2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007

Automated Tool Handling for the Trauma Pod Surgical Robot

Diana C.W. Friedman*, Jesse Dosher†, Tim Kowalewski†, Jacob Rosen† and Blake Hannaford†
*Dept. of Mechanical Engineeering
†Department of Electrical Engineering
University of Washington
{dwarden, jdosher, tmk, rosen, blake}@u.washington.edu

Abstract— **In order to enable robotic surgery without human assistance, a means must be developed to change tools. As part of the larger Trauma Pod Project, we developed the Tool Rack Subsystem — an automated tool rack capable of holding, accepting, and dispensing up to 14 tools for the da Vinci^{TM} surgical robot. Borrowing some techniques from industrial automation, we developed a robust system capable of presenting any stored tool in 700ms or less. Tools are positively retained in a sterilizable carousel in a compliant manner designed to accomodate misalignment during tool exchange. RFID equipment is integrated into the system and the tools so that tools can be inventoried and presented by function or serial number instead of rack position. The resulting device has completed testing and integration into the Trauma Pod system and met all its design requirements.**

I. Introduction

In one vision of the future of surgery, all humans will be removed from the operating room except the patient[1]. This vision has driven the requirements and system architecture for the Trauma Pod Project, a collaboration in which we have participated that is led by SRI International and includes General Dynamics, General Electric, Intuitive Surgical, Multi-Dimensional Imaging Inc., Oak Ridge National Laboratory (ORNL), Robotic Surgical Technology, the University of Maryland, the University of Texas at Austin, and the University of Washington at Seattle. The Trauma Pod Project website is `http://www.traumapod.org`

One component of the the Trauma Pod architecture is an automated tool changer, capable of changing tools on the da Vinci surgical robot without human assistance. The tool changer consists of an industrial, 7 DOF robot arm (Mitsubishi PA-10), a dual-headed gripper and compatible grasping fixtures on the da Vinci tools, and the Tool Rack Subsystem (TRS). This paper describes the features and performance of the TRS. Many requirements for the TRS were dictated by system considerations in the architecture of the overall Trauma Pod system and in particular, integration with the rest of the tool changer components by ORNL. These considerations cannot be fully described here due to limitations of space. Instead, this paper focuses on the design and performance of the TRS against these external specifications.

Related Work Automated tool changers have been used in CNC machining for decades. Like a CNC tool changer, a surgical tool changer needs to handle multiple tools and run at high speeds. Our requirements for a surgical tool changer require many modifications to the traditional CNC tool changer design, however. For instance, more compliance and a wider acceptance tolerance on presentation is necessary in order to interface with the Mitsubishi PA-10 arm. Being a medical device, the tool carousel also needs to be sterilizable. Since different operations require different tools, the location of each tool within the rack must be variable. Due to the size and shape of the tools, we also need positive grasping force on all tools. Finally, we require a high level networked control interface for the tool changer.

Treat[2] recently developed the Penelope Surgical Instrument Server and used it in surgery. Penelope uses voice recognition, speech synthesis, and machine vision to interface with human surgeons in the operating room. Currently, it is not designed to work with MIS tools, one of the requirements of our tool rack. The methods used to control and deliver tools to the surgeon appear to work for interactions with humans, but are suboptimal for interactions with another robot. The tools are delivered to the surgeon using an electromagnet, allowing for large errors in position and orientation upon delivery. The tools are not held in place, and may be jostled out of position or off the instrument tray accidentally. Finally, Penelope uses voice commands to transfer tools to the surgeon. When interacting with another robot, it is more efficient to use networked commands. Several of the same concerns can be raised about the Scrub Nurse Robot currently being developed by Miyawaki and Masamune[3], which mimics a human nurse's arm when delivering tools to a human surgeon. Our TRS attempts to address these concerns by creating a surgical tool rack specifically designed for interaction with another robot.

II. Requirements

The TRS design requirements were derived from overall system analysis by the Trauma Pod engineering team. They can be broken down into functional requirements, electromechanical hardware requirements, and software requirements.

A. *Functional Requirements*

The TRS needed to present an arbitrary tool for pickup and to grasp each tool with a positive retention. A typical operating sequence is:

1) Present an open tool position.
2) Open the tool-position grasper.
3) Receive the tool placed into position. Accomodate positioning errors in presentation of the tool.
4) Close the grasper around tool. Wait for tool to be released by robot.
5) Present a new tool.
6) Wait for tool to be grasped by robot.
7) Release grasper.
8) Wait for robot to remove tool.
9) Close grasper.

Additional functions included:

- Perform an initial calibration sequence.
- Report an inventory of current tools at any time after initial calibration.
- E-Stop, Shutdown, Startup, and Power Cycle sequences.

B. Hardware

The overall purpose of the system was to provide rapid and reliable machine access to up to 14 da Vinci tools. These tools, standard Intuitive Surgical products, were modified by substitution of a modified plastic housing. The housing has an added grasping fixture (Figure 5) and an internally mounted RFID tag for identification.

The tool changer also includes a calibration lug which can be presented during system intialization. By grasping the calibration lug and complying to interaction forces, the manipulator can periodically align itself with the proper gripping position.

The tool changer must positively retain the tools in position when not accessed by the robot arm. Force required to remove a presented tool must be less than 4.5N (1 pound-force).

Positioning Accuracy The TRS must be able to present tools with position accuracy and repeatability of ±0.65mm and orienatation accuracy and repeatability of ±0.2 deg.

Presentation Error Tolerance A key driving requirement was to tolerate two types of errors in positioning tools during grasp. First, the tools may be presented to the TRS (step 3 above) with incorrect positions and orientations. The tool-position grasper must successfully capture the tool. Second, when the robot initially grasps a presented tool, still held by the tool-position grasper (step 4 above), misalignment between the robot and tool fixture must be tolerated before and during simulataneous grasp by the two devices.

A tolerance analysis was conducted by ORNL of the external system including the Mitsubishi PA-10 arm and grippers. The analysis generated requirements for the TRS to meet in order to capture all presented tools. Table I summarizes the tolerance requirements. Tolerances for displacement and orientation are in all directions.

Positioning Speed The overall Trauma Pod system requirement was to complete an unassisted tool change of the da Vinci robot in 10 seconds or less, with steps as shown in Figure 1. Analysis of the timing budget among the various components and steps involved in the tool change resulted in the requirements for TRS positioning speed given in Table II.

TABLE I
CAPTURE TOLERANCE REQUIREMENTS FOR TOOL RACK SUBSYSTEM.

Parameter	Tol.	Units
Displacement	±4	mm
Orientation	±2.8	deg.
Stiffness	5.7-7.8	N/mm
Torsional Stiffness	0.8	Nm

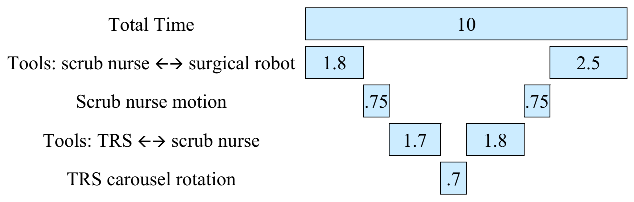


Fig. 1. Timeline for a 10 second tool change.

TABLE II
TRS SPEED REQUIREMENTS. SPEC IS EQUIVALENT TO AN AVERAGE SPEED OF 257°/SEC.

Parameter	Requirement	Units
Grasper Open/Close	100	ms
Worst Case Position Change	700	ms

RFID The TRS must be able to read from RFID tags embedded in the tools and to write a tool-identification string into the tags in response to commands over the network.

C. Software

Operating Modes The TRS software was required to run in three modes:

- *Emulator mode.* In this mode, all actuator motion is simulated by time delays. A system wide emulator testing facility was operated to test software interactions among the subsystems during development.
- *Real mode.* The operational mode of the system in which all functions are physically performed as specified when invoked over the network interface.
- *Manual Mode.* Individual functions can be invoked manually through a GUI presented over the network in this mode.

Communication In Real and Emulator modes, the Trauma Pod system architecture dictated that the TRS must respond to commands through a hierarchy of two levels of communication protocols.

The *Spread* protocol[4], [5] is used to distribute packets of information among subsystems of the Trauma Pod system. Spread is a peer-to-peer packet distribution scheme for local or distributed multiprocessing. Each spread user process can subscribe to one or more channels. Spread user processes that send a packet can optionally block until reception by all subscribing users is confirmed.

A set of *XML schemas* were designed for command and status packets to be exchanged between the subsystems and a central planning computer. From the full set of 60, the TRS had to support the following message types, each described by an XML schema.

- NodeIf - command messages related to the NodeState of the system
- NodeMonitorIf - periodic NodeState monitoring messages
- AlarmMonitorIf - alarm event messages
- TRSIf - messages related to the TRS only
- TRSModelIf - periodic messages describing the physical state of the system (carousel and gripper position)
- InventoryIf - messages for inventory updating and reporting

TRSIf contains the following message types:

- TRSSelectToolCmd - locate and present a tool of the requested type. The tools are requested by serial number as determined by the RFID tags and assignment of tools to rack slots was determined locally within the TRS.
- TRSSelectToolRsp - indicate that the tool has been presented. Also returns a data structure containing tool data such as RFID, and sterility status.
- TRSSelectEmptySlotCmd - locate and present an empty bay.
- TRSSelectCalibrationLugCmd - present the calibration lug.
- TRSAcquireToolCmd - close the grippers on a tool placed in the TRS.
- TRSAcquireToolRsp - indicate the TRS has grabbed a tool.
- TRSSurrenderToolCmd - open grippers, allowing a tool to be removed.
- TRSSurrenderToolRsp - indicate the TRS has opened its grippers.
- TRSGotoBayCmd - lower-level command to send the TRS to a requested bay.
- TRSSetGripperStateCmd - lower-level command to open or close the grippers.

III. Design Features

The completed TRS was required in the early phases of the Trauma Pod project and thus was developed in a single design-build-test-deploy cycle in 2005 and early 2006 lasting roughly nine months. The physical architecture is a base unit holding a detachable carousel on a rotary spindle. Each tool position has two spring-closed graspers holding the tool shaft for positive retention of the tools (necessary during rapid rotation) that are opened by a pushrod mechanism from below the presentation position. The graspers can only be opened via automation for the currently presented tool.

A. Mechanism and Basic Control

Tool Carousel The TRS was designed with 15 tool positions around a carousel 45cm in diameter. One of these positions contains a calibration lug to aid in registration of the Mitsubishi PA-10 arm. The calibration lug is mounted in a compliant suspension. The TRS carousel is a removable unit which is designed to be sterilized in an autoclave. It is entirely made of anodized aluminum, rubber, and stainless steel. By unscrewing a knob at the top of the TRS, the carousel and all attached tools can be lifted off by hand, sterilized, and remounted. All sensors and actuators are below the sterile barrier and do not touch the tools. RFID tags present in the tools can withstand autoclave temperatures.

Actuators We selected Smart Motors from Animatics Inc. for motion control because of the lack of project time for work on low level control and the relatively routine automation-style requirements for fast positioning of a predominantly inertial load. These devices are a single unit containing brushless servomotor, energy conversion, position sensing, and control system in a single package.

The mechanism features a rotating carousel driven by a servo actuator with an attached 30:1 gearhead mounted on a rigid aluminum base (Figure 2). After design of the carousel and tool graspers, the inertia of the carousel was determined to be 687×10^6 g mm^2 when holding 10 tools and the calibration lug. This inertia, and the requirement for moving 168° in 700ms, mandated selection of a relatively large 220 Watt servo motor for carousel rotation having a continuous torque rating of 1.09 Nm and a peak rating of 4.06 Nm.

A second, smaller Smart Motor, fitted with a linear lead screw mechanism, was selected for the tool-position grasper actuator. It is mounted below the tool-presentation position such that it can displace the pushrod upward and open the jaws. The rated speed of this actuator was sufficient to open the jaws with 18mm displacement of the pushrod in less than 100ms.

Control Control of the TRS is accomplished by a software thread that sends commands over a single RS-232 serial port to the two Smart Motors. The RS-232 line is arranged in a loop so that each actuator echoes packets down the chain and back to the PC. These actuators include trajectory generation and control functions so that high level motion commands are all that is required to execute controlled point-to-point motion.

Packets are sent to the Smart Motors at 38,400 bits per second. Packets are 2 to 32 bytes long averaging about 8 bytes per packet. The motor commands are:

- UpdateStatus - request a byte reporting the servo status
- GotoPosition - rotate to specified bay position (calculate desired encoder count, then command servo move to that count)
- Stop - halt all servo motion
- Calibrate - calibrate encoders using optical break-bar

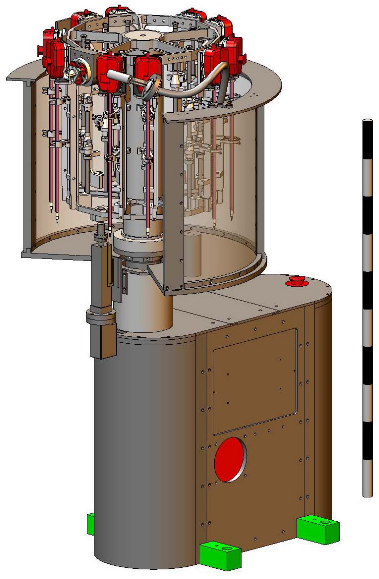


Fig. 2. CAD rendering of the completed Tool Rack Subsystem design. The column (right) is 1m high, divided into 10 equal segments and is included for scale. Overall height of the system is about 1.5m.

sensors

- Unservo - stop holding position, allows for manual (hand) positioning
- AtTargetPosition - return true/false if servo is at commanded position
- GetCalibPhotoInterrupted - low-level function to test calibration photointerrupter

The tool grasping mechanism (Figure 4, left) was reproduced 10 times, leaving 4 tool positions in the carousel open for further development. It consists of a vertical pushrod to which two conical cams are attached. The cams each open one of the two spring-closed tool-grasper jaws. The jaw motion range is designed to accomodate the required misalignment tolerances (Figure 4, right). Torsional stiffness of the closing springs is 0.8 Nm. The pushrods are also fitted with a knob at their top end for easy manual operation of the tool-position graspers.

Misalignment compliance is also designed into the tool-graspers by use of a set of 5 commercial shock mount devices to hold the tool head guide plates (Figure 5). Three of these commercial shock mount units (Lord 106 PDL-1) provide a stiffness of 5.7 N/mm each in the radial direction and two units (Lord J-3112-12-1) provide 7.9 N/mm each in the vertical direction.

B. E-Stop and Power Control Circuitry

E-stop and Power-Cycle circuits were provided to an external connector. These can be activated by remote contact closures so that motor power can be removed at any time and so the system can be completely powered down or up by remote control. A shorting plug was provided to allow local testing operation.

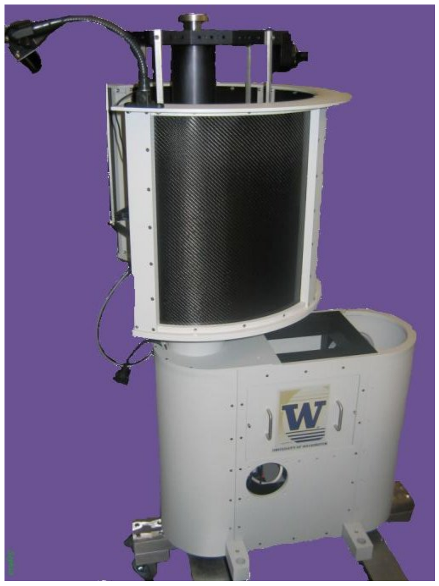


Fig. 3. Photo of completed Tool Rack Subsystem during testing. Graspers are not mounted onto the carousel in this photo. Video camera, mounted on a locking positioning stalk, is added to the frame to allow video monitoring of the tool exchange. Calibration fixture is visible pointing away from the tool presentation position (upper right).

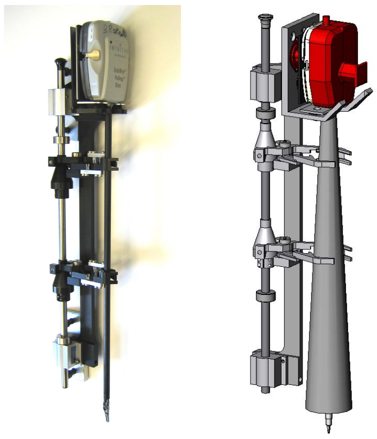


Fig. 4. Left: Photo of one tool-gripper module, removed from the carousel. Two conical cams are driven by a pushrod from the bottom to open the spring-loaded gripping jaws. Right: CAD analysis of grasp capture under angular misalignment of the tool. A 5° conical solid was used to represent the permissable tool misalignment.

The CTR line of a third RS-232 serial port on the computer senses the E-stop condition electrically and puts the software into the E-stop state. The transition into E-Stop state causes the software to send a node-state message to the system.

Should the software fail or system hang, and thus not be able to process the E-stop command, the E-stop signal is also sent to the ”Go” pin of the TRS motors. This pin is programmed to bring the system to an immediate, controlled stop irregardless of serial communication or internal status in approximately 10ms.

The initial E-stop circuit directly removed power to the actuators. If the E-Stop was activated during carousel motion,

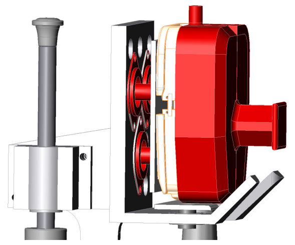


Fig. 5. Compliant elements are used to mount tool mating surfaces to the TRS carousel. Modified tool housing (red) shows gripping fixture projecting to right. RFID tag is mounted inside gripping fixture.

it caused violent deceleration as the brakes were automatically applied at high velocity. The circuit was rewired to interrupt AC power to the actuator power supply instead of DC power to the actuators. This allows the control software about 200ms of additional time after the E-stop event to initiate a controlled deceleration of the carousel before DC power ran out. This modification produces smooth stopping on E-stop but still provides a hardware guarantee of removal of actuator power robust to any software failure.

C. *Software Architecture*

Operating System The TRS computer is a standard 1.0 GHz PC in the mini ATX (Shuttle PC) form factor, integrated inside the TRS chassis. The operating system is standard Fedora Core 5 Linux. Hard real time scheduling features, such as those provided by RTAI Linux, were not required. Software was developed in `C++` with the Qt library used for the manual mode graphical user interface (below).

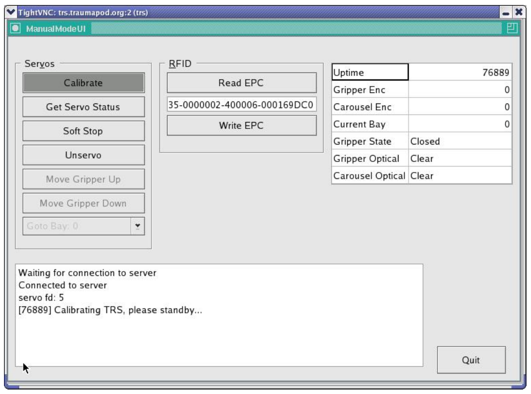
Single Threading A single thread handles actuator control and communication. An endless loop scanning all functions is set with a delay call to repeat at 1000 Hz.

Main Control Process The control part of the loop waits for incoming messages, identifies the message, and evaluates whether or not the message can be performed at that time. If not, a `NAK` is sent back to the source of the command, for example, if the system is in the Halt or E-stop state where motion is not allowed. If it can be, the command is performed.

Communication If a packet is received from Spread, the software identifies it and branches to code appropriate for parsing that message type.

Safety and Health Monitoring When any command is sent to the servos (e.g., at each loop iteration), normally the command is echoed back to the host through the RS-232 physical loop. If the echo fails or is corrupted, the software assumes a hardware fault and triggers an E-stop. Moreover, after a movement, the servo drive is interrogated to verify that it is now at the correct position.

RFID A Pepperl+Fuchs RFID tag reader was installed on the carousel housing 30° from the tool presentation position so that it comes within 10mm of each tool tag as it passes. Transmision power on the RFID reader is limited by design so that it can only interrogate the tag directly opposite the probe. It takes about 0.5 seconds to read an RFID tag. Empty positions take longer for the reader to time out. A software routine moves the carousel through all 15 positions and reads each RFID tag in about 10 seconds so that a complete tool inventory can be collected at any time after calibration.



35-0000002-400006-000169DC0

Uptime	76889
Gripper Enc	0
Carousel Enc	0
Current Bay	0
Gripper State	Closed
Gripper Optical	Clear
Carousel Optical	Clear

```
Waiting for connection to server
Connected to server
servo fd: 5
[76889] Calibrating TRS, please standby...
```

Fig. 6. Graphical User Interface for manual operation of TRS (used primarily for testing).

Manual Mode User Interface Despite its comprehensive network command set, a manual interface was found to be essential for system-level testing and diagnosis. The manual mode interface (Figure 6) allows a remote user to initiate all low-level TRS positioning functions without the Spread or XML network layers.

IV. Performance Testing

There were four phases of testing of the TRS.

1) Emulator software testing at SRI
2) Pre-delivery testing in our lab.
3) Integration with the tool-changing and supply robot at ORNL
4) System integration and testing at SRI.

Emulator testing The emulator was used initially as a development tool to test communications and avoid major interface problems when the real systems would be interconnected. It then evolved into a contemporary validation mechanism as an operational state on each system. Any series of commands can be exectued in emulation mode as a test run to reveal unforseen operational problems.

Each Trauma Pod subsystem contributed a software build to a set of servers operating on a private network for testing in emulation mode. This network could validate the communication protocols without requiring actual hardware. Overnight testing scripts probed each of the required TRS responses. While in emulator mode, the TRS software simulated the time delays for all hardware functions. Emulator testing was passed in February 2006.

Pre-delivery Testing Our initial TRS tests ran the device through testing loops overnight in real mode to measure reliability and repeatability of the complete system. In one test, we ran 1350 movements during the night and recorded movement time attained between each pair of positions.

The result showed worst case movement time of 648±8ms (Figure 7). The times to release and grasp the tool were measured at 88ms and 76ms respectively.

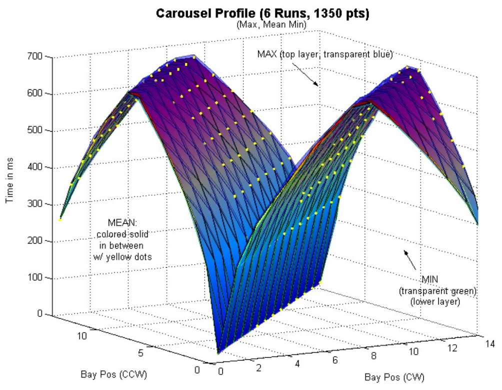


Fig. 7. Movement time measured between each possible pair of tool positions during overnight tests (n=1350). Worst case movement time was 648 ± 8 msec.

We measured tool capture tolerance by a static bench-top setup in which known static displacements were applied to tools and correct capture was verified. Due to the actuated grasp design with spring closure, no force was required to place the tool into a tool position. In normal operation, tools will only be removed with the jaws open. We measured tool removal force with the jaws closed in order to calculate the top speed at which centripetal forces would throw the tool out of the jaws. Tool removal force with the pincers closed was 12N, well above the requirement. Positioning repeatability was found to be ±0.177 deg.

System Integration at ORNL and SRI These testing phases included about two weeks of onsite support from our team at each location. Despite careful packaging, remedial engineering work was required at both locations to repair shipping damage to both purchased and fabricated components, including subtle damage to the computer motherboard. Robotic transfers of tools were performed at increasing speeds until the original specifications were achieved.

V. Discussion

The TRS design met its specifications and the working device was delivered and integrated into Trauma Pod. System level integration and evaluation was successfully completed.

In retrospect it should have been obvious that there is a relationship between grasping compliance and motion speed. Compliance was required in the tool graspers in order to accomodate error in the tool presentation position and orientation. However during rapid acceleration, this caused some tool motion and associated rattling noises. Although the tools were still retained in position, the noise was judged unaesthetic and speed requirements were subsequently reduced. Although it would be possible to design a clamping device which mechanically eliminated compliance during carousel motion, it seems more likely that operating experience will permit a less conservative error tolerance specification and allow higher stiffness in the shock mounts and grasper closing springs.

Another issue was the shaft coupling for the servo actuators. Despite our initial calculations, set screws and thread locking adhesive were not sufficient to keep the carousel drive spindle from slipping due to high torques on acceleration and deceleration. As a result, we modified the motor shaft to include two flats for extra set screws which solved the problem.

The TRS chassis had a very rigid frame designed to maintain alignment of the motion axes despite rough handling. However the commercial-grade computer was mounted inside with rigid holdown straps. Although the TRS chassis was successfully shipped accross the US, the computer inside sustained damage and was unreliable after shipping. Internal shock mounts for sensitive components should be considered in future designs.

There are many system-level lessons which are now being digested as Phase I of Trauma Pod nears completion. However the immediate focus of this paper, on the Tool Rack Subsystem, leads to the following conclusions for development of a future device:

- Resolve conflict between tool-holder compliance and motion specs.
- Reduce overall system size and weight.
- Use operating experience to reduce design conservatism.
- Consider a linear tool array to save floor space and allow wall mounting.
- Consider integration of tool rack with supply dispenser.

Acknowledgments

The authors would like to acknowledge support from DARPA and US-Army TATRC grant number W81XWH-05-2-0029, our colleagues at SRI, Oak Ridge National Labs, and the entire Trauma Pod team. We also acknowledge technical contributions from Brandon Martin-Anderson.

References

[1] R.M. Satava. Disruptive visions: The operating room of the future. *Surg. Endoscopy*, 17, 2003.

[2] M.R. Treat, S.E. Amory, P.E. Downey, and D.A. Taliaferro. Initial clinical experience with a partly autonomous robotic surgical instrument server. *Surg. Endoscopy*, 20, 2006.

[3] Fujio Miyawaki, Ken Masamune, Satoshi Suzuki, Kitaro Yoshimitsu, and Jüri Vain. Scrub nurse robot system - intraoperative motion analysis of a scrub nurse and timed-automata-based model for surgery. *IEEE Transactions on Industrial Electronics*, 52(5), Oct 2005.

[4] Y. Amir and J. Stanton. The spread wide area group communication system. Technical Report CNDS 98-4, Johns hopkins University, Department of Computer Science, 1998.

[5] Y. Amir, Y. Kim, C Nita-Rotaru, J.L. Schultz, J. Stanton, and G. Tsudik. Secure group communication using robust contributory key agreement. *IEEE Trans. Paralell and Distributed Systems*, 15(5), 2004.