RELIABILITY ENHANCEMENTS OF SURGICAL ROBOTS THROUGH SOFTWARE FAULT TOLERANCE

JIGNA B.PRAJAPATI[1], Dr. N. K. MODI[2]

[1] Acharya Motibhai Patel Institute of Computer Studies, Ganpat Vidyanagar, Kherva, India.
[2] S.V.Institute of Computer Studies Kadi, Gujarat, India.
Corresponding author's information: Jigna B. Prajapati, Assistant Professor, Acharya Motibhai Patel Institute of Computer Studies, Ganpat Vidyanagar, Kherva, India, E-mail: jigna.prajajapati@ganpatuniversity.ac.in , Phone: (M) 91-9925065025

Abstract

World is very speedily moving towards most hi-tech technology in each factor. Robotic surgery is a new and exciting emerging technology that is taking the surgical profession by storm. Surgical robots have become the essential for centers wanting to be known for excellence in minimally invasive surgery despite the current lack of practical applications. Present work discusses the performance issues of Robotic surgery and spotlight the weak root. The important issue of surgical robots is to deal with fault arise when the surgery is going on. Then try to make surgical robots system more reliable with fault tolerance with design diverse approach. The technique of design diverse applied to Surgical robots system will become more effective to handle faults and enable Surgical robots in reliable mode. Present approach can deal with the software and hardware faults by applying the techniques of software fault tolerance and hardware fault tolerance, respectively.

Keyword: Surgical robots, software fault, hardware fault, design diverse

Introduction

Surgical robots are medical robots which aid doctors or surgeons in administering surgical operations. In today's technology, long-distance surgeries are possible with the help of these modern surgical robots. These robots may be controlled remotely by trained surgeons over a secured network such as the internet [1, 2].

There are so many robots available today for tasks such as hip replacement in orthopedics, camera positioning for laparoscopic surgery, minimally invasive cardiac surgery, and needle placement for image-guided interventions. To take full advantage of robots, we must employ them to do things that humans cannot do, such as motion scaling and tremor reduction. We must establish safety protocols for the use of surgical robotics [1].

For the purpose of surgical applications, cooperative control systems promise significant advantages. A steady hand robot [2] can provide guidance, and enforce safety constraints. It can also use compliance models taking into account procedure-specific issues such as tissue properties and thereby provide tactile sensing for the human surgeon, which allows the surgeon's superior intelligence and experience to be used with greater precision and safety than before.

The new steady hand robot is [3] the first platform for these experiments. Designed for microsurgical applications this is a 7-degree-of-freedom manipulator. It has 3 XYZ translation stages at the base for coarse positioning, two rotational degrees of freedom at the shoulder [4], [5] an instrument insertion and an instrument rotation stage. Another

experiment was a LARS robot designed for camera holding applications in laparoscopic surgery; LARS is a 7 degree-of-freedom manipulator [5]. It has two rotational degrees of freedom at the shoulder and three translation stages at the base. LARS also has a force sensor built in the end-effectors and remote center of motion.
Computer-assisted surgery, uses special devices attached to the body to help your surgeon ensure proper positioning of the joint replacement implants. The devices are detected by an infrared camera that is connected to a computer terminal which constructs an electronic model of the knee or hip based on the information gathered by sensing the position of these probes. The computer can then help guide your surgeon's placement of the knee or hip replacement implants to ensure proper positioning of the joint replacement [6].

Recent Usage and Capabilities of Surgical Robots

There are number of surgical robotic devices available today having range of functions in the surgery environment. Few robots is working as a surgeon's "third hand" for moving the camera during minimally invasive procedures. Others exist to perform or facilitate telesurgery, telemonitoring, tele-mentoring, or true telepresence instruction. Still other robotic devices perform or assist with image-guided interventions. Transforming existing robotic devices into all-purpose devices or systems was a concept that emerged from discussions of this Working Group. It would be facilitated by combining both the image and information processing capabilities, and the revelation and task performance systems onto a multi-purpose workbench-like platform.
Robotic devices would be designed with automated tool changers, thereby enabling robotic devices to change tools rapidly and precisely in order to perform a multitude of tasks in the surgery environment. This capability could easily alter a robotic device's function and make it a more universal or multi-purpose device. As a result, robots could be made more useful in the neurosurgical, orthopedic, cardio-thoracic, and urological suites.
In addition, the use of such robots need not be limited to surgical task performance. The surgery environment needs capabilities than merely operating as tools to perform simple tasks. Robotics should be used to facilitate the overall performance of complex surgical interventions in the technologically advanced environment of surgery. It will involve information management, data processing, image processing, image-guided intervention, complex and minimally invasive task performance, and control of the OR assets, supplies, and personnel as well as management of the flow of patients within the process of surgical intervention.
Improving the capabilities of robotic systems must differentiate the machines' abilities to perform procedures which humans can do from those which humans cannot do. Robots and computer systems can process data and acquire data and images in manners far superior to humans. A challenge is to take the human ability to interact with the surgical environment and make decisions, then to translate these abilities into task performance needs for a surgical robotic system.

There are many devices used in Robotic surgery for different type of surgery. Here is The ROBODOC, a Curexo Tech, Inc. was founded in 2002 to marketize and profit from the Robodoc system. In August 2008 the system finally received a 510(k) clearance from FDA[7].

Figure 1: Robodoc- used in knee replacement surgery[7]

This system includes two components; ORTHODOC®, a computer workstation equipped with proprietary software for 3-D preoperative surgical planning, and the ROBODOC® Surgical Assistant, a computer-controlled surgical robot utilized for precise cavity and surface preparation for hip and knee replacement surgeries. The ROBODOC System, previously marketed by Integrated Surgical Systems (ISS), made medical history in 1992 as the first robot assisting in a human Total Hip Arthroplasty [7].

Performance issues of surgical robots

a. Steep learning curve device are difficult operate
b. high cost of the device- capital costs- running costs
c. Time gap between the instructor and the robot
d. When numbers are small and basic things are there at that time economies scale and the price of machinery, maintenance and training will all go up to reasonable levels.
e. Do specified work only
f. Only few robots can work on situation
g. Holding reasonable expectations about the accuracy and precision of robotic surgical systems.
h. Preparing contingency planning and undergoing training in the event of surgical robotic system failure or unanticipated events (accidents).

Beside these the main issue of surgical robots is to deal with fault arise when the surgery is going on. There are several ways to cope with different types of faults for the system control [8]. Here we would like give fault tolerance technique to manage the faults related to the architectural faults [9] t make surgical system more adaptive.
In order to ensure that these systems operate as indicated, even in extreme conditions, it is important to have fault-tolerant computer systems, hardware and software [8]. We include Design diversy and Environment diversy. Design diversy is identical service through individual design and implementations. As the exact copy of the software component redundancy cannot improve the reliability in terms of software Design malfunction, we must ensure it [8, 10, 11].
Environment diversity is the new approach to fault tolerance in software. Although this technique has been used for a long time in an ad-hoc manner, only recently has gained

recognition and importance [12]. To do so, we are approaching the design diversy with hardware and software fault tolerance.

Design Diversity

As the exact copy of the software component redundancy cannot get better reliability in terms of software malfunction, we must ensure diversity in the development and implementation of software. Present study aims to provide diverse through variants design to minimize the indistinguishable causes of errors. It also ensures reliability of server from each option as high as possible, so at least one option (for sever) would be commissioned at any time. Design diversity begins with primary requirements specification which becomes technical specifications of the program [13]. The variants perform their operations using inputs given by the user. Since there are multiple results, and this requires repetition and a way to decide which result to use. To achieve reliability study was focused on Recovery block and N-version programming.

A. Recovery Blocks(RcB)

The basic RcB scheme is one of the two original diverse software fault tolerance techniques. The RcB is categorized as a dynamic technique [14]. It chooses one alternate accepted execution path. If selected execution path and selected execution test passed then it would end execution, otherwise it will divert to another alternative. Such process will repeated up to the no. of alternative to passed successful. A less effective option (s) is consistently in the main block and are called (secondary) alternate or alternative units [15].

B. N-Version Programming

NVP was suggested by Elmendorf in 1972 and developed by Avizienis and Chen in 1977–1978. NVP gives us the different version of the same system [16, 17]. Compared with RCB, NVP is static method. It use different version of the same piece of code. Hardware fault tolerance architectures, associated with NVP is N-modular [16, 17]. The Processes can run simultaneously on different computers or sequentially on one computer [15].

Hardware Fault-Tolerance

Most fault-tolerant designs were aimed at building computers that automatically recover from random failures occurring in the hardware [8]. Each module backup protective redundancy so that if a module fails, others may assume its function. Special arrangements added to handle errors and recover. Generally when we any one of the device has problem we have immediately another device which can perform instead of previous module.

Implementation

No basis on the observation that the failure of most programs transient in nature, the environment and the diversity of approaches require reimplementation or reexecuting of programs in a different environment. Offers diversity of the environment effectively with Heisenberg by exploiting its definition and nature [12]. Adams has proposed restarting the system as the best approach to masking software faults [10]. Environment diversity is a generally give a way of restarting. This was proposed as cheap but effective techniques for fault tolerance in software. There are three components that determine the behavior of the process or the execution of.

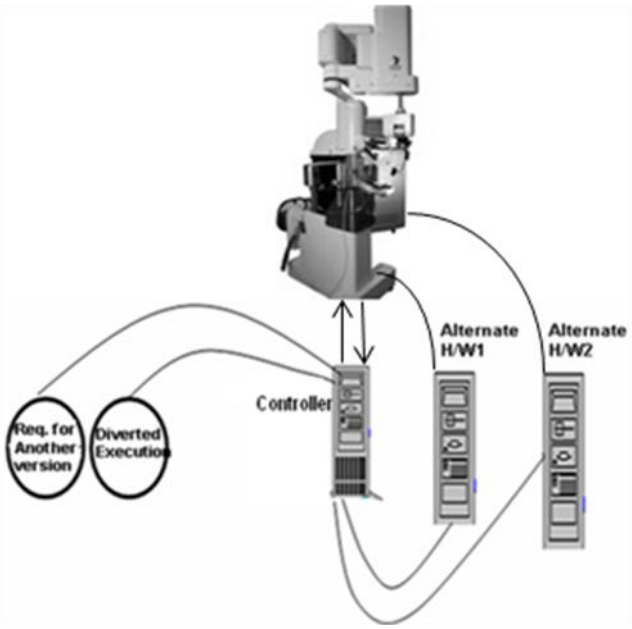

Figure 4 Robotic surgeries with H/W and Design diversy

The capricious state: It consists of a program stack and static and dynamic data segments. The constant state: Operating system (OS) environment: This refers to all resources through a program accesses an operating system such as paging, file systems, communication channels, keyboards and monitors. Transient deficiencies usually occur in the robotic surgery because of design faults in software which result in unacceptable and erroneous states in the OS environment [12]. Therefore various attempts were made to provide new or changed operating environment for the software. When software fails, it will be restarted in another, unmistakably OS environment that is achieved by some cleaning operations. Examples of methods of protection of environment diversity include the operation again, restart the system and rebooting the host, and though if the problem was not handling it will use a different h / w for specific hardware failures [12].
If any outside treatment is absent and does not work, we can use alternative equipment. As we have discussed an approach to handling software errors, mistakes and problems, or diversion of flow execution. In Robotic surgery the working f system is very most important in any situation l when it faces S / W difficult to answer in any way, distract performance S / W will not cause a system failure as we already discussed approach for handling software errors, faults and problems by either diverting the flow of execution. In Robotic surgery When such part facing S/W problem to answer the client node in any way, the divert execution of the S/W will not cause system fail. We can place some piece of information which provide the same type of dealing with inputs and assure with required output.
The RcB uses different acceptance test. When execution started first it check the primary execution , if it is according to the requirements or meet the satisfactory level then ok otherwise go to another alternative. The RcB gives us many alternatives until few extend. But in Robotic surgery is life critical system so it need Sound reliability in order to deal with faults by providing many alternatives of the any part of the system.
Using different versions in NVP, the execution began with first version of the system, if any fault occurred it will use another version of the same piece of problem. Once it pass by alternate version and execute complete with properly. It intimates about completing executing by successful handling the occurred faults.

Conclusion

Robotic surgery is an important option for doctors or surgeons in administering surgical operations. Implemented with fault-tolerant system, it can extensively improve the quality of surgeons' support and reduce risk. The important is evaluation and deployment planning properly. We discussed Robotic surgery system's usage and its performance issues. Then, we addressed software and hardware fault tolerance technique which either improved the traditional techniques or took a new approach to solve the issues of Robotic surgery system and make surgical robots more reliable. In conclusion, the application of all of these techniques and approach is relatively new to the area in Robotic surgery. The RcB and N-version work much effectively at any consign. But at Robotic surgery, these are the most adaptable solutions for dealing with different types of faults. Robotic surgery system such approached techniques increasing sound reliability and performance.

References

1. Anthony R. Lanfranco, Andres E. Castellanos, Jaydev P. Desai, William C. Meyers. Robotic Surgery a Current Perspective. Ann Surg. January 2004; 239(1): 14–21.
2. Taylor R, Jensen P, Whitcomb L, Barnes A, Kumar R, Stoianovici D, Gupta P, Wang ZX, deJuan E, Kavoussi L. A steady-hand robotic system for microsurgical augmentation. International Journal of Robotics Research. December 1999; 18(12):1201-1210.
3. Rajesh Kumar, Tushar M. Goradia, Aaron C. Barnes, Patrick Jensen, Louis L.Whitcomb, Dan Stoianovici, Ludwig M. Auer, Russell H. Taylor. Performance of Robotic Augmentation in Microsurgery-Scale Motions. Lecture Notes in Computer Science, 1999, Volume 1679/1999, 1108-1115.
4. Stoianovici D, Cadeddu JA, Demaree RD, Basile SA, Taylor RH, Whitcomb LL, Sharpe WN, Kavoussi LR. An efficient needle injection technique and radiological guidance method for percutaneous procedures. Lecture Notes in Computer Science. 1997; Vol.1205. 295-298.
5. Stoianovici D., Whitcomb LL. Anderson JH., Taylor RH., Kavoussi LR. A Modular Surgical Robotic System for Image Guided Percutaneous Procedures, Proceedings of MICCAI'98, Lecture Notes in Computer Science, Springer-Verlag, ,1998 Vol. 1496, 404-410.
6. Krackow KA, O'Connor MI. The Role of CAS in TKA. AAOS Now. March 2009; Vol 3, No 3, 19-21.
7. ROBODOC® FDA Cleared Active Robotic System to be launched at AAHKS. http://www.robodoc.com/media.html.
8. Laprie JC, Béounes C, Kanoun K. Hardware and Software Fault Tolerance: Definition and Analysis of Architectural Solutions. Proceedings of FTCS-17, Pittsburgh, PA, 1987, 116–112,
9. A Conceptual Framework for System fault tolerance. hissa.nist.gov/chissa/SEI_Framework/framework_1.html
10. Gray J., Siewiorek DP. High-availability computer systems. IEEE Computer, Vol. 24, No. 9, 39-48, 1991
11. Avizienis A., Kelly JP. Fault tolerance by design diversity - Concepts and experiments. Computer. Aug. 1984, Vol. 17, 67-80.
12. Ammann PE., Knight JC. Data diversity: an approach to software fault tolerance. 17th Int. Symp. Fault Tolerant Computing, Pittsburgh, 1988, 122-126.

13. Randell B., Xu J. The evolution of recover block concept. In Software Fault Tolerance,
Lyu, M. (ed), 1-22. Trends in Software, Wiley, 1994, ISBN 0-471-95068-8
14. Zaipeng Xie, Hongyu Sun, Kewal Saluja. A Survey of Software Fault Tolerance Techniques. citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.99.1320
15. Avizienis A. On the Implementation of NVersion Programming for Software Fault-Tolerance during Execution. COMPSAC '77, Chicago, IL, 1977, 149–155.
16. Chen L., Avizienis A. N-Version Programming: A Fault-Tolerance Approach to Reliability of Software Operation. Proceedings of the 8th International Symposium on Fault Tolerant Computing System (FTCS-8), Toulouse, France, 1978, 3–9.
17. Cristian F. Exception Handling and Tolerance of Software Faults, in M. Liu, ed., Software Fault Tolerance (John Wiley & Sons, 1994), 81-107