International Journal of Control and Automation
Vol.7, No.1 (2014), pp.339-346
http://dx.doi.org/10.14257/ijca.2014.7.1.30

Real-time Simulation Experiment Platform for Braking Pressure of Vehicle Based on xPC Target

Shan Yi[1], Li Wenjuan[2,*] and Wei Jingwei[2]

[1]Electrical & Electronic Teaching and Training Center
[2]School of Electrical and Electronic Engineering
Harbin University of Science &Technology, Harbin, China

liwenjuan@hrbust.edu.cn

Abstract

For the issue of being difficult to carry out various experiments of vehicle braking systems, a real-time simulation platform which is an vehicle braking pressure based on xPC Target is proposed. In the design of platform, the method is studied through writing CS-function driver development. Under MATLAB RTW, the problems of hardware simulation in the loop to support I / O device is resolved in xPC Target toolbox. Upon the analysis of the basic principles and structure of the driver, the solution to the key issues such as register operation and base seeking address are studied. And PCI-1710 driver is developed. The verifications about the driver validation of the data acquisition card and closed-loop regulation to the vehicle braking pressure are completed with the platform. The experimental results show that the A / D sampling frequency of acquisition card can be up to 10 kHz, signals can be displayed in real time and the test platform can be long-running. The development process is stable. Pressure waveforms consistent with the waveforms measured by the oscilloscope. The braking pressure can be adjusted by the real-time simulation platform. A new experimental method is provided for research on the vehicle braking performance.

Keywords: *xPC Target; real-time simulation; driver; braking system*

1. Introduction

The experimental platform is set up based on the real-time technology. The real-time simulation technology is that parts of the actual objects which are connected to the simulation model in computer through the acquisition card [1, 2]. The entire simulation system is simulated under the control of the real-time clock. When the braking condition of vehicles is complex, it is difficult to reflect the actual situation by the software simulation. High cost, long cycle and risk of large experiments are appeared in the test of vehicle roads [3]. There are many benefits by building the real-time simulation platform of braking pressure. Firstly, braking conditions are well reflected. In addition, development costs, development cycle and the risk of road experiments are reduced. The main method to establish real-time simulation platform is to download code to the real-time system by the real working space (RTW) of Matlab/Simulink [4].

There are mainly three kinds of typical real-time systems that include dSPACE, xPCTarget and VxWorks. dSPACE is a real experimental platform based on Matlab / Simulink for the control system which combines development software and semi-physical simulation experiment platform. dSPACE is a commercial real-time simulation platform, which has good real-time performances, high reliability, strong expandability and many I/O supports, but its

ISSN: 2005-4297 IJCA

price is expensive. xPC Target is a typical dual PC real-time simulation experiment platform , which has a host running Matlab/Simulink and the target machine with real-time core. The sampling frequency of this real-time system can be up to 100 kHz, and it is easy to build, economy and practical. xPC Target supports large numbers of I/O ports which are mostly produced by foreign manufacturers. Therefore hardware I/O drivers are required to be written with S function to build an xPC Target real-time system. VxWorks uses the RTW embedded code to generate Simulink model as ERT target, through Tornado real-time code is downloaded to the embedded real-time operating system VxWorks [5, 6]. This method can improve the VxWorks real-time performance and easily adjust the parameters of the model to achieve adjustment parameter online.

In this paper the real-time simulation platform for braking pressure of vehicle based on xPC Target is designed, the data acquisition card PCI-1710 driver is developed, and the experiments of pressure closed-loop real-time simulation is made.

2. xPC Target Real-time Simulation Experiment Platform

The real-time simulation experiment platform consists of xPC Target real-time simulation system, signal processing system and external hardware devices, such as solenoid valve, pressure transducer, air pump, air cyclinder and so on. Its overall structure is illustrated in Figure 1. The real time system is achieved by xPC Target toolboxes and Matlab/Simulink. Signal processing system converts and isolates the signals collected by the data acquisition card.

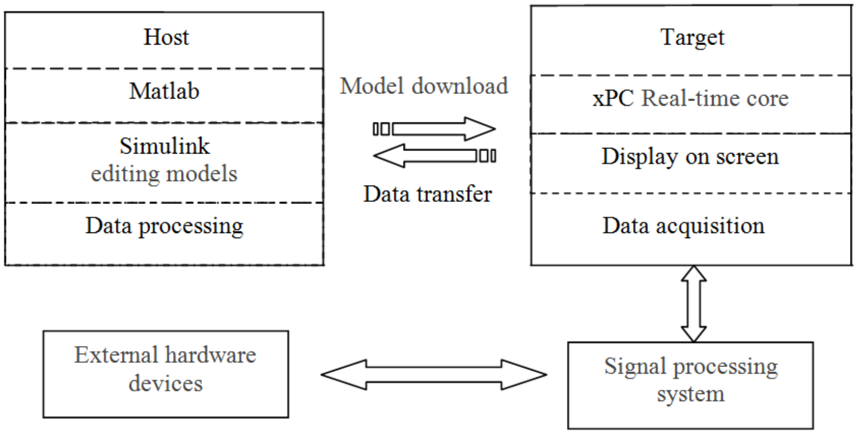


Figure 1. Structure of real-time simulation experiment platform

xPC Target uses host and target machines with dual PC running mode, where there is a platform as a host running Matlab/Simulink to establish real-time simulation model, RTW and C language editor are used to generate real-time code to run on the target machine. The achievement of xPC Target is based on the target machine with running high integration and 32-bit protected mode of real time operating core. The operation of the real core has no relation with the system installed in the target machine. The dual PCs of xPC Target communicate with TCP/IP or RS232 connected [7].

The xPC Target real-time simulation system applied by this experiment platform is mainly composed of one host, one target machine, I/O integrated circuit board of target machine and network connection devices between the host and target machine. The main function of data acquisition card is connected external devices with real-time system. In this experiment platform it sends the braking pressure signal to the target machine and outputs the solenoid valve control signal. It is reasonable to choose the external hardware devices and establish the controlled physical objects of real-time simulation experiment platform.

3. PCI-1710 Driver Development

PCI - 1710 is a 12-bit data acquisition card with multi-function produced by Yanhua company, which has a unique circuit design and perfect data acquisition and control functions and supports plug-and-play with high-speed cache FIFO, sampling frequency up to 100kHz. It can be flexibly to set the input and output types and range. Because it does not supported by xPC Target toolbox, I/O device driver need to be self-developed.

3.1. S function structure

The driver is based on the function C S Under the xPC which is S-function written by C language. The S-function modules is written into the Simulink model, and then compiled and downloaded to the target machine to run. The S-function written by M language cannot generate downloadable codes in Matlab.

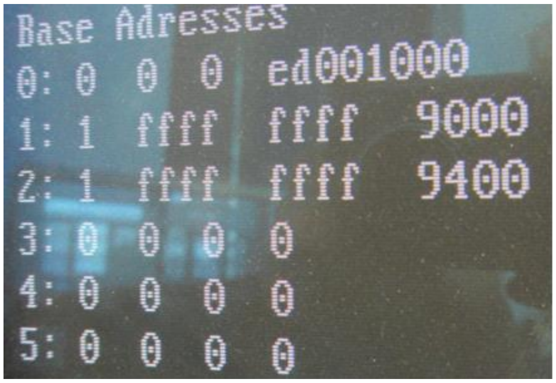
The S-functions typically include three main parts like initialization, continuous operation and end [8]. Among them, the initialization includes start preparing, mdlInitializeSizes, mdlInitializesampleTimes and mdlStart. Start preparing part is used to complete the input parameters and other predefined symbols, including the necessary header files. Input, output, state variables, parameters, and other related quantities of the size initialization number are completed by mdlInitializeSizes. The sampling time initialization of the system function is completed by mdlInitialize sample Times. The initialization of state variables and the required hardware are completed by mdlStart. Continuous operation is done by mdlOutputs function which is repeated when it is run in the target process, and the corresponding tasks which are based on inputs volume and the requirement of model is completed. The end of the part is mdlTerminate function which takes the equipment into a safe condition when it runs to the end.

3.2. Basic points of writing driver

A driver reads and writes to the device registers so that the device is controlled to complete the assigned functions. Before writing a driver, CS function is formatted, application program interface (API) is familiar, as well as board register address information. The main steps of writing driver are as follows.

1) **Obtain PCI card information.** The basic information of PCI-1710 can be obtained through the acquisition card which comes with the testing equipment or by the resource manager under Windows. The access method of PCI interface can be gotten through the device manual. PCI is visited by the firm ID 0x13fe and device ID 0x1710. The address mapping of PCI device is a pivotal problem for the program of the acquisition card, and is divided into two kinds: one is the I/O map; the other is the memory map. This two mapping methods directly affect the device how to get the base address and the access methods of device.

The most important information of the base address is stored in BaseAddress field. According to PCI protocol, PCI BIOS is configured to six base addresses for the devices [9], but some of the base address is invalid. To determine which base address is valid the information obtained earlier can be compared. After determined the base address, the corresponding value of AddressSpace Indicator is the value which is used to determine PCI mapping.The pciinfo value of PCI-1710 is shown in Fig. 2 from which PCI-1710 can be seen as I/O mapping.



```
Base Adresses
0: 0  0  0  ed001000
1: 1  ffff  ffff  9000
2: 1  ffff  ffff  9400
3: 0  0  0  0
4: 0  0  0  0
5: 0  0  0  0
```

Figure 2. Value of PCI-1710 pciinfo

2) **Write the S function.** First determine the S function as adadvpci1710.c. The parameters, such as channel, sample time, signal range, are given. If a target machine is installed in multiple similar equipments, the slot number is also given.When the S function is writen, S function of Matlab's own xPC I/O module and the S function templates are provided by Matlab as a reference, the file is matlabroot/simulink/src/sfutmpl_basic.c.

3.3. Module package and storage

For the convenience of using, the module must be masked and added into the custom module library after the driver finish to be debugged. The assembly editor opens through the mask subsystem, makes the module have its own icon and parameters dialog through module packaging. A custom module library mainly has the following two points:

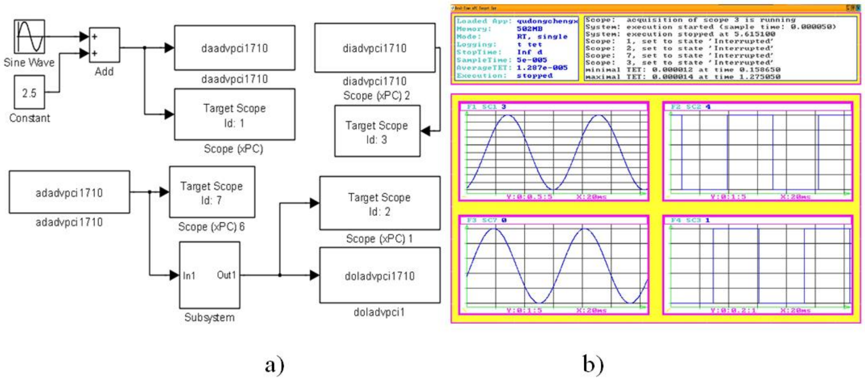
1) **Create a module library file.** The new library command is used to put drive module into the library window in Simulink, which is saved as xpc1710.mdl.

2) Display library browser. The module library created by user-defined is shown in library brower of Simulink. The module that needs to be displayed in the user library and the module name and other information for library browser are provided, which are achieved by the file slblocks.m [10].

The file is copied into the directory, where the user library file exists, and the directory of the user module library is set as the search path of Matlab.

4. Experiment and Results Analysis

After writing the driver program of the acquisition card, each procedure is separately tested, which includes the input and output program of A/D, D/A and DI, DO. The whole test is started. The test model of the driver program is shown in Figure 3 a). The D/A module outputs 0 ~ 5V sine wave, A/D module process the data collection, which are compared and outputted 0 ~ 5V square wave by DO module, DI module collects the waveform back into the system. The experimental results, given in Figure 3 b), show that the developed driver program is correct and stable.



```
Loaded App:  qudongchengx
Memory:      502MB
Mode:        RT, single
Logging:     t tet
StopTime:    Inf d
SampleTime:  5e-005
AverageTET:  1.287e-005
Execution:   stopped
```

```
Scope:   acquisition of scope 3 is running
System:  execution started (sample time: 0.000050)
System:  execution stopped at 5.615100
Scope:   1, set to state 'Interrupted'
Scope:   2, set to state 'Interrupted'
Scope:   7, set to state 'Interrupted'
Scope:   3, set to state 'Interrupted'
minimal  TET: 0.000012 at time 0.158650
maximal  TET: 0.000014 at time 1.275050
```

Figure 3. Driver test of data acquisition card

Solenoid valve and pressure transmitter are the main external devices of the experiment platform. The performances of the electromagnetic valve and transmitter are tested respectively, and then the pressure open-loop experiments are done, where the solenoid valve and pressure transducer constitute an open-loop system for the real time simulation. The closed-loop real-time simulation experiment on the pressure is made with the PID control method. Teal time simulation model is shown in Figure 4 and the experimental results are shown in Figure 5.

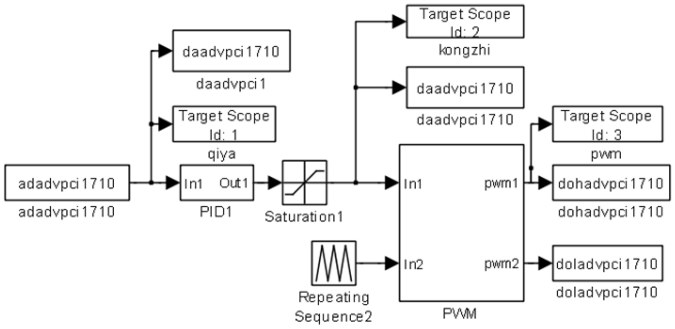
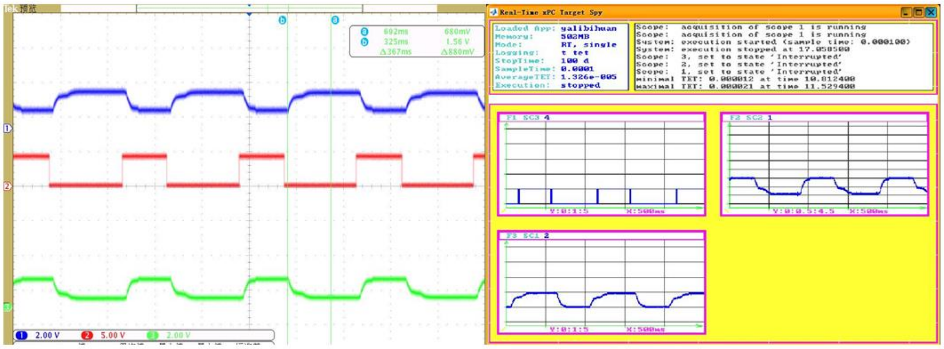


Figure 4. Model of pressure closed-loop test

The results show that the external equipment performance of real-time simulation experiment platform is stable and PID control effect is not perfect. Due to the purpose of this experiment is for testing equipment, the PID parameter is not given systematically. Comparing Figure 5(a) with Figure 5(b), it can be found that the waveforms displayed on oscilloscope are the same with those of the target machine. The waveforms of the target machine are not convenient for observation and measurement, and data are not convenient to save. There are three models in XPC Target module of Matlab/Simulink: Host Scope, Target Scope, and File Scope. Host Scope is used to display the waveform on the host. Target Scope is used to display the waveform on the target. File Scope is used to store the waveform data file into the target machine for data processing. Based on this, it can be sure that the File Scope can replace the oscilloscope to store the experimental data.



```
Loaded App:  yalibihuan
Memory:      502MB
Mode:        RT, single
Logging:     t tet
StopTime:    100 d
SampleTime:  0.0001
AverageTET:  1.326e-005
Execution:   stopped
```

```
Scope:   acquisition of scope 1 is running
Scope:   acquisition of scope 1 is running
System:  execution started (sample time: 0.000100)
System:  execution stopped at 17.058500
Scope:   3, set to state 'Interrupted'
Scope:   2, set to state 'Interrupted'
Scope:   1, set to state 'Interrupted'
minimal  TET: 0.000012 at time 10.812400
maximal  TET: 0.000021 at time 11.529400
```

a) b)

Figure 5. Results of pressure closed-loop test

5. Conclusions

Depending upon the analysis of the real-time simulation technology, the method is proposed on the braking pressure closed-loop control of vehicle in this paper. The real-time simulation experiment platform is designed for braking pressure of vehicle based on xPC Target and the driver of 12 multi-function data acquisition card PCI - 1710 produced by Yanhua Company, by which it can be solved that xPC Target toolbox is not supported by hardware, is developed. The simulation experiment platform is set up on this basis, and the debugging and validation of acquisition card driver has been carry out. Braking pressure closed-loop control of the real-time simulation model is established, and tested with the real-time simulation experiment platform. Experimental results show that the waveforms of target machine are consistent with those measured by the oscillograph. The effectiveness of the real-time simulation platform to adjust the brake pressure is further proved. These can provide the foundation and guarantee for the research on the anti-lock braking system of vehicles.

Acknowledgements

This work is supported by the Heilongjiang Natural Science Foundation (F201125), High Education Teaching Reform Project of Heilongjiang Province(JG2012010257).

References

[1] Y. Chiahan and V. Vyatkin, “Model transformation between MATLAB Simulink and function blocks”, 8th IEEE International Conference on Industrial Informatics, **(2010)** June, pp. 1130-1131.

[2] L. Huan, L. Wenjuan and C. Dongning, “Automatic code generation of SPWM for single phase inverter by model-based design”, 2ndInternational Conference on Measurement, Information and Control, **(2012)** August, pp. 1035-1038.

[3] L. Wenjuan,W. Xudong and C. Fenglin, “Modeling of vehicle braking system based on kinetic energy conversion and self-optimizing control for ABS”, Electric Machines and Control, vol. 2, no. 15, **(2011)**.

[4] The Math works.xPC Target User’S Guide, http://www.mathworks.com/access/helpdesk/help/pdf_doc/xpc/xpc_target_ug.pdf, **(2006)**, pp. 18-20.

[5] W. Chao, D. Jianmin and Y. Yongchuan, “A hardware in loop test system for pneumatic anti-lock brake system”, 2010 International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), **(2010)** August, pp. 105-106.

[6] S. Peilin, M. Lidong, Z. Guangde and X. Jiao, “Development of uniform hardware driver for real-time windows and xPC Target”, 2009 Second International Conference on Information and Computing Science, Manchester, **(2009)** August, pp. 378-379.

[7] Y. Di, L. Litao, Y. Xu and Z. Chengyuan, "Development environment and application of real-time simulation of the system", Beijing: Tsinghua University Press, **(2002)**.

[8] The Mathworks.Simulink Writing S-Functions, http://www.mathworks.com/access/helpdesk/help/pdf_doc/Simulink/sfunctions.pdf, **(2006)**, pp. 59-64.

[9] The Math works.xPC Target Device Drivers Guide, http://www.mathworks.com/access/helpdesk/help/pdf_doc/xpc/xpctarget_driver_gd.pdf, **(2006)**, pp. 26-29.

[10] L. Jie, "Model-based design and the embedded implementation", Beijing: Beijing University of Aeronautics and Astronautics Press, **(2010)**.