Proceedings of the IEEE
International Conference on Automation and Logistics
Qingdao, China September 2008

Research on Synchronous Control of Nodes in Distributed Network System*

Fang He
School of Control Science and Engineering
University of Jinan
Jinan, Shandong Province, China
cse_hf@ujn.edu.cn

Shouhua Zhao
School of Control Science and Engineering
University of Jinan
Jinan, Shandong Province, China
zxz_9999@163.com

Abstract –**The distributed network systems have been used widely in complicated electric drive and Mecatronics control systems. The research on synchronization control of distributed network nodes has been a hot point. This paper focuses on the hot point, synchronization environment and synchronization object are discussed firstly. And then, the concepts of activating and deactivating to the program of network nodes are proposed. In order to solve the problem of isochronous data sampling or isochronous drive output, we propose a novel kind of control idea of synchronization, which include dividing slave's main loop program into several processes and adopting twice trigger signals to activate synchronization control. Besides, we analyze the time characteristics of synchronous executing of nodes. Finally, we make a dissection to the realization method of synchronization control in PROFIBUS-DP master-slave network system, and discuss briefly their conditionality in actual application. By compiling application program according to idea proposed by this paper, high isochronous control demand can be satisfied for distributed nodes in many cases.**

Index Terms – synchronization, activation, deactivation, PROFIBUS

Ⅰ. INTRODUCTION

The mainstream trend is that the distributed control systems with multipoint transmission mode have replaced gradually the traditional point-to-point control systems. The distributed systems are increasingly used in complicated electric drive systems and Mecatronics control systems. In the distributed industrial control systems, fieldbus is specially used for the interconnection process controllers, sensors and actuators at the factory automation hierarchy. It is very important to insure network nodes running harmoniously and synchronizing. So it has been focus point to realize synchronous control of nodes in distributed network systems.

It is known that time constraints are more stringent in the automation hierarchy. Especially, in many distributed network application systems, it is necessary for distributed nodes to synchronous sampling, harmoniously acting and executing some process response at the same time. In these application cases, the synchronous modes of distributed nodes are very useful. These nodes can enhance quality of control and improve precision of products, and reduce remarkably the effect on the control precision of whole network system, which caused by the time distinction of process response of distributed nodes.

Currently, the mainstream fieldbuses include FF, PROFIBUS, CAN, and LONWORKS. Different MAC mechanisms are adopted for different fieldbuses communication systems. For different kinds of fieldbuses, the solution to the problems of synchronization control relies on their own protocols, and limited by their demands of hardware and software. The purpose of this paper is to study the synchronization of nodes in distributed network. The control strategies of synchronization are studied and their control method is proposed, which provide a more widely applicable range for users. That means the synchronization control of inputs or outputs nodes can be done through the design of users' own application program. Based on some ideas of program parallel process and multi-CPU parallel executing, the conception of synchronization drive activating and deactivating are introduced which aiming at the problems of synchronous control in distributed network control system. By setting global monitor, renewing process states of nodes and judging synchronization conditions, the nodes are be activated or deactivated. By this way, user can realize same interval synchronous sampling inputs and drive outputs of nodes in a distributed network control system to satisfy higher demand for synchronization control in many cases.

This paper is organized into six sections including this introduction. Section 2 discusses synchronization environment and synchronization object. In section 3, the concepts of activating and deactivating to the program of network nodes are proposed. In order to solve the problem of isochronous data sampling or isochronous drive output, a novel kind of control idea of synchronization is proposed, which include dividing slave's main loop program into several processes and adopting twice trigger signals to activate synchronization control. In section 4, some analysis of the time characteristics of synchronous behaviour of nodes has been done. Section 5 makes a dissection to the realization method of synchronization control in PROFIBUS-DP master-slave network system, and discusses briefly their conditionality in actual application. Finally, conclusions are presented in section 6.

Ⅱ. SYNCHRONIZATION ENVIRONMENT AND OBJECTS

Distributed control systems are used commonly for situations in the following:

* This work is supported by grant No.Y2006G25 from the Natural Science Foundation of Shandong Province P.R. China to Fang He.

978-1-4244-2503-7/08/$20.00 © 2008 IEEE

1) When system consists of various parts that can be clearly delineated and can be controlled relatively autonomously, run the program for several separate system sections as network node.

2) When the user program is too large for one CPU and memory is used up, the control program should be distributed among more than one CPU network node.

3) If part of the system must be processed quickly, take these program sections out of the main program and run them on a separate fast CPU network node.

Synchronization environment means that in distributed control systems, the participating CPUs or the other I/O units need their operating modes synchronously. It means they start up together and go into stop mode together in some cases. Especially, the high time consistency is requested. It contains that data acquisition inputs of each node is done synchronously, and data outputs and control command of each node is executed synchronously. The user program for each CPU executes or the other I/O units are executed independently of the user programs in the CPU of node. This means control tasks of each network node are run in parallel. By carrying out sampling and process data simultaneously, execute simultaneously users programs by the distributed nodes, transmit simultaneously results to related terminals, the control system of high quality would be achieved.

In distributed control systems, the CPUs in a segmented node form their own independent subsystem and behave like single processors. There is no shared address area. The CPUs are interconnected via the communication bus. In the synchronization environment of distributed control systems, each node can be defined a synchronization object.

Ⅲ. SYNCHRONIZATION ACTIVATING AND DEACTIVATING TO THE DISTRIBUTED NODES

A. Activating and Deactivating

The implement of synchronization environment in distributed control systems is based on a global state monitoring system. By a global monitor and on calculating consistent global states, the distributed application monitoring system gather information of each slave-node. In distributed control systems, the master-slave network is a kind of familiar structure. The master synchronization node is responsible for monitoring and analyzing the state of a parallel application and issuing signals to processes. The signal can be activation or deactivation.

There are three kinds of processing to the node of synchronization object. Two of them, activating and deactivating, will determine the regions sensitive to synchronization signals. The third is the waiting, which suspends a process until a signal arrives. Suspending is necessary feature, because not always a process has something useful to do before a synchronization condition for a group of processes is met.

Synchronization activating means that makes the current main loop program to be suspended by the trigger signal of synchronization control and turn to respond executing program of synchronization application, which satisfies synchronizing conditions defined by programmer. These programs of synchronization application can be the program of synchronization data acquisition or data outputs, etc.

Synchronization deactivating means that when the program of synchronization inputs or outputs of active nodes ends, the signs of synchronous executing will be cancelled. The executions of program return to the interruption, where the main loop program is interrupted, and continue the main program.

B. Control Idea and Implementation

In the master-slave network system, the master node is used as global monitor to promulgate synchronization control command. Each slave-node that is requested to perform synchronization action is considered as a synchronizing object. In order to structure a synchronization control program, something must be done. Firstly, it needs to set up address tables of slave-nodes. Secondly, it needs to set up process state tables of slave-nodes. Thirdly, programmer should confirm the states of synchronous conditions in response state table. The main loop program of each slave-node must be divided into several parts, and every part is regarded as a process. The function of each process is respectively independence. In the start and end of each process, the process state table of node will be refreshed. It is necessary for programmer to confirm the conditions of synchronization judgement, as well as the response to these conditions.

In order to realize synchronizing action control of inputs and outputs in distributed slaves, we proposed the control idea of activating synchronizing response by sending twice trigger signals to slave-node. We will describe this kind of control idea through a synchronizing sampling process for distributed nodes in a master-slave structure system.

We assume there is a master-slave network system, which holds a master and m slaves. There are n slaves need implement synchronous data sampling among m slaves. The main loop program of each node has been divided into i_n processes.

Address state table of slave-nodes can be described as:

10001111000100……

The total number is m. The state “1” expresses the slave is requested to carry out synchronization sampling program and “0” is reverse.

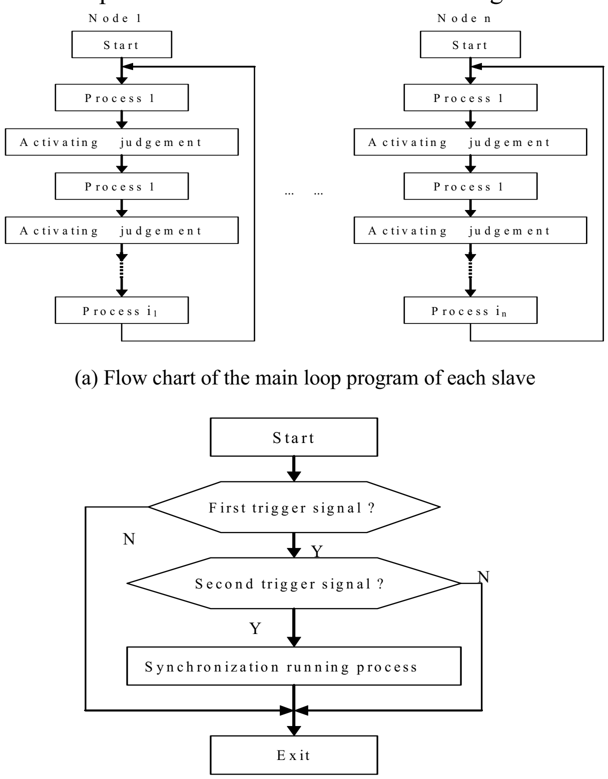
Process state table of single slave can be described as:

100111100110011……

The total number is i_n. The state “1” represents the process is running and “0” denotes the current process has finished. For each slave-node, the i_n can be different and the time of each process acting is usually different.

When the master sends the synchronization trigger signal of sampling to slaves, its global monitor collects the process states of slaves. In this time, some of slaves that have synchronization requests have not finished their current processes. The sign of the process state of slave is refreshed continuously. Until all of these slaves finished their current process, the sign of the process state of slaves become “1”. Immediately, the master sends its second trigger signal to slaves to make them executing synchronizing sampling program. All of sampling data are stored in process image area specified, and will be computed and transferred in next process cycle. Once the synchronous sampling is completed, the program will go back to main program. The master reset the process start signs and refresh process states table of slaves.

As an example, as shown in Fig. 1, we suppose that the program of each synchronization objects are separated into three parts: the main loop program of the slave-node, the monitoring and judging program of synchronization condition and the synchronizing running process program (input or out put). Several interruption points are set in the main loop program. These interruption points divide the main loop program into many processes. At the beginning of each process program segment, the sign of start is set. And at the end of segment, the sign of end is set. The sign of start is written into process states table of node according to number



(a) Flow chart of the main loop program of each slave

(b) Flow chart of the activating judgement program

Fig. 1 Flow chart of synchronous control of nodes

of process. At every interruption point, synchronization conditions judgement program is called.

If there are synchronization active signals that satisfy the synchronizing conditions at interruption points, the synchronizing objects can cut-in running sampling program synchronously through twice synchronization activating and judgement. This way guarantees to realize isochronous sampling in distributed network nodes.

When the synchronizing sampling program is running, the main program is suspended and the signs of process are stored. After the synchronizing sampling has been completed, the program of node returns back to the interruption points to continue running the main program.

Ⅳ DISCUSSION OF THE TIME CHARACTERISTIC OF SYNCHRONIZATION

A. Analysis to the Time of Node Course

The number of processes of synchronization objects is different in a distributed structure and their running time is distinct. If a process of a node is running, which means it is insensitive to synchronization request. When the master sends the first synchronization trigger signal of sampling, the synchronization objects, which have finished their processes, are suspended. They need wait for until the other nodes finish their processes in succession. Once the last synchronization object ends its process, the master will send the second synchronous trigger signal by judging the process states through process states table of each node. In this way, the synchronization objects can execute the sampling program simultaneously. The consistency of data can be guaranteed.

As an example, we suppose there are three synchronization object nodes, and their main loop program is divided into 8 parts (processes). The time characters of twice synchronization trigger by master are shown in Fig. 2. The variety of the process states tables of each node are described below.

When the master sends the first trigger signal, the process states tables of three nodes indication that the 4th process of node 1 and the 3rd process of node 3 are executing and the 3rd process of node 4 has finished.

Node 1：00001000

Node 2：00000000

Node 3：00000100

The master refreshes process states continually till the current process of each process is completed. Then the master sends the second trigger signal. Now, the process states of slaves will be:

Node 1：00000000

Node 2：00000000

Node 3：00000000

When the synchronization program has finished, the main loop program enters into next process. The process states of slaves are renewed by the following data.

Node 1：00010000

Node 2：00001000

Node 3：00001000

The more the number of processes made in main loop program, the more rapid the response of distributed slave-nodes to synchronization request will be. However, the process time of synchronizing response will be protracted due to increasing of synchronization monitoring and conditions judging.

B. Analysis to the Time of Processing Cycle

Sequence of cyclic program processing contains course in the following:

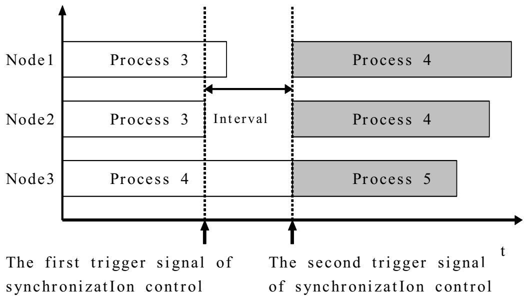


Fig. 2 The control strategy of twice synchronization activating signals

1) The operating system starts the cycle monitoring time.

2) The CPU reads the state of the inputs of the input modules and updates the process image table of the inputs or writes the values from the process image table of the outputs to the output modules.

3) The CPU processes the user program and executes the instructions contained in the program.

4) The CPU writes the values from the process image table of the outputs to the output modules or reads the state of the inputs of the input modules and updates the process image table of the inputs.

5) At the end of a cycle, the operating system executes any tasks that are pending, for example downloading and deleting blocks, receiving and sending global data.

6) Finally, the CPU returns to the start of the cycle and restarts the cycle monitoring time.

Majority of process cycles in a distributed network are not synchronous. So the prior matter done to realize synchronizing control is their timing base on a reliable point.

Users' program of slave can synchronize with system cycle by synchronizing cycle interruption, making synchronization when slaves read inputs, that is, they read the input at one time. Similarly, introducing of the same strategy of synchronization cycle interruption satisfies synchronization of output of slaves with system cycle, that is, all slave-nodes carry out sending at one time.

All nodes requested execute control synchronously. And the consistency of timing of inputs and outputs is fulfilled. Three basic characters underlie this model:

1) The synchronization between user program and I/O process, that is, regular all operations at identical time basis. All inputs should be registered in a limitation defined previously. The similar conception adapt to outputs too. In this way, the whole process is synchronous course.

2) The sampling occurring in one cycle will be always processed in next period, and at the end of third cycle execution will return to the sampling cycle. That means that a node read the inputs during the (n-1)th cycle, and then in the nth period the node compute the sampling and transferring the data, finally the slave transmits the result to terminals in the (n+1)th period. The Fig. 3 shows the process.

3) Consistency of timing between all distributed nodes is requested when they execute transmitting. That means the inputs and outputs of distributed nodes are related to each other and based on the same time base.

Process responding time can be described in the following:

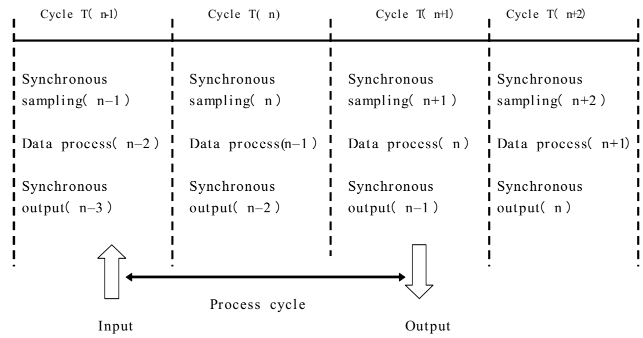


Fig. 3 Data processing cycle of input and output

up from "$Ti + T + To$" to "$Ti + (2\times T) + To$". *Ti* denotes the time of input. *To* means the time of output and *T* is cycle of system. In order to guarantee synchronizing states of inputs of distributed nodes before new cycle of system begin, reading inputs must be done previously. *Ti* consists of the time of signal preparing and conversion in electronic modules and the consumption of time in transmitting of message to slave-nodes along bus around nodes backboard. *To* can satisfies process response to make the consistency of transmitting to system I/O terminals. *To* comprise the time that is spent in transmitting between all slaves and master and the consumption of time in transmitting of message to I/O terminals along bus around nodes board. The process time from "$Ti + T + To$" to "$Ti + (2 \times T) + To$" depends on the interval of detecting between inputs and outputs in electronic modules.

V. THE REALIZATION OF SYNCHRONIZATION CONTROL BASED ON MASTER-SLAVE STRUCTURE OF PROFIBUS NETWORK

A. The master-slave structure and the process image accessing

PROFIBUS is considered as one of the fieldbuses solutions of the General-Purpose Fieldbus Communication System according with European Standard EN 50170. The MAC mechanism of PROFIBUS is based on a token passing to be used for data communication between master and their slaves. A simplified TT protocol is used for PROFIBUS token passing procedure. PROFIBUS-DP is most widely used in PROFIBUS network systems. The protocols of DP are applied mainly to data transmission in a high speed for PLC, distributed I/O or the other field equipments. The typical configuration of DP is single mater with multi-slave or multi-maser with multi-slave. A DP master links the CPU manager with distributed I/O systems. It exchanges data with the distributed I/O systems via bus, and monitors the running state of PROFIBUS DP bus. PROFIBUS DP supports all DP master or DP slave, which are compatible to IEC61784-1: 2002 Ed1 CP 3/1 standard.

When users' program access to address sections of inputs and outputs, the states of signal of digital signal modules are not accessed, conversely, the regions which it access to in memory is process image in memory. Process image can be divided into two sections: input process image and output process image.

Compared with I/O, advantage of accessing to process image is consistent process image signals can be provided for CPU during running of cycle. In the process of executing of program, when state of signal in an inputs' module change, the signal state in process image remains till next period. In addition, the rate of accessing is obviously fast than directly accessing to signal modules due to the process image stored in CPU. The Fig. 4 shows the sequence of operation in one cycle.

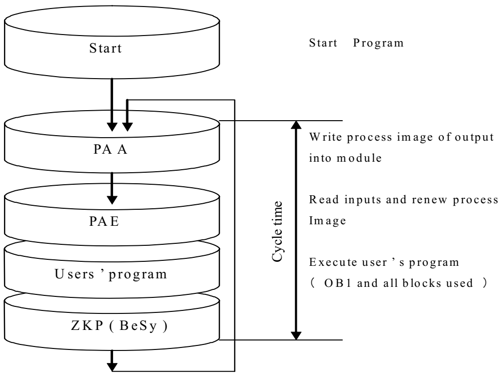


Fig. 4 The sequence of operation in one cycle

B. Synchronization control of output and input

1) Control of output

The writing of outputs of DP modules is triggered as follows:

- By transfer commands to the DP I/Os;
- By writing the process image output table to the modules (by the operating system at the end of OB1 or by calling SFC 27 "UPDAT_PO");
- By calling SFC 15 "DPWR_DAT." In normal operation, the DP master transfers the output bytes cyclically (within the cycle of the PROFIBUS DP bus) to the outputs of the DP slave-nodes;
- If certain output data are requested (possibly distributed on several slaves) applied to the outputs to the process at exactly the same time, the SYNC command can be sent to the relevant DP master using SFC 11 "DPSYC_FR".

SYNC control command of a DP master may broadcast to a group of DP slaves. Once master send SYNC control commands, the slave-nodes are active to make outputs synchronized. With the SYNC control command, the DP slaves of the selected groups are switched to the Sync mode. In other words, the DP master transfers the current output data and instructs the DP slaves involved to freeze their outputs. The DP slave stores the output data contained in the next frame. With the following output frames, the DP slaves enter the output data in an internal buffer and the state of the outputs remains unchanged.

Following each SYNC control command, the DP slaves of the selected groups apply the output data of their internal buffer to the outputs to the process. The outputs are only updated cyclically again when the UNSYNC control command is sent using SFC 11 "DPSYC_FR." The command "UNSYNC" means the cancellation of synchronizing outputs of nodes.

2) Control of input

The input data of the DP modules are read as follows:

- Using load commands to the DP I/Os;
- When the process image input table is updated (by the operating system at the start of OB1 or by calling SFC 26 "UPDAT_PI");
- By calling SFC 14 "DPRD_DAT." In normal operation, the DP master receives this input data cyclically (within the cycle of the PROFIBUS DP bus) from its DP slaves and makes them available to the CPU;
- If certain input data are requested (possibly distributed on several slaves) to obtain from the process at exactly the same time, send the FREEZE control command to the relevant DP master using SFC 11 "DPSYC_FR".

FREEZE control command of a DP master may broadcast to a group of DP slaves. The synchronization inputs of slave-nodes will be active, as the command FREEZE has been received. With the FREEZE control command, the DP slaves involved are switched to the mode of freezing, the slave freezes its current input status, and then transfers the frozen data to the input area of the CPU. The DP slave freezes its input status again after each new FREEZE command.

The DP slave does not resume the transfer input data to the DP master until the DP master has sent the UNFREEZE control command with SFC 11 "DPSYC_FR". The command "UNFREEZE" means the cancellation of synchronizing inputs of nodes.

3) Data Consistency

Before the control commands listed above are sent, it must be done to assign the DP slaves to groups using STEP 7. It must be confirmed which DP slave is assigned to which group with which number and the reactions of the various groups to SYNC/FREEZE.

In order to guarantee data consistency, something should be taken into account. Because SFC 11 "DPSYC_FR" functions are a cyclic and can be interrupted by higher priority classes, it should be made sure that the process images are consistent with the actual inputs and outputs when using SFC 11 "DPSYC_FR".

Data consistency is guaranteed if the following consistency rules are done:

- Define suitable process image sections for the "SYNC outputs" and the "FREEZE inputs" (only possible on the S7-400). Call SFC 27 "UPDAT_PO" immediately before the first call for a SYNC job. Call SFC 26 "UPDAT_PI" immediately after the last call for a FREEZE job.
- As an alternative: Use only direct I/O access for outputs involved in a SYNC job and for inputs involved in a FREEZE job. Must not write to these outputs when a SYNC job is active and not read these inputs when a FREEZE job is active.
- If using SFC 15 "DPWR_DAT," this SFC must be completed before a SYNC job to be sent the outputs involved. If using SFC 14 "DPRD_DAT," this SFC must be completed before a FREEZE job to be sent the inputs involved.

4) Conditionality

The basic function of PROFIBUS-DP（DP-V0） supports the function of synchronization and freezing, which can guarantee the realization of synchronizing control for inputs and outputs of distributed slaves. The extension function (DP-V2) of PROFIBUS-DP supports isochronous mode. Isochronous mode means equal interval synchronous mode, which can realize synchronization between master and its slave with error less than 1ms. All relevant equipments can synchronize with master cycle periodically through broadcasting messages of “global control”.

According to different demands, the configurations of PROFIBUS-DP network systems are different in actual application. So it is not always appropriate to adopt those methods introduced above to any case. In other word, the application of the synchronization function offered by the manufacturer is limited by their hardware and software. Software is required to be STEP 7 or higher versions Hardware is required to satisfy some conditions in synchronous mode. They include:

- CPU can support synchronizing mode;
- DP master supports invariable bus cycle;
- The interface of slave is required to support synchronizing mode.

Under condition of higher rank configurations of hardware and software, the synchronizing function offered by the manufacturer can make control easier for users. However, it is bounded the extension in use because of the limitation of hardware and software and encapsulation of system software by the manufacturer.

The strategy of synchronous activating and deactivating proposed by his paper in section Ⅲ gives out a more wide application for users in cases of synchronization control. Even if under a lower level configurations of hardware, user can realize synchronization control for distributed nodes in virtue of this strategy by program.

Ⅵ. CONCLUSION

In order to realize isochronous sampling inputs and isochronous drive outputs of nodes in the distributed network, this paper proposes a novel kind of synchronization control idea and gives its implementation. This kind of method can fetch up the limitation that the implement of synchronization control has to rely to hardware and system software offered by manufacturer. By compiling application program according to this kind of idea, high isochronous control demand can be satisfied for distributed nodes in many cases. This kind of synchronization control idea mainly includes several points below.

1) In the distributed control system, every node can be regarded a synchronization object in the synchronization environment.

2) The concepts of activating and deactivating to the drive program of network nodes are set. A global monitor is used. By updating the table of process state and adjusting the synchronization control condition, activating or deactivating to the program of network nodes is realized.

3) The master must send twice trigger signals for once synchronization action (input or output) in order to ensure data consistency of distributed nodes.

4) The structure of program is technically designed. The main loop program of node is divided into several processes, and a judging program of synchronization condition is cut in.

ACKNOWLEDGMENT

This work was supported by grant No.Y2006G25 from the Natural Science Foundation of Shandong Province P.R. China.

REFERENCES

[1] S. Vitturi, “On the effects of the cyclic traffic on Profibus DP networks,” *Computer Standards and Interfaces*, vol. 26, no. 2, 2004, pp. 131-144.

[2] Wu Xiaopeng, Xu Baoguo, “Control of multi-motor synchronization based on fieildbus,” *Computer Engineer and Design*, vol. 27, no. 3, 2006 pp. 367-269.

[3] Fang He, Weiming Tong, Qiang Wang, “Synchronization Control Strategy of Multi-motor System Based on Profibus Network,” *IEEE International Conference on Automation and Logistics*, pp. 3029-3034, 2007.

[4] Liu Huikang, Yang Yongli, Yang Xianzhao, Cheng Gengguo, “An approach for torque follow-up control system using model reference adaptive control based on neural network algorithm”, *Electric Drive*, vol.35, no. 4, 2005, pp. 27-29.

[51] Hou Weiyan, Zhang Haifeng, Fei Minrui, “Analyzing of Real Time Performance in Fieldbus Profibus,” *Journal of Electronic Measurement and Instrument*, vol. 18, no. 2, 2004, pp. 56-59.

[6] Shang Lina, Zhang Rongbiao, “Survey on Research of Time Delay Network Control System,” *Industrial and Control Computer*, vol. 20, no. 5, 2007, pp. 1-3.

[7] S. Vitturi, “Some features of two fieldbuses of the IEC 61158 standard,” *Computer Standards and Interfaces*, vol. 22, no. 3, 2000, pp. 203-215.