Microprocessors and Microsystems 39 (2015) 836–842

Contents lists available at ScienceDirect

Microprocessors and Microsystems

journal homepage: www.elsevier.com/locate/micpro

Formal verification of robotic surgery tasks by reachability analysis

Davide Bresolin [a,*], Luca Geretti [b], Riccardo Muradore [c], Paolo Fiorini [c], Tiziano Villa [c]

[a] University of Bologna, Italy
[b] University of Udine, Italy
[c] University of Verona, Italy

ARTICLE INFO

Article history:
Received 7 February 2015
Revised 13 September 2015
Accepted 9 October 2015
Available online 24 October 2015

Keywords:
Formal verification
Hybrid systems
Surgical robotics

ABSTRACT

In this paper we discuss the application of formal methods for the verification of properties of control systems designed for autonomous robotic systems. We illustrate our proposal in the context of surgery by considering the automatic execution of a simple action such as puncturing. To prove that a sequence of subtasks planned on pre-operative data can successfully accomplish the surgical operation despite model uncertainties, we specify the problem by using hybrid automata. We express the requirements of interest as questions about reachability properties of the hybrid automaton model. Then, we use the tool ARIADNE to study how the choice of the control parameters and the measurement error affect the safety of the system.

1. Introduction

In the last decade robotics played a relevant role in the progress of surgery. Even though *robotic surgery* has been a new field of investigation, in a short time the prototypes built in robotics laboratories gained a place in operating rooms. A new terminology witnesses this trend: computer-integrated surgery, medical robotics, rehabilitation robotics, telerobotics, telesurgery, robotic assistive systems, robot-assisted laparoscopic surgery, etc. [1–3]. Robotic surgery has already proven its advantages by improving safety, accuracy, reproducibility, and decreasing patient recovery time and surgeon fatigue [4–7]. This is both a blessing and a curse, in the sense that the advanced functionalities can be useful for surgeons, but at the same time the increased complexity makes the system more susceptible to design errors and poses new challenges to the designers. Advanced design, control, monitoring and deployment paradigms are needed for the next-generation robotic surgery systems.

In engineering practice, the analysis of a complex system is usually carried out via simulation, which allows the designers to explore one of the possible system executions at a time. Formal methods instead aim at exploring all possible executions, in order to ensure proper functionality of the system in all cases, or conversely to acquire information about potential fault cases. In computer science, the name Formal methods identifies a large family of mathematical languages, techniques and tools used to specify and verify systems and to help engineers to develop more reliable systems. Nowadays, they are standard practice in many ICT industries for the development of (discrete) HW/SW systems, and are becoming a vital aspect in the design of safety-critical cyberphysical systems, including robotics and automation systems [8,9]. An area where they can greatly improve the reliability of the design process is Autonomous Robotic Surgery (ARS) [10,11]. The aim of ARS is to perform simple tasks without the presence (or telepresence) of surgeons. Therefore, with ARS, basic tasks will be executed by robots, allowing the surgeons to focus only on the most difficult aspects of the intervention. This implies that the overall control architecture must respect strict safety constraints and must guarantee the successful accomplishment of the surgical tasks, independently of uncertainties and un-modeled subsystems.

In this work we will show how formal verification can provide accurate and reliable answers to help the designer in the development of ARS systems by considering the automatic execution of a simple surgical action such as puncturing. We first model the overall task as a finite sequence of atomic actions that should be accomplished to guarantee the success of the surgical action. This model takes the form of a *hybrid automaton* consisting of a discrete control part that operates in a continuous environment [12]. Then, we specify the safety constraints that the system should respect in a precise mathematical way as reachability properties of the hybrid automaton model. Finally, we use a state-of-the art tool for reachability analysis of hybrid automata (ARIADNE [13]) to find the values of control parameters that guarantee successful accomplishment of the surgical operation, even in presence of measurement errors. This paper focuses on the second step of the usual design process: mathematical modeling → simulation & formal verification → mechanical design → experimental validation, and has a twofold goal: (1) to highlight

* Corresponding author.
E-mail addresses: davide.bresolin@unibo.it (D. Bresolin), luca.geretti@uniud.it (L. Geretti), riccardo.muradore@univr.it (R. Muradore), paolo.fiorini@univr.it (P. Fiorini), tiziano.villa@univr.it (T. Villa).

http://dx.doi.org/10.1016/j.micpro.2015.10.006
0141-9331/

Table 1
Comparison of PHAVer, SpaceEx, HSOLVER and Ariadne.

	PHAVer	SpaceEx	HSOLVER	Ariadne
State space representation	Polyhedra	Support functions	Predicate abstraction	Image sets
Nonlinear dynamics	No	No	**Yes**	**Yes**
Composition of automata	**Yes**	**Yes**	No	**Yes**
Rigorous results	**Yes**	No	**Yes**	**Yes**
User-definable accuracy	**Yes**	**Yes**	No	**Yes**
Graphical output of results	**Yes**	**Yes**	No	**Yes**
Max. no. of variables[a]	~ 10	$\sim$ **100**	~ 10	~ 10

[a] these numbers of variables were reached in some cases reported in [15], [16], [17] and [13].

how a formal verification tool can be used to predict the performance of a robotic system, and (2) to help the designer during the tuning phase of the controller. We leave the study on the subsequent design steps to forthcoming publications.

The paper extends the preliminary results reported in [14] and is organized as follows. In Section 3 we model the surgical task and provide a mathematical model of the robotic manipulator. Section 4 formally defines the properties to be verified, whereas Section 5 describes the verification strategy and the results of the experiments. Some conclusions are drawn in Section 6.

2. Review of formal verification tools

When the system dynamics are simple, the evolution can be computed exactly, and most of the verification techniques for finite-state models can be used to obtain an exact answer to verification problems. When the dynamics is more complex, the reachable set cannot be computed exactly, and different techniques are needed to face the complexity of the verification problem. One of the most successful approaches is to use approximation techniques to under- or over-approximate the evolution of the system.

Among the publicly available state-of-the art tools that use approximation techniques to verify hybrid automata, the most relevant and actively developed are PHAVer [15], SpaceEx [16], HSOLVER [17], and Ariadne [13]. In the following we briefly describe the four tools. We refer the reader to the specific literature for a comprehensive description of their algorithms and state space representation choices. Table 1 summarizes their differences under the following criteria:

- *Class of system they can verify:* do they support nonlinear dynamics? Can the system be specified as a composition of smaller components?
- *Soundness of the results*: is the verification result guaranteed to be mathematically correct?
- *Accuracy control*: is it possible to choose the quality of the approximations?
- *Output:* is it possible to obtain a graphical output of the results, or is only a Yes/No answer provided?
- *Scalability:* what is the maximum size of a system that they can verify?

PHAVer [15] was one of the first tools that enabled verification of hybrid automata with complex dynamics: it handles affine dynamics and guards and supports the composition of hybrid automata. The state space is represented using polytopes. Results are formally sound by means of an exact and robust arithmetic with unlimited precision. Scalability is limited: systems with more than 10 continuous variables are usually out of the capabilities of the tool.

SpaceEx [16] is a modular open-source framework that improves upon PHAVer, with particular regard to scalability (systems with 100 variables have been analyzed with this tool). It combines polyhedra and support functions to represent the state space of systems with piecewise affine, non-deterministic dynamics. Local error bounds on the computation are guaranteed using variable time steps; however, differently from PHAVer, the result of SpaceEx is not guaranteed to be numerically sound. This means that when the tool finds the system safe, we can only conclude that more sophisticated methods are necessary to find bugs for that system.

HSOLVER [17] uses constraint propagation and abstraction-refinement techniques to verify safety properties of nonlinear hybrid systems. In this setting, the hybrid system under verification is first abstracted by a finite-state discrete model that approximates the original one. If the abstraction is not accurate enough to obtain an answer to the verification problem, it is improved using constraint propagation techniques, until either an answer is found or the maximum number of refinement steps is reached. HSOLVER supports systems with complex non-linear dynamics and guards, but not the composition of automata. Because of the particular state-space representation, it cannot provide a graphical output of the reachable set, but only a safe/possibly unsafe answer to the verification problem.

Ariadne [13] uses rigorous numerical methods for working with real numbers, functions and sets in the Euclidean space, to verify hybrid systems with nonlinear dynamics, guards and reset functions. It supports composition to build complex systems from simpler components, and can compute both upper-approximations and lower-approximations of the reachable set. By combining outer and lower approximations, Ariadne can provide both positive and negative answers to safety properties and other more complex verification problems. Its high expressivity, however, affects the performance and scalability of the tool, which is currently limited to systems with 10 continuous variables.

3. Modeling a surgical task

Puncturing is the act of penetrating a biological tissue with a needle, e.g. when performing a biopsy. Together with other elementary tasks such as cutting and suturing, this action can be used to build more complex surgical tasks. To model the puncturing task in a formal way, we divided its execution into three subtasks: (i) a *free motion* phase, where the end effector of the robot approaches the patient's tissue starting from its home position; (ii) a *perpendicular attitude* phase, where the end effector is in contact with the tissue, and the robot moves its wrist to have the tool orthogonal with the patient's surface; (iii) a *puncturing* phase, where the robot increases the force applied by the end effector until the tissue is penetrated.

We assume that the controller for each subtask stabilizes the plant, while the switching between controllers preserves the stability. Our goal is not to prove the stability of the overall system but to prove in a formal way that the *task* itself can be executed correctly. Thus, the focus of the analysis is to show the feasibility of the task rather than the stability of the system. The test case under consideration is a typical example of a *hybrid system*, i.e., a system mixing discrete and continuous behavior that cannot be characterized faithfully using either discrete or continuous models only. A hybrid system consists of a discrete part that operates in a continuous environment, and for this reason it is sensitive not only to time-driven phenomena but also to event-driven phenomena.