Virtual Commissioning Modeling and Simulation of Assembly System for Quasi Natural Language Programming

Ya jie Yang¹, Zhi jing Zhang* and Ming kuan Shi²

¹ School of Mechanical Engineering, Beijing Institute of Technology, 5 South Zhongguancun Street, Haidian District, Beijing, 100081, P.R.China
*Corresponding author’s e-mail: zhzj@bit.edu.cn

Abstract. This paper focuses on a kind of quasi natural language programming for automatic assembly system, and studies the modeling and simulation of virtual assembly prototype, including the modeling of mechanical system, control system and the error model of assembly prototype. A multi-domain model is formed based on interfaces between different professional modeling and analysis software. Based on this model, the quasi natural language program is simulated and commissioned. At the same time, the script program is run on the physical prototype to verify the consistency of the assembly system prototype. Tested by experiments, the virtual commissioning modeling and simulation technology of the assembly system can approximately restore the assembly process of the real prototype, perform lifecycle management on the development of the equipment, and improve the commissioning speed and production quality.

1. Introduction
In the process of continuously promoting the "Industry 4.0" and "Made in China 2025" strategies, traditional manufacturing is gradually transforming and upgrading to intelligent manufacturing[1]. However, in the process of transitioning to intelligent manufacturing, virtual commissioning technology for product lifecycle management is particularly important.

The virtual commissioning technology refers to the creation of digital models in the physical environment through virtual reality technology, which are used to test and verify the performance of a series of devices[2]. Products are often designed in the order of "design-debug-verify" during the design phase, which makes the period longer. Virtual commissioning allows the designer to make any modification and optimization before production. During the program verification process, the running state of the prototype can be simulated by 3D simulation software, meanwhile, the automatic assembly system can be improved in time.

This paper proposes a method for commissioning assembly systems virtually based on quasi natural language, and performs modeling and simulation verification for this purpose.

2. Quasi Natural Language Programming
Quasi natural language programming is a reconfigurable programming method for automatic assembly systems, acting on a high-precision pulse sequence multi-axes controller (G-MAC[3]). This programming method divides the automatic assembly process into the smallest action units, which can be configured in servo mode or step mode. Different assembly processes are realized through serial and parallel connection of different action instructions. The assembly system has certain generality and reconfiguration to complete the automatic assembly process for various assembly processes of
different parts. The programming instruction sample is shown in Fig. 1.

![Diagram](image)

Figure 1. Quasi natural language programming instructions.

In the virtual layer, this paper builds a virtual motion control system using NI Softmotion. It reads and parses quasi natural language programming script files, and uses motion functions to drive virtual prototypes in 3D software for automated assembly. In the real hardware layer, the edited control instructions are sent to the G-MAC through the serial port, which can automatically identify the instructions and complete the control of the multi-axes motion, I/O signals and sensors, limit switches and other signals. And then complete the assembly process of the automatic assembly system.

3. Data Parsing and Logical Controlling

Before performing virtual commissioning, the script files need to be parsed into the information required in NI Softmotion, including motion resources (axes numbers), enable, stop, speed, acceleration and position information. There is a certain relationship between these information, such as sequential structure, nested relationship, etc. The basic structure of data storage and query selects the Hash table structure for fast query of nodes. Hash table is a data structure that is accessed directly according to the key value. The bottom layer is generally an array structure [4].

When we store the data value in the hash table, we will get the key value corresponding to the input information. After the operation, we will get the slot in the corresponding array, which denoted by K; According to the original address value, the corresponding physical address under each different K value is obtained. Then, the data is obtained by querying the address. The address structure is:

$$\text{address } x = \text{address}0 + K \times M$$  
(1)

Where address is the address; K is the slot; M is the number of bytes; Taking the absolute motion instruction as an example, the expression structure in the hash table is shown in Fig 2:
By reading the script file through LabVIEW, the axis information required for NI Softmotion can be obtained after parsing. The implementation method in LabVIEW is shown in Figure 3.

The script instructions are read cyclically and configured as input information to the control unit. Different axes movements are controlled in parallel. After the simulation is completed, the current cycle of all axes is ended. There are three end conditions: fault stop, manual stop and automatic motion end. When the end condition is a fault stop or manual stop, the simulation is over. Restart the simulation by repairing the fault and checking the model. Until the programming script file is completely read, the operation ends. The interface display is shown in Fig 4.

4. Mechanical System and Control System of Virtual Prototype

4.1. Mechanical System
The basis of automation systems virtual commissioning is the establishment of a "mechanical-control" simulation environment. The mechanical model of the assembly system is built in Solidworks. The whole system builds 5 linear axes, 2 rotary axes and 5-axis robotic arms, as shown in Fig 5.
In order to achieve the precise assembly of microminiature parts, the assembly system needs to be able to independently complete at least one or more assembly processes. Therefore, the assembly system modeling must be reconfigurable and extensible. Finally, a two-level modular design method oriented to system adaptability is adopted.

According to the classification of assembly objects and requirements, the assembly system is divided into 5 modules to complete the entire process of "loading and unloading, flexible clamping, image detecting, micro-adjustment and assembly". The modular structure of the assembly system is shown in Fig 6.

In order to make the prototype closer to the real, some physical field attributes are added to the model, including attributes such as mass, acceleration, friction, and spring force, which make the virtual prototype closer to the real physical prototype.

4.2. Control System
The control model of the assembly system is built in LabVIEW [5-6], which simulates the input and output of the real motion controller, and optimizes the control program through combined simulation.

The advantage of using a script programming file as the control program is that it already includes path planning and sequential logic design for motion control. We only need to build the smallest motion control unit—axis motion model in NI Softmotion.

Each axis unit constitutes the entire system, and the specific functions include: linear / rotary motion, absolute / relative motion, position reading and feedback of limit signals, etc. The LabVIEW model is shown in Fig 7.

Part 1: First, configure the positive and negative limit switches of the motion axis. Then, read the sensor data from Solidworks through the read function. In the end, it is used to inform the end of the travel of the motion system with high / low level signals.

Part 2: Configure the absolute rotary motion of the axis and read the current position in real time.
When the feedback value matches the input, the next position value is read.

**Part 3**: Configure the end conditions of the cycle: Fault alarm, End of motion and Manual stop.

In addition, for the automatic assembly system, the main function of the IO signal is to control the cylinder movement. The output signal controls the movement of the cylinder through the control model, and the input signal detects the current position feedback by the magnetic switch. In NI Softmotion, the movement of the cylinder uses a point-to-point distance method.

### 5. Error Model of Virtual Prototype

The error between the assembly parts and the based parts during the assembly process is studied, and the topology of the assembly system is described using a low-order array [7], as shown in Table 1.

**Table 1. Assembly system low-order-body array.**

|   | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| 0^j | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 0^j | 0 | 1 | 2 | 3 | 4 | 0 | 6 | 7 | 8 |
| 0^j | 0 | 0 | 1 | 2 | 3 | 0 | 0 | 6 | 7 |
| 0^j | 0 | 0 | 0 | 1 | 2 | 0 | 0 | 0 | 6 |
| 0^j | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0^j | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

0- Main body; 1-X axis; 2-Y axis; 3-Micro-adjustment turntable; 4-Fixture; 5-Based parts; 6-Z axis; 7-A. turntable; 8-A. fixture; 9- Assembly parts.

For the assembly system, the two branches are the assembly branch of the assembly and the micro-adjustment branch of the based part.

According to the multi-domain system kinematics theory and homogeneous coordinate transformation, the feature change matrix between the relative motion structures is:

\[
T_{SV} = T_{SV,p}T_{SV,p}T_{SV,J}T_{SV,p}e
\]  

(2)

The feature change matrix between relatively stationary structures is:

\[
T_{SV} = T_{SV,g}T_{SV,pe}
\]

(3)

For the assembly unit system, adjacent moving bodies include 12, 23, and 67, and adjacent non-moving bodies include 01, 34, 45, 06, 78, and 89.

The actual space position of the based part in the assembly unit system can be determined.

\[
T_{05} = T_{01}T_{12}T_{23}T_{34}T_{45} = T_{01,p}T_{01,pe}T_{12,p}T_{12,pe}T_{23,p}T_{23,pe}T_{34,p}T_{34,pe}T_{45,p}T_{45,pe}
\]

(4)

Similarly, the actual space pose of the assembly part is:

\[
T_{09} = T_{06}T_{67}T_{78}T_{89} = T_{06,p}T_{06,pe}T_{67,p}T_{67,pe}T_{78,p}T_{78,pe}T_{89,p}T_{89,pe}
\]

(5)

The actual position of the based and assembly part in the coordinate system can be expressed as:

\[
P_3 = [P_x, P_y, P_z, 1]^T = T_{05}[0,0,0,1]^T
\]

(6)

\[
P_9 = [P_x, P_y, P_z, 1]^T = T_{09}[0,0,0,1]^T
\]

(7)

The position deviation is:
\[ \Delta P_s = [\Delta x_s, \Delta y_s, \Delta z_s, \Delta T_s]^T = P_s - P_i \]  

(8)

\[ \Delta P_b = [\Delta x_b, \Delta y_b, \Delta z_b, \Delta T_b]^T = P_b - P_i \]  

(9)

During the assembly process, the component errors of each motion axis are brought into the formula, and the maximum coordinate displacement deviation that can be caused in the three coordinate axis directions.

\[ |\Delta X_s|_{\text{max}} = |\Delta X_b|_{\text{max}} + |\Delta X_r|_{\text{max}} = 0.0264 mm \]  

(10)

\[ |\Delta Y_s|_{\text{max}} = |\Delta Y_b|_{\text{max}} + |\Delta Y_r|_{\text{max}} = 0.0264 mm \]  

(11)

\[ |\Delta Z_s|_{\text{max}} = |\Delta Z_b|_{\text{max}} + |\Delta Z_r|_{\text{max}} = 0.0174 mm \]  

(12)

6. Simulation verification of virtual prototype

In this paper, a visual virtual prototype trajectory, logic sequence and collision detection are obtained through simulation. Compensate the movement error of each axis, check the deviation values of the X, Y and Z directions between the assembly and the based parts after operation, and compare with the actual operation. The results are shown in Figure 8-9.

![Figure 8. Error on real prototype.](image)

![Figure 9. Error on virtual prototype.](image)

7. Conclusion

For the natural language-like programming of the automatic assembly system, this research studies the modeling and simulation of the virtual assembly prototype, and simulates the virtual operation of the program. Based on the kinematic error model of the multi-domain system, the running error of the assembly system is compensated to the virtual prototype, and the movement track, logical sequence and collision detection of the assembly system motion are obtained, which verifies the consistency of the assembly prototype operation.

References

[1] Jianzhong Fu. Development Status and Trend of Intelligent Manufacturing Equipment [J]. Mechanical and Electrical Engineering, 2014, 31 (8): 959-962.

[2] Wang Chunxiao, Luo Weichao, Liu Riliang, Hu Tianliang. Multi-domain Modeling and Virtual Debugging of NC Machine Tools Based on Modelica [J] . Modular Machine Tool & Automatic Manufacturing Technology, 2018 (10): 102-105 + 110.

[3] Gaoyang Zhang. Research on autonomous control system technology of non-silicon MEMS flexible micro-assembly line [D]. Beijing Institute of Technology, 2014.

[4] Longzhen Li. Research on the Data Structure of the Special Spreadsheet Software [J].
Production, 2014 (13): 62-63.

[5] Gadzhanov, S.D., Nafalski, A., Nedic, Z. An application of NI SoftMotion RT system in a motion control workbench [C]. REV.2013.6502909.

[6] Ke-qing Wang. Motion simulation of robot for nuclear reactor pressure vessel non-destructive detection based on SoftMotion[A], Technology of China:IEEE BEIJING SECTION,2015:8.

[7] Yan Li, Dapeng Fan. Modeling and Analysis of Assembly Error of Three-axis Turntable Based on Kinematics Theory of Multibody System [J]. Acta Armamentarii, 2007 (08): 981-987.