Proceedings of the 2017 Design of Medical Devices Conference
DMD2017
April 10-13, 2017, Minneapolis, Minnesota, USA
DMD2017-3347

Design of a robotic catheterization platform with use of commercial ablation catheter

Zhenkai Hu[1]
Jongseok Won[2]
Youngjin Moon[1,3]
Sanghoon Park[1]
Jaesoon Choi[1,3]

Biomedical Engineering Research Center, Asan Medical Center[1]
Graduate School of Convergence Science & Technology, Seoul National University[2]
College of Medicine, University of Ulsan[3]

1 Background

Cardiovascular diseases including atherosclerosis, thrombosis, aneurysm and arrhythmia remain the major cause of mortality in developed countries, accounting for 34% of deaths each year [1]. Commonly used minimally invasive vascular intervention with using catheters leads to higher success rate than open surgery [2].

Integrating robotic technologies into active control of catheters in teleoperation manner has promised to reduce radiation exposure to surgeons and improve accuracy during electro-physiological (EP) procedures [1]. Common used commercial robotic EP catheter platforms such as Sensei (Hansen Medical Inc., USA) and Niobe (Stereotaxis Inc., USA) are usually composed of a catheter driver (slave side) which can be remotely controlled by a console operator (master side). However, the Sensei catheters are more rigid and bigger than standard catheters because of their two-layer-sheath structure; and Magnetic Niobe systems are huge and expensive.

In this paper, we propose a mechanism of remote-driving catheterization platforms in which a commercial tip-steerable ablation catheter (St. Jude Medical Inc., USA) (Fig. 1) is manipulated by a catheter driver in three degree of freedoms (DOF) (insertion/withdrawal, rotation and tip deflection). In addition, we also present the design of the control software based on Object-Oriented Programming (OOP) method which is expected to give the other researchers a guide line during robotic catheter design.

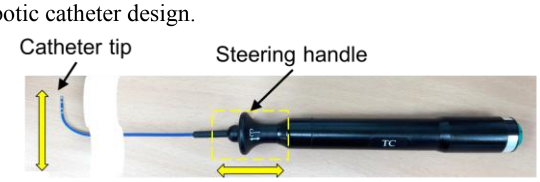

Fig.1. Tendon-driven steerable catheter. Catheter tip can bend in opposite directions by dragging the steering handle in respective directions

2 Methods

The catheterization platform is composed of a master console and a catheter driving system (slave) as shown in Fig. 2. In the master console, the operator can both send command to and get running status from the catheter driver through a GUI programmed in Microsoft Windows on a master PC. Another slave PC connecting with the master PC via Ethernet is used as controlling processor for the catheter driver. The catheter driver is actuated by three EC motors (MaxonMotor Inc., Switzerland) which are connected with the slave PC via controller area network (CAN).

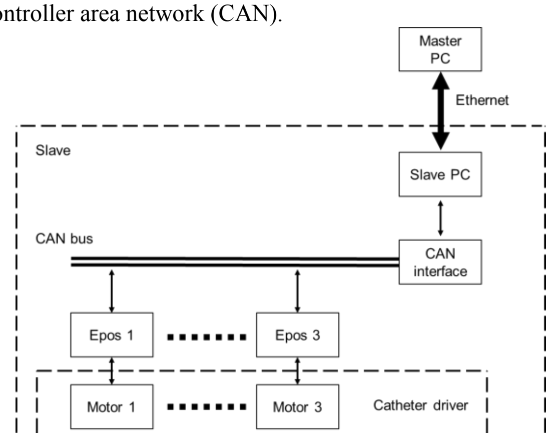

Fig.2. Structure of robotic catheterization platform

A slot where the steerable catheter (Fig. 1) can be mounted is on the catheter driver mechanism as shown in Fig. 3. For insertion/withdrawal motion, rotation of motor_1 shaft leads to sliding ball screw_1 to which positions of the motor_2, the motor_3, and the catheter handle are fixed. Motor_2 can rotate a catheter handle which is fixed on the shaft of motor_2 around axis X. Sliding of ball screw_2 actuated by motor_3 can bend catheter tip in 1 DOF. The opposed rolling belts are used to support the flexible catheter body in order to avoid catheter's buckling.

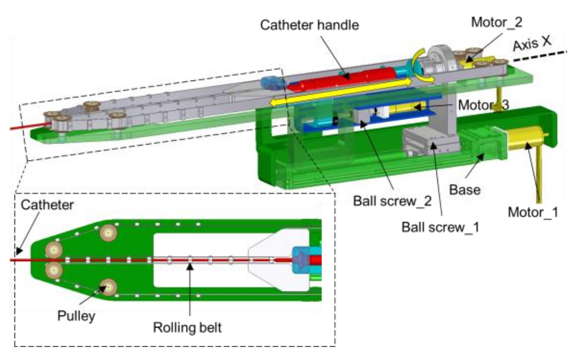

Fig.3. Mechanism of catheter driver

Considering functional extensibility of the catheter driver system, module based software is designed. As our present work, high-level module (catheter driver module) is composed of mid-level modules including motor control module, Ethernet module and kinematics module. The motor module is composed of low-level modules including CAN driver modules. Both the motor module and the CAN driver modules are hardware dependent while the catheter driver module is hardware independent. OOP method is used to design classes for the modules.

Figure 4 illustrates the software structure designed for the slave catheter driver control. Gray arrow in Fig. 4(a) shows relationship of the classes that the upper classes have variable members that are defined by low classes. CAN driver module includes classes CAN message, Object Dictionary, CAN driver and CAN message control that were designed for CAN communications based on CANopen protocol and library provided by CAN device hardware supplier. CAN message control can both send and receive a CAN message object. Derived classes (like CNmtMasterMsg etc., Fig. 5(b)) from CAN message were designed based on particular applications including network management (NMT) communication, service data objects (SDO) communication, process data object (PDO) communication, emergency message warning and synchronization. The class Maxon motor was designed to cover all the functions of the motor controller (EPOS, MaxonMotor Inc., Switzerland), to make Catheter drive codes hardware-independent. Since different Maxon motor products need different settings, derived classes from class Maxon motor were designed separately (Fig. 4(c)). The class Kinematics was designed for calculating translation from encoder position values to handle location of the master manipulator. Class EtherNet was designed for communication between the master and slave.

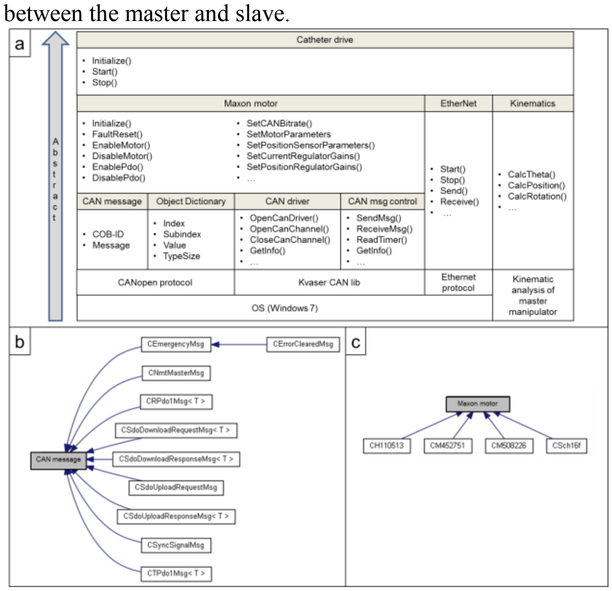

Fig.4. Software structure of catheter driver control

3 Results

Figure 5 shows the master GUI (Fig. 5(a)) and prototype of the catheter driving system (Fig. 5(b)). An operator set demanding position values to the catheter driver through configuring sliding bar or edit controls and then observes running status and actual positions of the catheter driver in the master GUI. The catheter driving system was mounted on a robot arm for initial positioning of catheter device.

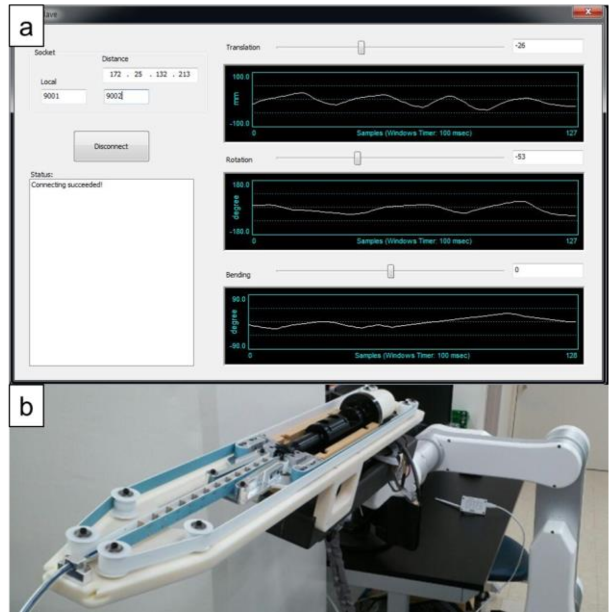

Fig. 5. Prototype of catheterization platform. a) Master GUI; b) Catheter driving system

4 Interpretation

This paper introduced a design of robotic catheterization platforms to control a tendon-driven steerable catheter with three DOF. Master-Slave structure was utilized to promise reducing radiation exposure of the operator and improve accuracy during endovascular procedures.

The software was designed based on considering both extensibility and feasibility of the platform. For example, new functional modules such as haptic sensing and image mapping can be designed separately from the existing codes and added into the catheter driver module easily. Another example is that update of a special hardware (e.g. CAN controller) only need to rewrite codes of the corresponding module but leave its interfaces and the higher level modules unchanged.

Acknowledgment

This research was supported by a grant of the Korea Health Technology R&D Project through the Korea Health Industry Development Institute (KHIDI), funded by the Ministry of Health & Welfare, Republic of Korea (grant number: HI14C0517).

References

[1] Rafii-Tari, H., Payne, C.J. & Yang G.Z. (2014). Current and Emerging Robot-Assisted Endovascular Catheterization Technologies: A Review. Ann. Biomed. Eng., Vol. 42, No. 4, 697-715

[2] Hausegger, K.A., Schedlbauer, P., Deutschmann, H.A. & Tiesenhausen, K. (2001). Complications in endoluminal repair of abdominal aortic aneurysms. Eur. J. Radiol., 39:22–33

