

Li MA, Linlin CI, Genyan GE

Design and implementation of a multi-axis precision movement machine based on MAS theory

© Higher Education Press and Springer-Verlag 2008

Abstract A model construction of a multi-agent system (MAS) and the basic function of the agent are described. The precision control method using the multi-CPU of a programmable logic controller (PLC) is introduced, and a distributed method using multiple CPUs to control different motion machines is given. The test results indicate that in industrial control fields, the combination of using the credible PLC to control the motion machine and multi-CPU task distributing methods can solve multi-axis machine linkage and implication, providing a more credible method for multi-axis motion units.

Keywords multi-agent system (MAS), token, multi-CPU system

1 Introduction

The traditional method of motion control based on precision mechanism is appliance transmission control. It can be considered innovative in terms of the realization method to combine the agent theory of artificial intelligence to control the mechanism, especially in a robot and its applications. The application of multi-agent system (MAS) theory of artificial intelligence in research on systems, which is a current research hotspot, enables each part of a system to have the possibility of self-organization [1]. Similar to the biosphere, such as that for ants, honeybees and other social insects, the community realizes a complicated target that seems impossible to a single individual through autonomy and cooperation. This shows that the community's cooperation behavior is significant

Translated from *Journal of North University of China (Natural Science Edition)*, 2007, 28(1): 32–37 [译自: 中北大学学报 (自然科学版)]

Li MA (✉)

College of Information Engineering, North China University of Technology, Beijing 100144, China
E-mail: mali@263.net

Linlin CI, Genyan GE
Beijing Institute of Information Technology, Beijing 100085, China

[2]. The robot team system CEBOT proposed by Fukuda adopted the way that several robots cooperated to work, making the system possess an assembly and disassembly ability [3]. The advancement of technology and demand of applications have made it very difficult for a single robot to complete complex tasks. This article puts forward a type of multi-axis precision movement control method based on MAS theory to precision system linkage. An MAS is then constructed based on the popular belief desire intention (BDI) structure and focused on the robot. The whole system involves factors that include logistics, assembly, test, simulation, visual inspection, speed restraint, and acceleration. This system also involves many agents, and they constitute a system that follows a certain protocol and completes a task together using the robot team. The authors place emphasis on the strategy and implementation of multi-axis precision movement machine linkage control by using multiple CPUs.

2 BDI system model

An intelligent agent is a kind of system in a dynamically changing environment. This type of system has an incomplete understanding of targets, and thus local control can be realized by carrying out the behavior. A typical agent can be assigned many duties (these duties may have mutual conflicts), and to meet these targets, the agent is required to generate an effective sequence of behaviors and make decisions [4].

To design an architecture that meets all needs, many solutions have been put forward and introduced in detail in Ref. [4]. The BDI structure model is a popular solution raised by Rao et al., and it is translated as a beliefs-desires-intentions model in Chinese [5]. A typical BDI structure model is shown in Fig. 1. There are 4 key data components in a BDI structure: the agent's beliefs, desires, intentions and plan library.

Beliefs are information the agent has with respect to the world. They are facts that have the probability of realization; desires are the assignments that are connected

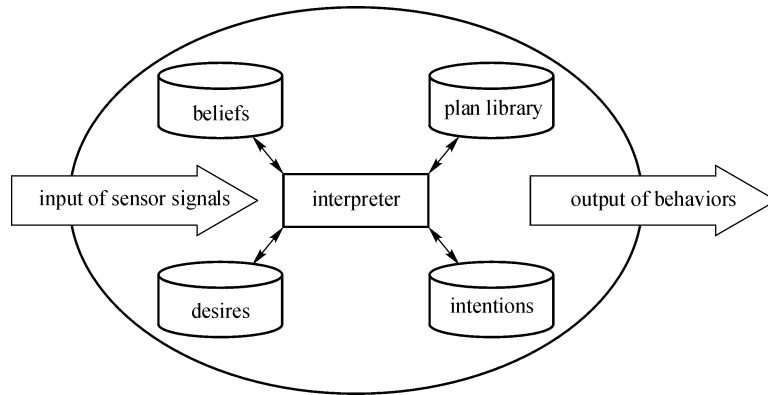


Fig. 1 A BDI agent architecture

with the agent, they may fail frequently, but the agent model requires them to be fixed; agent's intentions are the desires for success. Generally speaking, although desires are fixed, not all of them can be realized. Therefore, an agent must select some subsets of desires and find a way to implement them. These selected desires are called intentions.

The last data structure of the BDI agent model is the plan library. It is the set of plans that have formulated specific action proposals that may be used by the agent to achieve its purposes. For the sake of giving an official semantics to the BDI structure, Rao and Georgeff have developed a series of logical descriptions [6]. This logic is expanded on the basis of branch time logic CTL* [7], which is also a logic model that contains beliefs, desires and intentions. Most of the work concerning BDI logic concentrates on the possible relationship between 3 intelligence statuses (namely beliefs, desires and intentions), while the latest researches focus on restraints in proving the logic.

3 Control of precision movement machine within system

In general, a system can be divided into a central control system and a distributed control system depending on the presence of an intelligent agent. In the centralized control system, the administrator is responsible for planning and making decisions; its coordination efficiency is high, but the characteristics of real-time and dynamic property are poor. In the distributed control system, individuals have a high degree of autonomy and can complete tasks with cooperation by virtue of communication methods. Hence, it is much better compared with the centralized system in terms of fault-tolerant ability, reliability and elasticity.

Since a system is a multi-work organization that involves assembly, test, vision, and other factors, the centralized system adopts an MAS to improve the reliability. This system involves six agents in total: system general control agent (S-agent), robot agent (R-agent), test

agent (T-agent), vision agent (V-agent), assembly agent (A-agent) and other agent (O-agent). Each plays a different role in the system, and the structure between each is different. They can be made to exist in a system by adding constraints and modeling again. The set $D_{Ag} = \{S\text{-agent}, R\text{-agent}, T\text{-agent}, V\text{-agent}, A\text{-agent}, O\text{-agent}\}$ is made up of all the agents in the system, and each element includes its own plan library.

The authors shall only depict the design and implementation of a multi-axis precision movement machine in an auxiliary agent.

This system depends on an ultra precise assembly system, whose main duty is completing the adhesive fitting at a precision of 0.01 mm. This system involves many interdisciplinary subjects, including computing, control, mechanics, and metrology. To achieve assembly precision, the design of the machine is a very important part. However, it is not our emphasis here, and detailed information can be found in the references. Besides machine design, control strategies and methods also play key roles.

3.1 Constitution of a control system

The system general control model is shown in Fig. 2. This system contains six primary agents. They cooperate to work when needed; otherwise, each works separately to complete its corresponding task. Each agent submits its plans and behaviors that need exchanging with the interactive channel, and makes use of this channel to complete the information exchange between agents. The functions of each agent in the system are as follows:

1) System general control agent (S-agent)

S-agent is in charge of controlling the system's overall process; giving orders to let the system start, suspend, stop, reset; and monitoring the process. It is the system's coordinating agent.

2) Robot agent (R-agent)

R-agent is composed of seven degrees of freedom redundant robot PA-10-7C – a robot equipped with a power monitoring sensor and RCC pliancy in the terminal.

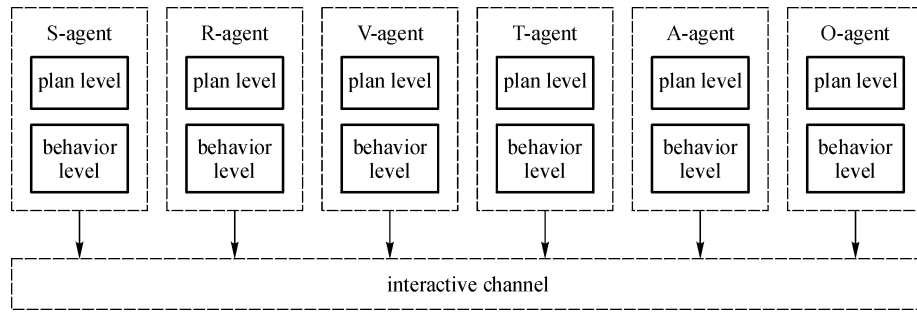


Fig. 2 Assembly of MAS model

R-agent is mainly responsible for transporting the workpieces and exchanging work and tools in the assembly assignments. In the process of finishing the assignments, there are two aspects of signals: input signals and output signals. Input signals are sent out from the system general controller. They are always command signals, such as ‘bring workpiece No.1 to location No.1’. Meanwhile, output signals have two forms. One is some parameter signals contained in the movement of the robotic arm, such as seven joints’ angle and extremity’s coordinate value; the other is the auxiliary sensor’s signals from three directions.

After receiving the command signals from the master controller, the arm finishes the corresponding behaviors by itself and does not need the system’s instructions. In this process, it finishes the assignments that the system requires automatically in normal conditions unless there appear exceptional issues, in which case the arm would be interrupted.

3) Test agent (T-agent)

T-agent is responsible for completing test assignments that the index requires. It is composed of a measurement device driver and test sensor. The measurement device driver can bring the test sensor to the designed location to make the sensor work in its activity scope and get the assignments done. T-agent is autonomous during the testing; it would encapsulate the results and transmit them to the interactive level.

4) Vision agent (V-agent)

V-agent’s primary mission can be separated into three aspects. The first is monitoring the entire work process: once there is unusual situation, it would send out the exceptional report to S-agent with the aim of ceasing the process, and people can participate when necessary to stop the process forcefully. The second is examining the coordination degree between the workpieces’ fitting surfaces, to see whether the coordination slit satisfies the matching requirement. The third is monitoring the workpieces’ surface to determine whether they have flaws and other issues.

5) Assembly agent (A-agent)

A-agent takes charge of assembling workpieces. The procedure involves holding the workpiece’s upper end on the assembly at the one-dimensional stage, and fixing the

workpiece’s lower part in the rotating floor’s assembly location. A-agent has a power sensor, RRC, to provide supplementary means for the process of assembling. The assembly behavior’s realization is controlled by the step motor.

6) Other agent (O-agent)

O-agent coordinates with the system’s primary agents to finish the task. Its prototype is a precise rotating-floor and one-dimensional platform that supports workpieces and tools. A servo-actuator and movement machine constitute the O-agent. In the process of finishing the task, it can accept other agents’ requests to drive the work platform to make suitable movements, and when finishing the task, it can inform the corresponding agent.

3.2 Control of precision movement machine

Among the agents described above, R-agent is made up of a multi-degree of freedom redundant robot PA-10-7C. Since it is relatively independent and the control policy is mainly manifested in its coordination with other agents, it is not the emphasis here. The other agents, T-agent, V-agent, A-agent, O-agent, are composed relatively independent structures. During the motion of these agents, the seven axes’ control drive are involved. According to the requests and different control precisions, the system uses one servo-actuator and six step motors to form seven independent control axes. As each axis’ movement is independent and parallel running and mutual coordination are needed, we must consider the relative independence and cooperation ability of each axis’ control in realization.

The control system is mainly composed of one industrial control computer and four Siemens PLC226s. The industrial control computer is regarded as the upper machine, while the programmable logic controller (PLC) is regarded as the down machine. The four down machines run in parallel, use a bus-oriented structure, apply a free interface communication method and formulate a concrete communication control protocol to realize contact between one PC and four down CPUs. The system constitution is shown in Fig. 3. In this article, the authors would no longer depict the down machine’s control of each precision

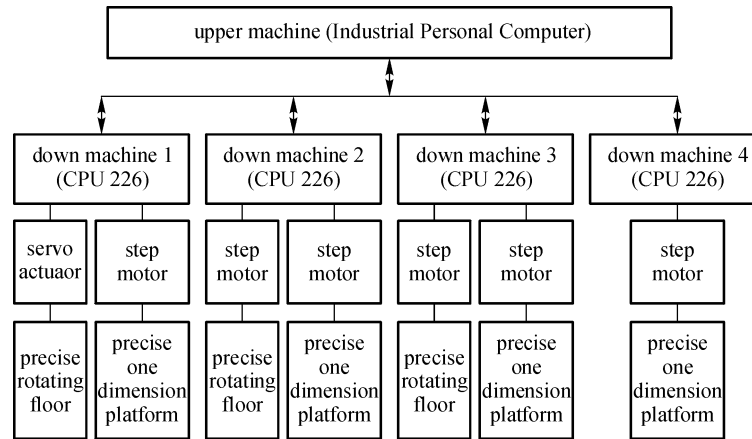


Fig. 3 Control structure

movement machine. Emphasis is on the communication control protocol and control strategy between the upper machine and the down machine.

The upper machine is a general-purpose computer. Considering the anti-interference ability and the necessity of real-time control, the down machine selected a CPU226 that is part of the Siemens series S7-200. The controlled objects and the restrictions of communication protocol must establish their own communication agreement, to implement the communication between the Siemens programmable controller and the general industrial personal computer.

Free interface communication was used in the realization and the communication protocol should be defined according to the specific situation.

3.2.1 Communication protocol

This system includes five CPUs of two kinds: one general purpose computer and four CPU226 in the Siemens series S7-200. It contains more than one CPU and is part of an isomeric multi-computer system. To guarantee the system's reliability and stability, we applied the PLC system of the reliable Siemens series S7-200 to control the movement machine during design.

The communication protocol adopted is an Echo protocol, which was divided into an upstream and a downstream. The PC sent out a downstream format to the PLC, which was called a command message. The upstream was sent out by the PLC to the PC, which was called a status message.

1) Format of a command message

In fact, the PLC controls two kinds of objects. One is the revolution of electric motors, which includes a step motor and servo-actuator. The other is various status switching values generated during the revolution of electric motors, which includes a paid-in test switch, zero switch, and limit switch. The format of the frame is shown in Table 1.

The meaning of each flag is as follows:

start mark: the start of the message, defined as '%';

flag: the current motion is in manual mode or automatic mode, 1 stood for automatic mode and 2 stood for manual mode;

type: what kind of object did this message send out to. 1 stood for motor, 2 stood for solenoid valves, 3 stood for inquiring the status of the down machine, 4 stood for ordering a down machine's movement mechanism to go back to zero, 5 stood for stopping the motor;

platform number: the number of the four PLCs that the command messages were sent out to. As there were four PLCs in the system, we could use the platform to tell them apart;

equipment number: the number of the specific equipment that was managed and controlled by the PLC;

direction/switch: when the 'object' was 1, this bit stood for the revolving direction of the motor; when it was 2, this bit stood for establishing the state of the switch. Additionally, 1 stood for on, while 0 stood for off;

frequency: described the revolving speed of the motor in kHz. This bit was only affected when the 'flag' was 1;

step: described the number of pluses generated by the revolving motor. This bit would not be affected unless 'flag' was 1;

check code: we used BBC to check. This bit could be

Table 1 Format of frame

name	start mark	flag	type	platform number	equipment number	direction/switch	frequency	step	check code	end mark
bytes	1	1	1	1	2	1	4	8	2	1
example	%	1	2	A	13	0	1090	11345678	1234H	#

obtained by doing the operations of XOR and adding the 14 bytes of data in front of the command message data;
end mark: the end of the message, defined as ‘#’.

2) Format of the status message

The format of the status message is shown in Table 2.

The meaning of each flag is as follows:

start mark: the start of the message, defined as ‘%’;

flag: had the same meaning as the command message’s ‘flag’;

type: had the same meaning with the command message’s ‘type’;

platform number: the number of the down machine PLC, it occupied one byte and was marked in the unit of the PLC CPU;

equipment number: the number of the equipment managed by the PLC;

status code: down machine’s status code. 0000 meant that the down machine was empty, other codes represented different statuses;

check code: we applied BBC to check;

end mark: the end of the status message, usually defined as ‘#’.

3.2.2 Communication control

Based on the communication protocol described earlier and the defined formats, the upper machine can send orders to down machines or receive the results returned from down machines. Because down machines had four CPUs, it was considered as multi-computer communication. The upper machine was the master station and the four down machines were slave stations. Communication was initiated by the upper machine, but if the down machines detected something unusual in the local place, it should report to the upper machine proactively. The communication mode in normal conditions was ‘the upper machine sent out orders→the down machine replied’. But once exceptions appeared in the down control, it should report the errors to the upper machine. If there was one more down machine at this moment, i.e., when two or more down machines sent out error reports to the upper machine, this would result in message conflicts and mistakes. To avoid such a situation, when the normal system ‘order→reply’ actually running was ended, the upper machine sent out a token and the four down machines could receive it at the same time. When a down machine detected an exception, it would receive the token. The bus would stop transmitting the token, and the down machine that received the token sends out a message to the upper machine while others were waiting. After the down

machine that held the token completed sending the message, the upper machine would send out the token to the bus again and other down machines could continue receiving the token and conveying information to the upper machine.

3.2.3 Assignments and implementation of down machine

Communication is very common between PCs, but utilizing PC and PLC to realize multi-computer control is not common in practice. In the MAS system, the implementing agency is multi-agent, where various intelligence agents may make choices and decisions according to their actual conditions. Applying several CPUs in the system to complete the jobs assigned by the system separately and in coordination seems to have a remarkable effect on the use of multi-intelligence agents in industrial control.

The PLC is an individual in multi-intelligence agents. It completes its independent movement as it coordinates with others to finish the overall task of the system. The system used in this article involved seven axes’ control. Moreover, the precision of the movement machine that each axis controlled was required to be within 0.01 mm. Therefore, it was very essential to complete the task alone with high quality. In the meantime, the system’s assignments were accomplished by several interacting movement machines, and thus cooperation was also very essential. In the precision control equipment’s type selection, the system chose the high accuracy revolution platform and its realization of control was implemented by the PLC.

The down machine used the PLC system to run. After the initialization of the system, the down machine stayed in a state of receiving orders, waiting for the upper machine’s commands. The upper machine broadcast orders to all down machines when needed, carrying a down machine’s platform number in the command message, and all the down machines could receive the orders at the same time. After checking the node and decoding the message, they compared their local numbers with the platform number separately. If the platform number was consistent with a certain down machine’s local number, it was the machine that the order should be received with. It would then call corresponding subprograms to finish the task as the order required. Otherwise, it would do nothing.

It should be pointed out that after down machines complete receiving orders, they should send out answering signals to the upper machine. To prevent network conflicts resulting from several down machines transmitting signals at the same time, when the down machines finish

Table 2 Format of status message

name	start mark	flag	type	platform number	equipment number	status code	check code	end mark
bytes	1(‘%’)	1	1	1	2	4	2	1(‘#’)
example	%	1	2	2	14	0002	1234H	#

transmitting answering signals, they set their own status as a receiving state, and would no longer transmit any data before receiving the next batch of orders.

The upper machine's communication flow is shown in Fig. 4.

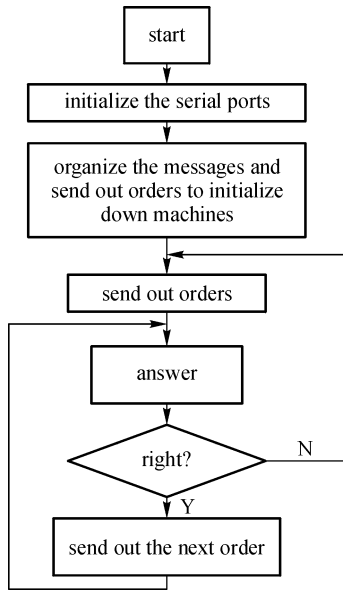


Fig. 4 Communication flow of upper machine

The structure of the down PLC's main code is as follows:

initialize port 0 (port 1): set port 0 (port 1) in the receiving state; put received data into the receiving buffer; decide whether the 'platform number' in the received data is consistent with the local number: if they are the same, the system would decode it; check the code and the transmitted return value, ... ; else, the system would do nothing. The simple flow chart is shown in Fig. 5.

4 Conclusions

The controls of different parts within the multi-agent system are different. Each agent can choose a suitable control method to accomplish a corresponding function based on its own condition, and the control method may not be the same for other agents. The system in this article uses a master agent to send out orders to programmable logic controllers, which would then implement the specific control to a multi-precision movement machine. Experiments prove that the motion system has high accuracy and can run steadily, and its control method is obviously effective. Reliability and stability are very important for industrial control in reality. The control method that employs a reliable PLC as the main part of industrial control has been used, although it is much more efficient to adopt a multi-CPU to achieve distributed control.

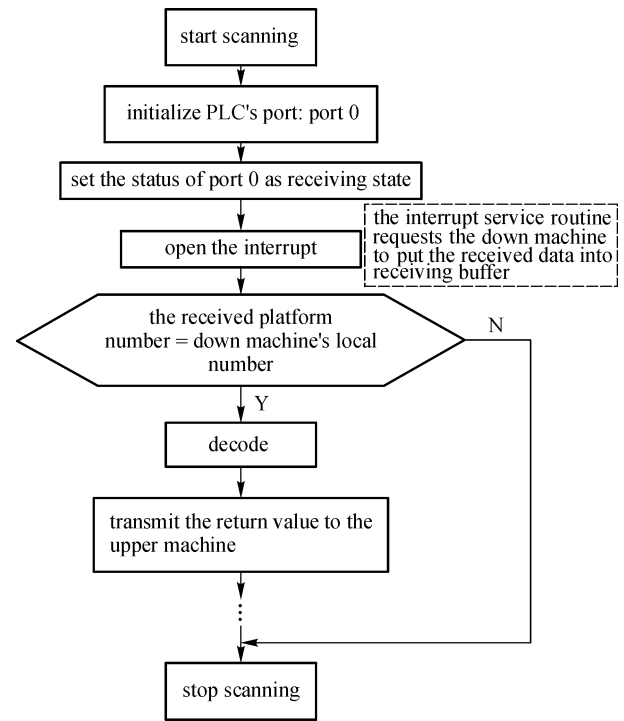


Fig. 5 flow chart of a down machine

Acknowledgements This work was supported by the Tenth five-year-plan of the China Fund of National Defense Advance Research (No. 4131607301).

References

- Nolfi S. Evolutionary robotics: exploiting the full power of self-organization. *Connection Science*, 1998, 10(3-4): 167-184
- Cao Y U, Fukunaga A S, Kahng A B. Cooperative mobile robotics: antecedents and directions. *Autonomous Robots*, 1997, 4(1): 7-27
- Fukuda T, Nakagawa S, Kawauchi Y, Buss M. Structure decision method for self organising robots based on cell structures-CEBOT. *IEEE International Conference on Robotics and Automation*, Scottsdale Arizona. 1989, 2: 695-700
- Wooldridge M J, Jennings N R. Intelligent agents: theory and practice. *The Knowledge Engineering Review*, 1995, 10(2): 115-152
- Rao A S, Georgeff M P. An abstract architecture for rational agents. In: *Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning (KR&R-92)*, 1992, 439-449
- Rao A S, Georgeff M P. Modeling rational agents within a BDI-architecture. In: *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR&R-91)*, 1991, 473-484
- Rao A S, Georgeff M P. Formal models and decision procedures for multi-agent systems. *Technical Note 61*, Australian AI Institute, Level 6, 171 La Trobe Street, Melbourne, Australia. 1995