A Scalable System for Real-Time Control of Dexterous Surgical Robots

Extended Abstract

Paul Thienphrapa, Renqiu Huang, Peter Kazanzides
{pault,rqhuang,pkaz}@jhu.edu
Tel: (410) 516-4779, Fax: (410) 516-4410

Department of Computer Science
Johns Hopkins University
3400 N. Charles St. CSEB 112
Baltimore, MD 21218 USA

Note: This paper is a follow-up to work presented at TePRA 2008.

The requisite cabling and control processing for surgical robots can become unwieldy as dexterity is increased, due to the additional degrees of freedom (dof). Motivated by dexterous snake-like robots for minimally invasive surgery (MIS), this paper details the development of a low-level control system that uses IEEE 1394 (FireWire), linking the computer to low-latency field-programmable gate arrays, to distribute I/O and centralize processing. A standard programming interface is defined as part of the control system to enhance its scalability. These features increase the viability of complex surgical robots and ease their development by reducing cables and enabling the processing of many axes of control on a single computer.

MIS is often beneficial due to reduction of trauma, leading to fewer complications and shorter hospital stays. However it poses a number of challenges for surgeons, including constrained workspaces, limited field of view, and lack of dexterity at the distal end. Existing MIS tools are rigid, difficult to manipulate through narrow insertion tubes, and they lack adequate suturing and tissue reconstruction capability.

One way to improve MIS is by providing surgeons with robotically-controlled tools with many dof for high dexterity. However, the corresponding hardware increase imposes a practical limit on the exploration of this idea. Research on different types of multi-axis surgical robots is often mired in the hardware construction effort. In response to these difficulties, we present the development of a system that is well suited for real-time control of robots with many axes.

These efforts originate from the need to improve the old multi-axis controller of the Snake Robot and replicate the controller for other research projects. The old controller utilizes a centralized I/O arrangement, whereby command and feedback signals are transmitted in raw analog form over long cables running between the robot and the computer. Though the design is conceptually straightforward, the cumbersome wiring associated with it introduces complications such as noise, cable drag, reduced reliability, and greater construction effort. The debug space is vast as there are many candidates for connectivity problems, so this approach limits the ability to develop increasingly dexterous surgical robots.

The long term benefits of developing a control system using IEEE 1394 are multifold. The high speed serial bus encourages a distributed I/O and centralized processing architecture. In such an approach the I/O processing logic is simple to implement and requires little maintenance. Signal integrity is improved because digitization occurs near the actuator sites. Cable complexity is reduced because distributed I/O hardware is accessed through a serial link, and the number of controllable axes can be more easily scaled up. Overall robustness is improved.

IEEE 1394 allows for centralized processing of a large number of channels on a centralized computer, so low-latency inter-task communication can be used instead of networking. The integration allows for a familiar software development environment and a standard API; this alleviates researchers and programmers from learning the idiosyncrasies of individual embedded micro-controllers, so they can instead focus on higher-level tasks. Furthermore, this architecture can more readily harness the power of high performance computing.

This work is intended to facilitate research by providing a generic interface and scalable mechanism for fine-grain real-time control. Such a custom solution was necessary for servoing the low power dc motors of the Snake Robot because an adequate commercial solution was not available. The solution allows for flexible customization of parameters, investigation of complex control laws, and high-density distributed I/O. This control system is designed to ease the development of dexterous surgical robots from both hardware and software perspectives.

In our previous work we found that the latencies for per-axis bus transactions were significant. Using quadlet (32-bit) transactions, the average transaction time was 34.5 μs for a read and 30.2 μs for a write. In a straightforward implementation (read-process-write) for a seven-axis robot, a combined read/write time of 453 μs leaves only 547 μs for control computations at 1 kHz, and is not even feasible at 8 kHz. Given the bus speed of 400 Mbps, we concluded that software overheads were a predominant factor. It became necessary to bundle the data for multiple axes into blocks in order to overcome these limitations.

Testing of this new approach shows the base latency to be about 33.2 μs for a read and 30.7 μs for a write, which agrees with our previous findings. The base latency dominates the total communication time, so potentially hundreds of axes can be controlled without communication time being an issue, so long as block transfers are used. The average speed is roughly 350 and 290 Mbps for reads and writes respectively. Neither value reaches the nominal 400 Mbps due to overhead, but it defies intuition that reads are faster than writes since read transactions are slightly more complicated and incur greater latency. A possible explanation may involve differences between how the computer and FPGA request bus access. Nevertheless, the results confirm real-time performance; in future work, the use of broadcast packets and isochronous transfers may yield further improvements.

Though the concept of this control system is not necessarily novel given existing technologies, we are not aware of work that uses the IEEE 1394 bus in this specific manner. The predominant use of IEEE 1394 is in video acquisition; in robot control it is mostly used with embedded servo controllers that do not require real-time updates. We further contend that the design will ease the development of dexterous robots and enable further research. The API will be compatible with a standard medical robotics framework, the Surgical Assistant Workstation.