2009 IEEE International Conference on Robotics and Automation
Kobe International Conference Center
Kobe, Japan, May 12-17, 2009

Planning and Control of a Teleoperation System for Research in Minimally Invasive Robotic Surgery

Andreas Tobergte, Rainer Konietschke, and Gerd Hirzinger

Abstract— **This paper introduces the planning and control software of a teleoperation system for research in minimally invasive robotic surgery. It addresses the problem of how to organize a complex system with 41 degrees of freedom as a flexible configurable platform. Robot setup planning, force feedback control and nullspace handling with three robotic arms are considered. The planning software is separated into sequentially executed planning and registration procedures. An optimal setup is first planned in virtual reality and then adapted to variations in the operating room. The real time control system is structured in hierarchical layers. Functions are arranged in the layers with respect to their domain and maximum response time. The design is flexible and expandable while performance is maintained. Structure, functionality and implementation of planning and control are described. The prototypic robotic system provides intuitive bimanual bilateral teleoperation within the planned working space.**

I. INTRODUCTION

In minimally invasive surgery (MIS) the surgeon works with slender instruments through small incisions. This leads to several benefits compared to open surgery, including: reduced pain and trauma, reduced loss of blood, shorter hospital stay and rehabilitation time, and cosmetic advantages. The operation through small incisions on the other hand leads to some drawbacks for the surgeon: (a) The instruments have to be moved around the entry point. The intuitive hand-eye coordination gets lost. The entry point furthermore binds two DoF, so that the surgeon looses manipulability and can only work with four DoF per instrument inside the patient. This makes complicated tasks such as suturing very time consuming. (b) The instruments need to be braced at the trocar, which is a little tube in the entry point. The contact forces can therefore hardly be sensed by the surgeon. To overcome the before mentioned drawbacks telesurgery systems are a promising approach. The surgeon uses a teleoperator station with haptic input devices (master) to control the remote telemanipulator (slave). The teleoperating system transfers the surgeon's commands into the patient's body and the surgeon feels interaction forces with the remote environment.

An advanced prototypic system for minimally invasive robotic surgery (MIRS) is developed at the German Aerospace Center (DLR). The system provides force-feedback and in combination with an auto-stereoscopic display allows for a high-grade of immersion of the surgeon into the remote side, thus, regaining virtually direct access to the operating area. A new versatile light-weight robot (MIRO) developed at the Institute of Robotics and Mechatronics is used as an instrument carrier [23], as shown in Fig. 1. It is kinematically redundant with 7 DoF and can be operated position or impedance controlled. The MIRO is adaptable to different applications as its predecessor the Kinemedic, e.g. for positioning of a biopsy needle with a single robot [18].

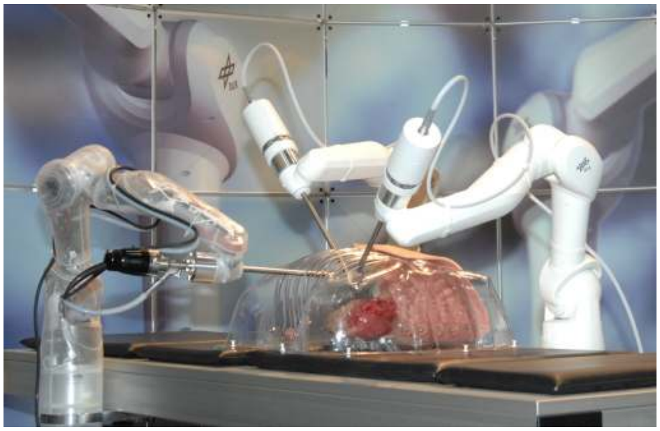

Fig. 1. The remote telemanipulator of the DLR system for minimal invasive robotic surgery, three versatile light-weight robots MIRO with 7 DoF and torque control, two surgical instruments with force-torque sensing attached to the white robot, a stereo endoscope carried by the transparent one.

DLR also developed an instrument that is dedicated to minimally invasive robotic surgery [6]. It has an actuated cardan joint to restore the two DoF lost at the entry point. Therefore the surgeon has full manipulability in six DoF inside the patient. Actuated forceps, which is another DoF, allow for manipulation of tissue. A miniaturized force-torque sensor between the joint and the forceps can measure manipulation forces in six DoF, and the grasping force inside the patient.

The surgeon's workstation (Fig. 2) is equipped with two commercially available haptic input devices omega.7 [9]. They feature seven DoF of which the translational DoF and the grasping are actuated. The rotational DoF are equipped with encoders.

Software design and system integration of such a distributed system with mechatronic devices that are heterogenous, and continously subject to change and development is challenging. The system integrates three robotic arms, two actuated instruments and two haptic devices with all together 41 DoF. It has to be flexible and expandable for researchers. At the time it shall be easily operated for the user which will be the surgeon in the future. This paper presents design

All authors are affiliated with the Institute of Robotics and Mechatronics, German Aerospace (DLR), 82234 Wessling, Germany andreas.tobergte@dlr.de

978-1-4244-2789-5/09/$25.00 ©2009 IEEE

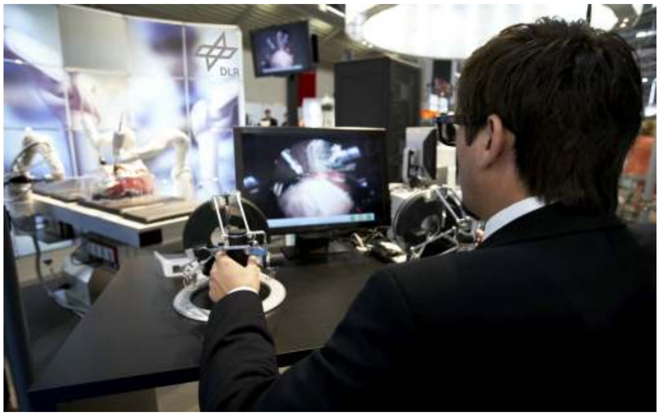

Fig. 2. The teleoperator station for the surgeon with two haptic master devices and a stereo display.

principles and a first implementation of the software system with planning and control.

In Section II the requirements for the planning procedure, and the real time control are defined, and a brief overview of the state of the art is given. The preoperative planning outside the operating room (OR) and the intraoperative refinement is described in section III. A conceptual control architecture for flexible rapid prototyping and details about the current functionality are depicted in section IV. The implementation in software, with results of planning and control, is explained in section V. Section VI concludes the paper and gives an outlook on future work.

II. Requirements and state of the art

A coarse structure of the software is given by the separation in planning and control software, as shown in Fig. 3. These two parts are essentially different because the planning processes functions that respond to user requests whereas the control functions are executed periodically processing actuating variables based on new sensor data or user actions. The interface between the two parts can be unidirectional. The control software requires data of the robotic setup but not vice versa. Setup knowledge is necessary for the interaction of mechatronic devices with each other or the environment, e.g. to avoid collisions and to keep the trocar point. A simple workflow has to be implemented to show the operational system. This workflow includes planning, setup, and the intervention. The workflow as a template can then be refined for several medical applications in MIRS.

A. Planning Procedure

Robotic assistance in minimally invasive interventions provides various advantages as mentioned in the introduction. Concomitantly the overall complexity of the intervention and accordingly the setup time as well as the number of error sources may increase. A preoperative planning (outside the OR) and computer-assisted setup procedure (inside the OR) may overcome these drawbacks. For planning, transparent optimization criteria have to be considered, and individual expertise of the user has to be included. Additionally the software should be usable without robotics knowledge. Preoperative planning is usually based on MRI/CT images of the patient. Intraoperatively, discrepancies might therefore occur due to e.g. soft tissue displacement. These differences have to be taken into consideration. Eventually, the automatically optimized configuration of the robotic arms has to be verified by the user and transfered into the OR. An assisting tool for the alignment of trocar positions and robot bases is inevitable to reduce setup time. Several approaches exist for the preoperative planning of MIRS procedures, mainly tailored to the commercial system daVinci [1], [20], [7], [16]. Most of them however use a trial and error approach to find an optimal setup. Other planning systems rely on performance measures that are not very transparent for the user or disregard collision avoidance or singular configurations. Only [8] considers the complete procedure including the setup in the OR. Setup was however quite time consuming and cumbersome due to missing individual breaks in the passive joints of the daVinci robot. Furthermore, based on the remote center of rotation design of the daVinci robot, a 2 step approach could be chosen to first optimize the robot's passive joints and then the entry point locations. This approach however is not advisable for general robot kinematics [14] and therefore is not adopted here.

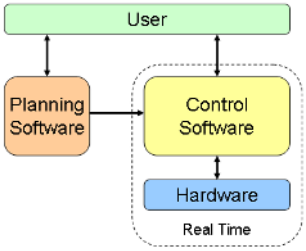

Fig. 3. Interaction of user, planning and real time control with mechatronic hardware.

B. Control Architecture

The control system has to handle different operating modes and various control loops, such as joint control, force feedback control or collision avoidance of the robotic arms. Due to computational limitations, robustness and flexibility the control system has to be distributed on several computers. The control architecture has to allow an efficient execution of control loops and still be flexible and expandable. The system should be easy to modify and adaptable to changing prototypic hardware. Interface definitions are strict but can change over time. Strict definitions are necessary to ensure that the system is successfully operating at any time and to give researchers a frame in which they can develop and experiment. However if interface definitions prove to be insufficient adaption of interfaces or a restructuring of functionality has to be possible. Therefore a conceptual architecture is required that gives a group of researchers a common understanding of the system and allows for rapid prototyping and short innovation cycles.

Common software frameworks provide an implementation of the interprocess communication, e.g. [5] or [21], but without description of the functionality that is implemented. On the other hand specific controller designs, e.g. [15] for teleoperation are limited to a master and slave with one DoF each. The problem of how to partition a system with 41 DoF into control tasks, such as nullspacemotion, bilateral control, and local robot control with kinematic constraints, is not addressed.

III. The Planning Procedure

The DLR planning procedure for MIRS as depicted in Fig. 4 is presented in the following. After preoperative planning in virtual reality (VR), the setup is aligned with the situation in the OR just before the operation (intraoperatively). In case of short notice changes the user can repeat the planning and after the final verification the setup data $\mathbf{S}_{\text{intra}}$ is transfered to the control system.

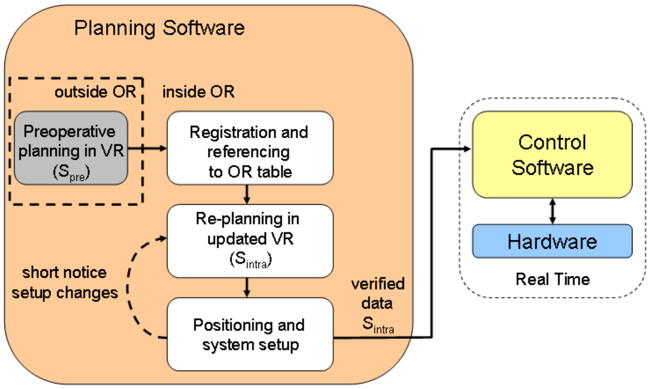
Planning Software
outside OR
inside OR
Preoperative planning in VR (S_pre)
Registration and referencing to OR table
Re-planning in updated VR (S_intra)
short notice setup changes
Positioning and system setup
verified data S_intra
Control Software
Hardware
Real Time

Fig. 4. Phases of the DLR planning procedure for MIRS.

Goal of the procedure is to achieve an optimized setup of robots relative to the patient in the OR. The developed procedure takes into account the robot kinematics and helps to decrease setup times in the OR as well as error sources during the intervention. For the latter, the robot positioning is optimized considering criteria to avoid collisions, singularities and workspace boundaries throughout the operation.

A. Preoperative Planning

Preoperatively, planning is done based on virtual reality and patient data such as segmented CT/MRI images [14]. The user provides details about the operating field inside the patient and the area of possible entry points into the patient[1]. An optimization algorithm that uses in the current implementation a combined Genetic Algorithm and gradient-based method then yields several setups which sufficiently satisfy the optimization criteria throughout the operating field. The optimization method has to be highly configurable to allow for optimization of robot base positions and entry points into the patient. The whole preoperative phase of the planning procedure takes place before the intervention and outside the operating room and, therefore, is less time critical. The result of the planning consists of the data $\mathbf{S}_{\text{pre}}$ as depicted in Fig. 5:

$$\mathbf{S}_{\text{pre}} = \{{}^{\text{world}}_{\text{baseS}}\mathbf{T}_i, {}^{\text{baseS}}_{\text{work}}\mathbf{T}_i, \mathbf{q}_{\text{work},i}, {}^{\text{baseS}}_{\text{app}}\mathbf{T}_i, \mathbf{q}_{\text{app},i}, {}^{\text{world}}_{\text{trocar}}\mathbf{p}_i, {}^{\text{baseS}}_{\text{elbow}}\mathbf{p}_i\},$$

with $i \in \{1,2,3\}$ denoting the respective robot and ${}^{\text{world}}_{\text{baseS}}\mathbf{T}$ the robot base pose. (${}^{\text{b}}_{\text{a}}\mathbf{T}$ defines the frame a in frame b) The center of the robot operating volume is denoted as ${}^{\text{baseS}}_{\text{work}}\mathbf{T}$, with $\mathbf{q}_{\text{work}}$ the corresponding joint angles. An approach position of the robot tool center point (tcp) such that the instrument is aligned with ${}^{\text{baseS}}_{\text{work}}\mathbf{T}$, but completely outside the patient with a safety distance of 5 cm is denoted as ${}^{\text{baseS}}_{\text{app}}\mathbf{T}$, with $\mathbf{q}_{\text{app}}$ the corresponding joint angles. The vectors ${}^{\text{world}}_{\text{trocar}}\mathbf{p}$ and ${}^{\text{baseS}}_{\text{elbow}}\mathbf{p}$ denote the entry position into the patient and a preferred position of the elbow, respectively. In the next steps of the planning procedure, the data has to be adapted from the virtual world to the real situation in the OR.

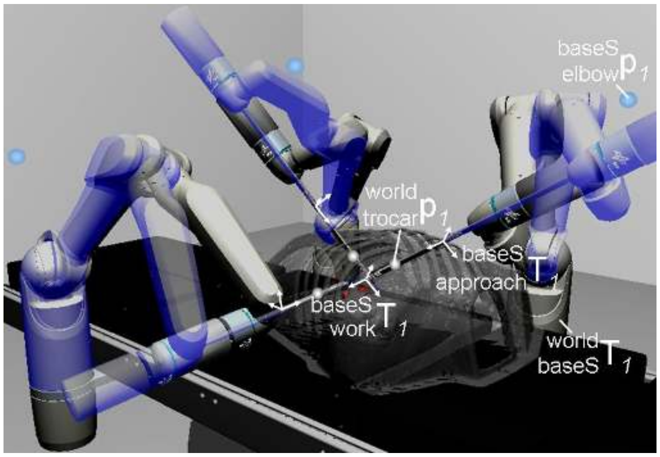

Fig. 5. Result of the planning procedure: The setup parameters for the right robot are shown exemplarily in the figure, the transparent robots are shown in the approach pose from where the user moves the robots through the trocar to the working pose (solid robots).

B. Transfer of planning results into the OR

After placing the patient on the operating table, a registration is performed to align the preoperative image data with the actual patient position. Furthermore, table referencing has to be done, i.e. the table position relative to the patient has to be measured. The medical robots are mounted to the operating table and can be positioned relative to the table only along its direct axis. Since the patient will be in a slightly different pose relative to the OR table than preoperatively planned, the optimal OR setup has to be recalculated taking into consideration the registration and table referencing results. Since good initial solutions are however known from the preoperative planning, this step only takes about 20 s and thus consumes only little of the valuable time in the OR. Eventually, the robots have to be positioned and the trocars set. In case the user decides on short notice to arrange robots or trocars different from the planned configuration, the updated trocar positions or robot

[1] Note that the planning procedure consists of various steps, and that each of these steps may be replaced or refined without compromising the other steps. E.g., the step of defining the operating field inside the patient may be done in a simple VR, directly in the CT slices or in any other planning software. This way, high flexibility and adjustability is achieved.

base poses are measured using an optically tracked probe and fed back to the planning software to calculate new valid data for e.g. $\mathbf{q}_{\text{app}}$ and ${}^{\text{baseS}}_{\text{elbow}}\mathbf{p}$. This way, the complete setup data $\mathbf{S}_{\text{intra}'}$ as realized in the OR is available for the control part described in the following.

IV. Control Architecture

In this section the control architechture of the MIRS-System at DLR is introduced. The control software is based on a signal oriented view. Functional blocks (components) with in and out ports are connected via signals. Signal oriented models are very well suited to closed loop control where periodic execution is necessary. A typical example for an implementation is Matlab/Simulink. Only for non-real time communication with the GUI, a request/reply communication is used. The system model is a static composition of components and connections. Context switches i.e. switching from one step in workflow to another result in different signal routing.

A. The Four Layer Architecture

The signal based control software is organized in different hierarchical layers. A layer is composed of different function based components. All layers communicate only with their neighboring layers or with the user being above the top layer, or the hardware below the lowest one. The architecture aims to satisfy two major goals:

(a) The components of the system are structured according to the demand of execution time. Higher priority is given to lower layers that are closer to the mechatronic hardware. Components in higher layers are less sensitive to delays and can run with lower sampling rates.

(b) The layer structure creates abstraction levels for developers and researchers. The higher the layer the more mechatronic hardware is comprised. On lower layers the level of detail is higher. The hardware is less abstracted.

The four layers from the lowest to highest are:

Layer 1 - Joint control: The joint control layer controls the joint positions and/or torques of a robot. This layer deals with highly non-linear effects such as friction and has to be executed fast with a high sample rate which is 3 kHz in the case of the MIRO.

Layer 2 - Local Cartesian control: In this layer the complete mechanical chains are considered with all joints and their kinematics and dynamics characteristics. A slave system combines a MIRO and an attached instrument, for example.

Layer 3 - Bilateral teleoperation: This layer connects two Cartesian devices to a one arm master-slave system for bilateral teleoperation as shown in Fig. 6. In this layer signals from force-torque sensors are integrated. A rate of about 1 kHz is typically desired in bilateral teleoperation [4].

Layer 4 - Multi arm coordination: The two master-slave systems for the left and the right hand of the user are integrated into a two arm system for bimanual teleoperation. The endoscope robot (disregarded in Fig. 6) that is only operated feed forward and all vision sensors are connected to this layer. In general all components that neither demand high rates nor low latencies are located here.

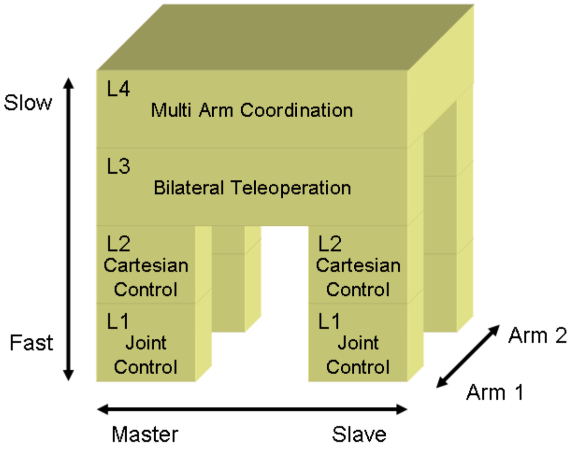

Fig. 6. Four Layer Architecture of MIRS in three dimensions.

The four layer structure clearly prioritizes local control over global control, force over vision and closed loop control over open loop control. It supports rapid prototyping with a team of researchers in a complex distributed system. Abstraction levels are created by grouping functional components without restricting research by strictly specifying interfaces or lowering performance by inefficient execution orders. The three following sections explain the architecture and some components exemplary as implemented. Changes in local or global control can be done while the layers with their abstraction levels remain. The next section describes the operating modes that are related to the MIRS workflow. Details of teleoperation are given with sections about bilateral teleoperation and inverse kinematics calculation.

B. Operating Modes

From the user's point of view the software has to be convenient to handle and must be adaptable to the setup in the OR. To increase the acceptance of the system by surgeons the user should always guide the robot whenever it is in contact with the patient. This can be done by either holding the robot or by remote controlling it. Five steps in the workflow were identified that are executed for all three robotic arms:

Step 1: Prepositioning The robot moves automatically from its initial pose to the approach pose where the instrument or endoscope is close to the human body.

Step 2: Manual Insertion The user guides the instrument through the trocar manually. The user is in full control of the robot's motion by keeping it in his hands.

Step 3: Teleoperation All three robotic arms are inside the human body and the instruments are visible on the stereo-screen of the operator station. The user starts teleoperation by pressing a footpedal to couple the masters and the slaves.

Step 4: Manual Removal The removal of the robotic arms from the patient is the reverse execution of step 2.

Step 5: Initial Positioning After being removed from the patient the robots can move back to their initial positions automatically.

The five steps of the workflow correspond to three basic operating modes in the system: (a) *Positioning*: The slaves move automatically to the patient and back. That is the mode for step 1 and 5, only the target pose changes. (b) *Manual Motion*: The user moves the slave arms with his hands on the robot. This mode corresponds to workflow steps 2 and 4. (c) *Teleoperation*: The user teleoperates the slaves from the master station. The mode is identical to the step 3 in the workflow. The currently implemented model of the Four Layer Architecture is shown in Fig. 7 from the front with functional components and desired frames and vectors that are sent to the mechatronic hardware. Layer 2 on the left belongs to the master. On the right side Layer 1 of the MIRO (left) and the instrument (right) can be seen. Both slave devices are connected to a complete slave system with layer 2. The motor/current controllers for each mechatronic device are shown as Layer 0 and not further regarded in this paper.

The Cartesian impedance controller is used for *Manual Motion* mode. This controller effectively reduces the motor and gear box masses felt by the user with a torque feedback loop in all joints. The cartesian behavior of the end effector can be modeled with a spring and a damper, for details see [19] and [2]. For the *Manual Motion* mode in MIRS it is configured with zero stiffness in translations and high stiffness in the rotations. Therefore the robot's end effector can only move unrestricted in translations. The user can hold the robot with his hands and guide it through the trocar. When entering the trocar two translational DoF are restricted and only motion longitudinal to the trocar is possible.

Positioning mode is implemented with an interpolator commanding a position controller. The MIRO controller implements a state feedback control with motor position control and additional torque feedback for vibration damping of the flexible coupled joints [17]. In *Teleoperation* mode the inverse kinematics sends the desired joint positions $\mathbf{q}_{1-7,d}$ to the position controlled MIRO and $\mathbf{q}_{8-9,d}$ to the instrument.

The alteration of operating modes is modeled with two switches. In *Manual Motion* mode for example, the path of the components *Configure Move Hands On*, *Impedance Control*, *Torque Control* is active, i.e. their out ports are connected to the robot, see Fig. 7. The components on the other paths are only connected with their in ports. They permanently reset their internal states according to the current hardware state, i.e. incoming sensor data from the hardware. This is done in a way that they always provide valid outputs and switching can be done in one discrete time step. Inactive components are always held in a proper initial state. Unsteady behavior that could lead to stability problems is avoided. In the next section components of layer 3 and layer 4 for teleoperation are described.

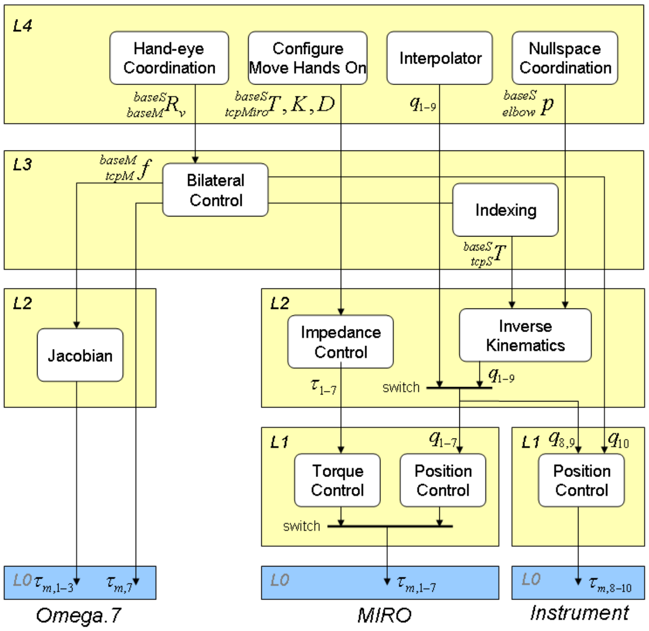

Fig. 7. Frontview of the four Layer Architecture with the master and the slave system consisting of the MIRO and an instrument.

C. Teleoperation

A prerequisite for a surgical teleoperation system is an intuitive hand-eye-coordination that requests a proper projection of the user motion into the remote environment. It is expressed with the virtual orientation of the master relative to the slave. The virtual orientation defines the coupling in teleoperation as contrast to the physical setup in the operation room. The user's display is aligned with the endoscopic camera with the virtual rotation matrix: ${}^{\text{tcpE}}_{\text{display}}\mathbf{R}_v$. Here, the camera focal point is considered the tcp of the endoscope robot ($tcpE$). The orientation of the endoscope in the base frame of the slave arm ${}^{\text{baseS}}_{\text{tcpE}}\mathbf{R}$ changes with motions of the endoscope. Note that slave denotes a robot with instrument and that the calculations in this section have to be done for both slaves separately. The hand-eye-coordination matrix

$${}^{\text{baseS}}_{\text{baseM}}\mathbf{R}_v = {}^{\text{baseS}}_{\text{tcpE}}\mathbf{R} \cdot {}^{\text{tcpE}}_{\text{display}}\mathbf{R}_v \cdot {}^{\text{display}}_{\text{baseM}}\mathbf{R} \tag{1}$$

is given with the physical orientation of the master device relative to the display ${}^{\text{display}}_{\text{baseM}}\mathbf{R}$, the virtual connection of the display with the endoscope and the orientation of the endoscope in the slaves's base. In other words, hand-eye-coordination is the alignment of the haptic channel to the visual channel. The processing of the hand-eye-coordination matrix is not time critical and requires robotic set up knowledge. It is therefore consequently computed in Layer 4 whereas the forward kinematics for the endoscope is computed in Layer 2. The hand-eye-coordination matrix is calculated for the left and the right master-slave arm as shown in Fig. 8.

In bilateral teleoperation a master and a slave robot are connected. Positions, velocities, and forces have to be transformed from master to slave and vice versa. The current version of force feedback is a position-force implementation.

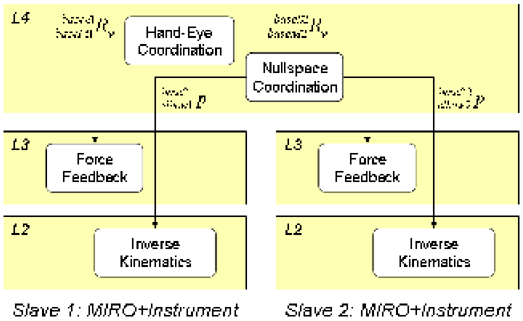

Fig. 8. Sideview of the four Layer Architecture with one slave system on the left and one on the right side.

The measured pose (or derived velocities) from the master are sent to the slave and measured forces are sent back.

The desired tcp of the slave in its base frame

$$ {}^{\text{baseS}}_{\text{tcpS}}\mathbf{T}_d(t) = {}^{\text{baseS}}_{\text{tcpS}}\mathbf{T}(0) + \int_0^t g({}^{\text{baseS}}_{\text{baseM}}\mathbf{R}_v(t), {}^{\text{baseM}}_{\text{tcpM}}\mathbf{v}(t), \mathbf{s}, c(t)) \tag{2} $$

is a function of the initial slave pose ${}^{\text{baseS}}_{\text{tcpS}}\mathbf{T}(0)$ and the master spatial velocity at a time ${}^{\text{baseM}}_{\text{tcpM}}\mathbf{v}(t)$. The velocities are transformed into the slave's base frame with the hand-eye-coordination matrix and scaled with $\mathbf{s} \in \mathbb{R}^6$ and coupled with $c(t) \in \{0,1\}$. The slave is coupled to the master and follows its motions if the user presses the footpedal and the slave does not move out of its workspace. The master automatically decouples when moving out of the slave's workspace and couples in again when moving away from the restricted area. Cartesian workspace limitations are expressed in virtual walls for example. An important limitation is to keep a minimum distance between the trocar point and the tcp to avoid a singularity in the inverse kinematics. The slave system with position controller, inverse kinematics, and indexing (see Fig. 7) can therefore be interpreted as a relative Cartesian slave that allows motions from any initial master pose.

The forces and torques commanded to the master device

$$ {}^{\text{baseM}}_{\text{tcpM}}\mathbf{w}_d(t) = h({}^{\text{baseM}}_{\text{baseS}}\mathbf{R}_v(t), {}^{\text{baseS}}_{\text{tcpS}}\mathbf{w}(t), \mathbf{p}, c(t)) \tag{3} $$

are obtained by transformation of the measured wrench ${}^{\text{baseS}}_{\text{tcpS}}\mathbf{w}$ with the inverse hand-eye-coordination matrix and amplification with $\mathbf{p} \in \mathbb{R}^6$. With (2) and (3) the system can be described as master, slave and two communication channels as shown in Fig. 9. Analysis of stability and transparency for such bilateral teleoperator systems is given e.g. in [15] [10] [11], where the bilateral controller is usually considered as part of the communication channel between master and slave.

D. Inverse Kinematics

The implemented inverse kinematics algorithm to calculate the joint angles $\mathbf{q} \in \mathbb{R}^9$ of a MIRO holding an instrument uses closed form solutions to exactly solve the

- Cartesian condition c_1 to reach the tcp pose ${}^{\text{baseS}}_{\text{tcpS}}\mathbf{T}$, and the

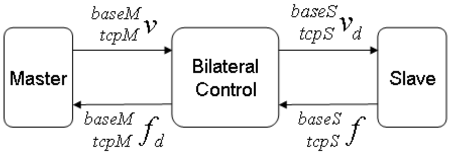

Fig. 9. Bilateral Teleoperation with two communication channels

- Trocar condition c_2 to intersect the instrument with the trocar ${}^{\text{baseS}}_{\text{trocar}}\mathbf{p}$.

The task space that includes the conditions c_1 and c_2 is 8-dimensional with 6 dimensions for the position and orientation of the tool tip and 2 dimensions for the trocar condition. Since the manipulating slaves have 9 DoF, a 1-dimensional nullspace is available for optimization of additional criteria such as joint limit avoidance.

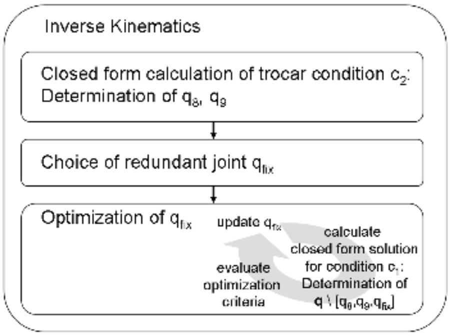
Inverse Kinematics
Closed form calculation of trocar condition c_2:
Determination of q_8, q_9
Choice of redundant joint q_{fix}
Optimization of q_{fix}
update q_{fix}
calculate closed form solution for condition c_1: Determination of $q \setminus [q_8, q_9, q_{fix}]$
evaluate optimization criteria

Fig. 10. Inverse kinematics algorithm with closed form solutions and nonlinear nullspace optimization.

The inverse kinematics algorithm is depicted in Fig. 10. In the first step, the trocar kinematics are solved and yield the joint angles of the articulated instrument q_8 and q_9. In the next step the nullspace angle q_{fix} is chosen based on the current robot pose $\mathbf{q}_{\text{init}}$. This is necessary to avoid algorithmic singularities that might occur when formulating the closed form solution for condition c_1, see [12] for further details. A Levenberg-Marquardt optimization then seeks the best solution in the task nullspace, incorporating the closed form solution of condition c_1. This way, the remaining joint angles $q_{1..7}$ are determined. Avoidance of joint limits and singular configurations as well as minimization of joint velocities and the elbow position itself are considered as optimization criteria. The elbow position criterion minimizes the distance of the robot elbow to the preoperatively planned preferred elbow position ${}^{\text{base}}_{\text{elbow}}\mathbf{p}$ such that collisions outside the patient become improbable. As shown in Fig. 8 the preferred ellbow position can be modified in layer 4 to avoid collisions during teleoperation. Since the task nullspace is 1-dimensional, the criteria are combined using weighting factors. Naturally, this may lead to concurrent goals which necessitates careful tuning of both weighting factors and optimization criterion functions. An advantage of the included closed form solutions is in this context that the conditions c_1 and c_2 are not compromised by the optimization in the task nullspace.

V. Implementation

Planning and real time control of the DLR MIRS system is implemented. The planning procedure is written in C++ on Linux with openGL for virtual reality. The result of the planning procedure is stored in a file that is used by the control system. The control system is developed with Matlab/Simulink and executed on the real time operating system QNX.

A. Planning Procedure

The planning procedure presented in this paper includes the complete workflow from patient specific preoperative planning based on MRI/CT data to the actual setup of the robots relative to the patient in the OR. The preoperative planning is the most time consuming part of the procedure. It takes about 15 min. Since it is done outside the OR, this is not time critical. Use of the software is easy and intuitive. The user just has to mark the operating field and an area for the entry points into the patient in the VR and then gets several proposals for the setup.

Inside the OR, patient registration and replanning take only few minutes. Patient registration is obtained through a surface scan of the upper body using the handheld 3D-Modeller as shown in Fig. 11 (left). A robust feature-based algorithm according to [3] then matches the patient surface with preoperative data. The position of the patient relative to the OR table is measured with the same optical tracking system as used for the 3D-Modeller. Therefore a tracking target is attached to the operating table. To show the calculated positions of trocars and robot bases to the user, the AutoPointer [13] is used: the optically tracked handheld device automatically projects the relevant data onto the patient resp. the OR table as shown in Fig. 11 (right). This way, the OR staff can set up the robots very conveniently. First tests with an experimental setup confirm the potential of the chosen approach. Registration is very robust and works also with incomplete patient scans. In the so far chosen optimized setups, the robots could operate without problems in the considered operating field.

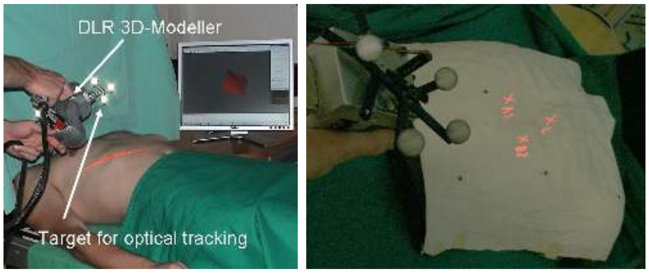

Fig. 11. Patient registration with the 3D-Modeller (left) and positioning with the AutoPointer (right).

B. Control

The control software was developed with Matlab/Simulink and Real Time Workshop for automatic code generation. Simulink enables programming on an abstraction level above source code that suits very well to closed loop control system design. The compiled code runs under the QNX Neutrino real time operating system, and is interfaced with Matlab/Simulink external mode for development and debugging. The executables are distributed on six off-the-shelf PCs with QNX. Interprocess communication is implemented with aRDnet (agile Robot Development, see [5]). The aRDnet software suite implements shared memory and ethernet/udp communication. It extends the Simulink signal flow over a distributed system for rapid prototyping. The control software is distributed over three models running with six instances, as shown in Fig. 12. The joint control Simulink model implements torque, position, and impedance control of layer 1 respectively 2. The executables are running on one PC each and are executed with 3 kHz synchronized on incoming sensor data from the MIROs. A hardware abstraction layer (HAL) provides an interface to the current controllers and the sensors of the robot [22]. The two MIROs holding the instruments communicate over aRD-udp with the force feedback model which integrates the inverse kinematics and the components of layer 3. The joint controllers of the instruments are implemented in hardware. The local control of the master, the omega.7, is provided by the manufacturer. The functionality of the world model implements layer 4 and the inverse kinematics of the endoscope robot. The planning output $\mathbf{S}_{\text{intra}}$ is treated as a set of parameters in the world model. The world and the force feedback models are running with 1 kHz. The roundtrip delay in teleoperation from the slave's force/torque sensor in the tool tip to the master and back to the slave's actuators is currently up to 10 ms, with most delay occurring in the serial RS-485 interface of the instrument that will be updated to the MIRO standard in the future.

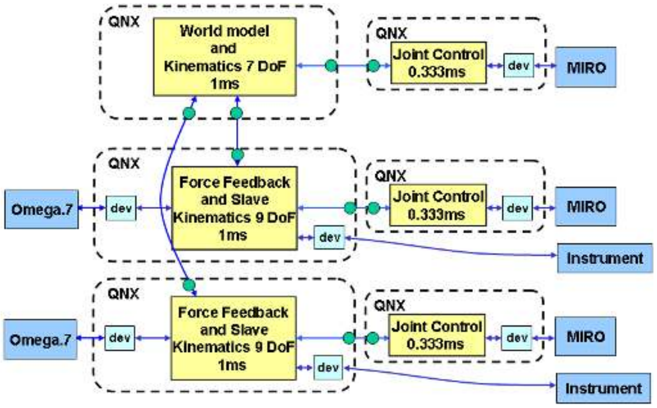

Fig. 12. Distributed Control Software for MIRS.

Start up and shut down is done with shell scripts. The software is expandable and the distribution over three different models leads to a reduced compile time. Collisions and joint limits were successfully avoided. The workflow is easily operated by a QT-GUI. The system provides an intuitive hand-eye-coordination in 7 DoF for each hand. Bimanual bilateral teleoperation with force feedback in 4 DoF per hand was implemented. In experiments with the setup shown in Fig. 1 and Fig. 2 it was possible to tie a knot, while clearly distinguishing the soft contact of a silicon heart and the hard contact when stretching the thread.

VI. Conclusions and Future Works

The paper presents the planning and the control software of the DLR robotic system for research in minimally invasive robotic surgery. The system consists of two essentially different parts, the robotic setup planning and the real time control. The planning is sequentially processed step by step on user requests. It is itself divided in an preoperative planning based on a 3D-patient model and an intraoperative planning that adjusts the setup to the actual situation in the OR after registration. The control software is periodically executed generating actuator variables from sensor data and user inputs. The control system is organized in hierarchical layers that give a functional view of the system. The four layers provide abstraction levels for researchers and priorities for execution. An overview of currently implemented functions is given. The conceptual approach is validated with an implementation of the whole system. It integrates e.g. robotic setup planning, force feedback and null space collision avoidance. The teleoperation system allows for an intuitive tying of a knot within the specified workspace. The user can effectively feel the corresponding forces in his hands when stretching the thread and closing the knot. The presented software structure is a guideline for the integration of future innovations in planning, control, vision and mechatronic hardware design. The system serves as a research platform in MIRS. Future works will include e.g. advanced collision avoidance strategies, different strategies for trocar handling and bilateral control. A challenging research topic will be motion compensation in beating heart surgery. The authors think that even if interfaces and functionality will change, the concept of the pre- and intraoperative planning and the four layers control architecture will remain because it bases on general considerations of clinical workflow and system dynamics.

References

[1] Louaï Adhami. *An Architecture for Computer Integrated Mini-Invasive Robotic Surgery: Focus on Optimal Planning*. Ph.D. Thesis, Ecole des Mines de Paris, Paris, 2002.

[2] Alin Albu-Schäffer, Christian Ott, and Gerd Hirzinger. A Passivity Based Cartesian Impedance Controller for Flexible Joint Robots Part II: Full State Feedback, Impedance Design and Experiment. In *in Proc. IEEE Int. Conf. on Robotics and Automation*, pages 2666–2672, 2004.

[3] Gill Barequet and Micha Sharir. Partial Surface Matching by Using Directed Footprints. In *In Proc. 12th Annual Symp. on Computational Geometry*, pages 409–410, 1996.

[4] C. Basdogan and M. A. Srinivasan. Haptic rendering in virtual environments. In K. Stanney, editor, *Virtual Environments HandBook*, pages 117–134. Lawrence Erlbaum Associates, Inc., 2001.

[5] Berthold Bäuml and Gerd Hirzinger. Agile Robot Development (aRD): A Pragmatic Approach to Robotic Software. In *IEEE/IROS International Conference on Intelligent Robots and Systems*, 2006.

[6] Bernhard Küebler, Ulrich Seibold and Gerd Hirzinger. Development of Actuated and Sensor Integrated Forceps for Minimally Invasive Robotic Surgery. *International Journal of Medical Robotics and Computer Assisted Surgery*, pages 96–107, 2005.

[7] Jeremy W. Cannon, Jeffrey A. Stoll, Shaun D. Selha, Pierre E. Dupont, Robert D. Howe, and David F. Torchiana. Port Placement Planning in Robot-Assisted Coronary Artery Bypass. *IEEE Transactions on Robotics and Automation: Special Issue on Medical Robotics*, October 2003.

[8] Ève Coste-Manière, Louaï Adhami, Fabien Mourgues, and Olivier Bantiche. Optimal Planning of Robotically Assisted Heart Surgery: Transfer Precision in the Operating Room. In *8th International Symposium on Experimental Robotics (ISER)*, Sant'Angelo d'Ischia, Italy, 2002.

[9] Force Dimension. The Omega Haptic Master Device. http://www.forcedimension.com. [Accessed on September 5th, 2008].

[10] Dong-Soo Kwon Jee-Hwan Ryu and Blake Hannaford. Stable teleoperation with time domain passivity control. In *IEEE Trans. on Robotics and Automation, Vol. 20, No. 2*, pages pp. 365–373, 2004.

[11] Septimiu E. Salcudean Keyvan Hashtrudi-Zaad. Analysis of control architectures for teleoperation systems with impedance/admittance master and slave manipulators. In *The International Journal of Robotics Research, Vol. 20, No. 6*, pages pp. 419–445, 2001.

[12] Rainer Konietschke. *Planning of Workplaces with Multiple Kinematically Redundant Robots*. Technische Universität München, Munich, Germany, 2007. PhD Thesis.

[13] Rainer Konietschke, Andreas Knöferle, and Gerd Hirzinger. The Autopointer: A New Augmented-Reality Device for Transfer of Planning Data into the Operating Room. In *Proceedings of the 21st International Congress and Exhibition of Computer Assisted Radiology and Surgery*, Berlin, Germany, June 2007.

[14] Rainer Konietschke, Holger Weiß, Tobias Ortmaier, and Gerd Hirzinger. A Preoperative Planning Procedure for Robotically Assisted Minimally Invasive Interventions. In *3. Jahrestagung der Deutschen Gesellschaft für Computer- und Roboterassistierte Chirurgie (CURAC)*, München, Germany, 8.–9. Dezember 2004.

[15] D. A. Lawrence. Stability and Transparency in Bilateral Teleoperation. *IEEE Transactions on Robotics and Automation*, pages 624–637, 1993.

[16] Roderick C. O. Locke and Rajni V. Patel. Optimal remote center-of-motion location for robotics-assisted minimally-invasive surgery. In *ICRA*, pages 1900–1905, 2007.

[17] Alin Albu Schäffer Luc Le Tien and Gerd Hirzinger. MIMO State Feedback Controller for a Flexible Joint Robot with Strong Joint Coupling. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2007.

[18] Tobias Ortmaier, Holger Weiß, Ulrich Hagn, Matthias Nickl, Alin Albu-Schäffer, Christian Ott, Stefan Jörg, Rainer Konietschke, Luc Le-Tien, and Gerd Hirzinger. A Hands-On-Robot for Accurate Placement of Pedicle Screws. In *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, 2006.

[19] Christian Ott, Alin Albu-Schäffer, Andreas Kugi, Stefano Stramigioli, and Gerd Hirzinger. A Passivity Based Cartesian Impedance Controller for Flexible Joint Robots Part I: Torque Feedback and Gravity Compensation. In *in Proc. IEEE Int. Conf. on Robotics and Automation*, pages 2659–2665, 2004.

[20] Donald L. Pick, David I. Lee, Douglas W. Skarecky, and Thomas E. Ahlering. Anatomic Guide for Port placement for DaVinci Robotic Radical Prostatectomy. *Journal of Endourology*, 18(6):572–575, August 2004.

[21] Scholl, Kay-Ulrich. Modular control architecture 2. http://www.mca2.org/. [Accessed on February 8th, 2009].

[22] Stefan Jörg, Mathias Nickl, and Gerd Hirzinger. Flexible Signal-Oriented Hardware Abstraction for Rapid Prototyping of Robotic Systems. In *International Conference on Intelligent Robots and Systems*, 2006.

[23] U. Hagn, M. Nickl, S. Jörg, G. Passig, T. Bahls, A. Nothhelfer, F. Hacker, L. Le-Tien, A. Albu-Schäffer, R. Konietschke, M. Grebenstein, R. Warpup, R. Haslinger, M. Frommberger, G. Hirzinger. The DLR MIRO: A Versatile Lightweight Robot for Surgical Applications. *Industrial Robot: An International Journal*, pages 324 – 336, 2008.