— A Journal-First Paper — Formal Specification and Verification of Autonomous Robotic Systems: A Survey

Matt Luckcuck, Marie Farrell, Louise A. Dennis, Clare Dixon, and Michael Fisher

Department of Computer Science, University of Liverpool, UK
{marie.farrell, m.luckcuck}@liverpool.ac.uk

Abstract. Autonomous robotic systems are complex, hybrid, and often safety-critical; this makes their formal specification and verification uniquely challenging. Though commonly used, testing and simulation alone are insufficient to ensure the correctness of, or provide sufficient evidence for the certification of, autonomous robotics. Formal methods for autonomous robotics has received some attention in the literature, but no resource provides a current overview. This journal-first paper provides an overview of a systematic survey of the state-of-the-art in formal specification and verification for autonomous robotics.

1 Introduction and Methodology

This paper summarises our recently published survey of the formal specification and verification techniques that have been applied to autonomous robotic systems[1] [5], which provides a comprehensive overview and analysis of the state-of-the-art, and identifies promising new research directions and challenges for the formal methods community. Previous work, which draws from this survey, advocates the use of integrated formal methods for autonomous robotic systems [2].

We define an *autonomous system* as an artificially intelligent entity that makes decisions in response to input, independent of human interaction. *Robotic systems* are physical entities that interact with the physical world. Thus, an *autonomous robotic system* is a machine that uses Artificial Intelligence (AI), has a physical presence in and interacts with the real world. Autonomous robotics are increasingly used in commonplace-scenarios, such as driverless cars [3], pilotless aircraft [6], and domestic assistants [1].

While for many engineered systems, testing, either by real deployment or via simulation, is deemed sufficient; the unique challenges of autonomous robotics, their dependence on sophisticated software control and decision-making, and their increasing deployment in safety-critical scenarios, require a stronger form of verification. This leads us towards using formal methods to ensure the correctness of, and provide sufficient evidence for the certification of, robotic systems.

[1] Accepted version available at: https://arxiv.org/abs/1807.00048

The corresponding journal paper identifies and investigates the following three research questions:

RQ1: What are the challenges when formally specifying and verifying the behaviour of (autonomous) robotic systems?
RQ2: What are the current formalisms, tools, and approaches used when addressing the answer to **RQ1**?
RQ3: What are the current limitations of the answers to **RQ2** and are there developing solutions aiming to address them?

In order to answer these questions we performed a systematic survey of the literature on *formal modelling of (autonomous) robotic systems*, *formal specification of (autonomous) robotic systems*, and *formal verification of (autonomous) robotic systems*. We restricted our search to papers that were published between 2007 and 2018, inclusive.

In addition to answering the research questions, the survey [5] illustrates opportunities for research applying formal methods (and Integrated Formal Methods (iFM)) to robotics and autonomous systems – either by identifying the popular languages that integration could use, or by showing the gaps that could be filled by iFM. It also provides a brief overview of some popular general software engineering techniques for robotic systems including middleware architectures, testing, and simulation approaches, domain specific languages, graphical notations, and model-driven engineering or XML-based approaches.

2 Answering the Research Questions

This section summarises how the results of our survey address the research questions described in §1.

To answer **RQ1**, we identified the challenges describe in the surveyed literature and categorised them as *external* or *internal* to the robotic system. External challenges come from the design and environment, independently of how the system is designed internally. We saw two major external challenges in the literature: modelling and reasoning about the system's environment, and providing enough evidence for public trust and regulation. Internal challenges stem from how the system is engineered. The three internal challenges that we found in the literature were related to using: agent-based, multi-robot, and adaptive or reconfigurable systems. These challenges, and the tools and techniques used to overcome them, are discussed at length in [5, §3–4].

Tackling internal challenges can have complementary benefits to mitigating external challenges. Reconfigurability is key to safely deploying robots in hazardous environments and vastly more work needs to materialise in order to ensure the safety of reconfigurable autonomous systems. Therefore, we see a clear link between a robotic system reacting to the changes in its external environment, and reconfigurable systems. Similarly, *rational* agent-based systems that can explain their reasoning provide a good route for providing evidence for public trust

or certification bodies. This is because they provide the transparency that is crucial for public trust and certification. A rational agent can provide reasons for its choices, based in the input and internal state information.

RQ2, asked what are the current formal methods used for tackling the challenges identified by answering **RQ1**. To answer this question we quantify and describe the formalisms, tools, and approaches used in the literature [**?**, §5–6]. We found that state-transition systems and logics (particularly temporal logic) are the most often used formalisms to specify the system and properties, respectively [5, Table 2]. We speculate that this is due to the fact that temporal logics and state-transition systems allow abstract specification, which is useful earlier in the development process.

A related finding is that model checkers are the most often used verification approach, which complements the wide use of state-transition systems and temporal logics [5, Tables 3–4]. We speculate that this is because model-checking as an approach is generally easy to explain to stakeholders who do not have experience using formal methods. Notably, theorem provers were used a lot less often, we believe that this is due to the level of expert knowledge required to operate them correctly and efficiently.

RQ3, asked what the limitations are of the best practice formalisms and approaches to verification that were identified in the answer to **RQ2**(see [5, §7]). One obvious limitation appears to be a resistance to adopting formal methods in robotic systems development [4]. The perception is that applying formal methods is a complicated additional step in the engineering process, which prolongs the development process while not adding to the value of the final product. A lack of appropriate tools also often impedes the application of formal methods. There are, however, notable examples of industrial uses of formal methods [7].

We found that there have been a variety of tools developed for the same formalism [5, Table 3]. This suggests that there is a lack of interoperability between different formalisms and tools. Often, models or specifications of similar components are incompatible and locked into a particular tool. Thus, a common framework for translating between, relating, or integrating different formalisms, would prove useful in smoothing the conversion between formalisms or tools. Further, this would serve a growing need to capture the behaviour of complex systems using a heterogeneous set of formalisms and integrated formal methods, each suited to the component being modelled or the properties of interest. This is currently an open problem in formal methods for robotic systems [2].

We note the lack of clear guidance for choosing a suitable formal method for a particular system. To provide some guidance for choosing formal methods for autonomous robotic systems, we describe the formalisms, tools, and the case study tackled for the surveyed literature [5, Table 1]. A more detailed analysis of this area would be useful future work.

Another limitation faced by formal methods for robotic systems (and more generally) is that of formalising the *last link*, the step between a formal model and program code. To guarantee that the program correctly implements the model requires formalised translation process. The lack of clarity about this limitation

points to another: a lack of open sharing of models, code, and realistic case studies that are not tuned for a particular formalism.

Field tests and experiments using simulations are both useful tools for robotic systems development [5, §2]; but formal verification is crucial, especially at the early stages of development when field tests of the control software are infeasible (or dangerous). A focussed research effort on the combination or integration of formal methods should improve their use in robotic systems development, because no single formalism is capable of adequately capturing that all aspects of a robotic system behave as expected. Ensuring that these tools are usable by developers and providing similar features in an IDE would also improve their uptake by simplifying their use. Work in this area could lead to an Integrated *Verification* Environment, allowing the use of different formalisms using same developer front-end, connecting them to their respective tools, and providing helpful IDE-like support.

3 Conclusion

The development of autonomous robotic systems is a novel, emerging, and fast-evolving field. Many of these systems are inherently safety- or mission-critical, so it is prudent that formal methods are used to ensure that they behave as intended. In the spirit of advancing research in this area, our survey provides a description of current formal languages and tools that are being applied to autonomous robotic systems. It also highlights the shortcomings of these approaches and outlines exciting and necessary future directions for the entire formal methods community.

References

1. C. Dixon, M. Webster, J. Saunders, M. Fisher, and K. Dautenhahn. "The fridge door is open" - Temporal verification of a robotic assistant's behaviours. volume 8717 of *LNAI*, pages 97–108. Springer, 2014.
2. M. Farrell, M. Luckcuck, and M. Fisher. Robotics and Integrated Formal Methods: Necessity meets Opportunity. In C. Furia and K. Winter, editors, *Integr. Form. Methods*, volume 11023 of *LNCS*, pages 161–171. Springer, 2018.
3. L. Fernandes, V. Custodio, G. Alves, and M. Fisher. A Rational Agent Controlling an Autonomous Vehicle: Implementation and Formal Verification. *Theor. Comput. Sci.*, 257(Fvav):35–42, 2017.
4. Y. Lopes, S. Trenkwalder, A. Leal, T. Dodd, and R. Groß. Supervisory control theory applied to swarm robotics. *Swarm Intell.*, 10(1):65–97, 2016.
5. M. Luckcuck, M. Farrell, L. Dennis, C. Dixon, and M. Fisher. Formal Specification and Verification of Autonomous Robotic Systems: A Survey. *ACM Comput. Surv. (Accepted)*.
6. M. Webster, M. Fisher, N. Cameron, and M. Jump. Formal Methods for the Certification of Autonomous Unmanned Aircraft Systems. volume 6894 of *LNCS*, pages 228–242. Springer, 2011.
7. J. Woodcock, P. G. Larsen, J. Bicarregui, and J. Fitzgerald. Formal methods: Practice and Experience. *ACM Comput. Surv.*, 41(4):1–36, 2009.