Journal of Medical Robotics Research

http://www.worldscientific.com/worldscinet/jmrr

Journal of Medical Robotics Research

A Cognitive Robot Control Architecture for Autonomous Execution of Surgical Tasks

Nicola Preda[*,§], Federica Ferraguti[†,¶], Giacomo De Rossi[‡,‖], Cristian Secchi[†,**],
Riccardo Muradore[‡,††], Paolo Fiorini[‡,‡‡], Marcello Bonfè[*,§§]

Engineering Department, University of Ferrara, Italy

†*Department of Science and Methods for Engineering*
University of Modena and Reggio Emilia, Italy

‡*Department of Computer Science, University of Verona, Italy*

The research on medical robotics is starting to address the autonomous execution of surgical tasks, without effective intervention of humans apart from supervision and task configuration. This paper addresses the complete automation of a surgical robot by combining advanced sensing, cognition and control capabilities, developed according to rigorous assessment of surgical requirements, formal specification of robotic system behavior and software design and implementation based on solid tools and frameworks. In particular, the paper focuses on the cognitive control architecture and its development process, based on formal modeling and verification methods as best practices to ensure safe and reliable behavior. Full implementation of the proposed architecture has been tested on an experimental setup including a novel robot specifically designed for surgical applications, but adaptable to different selected tasks (i.e. needle insertion, wound suturing).

Keywords: Surgical robotics; task modeling; cognitive systems.

JMRR

1. Introduction

Surgical robots provide more and more research and application perspectives to both medical and engineering domains. Robotics allows surgeons to improve the quality of many critical surgical tasks or makes possible interventions that otherwise would not be possible [1–3].

Most surgical robots, either commercially available such as the Da Vinci (Intuitive Surgical, Inc.) or developed by research entities like the DLR MIRO [4] and the RAVEN II platform [5], are teleoperated systems. This

means that, despite the fact that mechanical design and control hardware/software of such systems are highly sophisticated, they act as mere extensions of human surgeons, with limited (if any) autonomous capabilities provided by assistive forces or virtual fixtures [6] on the teleoperation master device. Embedding increasing levels of autonomy into surgical robots and giving them the possibility to carry out simple surgical actions automatically have been the subjects of recent academic research [7].

Needle insertion and suturing are among the most studied surgical tasks in the last years. The use of Magnetic Resonance Imaging (MRI) or Computed Tomography (CT) to guide a robot during the insertion of needles (e.g. for biopsies or other purposes) has been validated in laboratory setups or animals [8–10]. The execution of the suturing task with the automation of knot tying in laparoscopic or open surgery is described in many papers [11–14]. Since mimicking the human gesture involved by this operation is challenging, some works described the use of specifically designed mechanical adapters [15].

There are two critical aspects in any automatic or semiautomatic surgical procedure: the safety issue and the registration of the robotic system. The analysis of the safety goes back to the early stage of robotic surgery and medical robotics and it is still one of the possible showstopper for these technologies [16, 17]. Novel design approaches are needed to integrate safety and security in the early stage of the development phase, both at the hardware and software levels [18]. In this paper, the use of formal models and verification tools is proposed as a viable approach to address such design issues.

The registration is the other big challenge when more than one robot or instrument have to work together and to exchange information (e.g. multi-robot surgical platform). What makes the registration such a difficult problem is that in medical robotics the operating environment (i.e. human bodies) deforms during the intervention [19, 20]. This effect is even more critical in automatic procedures, when surgeons cannot manually compensate for mismatches. Almost all registration algorithms are based on optical tracking systems [21] for percutaneous interventions, and endoscopic images and/or ultrasound (US) images for laparoscopic interventions. A precise registration can be also used to compensate for respiratory and cardiac motions [22].

Other basic surgical tasks have received some attention in the research community. A remotely controlled catheter guiding robot was used to automatically perform cardiac ablation [23]. However, an experienced operator is required to perform all the procedures. Automatic scissors were proposed, so that surgeons can command an assisting robotic arm to cut the thread that he/she is holding [24]. Though all these works propose successful automation of simple surgical actions, validated and commercially distributed autonomous surgical robots are still hard to find. An example is ROBODOC [25], which is a system capable of interventions on rigid tissues (i.e. bones drilling or cutting). On the other hand, the properties of such tissues, rather than soft ones, greatly simplify the robotic task and allow the use of robots with stiff mechanical structure and predefined motion paths, which are standard features in industrial automation.

This paper describes results obtained during a research project, called *I-SUR* (Intelligent SUrgical Robotics, funded by the European Union), whose goal is to develop a robotic system that can autonomously execute selected surgical tasks on soft tissues, by combining sensing, cognitive capabilities and advanced control algorithms. The tasks addressed so far during the development of the proposed intelligent surgical robot are: the insertion of needles into soft bodies, guided by US imaging and emulating the surgical procedure for percutaneous cryoablation of small tumoral masses (this task will also be called simply *puncturing*); 3D vision-guided *suturing* of

planar wounds. The goal of this paper is to improve the full automation of such tasks in the following aspects:

(1) *puncturing*: from the CT acquisition to the planning and execution of the needle insertion, every phase has to be done automatically by the system, validated by the surgeon supervising the procedure, and executed by the cognitive robotic system;

(2) *suturing*: from the rough identification of the wound on an image by the surgeon, the system has to accurately detect the edges of the wound, to choose the number and location of the stitches, to plan the motion of the robot arms to perform the suture, and to validate at run time each stitch according to the specifications provided by surgeons (pre-operative knowledge).

This paper extends the analysis and implementation of these surgical tasks described in preliminary works [26, 27]. The proposed control system is able to adapt online the interaction between the robot and the environment and to switch from autonomous to teleoperated mode preserving stability.

A formal design framework is exploited to precisely specify the US-supervised puncturing and the vision-based suturing procedures and translate them into an engineering design, as proposed by Bonfe *et al.* [28]. The formal description enables automatic software design of the robotic control system and provides validation-oriented requirements that must be addressed during functional tests. Thanks to the modular and component-based architecture of the system, the same methodology and design approach can be applied to automate both the *puncturing* and *suturing* tasks.

The main contributions of the paper are:

- The definition of a requirements engineering approach to software design for complex cognitive robotic systems, capable of autonomous execution of surgical tasks. The proposed approach integrates formal modeling and verification methods to address safety issues from the very beginning of the development process.
- The integration of sensing, cognition and control capabilities into a modular software system, whose architectural properties allow to enhance reconfigurability and re-usability of its main components. Particular care is given to the implementation of robot motion planning, control and supervision with safety-related features, which is the most critical part of the system;
- The experimental validation of the proposed approach on a novel surgical robot developed within the I-SUR project.

The paper is organized as follows: Sec. 2 introduces the surgical tasks selected as case studies and the robotic setup prepared for experiments. Section 3 describes the

proposed methodology to collect the requirements and translate them into control-oriented specifications. Section 4 describes the proposed system architecture and its component-based software implementation. Finally, use case scenarios and results collected during the execution of the addressed surgical tasks are shown in Sec. 5, which is followed by conclusions in Sec. 6.

2. Surgical Tasks and Robotic Setup

This paper focuses on the development of a control architecture for a novel surgical robot prototype. The robot is designed to autonomously perform different surgical tasks with a minimal mechanical reconfiguration. Consequently, the control architecture is developed with the same focus on flexibility and reconfigurability. In the following, we briefly introduce surgical requirements of the considered tasks and the experimental robotic setup.

2.1. *Needle insertion*

Among possible applications of needle insertion, the particular case of percutaneous cryoablation of small renal tumors has been addressed more intensively, in order to emphasize the potential benefits of the proposed technology and the pre- and intraoperative analysis that it allows.

Percutaneous cryoablation is the act of killing tumoral cells by means of cycles of freezing and thaw [29]. Freezing is applied by hollow needles in which liquid nitrogen or argon gas is circulated, so that an *iceball* is grown surrounding the needle tip. More needles may be required to create a sufficiently large iceball covering the whole tumor. Since the clinical objective is to destroy the tumor and limit damages to healthy tissues, accurate planning and execution of needle positioning is crucial. The surgical workflow can be resumed as follows:

- Preoperative CT/MRI images are analyzed by the surgeon to plan the required number of needles and where to place their tip. The expected size of the iceball is evaluated from isotherm maps provided by the cryoablation needle manufacturer and from surgeon experience.
- Needles are inserted through the skin into the tumoral mass, avoiding as much as possibile bones, nerves and other organs. Intraoperative US imaging is used for needle insertion guidance and iceball formation monitoring. During this phase, it is important to align the US probe so that both the tumor and the needle appear on the US image. In case of needle trajectory misalignments, due to the deformation of soft tissues, its orientation can be corrected if the needle is up to 2 cm within the body, otherwise it must be extracted and inserted again.

- After the cryoablation freezing/thaw cycle, needles must be extracted giving particular care to the force required for removal. In fact, incomplete iceball melting can hardly be detected from US monitoring and if the needle is still *trapped* by residual ice its extraction would cause bleeding and organ damages.

It is commonly acknowledged by both surgeons and robotics researchers that using accurately calibrated mechanical arms, guided by properly registered US image processing, needle insertion could be executed more precisely. Moreover, robotic end-effectors equipped with force/torque sensors would promptly detect needle trapping conditions and react accordingly. Within the I-SUR project, the automation of this surgical task has been addressed following an increasing-complexity approach, which means that three cryoablation scenarios were considered: from the simplest case of a small tumor that can be treated with a single needle, whose insertion trajectory is specified manually by an expert surgeon, up to the most complex case of a tumor requiring up to five needles for treatment, whose positioning is fully autonomously planned from CT/MRI image analysis for optimal tumor coverage to robot motion generation. With this approach, the requirements for each case builds on those of previous ones by adding issues and desired features, but also technical solutions developed and validated for one case can be reused to address the next level.

2.2. *Suturing*

Since the aim of the I-SUR project is to develop modular and reconfigurable cognitive control architectures for autonomous surgical robots, a different surgical task has also been addressed, namely the act of suturing (i.e. closing a wound in a biological tissue by means of a thread). Even for the automation of this task, an increasing-complexity approach can be applied to define the following case studies:

(1) **Simple planar suture** of a linear incision on a flat surface.
(2) **Complex planar suture** of an irregular incision on a flat surface.
(3) **Complex suture on nonplanar surface**
(4) **Tubular suture**, a challenging task even for expert surgeons, required for example to repair blood vessels (i.e. aortic anastomosis).

In any of the previous cases, the suturing action requires the use of two tools, generaly a needle holder and a forceps. Semi-automatic suturing instruments also exist, especially for laparoscopic operations (e.g. Covidien Endo Stitch™, as will be described later). Even if different suturing techniques exist, the surgical workflow

can be generalized as an iterative process, whose iterations require the following steps:

(1) Plan the stitching point on the tissue, according to length and depth of the wound.
(2) Insert the needle on one edge of the wound using the needle holder.
(3) Grab the needle with the forceps and switch it back to holder.
(4) Insert the needle on the other edge and push it further.
(5) Grab the needle with the forceps and pull, return to Step 1.

Within the I-SUR project, the automation of the suturing action has been implemented referring to the simplest case (i.e. linear wound on a planar surface). However, the solutions developed for this case are expected to be applicable, with proper refinements, also to cases 2 and 3, while tubular sutures seems too much challenging for the current state-of-the-art robotic technologies.

2.3. *Robotic setup*

The surgical robotic setup developed to execute both emulated cryoablation and wound suturing has been deployed in two slightly different configurations. For the *puncturing* case study, the system is prepared as shown in Fig. 1, assembling the following parts:

– A UR5 industrial robot (Universal Robots A/S), a 6 Degrees-Of-Freedom (DOF) manipulator with a 5 kg payload and a reach radius of up to 850 mm, holding a US probe thanks to an *ad hoc* adapter.
– A robot specifically developed for surgical applications and based on a macro-micro mechanical design approach [30], that will be called *ISUR (Intelligent SURgical) robot*. The macro unit is a parallel robot, whereas the micro unit is composed of up to two serial arms, holding a cryoablation needle or a suturing device according to the desired task. The surgical tool is mounted on a 6-DOF force/torque sensor (ATI Mini40, ATI Industrial Automation) with a resolution of 0.01 N/0.00025 Nm for control and monitoring purposes. Further details about the mechanical system, designed by the RELab of ETH Zürich (partner of the I-SUR project), are described by Muradore *et al.* [27].
– A phantom accurately reproducing a human abdomen, manufactured at the Centre for Biorobotics of Tallinn University of Technology (another partner of the I-SUR project), that is shown as a red box in Fig. 1. More details about the anatomical characteristics of the phantom and its CT and US compatibility are described in a previous work [31].
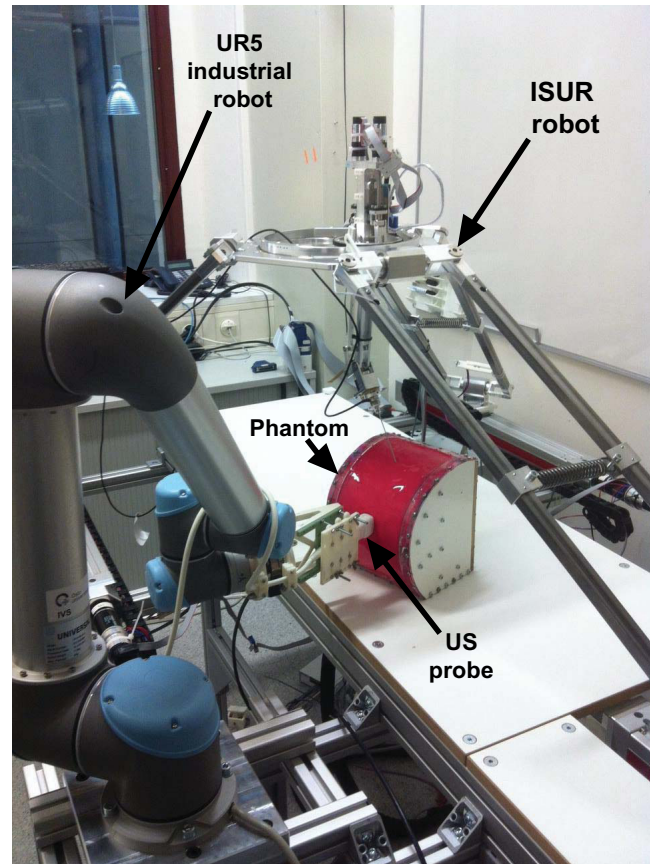– A couple of PHANTOM Omni® (Sensable) haptic devices, with 6 sensed DOF and force feedback on the

Fig. 1. Experimental setup for the needle insertion task.

translational DOF, allowing bilateral teleoperation of the robots.

The setup allows to emulate a cryoablation operation using real cryoablation probes (IceRod™ from Galil Medical, Inc.), apart from the actual freezing/defreezing cycles, since refrigerating gas circulation machines could not be installed in the academic laboratory hosting the experimental setup because of obvious safety issues. A 3D optical tracking system (Accutrack 500, Atracsys LLC, a system with active markers and a mean position error of 0.19 mm) is used to estimate relative coordinate transformations among the robots and the phantom. Finally, the setup includes an ultrasound imaging device whose images are visualized on a dedicated graphical interface for the surgeons and processed in real time to detect the position of needle tip, using the algorithm developed by Mathiassen *et al.* [32], and provide intraoperative adaptation of robot motion trajectories, as required by the surgical workflow previously described.

For the suturing task, the ISUR robot is equipped with two arms (i.e. the micro unit, mounted on the moving platform of the macro unit parallel robot). Instead of installing needle holder and forceps on such arms, whose use would require a larger workspace and more complex maneuvers, a semi-automatic instrument Endo Stitch™

(by Covidien) has been mounted on the right arm. The Endo Stitch is a specific tool for internal suturing during laparoscopic surgery, with two jaws: a tiny needle can be held in one jaw and passed to the other jaw by closing the handles and flipping a toggle levers. These operations have been automated by reverse engineering. The left arm of the ISUR robot micro unit, instead, has a gripper at its end effector, which is needed to grasp and move away the thread to avoid knots during the procedure. Even though the Endo Stitch is designed for laparoscopic interventions, the current setup of the ISUR robot is only addressing uses in open surgery suture, to simplify motion planning and execution issues.

The mechanical setup is shown in Fig. 2. The adapter holding the Endo Stitch is endowed with two motors: the first one for closing the jaws (and so executing the stitch) and the second one for switching the needle from one jaw to the other when the stitch is done.

The planning and monitoring of the task is vision based. A 3D camera system is used for the registration of the phantom with the micro-macro ISUR robot, for the detection of the wound, for the planning of the stitches and for compensating misalignment and mis-registration during the execution.

A phantom is used to reproduce the human skin. The phantom is a silicon-made pad with two layers having different stiffness and color. The difference in color is exploited by the camera system to detect the wound and to track the Endo Stitch tool and detect its correct insertion within the wound, whereas an admittance controller takes care of the contact of the tool with the soft tissue.
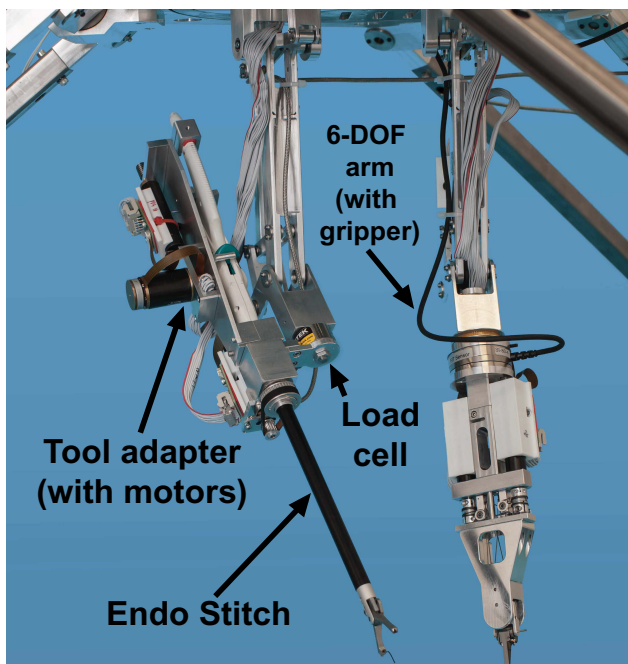

Fig. 2. Experimental setup for the suturing task.

3. Requirements Engineering and Design Specification

3.1. *Development process*

The design of cognitive autonomous robots for surgical applications must ensure a safe and reliable behavior of the final system. A careful use of formal modeling and verification tools is commonly acknowledged as a viable approach to address safety issues, especially within complex and software intensive design projects [33–35]. For this reason, design specifications for supervision, reasoning and control logic have been formalized using a requirements engineering approach, following the validation-oriented methodology described in a preliminar form in a previous work [28]. In particular, the proposed methodology aims to collect human knowledge about the desired surgical procedures and related safety issues, translate it into a formal model and automatically generate control-oriented specifications of the prescribed system behavior. The latter is then mapped into the supervision logic of the final software architecture, whose correctness properties can be further verified using formal tools.

In the initial phase of the development process, the knowledge of expert and specialized surgeons (e.g. urologists practicing cryoablation tasks) is captured to define, for each surgical task, a detailed definition of the main procedures ("best practice") to be performed, the elements of the domain (i.e. tools, gestures, preoperative and intraoperative data), the critical events related to the surgical actions and how they could be addressed to preserve safety. In the I-SUR case, this phase required surgeons interview, participation of developers and engineers to real surgical operations and execution of such operations on artificial phantoms and augmented reality simulators developed during the project, as described by Muradore *et al.* [36]. Then, surgical requirements are expressed using a goal-oriented methodology called FLAGS (Fuzzy Live Adaptive Goals for Self-adaptive systems [37]), which is focused on the essential objectives of an operation and on complications that may arise during its execution. The result of the knowledge formalization is a *goal model*, technically defined as a set of formal properties expressed in the *Alloy language* [38], which is a specification language expressing structural and behavioral constraints for complex software systems, based on First-Order Logic (FOL) and Linear Temporal Logic (LTL [39]). For example, a leaf goal of the cryoablation procedure, related to its safe completion, requires to avoid *forbidden regions* (i.e. bones, nerves, other organs) during needle insertion. The goal is specified by the following LTL formula:

$$G[\text{MP} => !(\text{FR} \wedge (\text{FR.needle} = \text{MP.needle}))] \qquad (1)$$

asserting that whenever (i.e. *G*lobally) a movement is performed (event MP), the *needle* entity associated to the movement must not touch a forbidden region (event FR).

The advantage of using a formal language for requirements specification is that the goal model can be automatically transformed into a sequence of operations and adaptations, satisfying the goals of the surgical procedure, thanks to the features of the Alloy Analyzer tool [40]. As a result, the process provides a sequential model equivalent to a state machine, representing the whole system behavior that guarantees the achievement of the root goal and does not violate safety constraints. Using this approach the requirements analysis is focused on the objectives of a surgical task, rather than the way (i.e. the operational sequence) in which they are obtained, since the latter is generated automatically by formal reasoning.

To enforce reconfigurability and reusability of the control software architecture, modular design is also commonly recommended. The state model obtained after goal-oriented analysis can be used for modular software design provided that its overall logic is refined and partitioned into the structural units of the system, as will be further described in next subsection, implementing a collaborative and coordinated behavior compatible with the requirements. This task is performed applying decomposition methods from classical discrete systems theory and using UML (Unified Modeling Language [41]) as a modeling tool, being the latter the current *gold standard* in object-oriented and component-based software design.

Finally, the UML model of the modular system is verified using formal tools for Model Checking [42] (namely, the tool *Symbolic Model Verifier*, SMV [43]), to prove that the design model preserves the properties expressed by the goal model. This task requires the formalization of an appropriate semantics of the UML behavioral specification (i.e. State Diagrams of system components), compatibly with the operational features of its run-time implementation. This step will be further analyzed in Sec. 3.3.

3.2. *System design*

The autonomous robotic system designed in this project is supervised and controlled by the following modules: a *Surgical Interface*, the *Robot Controllers* and the *Sensing* system with *Reasoning and Situation Awareness* capabilities. The Surgical Interface is a software-intensive system allowing surgeons and technicians to drive the system during both the preoperative and the intraoperative phase. In the first one, the focus is on detailed planning of the surgical intervention, while during the execution of operations the interface provides real-time visual navigation of the surgical scenario and, if necessary, allows the surgeons to take control of the system by switching into a teleoperated mode.

The Robot Controllers are the units implementing control of surgical actions and tasks sequencing during the intraoperative phase. The event-driven behavior extracted from the goal model is mapped into the robot control logic, which is specified by a UML State Diagram. Needs for a safe behavior require a strict coordination of these components with both the Surgical Interface and the Sensing/Reasoning module. The latter is a composite sub-system implementing advanced Sensing algorithms and Reasoning for Situation Awareness, whose role is to provide support to the planning task, during the preoperative phase, and prompt identification of anatomical changes or discrepancy between the tasks being executed and the *nominal* surgical plan, during the intraoperative phase. Bayesian Networks and Particle Filters [44] are used to detect the occurrence of undesired events and critical situations, so that appropriate corrective actions can be triggered.

In the following, the logic behavior of the Robot Controller, with integrated safety mechanisms, and its interactions with other modules is described for each surgical case study. For the puncturing case, the starting point of the procedure is the automated planning of cryoablation needles placement, a feature embedded in the Surgical Interface. A cryoablation planning algorithm (also called *cryo-planner* in the following), a novel contribution in itself described more precisely by Torricelli *et al.* [45], elaborates preoperative medical images to calculate the optimal placement of cryoprobe needles, providing full tumor coverage with the expected iceball and not interfering with other organs (i.e. *forbidden regions*). The plan generated by the cryo-planner specifies for each needle the skin entry point and the target point on the tumor. The needles placement is referred to the center of the tumor, therefore the task plan, once validated by the surgeon, must be mapped into the operational space of the robot by means of the registered coordinate transformations calculated by the Sensing/Reasoning module. During the actual needle insertion task, the US probe is first placed on the surface of the body, aligned with the expected needle tip trajectory, and then the planned needles are one by one mounted on the robot end-effector, when the latter is placed into a given needle changing pose, and then automatically inserted. Similarly, once that the cryoablation freezing cycle is completed, the needles are expected to be extracted one at a time, monitoring the applied force to detect if the iceball is not completely melt and consequently a needle is trapped. The complete behavioral specification of the robot control logic for needles insertion, compatible with the previously described workflow, is given by the UML State Diagram shown in Fig. 3.

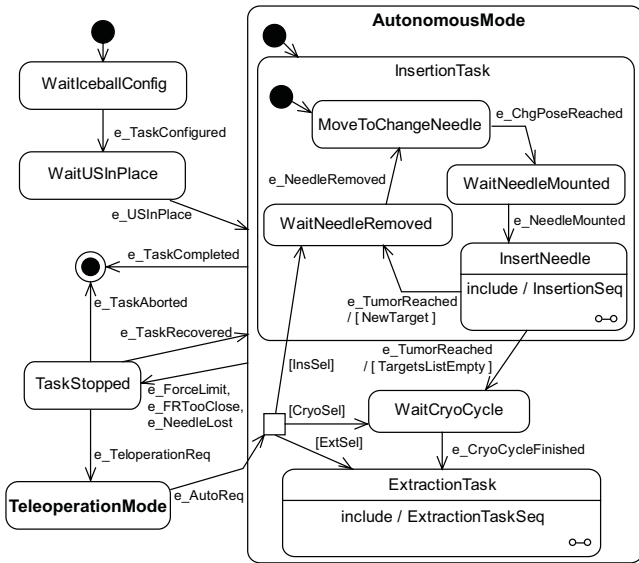The suturing surgical task instead, as addressed within the ISUR project, requires a more complex

Fig. 3. UML State Diagram of the behavioral specification for cryoablation needles insertion.

coordinated motion of the two arms installed on the moving platform of the parallel macro-unit, but on the other hand its control logic is simpler. In fact, the adoption of a semi-autonomous suturing device like the Endo Stitch prescribes a well-defined sequential behavior. The sequence of required robot motions is schematized in Fig. 4.

Assuming that a *Right Arm* mounts the Endo Stitch and an assisting *Left Arm* has a gripper to grasp the thread, the procedure starts with the right end-effector (EE_R) at P_0 (see Fig. 4) and the left end-effector (EE_L) at a relative distance from EE_R. Then, both end-effectors move to target poses such that EE_R reaches P_1 and EE_L keeps the initial distance. The application of a stitch requires to insert the EE_R inside the wound with the clamp oriented along the cut (EE_L holds in place) and then rotate it to clamp the left border of the wound in P_3.
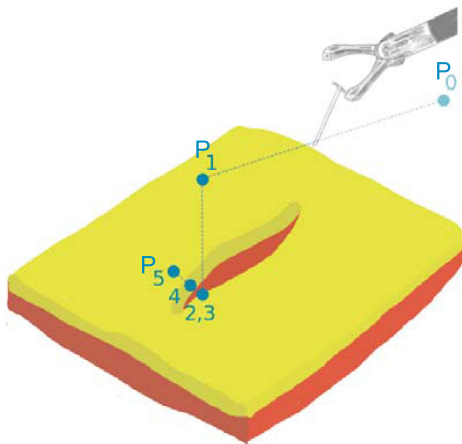

Fig. 4. Sequence of motions for a suturing task executed with the Endo Stitch^{TM} tool.

After that the Endo Stitch executes the *bite and switch* action, the EE_R goes to a target above P_1, stopping when a proper thread tension is detected. Finally, the EE_L grasps the thread and pulls it away, while EE_R moves to keep a proper thread tension and the sequence is repeated on the right border of the wound and again from the beginning for the next stitch. The suture planning is also done automatically and it consists of a pre-operative and an intra-operative phase. The first analyzes 3D images of the wound and calculates the required number of stitches, according to the length of the incision. The second one updates the wound detection after the execution of each stitch, since this action inevitably modifies the wound shape, and refines accordingly the proper placement of next stitch. The complete behavioral specification of the suturing task, that takes these requirements into account, is shown by the UML State Diagram of Fig. 5.

As can be seen, the hierarchical features of UML State Diagrams allow to embed exception handling mechanisms, by means of transitions exiting composite states. In both state machines, in fact, the robotic task can be stopped because of an exception event, that can be triggered either by the surgeons, through the Surgical Interface, or by the Sensing/Reasoning and Situation Awareness module. Events that are monitored and handled by the latter to preserve safety are: the needle is too
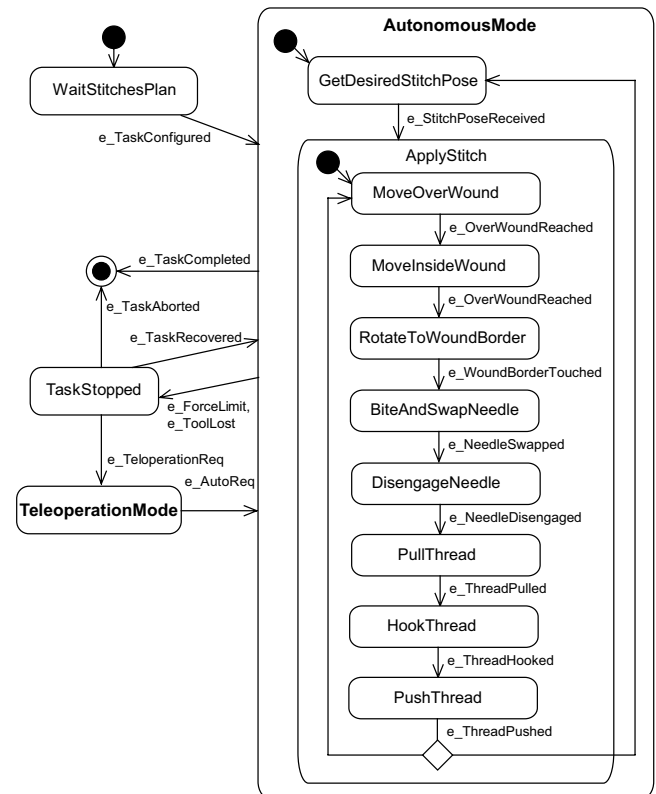

Fig. 5. UML State Diagram of the behavioral specification for the suturing task.

close to a forbidded region (e.g. a bone or another organ not involved in the cryoablation), namely `e_FRTooClose` in Fig. 3; the force applied by the robot exceeds a given limit, namely `e_ForceLimit` in both Figs. 3 and 5; the surgical tool is not properly tracked by US or stereo image processing algorithms, namely `e_NeedleLost` in Fig. 3 and `e_ToolLost` in Fig. 5. Whatever is the exception event, if the task execution can be restarted after appropriate validation of the surgeons, the transitions marked by the `e_taskRecovered` event are executed. Otherwise, the system allows the surgeon to teleoperate the robot, which is assumed to be the safest mode of operation.

3.3. *Model checking*

Formal verification of the UML design model requires the definition, first of all, of its operational semantics, according to the execution model of the target implementation framework. In particular, the proposed UML design has been implemented using the component-based Orocos framework [46] and rFSM (*r*educed *F*inite *S*tate *M*achines), an execution engine for hierarchical state machines written in Lua. The salient features of the operational semantics of an rFSM model [47] and its differences with the one defined by the UML standard can be summarized as follows:

- in UML events are supposed to be stored in a queue and processed one at a time to evaluate the enabled transition set of the state machine. Instead, rFSM collects all events occurred since the last executed machine *step* and uses them to evaluate enabled transitions and execute the next step, after which the whole set of occurred events is cleared;
- conflicting transitions are solved according to different structural priority schemes: UML gives higher priority to transitions whose source state is at lower hierarchical levels, while rFSM reverts the rule;
- rFSM does not support concurrency (i.e. so-called AND-states), since it assumed that this feature is provided by the higher level execution framework of the Orocos components deployer.

Assuming that an rFSM machine is embedded into a given Orocos component with input and output *event ports* to interact with rFSM machines in other components, it is possible to formalize the UML design model implemented in Orocos-rFSM as a modular transition system, following the approach described by Bonfè *et al.* [48]. In particular, the formal model of a component embedding an rFSM machine is a structure:

$$C = (S, T, P, r), \qquad (2)$$

where S is a (hierarchical) set of *states*, T is a set of *transitions*, $P = P^I \cup P^O$ is a set of "port" variables, each

one of a given data type (including *event*), and $r \in S$ is the root state. The full system is then defined as an ordered set of components and interconnections, together with a *scheduling function*. Such a formal model can be easily translated into the input language of the SMV tool [43], a model checker well-known for being able to efficiently handle the state-space explosion problem and allowing users to specify desired properties with either Computation Tree Logic (CTL) or Linear Temporal Logic (LTL).

The previously described rFSM events collection mechanism is different from the PLC-like execution model described by Bonfè *et al.* [48] and requires a specific adaptation. In particular, each event must be associated to an SMV module, whose internal boolean state is set true if the event has occurred and is reset when the event is cleared by the execution of the step of its "container" rFSM module. The module in SMV code is the following:

```
MODULE rFSM_EV(Event, Clear)
VAR
  Occurred : boolean;
ASSIGN
init(Occurred):=0;
next(Occurred):= case
             !Occurred & Event : 1;
             Occurred & Clear : 0;
             1 : Occurred;
             esac;
```

The SMV module related to an rFSM machine will include an `rFSM_EV` for each input and output event:

```
MODULE rFSM_Robot(ExecStep,e_ForceLimit,..)
VAR
e_ForceLimit_ev: rFSM_EV(e_ForceLimit,
                    (Exec = FINISHED));
Exec : {IDLE, STEP, FINISHED};
```

The module has a boolean input `ExecStep` that triggers the execution of its step, which is managed by the scheduling function mentioned before. When the step execution is completed, the enumerated variable `Exec` takes the value `FINISHED` and consequently input events are cleared. Finally, the hierarchical structure of the UML State Diagram specifying the behavior of an rFSM module is then encoded according to the same rules proposed by Bonfè *et al.* [48].

An SMV program is completed by the declaration of a `main` (i.e. container) module and by the specification of desired safety properties of the system. The SMV tool is then able to perform an exhaustive search of the state-space of the model, to confirm that such properties are never violated in any admissible execution path of the system. If instead a property is violated, SMV presents a counterexample (i.e. a path ending in a state not satisfying the property). As said before, the desired properties can be expressed using LTL, so that it is possible to verify

that the design model achieves the very same goals defined during the requirements specification, as described in Sec. 3.1 (for example, Eq. (1)).

The proposed verification procedure considers a discrete model of the surgical system in which robot motion is abstracted at an atomic level. Including a continuous model of motion into the verification process and then applying model checking algorithms for hybrid systems [49] is also possible, but at the price of an exponential growth of the computational complexity. On the other hand, formal verification of the hybrid model could consider a reduced version of the task state machine, modeling only the critical phases of the surgical procedure in which the robot is actually interacting with the patient. The puncturing task has been addressed with this approach, as described by Muradore *et al.* [50], and results obtained from different model checking tools for hybrid systems are presented by Bresolin *et al.* [51].

4. System Architecture

4.1. *Implementation and deployment*

Software development for the proposed cognitive surgical robotic system involves efforts from different research teams and, consequently, the integration of different software technologies. The core part of the robotic control system is implemented using Orocos components and runs on PCs with Ubuntu 12.04 Linux operating system and Xenomai hard-real time extension, while the Surgical Interface is developed using Microsoft .NET Framework and the low-level control (i.e. regulation of joints positions, velocities and currents) of the ISUR robot exploits National Instruments LabView 2013 and a CompactRIO control platform. Networked distribution of Orocos components is supported by CORBA [52] interoperability, while interconnection of Orocos-based software and other modules requires the development of socket-based exchange of TCP or UDP packets on Ethernet connection. A careful choice of network topology and the use of high-speed switches allowed to obtain a frequency of 1 kHz for the interaction between the Orocos-based high-level control system, that will be further described in Sec. 4.2, and the ISUR low-level control. Within the proposed architecture, the central role is played by the *Task Supervisor*, which contains the rFSM-based implementation of the UML State Diagrams described in Sec. 3.2, specifying the event-driven behavior required to coordinate robot actions and supervise the overall task execution. The cognitive part of the system is completed by *Sensing* software, which is in this case implementing real-time US image processing for needle tracking [32], and a *Situation Awareness* module, implementing Bayesian Networks [44] processing data received from Sensing and

robot control software to detect events and exceptions (e.g. forbidden regions touched, force limits exceeded, etc.).

The *Surgical Interface* manages the interaction with surgeons by showing intra-operative images, 3D rendering of the full robotics system in its current pose and accepts commands and inputs when required to progress with the task. Finally, a PHANTOM Omni haptic device, which is the master device when the robotic system is switched into the teleoperated mode, is installed on a dedicated PC together with a specifically designed interface software, a choice motivated only by issues related to device driver stability.

As described in Sec. 2, the experimental setup for the puncturing task includes two different robots: the ISUR robot, performing needle insertion, and a commercial UR5 robot, holding an ultrasound probe. Motion planning and control for the two robots are executed by duplicated instances of the same components, running on different platforms to simplify the interaction with low-level control hardware, as shown by the final deployment schematized in Fig. 6(a). The high-level control system for each robot is in charge of: searching *valid* Cartesian paths, satisfying task requirements and avoiding collisions among the two robots and other obstacles; generating timed trajectories satisfying dynamic constrains (i.e. maximum velocity and acceleration); generating low-level (i.e. joint position) commands for the control hardware. It is also useful to remark that the execution of the puncturing task does not require simultaneous motion of the two robots, so that their movements can be planned one at a time.

The suturing task, instead, requires a different setup and, consequently, a slightly different software deployment. In particular, the mechanical setup of the ISUR robot embeds two micro-units and no additional arms are required. Moreover, coordinated and simultaneous motion of the dual-arm robot requires a different configuration of the motion planning and control algorithms, as will be further analyzed in next subsection. Finally, the Sensing software is implemented within the ROS [53] environment, which can be straighforwardly connected to Orocos components, and performs the identification of wound borders by analyzing images from a Bumblebee 2 (Point Grey Research, Inc.) stereo camera with a 1024×768 resolution. The overall scheme of the deployed architecture is shown in Fig. 6(b).

4.2. *Task supervision and control*

The main objective of the proposed architecture is to embed autonomous behavior into a surgical robotics setup. Therefore, the *nominal* mode of operation of the control system corresponds to the automatic execution of supported surgical tasks. The components allowing the
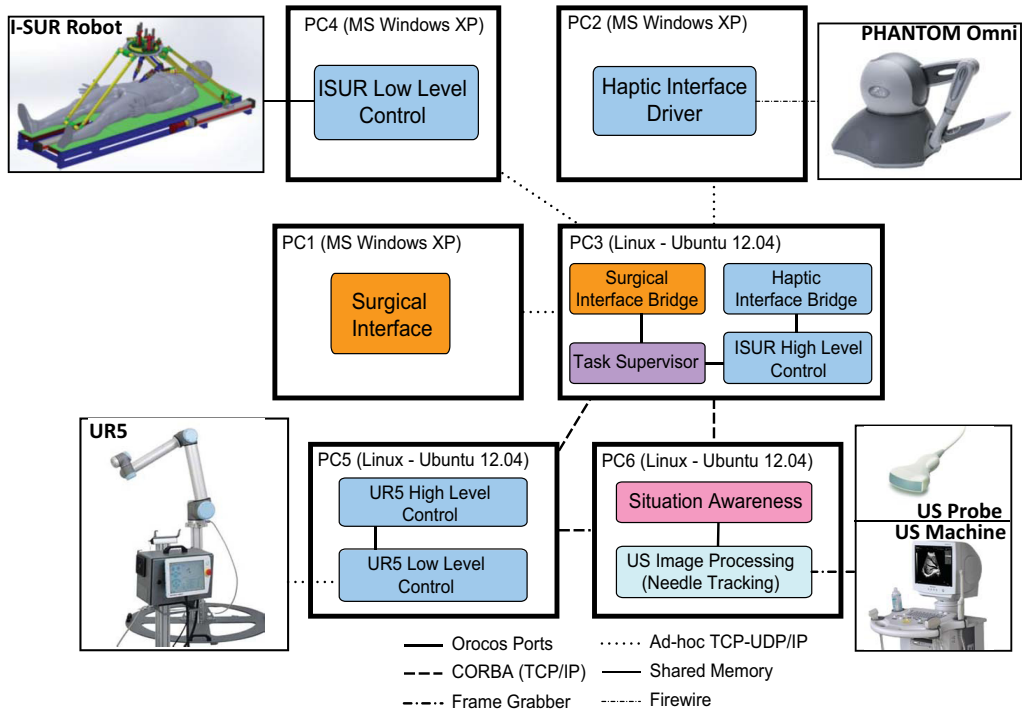
(a) *Puncturing* task
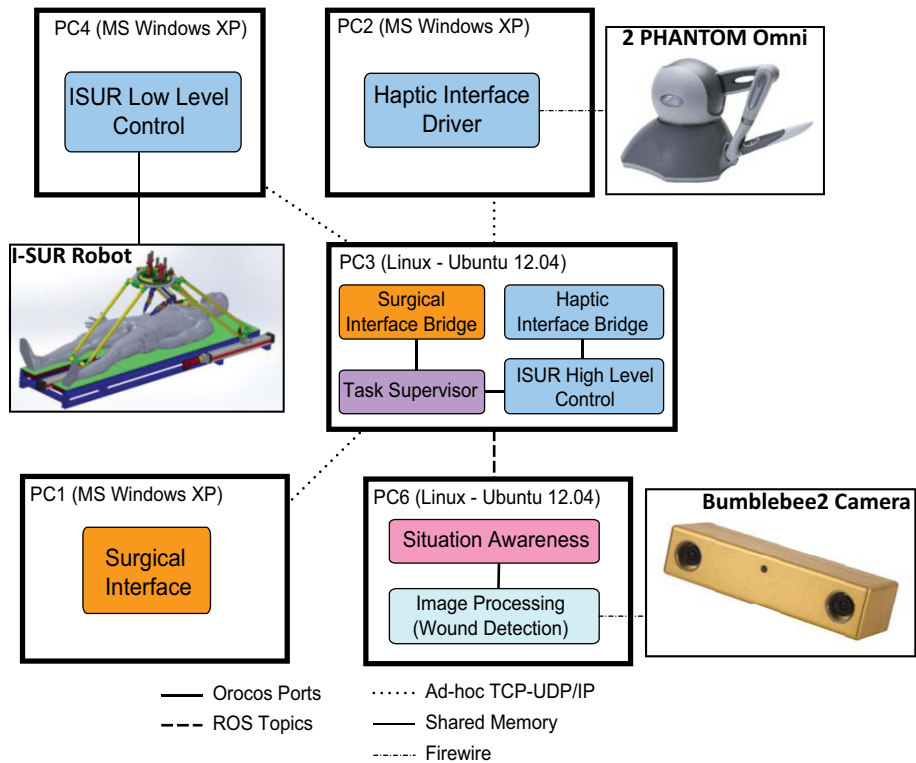
(b) *Suturing* task

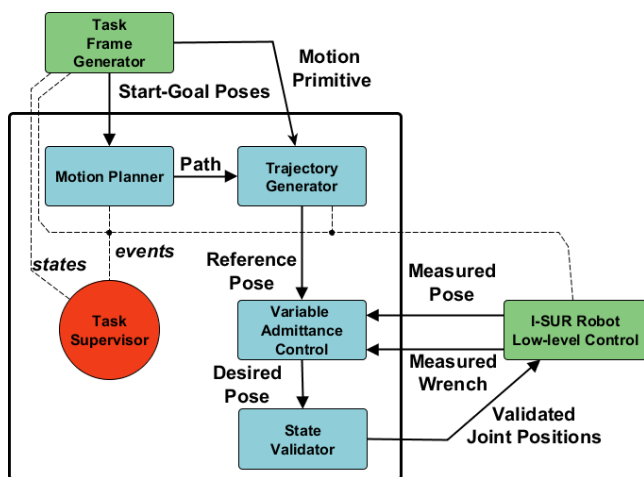**Fig. 6.** Cognitive control architecture for autonomous surgical robotics.

Fig. 7. Software components for ISUR robot motion planning and control (autonomous mode).

ISUR robot to implement the autonomous behavior are shown in Fig. 7.

The main director of the system is the *Task Supervisor* component, embedding the task state machine translated in Lua language using the rFSM framework. Any other component of the system interacts with the supervisor by means of events, such as the completion of a motion step or inputs from the Surgical Interface. In this way, each software component is developed to provide context-independent basic functionalities and its reusability is increased, since task-dependent coordination and configuration is demanded to the supervisor.

The only other component which is strictly task dependent is the *Task Frame Generator*: this component contains all the computations required to transform the data generated by preoperative planning tools, using transformation matrices calculated during both offline and online registration, so that the proper target points are assigned for motion planning. In particular, the required transformations are: between the reference frames of the two robots; from the end-effector of each robot to the specific tool tip; from US or stereo camera image coordinates to the robot frame; from the origin of patient/phantom 3D model to the robot frame. The latter is particulary important for the puncturing task, since the output of the *cryo-planner* software [45] (i.e. optimal needles placement) is referred to the center of the tumor. This output is post-processed by the Task Frame Generator also to find an adequate placement of the US probe that guarantees visibility of the needle during the insertion.

As shown in Fig. 7, the Task Frame Generator component can set the required motion command in two alternative ways, depending on the state of the task. In the first one, a *goal* pose is sent to a sampling-based *Motion Planner*. The Motion Planner generates online a collision-free path exploiting the *RRT-Connect* [54]

algorithm implemented by the the Open Motion Planning Library (OMPL [55]). For the puncturing case, the planner considers one robot at a time and the output is the path for the 6 DOF pose of the robot tip. For the suturing case, instead, simultaneous motion of two arms is required, so that the output is a composite path for the full 12 DOF dual-arm pose. In any case, the path is then transformed into a timed trajectory by the *Trajectory Generator*, applying multi-axis/multi-segment interpolation algorithms [56].

In the second way, the Task Frame Generator sends a motion primitive, specified by a fully predefined path, directly to the Trajectory Generator for interpolation. This alternative solution is necessary since sampling-based planning algorithms do not guarantee anything about the shape of the path, which is instead important for a correct execution of the surgical gestures. For the puncturing case, the only required motion primitive is a linear path connecting the skin entry point to the target on the tumor, while for the suturing task the motion primitives are those required to replicate the pattern schematized in Fig. 4 and described in Sec. 3.

The surgical robot must interact with the environment during the execution of the task. The regulation of the interaction behavior in the operational space is guaranteed by the *Variable Admittance Control* component. Admittance control and impedance control [57] are very effective schemes to achieve a desired robot/environment interaction, specified by a virtual multi-dimensional mass-spring-damper system. Loosely speaking, impedance control is more suitable for backdrivable robots while admittance control is more suitable for stiff robots. The robot specifically developed for this project, described in Sec. 2, has a stiff and not backdrivable structure. Therefore, a variable admittance controller has been implemented for the high-level control of the robot, introducing the possibility to modify online the stiffness of the interaction model without loosing passivity and, hence, stability of the closed-loop system [58]. Thanks to this dynamic behavior, the controller is adapted during the execution of the task, so that the robot is, for example, more compliant when approaching the surgical tool to the skin and stiffer when the tool (e.g. the needle) is pushed towards the final target.

A side effect of admittance control is to provide possible deviations from the reference trajectory, in case of environment interaction. Moreover, motion primitives provided by the Task Frame Generator are defined as incremental motions starting from current pose of the robot. As a result, the desired pose calculated by the Variable Admittance Control component is not always guaranteed to preserve a safe distance from collisions. To cope with this issue, the desired pose to be commanded to the low-level controller of the robot is validated by a specific *State Validator* component, that verifies that the pose is kinematically reachable and safely far from

collisions, otherwise the command is discarded and motion stops (the user may then decide to switch the system into teleoperated mode). More details about inverse kinematics of the ISUR robot and collision checking algorithms exploited for this purpose are described by Preda *et al.* [59].

Undesired events and direct surgeon requests may force the system into a *teleoperated* mode, in which the motion planning sub-system is not active and the surgeon takes control of the robots by means of dedicated haptic interfaces. Force feedback to such haptic devices is provided implementing a bilateral teleoperation control scheme [60]. Theoretical stability issues related to the autonomous/teleoperated mode switching have been addressed in the design of control algorithms [58]. Here, we can add that the software components Motion Planner and Trajectory Generator are replaced, in teleoperated mode, by components implementing the *Transparency Layer* and the *Passivity Layer* (according to the definitions of Ferraguti *et al.* [58]). Smooth mode switching is also supported by the fact the Variable Admittance Control component is designed to accept either pose commands, as required in autonomous mode, or wrench (i.e. 6 DOF force/torque vector) commands, as required by bilateral teleoperation.

5. Experiments

5.1. *Puncturing*

The automonous surgical robot and its cognitive control architecture have been tested during an experiment emulating a full cryoablation task, requiring the insertion of five needles, to validate the safety mechanisms embedded in the supervision system and the stability of control algorithms. The operations of the robotic system are described by the UML Sequence Diagram of Fig. 8.

The timeline of the Task Supervisor shows the states of the corresponding state machine along the *nominal* behavior of the robotic system. The system is actually able to manage deviations from the nominal case (i.e. undesired events) thanks to the design of the UML State Diagram of Fig. 3. A similar experiment, but ending with an undesired bending of the needle and the consequent request for a teleoperated mode switching, is reported by Ferraguti *et al.* [58], to demonstrate the stability of the control system during the commutation from autonomous mode to bilateral teleoperation.

In the nominal case, the surgical procedure is started when the ISUR robot and the UR5 robot are in a *home* position and the surgeon has confirmed the needle placement provided by the cryoablation planner. The
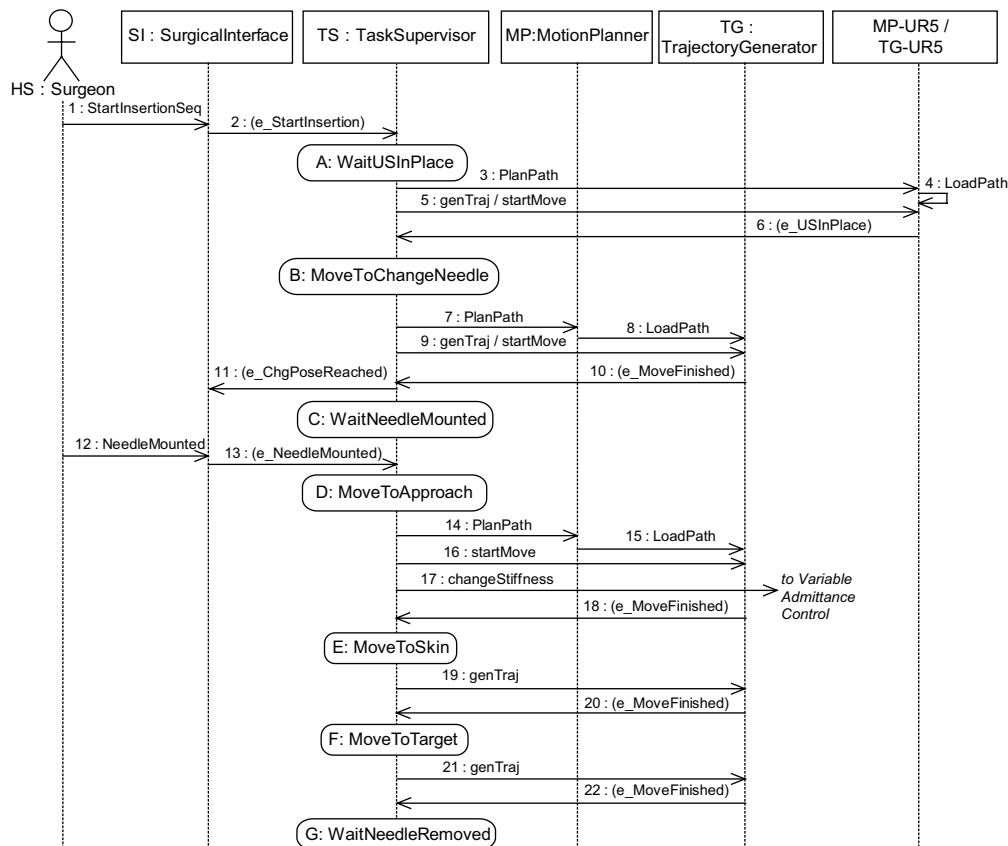

Fig. 8. UML Sequence Diagram describing the puncturing experiment.

UR5 robot is then moved along a collision-free trajectory, generated by the Trajectory Generator according to the path received by the Motion Planner, to a position allowing the needle tracking algorithm to detect the needle in the US image plane and the ISUR robot is moved to a position allowing the surgeon to mount the needle onto the end-effector. The system is required to wait for an acknowledgment from the surgeon, verifying the correct installation of the needle. Once that the needle is mounted, the ISUR robot moves towards the final target in three steps: first a collision-free trajectory to an approach position is generated and executed, then a sequence of two motion primitives (i.e. two aligned linear trajectories of specified lengths) is executed to get in contact with the skin and then penetrate it. During these operations, the stiffness of the Variable Admittance Controller is modified to ensure a precise insertion of the needle within the soft tissue. The stability during this phase is guaranteed by the technique presented by Ferraguti *et al.* [58]. The robot is finally stopped and waits for the disconnection of the needle

from its end-effector, then the sequence is repeated for the other four needles.

Figure 9 shows images captured during the actual execution of the experiment. Each image is related to a given state of the task sequence. The safe execution of the task is also demonstrated by the fact that forces applied by the ISUR robot at the needle tip are limited within prescribed bounds, as shown in Fig. 10.

Clinical effectiveness of the emulated cryoablation experiment is related to the correct implementation of the supervision and control systems, which is the focus of this paper, but also to the accurate registration and calibration of the robotic setup. The detailed explanation of the methods used for the latter operation has been published by Muradore *et al.* [27]. The overall accuracy of the needle tip positioning with respect to the tumor, verified during system testing, was 5.4 mm. Surgeons involved during the validation of these results admitted that this error is not acceptable for real applications. On the other hand, more precise execution of robot manufacturing and software-based compensation of
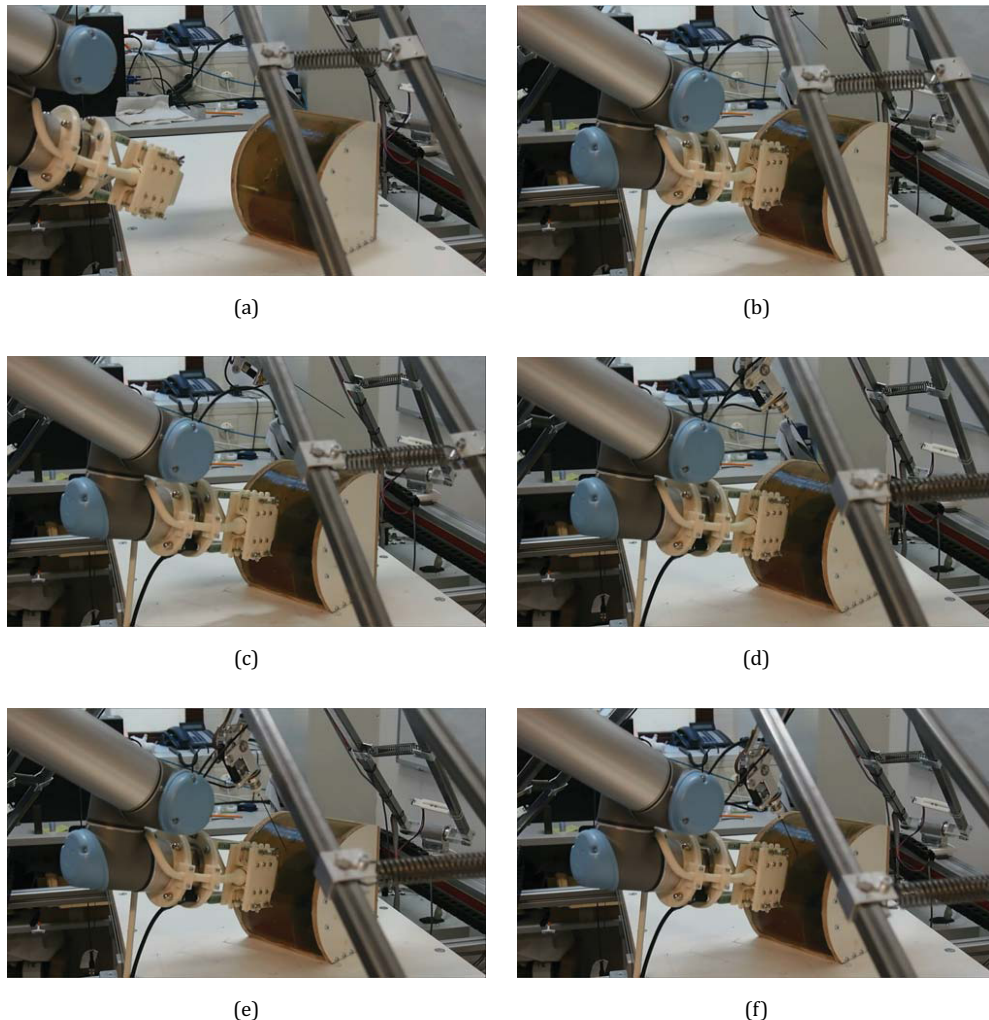
(a)

(b)

(c)

(d)

(e)

(f)

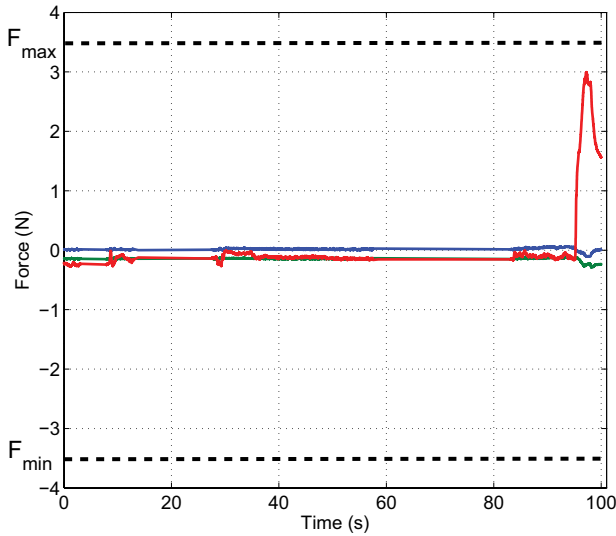**Fig. 9.** Images captured during the puncturing experiment.

**Fig. 10.** Forces applied by the robot during puncturing.

related tolerances should reduce the error within the millimetric bound.

5.2. *Suturing*

The second surgical task under study is the suturing of a planar and linear wound. Figure 11(a) shows the phantom pad where the skin is in yellow and the muscular layer below in red. The wound has an ellipsoidal shape and can be easily and accurately detected by a 3D vision system by color thresholding. Vision system calibration and its registration in the operational space of the robot has been performed using the `calib3d` module of the OpenCV library [61], that implements well-known chessboard-based methods. The absolute average error of this registration on the experimental setup was 0.57 mm. When the edge of the wound are detected and described in mathematical terms (Fig. 11(b)), it is possible to run the algorithm that selects the number of stitches needed to close the wound and their position with respect to the edges. In Fig. 11(c) the nominal position of the stitches on the left edge are indicated with red squares whereas the ones on the right edge with purple squares. This distribution comes from surgical specifications on stitch-to-stitch and stitch-to-edge distances, both required to be equal to 5 mm.

This is the initial step of the sequential procedure that executes the stitches one-by-one in an autonomous way. After the execution of each stitch the system has to:
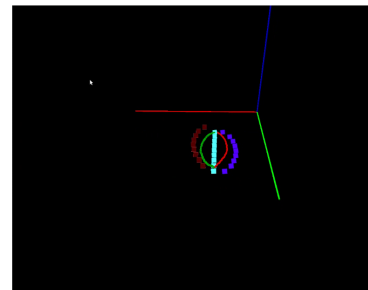
(1) verify that the two edges perfectly overlap nearby the stitch (i.e. no red tissue should be visible around the stitch),
(2) check the actual position of the stitch with respect to the nominal position (in particular the stitch-to-stitch and stitch-to-edge distances have to be monitored),
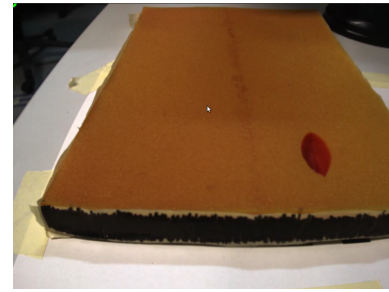
(a)
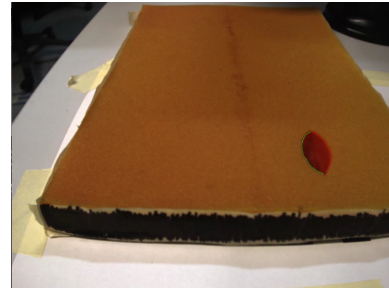
(b)

(c)

Fig. 11. (a) Two-layer phantom pad, (b) edges detection and (c) nominal position of the stitches at the beginning of the procedure.

(3) determine the edges of the wound and re-plan the stitches position according to the actual shape of the wound and the position of the previous stitches.

Once that the sensing module has detected the wound and stitches distribution has been planned, related data are passed to the motion planning and control subsystem for the execution of the task. Online sensing during task execution allows to track the Endo Stitch and verify its correct insertion within the wound. Moreover, online sensing would also be required to detect the thread by means of 3D image processing, so that the left arm of the ISUR robot can properly plan its grasping. However, thread detection turned out to be not reliable with the technology used in the current setup. As a workaround, preliminar experiments with manual motion of the ISUR robot arms revealed the

possibility to mount a small rod, instead of a gripper, on the left arm to simplify the task of pulling away the thread with a *sweeping* motion.

The overall control architecture and its interaction with the sensing module has been tested in parallel with the mechanical assembly of the dual-arm version of the ISUR robot, replacing the physical system with a specifically developed 3D viewer. The viewer animates a full CAD model of the robot, so that it can accept joint position commands just like the low-level controller of real robot. The 3D meshes shown by the viewer are exactly those used for collision-checking in the motion planner, which allows to visually verify the planned paths and the logical sequence of the task. Figure 12 contains
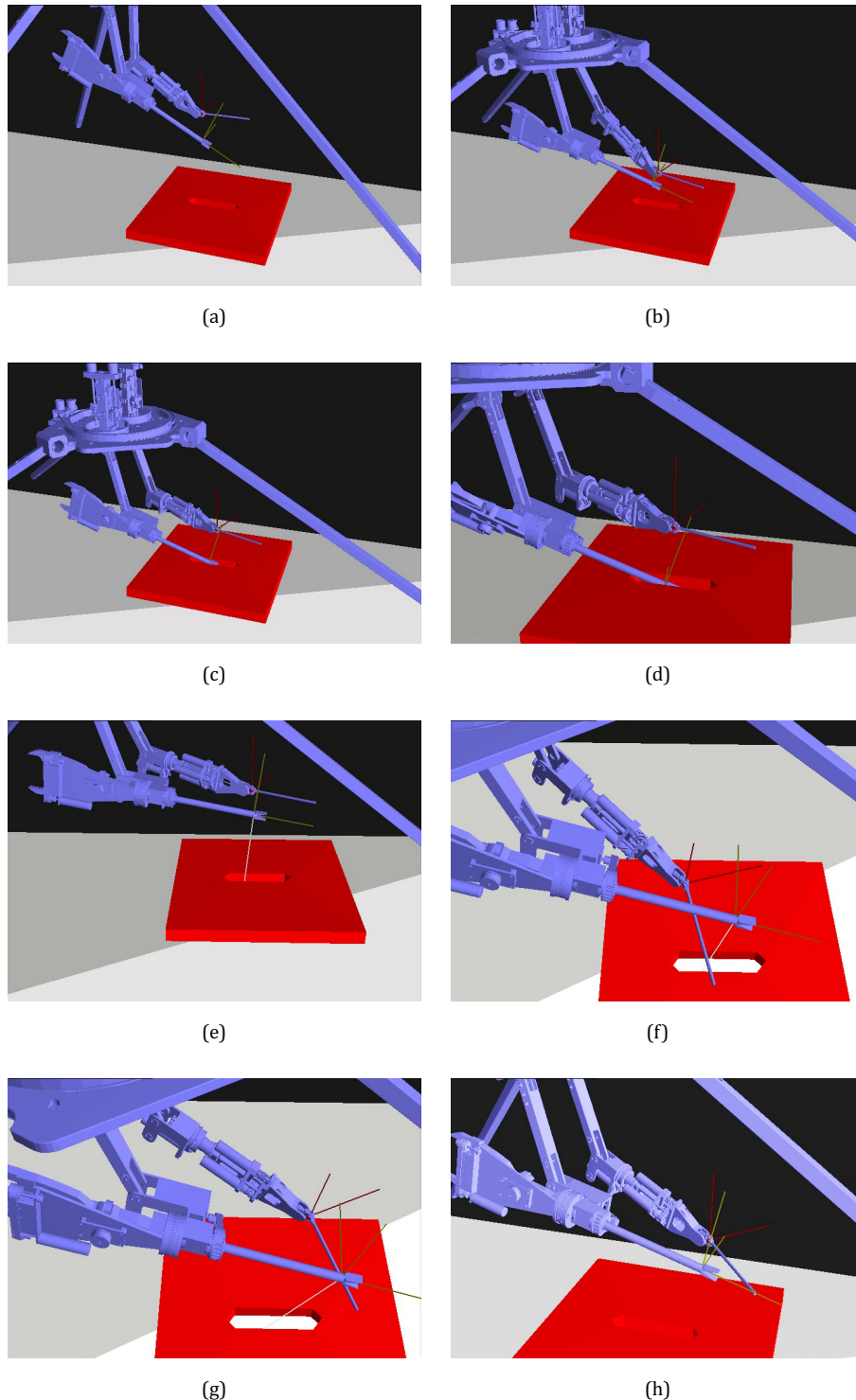

(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

Fig. 12. Frames from a simulated execution of the suturing task.

an excerpt of a full run of the stitching sequence described in Sec. 3.2.

More precisely, the frames show: A, the arms in their initial position; B, after an online planning, the two arms are moved over the wound; C, the tip of the Endo Stitch is positioned inside the wound; D, the Endo Stitch is rotated to touch one edge of the wound; E, both arms are moved following motion primitives to pull the thread; F, the left arm is moved around the thread with a motion primitive; G, the left arm pushes the thread; H, the two arms are moved back over the wound following a collision-free path planned online.

5.3. General assessment

The clinical partner in the I-SUR consortium (San Raffaele Hospital) did an evaluation of the robotic system at the end of the project, involving four urologists and a health manager and applying the so-called *Health Technology Assessment* approach [62]. The surgeons evaluated the following aspects:

- **Technical efficacy and safety features:** even though *accuracy* of the current prototype is an issue, *surgeon assistance* provided by the Reasoning system satisfied surgical requirements, especially those prescribing that the system should be able to easily and promptly switch into a teleoperated mode; *automated planning* phases were evaluated as important and satisfying.
- **Usability:** surgeons evaluated positively *extensibility* of the prototype, even though some concerns were expressed about limiting the suturing task, in the present setup, to planar and regular wounds; the *mechanical configuration* of the robot has its fixed base connected to the bed, but surgeons suggested that a better solution would be to connect the base to the ceiling, to increase the workspace for human operators; finally, *key factors* evaluation summarized as follows (mean values with marks from 0: not important/extremely bad to 5: very important/extremely good):

  – *Expected utility/generalization*: 4.8
  – *More safety margins*: 4.0
  – *Supervisory control*: 4.1
  – *Team coordination*: 3.8
  – *Expected safety improvement*: 3.6
  – *Increased resource perception*: 3.4
  – *Trust in automation*: 3.1

6. Conclusions

In this paper, we presented a robot control and coordination software architecture for the automation of simple surgical tasks, namely needle insertion and suturing. Design specifications were defined using a requirements engineering approach, allowing a formal verification of behavioral requirements and the generation of hierarchical finite state machines for the automated supervision of robotic tasks. Then, the proposed architecture has been implemented using component-based design tools in order to properly handle the distributed nature of the system and apply state-of-the-art robotics software design principles.

The proposed approach has been validated on an experimental setup including a novel surgical robot with a modular mechanical structure and, for the US-guided needle insertion case study, an additional industrial manipulator holding the US probe. The goals of the experiments was to show first the feasibility of full surgical robot automation and the flexibility and reconfigurability of the proposed software architecture, which is the focus of this paper. Clinicians involved during the execution of the experiments evaluated positively the possibility to promptly and smoothly switch the system from the autonomous to the teleoperated mode and the full automation of the pre-operative planning phase. Moreover, it was suggested to mount the fixed base of the ISUR robot on the ceiling, instead of the bed, to increase the workspace and to allow manual surgeons intervention.

Future work aims at:

- extending the proposed cognitive control architecture to address other mechanical setups, including the possibility to operate in a laparoscopic environment and take into account kinematic constraints and friction forces related to the tool-trocar interactions;
- detecting and compensating intra-operative organ motion;
- extending the suturing case to complex wounds on non-planar surfaces;
- improving the accuracy of the overall system, by means of a proper refinement of mechanical manufacturing and calibration.

References

1. P. Kazanzides, G. Fichtinger, G. D. Hager, A. M. Okamura, L. L. Whitcomb and R. H. Taylor, Surgical and interventional robotics-core concepts, technology, and design [tutorial], *IEEE Robot. Autom. Mag.* **15**(2) (2008) 122–130.

2. P. Gomes, Surgical robotics: Reviewing the past, analysing the present, imagining the future, *Robot. Comput.-Integr. Manuf.* **27**(2) (2011) 261–266.

3. R. A. Beasley, Medical robots: Current systems and research directions, *J. Robot.* **2012** (2012).

4. A. Tobergte, R. Konietschke and G. Hirzinger, Planning and control of a teleoperation system for research in minimally invasive robotic surgery, in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, Kobe, Japan, May 2009, pp. 4225–4232.

5. B. Hannaford, J. Rosen, D. W. Friedman, H. King, P. Roan, L. Cheng, D. Glozman, J. Ma, S. N. Kosari and L. White, Raven-II: An open platform for surgical robotics research, *IEEE Trans. Biomed. Eng.* **60**(4) (2013) 954–959.

6. S. A. Bowyer, B. L. Davies and F. R. Baena, Active constraints/virtual fixtures: A survey, *IEEE Trans. Robot.* **30**(1) (2014) 138–157.

7. G. Moustris, S. Hiridis, K. Deliparaschos and K. Konstantinidis, Evolution of autonomous and semi-autonomous robotic surgical systems: A review of the literature, *Int. J. Med. Robot. Comput. Assist. Surg.* **7**(4) (2011) 375–392.

8. E. Franco, M. Rea, W. M. W. Gedroyc and R. M., Needle-guiding robot for percutaneous intervention: Comparative phantom study in a 3T MRI scanner, in *Proc. Hamyln Symp. Medical Robotics*, London, UK, June 2015.

9. L. B. Kratchman, M. M. Rahman, J. R. Saunders, P. J. Swaney and R. J. Webster III, Toward robotic needle steering in lung biopsy: A tendon-actuated approach, *SPIE Medical Imaging*, International Society for Optics and Photonics (2011), p. 79641I.

10. W. Wang, Y. Shi, A. A. Goldenberg, X. Yuan, P. Zhang, L. He and Y. Zou, Experimental analysis of robot-assisted needle insertion into porcine liver, *Biomed. Mater. Eng.* **26**(s1) (2015) 375–380.

11. H. Mayer, I. Nagy, D. Burschka, A. Knoll, E. Braun, R. Lange and R. Bauernschmitt, Automation of manual tasks for minimally invasive surgery, in *Proc. Fourth Int. Conf. Autonomic and Autonomous Systems*, March 2008, pp. 260–265.

12. F. Nageotte, P. Zanne, C. Doignon and M. deMathelin, Stitching planning in laparoscopic surgery: Towards robot-assisted suturing, *Int. J. Robot. Res.* **28** (2009) 1303–1321.

13. J. Schulman, A. Gupta, S. Venkatesan, M. Tayson-Frederick and P. Abbeel, A case study of trajectory transfer through non-rigid registration for a simplified suturing scenario, *2013 IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, November 2013, pp. 4111–4117.

14. T. Liu and M. Cavusoglu, Optimal needle grasp selection for automatic execution of suturing tasks in robotic minimally invasive surgery, *2015 IEEE Int. Conf. Robotics and Automation (ICRA)*, May 2015, pp. 2894–2900.

15. Z. Baili, I. Tazi and Y. Alj, StapBot: An autonomous surgical suturing robot using staples, *2014 Int. Conf. Multimedia Computing and Systems (ICMCS)*, April 2014, pp. 485–489.

16. B. Fei, W. S. Ng, S. Chauhan and C. K. Kwoh, The safety issues of medical robotics, *Reliab. Eng. Syst. Saf.* **73**(2) (2001) 183–192.

17. M. Y. Jung, R. H. Taylor and P. Kazanzides, Safety design view: A conceptual framework for systematic understanding of safety features of medical robot systems, *2014 IEEE Int. Conf. Robotics and Automation (ICRA)*, (IEEE, 2014), pp. 1883–1888.

18. M. Y. Jung, M. Balicki, A. Deguet, R. H. Taylor and P. Kazanzides, Lessons learned from the development of component-based medical robot systems, *J. Softw. Eng. Robot.* **5**(2) (2014) 25–41.

19. A. Gulhar, D. Briese, P. W. Mewes and G. Rose, Registration of a robotic system to a medical imaging system, in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Hamburg, Germany (2015), pp. 3208–3213.

20. F. Vicentini, P. Magnoni, M. Giussani and L. Molinari Tosatti, Analysis and compensation of calibration errors in a multi-robot surgical platform, in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Hamburg, Germany (2015), pp. 3208–3213.

21. F. Šuligoj, B. Jerbić, M. Švaco, B. Šekoranja, D. Mihalinec and J. Vidaković, Medical applicability of a low-cost industrial robot arm guided with an optical tracking system, in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Hamburg, Germany (2015), pp. 3785–3790.

22. C. N. Cho, J. H. Seo, H. R. Kim, H. Jung and K. G. Kim, Vision-based variable impedance control with oscillation observer for respiratory motion compensation during robotic needle insertion: A preliminary test, *Int. J. Med. Robot. Comput. Assist. Surg.* (2015).

23. C. Pappone, F. Vicedomini, G. Manguso, F. Gugliotta, P. Mazzone, S. Gulletta, N. Sora, S. Sala, A. Marzi, G. Augello, L. Livolsi, A. Santagostino and V. Santinelli, Robotic magnetic navigation for atrial fibrillation ablation, *J. Am. Coll. Cardiol.* **47**(7) (2006).

24. N. Padoy and G. D. Hager, 3D thread tracking for robotic assistance in tele-surgery, in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, September 2011, pp. 2102–2107.

25. ROBODOC, Curexo Technology Corporation, http://www.robodoc.com.

26. R. Muradore, G. De Rossi, M. R. Bonfè, N. Preda, C. Secchi, F. Ferraguti and P. Fiorini, Autonomous execution of surgical tasks: The next step in robotic surgery, in *Proc. Hamyln Symp. Medical Robotics*, London, UK, June 2015, pp. 83–84.

27. R. Muradore, P. Fiorini, G. Akgun, D. E. Barkana, M. Bonfe, F. Borierol, A. Caprara, G. De Rossi, R. Dodi, O. J. Elle *et al.*, Development of a cognitive robotic system for simple surgical tasks, *Int. J. Adv. Robot. Syst.* **12** (2015) 1–20.

28. M. Bonfè, F. Boriero, R. Dodi, P. Fiorini, A. Morandi, R. Muradore, L. Pasquale, A. Sanna and C. Secchi, Towards automated surgical robotics: A requirements engineering approach, in *Proc. IEEE RAS & EMBS Int. Conf. Biomedical Robotics and Biomechatronics (BioRob)* (2012), pp. 56–61.

29. S. Permpongkosol, M. Nielsen and S. Solomon, Percutaneous renal cryoablation, *Urology* **68**(1 Suppl.) (2006) 19–25.

30. A. Sharon, N. Hogan and D. Hardt, The macro/micro manipulator: An improved architecture for robot control, *Robot. Comput.-Integr. Manuf.* **10**(3) (1993) 209–222.

31. R. Öpik, A. Hunt, A. Ristolainen, P. M. Aubin and M. Kruusmaa, Development of high fidelity liver and kidney phantom organs for use with robotic surgical systems, in *Proc. IEEE RAS & EMBS Int. Conf. Biomedical Robotics and Biomechatronics (BioRob)* (2012), pp. 425–430.

32. K. Mathiassen, D. Dall'Alba, R. Muradore, P. Fiorini and O. J. Elle, Real-time biopsy needle tip estimation in 2D ultrasound images, in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, Karlsruhe, Germany, May 2013.

33. J. Yoo, E. Jee and S. Cha, Formal modeling and verification of safety-critical software, *IEEE Software* **26** (2009) 42–49.

34. P. Kazanzides, K. Y., A. Deguet and Z. Shao, Proving the correctness of concurrent robot software, in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, May 2012, pp. 4718–4723.

35. Y. Kouskoulas, D. Renshaw, A. Platzer and P. Kazanzides, Certifying the safe design of a virtual fixture control algorithm for a surgical robot, in *Proc. 16th Int. Conf. Hybrid Systems: Computation and Control (HSCC)* (2013), pp. 263–272.

36. R. Muradore, D. Zerbato, L. Vezzaro, L. Gasperotti and P. Fiorini, From simulation to abstract modeling of surgical operations, *Joint Workshop on New Technologies for Computer/Robot Assisted Surgery*, Madrid, Spain, 9–10 July 2012.

37. L. Baresi, L. Pasquale and P. Spoletini, Fuzzy goals for requirements-driven adaptation, in *Proc. Int. Requirements Engineering Conf.* (2010), pp. 125–134.

38. D. Jackson, Alloy, (2012), Available at: http://alloy.mit.edu/.
39. C. Baier and J.-P. Katoen, *Principles of Model Checking* (MIT Press, 2008).
40. E. Torlak and G. Dennis, Kodkod for alloy users, in *Proc. 1st ACM Alloy Workshop*, April 2008.
41. Object Management Group, UML v. 2.2 Superstructure specification Document N. formal/2009-02-02 (2009), Available at http://www.omg.org/spec/UML/2.2/.
42. K. McMillan, *Symbolic Model Checking: An Approach to the State Explosion Problem* (Kluwer Academic Publishers, 1993).
43. K. McMillan, *The SMV Language* (Cadence Berkeley Labs., 2001 Addison St., Berkeley, USA, 1999).
44. H. Nguyen, O. Elle, D. Handini and K. Mathiassen, Intra-operative reasoning and situation awareness algorithm for kidney tumor cryoblation by robot, *24th Int. Conf. Society for Medical Innovation and Technology (SMIT)*, September 2012.
45. M. Torricelli, F. Ferraguti and C. Secchi, An algorithm for planning the number and the pose of the iceballs in cryoablation, in *Proc. Int. Conf. IEEE Engineering in Medicine and Biology Society (EMBC)*, Osaka, Japan, July 2013.
46. The Orocos Project, Smarter control in robotics and automation, Available at http://www.orocos.org.
47. M. Klotzbücher and H. Bruyninckx, Coordinating robotic tasks and systems with rFSM statecharts, *J. Softw. Eng. Robot.* **3** (2012) 28–56.
48. M. Bonfè, C. Fantuzzi and C. Secchi, Verification of behavioral substitutability in object-oriented models for industrial controllers, in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, Barcelona, Spain, April 2005.
49. E. Clarke and S. Gao, Model checking hybrid systems, in *Leveraging Applications of Formal Methods, Verification and Validation. Specialized Techniques and Applications*, eds. T. Margaria and B. Steffen, Lecture Notes in Computer Science, Vol. 8803 (Springer Berlin Heidelberg, 2014), pp. 385–386.
50. R. Muradore, D. Bresolin, L. Geretti, P. Fiorini and T. Villa, Robotic surgery — formal verification of plans, *IEEE Robot. Autom. Mag.* **18** (2011) 24–32.
51. D. Bresolin, L. Geretti, R. Muradore, P. Fiorini and T. Villa, Formal verification applied to robotic surgery, in *Coordination Control of Distributed Systems*, eds. J. H. van Schuppen and T. Villa, Lecture Notes in Control and Information Sciences, Vol. 456 (Springer International Publishing, 2015), pp. 347–355.
52. Object Management Group, CORBA (Common Object Request Broker Architecture) specifications, Available at http://www.corba.org.
53. ROS, An open-source robot operating system, Available at http://www.ros.org.
54. J. Kuffner and S. LaValle, RRT-connect: An efficient approach to single-query path planning, in *Proc. IEEE Int. Conf. Robotics and Automation* (2000), pp. 995–1001.
55. I. A. Şucan, M. Moll and L. E. Kavraki, The open motion planning library, *IEEE Robot. Autom. Mag.* **19** (2012) 72–82.
56. L. Biagiotti and C. Melchiorri, *Trajectory Planning for Automatic Machines and Robots* (Springer-Verlag, 2008).
57. L. Villani and J. De Schutter, Force control, in *Springer Handbook of Robotics*, eds. B. Siciliano and O. Khatib (Springer Berlin Heidelberg, 2008).
58. F. Ferraguti, N. Preda, A. Manurung, M. Bonfe, O. Lambercy, R. Gassert, R. Muradore, P. Fiorini and C. Secchi, An energy tank-based interactive control architecture for autonomous and teleoperated robotic surgery, *IEEE Trans. Robot.* **31** (2015) 1073–1088.
59. N. Preda, A. Manurung, O. Lambercy, R. Gassert and M. Bonfè, Motion planning for a multi-arm surgical robot using both sampling-based algorithms and motion primitives, in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Hamburg, Germany, 28 September–3 October 2015, pp. 1422–1427.
60. M. Franken, S. Stramigioli, S. Misra, C. Secchi and A. Macchelli, Bilateral telemanipulation with time delays: A two-layer approach combining passivity and transparency, *IEEE Trans. Robot* **27**(4) (2011) 741–756.
61. OpenCV, Open-source computer vision, Available at http://docs.opencv.org/2.4/index.html.
62. R. Dodi and A. Sanna, A new method for the assessment of percutaneous cryoablation, *IARMM 3rd World Congress of Clinical Safety*, Madrid, Spain, 10–12 September 2014.
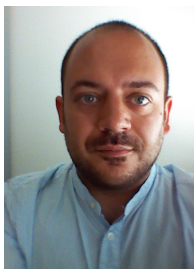
Nicola Preda received his M.Sc. degree in 2011, and the Ph.D. degree in 2015, both in Automation Engineering at the University of Ferrara. Currently, he is a Postdoctoral Research Fellow at the Department of Engineering of the University of Ferrara, Italy. His research interests include system and software architecture, software engineering and control of robotic systems.

Giacomo De Rossi received the Bachelor degree in Computer Science from the University of Verona, Verona, Italy, in 2012. He works in the ALTAIR Robotics Laboratory of the University of Verona, where he contributes to the development of algorithms for surgical robots.

Federica Ferraguti received the M.Sc. degree in Industrial and Management Engineering and the Ph.D. in Industrial Innovation Engineering from the University of Modena and Reggio Emilia, Italy, in 2011 and 2015. She has been visiting student at the Rehabilitation Engineering Lab at ETH Zurich, Switzerland in 2013. She is currently a Postdoctoral Research Fellow at the University of Modena and Reggio Emilia. Her research deals with surgical robotics, telerobotics, control of robotic systems and human–robot physical interaction.

Cristian Secchi graduated in Computer Science Engineering at the University of Bologna in 2000 and he received his Ph.D. in Information Engineering in 2004 from the University of Modena and Reggio Emilia, where he is currently Associate Professor. His Ph.D. thesis has been selected as one of the three finalists of the 5th Georges Giralt Award for the best Ph.D. thesis on robotics in Europe. He participated to the CROW project, selected as one of the finalists for the 2010 EUROP/EURON Technology Transfer Award for the best Technology transfer project in Europe. He has been an Associate Editor for the *IEEE Robotics and Automation Magazine* and he is currently serving as an Associate Editor of the *IEEE Transactions on Robotics* and of the *IEEE Robotics and Automation Letters*. His research deals with human–robot physical interaction, telerobotics, mobile robotics and surgical robotics and he has published more than 100 papers in international journals and conferences.

Riccardo Muradore received the Laurea degree in Information Engineering in 1999 and the Ph.D. degree in Electronic and Information Engineering in 2003 both from the University of Padova (Italy). He held a post-doctoral fellowship at the Department of Chemical Engineering, Univ. of Padova, from 2003 to 2005. Then he spent three years at the European Southern Observatory in Munich (Germany) as control engineer working on adaptive optics systems. In 2008, he joined the ALTAIR robotics laboratory, University of Verona (Italy). Since 2013 he is an Assistant Professor. His research interests include robust control, teleoperation, robotics, networked control systems and adaptive optics.

Paolo Fiorini received the Laurea degree in Electronic Engineering from the University of Padova, (Italy), the MSEE from the University of California at Irvine (USA), and the Ph.D. in ME from UCLA (USA). From 1985 to 2000, he was with NASA Jet Propulsion Laboratory, California Institute of Technology, where he worked on telerobotic and teleoperated systems for space exploration. From 2000 to 2009 he was an Associate Professor of Control Systems at the School of Science of the University of Verona (Italy) where he founded the ALTAIR robotics laboratory with his students. He is currently a Full Professor of Computer Science at the University of Verona. His research focuses on teleoperation for surgery, service and exploration robotics funded by several European Projects. He is an IEEE Fellow (2009).

Marcello Bonfè received the M.Sc. degree in Electronic Engineering in 1998, and the Ph.D. in Information Engineering in 2003. He is an Assistant Professor of Automatic Control at the University of Ferrara, Italy. He has published more than 70 refereed journal and conference papers and his main research interests are: formal verification of discrete event systems, modeling and control of mechatronic systems, fault detection and fault tolerant control, robotics and motion planning.