Design and Automatic Modeling of Reconfigurable Software for Modular Robot

Zheng Li, Xin-an Pan, Wen-bin Gao
Shenyang Institute of Automation, Chinese Academy of Science, Shenyang
lizheng@sia.cn, panxinian@sia.cn, gaowenbin@sia.cn

Abstract

A kind of joint modular robot is introduced in the paper. The robotic modules, the expression method of software configuration, modeling and design of reconfigurable software have been described in detail. 3D model and kinematics of the robot are realized by using OpenGL techniques. The software can generate software model automatically based on a method of dynamic index of file system by user's configuration. Finally, its correctness is verified by experiments of kinematics simulation.

Keyword: *Modular Robot, Reconfigurable Software, Robotic Configuration, Automatic Modeling*

1. Introduction

Not only reconfigurable robot is necessary to achieve the reconfigurable hardware, but also real-time control software must be reconfigurable, so as to adapt to the rapid changes of robotic application range [1]. As an important branch of the reconfigurable robots, modular robot can be flexible combined and installed to a different configuration according to tasks need. Therefore, its control software needs to generate the robotic model of kinematics and dynamics to meet the control requirements. Modular reconfigurable robot is comprised of joints, links, the end-effector modules which have different sizes and features[2]. They can be used as the robotic body by the approach of "building-block" through a simple, rapid assembly and disassembly of the modules [3-8]. But the software method in reconfigurable robot is less presented. An approach of dynamic index table based on file system to achieve the robot configuration and automatic modeling is introduced in the paper. The correctness of reconfigurable software is validated by OpenGL simulation. It is a workable solution when the robot mechanical configuration changed, the software model how to be reconstructed.

2. Introduction of modular robot and its modules

The design takes a self-developed MRRS (Modular Reconfigurable Robot System) as a research object in the paper, its mechanical structure and modules are described below in Table 1.

Table 1. Robotic modules

index	module	description
1	Rl	Large Rotational joint
2	Rm	Middle Rotational joint
3	Rs	Small Rotational joint
4	T	Streach joint
5	L(Rl,Rl)	Right-bottom link,link to Rl
6	L(Rm,Rm)	Right-bottom link, link to Rm
7	L(Rm,Rs)	Right-bottom link, link to Rm, Rs
8	L(Rl,RmRsT)	Left-top link, link to Rl, Rm, Rs, T
9	L(Rm,RmRs)	Left-top link, link to Rm, Rs
10	L(Rs,G)	Left-top link, link to Rs, G
11	I(Rs,G)	Line link， link to Rs, G
12	P	Parallel link
13	B	Base of robot
14	G	Grape, end of robot

Each module has a standard mechanical and electrical interface to achieve the combination between different modules [2]. The free combination of 3 to 7 degrees of freedom of the robot and the same degree of freedom under different robotic configurations can be made by a flexible combination of the above modules to meet different tasks need.

There is an embedded controller in each joint module, it can communicate with PC reconfigurable software by CAN BUS, and receive control command to complete the control task.

3. Design of reconfigurable software

3.1. Design of robotic parts and parts library

The MRRS has various "hardware" modules, thus building the same "software" model corresponding to its mechanical module in software. Modular robotic parts library is comprised of these models. As same as mechanical assembly, the same method is used to "assemble" robot in software. Each part is comprised of part diagram, part pose matrix and 3D model (later will be described). Part diagram is a BMP icon file, using the image not only can show part type, but also can let user draw their own BMP part easily; pose matrix of part is described in text file;3D model is generated by 3Dmax tools. The interface is shown in Figure 1.

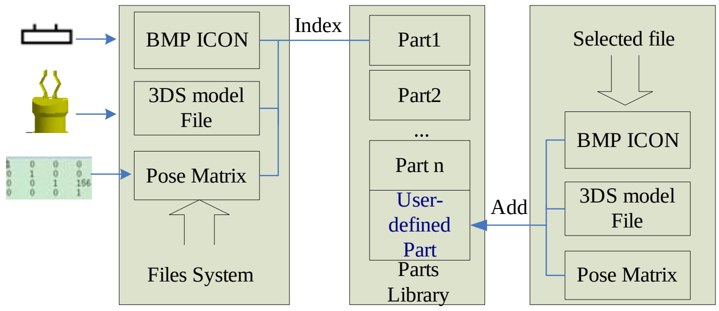

Figure 1. Introduction of Parts Library

The parts library has an interface for parts editing, users can draw the icon of part and set up its matrix parameters. The new part can be added and saved into parts library, and can be used in robotic configuration tree by user.

3.2. Design of robotic configuration tree

Software method of various parts of the robot is defined and described on above. When a robot is assembled, its configuration is fixed. The same model in software needs to be built according to the hardware. In this paper, a topology description method of robotic configuration tree is represented [9]. The "robotic configuration tree" just likes building blocks, user can place the robotic parts in the trunk optionally. The robotic parts can be selected from parts library freely by user. Therefore, user can "assemble robot" in software like mechanical assembly. Any configurations can be done in this mode. User interface is shown in Figure2.

Users can choose their parts form parts library and place in robotic configuration tree to "assemble robot". The BMP icon of part describes the formation of the robot configuration vividly. Each part can be set the rotational axis and the rotational angle; it is easy to determine the position and orientation of the part. When the graphic formation of the robot is saved, formed data in software is stored in a text file, which is stored into the robotic configurations library as a robotic configuration file.

3.3. Design of robotic configurations library

The robotic configurations library is comprised of a series of files. It includes two types of configurations. One is for the robot, the other is defined by user. User-defined configuration can be done by the robotic

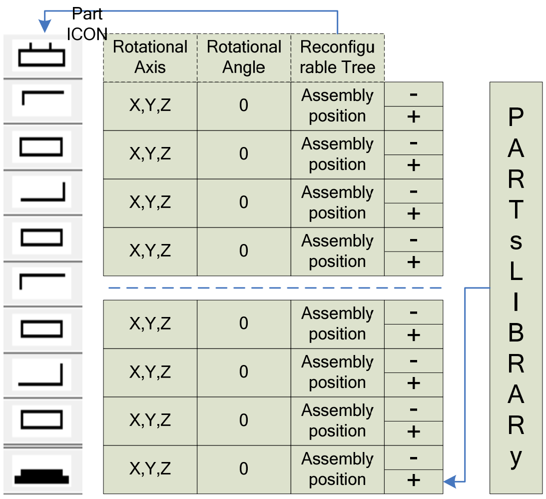

Figure 2. Introduction of robotic configuration tree

configuration tree as discussed above. Each configuration is described in a text file, and each file has a unified file format. As shown in the Table 2, the format and parameters of configuration is described.

Table 2. Format of robotic configuration file

Flag	Name	Position	Angle	Axis
Flag Header	0	0	0	0
Part description i	Part Name i	XYZ position i	Rotational Angle i	Rotational axis (XYZ) i
Flag tail	1	1	1	1

Here, “i” is the number of part added by user. It is obvious that a file is comprised of multiple parts. Each part can be set “name, position, angle and axis”. When the file is loaded by reconfigurable robot software, it can find 3ds model file and pose matrix file according to “part name” key by file system index. The two files are associated with this part. They can provide all data (position, orientation, 3ds model) for each robotic configuration and can be used as the data source of automatic modeling.

3.4. Dynamic file system index table

With the robotic text description of every configuration file, reconfigurable software should parse the name of various parts and extract parameters from the file. File system of reconfigurable robot software is comprised of four folders:

A. robotic configurations files folder

Robotic configurations file is saved in the folder. Each file is a robot configuration description. Its format has been described on above.

B. part icon folder

Each robotic part BMP file is saved in the folder. The BMP icon mapping the MRRS’s modules. Of course, users can add their own icon file to the folder.

C. part model folder

Each 3ds model of robotic part is saved in the folder. These model files are consistent with the physical modules in size, which can be generated by software tools, such as 3DMax. User can add 3ds model file in the folder for robotic simulation.

D. pose matrix folder

Matrix parameter file of each part is saved in the folder. Also user can add user-defined matrix parameters file to the folder.

File system index table is shown as Figure 3:

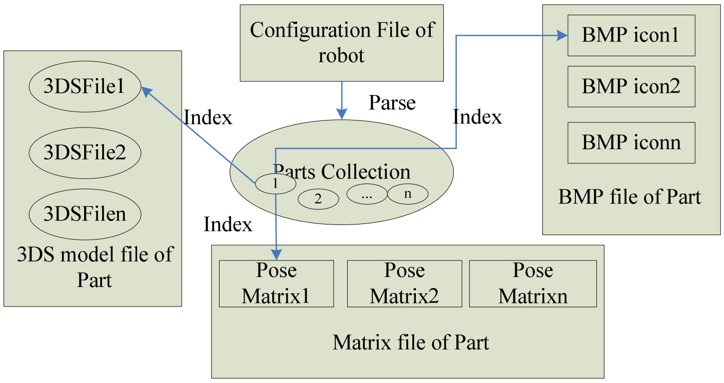

Figure 3. File system index table

It is obvious that the configurable software can find its file in folders by part name in configuration file dynamically. Therefore, the part information is obtained. The way of dynamic extraction and combination parts make it possible to support reconfigurable software modeling.

4. Design of automatic modeling in reconfigurable software

4.1. OpenGL simulation and coordinates system

OpenGL is a commonly shared open 3D graphics standard. It plays an important role in the field of simulation of three-dimensional modeling, virtual reality technology, 3D game development [10,11,12]. Because of 3ds file [13] is easy to be parsed by OpenGL, 3ds file model commonly used in software for 3D simulation. In robotic field, it can describe robotic simulation and display the dynamic process of robot working vividly. This is very useful for verifying the robot works, the workspace and the collision detection [14].

OpenGL's coordinates system complies with the computer screen coordinate system. As shown in Figure 4, the modular robot coordinate system is a defined coordinate system. They can be exactly equal by turning 90 degrees around X axis. Furthermore, OpenGL provides many rich API functions such as translation, rotate. Pose transformation can be done conveniently in program by the functions. They provide a basis for the automatic combination of parts in program.

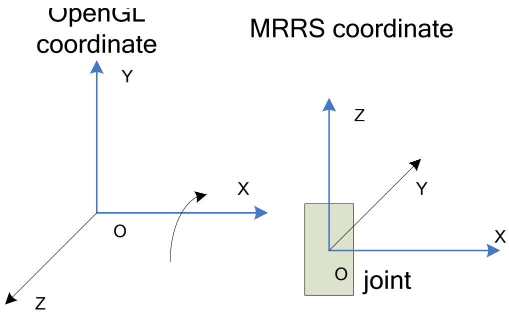

Figure 4. Coordinates system

4.2. Automatic modeling in OpenGL

Because modular robotic configuration is often not predictable, the robotic model needs to be generated automatically. The aim is achieved by 3DS files' automatic combination. The position and orientation of each part can be obtained by the rotation and translation transformation in OpenGL. As we know the angle and coordinates from each part in robotic configuration file, the robotic automatic modeling can be done by

OpenGL functions. Firstly, the reconfigurable software loads the configuration file. Secondly, information of each part is obtained from the file. Thirdly, according to method of file system index, getting each model file, pose file from their folders. Finally, from start to end, drawing each part model automatically in OpenGL workspace.

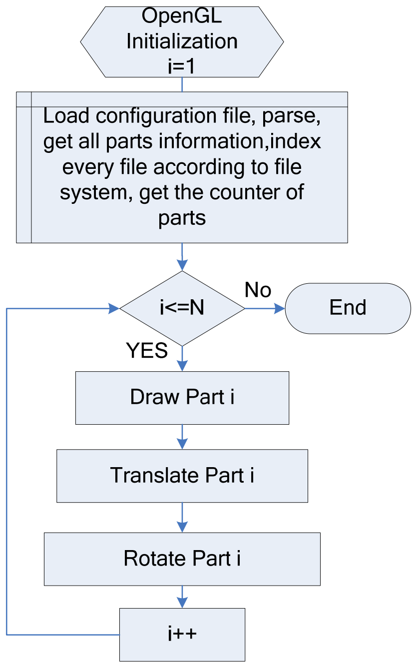

Figure 5. Program flow chart

5. Experiment

5.1. Model building

After the development is complete, it is necessary to make a test to verify the realization using the reconfigurable software. According to the physical robot (6 degrees of freedom), the "soft assembly" is done in the reconfigurable software. The two steps are as follows:

A. Assemble the robot model in robotic configuration tree. As be shown in left of Figure 6.

B. Generate 3D model automatically. As be shown in right of Figure 6.

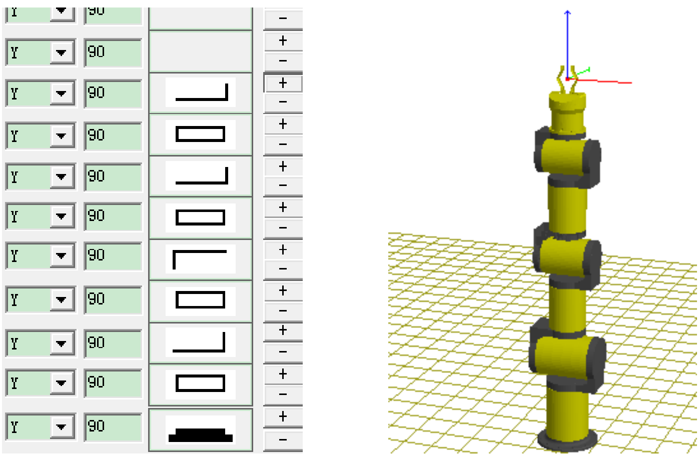

Figure 6. Model building

5.2. Kinematics test

The kinematics of the modular reconfigurable robot is based on the results of the topological analysis[15].

A. Forward kinematics

Each joint is given an angle value, 3D simulation environment will finish forward kinematics sport and give simulation results. As be shown in left of Figure 7. The right picture shows the pose matrix of the current status.

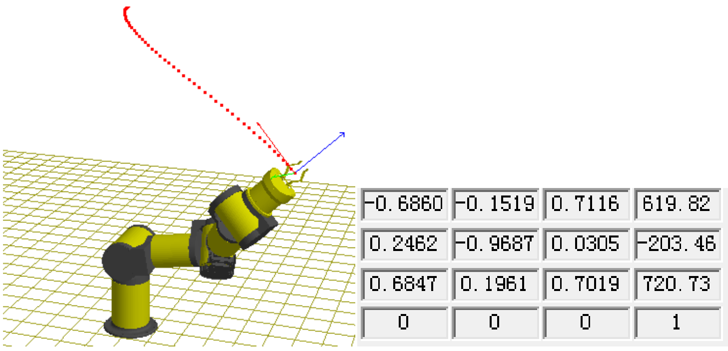

Figure 7. Forward kinematics test

Jionts' values are listed in the Table 3.

Table 3. Test values

name	Joint1	Joint2	Joint3	Joint4	Joint5	Joint6
degree	90	-80	50	60	40	-45

B. Inverse kinematics

Setting the end of the robot position and orientation, The pose matrix is shown in left of Figure 8. After kinematic calculations, the angle of each joint is obtained (the right picture) and set to simulation environment (the middle picture).

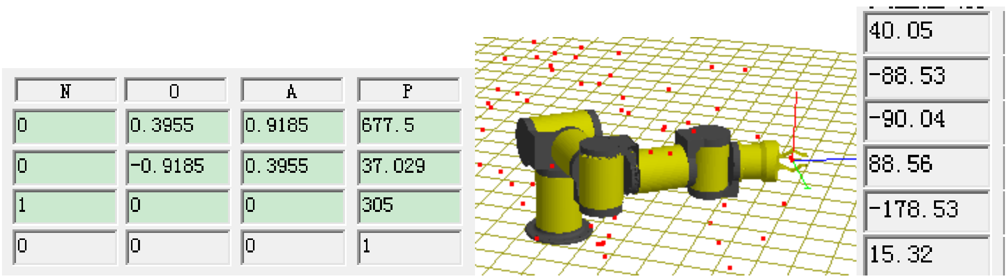

Figure 8. Inverse kinematics test

The result of experiments shows that the robotic models built by reconfigurable software can be generated automatically. The kinematic sport shows that the combination is right. Therefore, the realization of reconfigurable software is feasible and correct.

6. Conclusion

A kind of reconfigurable software based on file system index method is designed according to the research of MRRS, and the parts library and robotic configurations library are all developed in the software. Automatic modeling and reconfigurable development are all realized by the method of robotic configuration tree. Furthermore, it is very important to discuss robotic kinematics and dynamics for reconfigurable software.

7. References

[1] Bing Wang, Qin Jiang, “Review on the Status and Development of Modular Reconfigurable Robot Technology”, Mechanical & Electrical Engineering Magazine, vol. 5, no. 5, pp. 1-4, 2008.
[2] Dongxing Tian, “Kinematics Simulation 0f 3-D0F Planar Parallel Manipulator Based on OpenGL”, Coal Mine Machinery, vol. 31, no. 9, pp. 55-56, 2010.
[3] Yong Jiang, Hongguang Wang, Xinan Pan, Cen Yu, Neng He, “Autonomous Online Identification of Configurations for Modular Reconfigurable Robot”, Chinese Journal of Mechanical Engineering, vol. 47, no. 15, pp. 17-24, 2011.
[4] Yanli Zhang, “Experimental Study of Reconfigurable Modular Robot Systems”, Journal of Shenyang Institute of Aeronautical Engineering, vol. 26, no. 1, pp. 35-37, 2009.
[5] Yanli Zhang, Jingkui Li, Shujun Li, “Kinematic Analysis of Reconfigurable Modular Robots”, Machinery Design & Manufacture, no. 10, pp161-163, 2008.
[6] Yanli Zhang, Shujun Li, Jingkui Li, “Inverse Kinematics Modeling for the Reconstruct-able Modularized Robots”, Journal of Machine Design, vol. 25, no. 11, pp. 29-31, 2008.
[7] Yanli Zhang, Jinfeng Che, Shujun Li, “Kinematic and Workspace’s Analysis of Reconfigurable Modular Robots”, Machinery Design & Manufacture, no. 10, pp. 146-148, 2010.
[8] Mingyao Liu, Dalong Tan, Bin Li, “Status and Development of Reconfigurable Modular Robots”, Robot, vol. 23, no. 5, pp. 275-279, 2001.
[9] H. W. Liu, H. G. Wang, S. J. Li, L. B. He, “Research on the Topology Description and Modeling Method for Reconfigurable Modular Robots”, In Proceedings of ASME/IFToMM International Conference on Reconfigurable Mech. & Robots, pp. 129-135, 2009.
[10] Zhixiao Yang, Jinchi Han, Erhui Ou, “Optimization of Virtual Robot Modeling Based on OpenGL”, Computer Systems & Applications, vol. 11, pp. 65-74, 2008,.
[11] Guangzhi Zhao, Hongguang Wang, Yuwang Liu, “Research on Configuration and Kinematics of Joint Modular Robots”, Machinery Design & Manufacture, no. 4, pp. 178-180, 2010.
[12] Chenxu Zhang, “Technology Research Based on OpenGL about Robotics Simulation”, Mechanical Management and Development, vol. 24, no. 3, pp. 107-171, 2009.
[13] Hongqiang Chang, Shuang Liu, Guohao Li, Na Wu, “3DS File Parsing and Implementation of Viewpoint Movement and 3D Selection”, International Electronic Elements, vol. 19, no. 20, pp. 14-17, 26, 2011.
[14] Songtao Zhang, “Stabilization for Discrete-time Takagi-Sugeno Fuzzy Control System”, AISS: Advances in information Sciences and Service Sciences, vol. 3, no.10, pp. 355-362, 2011.
[15] Songtao Zhang, Xiaowei Zhao, “Fuzzy Robust Control for an Uncertain Discrete Switched Closed-loop Supply Chain with Time-delay”", IJACT: International Journal of Advancements in Computing Technology, vol. 4, no. 14, pp. 92-104, 2012.