2010

Implementation and Application of a Real-Time Control System using Matlab/xPC

Xiaofeng He
xhe@uow.edu.au

Zhengyi Jiang
University of Wollongong, jiang@uow.edu.au

Christopher David Cook
University of Wollongong, chris_cook@uow.edu.au

Yinshui Liu
Huazhong University of Science & Technology, China

IMPLEMENTATION AND APPLICATION OF A REAL-TIME CONTROL SYSTEM USING MATLAB/xPC

XIAOFENG HE

School of Mechanical Science & Engineering, Huazhong University of Science & Technology, No. 1037 Luoyu Road, Wuhan, 430074, China

ZHENGYI JIANG

Faculty of Engineering, University of Wollongong, Northfield Road, Wollongong, NSW 2522, Australia

CHRISTOPHER COOK

Faculty of Engineering, University of Wollongong, Northfield Road, Wollongong, NSW 2522, Australia

YINSHUI LIU†

School of Mechanical Science & Engineering, Huazhong University of Science & Technology, No. 1037 Luoyu Road, Wuhan, 430074, China

A real-time control system for rolling mill control such as the Hille 100 experimental rolling mill was developed based on Matlab Simulink, Real-Time Workshop and xPC Target. The Host-Target PCs architecture for the control system was introduced, then the approach for the implementation of data acquisition, real-time control and the associated graphic user interface (GUI), were discussed in detail. Some limitations of the commercial software were also presented. The developed control system has been verified with Hille 100 rolling mill in laboratory, indicating the effective application in rolling mill control.

Keywords: Rolling mill, Data Acquisition, Real-time Control, Matlab/xPC

1. Introduction

Modern control technologies are being applied in almost every field of industries. In general, a control system design process consists of design, modelling, simulation, implementation and evaluation.[1] Subsequently the design is modified until all the system demands are fulfilled. These processes are often complicated, and need a long-term to be finished. However, with the

† Corresponding author, email: liu_yinshui@163.com

development of computer science, the complicated work is becoming simple. The Matlab software provides such rapid prototype tools for achieving hardware-in-the-loop simulation or constructing a real-time control system.[2-7]

An AGC (Auto Gauge Control) control system for a Hille 100 experimental rolling mill was developed successfully in 2008 by using tools of Simulink, Real-Time Workshop and xPC Target in Matlab 2009a. The goal of the system is to achieve real-time control of the roll gaps, as well as the data acquisition during rolling process. In this study, the detail of how to implement a data acquisition, real-time control and the associated GUI will be presented.

2. Architecture of Real-Time Control System

The developed real-time control system adopted the Host-Target type architecture. The Host PC uses Windows XP as an operating system combined with Matlab 2009a, which contains Simulink 7.3, RTW 7.3 and xPC Target 4.1 toolboxes.

Simulink provides a graphical user interface (GUI) for building models as block diagrams. xPC Target 4.1 supports a serials of I/O hardware in its block library, which can be added to the model.[8] RTW is used to generate code from the block diagram model. Then the code can be compiled to the executable real-time kernel using Microsoft Visual C/C++ or Watcom as a compiler, and downloaded directly to the Target PC via TCP/IP or RS-232 protocol.[2, 9] The Target PC is booted by a disk created from Host PC, and runs the real-time kernel in a DOS environment. The DAQ Card installed on the Target PC will be connected to the device sensors and amplifiers with its AD and DA ports, respectively.

So a great feature of this architecture is to build a bridge between the simulation model and hardware without writing low-level programming language, as well as to ensure the system's real time performance.

3. Implementation of Data Acquisition

Figure 1 gives the roll gap control model on the operation side for the rolling mill. It contains rolling forces data acquisition blocks, which is not only for monitoring process parameters, but also providing data source for calculating and adjusting the input signals of the control system.

Because the measured signals are inevitably coupled with many kinds of disturbance with high frequency in work site, therefore the low pass filters are often adopted for data acquisition. Matlab software provides very powerful Filter Design & Analysis Tool (Fdatool) to design digital filter blocks which can be added to the model. However, the characteristics of different types of filters are different. Taking two typical low pass filters, Butterworth and Equiripple, as example, it can be found by simulation that the Equiripple filter will cause more steady error compared with the Butterworth filter. Therefore, the Butterworth filters were finally selected for the model in Figure 1.

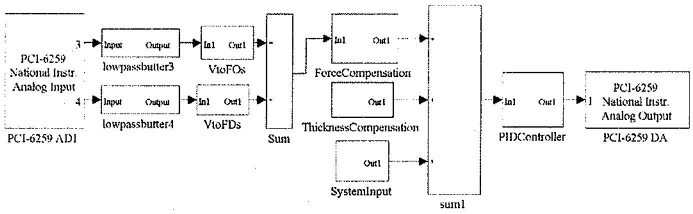

Figure 1 Block diagram of the roll gap control on the operation side for Hille 100 rolling mill

4. Implementation of Real-Time Control

The main function of the model in Figure 1 is to send the overall input signals to the PCI-6259 DA port and control the servo valve by a G122-202A1 servo amplifier. The overall input signal is integrated by the preset value, the force compensation value and thickness compensation value. The preset value is composed of the Entry and Desired Strip Thicknesses input through the Host PC by operator, and the Initial Roll Gaps which are measured before rolling.

The force compensation value C_F adopts the following fundamental equation:

$$C_F = \alpha_F dPM/2 \text{ (mm)} \tag{1}$$

where M is the rigidity of the rolling mill, α_F is the force compensation coefficient which can be obtained by test, α_F=0-1. dP is the rolling force variation compared with a base value.

The thickness compensation value C_{TH} is obtained by

$$C_{TH} = (h_{ave} - h_d)((W + M)/(W + M - \alpha_{TH}W)) \text{ (mm)} \tag{2}$$

where h_{ave} is the average exit strip thickness during last rolling process, h_d is the desired exit strip thickness. α_{TH} is the thickness compensation coefficient, α_{TH} =0-1, W is the deformation resistance of the rolled material.

5. Design of Graphic User Interface (GUI)

After modelling, xPC Target Explore tool can be used to load, build and download the model to the Target PC, then run in real-time, as well as modify the blocks' parameters on-line. However, this GUI is not friendly for users.

Figure 2 illustrates the GUI for Hille 100 short strip rolling process designed by using GUIDE layout panel.[10] This GUI has the following main functions.

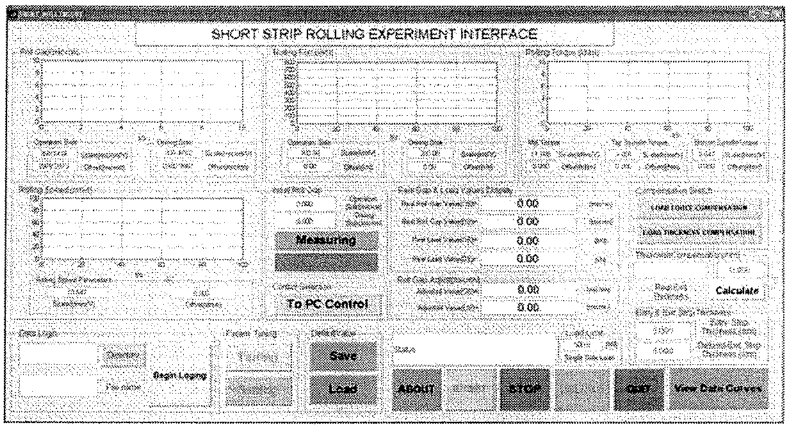

Figure 2 GUI for Hille 100 short strip rolling process

1) Measuring and Submit buttons: measure the initial roll gaps of the work rolls, then take their average values and display on the screen.

2) Tuning and Setting buttons: tune the controller parameters, select the mill type and material, and set the sampling parameters, etc.

3) START button: set xPC environment, load and build the model and download it to the Target PC, set the parameters of the Target object, then start the application. When the program is running, the acquired data can be displayed on the screen with curves and numerical values. In addition, the rolling mill can be switched to the PC or Manual control.

4) Begin Logging toggle button: to select if the experimental data is saved or not.

5) View Data Curves button: open another GUI and plot data curves for viewing and analysing the results after experiments. Figures 3 to 4 are the displacement (roll gap), rolling force on the operation side respectively when rolling a hot strip with entry thickness 12mm, reduction 5.95mm. From Figures 3 and 4, it can be seen that when the strip was moved into the roll, the gap increased about 150 μ m with the increase of rolling force, and then they kept relatively stable values. When the strip was running out of the work rolls, the roll gap decreased with the rolling force and was recovered to its original state after a short transient process.

6. Limitations of the Matlab/xPC

During using Matlab/xPC, some limitations of the Matlab/xPC were found:

- Not all the xPC Target blocks have been tested to be perfect. For example, the PCI6259 A/D block in the xPC Target Library (Version 4.1) was verified to have bug. When this block was used to achieve multi-channels data acquisition, the correct results could not be obtained until a file named 'xpcionim.c' was replaced with a patch provided by The MathWorks Inc.
- The compatibility between different versions of software is not so good. Some commands' formats from Matlab Help files or its website may not be supported by the using version.

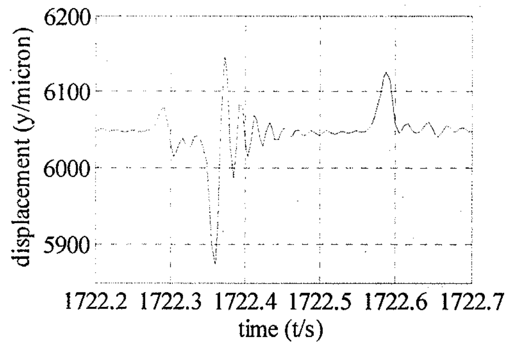

Figure 3 The displacement (roll gap) curve on the operation side when rolling a hot strip with entry thickness 12mm, reduction 5.95mm

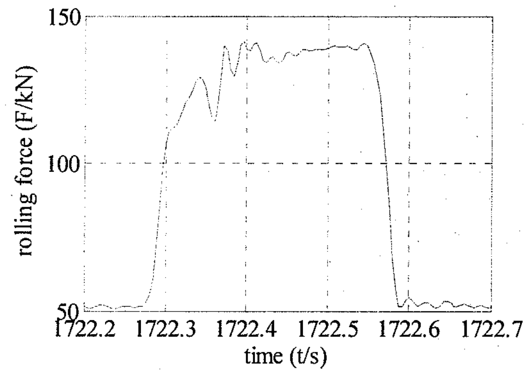

Figure 4 The rolling force curve on the operation side when rolling a hot strip with entry thickness 12mm, reduction 5.95mm

7. Conclusions

This paper presents the implementation of a real-time control system for a Hille 100 experimental rolling mill based on Matlab/xPC. The details of constructing data acquisition, real-time control system and the design of the associated GUI are discussed. The developed control system was successfully applied to practice and upgraded the performance of the rolling mill.

Although limitations of Matlab/xPC have been found during the implementation process, they will not handicap to verify that the Matlab/xPC has powerful functions which will make the modelling, simulation and implementation of a real-time control system easy and fast. This work will provide a reference for the rapid implementation of real-time control or monitoring systems not only in steel industries, but also in many other automatic control fields.

Acknowledgements

The work was financially supported by the Faculty of Engineering, University of Wollongong, Australia.

References

1. C. Ridderström, J. Ingvast, J. Wikander: Proceedings 2001 ICRA. IEEE International Conference on 'Robotics and Automation', Seoul, Korea, May 2001, IEEE, 1327-1333.
2. P. S. Shiakolas, D. Piyabongkam: Proceedings of the 40th IEEE Conference on 'Decision and Control', Orlando, Florida USA, December 2001, IEEE, 1348-1351.
3. A. Adgar, C. S. Cox: UKACC Control 2004 Mini Symposia, Bath UK, September 2004, University of Bath, ID-095.
4. Asan Gani, M. J. E. Salami-SMIEEE, Md. Raisuddin Khan: Proceedings of 2003 IEEE Conference on 'Control Applications', Istanbul Turkey, June 2003, IEEE, 538-544.
5. K. H. Low, Heng Wang, Michael Yu Wang: Proceeding of the 2005 IEEE International Conference on 'Automation Science and Engineering', Edmonton Canada, August 2005, IEEE, 345-350.
6. F C Teng: 2000 IEEE International Conference on 'Systems, Man, and Cybernetics', Nashville USA, October 2000, IEEE, 2697-2702.
7. Cheol Hoon Park, Sang-Kyu Choi, Young Su Son, Young Hee Han: 2009 World Congress on 'Computer Science and Information Engineering', Los Angeles USA, March 2009, IEEE, 701-705.
8. 'xPC Target User's Guide', 10, 2002, Natick USA, The MathWorks, Inc..
9. 'Real-Time Workshop User's Guide', 1, 1999, Natick USA, The MathWorks, Inc..
10. 'Creating Graphical User Interfaces', 1, 2008, Natick USA, The MathWorks, Inc..