International Journal of u- and e- Service, Science and Technology
Vol.9, No. 6 (2016), pp.201-210
http://dx.doi.org/10.14257/ijunesst.2016.9.6.19

Rapid Development for Process Control Systems Based on PC/104

An Bao-Ran[1], Liu Guo-Ping[2] and Chai Sen-Chun[3]

[1]. Institute of Computer Applications, China Academy of Engineering Physics, Mianyang, 621900
[2]. School of Engineering, University of South Wales, Pontypridd CF37 1DL, United Kingdom
[3]. School of Automation, Beijing Institute of Technology, Beijing 100081, China
baoranan@126.com[1], guoping.liu@southwales.ac.uk[2], chaisc97@hotmail.com[3]

Abstract

The paper investigates a rapid-prototyping methodology for the design of process control systems using PC/104 and MATLAB/RTW. Experimental results obtained from the real-time control of the torque motor speed control system are presented.

***Keywords**: PC/104, Process Controller, MATLAB/RTW*

1. Introduction

Over the past several decades, control theory has attracted more and more attention because of its applications in various areas such as digital weapon systems [1], missile guidance [2], wireless networked control systems [3], complex industrial processes [4], precision agriculture systems [5], and mining and mineral engineering [6]. The rapid development of microprocessor technology has offered very powerful hardware architectures and reliable solutions for implementation of different automation systems, such as MCS-51 [7], PIC [8], ARM [9], DSP [10], Field Programmable Gate Array (FPGA) [11], PC/104 [12, 13], Programmable Logic Controller(PLC) [14], *etc.* In general, the implementation of a control system has to pass six phases [15], including requirement analysis, modelling, control design, simulation, implementation, and verification. However, researchers face various difficulties especially during the implementation phase, which is mainly due to their lack of mature design and development experiences. It has seriously affected engineering development progress and system flexibility. The increasing demand for rapid development and flexibility in process control systems prompt researchers to explore convenient modelling and implementation tools [16-17]. As is well known, MATLAB is a software environment that allows a user to easily integrate computation, algorithm development, modeling, simulation, data analysis, visualization, and application development. SIMULINK, a software package for modeling, simulating, and analyzing dynamic systems in the MATLAB environment, supports both linear and nonlinear systems that are modeled as transfer functions or state-space equations in continuous time and discrete time. Real-Time Workshop (RTW) toolbox can generate C code directly from the SIMULINK models and executable codes is created for various platforms by the secondary development [18]. At present, dSPACE provides a multiprocessor DSP-based solution, while this system is too expensive to expand many users' experimental program [19]. A low-cost rapid-prototyping system using Texas Instruments' (TI) TMS320C30 Evaluation Module (EVM) is constructed in [20], which includes compiling, assembling, and downloading of the real-time algorithms. For laboratory automation and measurement, a LabVIEW application for the NI PXI system through rapid-prototyping techniques has been developed in [21]. Taking into account hard real-time capability and rapid implementation, LinuxRTAI and

ISSN: 2005-4246 IJUNESST

MATLAB/RTW is employed to develop Rapid Controller Prototyping (RCP) in [22] and [23].

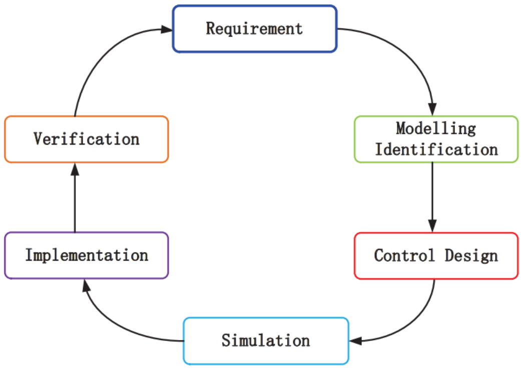

Figure 1. Control System Design Phases

In general, controlling some slow dynamic processes, such as temperature, pressure and level, need not meet the strict real-time requirement [24]. In addition, a process controller should achieve remote monitoring and adjustment of plants over a communication network. Therefore, an embedded computer (*e.g.*, PC/104) running a non-real-time operating system (*e.g.*, Windows XP) can execute control programs under acceptable sampling period.

This paper aims to investigate a rapid-prototyping methodology for the design of process control systems and to develop a PC/104-based process control system involving a torque motor as a test bed for demonstrating our design methodology. The rest of this paper is organized as follows. Section 2 describes the overall architecture of PC/104-based process control system for controller side and PC side, including its hardware structure and software development. Experimental results obtained from the torque motor speed control system are given in Section 3. Section 4 provides the conclusions.

2. System Architecture

The general scheme of the system architecture is shown in Figure 2.

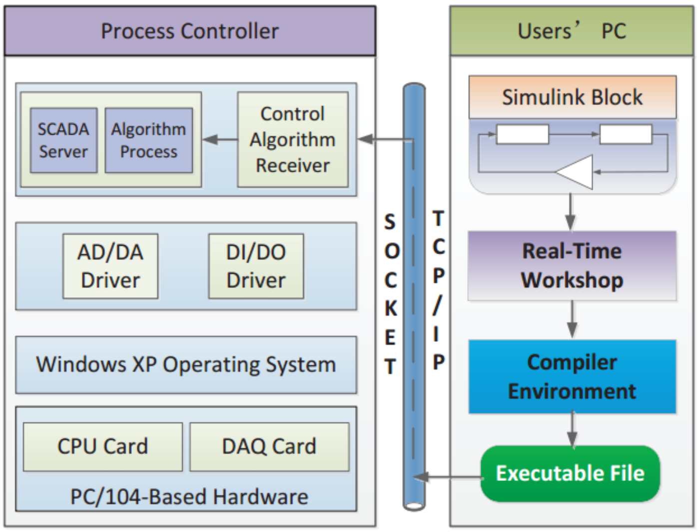

Figure 2. Control System Design Phases

In this diagram, some hardware and software elements are designed in two main parts linked by Ethernet: PC side where system modelling, simulation and implementation are carried out using MATLAB/Simulink blocks and RTW generates ANSI C codes and executable codes for the target platform (the process controller), and the plant side where the physical process system and PC/104-based control elements are located.

The following subsections explain these components in more detail.

2.1. PC/104-based Controller Platform

With the advances in reliability, flexibility, antiinterference and lower costs, a PC/104-based implementation of the modular controllers, which has been used successfully for the detector head interface electronics [25], unmanned rotorcraft vehicles [26], environmental remote sensing [27], *etc*, is considered. The control algorithm runs on PC/104-based Controller which drives physical plants.

2.1.1. Hardware Configuration: PC/104 is an industrial control bus that offers full architecture, hardware and software compatibility with the PC bus, but in ultra-compact and stackable modules. Thus, it is ideally suited to the unique requirements of embedded control applications [28]. Two cards utilized in the Windows XP system incorporate the PC/104 bus.

A full function industrial PC board '104-1647CLD2N' manufactured by EVOC Company, which is with embedded PC/104 structure and low power consumption, was chosen (see Figure 3).

Figure 3. Component Layout Diagram of 104-1647CLD2N

The highly integrated CPU board is equipped with

- AMD LX800 (standard configuration)/ 700 500MHz CPU;
- On board 256MB DDR memory;
- Two USB2.0, BIOS support USB start-up;
- One CF card interface;
- Four serial ports, two of which are RS-232 mode, one is RS-232/RS-422/RS-485 optional mode and the other one is RS-232/RS-485 optional mode;
- Two 10/100Mbps Ethernet controllers.

A PC/104 data acquisition card 'ART2953' manufactured by ART Company [29] is added to the stack for data acquisition and analog output (see Figure 4).

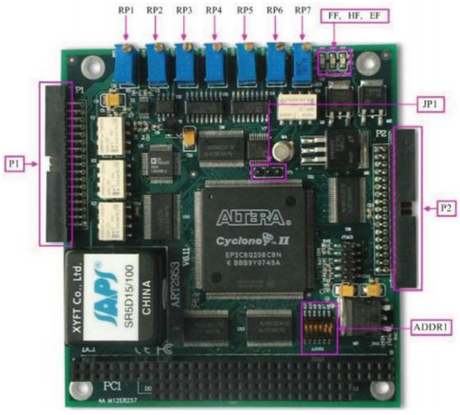

Figure 4. Component Layout Diagram of ART2953

It is equipped with the following functions:

- a 16-bit A/D convertor with 250kS/s multifunctional data acquisition;
- 16-bit 16-channel single ended or 8-channel differential analog input;
- 12-bit 4-channel analog output;
- 8-channel digital input and 8-channel digital output;
- 16-bit 3-channel timer/counter.

2.1.2. Software Configuration: One Windows-based PC/104 system is applicable in general to process control. It could integrate existed network communication services, necessary batch scripts and device drivers. The development and configuration for the PC/104-based Controller Platform is given below.

- ART2953 card driver: It includes a set of the device-specific codes such as AD, DA, DI, DO and Timer/Counter, which is needed to carry out actions on the device;
- One receive file server ‘recvnfile.exe’: It is responsible for receiving executable files and creating a new process that runs the received executable file;
- Batch scripts: They have been developed in the powerful shell and can automatically start control security related services such as recvnfile.exe, telnet, http, *etc.*

2.2. Design Framework of PC HOST System

In order to guarantee flexibility and efficiency of the control solution, the design framework of PC HOST System is presented for a successful software configuration and development. Figure 5 describes the process to create, compile, execute and monitor a control system block diagram. Below is a list of how to make full use of Simulink, RTW and Microsoft Visual Studio.

➢ Process Control Library for Simulink

From the viewpoint of the computer control technology, some blocks in Simulink library are not available for the implementation of a practical process control system. To access the physical peripherals, several general purpose blocks, such as analog input, analog output, digital input, digital output and timer-counter, are developed using the S-function. Those customized S-function blocks are added to Simulink as a process control library in order that users can use them directly.

➢ Configuration of Template Files

In order to automate code generation for the PC/104-based controller platform, one target script named ‘*grtVSCAPI.tlc*’ is developed, which provides users with the flexibility to customize the application-specific C-coded files generated by RTW. Meanwhile, the template makefile named ‘*grtVSCAPI.tmf*’ which is used to create a set of macro definitions and rules describing how to produce the executable file from the C-coded files (composed of .c, .h) when program building is invoked. As to make utility, it is required to install Microsoft Visual Studio and use the compile command ‘*nmake.exe*’. Below describes the main configuration flow that includes specific solutions upgrade and automatic downloading of executable files in ‘*grtVSCAPI.tmf*’.

- SCADA Software

The supervisory control and data acquisition (SCADA) software through Ethernet can be divided into two parts: one is the server side on PC/104 controllers, the other is the client side on the PC.

```
$(MODEL).exe: $(MODEL).mak
devenv.exe /Upgrade $(MODEL).mak
devenv.exe /Build Debug $(MODEL).vcproj
cd Debug
$(SENDFILE) $(Netcon WinPC104 IP) 17728 $(MODEL).exe
cd ..
                              cd ..
echo ### $(MODEL).exe: $(MODEL).mak
```

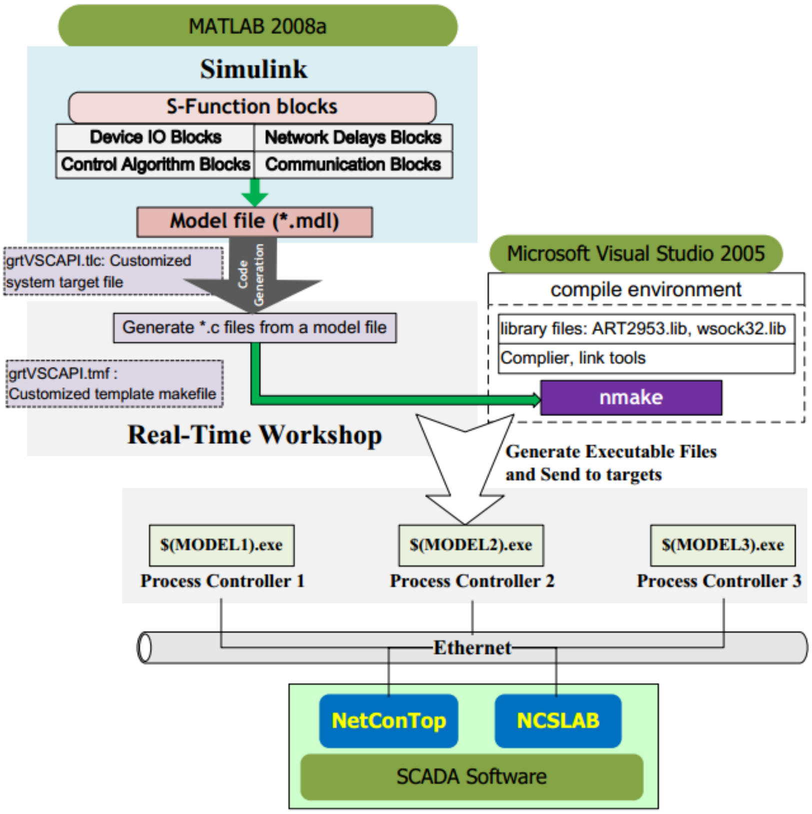

Figure 5. System Design Framework of PC HOST

In our previous work, there are two software projects on the client side: NetconTop [30] developed in Visual C++ programming language and NCSLAB [31] developed in the Java programming language, which can establish the communication with the server through the TCP/IP protocol and enable users to send commands, adjust parameters and display the corresponding variables.

The server program on PC/104 controllers, which is developed in C language, mainly responsible for dealing with the commands from the client side, sending the real-time data of all selected signals and parameters or changing the specified parameters of the control algorithm on-line.

3. A Case Study: Torque Motor Speed Control Experiment

Figure 6 illustrates a PC/104-based experimental system for torque motor speed control. It mainly consists of a torque motor, a signal converter unit for actuator/sensor, and PC/104-based controller. The THBLD-1 torque motor and signal converter unit manufactured by ZHEJIANG SUPCON Group Co., Ltd. are selected. Our PC/104-based controller serves the speed control experiment to obtain the speed data of the THBLD-1 torque motor and produce an appropriate control effort signal for driving the experimental system at each sampling step.

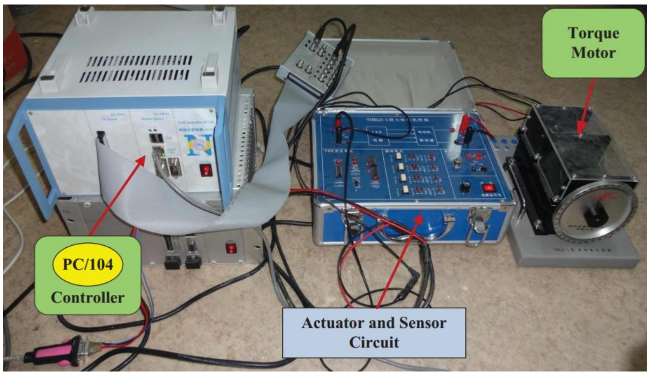

Figure 6. DC Torque Motor System

The first order plus dead-time (FOPDT) model is used to approximate the dynamics of the torque motor speed control process. Based on a closed-loop identification method of Proportional-Integral(PI) control that is proposed in [32], a transfer function model for the torque motor speed control process can be obtained as

$$G_p(s) = \frac{3.392}{s + 0.442} e^{-0.08s} \qquad (1)$$

The identification result, as shown in Figure 7, indicates the practical out is very similar to the model-based output of the same proportion and integration control parameters in the closed-loop control experiment. Using the normal PI design method, the transfer function of the PI controller is given as

$$G_c(s) = \frac{0.25s + 0.15}{s} \qquad (2)$$

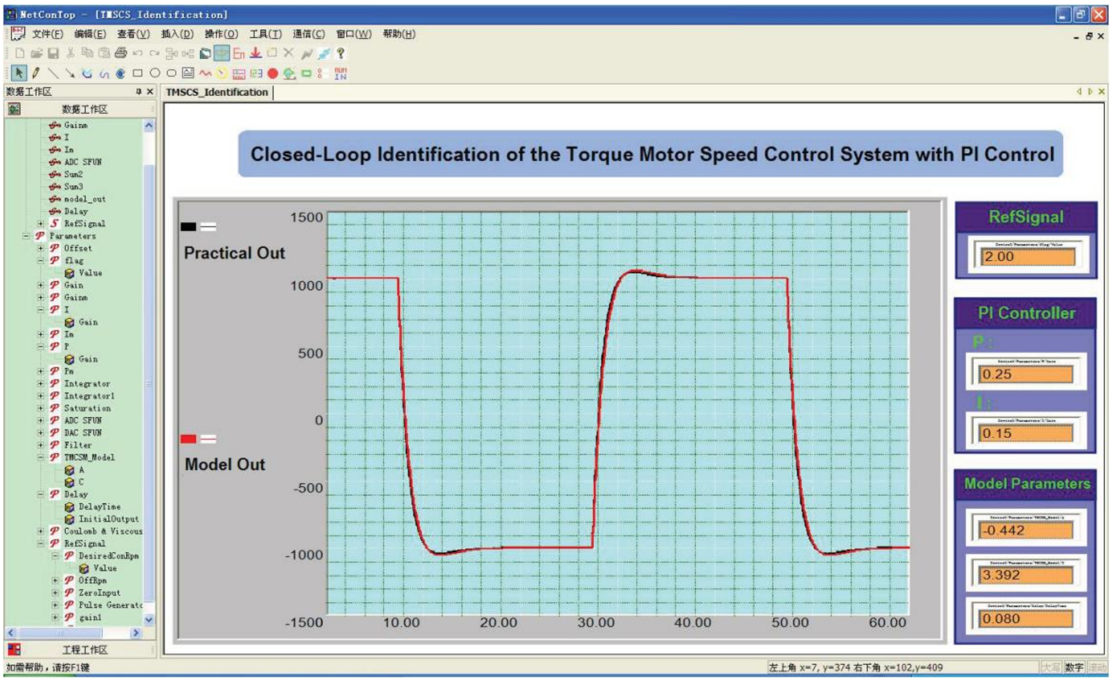

Figure 7. Comparison of the Practical Out of the Torque Motor Speed System and the Closed-loop Identification

To implement the speed control, a control diagram with the above PI controller in Figure 8 is designed in Simulink. The RTW is employed to generate the executable code, and then the executable code is uploaded to our PC/104-based controller via Ethernet. The response of the closed-loop PI torque motor speed control system is given in Figure 9. Clearly, it has good control performance.

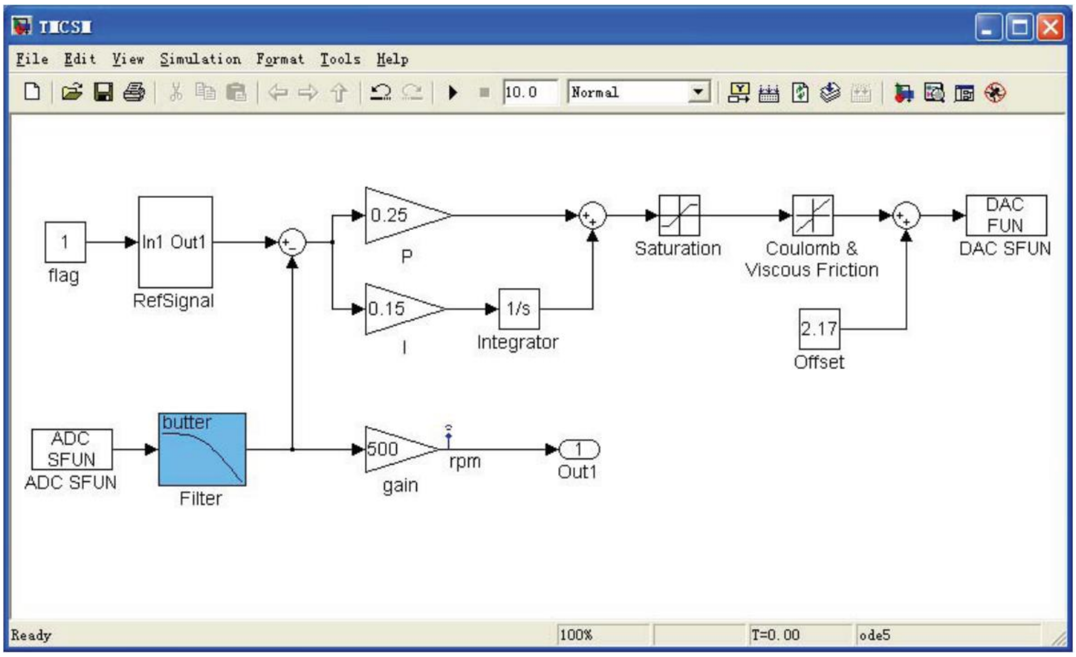

Figure 8. Control Block Diagram in Simulink

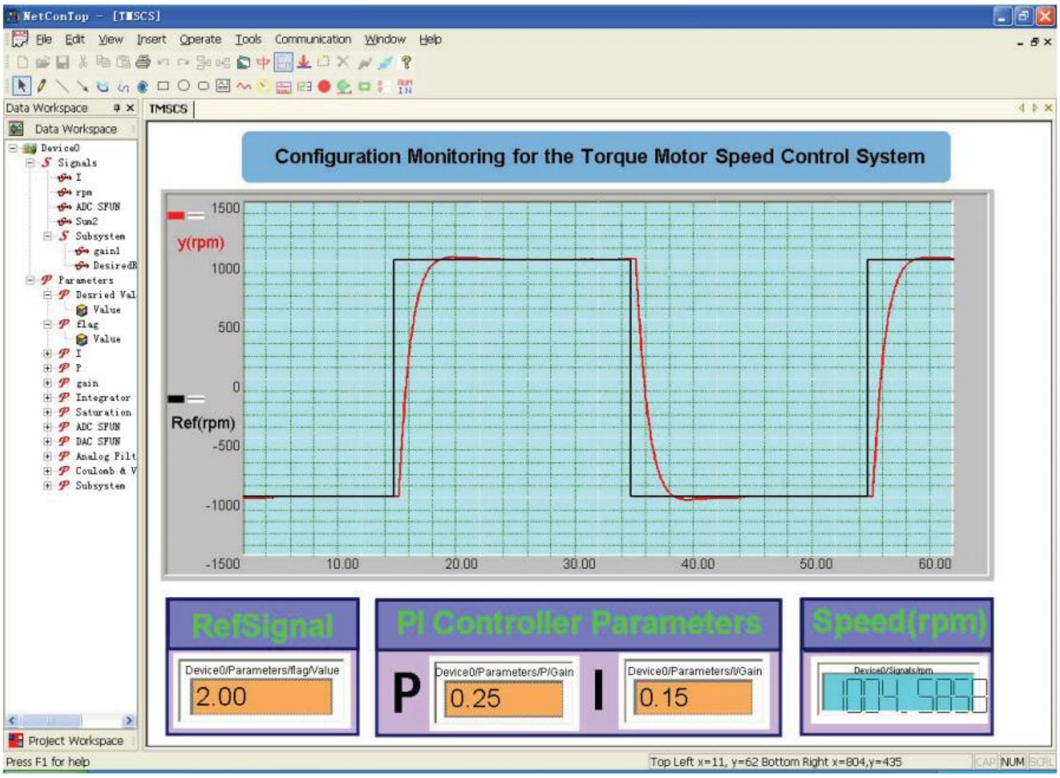

Figure 9. Monitoring Interface Based on NetConTop

4. Conclusions

In order to satisfy the requirement of fast development and implementation for process control systems, this paper takes Matlab/RTW and Visual Studio as software combination, and it realizes the modularization and configuration of the whole process control procedure. The system architecture is described and the hardware setting and software design are detailed. At last, an application into the torque motor speed control system is achieved.

References

[1] J. Chen, H. Fang, B. Xin and F. Deng, "Zidonghua Xuebao/Acta Automatica Sinica", vol. 39, no. 7, **(2013)**
[2] Z. Zhu, D. Xu, J. Liu and Y. Xia, "IEEE Trans. Ind. Informat", vol. 60, no. 12, **(2013)**.
[3] T. Kohtamaki, M. Pohjola, J. Brand and L. M. Eriksson, "PiccSIM toolchain - Design, simulation and automatic implementation of wireless networked control systems", Proceedings of the 2009 IEEE International Conference on Networking, Sensing and Control, Okayama, Japan, **(2009)** March 26-29.
[4] T. Chai, L. Zhao, J. Qiu, F. Liu and J. Fan, "IEEE Trans. Ind. Informat", vol. 9, no. 1, **(2013)**.
[5] I. M. Santos, F. G. da Costa, C. E. Cugnasca, F. Liu and J. Ueyama," Precis. Agric.", vol. 15, no. 3, **(2014)**.
[6] A. A. Uppal, A. I. Bhatti, E. Aamir, R. Samar and S. A. Khan, "Journal Process Cont.", vol. 15, no. 3, **(2014)**.
[7] T. C. Chiang, C. Y. Chang and Y. W. Wang, "J. Vib. Control", vol. 17, no. 3, **(2011)**.
[8] K. Khairurrijal, M. Abdullah and M. Budiman, "Comput. Appl.", Eng. Educ., vol. 19, no. 1, **(2011)**.
[9] P. Rong, J. Sun, W. Dong and L. Yu, "Journal of Harbin University of Science and Technology", vol. 18, no. 125, (**2013**).
[10] W. Zhang and Y. Zhang, "Journal of Harbin University of Science and Technology", vol. 10, no. 19, **(2005)**.
[11] R. F. Fung, M. H. Weng and Y. S. Kung, "Mech. Syst. Signal Pr.", vol. 23, no. 8, **(2009)**.

[12] B. An and G. P. Liu, “Real-time experimental platform of networked control systems based on PC/104 controller and web laboratory”, Proceedings of 2013 Chinese Automation Congress, Changsha, China**, (2013)** November 7-8.
[13] C. E. Hall Jr., “A Real-Time Linux system for autonomous navigation and flight attitude control of an uninhabited aerial vehicle”, Proceedings of AIAA/IEEE Digital Avionics Systems Conference, 1A11-1A19, **(2001)**.
[14] R. Marques, J. Rocha, S. Rafael and J. F. Martins, “IEEE Trans. Ind. Electron.”, vol. 55, no. 6, **(2008)**.
[15] R. Bucher and S. Balemi, Control Eng. Pract., vol. 14, no. 2, **(2006)**.
[16] C. H. Yang, V. Vyatkin and C. Pang, “IEEE Trans. on Syst. Man Cy-S.”, vol. 44, no. 3, **(2014)**.
[17] G. Doukas and K. Thramboulidis, “IEEE Trans. on Ind. Electron.”, vol. 58, no. 3, **(2011)**.
[18] S. Wang, G. P. Liu, R. David and M. Tan, “Development of networked robot fish control systems”, Proceedings of the World Congress on Intelligent Control and Automation, **(2006)**, pp. 9317-9321**.**
[19] D. C. Xie, Z. W. Wang, J. T. Cheng and Q. H. Zeng, “Journal of Astronautics”, vol. 31,no. 11, **(2010)**.
[20] W. S. Gan, Y. K. Chong, W. Gong and W. T. Tan, “IEEE Trans. on Edu.”, vol. 43, no. 1, **(2000)**.
[21] M. Canale, V. Cerone, V. Razza and D. Regruto, “Rapid prototyping of predictive controllers through an industrial platform”, Proceedings of American Control Conference, **(2012)**, pp. 4715-4720.
[22] R. Bucher and S. Balemi, “RScilab/Scicos and Linux RTAI - A unified approach”, Proceedings of the IEEE International Conference on Control Applications, **(2005)**, pp. 1121-1126**.**
[23] B. An, G. Liu and S. Chai, “Design and implementation of realtime control system using RTAI and Matlab/RTW”, Proceedings of the 2012 UKACC International Conference on Control, **(2012)**, pp. 834-839.
[24] Y. Wang, Y. Yang and H. Wu, “Yi Qi Yi Biao Xue Bao/Chinese Journal of Scientific Instrument”, vol. 28, no. 12, **(2007)**.
[25] J. W. Young, J. C. Moyers and M. Lenox, “IEEE Trans. on Nucl. Sci.”, vol. 47, no. 4, **(2000)**.
[26] G. Cai, B. M. Chen and T. H. Lee, “Frontiers of Electrical and Electronic Engineering in China”, vol. 5, no. 1, **(2010)**.
[27] V. V. Klemas, “Journal of Coastal Res.”, vol. 31, no. 5, **(2015)**.
[28] R. Lehrbaum, “Using PC/104 embedded-PC modules for embedded applications”, In WESCON/’93. Conference Record, IEEE,**(1993)**, pp. 633-638.
[29] http://art-control.com/Product/T17/769.html.
[30] Y. Zhu, G. Zheng and G. P. Liu, “System architecture design of supervisory software for networked control systems”, Proceedings of the 2007 Chinese Control Conference, **(2007)**, pp. 555-560.
[31] B. An, G. P. Liu and S. Chai, “Real-time experimental platform of networked control systems based on PC/104 controller and web laboratory”, Proceedings of the 2013 Chinese Automation Congress, (2012), (**2013),** pp. 749-754.
[32] R. Mamat and P. J. Fleming, “Electron. Lett.”, vol. 31, no. 15, **(1995)**.