**Proceedings of the Second IASTED International Conference**
**HUMAN-COMPUTER INTERACTION**
**March 14–16, 2007, Chamonix, France**
**ISBN Hardcopy: 978-0-88986-654-6 / CD: 978-0-88986-655-3**

# HUMAN COMPUTER INTERFACES OF A SYSTEM FOR ROBOTIC HEART SURGERY

H. Mayer, I. Nagy and A. Knoll
Robotics and Embedded Systems
Technical University Munich
Boltzmannstr. 3
85748 Garching, Germany
email: {mayerh|nagy|knoll}@in.tum.de

E.U. Braun and R. Bauernschmitt
Department of Cardiovascular Surgery
German Heart Center Munich
Lazarettstr. 36
80636 Munich, Germany
email: {brauneva|bauernschmitt}@dhm.mhn.de

**ABSTRACT**
In this paper a system for robotic heart surgery is presented, which provides force feedback and semi-automatic performance of certain surgical tasks. In order to operate this system, a variety of human-computer interactions are necessary for which we have implemented and evaluated corresponding interfaces. Since we are dealing with a telepresence system, it is essential to provide the user with as much immersion as possible. This comprises at least interfaces for manual input, vision and haptics. The main focus of the following paragraphs will be on an interface for interaction with the robotic system. This interface is based on a keyframing approach like it is used in computer animation. The user can interact with a realistic 3D-model of the scenario. Within this virtual environment it is possible to drag the end-effectors of the system to certain positions and save them as keyframes. Afterwards, different interpolation techniques can be used to accomplish movements from keyframe to keyframe. Since not all configurations within the virtual environment are technically feasible, inverse kinematics and collision detection prevents the user from executing harmful movements. Beside this offline planning of trajectories, the proposed interface is also capable of real-time interactions with the robotic system.

**KEY WORDS**
haptic interface, keyframing, robotic surgery, bi-manual interface, human-robot interface

## 1. Introduction

Endoscopic surgery is a challenging technique for thoracic interventions. Its application is especially expedient in the field of thoracic interventions like heart surgery, because sternotomy or large intercostal cuts can be avoided. Therefore, the collateral surgical trauma of the patients is minimized, which results in quicker recovery of patients. In addition, the time of hospitalization and the infection rate can be reduced. Therefore patients massively profit from this endoscopic treatment option. On the other hand, surgeons have to cope with increasingly complex working conditions, but the design of intuitive user interfaces can help to overcome these barriers. Since endoscopic surgery is performed through a small port or "key-hole" in the patient's chest (cf. fig. 1), surgeons must learn to operate with unfamiliar and often awkward surgical instruments. All movements have to be performed using "P" as fulcrum and visual impressions of the field of operation can only be provided by means of an endoscopic camera. Hence,

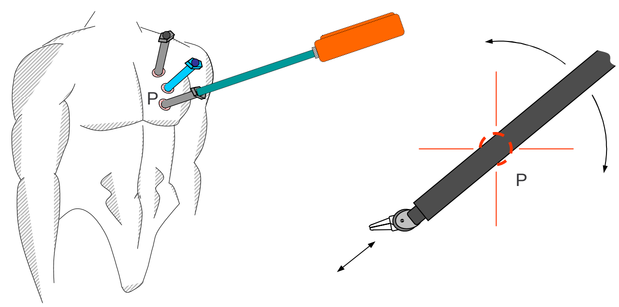


**Figure 1. Location of endoscopic ports for instruments and camera**

the techniques of endoscopic surgery have been applied uncommonly, particularly in the field of heart surgery. An important step to push this technology was the introduction of telemanipulation, which was especially designed to overcome the fulcrum effect of endoscopic instruments. The surgeon no longer operates the instruments directly, but they are driven by a special device with a Cartesian user interface which surgeons can handle as usual, i.e. like instruments for open surgery. Commercial examples for such systems are the *daVinci*$^{TM}$ [1] and *ZEUS*$^{TM}$ systems (the latter has been discontinued). They are good examples of how the proper design of user interfaces can push forward new technologies like minimally invasive and endoscopic surgery. They offer as much freedom of movement as the hand of the surgeon in conventional open surgery, thus providing six degrees of freedom instead of four like conventional endoscopic instruments. In addition, they assist the surgeon with motion scaling, tremor filtering and a stereo vision interface at the input console. Surgeons can now operate with a surgical mechatronic assistant in a comfortable, dextrous and intuitive manner [2, 3]. Despite the obvious potential advantages of robot assisted surgery, most researchers and surgeons in this area agree that the lack of a haptic interface is a crucial drawback of currently available systems [4]. The inability of the operator to sense the applied forces causes increased tissue trauma and frequent suture material damage. The systems are telemanipulators with no Cartesian position control (the control loop is implicitly closed by visual servoing of the surgeon). In addition, it is not possible for users from other fields to program new trajectories for those devices. Therefore, our main research interests are the construction and evaluation of force sensory / force feedback and the development of an easy-to-use interface for trajectory planning. After a short introduction of our hardware and software for force feedback we will focus on the presentation of a novel human-robot interface based on keyframing.

## 2. Previous Work

Since the field of robotic surgery has attracted many researchers, a variety of systems exists with different features implemented by other groups. At the University of California, Berkeley, a robotic system was developed, which has already been used to perform certain surgical tasks like suturing and knot-tying [5]. The Korean Advanced Institute of Science and Technology has developed a micro-telerobot system that also provides force feedback [6]. In Germany two systems for robotic surgery were built at the Research Facility in Karlsruhe [7] and at the DLR in Oberpfaffenhofen [8]. While the first system provides no force feedback, the latter system is equipped with PHANToM$^{TM}$ devices for haptic display. A system that also incorporates force feedback was the previous version of our system at the Technical University of Munich [9]. Regarding human-robot interaction based on keyframes, Hein et al. have proposed a plannig environment for an humanoid football-playing robot, which supports keyframing [10]. A similar approach was proposed by Wama et al. to teach dancing motions to a humanoid robot [11]. Apart from robotic applications, there have been recent endeavors to improve the technical soundness of keyframed computer animations. Two frequently employed techniques are physics engines [12] and collision detection [13].
The ergonomic features of the new user interface, presented below, have been determined by a clinical trial with an earlier version of our system in 2004 [14]. The study involved testing of the previous user interface for its ability to conduct basic surgical procedures. Knot tying, breaking suture material and detection of arteriosclerosis had to be performed in a deˇ ned cycle with double blinding. These tasks imply at least basic knowledge of and familiarity with surgical principles and procedures. The participants, 25 surgeons at different levels of surgical training and age, assessed the work flow and ergonomics of the system. They also evaluated three different levels of haptic feedback: no feedback, standard force feedback and `enhanced' force feedback. During the trials, the critical flicker fusion frequency (CFF) was measured, analyzing the progression of fatigue in between three blocks of tasks using three different degrees of haptic feedback. The most important finding of this study showed that haptic feedback can greatly improve the surgeon's impression of telepresence, resulting in less disruption of suture material and injuries to the tissue. The amount of applied force during surgical manipulation decreased significantly with haptic feedback. The surgeon's subjective sensation of safety and confidence while manipulating with implemented haptic feedback was enhanced compared to operating in a non-haptic environment. Moreover, a decrease of visual stress was detected with haptic feedback.

## 3. Materials and Methods

Our new version of the system is named ARAMIS for Autonomous Robot Assisted Minimally Invasive Surgery System. Four small robotic arms are mounted on an aluminum framework (see figure 2). Although there are four robots, it is easy to access the workspace due to the ceiling mounted setup. The arms are equipped with force-feedback instruments and an endoscopic stereo camera system (details on the user interface, kinematics and force feedback are outlined below).

**Figure 2. System setup: Four ceiling mounted robots with three instruments and an endoscopic stereo camera attached. Size of the system (including gantry) is approx. 2.5m x 5.5m x 1.5m.**

### 3.1 User Interface

We have developed a human-robot interface where positions and orientations of the robotic manipulators are controlled by two PHANToM$^{TM}$ devices, Sensable Inc. This device is available in different versions with different capabilities. We have chosen the version Premium 1.5 which provides a $20 \times 25 \times 40$ cm workspace that is large enough for surgical procedures. The user controls a stylus pen that is equipped with a switch that can be used to open and close the micro-grippers. The most interesting feature of the employed PHANToM™devices is their capability of displaying forces to the user. Forces are fed back by small servo motors incorporated in the device. They are used to steer the stylus pen in a certain direction. This creates the impression of occurring forces, while the user is holding the pen at a certain posture. Our version of the PHANToM™device can display forces in all translational directions, while no torque is fed back. In order to be able to display realistic forces during operations, we have equipped the instruments with force sensors.

Since the shaft of the surgical instrument is made of carbon fiber, force sensors have to be very sensitive and reliable. Therefore we decided to apply strain gauge sensors, which are employed in industrial force registration. For efficient telemanipulation, it is critical to have a 3D-interface providing a clear view of the operating area. In order to allow for such a feature, we equipped an additional robot with a 3D endoscopic camera. Like the surgical instruments, this camera can also be moved by means of trocar kinematics and can either be actively controlled by the operator or automatically tracked by the system. We have evaluated four options for displaying stereo camera images:

1. Head mounted display (HMD).
2. Alternately displaying left and right images on a CRT-screen. In this case the operator has to wear shutter glasses, which are triggered by the output on the screen.
3. Projecting the images on a silver screen with two video projectors. The projectors have to be equipped with polarizing filters which are orthogonally arranged. Observers have to were glasses with an appropriate polarization for the corresponding eye.
4. An optical system with a semi transparent mirror that displays for each eye the corresponding camera view.

The last configuration has been realized in our master console since it provides the best option for surgeons.

### 3.2 Planning Interface

Apart from the manual interface, described above, our system also comprises an interface for offline and real-time trajectory planning. The central part of this tool is a virtual emulation of the system where the user can easily manipulate its state. In fig. 3 a robotic arm of the system is selected. Items in the scene can either be selected by directly clicking on them or by choosing them from the scene browser on the upper right side of the GUI. The scene browser can be used as a basic CAD program. It is possible to insert new primitives (like cones, spheres , cubes etc.) or VRML objects, e.g. an endoscopic instrument. If those are selected, a context menu for the corresponding object is displayed. Therefore in fig. 3 a corresponding context menu to adjust the different joints of the robot is displayed. Each context menu of an object contains functionality to translate or rotate the object. With the help of the scene browser it is also possible to aggregate objects to groups which can be manipulated on their part. It is also possible to move objects in the hierarchy of the scene graph or to remove them completely. In addition, it is possible to copy and paste objects in order to reuse preassembled parts. The scene graph or parts of it can be stored to disk in order to store them safely. This provides an intuitive interface for users to manage different scenes and make certain modifications. All operations on the scene graph are implemented by means of the open source Coin 3D interface from Systems in Motion AS. This is a high-level graphics language based on OpenGL.

The GUI provides different modes for interacting with the

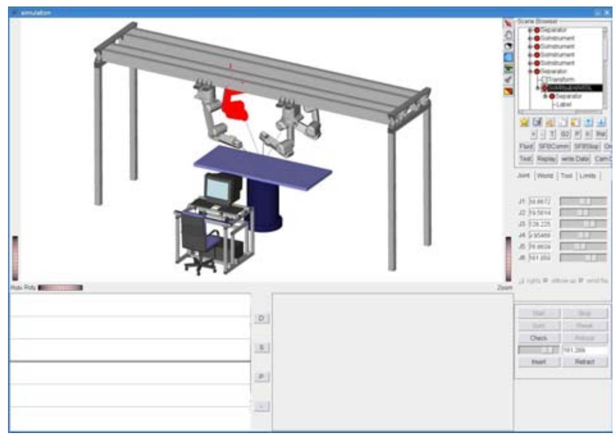

**Figure 3. Graphical User Interface**

robots or surgical instruments. As mentioned above, the robots can be moved by means of sliders; one for each joint of the robot. In addition, the robots can be moved in Cartesian space, i.e. linear translations in x,y and z direction and corresponding rotations about these axes. After the configuration of the robot has been determined, Cartesian movements will be mapped onto joint angles by inverse kinematics. The same applies to the minimal invasive instruments which require a special inverse kinematics if moved in Cartesian space (so-called port kinematics [15]). Port kinematics arranges for movements of the instrument about a small incision in the patient's body and is indispensable for robotic applications in endoscopic surgery. Since each instrument is linked to a dedicated robot, any movement that changes the position of the instrument's base will consequently induce corresponding movements of the robot where the base is attached to. So far, we can use this interface to move the robots or instruments from one posture to another. This can either be executed in real-time or offline. In realtime mode, the robot directly follows the movements that are instructed by the GUI sliders. Since this is quite dangerous (particularly in Cartesian mode: small slider changes can result in wide-ranging robot movements), we have disabled this feature during normal operation. In contrast, offline operation provides more safety. After adjusting the posture of the robot by the sliders in offline mode, the robot will not move until the user has acknowledged the new stage. For more sophisticated trajectories, as they may occur in robotic knot-tying, this kind of interface for point-to-point movements will not suffice. Therefore we have developed a planning interface based on keyframing. Speaking of keyframing regarding trajectory planning, the robots are moving to certain consecutive positions which are saved as keyframes. Afterwards we apply a certain policy (e.g. linear or spline interpolation) to generate all other points of the trajectory that lie between those keyframes (see fig. 4). There are two different modes of moving

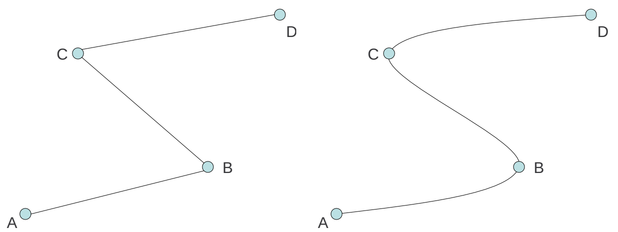


**Figure 4. Linear and spline interpolation between keyframes**

on a trajectory with keyframes: one is to stop at each keyframe, the other, more complex possibility is to make an continuous movement through all keyframes between start and end. Being the most difficult possibility, we restrict ourselves to the description of continuous movements via spline interpolation. We have to take into account that every robot needs a certain time for acceleration after starting and for deceleration before stopping. Otherwise it is not possible to achieve jerkfree motions. Another prerequisite for our application is, that keyframes occur at certain points in time which have to be met exactly. I.e. if the robot starts in point A (cf. fig. 4) it will first accelerate to a certain speed which depends on the time when point B has to be reached. Accordingly there is a fixed time to move from B to C. Therefore, the robot will have to adapt its speed after leaving point B. For calculating the speed during acceleration and decelerations, we have employed the functions $v_a(t)$ and $v_b(t)$, respectively:

$$v_a(t) = \frac{v_c}{1 + e^{n(1-\frac{2t}{t_a})}} \qquad v_d(t) = \frac{v_c}{1 + e^{-n(1-\frac{2t}{t_d})}} \qquad (1)$$

Those are sigmoid functions shifted along the positive $t$-axis. The factor $n$ changes the acclivity of the curve, which reaches its maximum at $t = \frac{t_a}{2}$ (see fig. 5). The time needed for acceleration and deceleration is denoted as $t_a$ and $t_d$, respectively. Another nice feature of these functions is, that the area underneath the curve (i.e. the traveled path) amounts to $\frac{1}{2}t_a v_c$. Therefore, we can easily determine the residual speed $v_c$, given a certain path length and frame time (the same holds for deceleration). Determining the path length is easy for linear keyframe interpolations, but this yields the adverse effect, that trajectories will show

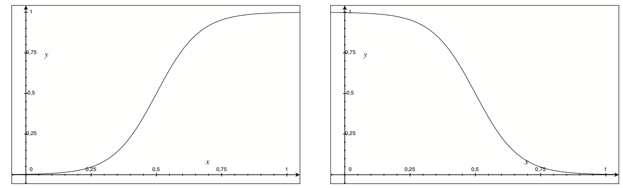

**Figure 5. Sigmoid functions for acceleration and deceleration**

discontinuities at keyframe positions (cf. fig. 4 left side). In order to overcome this disadvantage, it is favorable to apply a spline interpolation to these points (cf. fig. 4 right side). We have chosen Hermite splines for interpolation. They assure smooth transitions at intermediate keyframes. Given a starting point $\overrightarrow{a}$ and an end point $\overrightarrow{b}$ and the corresponding tangent vectors $\overrightarrow{a}'$ and $\overrightarrow{b}'$, each intermediate point can be determined by

$$p(s) = h_1(s) \cdot \overrightarrow{a} + h_2(s) \cdot \overrightarrow{b} + h_3(s) \cdot \overrightarrow{a}' + h_4(s) \cdot \overrightarrow{a}' \qquad (2)$$

where the Hermite weightings are calculated as follows:

$$h_1(s) = 2s^3 - 3s^2 + 1 \qquad (3)$$
$$h_2(s) = -2s^3 + 3s^2 \qquad (4)$$
$$h_3(s) = s^3 - 2s^2 + s \qquad (5)$$
$$h_4(s) = s^3 - s^2 \qquad (6)$$

depicted in fig. 6. However, determination of the path

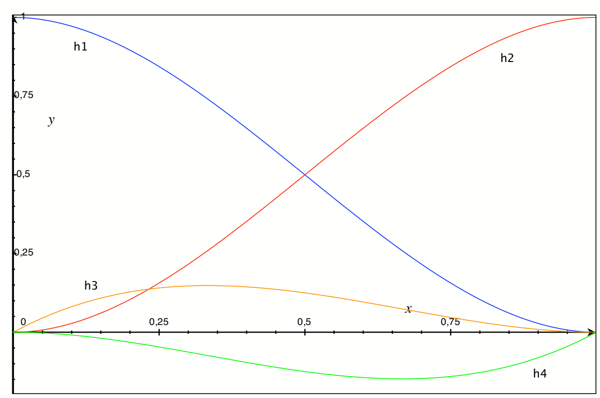


**Figure 6. Weightings for Hermite splines**

length of a spline curve is analytically not feasible. The only possibility would have been peacewise discretization, but this would have yielded inevitable aberrations. Therefore we decided to use another method, the so-called Hermite non-stop interpolation. The basic idea is to interpolate between the tangent vectors and use their length as manipulator speed. I.e. the length of $\overrightarrow{a}'$ is the initial speed of a section while the length of $\overrightarrow{b}'$ is the terminal velocity. Intermediate velocities can be determined by differentiating equation 2:

$$v(s) = \dot{h}_1(s) \cdot \overrightarrow{a} + \dot{h}_2(s) \cdot \overrightarrow{b} + \dot{h}_3(s) \cdot \overrightarrow{a}' + \dot{h}_4(s) \cdot \overrightarrow{a}' \quad (7)$$

where the derivatives of the Hermite weightings with respect to $s$ are determined as:

$$\begin{aligned} \dot{h}_1(s) &= 6s^2 - 6s && (8)\\ \dot{h}_2(s) &= -6s^2 + 6s && (9)\\ \dot{h}_3(s) &= 3s^2 - 4s + 1 && (10)\\ \dot{h}_4(s) &= 3s^2 - 2s && (11) \end{aligned}$$

Using these formulas, it is guaranteed that the next keyframe will be reached at the right time and with the right velocity. I.e. the important parameters for robot movements will be automatically determined by the system. The only thing left to do for the user is to set the keyframes on a timeline. This can be achieved by the interface depicted in fig. 7. With the help of this interface, the user

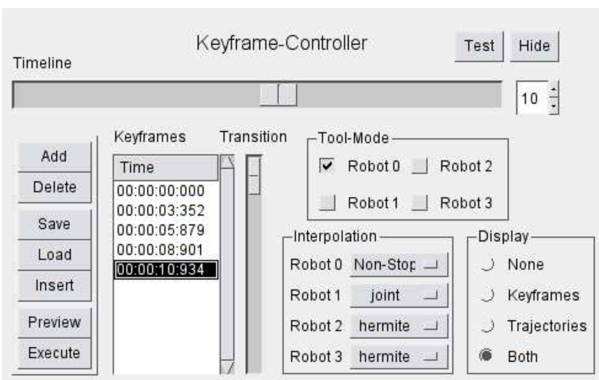


**Figure 7. GUI element for keyframe adjustment**

can add and delete keyframes on a timeline. The time for the keyframe can also be altered afterwards by means of the timeline slider. The user can select which robot should be integrated into the program and how the corresponding trajectory will be interpolated. In addition, trajectories can be stored and saved, and it is even possible to insert stored trajectory parts into new ones. All features can be displayed in the simulation environment and a preview of the programmed trajectories is possible. Anyway, two bad things still can happen. The first is, that a robot exceeds its speed limit, because the user has planned too little time for its movement. In this case the user gets an error message which recommends to extend the time between the coresponding keyframes. The other issue is, that instruments or robots may collide if the planned trajectory is executed. In order to avoid this, all trajectories are processed by an inverse kinematics and collision detection unit. If collisions occur, it is not possible to execute the trajectory before it is safely replanned. In order to speed up this complex computation, the complete inverse kinematics of the robot can be performed on the graphical processing unit (GPU) of a graphics card. Therefore, the desired Cartesian positions

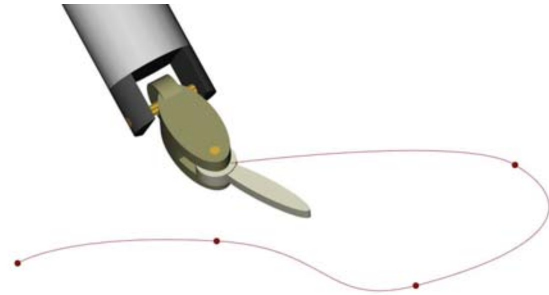

**Figure 8. Display of the keyframes and the calculated trajectory in the planning environment**

are transferred to a texture on the graphics card. Then, a so-called fragment shader, which is written in the OpenGL Shading Language and contains the algorithm for inverse kinematics, is applied to the texture. The results (the joint angles of the robot) are transferred to the framebuffer of the card and can be moved back to the CPU. Afterwards the angles can be directly applied to the robot. Employment of a GPU (NVIDIA GeForce 5200 graphics card) leads to a computation time, which is 20 times faster in comparison with a modern CPU (Athlon64 2200 MHz).

## 4. Conclusion

In the above sections an experimental system was presented for endoscopic surgery which poses a variety of challenges for the design of user interfaces. Like comparable systems, it features an input console with 3D display and the possibility for user input in six degrees of freedom for each hand. In addition, we have implemented force measurement and feedback. The focus of the work presented here was on the design of a novel planning interface for endoscopic surgery. The main criterion was its usability by users from other fields (i.e. users from another domain outside robotics). We have adopted a keyframing approach like it is known from computer animation. All relevant robot parameters which are difficult to determine will be calculated automatically. The operator can concentrate on task-specific work, e.g. planning the trajectory for endoscopic knot-tying. We hope that this work will simplify the handling of complex technical systems and hopefully will be adopted by programmers from other fields, e.g. industrial robotics. At least in the field of robotic surgery, this technique has helped to increase the acceptance of robots in the operating room by surgeons. In the near future we will provide the possibility to integrate preoperative, patient centered image data like CT or MR scans.

## Acknowledgment

This work is supported by the German Research Foundation (DFG) within the Colloborative Research Center SFB 453-I4 on "High-Fidelity Telepresence and Teleaction" and by the German Heart Center Munich.

## References

[1] G.S. Guthart, & J.K. Salisbury, The Intuitive™ Telesurgery System: Overview and Application, *Proc. of IEEE International Conf. on Robotics and Automation*, San Francisco, CA, 2000, 618-621.

[2] V. Falk, S. Jacobs, J. Gummert, & T. Walther, Robotic coronary artery bypass grafting (CABG): the Leipzig experience, *The Surgical Clinics of North America*, *83*(6), 2003, 1381-1386.

[3] V. Falk, S. Jacobs, J. Gummert, T. Walther, & F.W. Mohr, Computer-enhanced endoscopic coronary artery bypass grafting: the da Vinci experience, *Seminars in Thoracic and Cardiovascular Surgery*, *15*(2), 2003, 104-111.

[4] M. Mitsuishi, S. Tomisaki, T. Yoshidome, H. Hashizume, & K. Fujiwara, Tele-micro-surgery system with intelligent user interface, *Proc. of IEEE International Conf. on Robotics and Automation*, San Francisco, CA, 2000, 1607-1614.

[5] M. Cavasoglu, W. Williams, F. Tendick, & S.S. Sastry, Robotics for Telesurgery: Second Generation Berkeley/UCSF Laparoscopic Telesurgical Workstation and Looking towards the Future Applications, *Industrial Robot, Special Issues on Medical Robotics*, *30*(1), 2003, 22-29.

[6] D.S. Kwon, K.Y. Woo, S.K. Song, W.S. Kim, & H.S. Cho, Microsurgical Telerobot System, *Proc. of IEEE/RSJ International Conf. on Intelligent Robots and Systems*, Victoria B.C., Canada, 1998, 945-950.

[7] U. Voges, E. Holler, B. Neisius, M. Schurr, & T. Vollmer, Evaluation of ARTEMIS, the Advanced Robotics and Telemanipulator System for Minimally Invasive Surgery, *Proc. of the IARP 2nd Workshop on Medical Robotics*, Karlsruhe, Germany, 1997, 137-148.

[8] R. Konietschke, T. Ortmaier, H. Weiss, & G. Hirzinger, Manipulability and Accuracy Measures for a Medical Robot in Minimally Invasive Surgery, *Conf. on Advances in Robot Kinematics*, Sestri Levante, Italy, 2004, 191-198.

[9] I. Nagy, H. Mayer, A. Knoll, E.U. Schirmbeck, & R. Bauernschmitt, The Endo[PA]R system for minimally invasive robotic surgery, *Proc. of the IEEE/RSJ International Conf. on Intelligent Robots and Systems*, Sendai, Japan, 2004, 3637-3642.

[10] D. Hein, & M. Hild, Simloid: Research on Biped Robots Controller Using Physical Simulation and Machine Learning Algorithms, *Concurrency, Specification and Programming*, *1*, 2006, 143-151.

[11] T. Wama, M. Higuchi, H. Sakamoto, & R. Nakatsu, Realization of Tai-chi Motion Using a Humanoid Robot - Physical Interactions with Humanoid Robot, *Proc. of the 14th International Conf. on Artificial Reality and Telexistence*, Seoul, Korea, 2004, 71-74.

[12] S. Nourian, X. Shen, & N. Georganas, Role of Extensible Physics Engine in Surgery Simulations, *IEEE International Workshop on Haptic Audio Visual Environments and their Applications*, 2005, 61-66.

[13] J. Nebel, Keyframe Interpolation with Self-Collision Avoidance, *Conf. on Computer Animation and Simulation, Eurographics Animation Workshop*, Milano, Italy, 1999, 77-86.

[14] R. Bauernschmitt, E.U. Schirmbeck, A. Knoll, H. Mayer, I. Nagy, N. Wessel, S. Wildhirt, & R. Lange, Towards robotic heart surgery: Introduction of autonomous procedures into an experimental surgical telemanipulator system, *The International Journal of Medical Robotics and Computer Assisted Surgery 1*(3), 2004, 74-79.

[15] H. Mayer, I. Nagy, & A. Knoll, Kinematics and Modeling of a System for Robotic Surgery, *Conf. on Advances in Robot Kinematics*, Sestri Levante, Italy, 2004, 181-90.