ELSEVIER

Reliability Engineering and System Safety 73 (2001) 183–192

RELIABILITY ENGINEERING & SYSTEM SAFETY

www.elsevier.com/locate/ress

The safety issues of medical robotics

Baowei Fei[a,1], Wan Sing Ng[a,*], Sunita Chauhan[a], Chee Keong Kwoh[b]

[a]*School of Mechanical and Production Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798*
[b]*School of Computer Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798*

Received 22 June 2000; accepted 10 April 2001

Abstract

In this paper, we put forward a systematic method to analyze, control and evaluate the safety issues of medical robotics. We created a safety model that consists of three axes to analyze safety factors. Software and hardware are the two material axes. The third axis is the policy that controls all phases of design, production, testing and application of the robot system. The policy was defined as hazard identification and safety insurance control (HISIC) that includes seven principles: definitions and requirements, hazard identification, safety insurance control, safety critical limits, monitoring and control, verification and validation, system log and documentation.

HISIC was implemented in the development of a robot for urological applications that was known as URObot. The URObot is a universal robot with different modules adaptable for 3D ultrasound image-guided interstitial laser coagulation, radiation seed implantation, laser resection, and electrical resection of the prostate. Safety was always the key issue in the building of the robot. The HISIC strategies were adopted for safety enhancement in mechanical, electrical and software design. The initial test on URObot showed that HISIC had the potential ability to improve the safety of the system. Further safety experiments are being conducted in our laboratory.

Keywords: Medical robotics; Safety-critical system; Image-guided surgery; Transurethral resection of the prostate; Interstitial laser coagulation; Radiation seed implantation; Laser resection; Electrical resection; Benign prostate hyperplasia and Prostate cancer

1. Introduction

Medical robots have played an increasingly important role in the recent decade. Applications include neurosurgery [1–3], orthopedics [4–7], urology [8–10], laparoscopic surgery [11–13] and others [14–16]. Detailed reviews were reported in the literature [17,18]. Commercial products include ROBODOC [19], AESOP [20], CASPAR [21], da Vinci [22] and others [17,18].

Safety is one of the key issues in designing a medical robot but traditional safety methods for industrial robot are not enough for medical robots [27]. For example, safety requirements for industrial robots suggest that they should be isolated in a cell with safety interlocks to prevent people from directly interacting with the robot. However, surgical robots require direct contact with the patient and surgeon. Isolation will limit their capability and applications [27]. Medical and industrial robots have other huge differences such as operating targets and working environments. Medical robots are used in medical settings for patients and concern human life. Industrial robots are operated in factories for products. The safety issues of medical robots are more stringent, dedicated and critical.

Mechanical safety is the basic requirement for a robot. Earlier enhancement methods included redundancy sensor, mechanical constraints and fault detection [23–27]. Davies [27] put forward the concept of a long no hazard period for safety analysis of medical robots. Ng et al. [28] reported mechanical safety enhancement methods for a surgical robot. Radermacher and co-workers [29,30] used ergonomics analysis method for man–machine interface design. Maciejewski et al. [31,32] put forward methods to analyze joints failures. Ikuta et al. [33] used impact force and impact stress as the safety value for human-care robots.

Software safety received more attention with the intensive use of computers, integrated circuits and advanced functionality in robots. Safety approaches included fault tree analysis (FTA) [34,35], even tree analysis [35], fault tolerance algorithm [36], dependability principles [37], and others [38,39]. Valey [40] reported practical techniques for software development for an endoscopic camera manipulator.

There are currently no specific standard safety guidelines

* Corresponding author.
E-mail address: mwsng@ntu.edu.sg (W.S. Ng).
[1] Current address: Department of Biomedical Engineering of Case Western Reserve University, 10900 Euclid Avenue, Cleveland, OH 44106, USA..

0951-8320/01/$ - see front matter
PII: S0951-8320(01)00037-0

for medical robots. However, several standards can be used as references. EN 755 (ISO 10218) is a standard safety guideline for industrial robots [41]. The FDA (Food and Drug Administration of the USA) provided a safety standard for computer controlled medical devices [42]. IEC 1508 is the standard for a safety-related system [43]. With increasing applications of medical robots, more specific systematic safety guidelines are necessary. There are no reports on systematic approaches on the safety issues of medical robots.

In this paper, we will propose a systematic methodology named hazard identification and safety insurance control (HISIC) to analyze, control and evaluate the safety of medical robots. In Section 2, we will first introduce a useful model to analyze safety factors and potential hazards, and then we will explain the HISIC principles in detail. In Section 3, we will use an example robot named URObot to explain the practical considerations in both software and hardware. At the end, we will discuss other safety methods.

2. Hazard identification and safety insurance control

2.1. HISIC model

Many factors can cause safety problems for medical robots. Human error is one aspect, such as wrong instructions or manipulation. System error is another important aspect, which can also be divided into four categories: pure hardware, pure software, hardware triggered by software, and software triggered by hardware.

Any robot system should have basic safety requirements. First, hardware should be proved safe when it stands alone or when no software controls it. Second, software must run and test correctly during simulation or when no hardware is involved.

We put forward an HISIC model to study the safety issue. Overall safety index SF is defined as:

$$\mathrm{SF} = f(\mathrm{SW(PL)}, \mathrm{HW(PL)}),$$

where SW is the software factor, HW the hardware factor, and PL the policy factor. PL is not a material factor, but it does through all phases of the medical robot.

We believe that policies play a key role in safety controls because they regularize the whole lifecycle including research, design, production, test, and end use. We defined the HISIC principles to identify and control safety factors from physical, medical and chemical aspects.

2.2. HISIC principles

HISIC includes seven core principles for systematically analyzing and controlling safety issues. Ideally, every medical robot project should have an HISIC Team that is responsible for developing, implementing and maintaining HISIC system. The team should include designers, surgeons, physicians, patients and administration staff.

2.2.1. Definitions and requirements (principle 1)

At the start, the HISIC team should create detailed descriptions on functionality, requirements and working environments of the medical system. These descriptions form the basis for detailed requirements, specifications and design methodology of the whole system including mechanical, electrical, software, and safety sub-systems as well as an independent test system. For example, software should be developed according to a properly documented methodology. The FDA requirements for computer controlled medical device imply that the traditional 'top–down' methodology is preferred [42]. At the end, detailed instructions for operation, management and maintenance are put together.

2.2.2. Hazard identification (HI) (principle 2)

Being compatible with IEC601 [44,45], any hazard discussed here is a potentially detrimental effect, arising directly from a medical robot, on the patient, other persons, or surroundings. HI is the process of collecting and evaluating information on hazards associated with the robot under consideration in order to decide significant issues that must be addressed in the HISIC plan. The purpose of the HI is to develop a list of hazards that are reasonably likely to cause harm, injury or death if not effectively controlled.

It is important to consider medical, chemical and physical factors, such as mechanics, electrical controlling components, software, transportation, sterilization and operation. Mechanical injury may be due to shock that might cause internal bleeding and/or bone fracture, and scar that leads to bleeding. Electrical injury may include electrical shock that may cause burn or death, fire, electromagnetic radiation that may lead to induction of certain cancer, leukemia and interference. Software hazards may lead to the robot being out of control, system crashes or malfunctions. Not only the hazards derived directly from the system itself should be identified, but also those related with patient movement and surgeon operation should be inspected.

Thorough HI is the key to prepare an effective HISIC plan. If the HI is not done correctly or the hazards are not identified, the HISIC plan will not be effective regardless of how well it is followed. The HI has three goals. First, hazards and associated control measures are identified. Second, the process may identify required modifications to product so that the safety is further improved. Third, the analysis provides a basis for safety insurance control (SIC) in principle 3.

2.2.3. Safety insurance control (principle 3)

SIC is a control step to eliminate a hazard or to reduce it to an acceptable level. The tasks of SIC are to specify detailed guidelines for implementation and to set up a monitoring mechanism. SIC consists of many critical control

points that are important for safety. These points may be located at any sub-system such as a joint. The information collected in HI is essential to SIC. Complete SIC is very significant for safety control. The SIC plan must be carefully developed and documented.

Typical examples are discussed. Quantitative mechanical data of a joint should be specified to prevent mechanical failure, and loading tests shall be conducted after design. The positive and negative switches shall be set to prevent motors from running out of travel limits. Redundancy sensors shall be installed to monitor motor position. Software quality metrics such as flow charts, fault trees, even trees and statistical methods shall be applied to measure the reliability of modules and whole software.

2.2.4. Safety critical limits (SCL) (principle 4)

SCL are specific ranges to limit safety parameters for mechanical, electrical and software systems. A critical limit distinguishes a parameter as within a safe or unsafe range. Each SIC will have one or more control measures to assure that the identified hazards are eliminated or reduced to acceptable levels. Each control measure has one or more associated critical limits. SCL data shall have scientific and experimental basis for safety control. Quantitative data of mechanical and electrical parts, such as the force, dynamic constraints, velocity, current, voltage, power, must be limited to a safety range. For example, a gantry should have the ability to support a possible maximum load.

2.2.5. Monitoring and control (MC) (principle 5)

Monitoring is a sequence of observations and measurements for the assessment of SIC implementation. It also produces an accurate record for future verification. Control is to manage the condition of an operation to comply with established criteria. Control also takes actions to correct deviation.

MC has two important concepts. The implementation of HISIC plan itself needs MC and the HISIC team should be well managed. The robot itself is an advance self-monitoring and controlling system. It shall have the ability of collecting data, comparing to criteria and correcting deviation. The information obtained in MC can also be used for verification in principle 6.

The tasks of control are to specify what is done when a deviation occurs, to determine and correct the cause of non-compliance, and to record the correct actions. Parameters to be monitored include motor position, working envelope, force, voltage, current, software run path, and many others. The system for MC should be independent of the working system to increase safety. The control system should be a closed loop. A redundant computer, parallel processor, or microprocessor system can be used.

2.2.6. Verification and validation (VV) (principle 6)

Validation ensures the end product is correct as compared to its requirements. It comprises the evaluation of scientific and technical information to determine if the HISIC plan, when properly implemented, will effectively control the hazards.

Verification ensures a correct product in its developing phase and sub-phase. It determines whether a product is correct, complete and consistent with itself and predecessor. The activities ensure the validity of HISIC plan and correct operating of the plan.

VV should be applied to the HISIC plan itself, system requirements, software, mechanical, and electrical systems, and applications including transportation, maintenance and operation. VV can improve the correctness, reliability, quality and therefore safety of medical robots.

2.2.7. System log and documentation (principle 7)

The system log is used to record system status and actions. Typical statuses include system initial parameters and states of emergency button, locks, gantry, key button, and laser switches. Typical actions include robot homing, gantry adjustment, motor movement, locks switching, laser turning on, emergency button pressed and any other actions generated by robot. The system log traces and watches the robot. It provides very useful data to analyze faulty action.

Documenting the designs, tests and operations of a medical robot is important for safety. It is the designer's responsibility to achieve safety requirements such as accuracy, repeatability and reliability. It is a critical requirement that a tester tests all criteria in extreme conditions, simulates applications many times and provides feedback to designer. The tester shall give suggestions to users in an operation manual. The user shall operate the system according to the manual. It is the user's responsibility to correctly operate the system and design an appropriate plan. Documentation helps to prevent human error and demarcates responsibility.

The HISIC principles are important for safety enhancement of medical robots. We next discuss the safety issues of an example robot.

3. URObot

3.1. System overview

A robot for urological application was developed in the Computer Integrated Medical Intervention Laboratory (CIMIL). The system, known as URObot, has a history traced back to 1989 when a laboratory prototype was built for robotic transurethral resection of the prostate (TURP) in Imperial College, London. Ng, one of the authors, was among the pioneer development team in Imperial College (1989–1992) [8,9]. From September 1995 to December 1998, a commercial prototype for robotic electrical-TURP, named surgical programmable urology device (SPUD), was produced by Dornier Asia Medical System Pte. Ltd in collaboration with CIMIL. VV was conducted before SPUD commenced clinical trials in Changi General Hospital

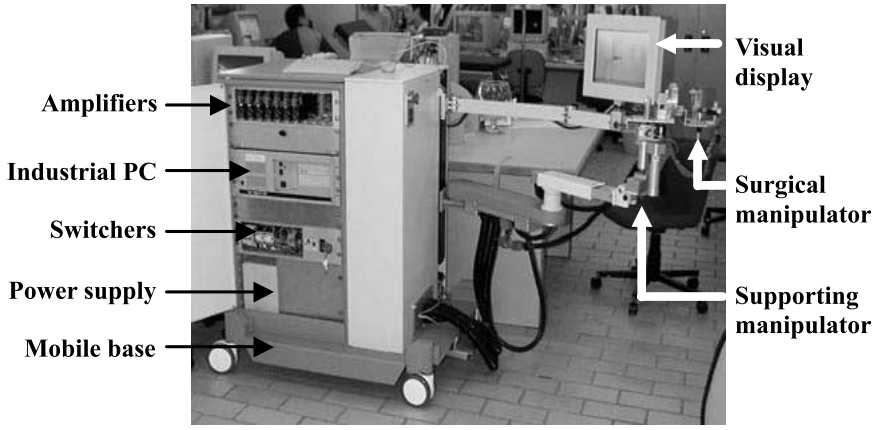

Fig. 1. The URObot system with ILC module. The top-right is an 800 × 600 LCD covered by a touch screen. Immediate below is the surgical manipulator where four motors control four motion freedoms. The supporting manipulator can change positions vertically and horizontally. The big cabinet on the left is the main body of the system that includes amplifiers, an industrial PC, the motion controller, power supply and electronic circuits. The whole system with four wheels can move and be fixed.

(Singapore) in 1998. We are developing the second-generation robot that provides a general platform for interstitial laser coagulation (ILC), laser resection (LR), radiation seed implantation (RSI) and electrical resection (ER).

Fig. 1 shows the URObot system with ILC module. The visual display unit is an 800 × 600-pixel LCD for ultrasound image display and user interface (UI). The surgical manipulator consists of four motors that control four-freedom motions. The manipulator can hold a laser fiber, ultrasound probe or other surgical tools. The electrical unit includes an industrial PC, a motion controller, amplifiers and electronic circuits. The robot has a supporting manipulator, a cabinet and a mobile base.

We now briefly describe the ultrasound image-guided robot assisted surgery. The first step is the surgical preparations that are the same as the conventional TURP operation. A catheter is inserted into the urethra under the guidance of endoscope. The second step is ultrasound image acquisitions and processing (IAP). A transurethral ultrasound probe is mounted on the surgical manipulator. The robot drives the probe into the urethra step-by-step. At the same time, series of 2D images are acquired slice-by-slice covering the whole prostate from verumontanum to bladder neck. The surgeon then uses graphical tools to semi-automatically outline the prostate boundary in each image and the 3D prostate is automatically reconstructed. The third step is treatment planning and surgery. Since the ultrasound probe is mounted on the robot manipulator, the prostate position derived from images is transformed to robotic coordinate system. The surgeon plans the treatment and robot generates corresponding motion sequences. Many calibration experiments were conducted on phantoms before clinical trial.

This paper focuses on the safety issues and HISIC implementation in URObot. Partial issues mainly on hardware were addressed in Ref. [28]. We will further discuss some safety considerations on software and hardware. More details were reported in Refs. [10,46–49].

3.2. Software considerations

3.2.1. Software design methodology

Fig. 2 shows the software design method for URObot. The entire development consisted of eight phases: system requirements, software requirements, module design, program development and test, module test, software integration and test, and system integration and test. It was a top–down design method that complies with the FDA preference. Safety check and VV were implemented in the whole cycle at each phase. The development of the whole system and its sub-systems were always in a closed loop where an independent test improved the safety. These special features are important for software safety, reliability and quality.

The method was the result of HISIC application in software design. The seven principles were adopted in each phase and sub-phase of the software development, such as requirements, safety check, verification, validation, and test.

3.2.2. Software layers

The URObot software system consisted of five main layers as shown in Fig. 3. The first layer is the graphics user interface (GUI), the interface between surgeon and robot. The second layer includes ultrasound image acquisition, boundary detection and 3D modeling of the prostate. The third layer includes the treatment planning, robot controlling software (RCS) and robot controller interface (RCI). RCI is the interface between RCS and controller. RCI also regularly checks the status of the peripherals such as controller circuits, motors, amplifiers and mechanical switches. The fourth layer is the controller driver that is responsible for the communication between the industrial

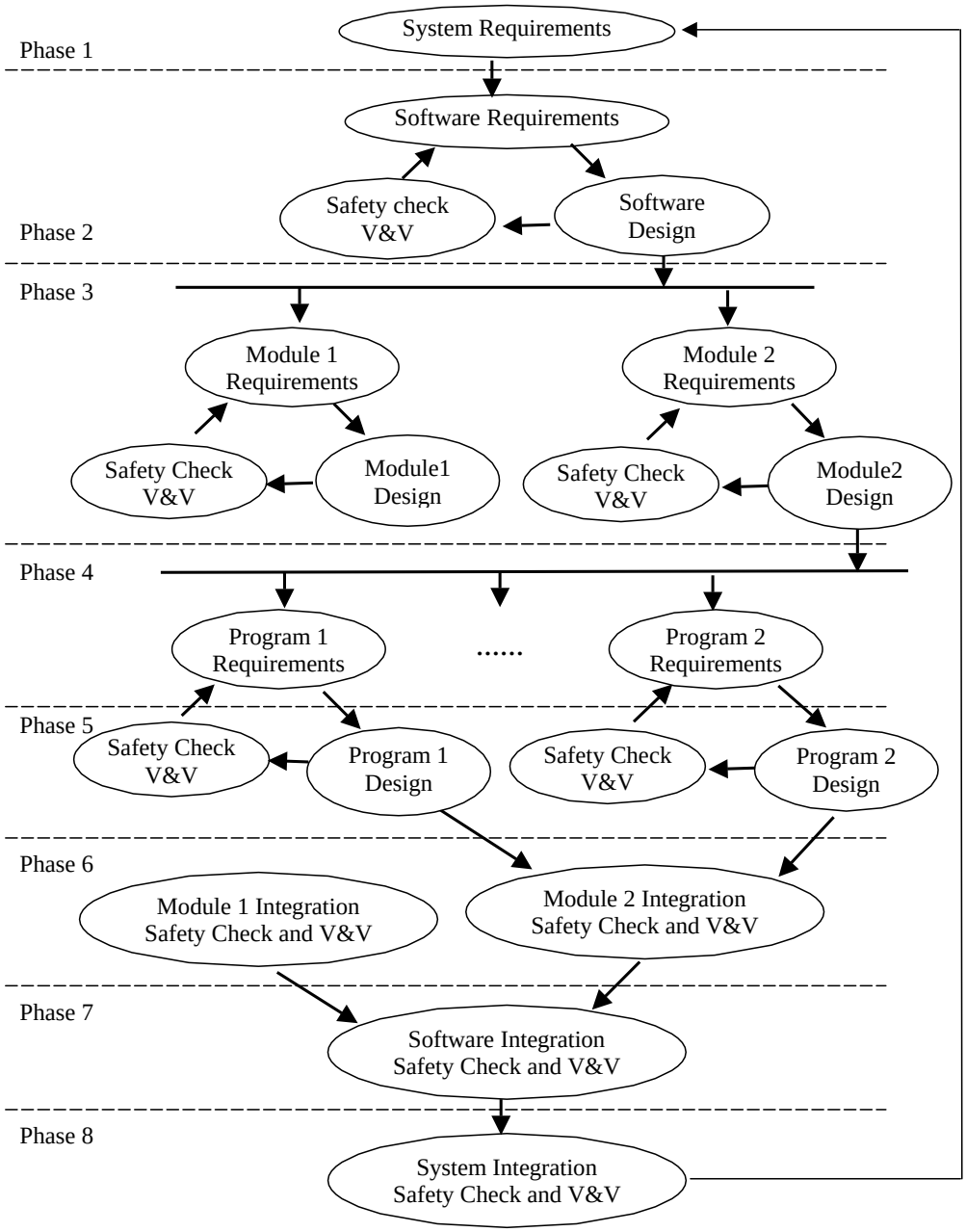

Fig. 2. Software design method. The framework is a top–down approach. Horizontal dash lines separate the whole flow chart into eight development phases. The top is the system requirements design. The bottom is the system integration and test. Requirement, design and test are the three basic steps that form a close loop to ensure safety. This concept is implemented in each phase. Safety check and VV are always parts of the development recycle.

personal computer (IPC) and the controller, programmable multi-axis controller (PMAC, The Delta Tau Data System, USA).

The layers concepts improved the quality and safety of the software. Different layers are responsible for different tasks. Data encapsulation and access privilege of each layer provided independence and security between modules.

3.2.3. Software reuse

Software reuse was applied to URObot and it improved the software usability and reliability [50,51]. The purposeful creation, management, support, and reuse of software modules were implemented (Fig. 4). URObot provides a universal platform for ILC, RSI, LR and ER. The four modules have many common parts such as IAP, GUI, RCS and RCI. We created very general classes for these functionalities. These classes were carefully designed and well tested according to the HISIC plan. Different applications can directly use or inherit them for their own purposes.

There are several implementation details. We used standard ANSI C++ as the programming language since

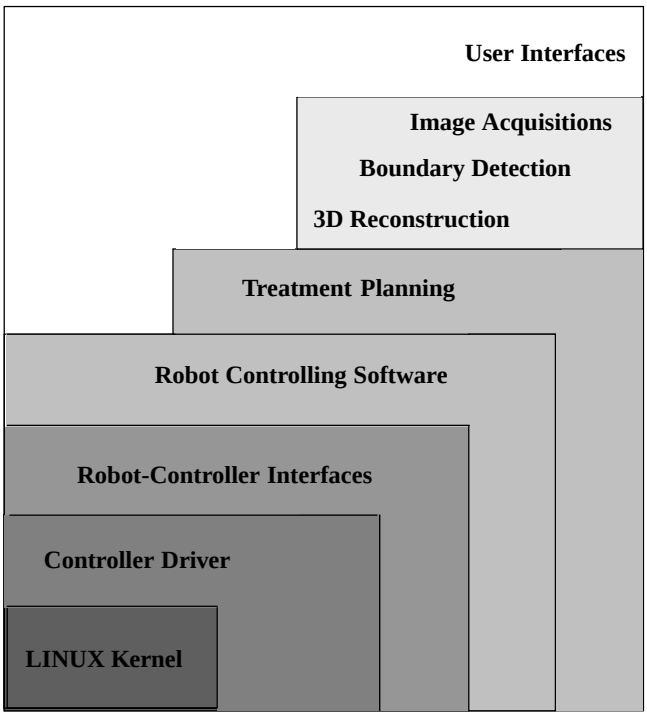

Fig. 3. Software layers. From top to bottom, the software levels become more and more low. The highest level is the GUI. The lowest is the controller driver that is a part of the LINUX kernel. If two layers are neighbors, they can exchange data. Otherwise, no access is allowed for software safety.

it can be compiled and run in multiple platforms. We use fast light tool kit (FLTK) as the GUI development tool. Redhat (Red Hat) Linux 6.1 was chosen as the operating system for its high reliability.

3.2.4. User interface

There are several reasons why UI is important for medical robots. First, a medical robot is usually a complicated system that may include precise mechanical parts, advanced electronic circuits, automatic control system and sophisticated computer software. Second, the users of medical robots may not have expertise in engineering especially computer. Third, medical robots have special requirements on operations and working environments.

General UI considerations are discussed. URObot directly contacts the patient. A surgeon controls the robot by GUI and mechanical buttons. An easily accessed emergency button can immediately cut off the power supply for motors and thus stop robot. Magnetic and mechanical

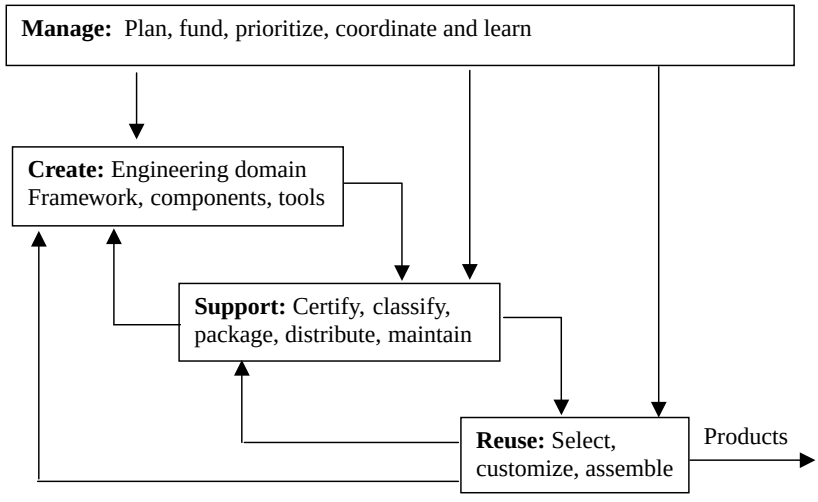

Fig. 4. Software reuse. There are four blocks that indicate different functional groups in software development and applications. They are management, creation, support and reuse. Arrows mean interaction between them. Strong management combines the other three together. The water flow and feedback among creation, support and reuse provide high efficiency for new products.

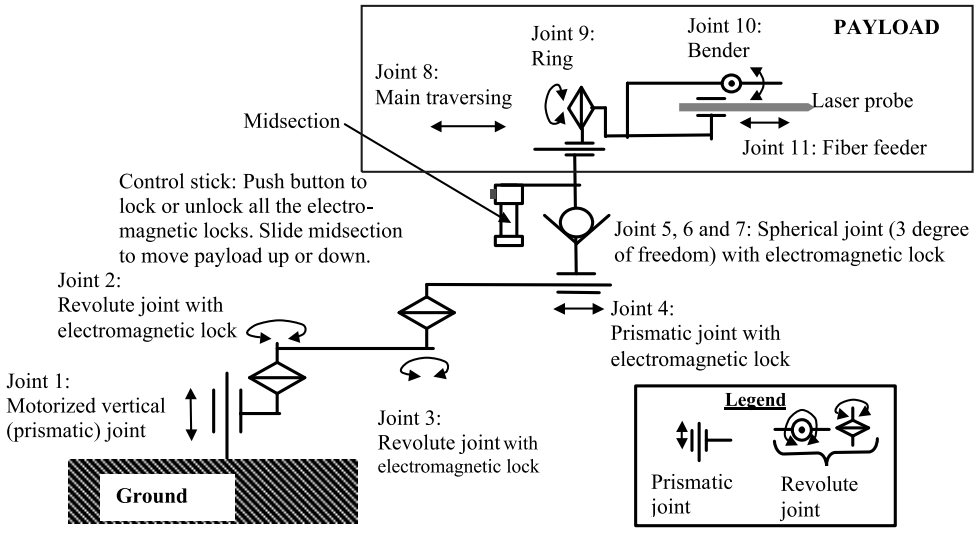

Fig. 5. Schematic of the manipulators. From bottom to top, the joints 1–7 are the support manipulator that provide motion freedoms robot setup. Joints 8–11 are the surgical manipulaors. There are four motors providing four motions freedoms for the robotic surgery.

locks can constrain joint motion. Sterilization can be easily performed. A surgical drape does not affect the manipulation. Mechanical parts were designed without sharp edges to avoid hurting the surgeon and patient. The HI principle was applied to the UI design.

We report on some GUI considerations for URObot. It is important to keep in mind that the surgeon is the decision maker and GUI shall always provide choices. For example, after the surgeon outlined the prostate boundary, the GUI shall allow the surgeon to accept or reject the results. GUI shall comply with surgical steps and surgeon's requirements.

It is also important that GUI provides a simple and effective way for surgeon to control the robot. We used a touch screen as the input because it is flexible, simple and requires no keyboard. Typically, one screen only has 1–3 buttons and no menu. This keeps selection simple and avoids wrong operations. Help information was displayed at the bottom of the screen. Users can simply follow the instructions to go through the entire operations.

It is critical that GUI prevents users from performing wrong operations and gives warning messages for serious decisions. For example, the robot accepts instructions to perform surgery, the GUI warns the surgeon of possible effects. The surgeon can then confirm or reject the decision. During surgery, the GUI pops up a huge button that can be used to immediately pause robot motion. The GUI shall intelligently disable some buttons to avoid wrong operation. For example, motor jogging is disabled before an ultrasound slice image acquisition is over and letter keys are disabled when the input is a number.

The GUI shall report system error and status. The software has the ability to automatically detect system errors such as failing to home, no image signal, PMAC off line, motor following error, amplifier fault and others. The GUI provided users with an interface to inspect the internal system.

3.3. Hardware considerations

3.3.1. Manipulator design

Fig. 5 shows the mechanical design of the URObot with ILC module. There are 11 joints in total. Joint 1 is motorized and controls the vertical motion of the whole robot arms. Joints 2 and 3 are revolute joints that control the horizontal motion of the platform. Joint 4 is a prismatic joint with electromagnetic lock that extends the robot arms. Joints 5–7 are spherical joints with electromagnetic lock that precisely control the posture of the surgical platform. After the robot is set up, the joints 1–7 are locked. Joints 8–11 are in the platform to mount the ultrasound probe, surgical tools and laser fiber. Joints 8–11 are the motion parts for image acquisition and robotic operation.

3.3.2. Critical analysis

The HI principle was applied to the joints analyses. The mechanical hazards are identified. They are (1) joint mechanical failure, joint cracking; (2) arm mechanical failure, arm cracking; (3) lock failure, the motors slide under gravity or other unexpected influence, motors jam; (4) the surgical platform moves during operation. These joints were very important in the safety enhancement. The systematic analysis methods were reported in Refs. [31,32].

We used the SIC principle to control the identified hazards. The failures of any joint may lead to disastrous results. For example, the failure of joint 1, which supports the whole robot will injure or possibly kill the patient. During design, the mechanical parameters should be clearly

specified in safety requirements. The loading test for these joints is the first step before any experiment.

The gantry including joints 1–7 plays a key role in the robot setup. Each joints has an electromagnetic and mechanical lock. Mechanical locks are freed and electromagnetic locks are locked before setup. When a button is pressed, magnetic locks are freed and the whole gantry can be manually repositioned. This step allows the surgical platform locate in an appropriate position for surgery. Mechanical locks can be manually locked to fix the gantry position after setup.

Other HISIC principles such as SCL and VV were applied to the URObot for safety enhancement. Positive and negative limit sensors and dead mechanical limits were installed to prevent motors from escaping away when there are voltage glitches in the servo amplifier output. Each motor was equipped with an encoder for positioning. Backup power supply ensures the system works continuously and shut down gracefully in case of power failure. Electrical, mechanical and software system underwent VV. The whole system met the standards of IEC 601 [44,45] and FDA [42].

4. Discussions

4.1. Implementation of HISIC plan

It is important for a medical robot that an HISIC plan is addressed specifically and conducted consistently. Safety is not only an engineering issue but also a management project that involves the scientist, engineer, surgeon, radiologist and patient. HISIC provides a systematic approach that can be applied to different robots. We discussed some general methods in Section 2 and gave some specific techniques in Section 3. We found that it is not enough to only test the end product. HISIC should run through all phases including research, design, production, test and applications.

4.2. Software safety methods

Software safety plays an important role in a medical robot because it is a complicated computer-based safety-critical system. Software failure can be catastrophic if it loses control of hardware from dangerous actions. It is sometimes difficult to identify a bug in a huge program because it describes a logical process not a physical entity and because there are numerous branches and conditions to be tested. We next describe some considerations on how HISIC principles such as HI and SIC to be implemented in software design and test.

Many factors may cause software failures. They include inadequate specifications, poor logic design, improper implementation, support software design error or failure. Others include erroneous pointer, memory allocation, call, jump, stack, unexpected timing or a combination of these conditions. Some errors may not be detected during the test phases.

Formal methods demonstrated approach to assure software safety [52]. Formal specifications and formal reasoning are more convincing than informal evidence for assurance of a safety case. This is a compelling reason to use formal methods, particularly when one is trying to satisfy legal constraints, or to demonstrate the best practice or adherence to principles.

Standard methods were another approach that was adopted by NASA [53–55] and IEEE [56]. The first cycle of software hazard analysis is top–down only. Bottom–up analyses take place after a sufficient level of design detail is available. The goal is to identify all credible hazards up front. In practice, coding standards that provide some generic software requirements are ‘safe’ subsets of programming languages. These are needed because most compilers can be unpredictable in how they work. For example, when some portions of memory are safety critical, the defaults chosen by the compiler might be unsafe. It is important to control which memory elements are assigned in a particular compilation process.

Structured design techniques are encouraged for safety software. They greatly reduce the number of errors, especially requirements errors that are the most expensive to correct. These errors may have the most impact on overall safety of a system. Object oriented design (OOD) and object oriented analysis (OOA) is the most recent technique [57]. To date, popular analysis methods included functional decomposition (FD) [58], data flow (DF) [59], structured analysis (SA) [60], and information modeling (IM) [61].

Code analyses are another ways to verify if programs correctly implement the design and comply with safety requirements. Techniques include code logic analysis, FTA [34], event tree analysis [35], petri-nets [62], code data analysis, code interface analysis, measurement of complexity and code constraint analysis [63–65]. When test is based on debug level, each line of code shall be executed at least once. Test data shall try every possible condition. Tests include normal, extremes and exception cases. Unusual inputs shall be produced to test special cases. ‘Black box’ methods are used to test functional specifications.

4.3. Hardware safety

Many mechanical methods were reported to ensure hardware safety. General methods include emergency button, position feedback [28], passive arm with mechanical constraints [24], force monitoring and redundancy sensors [66] and brake [27]. Before tools contact the patient for surgery, the moving speed shall be low [67].

The applicator position shall be correctly monitored. Here, we define applicator as a tool that directly contacts the patient and executes surgery, for example, the laser fiber for ILC. The applicator may be mounted on a platform

driven by motors. Traditionally, motor position is monitored by an encoder feedback. However, we think encoder feedback may not always reliably reflect the applicator position. This is because motor movement does not mean applicator movement and because applicator movement is unnoticeable when encoder output is not correct.

An effective control system that directly monitors applicator position is necessary for safety. For example, laser fiber may penetrate into prostate and wrongly enter the bladder. To prevent this from happening, we can install positioning sensors, mechanical constrains and limit switches on surgical platform and movable arms. We can also use an independent microprocessor to detect the signal from these sensors and to control safety switches. Another concern is the main platform. Since it supports the ultrasound probe and other tools, it must have enough force to stand firmly for long time. It shall be constrained within a safe range. Optical methods can also be used to monitor applicator position. We use OPTOTRAK 3020 (Northern Digital, Canada) as a redundancy position sensor. Surgical tools can also be monitored in real time.

The combination of hardware and software could be a potentially good approach for safety enhancement. Software is intelligent and flexible. Hardware lays material ground. The double check of software and hardware can save failures that may escape from one guard.

5. Conclusions

Safety is a key issue in medical robots. We put forward a systematic method named HISIC to analyze, control and evaluate the issue. The HISIC principles possibly provide a guideline for the safety enhancement of medical robots. The initial implementation of HISIC in URObot was successful. Safety considerations on software and hardware were discussed in detail. Tests showed that HISIC had the potential ability to improve safety. Further experiments are still being conducted in our laboratory.

References

[1] Tseng CS, Chen HH, Wang SS, Tseng HM. Image-guided robotic navigation system for neurosurgery. J Robot Syst 2000;17(8):439–47.
[2] Masamune K, Ji LH, Suzuki M, Dohi T, Iseki H, Takakura K. A newly developed stereotactic robot with detachable drive for neurosurgery. In: Medical image computing and computer-assisted intervention — MICCAI'98, 1998, vol. 1496. p. 215–22.
[3] Benabid AL, Hoffmann D, Ashraf A, Koudsie A, Esteve F, Le-Bas JF. Robotized neurosurgery: state of the art and future developments. Bull Acad Natl Med 1997;181(8):1625–36.
[4] Taylor RH, Mittelstadt BD, Paul HA, Hanson W, Kazanzides P, Zuhars JF, Williamson B, Musits BL, Glassman E, Bargar WL. An image-directed robotic system for precise orthopedic-surgery. IEEE Trans Robot Automat 1994;10(3):261–75.
[5] Brandt G, Radermacher K, Lavallee S, Staudte HW, Rau G. A compact robot for image guided orthopedic surgery: concept and preliminary results. Lect Notes Comput Sci 1997;1205:767–76.
[6] Howe RD, Matsuoka Y. Robotics for surgery. Annu Rev Biomed Engng 1999;1:211–40.
[7] Harris SJ, Lin WJ, Fan KL, Hibberd RD, Cobb J, Middleton R, Davies BL. Experiences with robotic systems for knee surgery. Lect Notes Comput Sci 1997;1205:757–66.
[8] Davies BL, Hibberd RD, Ng WS, et al. A surgeon robot for prostatectomies. In: Proceedings of the Fifth International Conference on Advanced Robotics, Robots in Unstructured Environments, 1991. p. 871–5.
[9] Ng WS, Davies BL, Hibberd RD, Timoney AG. Robotic surgery. IEEE Engng Med Biol Mag 1993;12(1):120–5.
[10] Fei BW, Kwoh CK, Ng WS. The software design for a medical robot for urological applications. In: Proceedings of the First Joint IEEE BMES/EMBS Conference, 1999 Oct 13–16; Atlanta, GA. p. 896.
[11] Hoenig DM, Shalhav AL, Arcangeli CG, Ostrander DD, Elbahnasy AM, Clayman RV. Under-table mounting for AESOP robot for laparoscopic flank surgery. Minimally Invasive Therapy Allied Technology 1997;6(5–6):460–2.
[12] Taylor RH, Funda J, Eldridge B, et al. A telerobotic assistant for laparoscopic surgery. IEEE Engng Med Biol 1995;14(3):279–88.
[13] Falcone T, Goldberg J, Garcia-Ruiz A, Margossian H, Stevens L. Full robotic assistance for laparoscopic tubal anastomosis: a case report. J Laparoendosc Adv Surg Tech, Part A 1999;9(1):107–13.
[14] Hein A, Lueth TC. Robot control in maxillofacial surgery. Exp Robot VI 2000;250:173–82.
[15] Shennib H, Bastawisy A, McLoughlin J, Moll F. Robotic computer-assisted telemanipulation enhances coronary artery bypass. J Thorac Cardiovasc Surg 1999;117(2):310–3.
[16] Schweikard A, Bodduluri M, Adler JR. Planning for camera-guided robotic radiosurgery. IEEE Trans Robot Automat 1998;14(6):951–62.
[17] Dario P, Guglielmelli E, Allotta B, Carrozza MC. Robotics for medical applications. IEEE Robot Automat Mag 1999;3(3):44–56.
[18] Davies B. A review of robotics in surgery. Proc Inst Mech Engrs, Part II, J Engng Med 2000;214(H1):129–40.
[19] Pransky J. ROBODOC — surgical robot success story. Ind Robot 1997;24(3):231–3.
[20] Mettler L, Ibrahim M, Jonat W. One year of experience working with the aid of a robotic assistant (the voice-controlled optic holder AESOP) in gynaecological endoscopic surgery. Human Reprod 1998;13(10):2748–50.
[21] Paul A. Surgical robot in endoprosthetics. How CASPAR assists on the hip. MMW Fortschr Med 1999;141(33):18.
[22] Cichon R, Kappert U, Schneider J, Schramm I, Gulielmos V, Tugtekin SM, Schueler S. Robotically enhanced 'Dresden technique' with bilateral internal mammary artery grafting. Thorac Cardiovasc Surg 2000;48(4):189–92.
[23] Taylor RH, Paul HA, Kazanzides P, et al. Taming the bull: safety in a precise surgical robot. In: Proceedings of the Fifth International Conference on Advanced Robotics, Robots in Unstructured Environments, 1991. p. 865–70.
[24] Troccaz J, Lavallee S, Hellion E. A passive arm with dynamic constraints: a solution to safety problems in medical robotics? Proc Int Conf Syst Man Cybern Syst Engng Service Humans 1993;3:166–71.
[25] Wikman TS, Branicky MS, Newman WS. Reflexive collision avoidance: a generalized approach. In: IEEE International Conference on Robotics and Automation, Atlanta, GA, 1993. p. 31–6.
[26] Visinsky ML, Cavallaro JR, Walker ID. Robotic fault detection and fault tolerance: a survey. Reliab Engng Syst Safety 1994;46(2):139–58.
[27] Davies BL. A discussion of safety issues for medical robots. In: Taylor RH, Lavalle S, Burdea GC, Mosges R, editors. Computer-integrated surgery technology and clinical applications, Cambridge, MA: MIT Press, 1995. p. 287–96.
[28] Ng WS, Tan CK. On safety enhancements for medical robot. Reliab Engng Syst Safety 1996;54(1):35–45.
[29] Rau G, Radermacher K, Thull B, Pichler C. Aspects of an ergonomic system design of a medical worksystem. In: Taylor RH, Lavallee S,

Burdea GC, Mosges R, editors. Computer integrated surgery, Cambridge, MA: MIT Press, 1996. p. 203–21.
[30] Erbse S, Radermacher K, Anton M, Rau G, Boeckmann W, Jacke G, Staudte HW. Development of an automatic surgical holding system based on ergonomic analysis. In: Troccaz J, Grimson E, Mosges R, editors. CVRMED II and MRCAS III, Lecture notes in computer science, Berlin: Springer, 1997. p. 737–46.
[31] Lewis C, Maciejewski AA. Dexterity optimization of kinematically redundant manipulators in the presence of joint failures. Int J Comput Elect Engng 1994;20(3):273–88.
[32] English JD, Maciejewski AA. Measuring of reducing the Euclid-space effects of robotic joint failure. IEEE Trans Robot Automat 2000;16(1):20–8.
[33] Ikuta K, Nokata M. General evaluation method of safety for human-care robots. In: Proceedings of the International Conference of IEEE on Robotics and Automation, Detroit, MI. May 1999. p. 2065–72.
[34] Connolly B. Software safety goal verification using fault tree techniques: a critically III patient monitoring example. In: Proceedings of the Fourth Annual Conference on Computer Assurance, Systems Integrity, Software Safety and Process Security, 1989. p. 18–21.
[35] Khodabandehloo K. Analysis of robot systems using fault and even trees: cases studies. Reliab Engng Syst Safety 1996;53(3):247–64.
[36] Hamilton DL, Visinsky ML, Bennett JK, Cavallaro JR, Walker ID. Fault tolerant algorithms and architectures for robotics. In: Proceedings of the IEEE Seventh Mediterranean Electrotechnical Conference, vol. 3, Antalya, Turkey. April 1994. p. 1034–6.
[37] Dowler NJ. Applying software dependability principles to medical robotics. Comput Control Engng J 1995(Oct):222–5.
[38] Seward DW, Garman A. The software development process for an intelligent robot. Comput Control Engng J 1996;7(2):86–92.
[39] Gowen LD. Specifying and verifying safety-critical software system. In: Proceedings of the Seventh IEEE Symposium on Computer-based Medical System, 1994. p. 235–40.
[40] Valey P. Techniques for development of safety-related software for surgical robots. IEEE Trans Inform Technol Biomed 1999;3(4):261–7.
[41] EN 775, Manipulating Industrial Robots — Safety.
[42] Reviewer Guidance for Computer Controlled Medical Devices Undergoing 510(k) Review. Office of Device Evaluation, Center for Device and Radiological Health, Food and Drug Administration, August 1991.
[43] IEC 1508 (Parts 1–7). Functional safety: safety related systems. International Electrotechnical Commission, 1996.
[44] IEC 601-2-18. Medical electrical equipment. Particular requirements for safety, Part 2, Section 2.18. Specification for endoscopic equipment, 1996.
[45] IEC 601-2-22. Medical electrical equipment. Particular requirements for safety, Part 2, Section 2.22. Specification for diagnostic and therapeutic laser equipment; 1995.
[46] Ng WS, Chung VR, Vasan S, Lim P. Robotic radiation seed implantation for prostatic cancer. Proc 18th Annu Int Conf IEEE Med Biol Soc 1996;1:231–3.
[47] Shee CY. Development of robotic interstitial laser therapy for benign prostatic hyperplasia. MSc Thesis. Singapore: Nanyang Technological University, 1999.
[48] Shi XM. Robotic radioactive seed implantation for prostate cancer. MSc Thesis. Singapore: Nanyang Technological University, 1999.
[49] Gideons H. Computer assisted laser resection of the prostate. MSc Thesis. Singapore: Nanyang Technological University, 2000.
[50] Jag S. Software reuse: domain analysis and design processes. New York: McGraw-Hill, 1999.
[51] Andre KE. Software reuse: a holistic approach. Chichester: Wiley, 1995.
[52] Muffy T. Formal methods and their role in developing safe systems (report of a workshop organized jointly with the BCS). High Integr Syst 1996;1(5):447–51.
[53] NSS 1740.13 NASA Software Safety Standard. Interim Release, June 1994.
[54] NSTS 13830B Implementation Procedure for NASA Payload System Safety Requirements.
[55] SMAP-GB-A201 NASA Software Assurance Guidebook, September 1989.
[56] IEEE standard for software safety plans. New York: IEEE, 1994.
[57] Leach. Object oriented design and programming with C++. New York: Academic Press, 1995.
[58] Wing J. A specifier's introduction to formal methods. Computer 1990;23(9):8–24.
[59] Meyer B. On formalism in specification. IEEE Trans Software Engng 1985:6–26.
[60] Wayne SP. Software design: concepts and methods. New York: Prentice Hall, 1991.
[61] Boehm BW. A spiral model of software development and enhancement. Computer 1988;21(5):61–72.
[62] Leveson N, Stolzy J. Safety analysis using Petri-Nets. IEEE Trans Software Engng 1987:386–97.
[63] Leveson N, Harvey P. Analyzing software safety. IEEE Trans Software Engng 1983;9(5).
[64] Leveson N, Cha S, Shimeall T. Safety verification of programs using software fault trees. IEEE Trans Software Engng 1991;8(4).
[65] Garrett C, Yau M, Guarro S, Apostolakais G. Assessing the dependability of embedded software systems using the dynamic flowgraph methodology. In: Proceedings of the Fourth International Working Conference on Dependable Computing for Critical Applications, 1994 Jan 4–6; San Diego.
[66] Kazanzides P, Zuhars J, Mittelsstadt B, Taylor RH. Force sensing and control for a surgical robot. In: Proceedings of the IEEE Conference on Robotics and Automation, 1992; Nice. p. 612–17.
[67] Lavellee S, Troccaz J, Gaborit L, et al. Image guided operation robot: a clinical application in stereotactic neurosurgery. In: Proceedings of the IEEE Conference on Robotics and Automation. May 1992; Nice. p. 618–24.