A Concept for Multi-Phase Incremental Formal Verification in Robotic Guided Surgery

Mattias Ulbrich, Luzie Schreiter, Sarah Grebing,
Jörg Raczkowsky, Heinz Wörn, and Bernhard Beckert

Karlsruhe Institute of Technology,
Karlsruhe, Germany
{ulbrich,luzie.schreiter,sarah.grebing,
raczkowsky,woern,beckert}@kit.edu

Abstract. This work-in-progress paper outlines the concepts of an approach for the formal verification of robotic guided surgery interventions at three different stages of the procedure. The central idea is that complex modelling and verification tasks are performed preclinically during component design yielding simpler safety conditions that can be checked more efficiently shortly before or during the intervention.
A simplified example is presented to illustrate this central idea.

1 Introduction

Robotic guided surgery is a relevant technology in modern health care. Since surgery is a critical application area in which human life is at stake, safety is of utmost importance when designing, configuring and applying robotic guided surgery setups. In this paper we outline how formal methods can be used to enforce safety properties in order to make the application of robotic devices in operation theatres safer.

The main contribution of this paper is a concept for a verification approach for the safety of the entire surgical process from the preclinical phase to the intraoperative phase. To this end, safety properties are checked incrementally at three different stages of the procedure: at component *design time*, at operation theatre *configuration time*, and at *intervention time* (see Sect. 2 for a detailed description). The modelling, specification and verification endeavours concentrates on those safety properties which can be accidentally violated by robot behaviour. The system cannot be completely verified already at design time since information about the intervention setup and procedure is not yet available. Right before and during the actual surgical intervention more detailed data (in particular, the patient's anatomic idiosyncrasies and the workflow of the intervention) is available and concrete safety guarantees can be verified. By doing as much formal analysis as possible already during component design, the time critical verification checks at later phases can be done more efficiently.

To achieve a smooth cooperation between the surgeon and the robotic devices, *surgical workflows* describing the procedure for both robots and staff are

used. The approach gains expressiveness by incorporating the workflows into formal verification during configuration and during the intervention.

The application of formal methods in the area of computer and robot assisted surgery is not new. In [1] model checking of robotic guided workflows by using NuSMV2[1] is introduced. The presented approach covers the robotic guided surgery process only partly; in particular, the dynamics of technical equipment are completely excluded. Muradore et al. [2] employ hybrid automata to model the workflow for a simple tissue puncturing to be performed by an autonomously acting surgical robot. Kouskoulas et al. [3] specified and verified a directional force feedback algorithm which guarantees safety for all possible inputs using KeYmaera [4]. The existing approaches focus on separate aspects of surgical interventions and do not span over more phases of the procedure.

The paper is structured as follows: In Sect. 2 we introduce the concepts of the multi-phase incremental verification, Sect. 3 illustrates the idea with a simplified example. We summarise our contribution and list future work in Sect. 4.

2 Three-phase Verification

We propose a verification approach spanning over three phases which differ in two respects: (1) the available detail data about the surgical setup and the impending operation and (2) the affordable effort that can be spent on verification. Table 1 shows an overview over the three phases.

The Design Phase takes place preclinically, when the robotic device is designed. Robotic components are verified individually with the intended intervention scenarios in mind. A robotic device is modelled as a cyberphysical system, thus going beyond an analysis of its controller software. Continuous values are used to model the physics of the robot's movements. To ensure that such devices obey their safety specifications, they have to be modelled using hybrid modelling languages and verified using hybrid verification techniques.

In this phase it is shown that the component satisfies specified continuous *safety properties* (e.g., that a device's tooltip does not collide with obstacles). To this end, assumptions (called *safety conditions*) are made about the actor stimuli provided by the controller software. If these are obeyed by the controller implementation, the device will be guaranteed to never violate the safety properties. In later phases, safety conditions are checked to hold for the actual implementation of the controllers.

The verification in this phase is the most expensive one which requires an expert to come up with hybrid models (at the right degree of abstraction) and to verify them (possibly interactively). As techniques to be applied here we envision deductive hybrid theorem proving (e.g., with KeYmaera [4]) or hybrid model checking (e.g., with HyperTech [5]).

The Configuration Phase is the time during which the operation theatre is prepared for the upcoming robotic guided surgery. It is only directly prior to the

[1] http://nusmv.fbk.eu/

Table 1: Overview of phases and proposed methodologies

Phase	*Design*	*Configuration*	*Intervention*
Considered Components	robotic device	robotic device, patient	robotic device, patient, med. staff
Proof Obligation	hybrid safety property	discrete safety condition	discrete safety condition
Methodology	hybrid theorem proving / hybrid model checking	discrete model checking	runtime verification / monitoring

actual intervention that detailed data about the patient's health state, anatomic idiosyncrasies, or the exact position of the surgical situs are available. The medical and technical workflows[2] are assembled. With this data, the safety properties can now be substantiated; the verification goal in this phase is to ensure that the setup remains within safe bounds with the given parameterisation.

The requirements on the verification techniques are different from those in the design phase: it must run automatically, user interaction with a verification engine is not an option, and the results must be reached within reasonable time bounds (a matter of minutes to not make verification a bottleneck during configuration). It is therefore a valuable benefit that the verification at design time (which operates on hybrid models) yields *discrete* safety conditions which have to be fulfilled for every decision made by the controller. It is hence sufficient to work with discrete state model checking (instead of more challenging hybrid verification techniques) in this phase which simplifies and extends the reach of this verification task. Ensuring safety conditions at configuration time is important as it reveals potential safety risks before the operation has actually begun.

The Intervention Phase covers the time during which surgery is performed, from the time the patients are sedated up to the moment they are removed from the theatre. All safety conditions cannot be guaranteed before the intervention; there are several reasons why safety checks must be delayed to intervention time: Even though thoughtfully planned, unexpected deviations from the plan (like complications) are always to be considered in surgical interventions. The staff operating the devices may move about independently, this is another factor which requires that safety conditions are closely monitored at runtime.

On the other hand, all facts are not known that early, and a verification/monitoring during the surgery is necessary to guarantee safe operation. Nondeterminisitic unforeseeable control actions, e.g. by telemanipulated devices, unexpected modifications of parameters (no-go areas may move if the patient body moves, system failures) make monitoring necessary.

[2] We assume workflows to be hierarchical behaviour descriptions: A *medical workflow* captures the intended steps of the surgical procedure (including cases of possibly occurring complications), and the behaviour of the robot (the *technical workflow*) is a state machine that depends on the current state in medical workflow.

Runtime monitoring must happen in realtime and, hence, poses a strict limit on the available time for performing verification tasks. A dynamic online verification (with forward-prediction) is targeted to identify deviations from the intended plan of the configuration phase which may violate safety conditions and hence safety properties. It is vital for efficient real-time runtime checks that the originally difficult cyberphysical safety properties have been broken down to easier to check discrete safety conditions in the design phase.

3 Example

To illustrate the concept of the approach, we consider a simplified version of a minimal invasive robotic guided surgery (e.g., an appendectomy). For this purpose, two robotic devices (arms) are used: one moving *freely*, operated in telemanipulation mode by a surgeon and one holding an endoscopic camera. We assume that the latter robot autonomously follows the tooltip of the telemanipulated robot in a specified distance.

In the *design phase*, hybrid models for the robotic devices are constructed. In the example, we model the autonomous camera robot by means of a hybrid model. The examined safety property is that the camera does not come into contact with areas where it might cause damage (e.g., blood vessels, nervous tissue). In the following these areas are called *no-go areas* and are modelled as a set $\{n_1, n_2, \ldots, n_k\} \subset \mathbb{R}^3$ of points (within the patient's body) and a safety distance $s \in \mathbb{R}^+$. The safety property says hence that the (euclidean) distance between the position of the camera $e \in \mathbb{R}^3$ and any n_i is always at least s, i.e., that $\forall i \in \{1, \ldots, k\}.\ \|e - n_i\| \geq s$.

For simplicity we assume that due to slow instrument movement within the body, acceleration effects can be neglected. We will hence model the movement of the camera as a piecewise uniform linear movement with a (piecewise) constant velocity vector $v \in \mathbb{R}^3$.

Moreover, it is modelled that the system has a reaction latency time $T \in \mathbb{R}^+$ pooling all delays between two consecutive decisions taken by the controller program (sensor latency, sensor/actor data processing time, clock cycles, etc.)

The example model of the device is formulated in Differential Dynamic Logic (dDL) [4]. A proof obligation in dDL usually has the shape

$$pre \rightarrow \left[(ctrl\ ;\ dyn)^*\right] safety \tag{1}$$

stating that under assumption of the precondition *pre* the safety property *safety* holds in all reachable states of the hybrid system described by the discrete controller *ctrl* and the continuous dynamics *dyn*.

The instantiations in the schematic proof obligation (1) for the i-th no-go area around n_i are

$$pre := s > 0 \ \wedge\ T > 0 \ \wedge\ \|e - n_i\|^2 \geq s^2 \qquad safety := \|e - n_i\|^2 \geq s^2$$
$$ctrl := \textbf{choose}\ v\ \textbf{such that}\ \Psi(e, n_i, v, T) \qquad dyn := t := 0\ ;\ \{e' = v, t' = 1\ \&\ t \leq T\}\ .$$

The precondition (besides assuming the safety distance and the latency time positive) is that the camera is initially in a safe distance from n_i. This latter

fact is also the safety property to be shown. The discrete component controller software is modelled as a non-deterministic choice of the velocity vector v. This choice is not arbitrary but coupled to a safety condition $\Psi(e, n_i, v, T)$. In the present example, condition

$$\Psi = (p \geq T \rightarrow \|e + T \cdot v - n_i\|^2 > s^2) \ \wedge \ (0 \leq p < T \rightarrow \|e + p \cdot v - n_i\|^2 > s^2)$$

with $p = \frac{\langle n_i - e, v \rangle}{\|v\|^2}$ is a sufficient safety condition guaranteeing that the hybrid system is safe. The fact that for even such a simple model a complex safety condition is required underlines the need of formal verification of such models. The continuous part of the model is described by the differential equation $e' = v$ relating position and velocity. Another continuous variable $t \in \mathbb{R}$ with $t' = 1$ is used to model the elapsing time and the constraint $t \leq T$ ensures that *ctrl* is invoked at least once during the period T. We discharged the hybrid proof obligation (1) in an interactively conducted proof with the dDL theorem prover KeYmaera [4].

The verification result of the design phase is the guarantee that the camera does not intrude into the no-go areas if $\Psi(e, n_i, v, T)$ is satisfied for all $i \in \{1, \ldots, k\}$ for all choice of velocity v that are made.

The *configuration phase* combines the components for the specific surgical intervention with the medical and technical workflow for the intervention. Furthermore, formal parameters have now been fixed, in particular the positions n_i and dimension s of the no-go areas.

The workflow models the process for a specific surgical intervention and is modelled as a non-deterministic finite hierarchical state machine M. In the hierarchy the outermost states model the medical surgery procedure whereas the innermost states encode the behaviour of the robot (the program) w.r.t. the input sensor signals.

The verification obligation in the configuration phase is to show that the safety condition Ψ obtained in the design phase is met for every actuator output signals that the controller implementation comes up with. It is therefore a discrete, classical LTL model checking task to show that $M \models \mathbf{G} \bigwedge_{i=1}^{k} \Psi(e, n_i, v, T)$ in which v is the computed output signal chosen by the controller state machine.

The verification in the *intervention phase* differs from earlier phases in that a detected safety condition violation (or a potential violation in short time) can only raise an alarm and not abort the operation.

In case of the simple example surgery scenario, a runtime monitor can be installed that evaluates $\bigwedge_{i=1}^{k} \Psi(e, n_i, v, T)$ after every invocation of the controller implementation. An alarm can be raised catching the attention of the responsible surgeon as soon it is violated.

In more elaborate models, verification in this phase may not be restricted to runtime expression monitoring but will also integrate monitoring with more advanced formal techniques checking temporal logic constraints for a bounded time span.

4 Conclusion

The presented example shows that the proposed approach for an incremental multi-level formal verification of computer guided surgery is conceptionally feasible.

It remains for future work to extend the presented concept to a more general approach that can deal with various degrees of incrementality and a broader notion of safety condition. As a test-bed for the configuration and intervention phase and for the evaluation of the presented approach the platform OP:Sense, a flexible and modular research platform to perform and monitor robot guided surgeries [6, 7], will be adapted to the proposed verification and monitoring concepts. On this system we will be able to apply the developed verification techniques to emulate telemanipulated and autonomous robot guided surgeries under realistic conditions.

References

1. H. Moennich, J. Raczkowsky, and H. Wörn, "Model checking for robotic guided surgery," in *Electronic Healthcare*, 2010.
2. R. Muradore, D. Bresolin, L. Geretti, P. Fiorini, and T. Villa, "Robotic surgery: Formal verification and plans," *Robotics Automation Magazine, IEEE*, vol. 18, no. 3, pp. 24–32, 2011.
3. 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. on Hybrid Systems: Computation and Control (HSCC 2013)*, ACM, 2013.
4. A. Platzer, *Logical Analysis of Hybrid Systems*. Springer, 2010.
5. T. A. Henzinger, B. Horowitz, R. Majumdar, and H. Wong-toi, "Beyond hytech: Hybrid systems analysis using interval numerical methods," in *in HSCC*, pp. 130–144, Springer, 1999.
6. P. Nicolai, T. Brennecke, M. Kunze, L. Schreiter, T. Beyl, Y. Zhang, J. Mintenbeck, J. Raczkowsky, and H. Wörn, "The op:sense surgical robotics platform: first feasibility studies and current research," in *International Journal of Computer Assisted Radiology and Surgery*, pp. 136–137, 2013.
7. A. Bihlmaier, T. Beyl, P. Nicolai, M. Kunze, J. Mintenbeck, L. Schreiter, T. Brennecke, J. Hutzl, J. Raczkowsky, and H. Wörn, "Ros-based cognitive surgical robotics," p. Forthcoming, 2015.