Architecture Design for a MIS Robot Control System

Xingpeng Da[1] Qixin Cao[1] Peng Chen[1] Masaru Adachi[2]

daveying@sjtu.edu.cn qxcao@sjtu.edu.cn cp_dream_sjtu@qq.com Masaru.Adachi@yaskawa.co.jp

1. Institute of Biomedical Manufacturing and Life Quality Engineering, Shanghai Jiao Tong University, Shanghai, China

2. Tsukuba Research Laboratory, Yaskawa Electric Corporation, Tsukuba, Japan

1. Introduction

Minimally invasive surgery (MIS) robot has been more and more used in hospital because of its no opening, small incisions, less pain and quicker recovery. The MIS robot has great advantages when compared with traditional minimally invasive surgery methods, since it can make up the weakness of small view perspective and low flexibility of traditional minimally invasive surgery, surgeons could operate minimally invasive surgeries better with MIS robot, because MIS robot can realize precision operations of surgical instruments, so MIS robot has great application prospects.

There are many MIS robots have been used in many MIS procedures such as da Vinci [1], Zeus system [2], and RAVEN-II[3,4]. da Vinci and Zeus system have its own software control system, and RAVEN-II is based on Robot Operation System[5] (ROS) which is widely used in industrial robot and service robot.

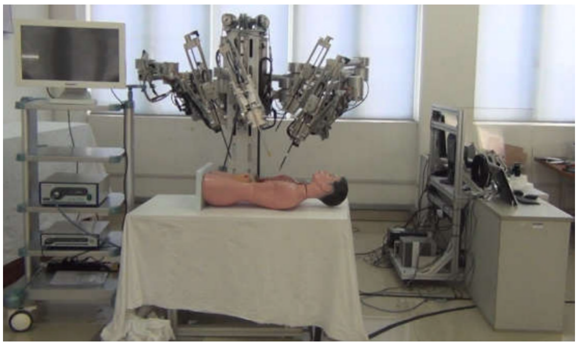

Fig. 1 MIS robot developed in SJTU

Figure 1 shows a MIS robot system developed in SJTU, which has four arms, two of them are used to hold instruments, and one of them at front is used to hold endoscope and a backup arm at back. Besides the main structure, the MIS robot system also has a video platform and a doctor's console. Doctors' console is the place at where surgeons operate, which consists of two master manipulators, a naked eye 3D screen, a set of treadles, a touch screen and some indicators. Video platform consists of endoscopic video processing system and a screen which is for all operating room staff. The control system consists of two sub systems and run in two separate computers, one is for MIS robot control that is called slave system and located at robot side, another is for dealing with surgeon's operations, such as master manipulator data sampling and treadles event processing, this part is called master system.

2. Architecture Design for Control System

2.1 ROS and OpenRTM

ROS is an open source platform for robot control and simulation; it has attracted a lot of people because it's flexible communication mechanism, large number of open source packages and efficient hardware description method. It is very easy to construct a complex system with ROS.

But ROS also has limitations in our system, because we cannot run two roscore (core services of ROS) in the same system, and our control system consists two parts and run in two computers separately, if we just use ROS, that means we must choose one computer to run roscore, and another should communicate with roscore via network connection, if connection between two computers fails, the MIS robot may out of control, this is very dangerous.

So we introduced OpenRTM technology to deal with the communication task between these two computers, and we can run two roscore in two computers with no conflict. Every component can be abstracted as robot technology component (RTC) with OpenRTM technology, RTC can be ran in different operation systems (OS) and in different computers, and these RTCs can communicate with each other freely as long as these RTCs are in the same local network.

2.2 Communication Frame

For our surgical robot control system, it has two sub-systems, which is master system and slave system. The communication frame between these two sub-systems is illustrated bellow.

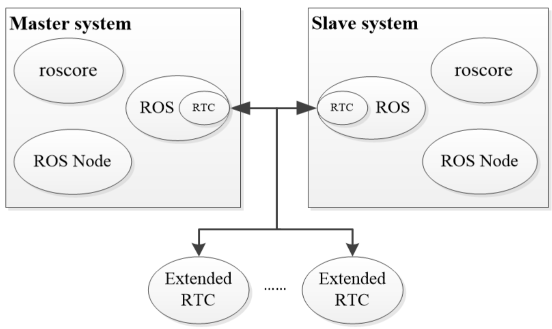

Fig. 2 Communication frame between master system and slave system

We use OpenRTM in a ROS node code, so in one aspect, this is a RTC which can communicate with other RTC in other system, in the other hand, this is also a ROS node, and it can communicate with other ROS nodes via ROS topic or ROS service. With this communication frame, each sub-system is a complete ROS system, so this sub-system will not rely on the other sub-system, this increase the safety of the whole system. Each sub-system is also a RTC, which enables communication between these sub-systems. Move over, the surgical control system is very easy to extend with OpenRTM framework, we can add a series of new component as long as it is packaged as RTC.

There are two kinds of data that transfer between these two systems, one is command data and another is status data. Command data are important relatively, so these data are transferred using service port so that we can make sure the data are transferred successfully. In addition, status data are transferred using data port to increase the efficiency of communication.

2.3 Architecture of Master System and Slave System

We designed a kind of four-layer control architecture for master and slave system, which are hardware layer, driver layer, control layer and RTC communication layer. Hardware layer is consists of different hardware, and driver layer is formed by the ROS nodes that dealing with corresponding hardware. Control layer is formed by the ROS nodes that implements main control logic, at last, the RTC network layer is consists of the ROS nodes that dealing with communications with another system.

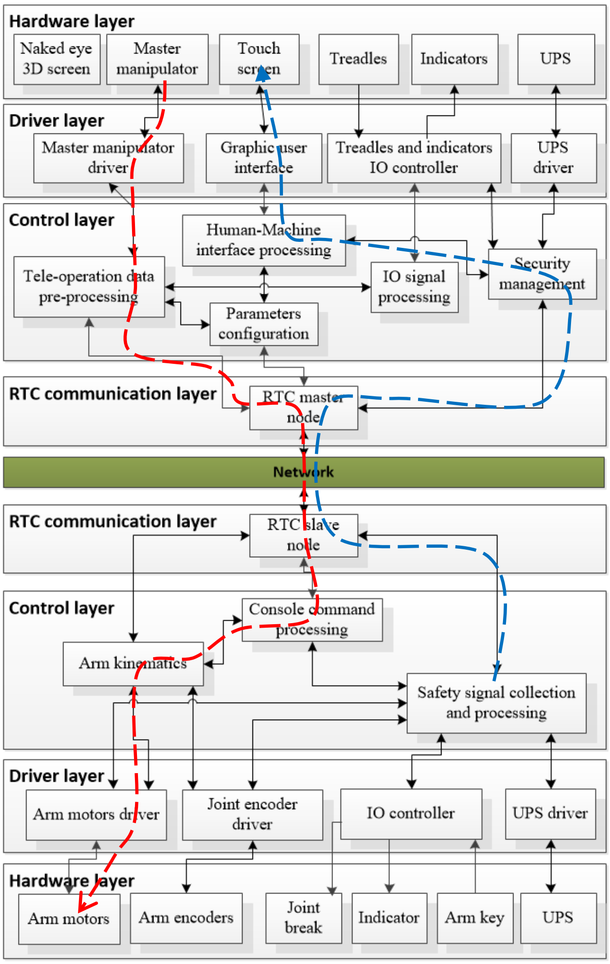

Fig. 3 Architecture of Master System and Slave System

Whole architecture diagram is illustrated in figure 3, the upper part is master system, and the lower part is slave system. The black arrows in the figure 3 indicate the information flow. A typical information flow is tele-operation control flow as red dash arrow indicated in figure 3, which begins from master manipulator which is Omega-7, then a ROS driver node sample the data from Omega-7, and sends these data to tele-operation data pre-processing node which will processing the raw data, and then processed data will send to RTC master node which is a ROS node dealing with communication with slave system, in this node, data will transfer to RTC slave node in slave system using OpenRTM techniques. Finally, data will send to arm motors through console command processing node, arm kinematics node and arm motor driver node. Type of data will change all the way down, and finally this information will drive the motors to desired position or velocity. Another important information flow is safety information collection and processing, which indicated in figure 3 with blue dash arrow. All essential information of slave system will come together to safety signal collection and processing node, and this node will send the emergency information to security management node in master system, then these information will send to human-machine interface processing node and will display on touch screen, on which a popup window will inform and ask doctor to react to these information.

Acknowledgments

This work is supported by the National Natural Science Foundation of China. (Grant No.61273331)

References

[1] G. D. Hong, T. Nakamura and K. G. Kim, “Article Title,” Journal Title, Vol. 10, No. 6, pp. 1-10, 2006.

[2] A. C. Smith, Introduction to Robotics, Publisher, New York, 2006.

[1] I. Broeders, J. Ruurda, “Robotics revolutionizing surgery: the Intuitive Surgical ‘Da Vinci’ system”, Industrial Robot: An International Journal, Vol. 28 Iss 5 pp. 387 – 392, 2001

[2] Marescaux J, Rubino F. “The ZEUS robotic system: experimental and clinical applications.” , Surgical Clinics of North America, Vol. 83, No. 6, pp. 1305-1315, 2003

[3] Hannaford B, Rosen J, Friedman D W, et al. “Raven-II: an open platform for surgical robotics research”, Biomedical Engineering, IEEE Transactions on, Vol. 60, No. 4, pp. 954-959, 2013

[4] King H, Kosari S N, Hannaford B, et al. “Kinematic analysis of the Raven-II research surgical robot platform”, University of Washington Electrical Engineering Department Technical Report, 2012, 2012: 0006.

[5] Kazanzidesf P, Chen Z, Deguet A, et al. “An open-source research kit for the da Vinci® Surgical System”, Robotics and Automation (ICRA), 2014 IEEE International Conference on. IEEE, 2014: 6434-6439.