Software Design for Real-Time Measuring and Control System Based on LabView and 1553B

Xudan Xu, Ping Zhang
School of Automation Science and Electrical Engineering
Beihang University
Haidian, Beijing, China
Email: xxd@asee.buaa.edu.cn

Abstract—The article described the real-time measuring and control subsystem (MCS) of a certain actuation system briefly first. According to the structure and the specific developing requirements for the subsystem, the article gave the software design thinking based on Labview and 1553B communication standard. The article then introduced the design guidelines, software structures and design details specifically. By the actuation system application, the software was proved to satisfy the application requirements fully.

Keywords- LabView; software design; real-time; 1553B

I. INTRODUCTION

The more complex modern applications for actuation systems generate stricter requirements for control accuracy and real-time capability. They both have become the evaluation standards of actuation systems[1]. To test a new actuation system’s control accuracy and execution efficiency and to get the first-hand characteristic datum of the system, a new measuring and control subsystem (MCS) is developed. Compared with traditional ones, the new MCS software is developed by LabView language based on 1553B communication standard, which confirms the comfortable human-computer interaction interface and the real-time capability of the system. What’s more, the whole software is divided into two parts: the one in the industry computer and the onboard one. This offers high execution efficiency and high control accuracy. The MCS hardware is realized by DDC and NI developing kits, featuring high reliability and good maneuverability.

A. The Developing Requirements of the MCS

In order to observe the new system’s working character effectively and to obtain high control accuracy, the real-time capability and mass data processing are requested by the MCS. The sampling interval is set to 4ms. And the subsystem sends 8 control words and samples 50 remote data words during one period. At the same time, it saves the sampled data every 50 intervals. After finishing real-time data sampling, the MCS needs to process about up to 4 million data, plotting, displaying, classifying and then saving them in files.

The measuring software should also be featured by the functions hereunder in order to satisfy the MCS’s flexibility and observability.

- Optional control commands: operators could choose the control commands according to testing tasks;
- Optional channels for plotting and displaying: operators could choose the channels for data being plotted and displayed according to testing tasks;
- Saving the sampled data: save the sampled data in files in case of data processing later;
- Automatic measuring process: the program can do measurement and control automatically after the control commands needed are reordered;
- Optional start-up method: operators could choose to start the program by hardware triggering or by software;
- Sampling 4-channel analog data timely: sample 4-channel data with the frequency of 100Hz and then save them in files
- Pause function: end the process forcibly in case of emergency, and then save the present data in files.

B. A Brief Introduction for System Hardware Structure

From Figure 1, it can be seen that the MCS is divided into two portions: the local one and the remote one. In the local one, Siemens industrial computer connects PXI bus expanded card in remote NI chassis through NI PCI Bus expanded card by fibers, so that the local industrial computer can control remote 1553B bus by being expanded to the remote one, as sending local control commands and sampling remote terminal data, etc. The industrial computer samples triggering signal by NI DAQ. In the remote one, the remote NI chassis integrates DDC...
1553B communication cards and NI DAQ by PXI slots. And the chassis connects 3 Remote Terminals (RTs) by 1553B bus. The three RTs are the same and receive same commands from the bus simultaneously. So that the MCS has three redundancy.  


1553B digital time division command/response multiplexing data bus is used widely in various avionics systems. The bus standard has double-redundancy, and transmission can be switched between Bus A and Bus B according to present situations. Its transmission speed can be up to 1Mb per second. One 1553B system could consist of one Bus Controller (BC), up to 31 Remote Terminals (RTs) and one Bus Monitor (BM). The BC controls bus communication, like self-test, synchronization, sending data and receiving data, etc. The RTs reply to control commands and the BM monitors transmissions on the bus. The bus standard follows the command/response model, that is RTs will not reply unless control commands are given by the BC. And the standard does communication by “message”. Every message contains status words, command words, model words and data words. The standard has been proved to feature high reliability, good stability and is easy to be developed. In the MCS, all control commands and remote data are exchanged between BC and RTs following 1553B communication standard.

II. SYSTEM SOFTWARE DESIGN

To obtain a comfortable human-computer interaction interface, the MCS software is developed by LabView language. LabView is a graphics language based on data flow, which is visible to be programmed and easy to be debugged. At the same time, LabView language integrates a lot of developing functions in accordance with NI products[3].

The MCS software features a function of sending control commands to three RTs from BC with a preprogrammed and adjustable interval, which is decided by the given parameters and selected control commands. After finishing the sampling process, the MCS software plots and displays the characteristic graphs in accordance with the operators’ selection. And it saves all the collected data in files. At the same time, the MCS software samples 4-channel analog data in each RT with the frequency of 100Hz.

A. Software Design Guidance

To satisfy the real-time request, the MCS software consists of software in the industrial computer and the onboard software. The onboard software fixes the sampling interval accurately by the onboard timers, then sends control commands and saves the sampled data; while the software in computer reads the data saved in onboard RAM and rewrites new control commands into it during every interval. With the help of hardware timers on board, the system achieves the timing accuracy of 0.001ms, which is a big progress [4,5].

B. Software Architecture

The MCS software has a sequence structure. It is divided as six frames: Initiation Frame, Command Ordering and Data Pretreatment Frame, Characteristic Array I Frame, Characteristic Array II Frame, Characteristic Array III Frame, Data Processing Frame and End Frame. And there are two “whiles” of “Analog Data Sampling” and “Pause” besides the main structure. During real testing, Characteristic Array II and Characteristic Array III are empty if no commands of a certain set of characteristic testing tasks are chosen. Please refer to Figure 2 for the software block.

Hereunder is the brief introduction for each frame:

- Initiation Frame: all controls are reset, data saving space is opened up, and all parameters are set to default values according to chosen character testing tasks.
- Command Ordering and Data Pretreatment Frame: all the hardware are reset, the value changing in front panel is recorded. All control parameters and sampling channels in which data need to be plotted and displayed are saved. The commands sequence is decided according to the operators’ configuration and it is saved in the “control command sequence array”. At the same time, relevant arrays are initiated with appropriate values in this frame according to the selected commands. Timers and analog sampling channels are also reset.
Characteristic Array I Frame: a set of character testing I commands is carried out with the interval of 4ms. The control command table in the BC card is refreshed, and the data being sent from RTs to BC’s data table every 50 periods are acquired. The analog data in the two redefined channels are sampled with a frequency of 100Hz. And at the same time, the software is polling the “pause” signal. The frame is ended after BC finished the expected commands.

Characteristic Array II Frame: a set of character testing II commands is carried out with the interval of 4ms. The control command table in the BC card is refreshed, and the data being sent from RTs to BC’s data table every 50 periods are acquired. The analog data in the two redefined channels are sampled with a frequency of 100Hz. And at the same time, the software is polling the “pause” signal. The frame is ended after BC finished the expected commands.

Characteristic Array III Frame: a set of character testing III commands is carried out with the interval of 4ms. The control command table in the BC card is refreshed, and the data being sent from RTs to BC’s data table every 50 periods are acquired. The analog data in the two redefined channels are sampled with a frequency of 100Hz. And at the same time, the software is polling the “pause” signal. The frame is ended after BC finished the expected commands.

Data Processing Frame: the 1553B cards and sampling process of analog data are stopped in this frame. The relevant graphs in accordance with the sampled data and operators’ configuration are plotted and displayed. Sampled remote data are saved respectively in the files of “RT1”, “RT2”, “RT3” and “Analog data sampling”.

End Frame: the 1553B cards are closed in this frame. Sampling task is cleared, and the industrial computer EMS memory is released. Then the software exists LabView developing environment after all above are finished.

Analog Data Sampling: the “Analog Data Sampling” bit is set every 1ms and saved in FIFO in this frame.

Pause: the software polls hardware triggering and set the “Pause” bit in this frame.

C. Detail Design

The software realizes the data communication between software in the industrial computer and the onboard software utilizing interrupts. The software in the industrial computer understands the present data transmissions through the interrupt generated by remote onboard software and refreshes the onboard data every 50 periods.

The three array frames in the program are combined by the way of being hung up, being rewritten and then operating. When the former frame is finished, the 1553B BC is hung up. Control commands are calculated again and rewritten into BC’s onboard control command table. And the 1553B control frame is reconfigured and downloaded to the 1553B BC card, then BC runs. In the program, the industrial computer rewrites the BC’s onboard control command table every 50 periods, for these are so many control commands for every character testing that they can not be written into the command table at one time. During the rewriting, the onboard software is still running, ensuring the real-time capability.

The MCS software needs to process about up to 4 million data for every complete character testing of the actuation system. In order to utilize the industrial computer EMS memory effectively, queue technology is chosen to save data in the program. For the queue has the feature of “being opened up if these are data and being released as soon as possible if no[3]”. And the program adds C code in “Formula Node” to simple the calculation process because of the language C’s super calculation ability[6].

III. TESTING RESULTS

Hereunder are figure 3 and figure 4, two of character testing tasks in the MCS. The graphs are plotted according to control commands. In figure 3, the command is a sine wave signal with the amplitude of 4 and frequency of 0.02Hz. And in figure 4, the command is a round wave signal with the amplitude of 4 and frequency of 0.02Hz. It can be seen from the figures that the signals satisfy the requirements completely. And from the figures, it’s known that the parameters in this program could be configured optionally and commands could be reconstructed. The control signal has good accuracy, which meets the testing requests.

![Figure 3. Control commands for one character testing task](image1)

![Figure 4. Control commands for another character testing task](image2)
IV. CONCLUSION

After being used as a measuring and control subsystem in a certain actuation system for nearly one year, the MCS owns the attributes of configuring parameters optionally, selecting control commands autonomously, and sampling and processing remote data automatically. And the “Pause” function avoids the emergency effectively. By the actuation application, the MCS is proved to possess qualities of high control accuracy, great stability and comfortable human-computer interaction interface. In a word, it fully satisfies the real application requirements.

ACKNOWLEDGMENT

The authors gratefully acknowledge Professor Chen’s advice and supervision. And the authors sincerely thank the reviewers for their careful reading of the manuscript and their valuable recommends.

REFERENCES


