A Summary of Formal Specification and Verification of Autonomous Robotic Systems*

Matt Luckcuck[0000−0002−6444−9312], Marie Farrell[0000−0001−7708−3877], Louise A. Dennis[0000−0003−1426−1896], Clare Dixon[0000−0002−4610−9533], and Michael Fisher[0000−0002−0875−3862]

Department of Computer Science, University of Liverpool, UK
{m.luckcuck, marie.farrell}@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 have received some attention in the literature, but no resource provides a current overview. This short paper summarises the contributions published in [5], which surveys the state-of-the-art in formal specification and verification for autonomous robotics.

1 Introduction and Methodology

This short paper summarises our recently published survey of the formal specification and verification techniques that have been applied to autonomous robotic systems [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].

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

* Work supported by UKRI Hubs for Robotics and AI in Hazardous Environments: EP/R026092 (FAIR-SPACE), EP/R026173 (ORCA), and EP/R026084 (RAIN).

The corresponding journal paper [5] 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?

To answer these questions we performed a systematic literature survey 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 published from 2007 to 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 found 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 much 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 on input and internal state information.

RQ2, asked what are the current formal methods used for tackling the challenges identified by answering **RQ1**. Thus, we quantify and describe the formalisms, tools, and approaches used in the literature [5, §5–6], summarised in Table 1 (right). We found that state-transition systems and (temporal) logics are the most used formalisms to specify the system and properties, respectively; which may be because they allow abstract specification, which is useful early in development.

Formalism	System	Property
Set-Based	5	0
State-Transition	33	0
Logics	6	32
Process Algebra	3	1
Ontology	4	0
Other	5	8

Table 1. Summary of the types of formalisms for specifying the system and the properties to be checked, summarising [5, Table 2].

We found 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]. This may be because the model-checking approach is generally easy to explain to stakeholders with no experience of 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 are the limitations of the 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 development 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].

There have been a variety of tools developed for the same formalism [5, Table 3] signaling a lack of interoperability between different formalisms and tools. Often, models or specifications of similar components are incompatible and locked into a single tool. Thus, a common framework for translating between, relating, or integrating different formalisms, would help smooth the conversion between formalisms/tools. This would also serve a growing need to capture the behaviour of complex systems using a heterogeneous set of formalisms and integrated formal methods. This is an open problem in formal methods for robotic systems [2].

Another limitation faced in this domain is in formalising the *last link*, the step between a formal model and program code. Ensuring that the program correctly implements the model requires a formalised translation. 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 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 combining or integrating formal methods should improve their use in robotic systems development, because no single formalism

is capable of adequately checking 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 quickly-changing 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. To advance research in this area, our survey provides a description of the formalisms and tools that are current being applied to autonomous robotic systems. To provide some guidance for choosing these languages and tools, we describe them and the case study tackled for the surveyed literature [5, Table 1], however a detailed analysis of this is left as future work. The survey also highlights the shortcomings of these approaches and outlines exciting and necessary future directions for the entire formal methods community.

References

1. Dixon, C., Webster, M., Saunders, J., Fisher, M., Dautenhahn, K.: "The fridge door is open" - Temporal verification of a robotic assistant's behaviours. In: Towar. Auton. Robot. Syst. LNAI, vol. 8717, pp. 97–108 (2014). https://doi.org/10.1007/978-3-319-10401-0_9
2. Farrell, M., Luckcuck, M., Fisher, M.: Robotics and Integrated Formal Methods: Necessity meets Opportunity. In: Integr. Form. Methods. LNCS, vol. 11023, pp. 161–171. Springer (2018). https://doi.org/10.1007/978-3-319-98938-9_10
3. Fernandes, L.E.R., Custodio, V., Alves, G.V., Fisher, M.: A Rational Agent Controlling an Autonomous Vehicle: Implementation and Formal Verification. Electron. Proc. Theor. Comput. Sci. **257**(Fvav), 35–42 (2017). https://doi.org/10.4204/EPTCS.257.5
4. Lopes, Y., Trenkwalder, S., Leal, A., Dodd, T., Groß, R.: Supervisory control theory applied to swarm robotics. Swarm Intell. **10**(1), 65–97 (2016). https://doi.org/10.1007/s11721-016-0119-0
5. Luckcuck, M., Farrell, M., Dennis, L.A., Dixon, C., Fisher, M.: Formal Specification and Verification of Autonomous Robotic Systems. ACM Comput. Surv. **52**(5), 1–41 (sep 2019). https://doi.org/10.1145/3342355
6. Webster, M., Fisher, M., Cameron, N., Jump, M.: Formal Methods for the Certification of Autonomous Unmanned Aircraft Systems. In: Comput. Safety, Reliab. Secur. LNCS, vol. 6894, pp. 228–242. Springer (2011). https://doi.org/10.1007/978-3-642-24270-0_17
7. Woodcock, J., Larsen, P.G., Bicarregui, J., Fitzgerald, J.: Formal methods: Practice and Experience. ACM Comput. Surv. **41**(4), 1–36 (2009). https://doi.org/10.1145/1592434.1592436