State Monitoring for Surgical Robot System Based on Robot Operating System

Haoruo Zhang
Shanghai Jiao Tong University
zhr@sjtu.edu.cn

Qixin Cao
Shanghai Jiao Tong University
qxcao@sjtu.edu.cn

Weibang Bai
Shanghai Jiao Tong University
bwb@sjtu.edu.cn

Shuanghai Dong
Shanghai First People's Hospital affiliated to Shanghai Jiao Tong University
dale8188@163.com

1. Introduction

Nowadays, the minimal invasive surgery (MIS) is increasingly being applied to laparoscopic surgery. There are numerous surgical robot system available for MIS, such as da Vinci and Zeus. And they become very popular among patient [1]. However, these surgical robot systems are too expensive for using in most general hospital. Certainly, surgical robot system have many advantages such as convenience, accuracy and stability, and it will play the important role in MIS. So we develop a new surgical robot system which has proprietary intellectual property rights. As shown in Figure 1, the surgical robot system consists of main structure, control platform and vision platform [2]. The control system of this surgical robot is based on Robot Operating System. In order to improve the security capability of our surgical robot system, monitoring the working state of each module is essential.

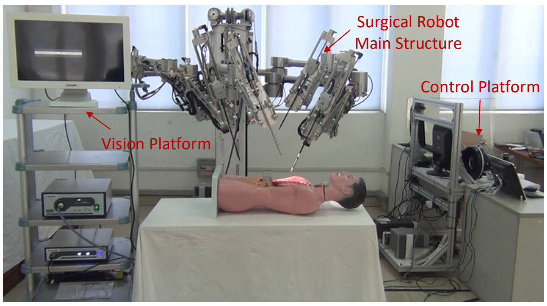

Fig. 1. Surgical Robot System

In this paper, we propose the state monitoring for our surgical robot system based on Robot Operating System (ROS). And it is a crucial part of the whole system. So far, the surgical robot system still need many modules to achieve operating function. Because it is complex system, errors may occur in different modules. In order to ensure the normal operation of surgical robot system, the function of state monitoring is necessary. Due to the complex system, we will utilize the ROS which is based on the modularization thinking to build software platform of surgical robot system. In Section 2, we briefly introduce ROS technologic and its modularization thinking. Then we will present the key points of state monitoring for our surgical robot system based on ROS in Section 3. Our conclusions and some ideas for future research on state monitoring are given in Section 4.

2. ROS Technology

The Robot Operating System (ROS) is a flexible framework for writing robot software. It is a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust robot behavior across a wide variety of robotic platforms [3]. In ROS, a whole software is consist of a master node which called roscore and some other functional nodes. Different nodes will perform different threads independently. Therefore, it is very simple to build distributed control system based on ROS. As we known, a communication system is often one of the first needs to arise when implementing a new robot application. The messaging system in ROS saves our time by managing the details of communication between distributed nodes via the anonymous publish/subscribe mechanism and topics are named buses over which nodes exchange messages. In addition, because many-to-many one-way transport of publish/subscribe model is not appropriate for RPC request/reply interactions. So, request/reply is done via a Service, which is defined by a pair of messages: one for the request and one for the reply in ROS. The brief communication graph is shown in Figure 2.

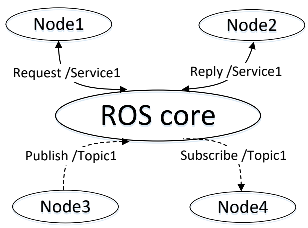

Fig. 2. Communication Graph in ROS

3. State Monitoring Mechanism

As shown in Figure 3, the control system of surgical robot is consist of slave system and master system. In master system, state monitoring main node is the main

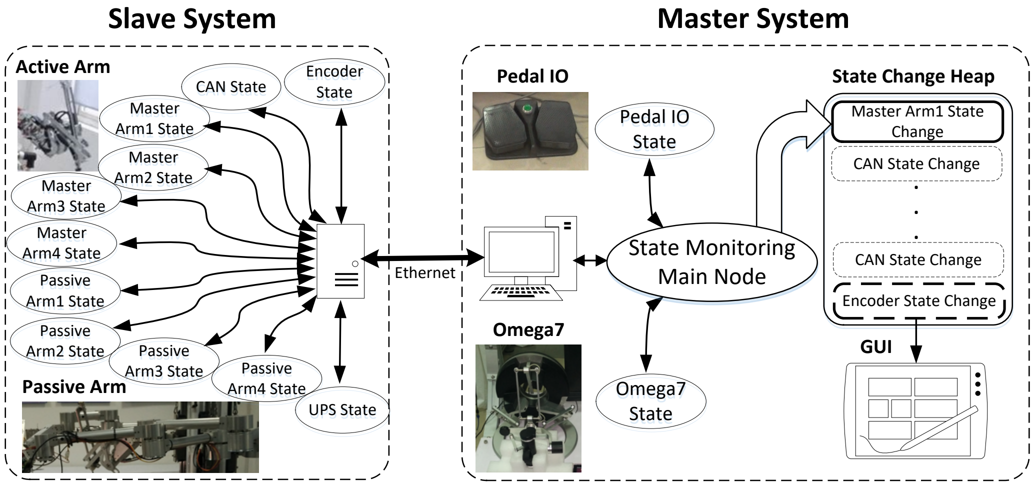

Fig. 3. State Monitoring Mechanism based on ROS

program of state monitoring mechanism. And this node has two function, including Self-checking Initialization and Component State Monitoring. Above all self-checking initialization is very important to the whole system [4].

After the system starts, State Monitoring Main Node will request the reply of each node through service interactions in ROS. As long as the reply is TRUE, the node could be considered as activated. And only if all the replies are TRUE, the Self-checking Initialization would be done. Once a FALSE reply comes out, State Monitoring Main Node would push the FALSE message into the State Change Heap. According to Figure 3, State Change Heap contains the message about the change of each node.

As for the second function of State Monitoring Main Node, the aim of it is to monitor the change of node state. Among the whole control system, there are 11 nodes in slave system and 2 nodes in master system, including encoder node, CAN node, master arm1 node, master arm2 node, master arm3 node, master arm4 node, passive arm1 node, passive arm2 node, passive arm3 node, passive arm4 node, UPS node, pedal IO node and Omega7 node. When an error occurs in certain node, the node would publish a message. In addition, the number of module state in surgical robot control system may be more than two [5]. So, this message records state change, and it could be warning or error. In slave system, those messages will be received by the slave roscore. Meanwhile, the slave roscore corresponds with the master roscore based on Ethernet. In this way, the message in slave system could be subscribed by State Monitoring Main Node. In master system, the messages of pedal IO node and Omega7 node are directly subscribed. Due to response frequency of GUI is about 5Hz, messages of state change cannot be received simultaneously. Therefore, it is necessary to set up a heap to process this data stream. As long as State Monitoring Main Node has received messages from several nodes, it would push these messages into the heap. Then, State Change Heap will publish messages at 5Hz frequency until all messages have been sent out. GUI module will receive these messages and refresh the display. So, if there is something wrong in the control system, users can get the anomaly message from GUI right away.

4. Conclusion and Future Work

It is clearly important to monitor each module in the surgical robot control system. The state monitoring mechanism is based on Robot Operation System which is a flexible framework of writing robot software. And there are two novel points in this paper. The first one is that every module in control system is encapsulated as distributed node in ROS, so as to it can be controlled and monitored individually. And utilizing state monitoring heap is the second one, which can solve the problem caused by response frequency of GUI.

So far, the function of the state monitoring is not enough. It is also important that a function about exception handling is included. When certain error occurs, the system can not only publish this message to the GUI but also take some measures to fix this error. And we will integrate it into the whole system in the future.

Acknowledgement This work was supported by Cross Research Fund of Biomedical Engineering of Shanghai Jiao Tong University (No.YG2013MS08).

References

[1] Giulianotti, Pier Cristoforo, et al. "Robotics in general surgery: personal experience in a large community hospital." Archives of surgery 138.7 (2003): 777-784.

[2] Goldberg, Randal P., et al. "Ergonomic surgeon control console in robotic surgical systems." U.S. Patent No. 8,120,301. 21 Feb. 2012.

[3] Robot Operating System available at http://www.ros.org/.

[4] Noh, Jong Ho, Dong Hyun Yun, and Jong Ho Park. "Development of the control system for stage sets equipped with the self-checking and counteracting functions." International Journal of Precision Engineering and Manufacturing 16.12 (2015): 2451-2458.

[5] Woodall, William H., Sandy L. Fogel, and Stefan H. Steiner. "The monitoring and improvement of surgical-outcome quality." Journal of Quality Technology 47.4 (2015): 383.