Design Control System of Telescope Force Actuators Based on Linux

Shuai Xiaoying [1,2,3]
1. National Astronomical Observatories/Nanjing Institute of Astronomical Optics & Technology
Chinese Academy of Sciences
Nanjing 210042, China
e-mail: xyshuai@niaot.ac.cn

Zhang Zhenchao [1] , Wang You [1], Ni Jijun [1]
2. Graduate School of the Chinese Academy of Sciences
Chinese Academy of Sciences
Beijing 100049, China
e-mail: zczhang@niaot.ac.cn

3. Dept.of Computer Science, Chizhou College, Chizhou Anhui 24700, China
e-mial:xyshuai@niaot.ac.cn

***Abstract*—By analyzing the characteristics of large telescope and the function structure of telescope actuator control system, distributed Ethernet force actuators control system is applied. To improve the performance of stability and real-time, developed software of force actuators control system in real-time Linux, a hiberarchy schedule scheme based on RTAI is presented. Adopt QT designer to design the user interface of control system in master computer. Master computer and controllers intercommunicate asynchronously no-block via QServerSocket, QSocket and DSocket, design communicate frame format. In order to communicate among processes, SLOT and SIGNAL are adopted. Finally depicted the main algorithm of data received in main controller.**

Keywords- Force actuator control system, Real-time Linux, RTAI, Qt designer

I. INTRODUCTION

In the large telescope, active optics system adopted deformable mirror active optics technique and segmented mirror active optics technique. Control system of force actuators is the key components of deformable mirror active optics. By analyzing the characteristics of telescope and the function structure of actuator control system, distributed Ethernet force actuators control system is applied.

To improve the performance of stability and real-time, we developed master controller software of force actuators control system in real-time Linux, a hiberarchy schedule scheme based on RTAI is presented. Adopt QT designer to design the user interface of control system in master computer. Main controller and slave controllers inter-communicate asynchronously no-block via QServerSocket, QSocket and DSocket with special frame format. Parser and encoder parsed and coded the communicate data between Master and external servers on XML format. Data module deal with the information received from controllers or sent to controllers, expatiated on the process of information. In order to communicate among processes, SLOT and SIGNAL were adopted.

The reminder of this paper is organized as follows. Section II gives a brief descript of topologic structure of telescope force actuators control system. Section III describes QT designer and the design of master control system GUI, signal and slot, and real-time scheme. In section IV, we present intercommunication between main controller and slave controllers. The algorithm of data received in main controller is shown in section V. Finally, we present the conclusion in section VI.

II. FORCE ACTUATORS NETWORK TOPOLOGIC STRUCTURE

Large telescope was adopted distributed Ethernet network structure in order to allow for distribute, real-time, reliability and security of system[1]. TCP/IP has become the de-facto standard for communication. TCP provides reliable full-duplex stream transmission[2]. Controllers communicate with actuators in large telescope control system via TCP/IP.

Fig. 1 shows framework of telescope force actuators control system. The system consists of computer control system, force actuators and optic system. The main task of optic system is wave-front measure and wave-front analysis. Active optic server computed the force value of force actuators by the result of wave-front measure. Main controller and slave controllers compose computer control system. The main controller receives the force value from TCS server, then, sends command to a slave controller. The slave controller control force actuators. Force sensors feed back running result of actuators to computer control system which corrects the error. Each submirror of MA is controlled by n (such as 5) slave controllers. Each controller controls 8 force actuators, the last control the odd actuators

III. DESIGN OF MAIN CONTROLLER SYSTEM

A. Design of user interface

To improve the performance of stability and real-time, we developed master controller software of force actuators control system in Linux. We developed GUI of control system with Qt designer. Qt is the de facto standard C++ framework for high performance cross-platform software development. The Qt toolkit is a set of tools for building multiplatform GUI programs using a "write once, compile anywhere" approach.

The Fig. 2 shows how Qt works[3].

978-0-7695-3679-8/09 $25.00 © 2009 IEEE
DOI 10.1109/ICIE.2009.123

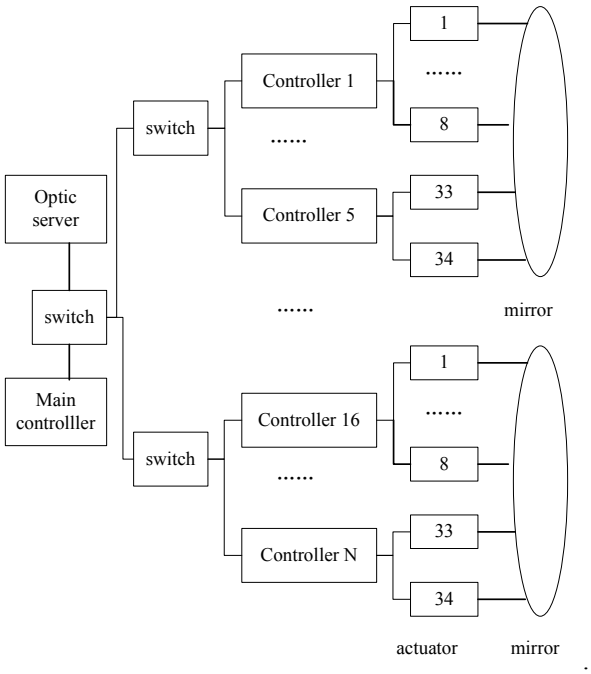

Figure 1. Framework of telescope force actuators control system

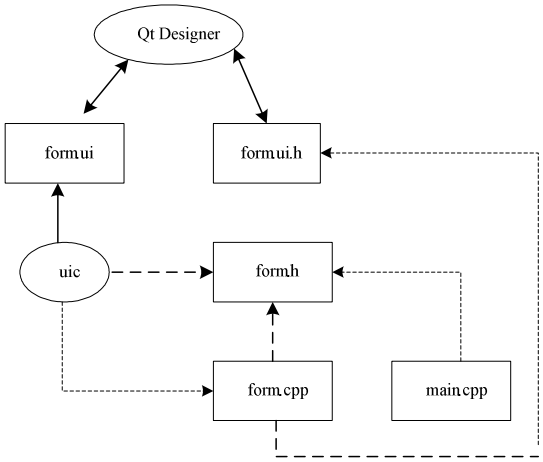

Figure 2. Work process of QT desginer

In addition to the .ui file, form.ui, Qt Designer reads and writes another associated file form.ui.h. This .ui.h file is an ordinary C++ source file that contains implementations of custom slots The user interface compiler, uic, creates both a header file and an implementation file, from the .ui file. The application code is in main.cpp.

The main controller system is designed by QT. It consists of main window application (main telescope force control form), is shown as Fig. 3.

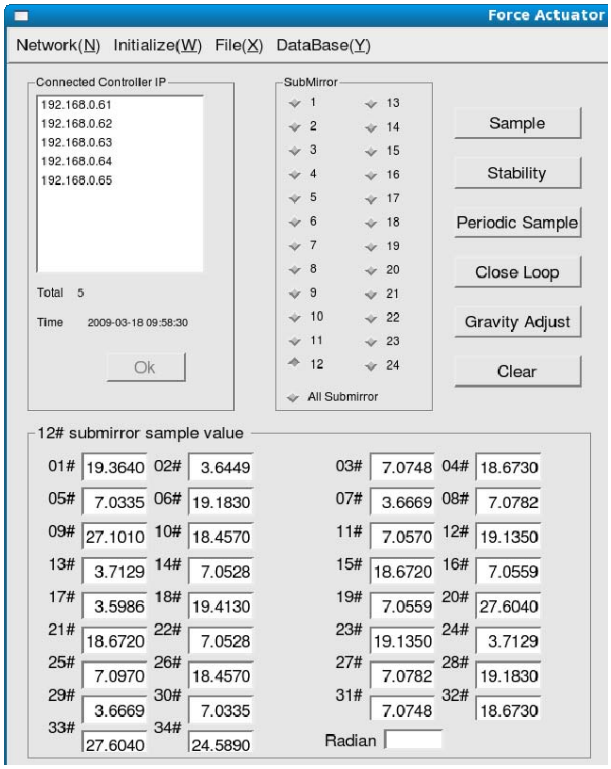

Figure 3. Main telescope force control form

B. *Signal and slot*

Signals and slots provide inter-object communication facilities. Signals are emitted by an object when its internal state has changed in some way that might be interesting to the object's client or owner. When a signal is emitted, the slots connected to it are executed immediately, just like a normal function call[4]. Signal and slots are used for communication between objects of telescope force actuators control system. The most signal and slots are shown in Fig. 4.

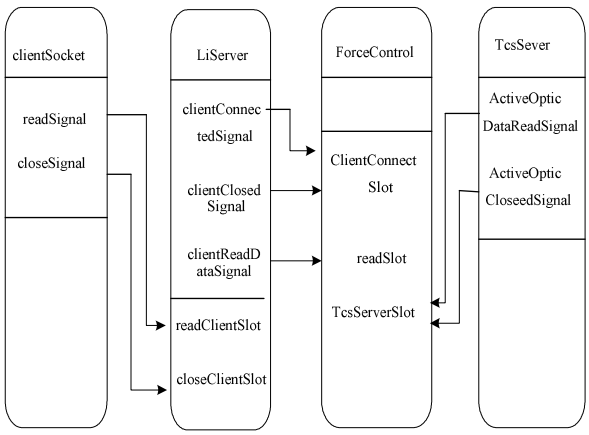

Figure 4. Signal and slot

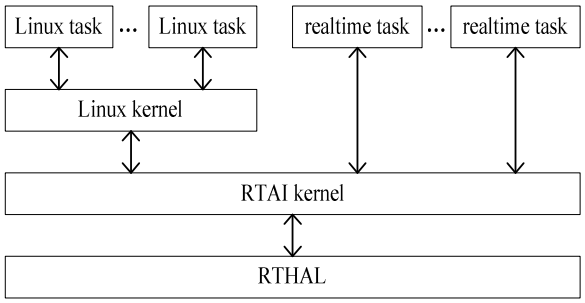

Figure 5. Linux RTAI system architecture

C. *Real-time scheme*

All interrupts are initially handled by the real time kernel and are passed to Linux only when there are no active real time tasks under RTAI (Real-Time Application Interface). RTAI was developed as a real-time operating environment [5]. Fig.5 shows Linux and RTAI system architecture. RTHAL (Real Time Hardware Abstract Layer) provides a frameworks onto which all hard interrupts are intercepted and routed. RTAI offers typical switch times of 4uSec, 20uSec interrupt response, 100KHz periodic tasks and 30KHz one-shot task rates. Kun Ji[6] proposed a real-time control over networks with RTAI, Ambike[7] proposed a real-time control system by using Linux with RTAI.

The transmitted data include: burst real-time data, periodic real-time data, and not real-time data. Burst real-time data were provided with the highest priority, periodic data took second place, not real-time data were deal with linux kernel. We expand RTAI scheduler, add RM schedule and EDF schedule. EDF scheduled burst real-time data, RM schedule periodic data. RTAI transmitted data to user interface with real-time FIFO. The algorithm is follow:

```
int rtai_schedule(void)
{
switch(RTAI_SCHEDULE_TYEP)
{
case SCHED_OLD:
    return rtai_schedule_old();
    break;
case SCHED_RM:
    return rtai_schedule_period();
    break;
case SCHED_EDF
    return rati_schedule_burst();
    break;
}}
```

The modified scheduler achieved better performance.

IV. DESIGN OF NETWORK COMMUNICATION

Main controller communicates with slave controllers and TCS server. Asynchronous no-block QServerSocket and QSocket allow system to work in a portable way with TCP/IP sockets[8]. The class LiServer inherited QServerSocket, work as the role of server in main controller. Slave controllers run in DOS OS, achieved socket via DSOCK[9]. When a slave controller send a request for connecting LiServer, LiServer new a clientSocket inherited QSocket to exchange information with the slave controller. Fig.6 shows the process of communication between the main controller and a slave controller. Data format of exchange is an aptotic frame, shown as Fig. 7.

The main controller and TCS server exchange data based on XML. TCS server get force value that force actuator will execute. Then, it encodes the value into XML stream by XML format and sends the stream to main controller with socket. The main controller parses the package with XML parser after receives the XML stream, vice versa.

V. DATA PROCESS

The main function of main controller is deal with the data feed back from slave controllers. The main controller stores data from the slave into Recv, it is a two dimension array. Then program reads the data from Recv to compute and analysis, the algorithm as follows:

1)Read Recv[0], if the value is consisted with the number of selected sub-mirror then go on, else stop.

2)Read Recv[1], if the value is equal to the number of a slave controller then deal with the data, else stop.

3)Read Recv[2], if it is equal to 1 then goto 4; if it is equal to 2 then goto 5; if it is equal to 3 then goto 6; if it is equal to 4 then goto 7.

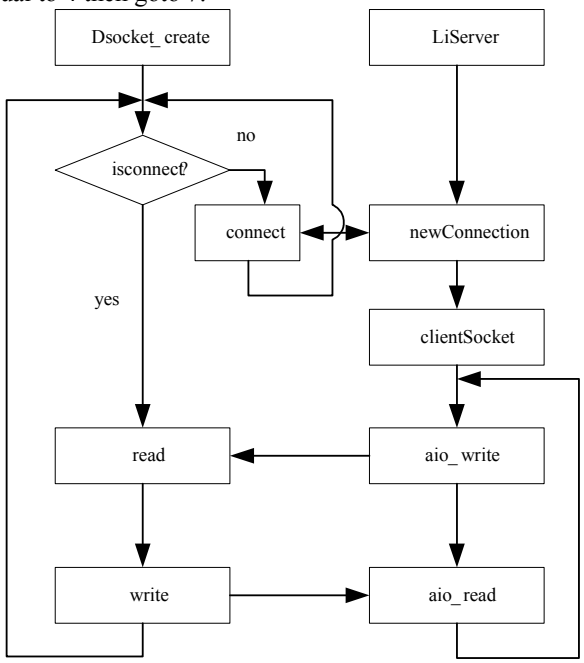

Figure 6. Communication between main controller and a slave controller

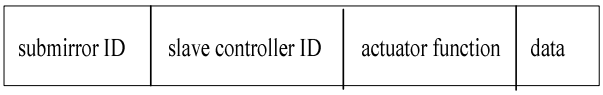

Figure 7. Frame format

4)Read data from 8 force actuator channels to compute the force value, if the sign is equal to 1, the force value is positive, else is minus.

5)Read Recv[3], it denotes the actuators loop control is successful if the value is 1 else fail.

6)Compute the error between the required force value and the fact force value.

7)If the sign is 1, transmitted data is positive, else is minus.

The flow of execution as follow Fig. 8:

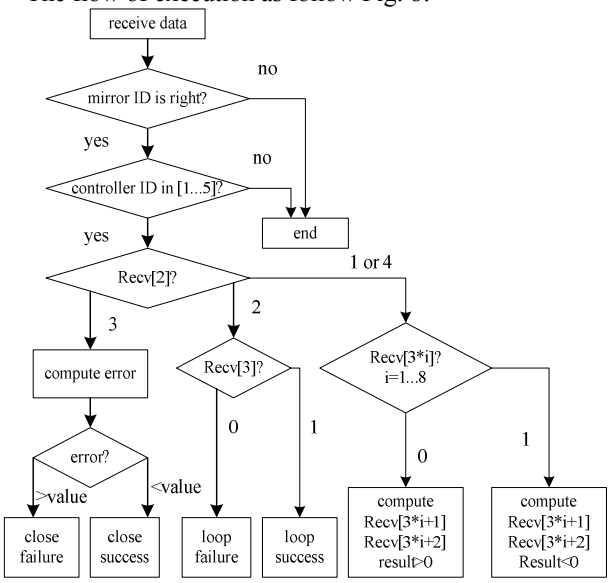

Figure 8. The data flow of execution

VI. CONCLUSION

Large telescope adopted deformable mirror active optics technique and segmented mirror active optics technique. There are thousands of force actuators in control system. Distributed Ethernet network improves the performance of control system. The main controller system was developed in Linux, made use of QT designer to design the GUI. The control system was provided with favorable user interface and real-time response.

REFERENCES

[1] Xu Xinqi, "The control system of LAMOST", Acta astrophysica Sinica, Vol.20 Supp., 2000,pp.43-52。

[2] Andrew S. Tanenbaum, Computer Networks, fourth edition, Prentice Hall, 2003.

[3] Trolltech, The Designer Approach, Qt Assistant,2007

[4] Jasmin Blanchette, Mark Summerfield, C++ GUI Programming with Qt 3, Prentice Hall PTR, January 15, 2004.

[5] Mike Angelo, Paolo Mantegazza, "RTAI Programming Guide", http://www.rtai.org, 2006.

[6] Kun Ji,"Real-time control over networks", Texas A&M University,2006.

[7] A. Ambike, "Closed-loop real-time control on distributed networks",M.S. Thesis, Texas A&M University, 2004.

[8] Wu Jun, Xiao Zhitao,"Design and implementation of embedded communication system based on S3C2410A", Application Research of Computers, China, June, 2008, pp.1897-1900.

[9] Zhang Shengtao, Zhang Zhenchao, Wang You,"Control and network system of force actuators for deformable mirror active optics in LAMOST", Proceedings of the SPIE, Volume 6721, 2007,pp. 1-8.