Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

Distributed Control Architecture for Automated Surgical Task Execution with Coordinated Robot Arms

Marcello Bonfè * **Nicola Preda** * **Cristian Secchi** **
Federica Ferraguti ** **Riccardo Muradore** *** **Luisa Repele** ***
Giovanni Lorenzi *** **Paolo Fiorini** ***

* *Engineering Department, University of Ferrara, Italy (e-mail: name.surname@unife.it).*
** *Department of Science and Methods for Engineering, University of Modena and Reggio Emilia, Italy (e-mail: name.surname@unimore.it)*
*** *Department of Computer Science, University of Verona, Italy, (e-mail: name.surname@univr.it)*

Abstract: The paper describes a robot control and coordination framework for the automation of surgical tasks. In the proposed framework, surgeons are supported by autonomous robotic assistants and do not teleoperate robots, unless in case of exceptions in the tasks of the robots. Such robots perform basic surgical actions by combining sensing, dexterity and cognitive capabilities. The goal is achieved thanks to rigorous assessment of surgical requirements, formal specification of robotic system behavior, including multiple arm coordination and human/system interaction, and control software development with *state-of-the-art* component-based technologies. The paper presents an experimental setup composed of two robots operating on a US-compatible phantom, demonstrating the feasibility of the approach.

Keywords: Teleoperation, Finite State Machine, UML, Force Control.

1. INTRODUCTION

The introduction of minimally invasive surgery first and, more recently, of surgical robots, has brought new perspectives to surgery and has significantly improved the quality of many critical surgical tasks. Nowadays, surgical robots are usually teleoperated by the surgeons, as in the case of the well-known da Vinci by Intuitive Surgical (`www.intuitivesurgical.com`). A more recent telesurgey system has been developed by the German Aerospace Centre (DLR) Tobergte et al. (2009). However, teleoperated robots are not the final answer to surgeon's accuracy demands. The possibility to carry out simple surgical actions automatically has been the subject of academic research. A remotely-controlled catheter guiding robot was used in Pappone et al. (2006) to automatically perform cardiac ablation. However, an experienced operator is required to perform all the procedures. Several works (see e.g. Mayer et al. (2008)) have been done towards the automation of knot tying in suturing tasks, but requiring manual help in several preparatory stages (e.g. grasping the needle). Automatic scissors were proposed in Padoy and Hager (2011), namely the possibility for a surgeon to invoke a third robotic arm to come and automatically cut the thread that he/she is holding. Though all these works constitute successful automation of simple surgical actions, validated and commercially distributed autonomous surgical robots are quite rare. A notable exception is represented by ROBODOC (`www.robodoc.com`), a system capable of interventions on rigid tissues (i.e. bones drilling or cutting).

This work is a part of a research project whose goal is to develop a robotic system that can autonomously execute surgical tasks on soft tissues. In particular, this paper describes the design and realization of a robotic system, controlled by a distributed architecture, capable of autonomously executing US-guided insertion of needles into soft bodies, emulating the surgical procedure for percutaneous cryoablation of small tumoral masses. This task is also called simply *puncturing*. Exploiting the framework proposed in Bonfè et al. (2012), we show that formal methods can be applied to describe the US-supervised puncturing procedure in a precise way. This formal description enables automatically the design and software implementation of the robotic control system. Furthermore, for validation purposes, we implemented the proposed architecture on an experimental setup, made by two robotic manipulators that autonomously perform the puncturing task: one robot holds the needle and moves it according to a planned trajectory to perform the puncturing, while the other robot holds an US probe which images are used to intra-operatively guide and control the needle insertion. Even the pre-operative planning is performed automatically, by means of a software, in the following called *cryo-planner*, that implements the algorithm proposed in Torricelli et al. (2013). Thanks to the modular and component-based architecture of the system,

⋆ The research leading to these results has received funding from the European Union Seventh Framework Programme FP7/2007-2013 under grant agreement n. 270396 (I-SUR).

the same methodology and design approach can be applied to automate other simple surgical tasks.

The paper is organized as follows: Section 2 introduces the surgical task selected as case study 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 UML design and the system architecture. Finally, results collected during the execution of needle insertion experiments are shown in Section 5, which is followed by conclusions in Section 6.

2. CASE STUDY AND ROBOTIC SETUP

This paper focuses, among the surgical procedures selected to evaluate the feasibility of robotic automation, on percutaneous cryoablation of small tumors. Nevertheless, the same methodology and design approach will be applied in a near future to automated suture. Percutaneous cryoablation requires the use of pre- and intra-operative images (CT, MRI/US) to insert, through the skin, one or more cryoprobe needles into the tumoral mass to be destroyed and to check the real-time position of the tools inside the patient. Trajectory misalignments are usually due to the deformation of soft tissues and organ displacement because of respiration. Thanks to real-time image registration and accurately calibrated mechanical arms, needle insertion would be precisely executed by the robotic system. The cryoablation operation involves, once needles are correctly inserted, cycles of freezing and thaw to create an *iceball* covering and killing the tumoral cells, while preserving the healthy tissue and the surrounding abdominal structures (see Permpongkosol et al. (2006)). Major complications refer to bleeding and organ damages caused by the extraction of the probes when the ice ball is not melted enough to release the needles. The evaluation of the force applied to extract a needle from the patient is, for human surgeons, the only way to detect the melting status of the iceball. Even in this case, robotic assistants equipped with force sensors and intraoperative processing of US images would increase safety margins during completion of the surgical procedure.

To evaluate practical issues and benefits of cryoablation execution by means of automated robots, an experimental setup has been prepared, as shown in Figure 1. The setup allows to emulate a cryoablation operation, apart from the actual freezing/defreezing cycle. In fact, cryoablation devices can only be used in real operating rooms, while the experimental system is installed in an academic laboratory. The proposed system includes two industrial robotic manipulator (Unimate Puma260 robots retrofitted with modern control hardware and software): the first robot holds the needle, while the second holds the US probe. The end-effector of both robots is equipped with a specific tool adapter that integrates a 6-DOF force/torque sensor. Two types of phantom emulating a human abdomen have been used for experiments: in the first, artificial organs, produced using high-fidelity CAD models as described in Öpik et al. (2012), are enclosed by a tissue that replicates the features of human skin; in the other, providing a simpler, lower cost and easily disposable alternative to the higher-fidelity one, ex-vivo animal tissues are embedded into a mixture of water and corn flour. In both cases, the phantom is compatible with US imaging. An 3-D optical tracking system is used to estimate relative coordinate transformations among the robots and the phantom. Finally, the setup includes an ultrasound imaging device whose images can be visualized on a dedicated graphical interface for the surgeons and processed in real-time to detect the position of needle tip, as shown in Mathiassen et al. (2013), and provide intra-operative adaptation of robot motion trajectories.

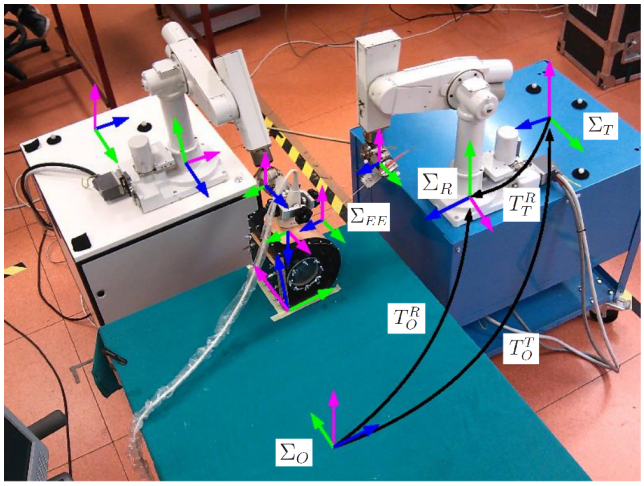

Fig. 1. Experimental setup with coordinate systems and related transformations

3. CONTROL AND COORDINATION

3.1 Development process

Validation-oriented design is mandatory for the application domain of surgical robotics. Therefore, design specifications for control algorithms and supervisory/coordination logic have been formalized using a requirements engineering approach, which is a more and more recommended practice for safety-critical systems design. In particular, the methodology applied in this project, described more precisely in Bonfè et al. (2012), is developed as follows:

(1) **Requirement collection**: a group of expert surgeons is interviewed on the objectives of the surgical process, the main procedures ("best practice") to be performed, the elements of the domain and the critical events related to the surgical actions.
(2) **Requirements engineering**: surgical requirements are expressed using a goal-oriented methodology called FLAGS (Fuzzy Live Adaptive Goals for Self-adaptive systems, see Baresi et al. (2010)). In particular, the *goal model* formalizes structural and behavioral constraints with the *Alloy language* (see Jackson (2012)) and Linear Temporal Logic (LTL, Pnueli (1977)).
(3) **Operationalization**: the goal model is transformed into a sequence of operations and adaptations, satisfying the goals of the surgical procedure. This task is formally defined as a constraint satisfaction problem, whose solution is obtained by using the SAT-solver Kodkod embedded in the Alloy Analyzer tool, as

shown in Torlak and Dennis (2008). As a result, the tool provides a state machine representing the whole system behavior.

(4) **Modular System Design**: the state model obtained after goal-oriented analysis is refined and partitioned into the structural units of the overall automated system, applying decomposition methods from classical discrete systems theory and using UML (Unified Modeling Language, `www.uml.org`) as a modeling tool. The UML model is then mapped into a component-based software architecture.

(5) **System Verification**: formal tools like Model Checking (see McMillan (1993)) are applied to verify that the UML system 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), since the UML language itself does not include a strictly formal one.

(6) **Experimental Validation**: this task is of course mandatory before the clinical use of automated devices. Though the aim of the proposed research is to evaluate the feasibility of surgical robotics automation only using artificial phantoms, demonstrations involving expert surgeons directly interacting with the system are primarily important even to achieve this goal.

3.2 System design

The autonomous system being designed in this project is supervised and controlled by the following modules, corresponding also to software units distributed on different computational platforms: a *Surgical Interface*, the *Robot Controllers* and the *Sensing* system with *Reasoning and Situation Awareness* capabilities. In particular, the Surgical Interface is a software-intensive system allowing humans (i.e. surgeons and technicians) to drive the system during both the pre-operative and the intra-operative phase. In the first one, the focus is on detailed planning of the surgical intervention (e.g. enumeration and placement of cryoablation needles for maximal tumor coverage thanks to the *cryo-planner* of Torricelli et al. (2013)). During operations, the interface should provide real-time visual navigation of the surgical scenario and, if necessary, let surgeons take control of the system (e.g. by switching to 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 control logic of each robot, specified by a UML State Diagram. Safety-critical requirements put a strong demand for strict coordination of these components with both the Surgical Interface and the Sensing/Reasoning module. Finally, the composite sub-system implementing advanced Sensing algorithms and Reasoning for Situation Awareness provides 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, so that appropriate corrective actions can be triggered. The interaction among such system components has been specified with the help UML Sequence Diagrams, which represents scenarios compatible with a given collaborative behavioral specification, including nominal task execution and possible adaptations.

The complete behavioral specification of the robot control and supervision units is given by UML State Diagrams associated to the control logic for the robot holding the needle and for the robot holding the US probe. Figure 2 shows the hierarchical state machine related to the needle inserting robot. As can be seen, the hierchical 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 `e_STOP` event, that can be triggered either by the surgeons, through the Surgical Interface, or by the Sensing/Reasoning and Situation Awareness module. In particular, the latter is in charge of detecting if the needle is too close or even touching a forbidded region, like for example a bone, a nerve or another organ not involved in the cryoablation, by means of real-time monitoring of the needle motion within an anatomical atlas of the patient. Another undesired event is triggered if any force value measured by the sensors exceeds a given limit. 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. Eventually, the system allows the surgeon to switch to a teleoperated mode.

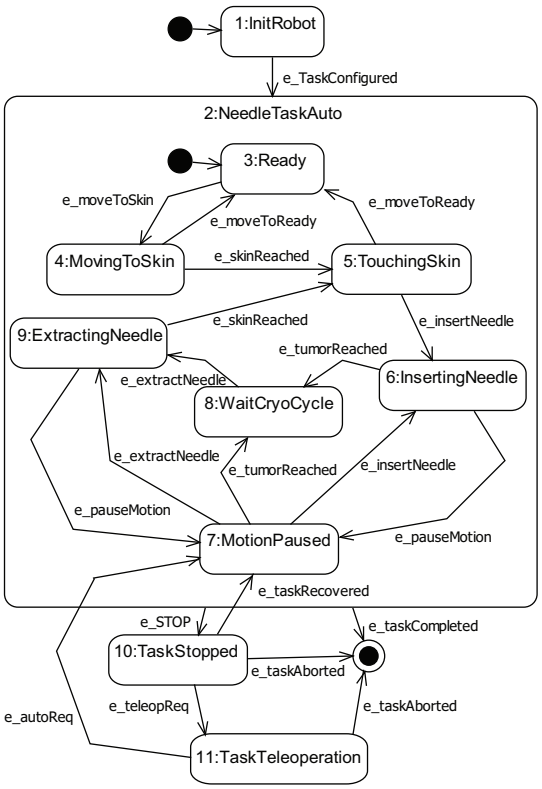

Fig. 2. UML State Diagram of the behavioral specification for the controller of robot holding the needle

3.3 Model Checking

Formal verification of the UML design model requires the definition of its operational semantics, according to

the execution model of the target implementation framework. Since the proposed UML design has been implemented using the component-based Orocos framework `www.orocos.org`) and rFSM (written in Lua, `www.lua.org`), an execution engine for hierarchical state machines, the peculiar features of the latter must be carefully considered. A detailed analysis of the semantics of the rFSM model and its differences with the one informally described by the UML standard can be found in Klotzbücher and Bruyninckx (2012). Summarizing, the main differences are: UML assumes that events are queued and processed one at a time, while rFSM collects all events occurred since the previous *step* of the machine and clears them after the step execution; UML gives higher priority to transitions whose source state is at lower hierarchical levels, while rFSM reverts the rule; rFSM does not support concurrency.

Assuming that an rFSM machine is embedded into a given Orocos component with input and output *event ports*, it is possible to formalize the UML design model implemented in Orocos-rFSM as a modular transition system, as described in Bonfè et al. (2005). The reference provides also rules to translate the formal model into the input language of the Cadence SMV (Symbolic Model Verifier) tool, which is still one of the model checkers that handles most efficiently the well-known state-space explosion problem.

It is important to remark that the events collection mechanism of rFSM is quite different from the PLC-like execution model described in Bonfè et al. (2005) and required a specific adaptation. In particular, the modular features of the SMV language have been exploited to define an event module, whose internal boolean state is true if the event has occurred, but has not been 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_STOP,..)
VAR
e_STOP_ev: rFSM_EV(e_STOP, (Exec = FINISHED));
..
Exec : {IDLE, STEP, FINISHED};
```

The module has also a boolean input `ExecStep` that triggers the execution of its step. This input will be set by an external *scheduling function*. Input events are cleared when the step execution is completed, a condition defined by a value of the enumerated variable `Exec`.

The UML State Diagram specifying the behavior of an rFSM module is encoded preserving the hierarchy of states into variables with enumerated values like:

```
Root : {InitRobot, NeedleTaskAuto, TaskStopped, ..};
SUBNeedleTaskAuto : {Ready, ..., MotionPaused};
```

and the predicates to evaluate the current configuration of the machine, the set of enabled and firable (i.e. higher piority) transitions and the initialization and execution of a step, similar to those described in Bonfè et al. (2005).

An SMV program is completed by the declaration of a `main` module (i.e. the top-level) and by the specification of desired properties of the system. As said before, the desired properties can be expressed using LTL, in the same way that leaf goals are specified by the goal model of the requirements specification.

4. SYSTEM ARCHITECTURE

In this section the main Orocos software components of the control architecture developed for the experimental setup presented in Section 2 are described. Such components are highly configurable and reusable, so that in a near future they will be easily adapted to a newer robotic structure, specifically designed for the surgical applications addressed within the I-SUR project (`www.isur.eu`).

OptiTrack interface and Registration components: these components wraps respectively the communication via TCP/IP with the OptiTrack device and the algorithms for estimating the required homogeneous transformations among the reference systems of interest, shown in Figure 1. The first component provides the poses of the groups of markers associated to the robot with the needle and the robot holding the US probe. The actual registration, instead, allows to estimate the transformation T_R^O mapping points in the robot reference system Σ_R into the OptiTrack reference system Σ_O and transformation T_T^O relating the reference systems of the trolley Σ_T and of the robot.

All these homogeneous transformations allow us to perform the following pre-operative initialization:

(1) map the points on the skin and inside the tumor coming from the cryo-planner and given in the phantom reference system P_{skin}^P, P_{tumor}^P into the global reference system P_{skin}^O, P_{tumor}^O,
(2) map these points into the two robot reference systems: P_{skin}^{Rn}, P_{tumor}^{Rn}, P_{skin}^{Rus}, P_{tumor}^{Rus}, where the n stands for needle and us for US,
(3) plan the movement of the robot holding the needle to guarantee a straight line trajectory in Cartesian coordinate between P_{skin}^{Rn} and P_{tumor}^{Rn},
(4) plan the movement of the robot holding the US probe to guarantee that the needle (i.e. the straight line trajectory designed before) is in the working region of the US probe.

Trajectory Generator component: this component can compute reference trajectories either in joint space or in Cartesian space, from a given initial position to the target position.

ForceSensor interface component: in the experiments two instances of this component have been used, one for an AtiMini45 force-torque (F/T) sensor, mounted on the US probe holding adapter, and the other for an AtiNano17 F/T sensor, mounted on the needle holding adapter. The component receives sensor measurements through a TCP/IP communication with Ati NetBox data-acquisition system and provides a vector of double precision values,

containing forces (F_x, F_y, F_z) and torques (T_x, T_y, T_z), on a real time output port.

Force/Position Control component: this component implements the joint space position control (using a PID scheme) and a direct force control algorithm, implemented as an outer loop cascaded with position control, as described in Siciliano and Villani (1999). The outer force control loop is not always active, because the robots move freely at the beginning and when they touch the phantom a force controller is needed.

SOEM Interface component: this component uses the SOEM (Simple Open EtherCAT master) to communicate with the robot control electronics developed specifically for the proposed experimental setup. SOEM is an open-source project implementing an EtherCAT (Ethernet for Control Automation Technology) master node. The control electronics includes a EtherCAT slave modules for encoder acquisition (Beckhoff EL5152) and analog outputs (Beckhoff EL4004), the latter connected to Maxon Motors power amplifiers. Thanks to this software and hardware setup, the complete control system for a Puma260 robot can be deployed as shown in Figure 3.

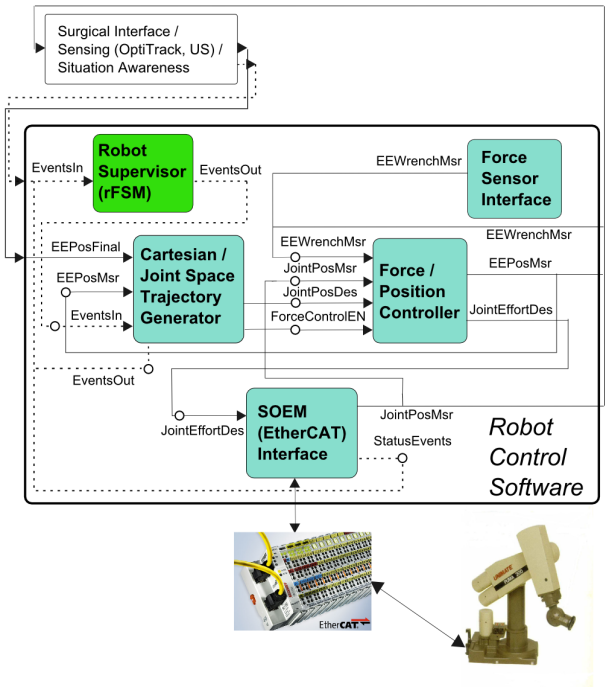

Fig. 3. Orocos components deployment for the control software implementation of the robot holding US probe

Communication Bridge component: this component handles the communication among components running in different computers, so that the full collaborative behavior of the overall architecture (i.e. including the Situation Awareness module and the Surgical Interface) can be executed. The component provide the additional degree of flexibility and extensibility of the system, thanks to the definition of a specific Orocos-CORBA interface.

5. EXPERIMENTAL RESULTS

This section reports the result of an experiments which can be also seen in the video that can be found at `http://youtu.be/HQkhNzm6O8I`. The goal is to show and compare the time series of forces and the states of the robots control logic (see Figure 2) The proposed experimental setup goes through the following steps triggered by the surgeon through the user interface:

(1) The surgeon pushes the **Ready** button. Both robots move from their nest position to the ready position where the needle and the US probe are mounted. At the end of this phase both robots are in the "Ready" state.
(2) The surgeon pushes the **Start** button. The robots receive their nominal trajectories and move the needle and the US probe in contact with the phantom. During this phase the robot controllers go through the following states: "MovingToSkin" and "TouchingSkin".
(3) When the surgeon pushes the **InsertNeedle** button, the robot holding the needle starts the insertion ("InsertingNeedle") until it reached the target point. In this case the state of its control logic is "WaitCryoCycle".
(4) Since the actual cryoablation is not available in the current setup, the final step is for the surgeon to push the button **Finish** to bring the robots back in the "Ready" state.

Figure 4 reports the force applied by the US probe on the phantom, along the approaching axis only.

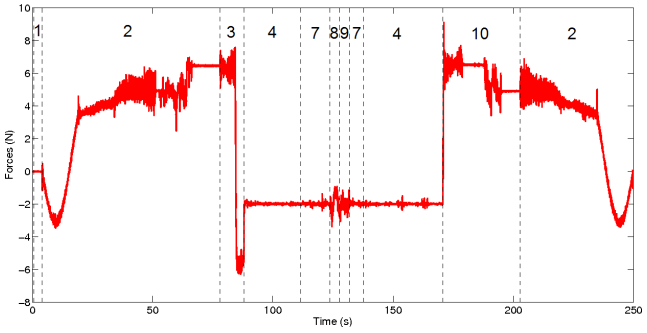

Fig. 4. Force applied to the US probe during the emulated cryoblation task.

The robot starts from an initial idle state (1). When the surgeon presses the **Ready** button the robot goes to a "ready" position (2). During this phase the force is not controlled because the robot is moving in free motion. In "Ready" the US probe is mounted on the end-effector. During the phase (3) the US probe reaches the skin and when the measured force along the main axis of the probe exceed a threshold (4), the force controller is switched on. This controller stabilizes the force around 2 N which guarantees a good quality of the US images. During the procedure, the probe executes some rotations to emulate the research of the needle in US image, in phases (7),(8),(9). These rotations change a little bit the force as it is possible to see in Figure 4. At the end of the procedure the robot is in a "TaskStopped" state of the rFSM. Figure 5 shows the force measured by the F/T sensor located on the robot holding the needle along the

main axis of the needle. The oscillations in the phase "Ready" (3) are only due to a small flexibility of the current prototype of the needle holding adapter.

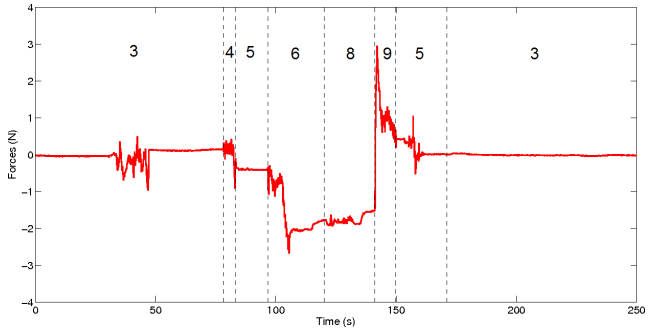

Fig. 5. Force applied to the needle during the emulated cryoblation task. (The numbers refer to the states in Figure 2.)

In the "MovingToSkin" (4) phase the force changes because the robot moves in free motion and the sensor is extremely sensitive. At the end of this movement the tip of the needle touches the skin and so the force is constant. During the insertion, the needle passes through different layers (skin, tumor) and this explains why the profile of the force increases (in absolute value). The force becomes stable during the "WaitCryoCycle" (8) state. After few seconds the needle is extracted and so the force values have opposite sign. The needle is removed in state (5) and the robot goes back in nest under pure position control.

6. CONCLUSIONS

In this paper we presented a robot control and coordination framework for the automation of simple surgical tasks. We formalized the design specifications using a requirements engineering approach and derived the state machines for the control of the robots involved in the operation. Then, we implemented the proposed architecture using component-based design tools, i.e. Orocos framework and rFSM, in order to handle properly the distributed nature of our system. The proposed approach has been validated through an experimental setup where two robots execute autonomously a puncturing task on a phantom replicating the human abdomen. The goal of the experiments was to show and compare the time series of forces and the states of the state machines controlling the two robots. Future work aims at extending the task-related state machines to include different operating scenarios. Then, we will implement the Situation Awareness module as it has been defined in the architecture proposed in this paper and we will integrate it in the experimental setup. Finally, we will apply the same methodology proposed in this paper to perform other simple surgical tasks, e.g. suturing of planar wounds.

REFERENCES

Baresi, L., Pasquale, L., and Spoletini, P. (2010). Fuzzy Goals for Requirements-Driven Adaptation. In *Proceedings of the International Requirements Engineering Conference*, 125–134.

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

Bonfè, M., Fantuzzi, C., and Secchi, C. (2005). Verification of behavioral substitutability in object-oriented models for industrial controllers. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*. Barcelona, Spain.

Jackson, D. (2012). Alloy. `http://alloy.mit.edu/`.

Klotzbücher, M. and Bruyninckx, H. (2012). Coordinating robotic tasks and systems with rFSM statecharts. *Journal of Software Engineering for Robotics*, 3(1), 28–56.

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

Mayer, H., Nagy, I., Burschka, D., Knoll, A., Braun, E., Lange, R., and Bauernschmitt, R. (2008). Automation of manual tasks for minimally invasive surgery. In *Proceedings of the Fourth International Conference on Autonomic and Autonomous Systems*, 260–265.

McMillan, K. (1993). *Symbolic Model Checking: an Approach to the State Explosion Problem*. Kluwer Academic Publishers.

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

Padoy, N. and Hager, G.D. (2011). 3D thread tracking for robotic assistance in tele-surgery. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2102–2107.

Pappone, C., Vicedomini, G. Manguso, F., Gugliotta, F., Mazzone, P., Gulletta, S., Sora, N., Sala, S., Marzi, A., Augello, G., Livolsi, L., Santagostino, A., and Santinelli, V. (2006). Robotic magnetic navigation for atrial fibrillation ablation. *Journal of the American College of Cardiology*, 47(7).

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

Pnueli, A. (1977). The Temporal Logic of Programs. In *Proceedings of the Annual Symposium on Foundations of Computer Science*, 46–57.

Siciliano, B. and Villani, L. (1999). *Robot Force Control*, volume 540 of *Springer International Series in Engineering and Computer Science*. Springer.

Tobergte, A., Konietschke, R., and Hirzinger, G. (2009). Planning and control of a teleoperation system for research in minimally invasive robotic surgery. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 4225–4232. Kobe, Japan.

Torlak, E. and Dennis, G. (2008). Kodkod for Alloy users. In *Proceedings of the 1st ACM Alloy Workshop*.

Torricelli, M., Ferraguti, F., and Secchi, C. (2013). An algorithm for planning the number and the pose of the iceballs in cryoablation. In *Proc. of the Int. Conf. of the IEEE Engineering in Medicine and Biology Society (EMBC)*. Osaka, Japan.