the assembly methods.

2. Composition of the control system
The system is mainly consists of robot control system, PLC control system, superior system, and docking experimental parts [2]. The robot control system adopts KR 6 R700 sixx small industrial robot of Kuka company, supporting KR C4 Compact controller, with a load of 6kg and a repetitive positioning accuracy of +/−0.03mm. The PLC control system adopts Siemens s7-1515-2pn PLC, with 500KB code storage and 3MB data storage area, bit instruction execution time up to 30ns. The superior system adopts Advantech IPC610 drawer type industrial controller, which is equipped a 24 "industrial display. The specific configuration is shown in ‘Figure 1’:
The system adopts Siemens Profinet communication bus and use the Siemens PLC as the main station controller. The KRC4 Compact integrate a Profinet slave station module inside and worked as a slave station on the system bus. The programming software adopts the latest version of TIA Portal V15, and KUKA. PLC mxAutomation library is installed. The specific hardware model and software version are shown in the ‘Table 1’:

Table 1. The configuration list of control system.

| Name of the element            | Specific model                        | Remarks          |
|--------------------------------|---------------------------------------|------------------|
| Simatic Control                | S7-1515-2PN                           | Firmware version V2.5 |
| PLC memory card                | 6ES7954-8LE03-0AA0                    | 12M              |
| Kuka robot                     | KR 6 R700 SIXX                        |                  |
| Robot controller               | KRC4 Compact                          | Firmware version V2.5 |
| Industrial computer            | IPC 610                               |                  |
| Configuration software         | TIA Portal                            | Software version V15 |
| Library files                  | KUKA-TIA Library                      | Library version V2.13 |
| Kuka external control interface| Kuka.PLC.mxAutomation                 | Software version V2.13 |
| Kuka slave station module      | Kuka.Profinet Device                  | Software version V3.2 |
| Kuka configuration file        | Kuka GSD file for KRC4                | Software version V2.3.1 |
| Kuka programming software      | Work Virtual                          | Software version V4.0.29 |

3. Configuration and application of Kuka.PLC. mxAutomation

As an optional software package of Kuka robot, Kuka.PLC.mxAutomation [3] is used to convert the programming instructions of PLC into KRL language of KUKA robot and submit it to the robot for execution. Through this software package, the programmer can control the robot to perform absolute and relative positioning movement, definition and switching of tool coordinate system and base coordinate system dynamically and flexibly. The establishment of point positions and coordinate systems of these motions is carried out dynamically, and it is not necessary to teach or define them in advance in the robot. Through this interface, up to 5 robots can be controlled at the same time.

When using the software package, it is necessary to configure the robot controller side and PLC controller side separately. On the robot side, the software package is installed on the robot controller in
the form of KOP installation package through WorkVistual software [4]. After installation, we configured robot controller as Profinet device and selected the corresponding communication I/O Numbers are 2032 (254 bytes), see ‘Figure 2’ for details.

![Figure 2. Configuration of KUKA robot in WorkVistual.](image)

In the configuration software of PLC, through the installation of Kuka-Tia-Library file, and in accordance with the provisions of the use of Library file function programming operation, PLC can achieve the external automatic control function of the robot. In the HMI software simulation interface of PLC, you can start, stop, reset, set the running speed of the robot, manual JOG mode, manual teach the point and save to PLC’s data blocks, also can read real-time robot position information, status information and other important parameters of the robot. The HMI interface look like ‘Figure 3’.

![Figure 3. The HMI software simulation interface.](image)

4. Control of the KUKA robot in TIA Portal

When using Siemens PLC to program control KUKA robot, it must follow a certain process. The flow chart in ‘Figure 4’ briefly illustrates the control logic:
The starting program of the PLC must use KRC_ReadAxisGroup() function to read the information of the robot from the communication interface’s receiver cache to the local data block. Similarly, the ending program must use the KRC_WriteAxisGroup() function to write all control programs to the communication interface’s sender cache. The effective programming interval is between these two functions, and all control and motion instructions need to be written between these two functions.

At the beginning of the robot control program, we use KRC_Initialize() function to initialize mxA interface, then using KRC_ReadMxaStatus() to check the real-time Status of the interface, when the parameters of the Status is 3 or more which indicated interfaces on both sides are showed normal. Otherwise, KRC_ReadKrcError(), KRC_ReadMxa_Error() and KRC_Diag() need to be run to detect robot KRC program error, robot mxA interface data error and robot self-diagnostic error, respectively. According to the feedback of detection results, KRC_Message_Reset() can be run to successfully reset the fault information, and then KRC_AutoExt() function can be run (at this time, the robot needs to switch to the external automatic mode AUT EXT on the robot teacher). When the external automatic mode is started successfully, the robot can be programmed for motion control.

In a single scanning cycle, the PLC writes all the work to the Profinet IE bus and passes it to the Profinet bus module of the robot controller. After receiving the data, the robot controller first run the Kuka.PLC.mxAutomation interpreter internally to interpret the transmitted PLC data into KRL language that can be operated by KUKA robot and pass into the program storage area of the robot. At this point, the robot interpreter starts to reads the commands in the program storage area and controls the operation of the robot. It should be noted that the global motion speed of the robot needs to be limited in the control process, so the KRC_SetOverride() function is used to limit the axial motion and linear motion to an acceptable range.

5. The experimental process of precision docking system
In the construction of this experimental system, the robot is mainly used to docking the two parts with very small clearance automatically and accurately, which is difficult to achieve by ordinary teaching function. The stationary docking piece is a circular hole structure, which simulates the flange hole commonly used in aviation assembly field, and is fixed on the ground by a right-angle steel plate base. The mobile docking piece is a disc structure, simulating the lining ring in the supporting flange hole, which can be directly installed in the center of the robot flange plate and ensure that the connection between the two center points is perpendicular to the disk. The specific size is shown in the ‘Figure 5’.
As you can see the fit clearance between is only 0.5 mm, and it requires the mobile docking piece to be vertically inserted along the center of stationary docking piece to complete the precision assembly. So we set the center of the stationary docking piece as the origin of the robot base coordinate, the horizontal plane where the whole circular hole is located is set as the x-y plane of the coordinates, the line perpendicular to the x-y plane and passing through the origin serves as the z-axis of the coordinate. This coordinate system serves as the workpiece coordinate system (Base coordinate system) for robot docking [5].

In order to obtain the spatial expression of the circular hole of the stationary docking piece, firstly, a known probe tool is installed at the end of the robot flange, and the physical parameters of the tool are set in the tool coordinate system of the robot. Then, the robot is controlled to touch the probe to some points on the edge of the circular hole, and the recorded this points to the robot controller. This measurement process is the result of manual instruction, and some human errors will be introduced. After the subsequent introduction of the measurement system, the errors can be gradually eliminated and automated [6].  

The ‘Figure 6’ above shows three spatial coordinate points p1, p2 and p3 found on the circular hole plane by manual instruction. The coordinate values of these points are all based on the values of the world coordinate system of the robot. The plane where the circular hole is located and the position of the center point o of the circular hole are fitted by these three points, which serves as the origin of the base coordinate system.
With the origin as the center, point \( p_1 \) points to \( p_2 \) as the vector \( \vec{x} \), point \( p_1 \) points to \( p_3 \) as the middle vector \( \vec{m} \). After the vector \( \vec{x} \) cross multiplied by vector \( \vec{m} \), we can get the normal vector \( \vec{z} \) of the plane, then we use vector \( \vec{x} \) cross multiplied by vector \( \vec{z} \) to get the vector \( \vec{y} \). Finally, by standardizing the three axes, the specific operation results can be established as follows:

\[
\vec{x} = (p_2 - p_1) / \text{norm}(p_2 - p_1) = [-0.3886, 0.212, -0.8967] \quad (1)
\]

\[
\vec{m} = (p_3 - p_1) / \text{norm}(p_3 - p_1) = [0.1112, 0.7248, -0.6799] \quad (2)
\]

\[
\vec{z} = (\vec{x} \times \vec{m}) / \text{norm}(\vec{x} \times \vec{m}) = [-0.7289, -0.5245, -0.44] \quad (3)
\]

\[
\vec{y} = (\vec{z} \times \vec{x}) / \text{norm}(\vec{z} \times \vec{x}) = [0.5636, 0.8246, -0.0493] \quad (4)
\]

Rotation matrix of the base coordinate system:

\[
\text{Rot} = \begin{bmatrix}
\vec{x}^T & \vec{y}^T & \vec{z}^T
\end{bmatrix} = \begin{bmatrix}
-0.3886 & 0.5636 & 0.7289 \\
0.212 & 0.8246 & -0.5245 \\
-0.8967 & -0.0493 & -0.44
\end{bmatrix} \quad (5)
\]

According to the position of the center 'O' of the circular hole, the homogeneous transformation matrix of the base coordinate system can be established:

\[
\begin{bmatrix}
465.5653 & -475.2322 & 665.6352
\end{bmatrix} \quad (6)
\]

\[
T = \begin{bmatrix}
\text{Rot} & \vec{O} \\
0 & 1
\end{bmatrix} = \begin{bmatrix}
-0.3886 & 0.5636 & 0.7289 & 465.5653 \\
0.212 & 0.8246 & -0.5245 & -475.2322 \\
-0.8967 & -0.0493 & -0.44 & 665.6352 \\
0 & 0 & 0 & 1
\end{bmatrix} \quad (7)
\]

In Matlab, the function tr2eul in Robotics Toolbox is used to obtain the Euler angle required by the corresponding robot controller:

\[
eul = \text{tr2eul}(T, \ 'deg') = [-35.7398, 116.1018, -3.1467] \quad (8)
\]

In KRL language, when defining a workpiece coordinate system (Base coordinate system), it is necessary to input 6 parameters: the position of the origin \([x, y, z]\), and the rotation angle \([R_x, R_y, R_z]\) which are relative to the robot world coordinate system. According to the above calculation, the coordinate value of origin position 'O' can be assigned to \([x, y, z]\), and the eul assigned to \([R_x, R_y, R_z]\), so that every time the robot makes precise docking, it can make accurate docking along the Z axis in the workpiece coordinate system. Using the above calculation results, a complete docking process was achieved in the actual docking experiment, as shown in 'Figure 7'.
6. Conclusion
In the simulated docking experiment, the control system dynamically controls the robot to successfully complete the precision docking. By means of manual measurement, some characteristic points of the docking workpiece in the robot coordinate system are found, and the workpiece coordinate system of the robot is established according to these characteristic points, so as to complete a precision docking work. In the following work, the method of manually measuring feature points will be upgraded to the method of robot vision measurement, which will greatly improve the accuracy and stability of the system and can be applied to the field of aerospace assembly.

Acknowledgments
This work was supported by National Key R&D Program of China(No. 2017YFB1303701), the Project of Intelligent Manufacturing Integrated Standardization and New Model Application, Chongqing Artificial Intelligence Technology Innovation Major Thematic Projects (No. zdyf0027), Made-in-China 2025 Sichuan Action Fund Project(2017ZZ003) and the Project of Intelligent Manufacturing Integrated Standardization and New Model Application 2017.

References
[1] Liu wei, Liu feng, NI yang yong, Bai xue and Li fei 2018 Astronautical Systems Engineering Technology 2 33-36
[2] Yao li zhao, Fang wei and Zhu xiao jin, 2018 Instrumentation Technology 9 9-12
[3] Li shi cheng, Xu shi bao, Zhang yun jian, Cheng xu and Wang shi gang 2017 36 75-78
[4] Wu hang 2017 research of active compliant technology and application software development for handling and assembly robot Southeast University

[5] John J. Craig 2005 Introduction to Robotics: Mechanics and Control chapter 2 pp 14-29

[6] A. Glamnik and R. Safaric 2012 9th International Conference on Remote Engineering and Virtual Instrumentation control of kuka kr 5 robot with a haptic device