Planning and Real Time Control of a Minimally Invasive Robotic Surgery System

Andreas Tobergte, Rainer Konietschke, and Gerd Hirzinger

***Abstract*— This paper introduces the planning and control software of a teleoperating robotic system for minimally invasive surgery. It addresses the problem of how to organize a complex system with 41 degrees of freedom including robot setup planning, force feedback control and nullspace handling with three robotic arms. 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 composed of hierarchical layers. The design is flexible and expandable without losing performance. Structure, functionality and implementation of planning and control are described. The robotic system provides the surgeon with an intuitive hand-eye-coordination and force feedback in teleoperation for both hands.**

I. INTRODUCTION AND MOTIVATION

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 enviroment.

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 [20], 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, i.e. for positioning of a biopsie needle [15].

DLR also developed an instrument that is dedicated to minimally invasive robotic surgery [5]. 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.

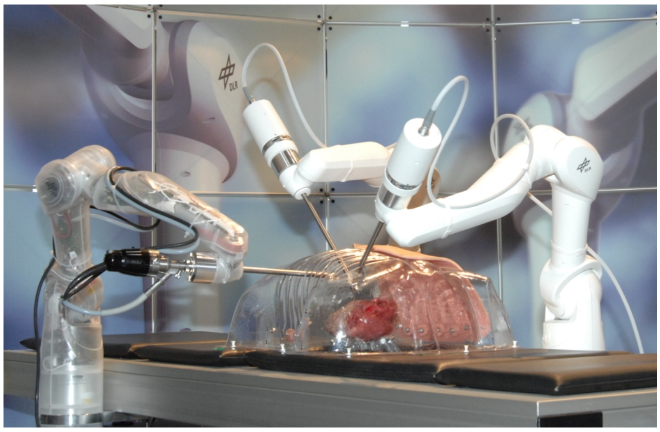

Fig. 1. The remote telemanipulator of the DLR system for minimal invasive robotic surgery, three versatile light-weight robots MIRO with 7 DoF and torque control, two surgical instruments with force-torque sensing, one stereo endoscope.

The surgeon's workstation (Fig. 2) is equipped with two commercially available haptic input devices omega.7 [8]. They feature seven DoF of which the translational DoF and the grasping are actuated. The rotational DoF are equipped with encoders.

Software design for such a distributed system with heterogenous, changing, and developing mechatronic devices is a challenging task. The system integrates three robotic arms, two actuated instruments and two haptic devices with all together 41 DoF. It shall be easily operated for the surgeon but also flexible and expandable for researchers.

In Section II the requirements for the planning procedure, and the real time control are defined, and brief overview of the state of the art is given. The preoperative planning outside the operating room (OR) and the intraoperative refinement is

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

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

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 conludes the paper and gives an outlook on future work.

II. Requirements and State of the Art

The coarse structure of the software is given by the separation in an offline part, for planning and a real time control software, as shown in Fig. 3. The surgeon is responsible for accurately completing the planning procedure outside and inside the OR before starting with the actual surgical intervention. The control software requires data of the robotic setup in the operating room from the planning output. This is necessary to avoid collisions and to keep the trocar point.

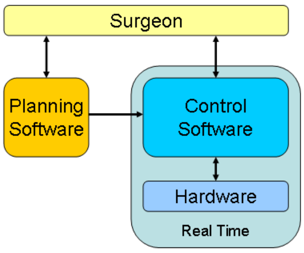

Fig. 3. Interaction of Surgeon, Planning and real time control with Hardware.

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 surgeon 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 surgeon 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 taylored to the commercial system daVinci [1], [17], [6], [13]. 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 surgeon or disregard collision avoidance or singular configurations. Only [7] considers the complete procedure including the setup in the OR. None of the approaches is however adaptable to the robotic system presented in this paper.

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 and robustness 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 flexibel and expandable. The system should be easy to modify and adaptable to changing prototypic hardware. It is clear that strict interface specifications cause restrictions for research. On the other hand unstructured rapid prototyping leads to systems that are hard to maintain. 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 architectures and frameworks such as [18] focus on modularity and implementation aspects. Modularity is achived by the definition of standard objects and interfaces. In this paper the authors promote a functional driven view of the control model. It is adaptable to changes of the mechatronic hardware and new ideas on control. To the authors knowledge there is no robotic system that combines bimanual force feedback with nullspace collision avoidance in MIRS.

From the surgeon'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 surgeon 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 should be executed for all three robotic arms:

Step 1: Prepositioning The robot moves automatically from its initial pose to the approach pose ${}^{\text{base}}_{\text{app}}\mathbf{T}$ where the instrument or endoscope is close to the human body. (${}^{\text{b}}_{\text{a}}\mathbf{T}$ defines the frame a in frame b)

Step 2: Manual Insertion The surgeon guides the instrument through the trocar manually. The surgeon is in full control of the robot's motion by keeping it in his hands. Steps 1 and 2 are executed for the endoscope robot first and for the instrument robots afterwards. The human operator can see the instrument on a screen when coming into the view of the endoscope.

Step 3: Teleoperation All three robotic arms are inside the human body and the instruments are visible on the stereoscreen of the operator station. The surgeon starts teleoperation by coupling the masters and the slaves by pressing a footpedal.

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.

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 surgeon can repeat the planning and after the final verification the setup data $\mathbf{S}_{\text{intra}}$ is transfered to the control system.

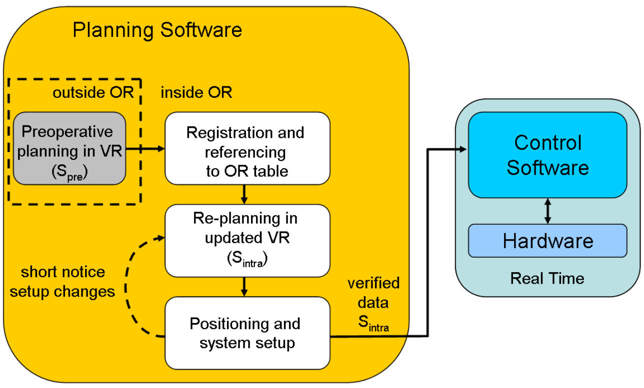

Fig. 4. Phases of the DLR planning procedure for MIRS.

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 [11]. The surgeon provides details about the operating field inside the patient and the area of possible entry points into the patient. An optimization algorithm that uses a combined Genetic Algorithm and gradient-based method then yields several setups which sufficiently satisfy the optimization criteria throughout the operating field. This 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. 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.

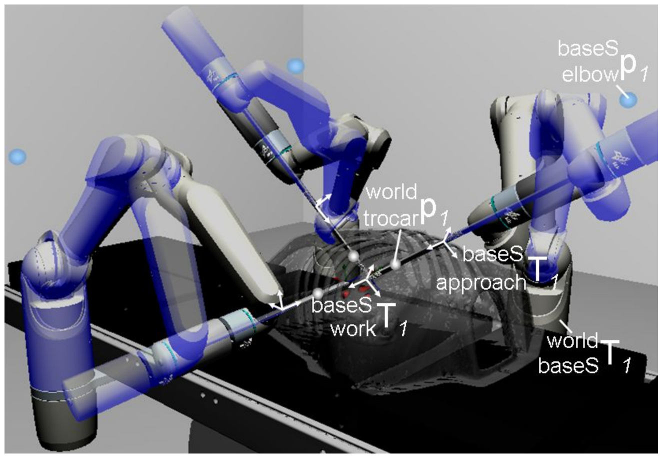

Fig. 5. Result of the planning procedure: The setup parameters for the right robot are shown exemplarily in the figure, the transparent robots are shown in the approach pose from where the surgeon moves the robots through the trocar to the working pose (solid robots).

B. Transfer of planning results into the OR

Patient registration is obtained through a surface scan of the upper body using the handheld 3D-Modeller as shown in Fig. 6 (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.

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.

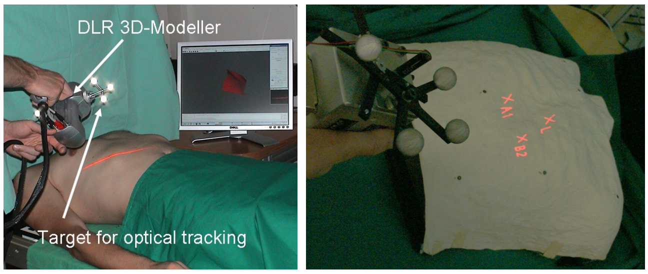

Fig. 6. Patient registration with the 3D-Modeller (left) and positioning with the AutoPointer (right).

Eventually, the robots have to be positioned and the trocars set. To show the calculated positions of trocars and robot bases to the surgeon, the AutoPointer [10] is used: the optically tracked handheld device automatically projects the relevant data onto the patient resp. the OR table as shown in Fig. 6 (right). In case the surgeon decides on short notice to arrange robots or trocars different from the planned configuration, the updated trocar positions or robot 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. This is not an issue because the MIRS-Scenario can be realized with a very simple workflow. Priority was given to closed loop control design, i.e. local controllers, force feedback and collision avoidance during teleoperation.

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 surgeon 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. Componends 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. 7. In this layer signals from force-torque sensors are integrated. A rate of about 1 kHz is typically desired in bilateral teleoperation.

Layer 4 - Multi arm coordination: The two master-slave systems for the left and the right hand of the surgeon are integrated into a two arm system for bimanual teleoperation. The endoscope robot (disregarded in Fig. 7) 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.

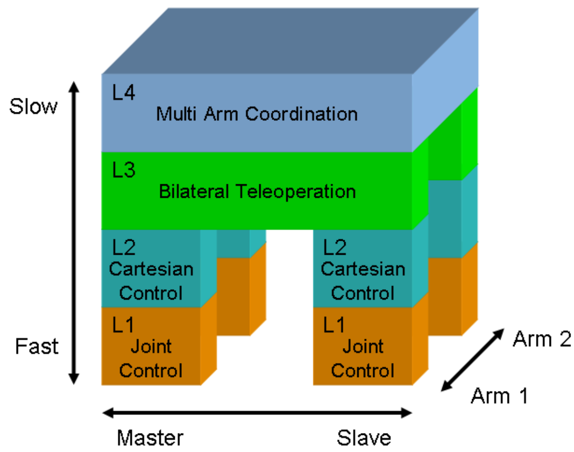

Fig. 7. Four Layer Architecture of MIRS in three dimensions.

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 nest section describes the operating modes that are related to the workflow. Afterwards details of teleoperation are given with sections about bilateral teleoperation and inverse kinematics calculation.

B. Operating Modes

The five steps of the workflow correspond to three basic operating modes in the system: (a) *Positioning*: The slaves move automaticaly to the patient and back. That is the mode for step 1 and 5, only the target pose changes. (b) *Manual Motion*: The surgeon moves the slave arms with his hands on the robot. This mode correponds to workflow steps 2 and 4. (c) *Teleoperation*: The surgeon teleoperates the slaves from the master station. The mode is identical to the step in the workflow. The currently implemented model of the Four Layer Architecture is shown in Fig. 8 from the front. 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. It is configured with zero stiffness in translations and high stiffness in the rotations, for details see [16] and [2]. Therefore the robot has three translational DoF for free motion in space. The surgeon 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. An advantage of this procedure is its robustness with respect to little variations in the setup. Positioning mode is implemented with an interpolator commanding a position controller. The MIRO controller implements a state feedback control with motor position and torque feedback for flexible coupled joints [14]. In Teleoperation mode the same position controller is used but the desired joint positions $\mathbf{q}_{1-7,d}$ are received from the inverse kinematics. Joints 8 and 9 are sent to the instrument.

The master is running force controlled in all modes. If the surgeon is not working with teleoperation, the master devices are gravity compensated. When coupling master and slave in teleoperation, forces feed back from the sensor in the instrument is enabled. For moving the endoscope the surgeon can connect one of the masters to the endoscope robot through layer 4. This is consistant with the Four Layer Architecture because there is no force feedback. Therefore, the task is not time critical.

The alteration of operating modes is modeled with two switches. Manual Motion mode for example, the path of the components Configure Move hands on, Impedance Control, Torque Control is active, i.e. its out port is connected to the robot. 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 hold in a proper initial state. Unsteady behaviour that could lead to stability problems is excluded. The frames and vectors in Fig. 8 can all be interpreted as desired values of one Master-Slave arm that are sent to the mechatronic hardware.

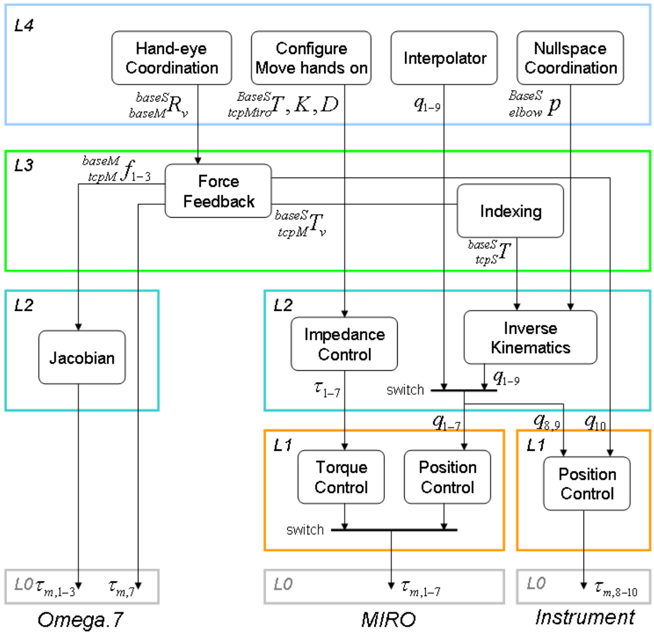

Fig. 8. Frontview of the four Layer Architecture with the master and the slave system consisting of the MIRO and an instrument.

In the next section components of layer 4 and layer 3 for teleoperation are described.

C. Teleoperation

A prerequisite for a surgical teleoperation system is an intuitive hand-eye-coordination. 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 surgeon'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

$$ {}^{\text{baseS}}_{\text{tcpE}}\mathbf{R} = {}^{\text{baseS}}_{\text{world}}\mathbf{R} \cdot {}^{\text{world}}_{\text{baseE}}\mathbf{R} \cdot {}^{\text{baseE}}_{\text{tcpE}}\mathbf{R} \tag{1} $$

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 seperately. The hand-eye-coordination matrix

$$ {}^{\text{baseS}}_{\text{baseM}}\mathbf{R}_v = {}^{\text{baseS}}_{\text{tcpE}}\mathbf{R} \cdot {}^{\text{tcpE}}_{\text{display}}\mathbf{R}_v \cdot {}^{\text{display}}_{\text{baseM}}\mathbf{R} \tag{2} $$

is given with the orientation of the master device base frame relative to the display. 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. It only changes when the endoscope is moving, that means the manipulator arm stands still. 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. 9.

In bilateral teleoperation a master and a slave robot are connected. Positions, velocities, and forces have to be

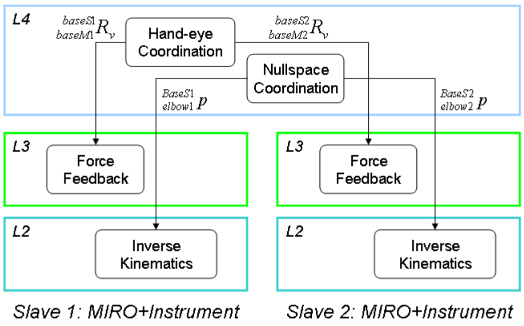

Fig. 9. Sideview of the four Layer Architecture with one slave system on the left and one on the right side.

transformed from master to slave and vice versa. The current version of force feedback is a position-force implementation which can be seen as a subset of the general Lawrance four channel architecture [12]. The measured positions from the master are sent to the slave and measured forces are sent back. The master pose is first transformed into the slave base frame

$$ {}^{\text{baseS}}_{\text{tcpM}}\mathbf{T}_v = \begin{pmatrix} {}^{\text{baseS}}_{\text{baseM}}\mathbf{R}_v & \mathbf{0} \\ \mathbf{0} & 1 \end{pmatrix} \cdot {}^{\text{baseM}}_{\text{tcpM}}\mathbf{T} \tag{3} $$

with the hand-eye-coordination matrix. The result is the virtual master frame in the slave base frame. The desired tcp of the slave in its base frame

$$ {}^{\text{baseS}}_{\text{tcpS}}\mathbf{T}_d(n) = g({}^{\text{baseS}}_{\text{tcpS}}\mathbf{T}(0), {}^{\text{baseS}}_{\text{tcpM}}\mathbf{T}_v(i), couple(i)) \tag{4} $$

is a function of the pose of the master at a timestep ${}^{\text{baseS}}_{\text{tcpM}}\mathbf{T}_v(i)$ with $i = 0..n$ and the corresponding signal $couple(i)$ and the initial slave pose ${}^{\text{baseS}}_{\text{tcpS}}\mathbf{T}(0)$. 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 can be 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. 8) can therefore be interpreted as a relative Cartesian slave that allows motions from any initial master pose. The desired tcp ${}^{\text{baseS}}_{\text{tcpS}}\mathbf{T}_d$ is sent to the inverse kinematics and transformed into joint space $\mathbf{q}_d$. Joints 1-7 are sent to the MIRO position controller whereas joints 8 and 9 are sent to the instrument controller respectively. The 10th DoF which connects the gripper of the master with the forceps of the slave is treated separately.

The transformation of wrenches measured with the sensor of the instrument

$$ {}^{\text{baseM}}_{\text{tcpS}}\mathbf{w}_v = \begin{pmatrix} {}^{\text{baseS}}_{\text{baseM}}\mathbf{R}_v^{-1} & \mathbf{0} \\ \mathbf{0} & {}^{\text{baseS}}_{\text{baseM}}\mathbf{R}_v^{-1} \end{pmatrix} \cdot {}^{\text{baseS}}_{\text{tcpS}}\mathbf{w} \tag{5} $$

is done with the inverse hand-eye-coordination matrix. The desired forces for the master are calculated according to

$$ {}^{\text{baseM}}_{\text{tcpM}}\mathbf{f}_d = \mathbf{h}\left({}^{\text{baseM}}_{\text{tcpS}}\mathbf{f}_v\right) \tag{6} $$

where $\mathbf{h}$ is a linear controller. The forces in joint space are commanded to the master hardware directly without a joint controller in between. The haptic master is assumed to have rigid dynamics and does not distinguish between motor and joint torque. Unlike the MIRO, where measured joint torques are fed back to control a flexible joint model.

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
- 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.

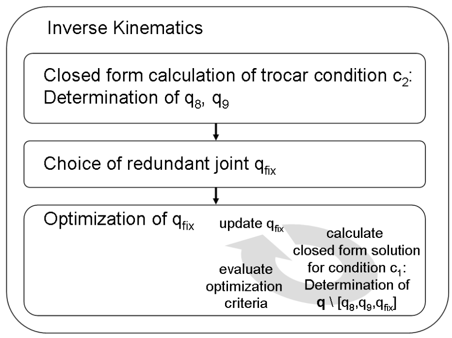

Fig. 10. Inverse kinematics algorithm with closed form solutions and nonlinear nullspace optimization.

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 [9] 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. 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 and Results

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 in 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. With the AutoPointer, the results of the planning procedure are projected directly onto the patient, and 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.

B. Control

The control software was developed with Matlab/Simulink and Real Time Workshop for automatic code generation. 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 [4]). 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. 11. 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 sychronized 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 [19]. 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

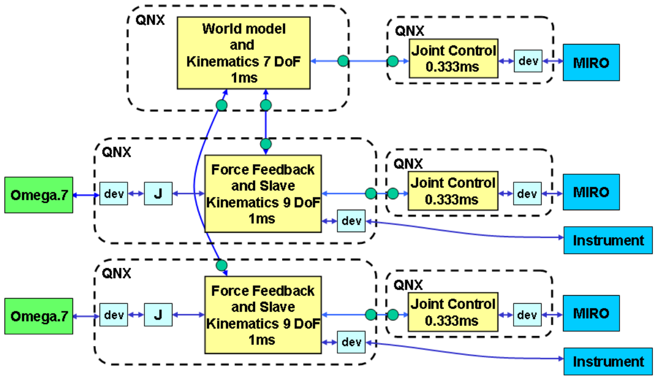

Fig. 11. Distributed Control Software for MIRS.

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.

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 succesfully 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.

VI. Conclusions and Future Works

The paper presents the planning tool and the control software of the DLR robotic system for minimally invasive surgery. The planning optimizes the setup in the operating room preoperatively and adapts it to variations intraoperatively. The control system is structured according to real time requirements and abstraction level of the components. The conceptual result is the Four Layer Architecture that gives a functional view of the system. According to this architecture a control system was modeled and implemented. The system is flexible and adaptable to future innovations in control or hardware design.

Future works will include advanced collision avoidance strategies in the case that the specified operating area leads to disadvantageous robot configurations. Bilateral teleoperation can be extented to more channels (positions, forces) or based on impedance control. There will also be additional kinds of instruments integrated into the system. A challenging and exciting feature will be motion compensation in beating heart surgery.

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] 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.

[5] 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.

[6] 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.

[7] 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.

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

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

[10] 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.

[11] 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.

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

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] S. Galvan, A. Castellani, D. Botturi, P. Fiorini. Advanced Teleoperation Architecture. In *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006.

[19] 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.

[20] 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.