Increasing Confidence in Autonomous Systems

Michael Fisher
University of Manchester
Manchester, UK
michael.fisher@manchester.ac.uk

Angelo Ferrando
University of Genova
Genova, Italy
angelo.ferrando@unige.it

Rafael C. Cardoso
University of Manchester
Manchester, UK
rafael.cardoso@manchester.ac.uk

ABSTRACT

This presentation will describe how we are using, and aiming to use, runtime verification, along with other varieties of formal verification and simulation-based testing, to together provide increased confidence in a range of autonomous systems.

CCS CONCEPTS

• **Software and its engineering → Formal software verification**.

KEYWORDS

Runtime verification; Autonomous systems

ACM Reference Format:
Michael Fisher, Angelo Ferrando, and Rafael C. Cardoso. 2021. Increasing Confidence in Autonomous Systems. In *Proceedings of the 5th ACM International Workshop on Verification and mOnitoring at Runtime EXecution (VORTEX '21), July 12, 2021, Virtual, Denmark.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3464974.3468452

1 BACKGROUND

Autonomous systems, such as "driver-less cars", unmanned air vehicles ('drones'), domestic robots, etc, are popular in the media but much rarer in practice. The key aspect of autonomy is the possibility, and often the need, that the system will make its own decisions and take its own actions without human intervention. Many of the systems we see today do not have these levels of autonomy, being either controlled by a human (e.g. a driver) or being deployed in environments that are so constrained that we can predict all the possibilities (and so map out all optimal decisions) beforehand. However, there will eventually be truly autonomous systems that work in non-trivial (and unpredictable) environments.

But what is holding this development back? The hardware for robotic systems and vehicles exists. The software for autonomous decision-making, either through automatic control or autonomous agents, has been available for many years. However, the main barriers appear to be around regulators, the public, and even the organisations that must develop these systems. There is a lack of trust, and even suspicion, about truly autonomous systems. Humans can be trusted, but can we really rely on software to make key, and sometimes, critical decisions? This barrier to true autonomy occurs across sectors and explains why human operators/pilots/drivers are still responsible for many aspects of so-called 'autonomous' systems.

So, do we need to go beyond this state-of-affairs, especially as many applications work perfectly well with close human oversight? But there are increasingly many areas where 'autonomy' would be beneficial, as described below.

Hazards. In hazardous environments, such as at nuclear sites or in harsh offshore deployments, then we clearly want robotic solutions that avoid the need for humans to be present. This obviously improves human safety. However, if we do not want the robotic system to be fully autonomous then we are left with remote-control, or remote piloting, of a robotics system at a distance. Even if we do not need the robotic device to manipulate/intervene in its environment, then this remote operation is fraught with problems. If you have ever tried accurate remote control, especially where you cannot see the target or the environment directly then you will know how difficult this can be. This is often very inaccurate and inefficient and delegating increasing autonomy to the robotic device might well improve the situation [14].

Delay. There are a number situations in which waiting for a human to make a key decision will take much too long. One example is in the hazardous environments described above where critical problems can occur quickly and so solutions/decisions must also be reached quickly. A more compelling example, however, concerns activity in space. As we move beyond Earth orbits there is an unavoidable delay in communications, both in transferring images back to Earth and then transmitting decisions to the relevant robot or vehicle. In very static situations this will only lead to inefficiency. However, if robots or vehicles are to be deployed on Mars and beyond, the significant delays mean that critical decisions cannot be made in a timely fashion. The robot or vehicle will have already become stuck in soft sand, run out of battery, or lost its communications link. Autonomous operation clearly becomes crucial in these distant environments.

Mundane. A final class concerns situations in which we *could* directly control a robot or vehicle, but either it is too mundane for us to continually do this, or too expensive. Imagine that everyone had a domestic robot in their home. The human resource required to monitor and make timely decisions about all these will be huge (even when multiple robots are monitored). Similarly with "driver-less" cars; drivers expect that eventually they will be able to completely delegate control and, for example, sleep in their seat. If the vehicles are not going to be fully autonomous, who will monitor and decide? In particular, once we have millions of such vehicles.

2 WAYS FORWARD

There are a number of initiatives aimed at improving this situation, all of which will require stronger and more comprehensive *Verification and Validation* (V&V). Let us highlight several of these.

VORTEX '21, July 12, 2021, Virtual, Denmark

ACM ISBN 978-1-4503-8546-6/21/07.
https://doi.org/10.1145/3464974.3468452

Standards & Regulation. Both national and international standards bodies are beginning to look at autonomous systems. Particularly important are the P70** series of standards under the umbrella of the *IEEE Global Initiative for Ethical Considerations in the Design of Autonomous Systems*[1], for example P7001 on *Transparency* [16] and P7009 on *Failsafe Design* [17]. These, and other standards, lean heavily on comprehensive V&V to provide evidence and confidence and so regulators, across sectors, are increasing looking at these for routes to certification [15].

Transparency & Trust. A key issue with *truly* autonomous systems is 'trust'. In order for us to trust such systems we must not only have confidence that they are reliable but also that they are working, especially when making their own decisions, for our benefit [4]. A key part of this is that these systems must be transparent, not only concerning what they decide, but *why* they decided on an action [20]. Again, standards on transparency, such as P7001, are important but the *truthful* exposure of reasons for decisions is crucial and will again require V&V to ensure.

Systems Architectures & Verification. How we put together elements within the system, and how we can verify distinct elements to give a holistic verification, will be crucial to these types of systems [22]. Fortunately, most practical autonomous systems are modular; for example in robotics the modular Robotic Operating System (ROS) [25] is predominant. We are then able to construct architectures with heterogeneous components. In particular, we combine symbolic agents, for high-level decision-making, with standard control/feedback components, for low-level interactions [5, 7, 21]. Then, verification for the whole system requires bringing together potentially different verification methods for the distinct components [3, 8]. This requires a compositional approach, with the one we primarily use being based on first-order temporal representations [1]. A further important aspect is that verification of any individual component is often not dependent on a specific verification technique; we utilise *corroborative* verification [27] with multiple different verification techniques being used on the same system component, which provides increased confidence. For example, in verifying UAV behaviour we may use simulation-based testing to examine and "stress-test" the formal verification that has been carried out [26].

3 ROLE OF RUNTIME VERIFICATION?

Essentially, the way we use (or plan to use, in some cases) runtime verification is to *recognise situations we do not expect*. This covers a range of possibilities, in terms of component behaviour, safety, user models, security, verification boundaries, predictions, etc. In most cases, runtime monitors are easy to generate — this is obviously the case where we have carried out some form of modelling or formal verification where the monitor is essentially a by-product of the modelling/verification. In other cases, the monitors may be derived from the predictive elements used, such as prognostic models. So, to give a flavour of where we have, and might, use runtime verification to help with verification and with increasing confidence, we provide a range of examples below. Predominantly, we use our ROSMonitoring tool [10] to create monitors that can observe events generated between ROS components and verify properties at runtime.

Certification/Assurance. Practical systems have to be certified or approved by appropriate regulators. As part of this a "Safety Case", or some other form of analysis and evidence, must be provided. And as part of the Safety Case, some assumptions are made. So long as those assumptions remain true, the system is (legally) certified to run. However, if these assumptions no longer hold then this certification fails. This does not necessarily mean the system is unsafe, just that the conditions under which approval was given no longer hold [13].

Safety. Practical safety for autonomous systems is covered through adherence to a range of safety standards that developers must assess their system against. For example, [18, 24] describes the proximity and speed limits for robotic devices near humans. While developers must assess these requirements within their Safety Case, their assessments are only estimates and so there remains a risk of robots actually coming too close to humans. So, an obvious route is to add runtime monitors to ensure that the proximity/speed requirements are maintained in practice. We have done this as part of the ARIAC automated manufacturing competition [11].

Security. There are a range of security threats that our autonomous systems might face and so another obvious route is to add runtime monitors to recognise either the patterns leading up to attacks or the effect of attacks. However, there are so many potential threats that we cannot realistically add monitors for all of these. So, our approach here is to use general security analysis to highlight the most likely, or most significant, threats and aim to recognise these through runtime verification [23].

Verification Violations. Runtime verification naturally fits well with static formal verification techniques. Typically, we will carry out some form of model checking [7] before deployment, but will have to make some abstract assumptions about the environment. Under these assumptions we can guarantee correctness. Based on this verification we can then produce runtime monitors to recognise when the assumptions have been 'violated' [9]. If such a situation is recognised then the agent in control of the system knows that the expected behaviour is no longer guaranteed. The role of runtime monitors in checking for assumption violations (for formal verification) is then quite natural and appropriate [12].

Failure. As described above, our autonomous systems comprise various modules related to the distinct aspects of the system: agent, planners, manipulation, movement, object recognition, etc. In sophisticated autonomous systems, the high-level agent might have representation and awareness of the expected behaviour of each of these components [5]. Runtime verification can usefully be used to monitor whether the behaviour of various components match expectations. For example, we might add a monitor to recognise whether the movement module actually achieves its target destination or not. In case of failure, the agent might choose to use other movement options or simply adapt its description of the module's capabilities [2, 6].

Predictions. Monitors are not only used to report violations at the moment they occur, but they can also be used to predict unexpected behaviour according to some previous knowledge of the system. In such scenarios we talk about predictive runtime verification [28]. This becomes relevant when detecting a failure occurs too late. For

[1] standards.ieee.org/develop/indconn/ec/autonomous_systems.html

instance, in the context of UAV battery consumption, we might add a monitor to detect when the battery is not going to be enough for the UAV to conclude its mission and get back to base. In such scenarios, if the monitor considered only the current battery consumption, it would not be capable of warning the system in time to react, and the UAV would risk running out of battery. Hence, we need the monitor to predict a failure before the latter occurs. This can be done by performing a prognostic analysis on the battery [29], and using the resulting model to predict future events at runtime for the monitor.

User Models. Typically, complex autonomous systems incorporate models of the users they deal with. Often these are of the "Theory of Mind" kind [19]. Behaviour of the system, when interacting with humans, is optimised with respect to the modelled behaviour/expectations of the human. However, it will be important to recognise when human behaviour regularly slips beyond the expected envelope [13], which might then lead to theory revision or reduced interaction.

4 SUMMARY

The demand for the use of autonomous systems has been increasing, especially in applications domains that contain hazardous environments, require timely decisions/reactions, or are simply too expensive or too mundane to operationalise. With this increase in demand, autonomous systems are also facing higher scrutiny in terms of trust that the autonomous behaviour works as intended and does not violate any safety constraints.

Thus, it is important to provide assurances that can increase the confidence we (regulators, public, developers, stakeholders, etc) have about autonomous systems. Recent autonomous systems (e.g., robots) have a high degree of modularity due to the assortment of different components that interact with each other. Some of the approaches we have been developing use a variety of V&V techniques (such as runtime verification, model checking, theorem proving, simulation-based testing, etc.) that can be combined to provide increased confidence in a range of autonomous systems.

ACKNOWLEDGMENTS

The work described here has been supported through UK research funding from the *Royal Academy of Engineering*, under the Chairs in Emerging Technologies scheme, and the *UK Research and Innovation's* "Robots for a Safer World" programme (EP/R026092, EP/R026084, EP/R026173) and Trustworthy Autonomous Systems *Verifiability Node* (EP/V026801).

REFERENCES

[1] R. C. Cardoso, L. A. Dennis, M. Farrell, M. Fisher, and M. Luckcuck. Towards Compositional Verification for Modular Robotic Systems. In *Proc. Second Workshop on Formal Methods for Autonomous Systems, Virtual, December 2020*, volume 329 of *Electronic Proceedings in Theoretical Computer Science*, pages 15–22. Open Publishing Association, 2020. doi: 10.4204/EPTCS.329.2.

[2] R. C. Cardoso, L. A. Dennis, and M. Fisher. Plan Library Reconfigurability in BDI Agents. In *Engineering Multi-Agent Systems*, pages 195–212, Cham, 2020. Springer. doi: 10.1007/978-3-030-51417-4_10.

[3] R. C. Cardoso, M. Farrell, M. Luckcuck, A. Ferrando, and M. Fisher. Heterogeneous Verification of an Autonomous Curiosity Rover. In *Proc. NASA Formal Methods*, pages 353–360, Cham, 2020. Springer. doi: 10.1007/978-3-030-55754-6_20.

[4] R. Chatila, V. Dignum, M. Fisher, F. Giannotti, K. Morik, S. Russell, and K. Yeung. Trustworthy AI. In *Reflections on Artificial Intelligence for Humanity*, pages 13–39. Springer, 2021. doi: 10.1007/978-3-030-69128-8_2.

[5] L. A. Dennis and M. Fisher. Verifiable Self-Aware Agent-Based Autonomous Systems. *Proceedings of the IEEE*, 108(7):1011–1026, 2020.

[6] L. A. Dennis, M. Fisher, J. M. Aitken, S. M. Veres, Y. Gao, A. Shaukat, and G. Burroughes. Reconfigurable Autonomy. *KI - Künstliche Intelligenz*, 28(3):199–207, 2014. doi: 10.1007/s13218-014-0308-1.

[7] L. A. Dennis, M. Fisher, N. K. Lincoln, A. Lisitsa, and S. M. Veres. Practical Verification of Decision-Making in Agent-Based Autonomous Systems. *Automated Software Engineering*, 23(3):305–359, 2016. doi: 10.1007/s10515-014-0168-9.

[8] M. Farrell, M. Luckcuck, and M. Fisher. Robotics and Integrated Formal Methods: Necessity Meets Opportunity. In *Proc. 14th Int. Conf. Integrated Formal Methods (iFM)*, volume 11023 of *LNCS*, pages 161–171. Springer, 2018. doi: 10.1007/978-3-319-98938-9_10.

[9] A. Ferrando, L. A. Dennis, D. Ancona, Michael, and V. Mascardi. Recognising Assumption Violations in Autonomous Systems Verification. In *Proc. 17th Int. Conf. Autonomous Agents and MultiAgent Systems*, pages 1933–1935. ACM, 2018.

[10] A. Ferrando, R. C. Cardoso, M. Fisher, D. Ancona, L. Franceschini, and V. Mascardi. ROSMonitoring: A Runtime Verification Framework for ROS. In *Towards Autonomous Robotic Systems*, pages 387–399, Cham, 2020. Springer. doi: 10.1007/978-3-030-63486-5_40.

[11] A. Ferrando, Z. Kootbally, P. Piliptchak, R. C. Cardoso, C. Schlenoff, and M. Fisher. Runtime Verification of the ARIAC competition: Can a robot be agile and safe at the same time? In *Proceedings of the 7th Italian Workshop on Artificial Intelligence and Robotics co-located with the 19th Int. Conf. Italian Association for Artificial Intelligence (AIxIA 2020), Anywhere, November, 2020*, volume 2806 of *CEUR Workshop Proceedings*, pages 7–11. CEUR-WS.org, 2020. URL http://ceur-ws.org/Vol-2806/short2.pdf.

[12] A. Ferrando, L. A. Dennis, R. C. Cardoso, M. Fisher, D. Ancona, and V. Mascardi. Toward a Holistic Approach to Verification and Validation of Autonomous Cognitive Systems. *ACM Trans. Software Engineering and Methodology*, 30(4), 2021. doi: 10.1145/3447246.

[13] M. Fisher, E. Collins, L. A. Dennis, M. Luckcuck, M. Webster, M. Jump, V. Pagé, C. Patchett, F. Dinmohammadi, D. Flynn, V. Robu, and X. Zhao. Verifiable Self-Certifying Autonomous Systems. In *Proc. Int. Symp. Software Reliability Eng. Workshops*, pages 341–348. IEEE, 2018. doi: 10.1109/ISSREW.2018.00028.

[14] M. Fisher, R. C. Cardoso, E. C. Collins, C. Dadswell, L. A. Dennis, C. Dixon, M. Farrell, A. Ferrando, X. Huang, M. Jump, G. Kourtis, A. Lisitsa, M. Luckcuck, S. Luo, V. Page, F. Papacchini, and M. Webster. An Overview of Verification and Validation Challenges for Inspection Robots. *Robotics*, 10(2), 2021. doi: 10.3390/robotics10020067.

[15] M. Fisher, V. Mascardi, K. Y. Rozier, B. Schlingloff, M. Winikoff, and N. Yorke-Smith. Towards a Framework for Certification of Reliable Autonomous Systems. *Autonomous Agents and Multi Agent Systems*, 35(1):8, 2021. doi: 10.1007/s10458-020-09487-2.

[16] Institute of Electrical and Electronics Engineers. P7001 – Transparency of Autonomous Systems, 2021. URL https://standards.ieee.org/project/7001.html.

[17] Institute of Electrical and Electronics Engineers. P7009 – Fail-safe Design of Autonomous Systems, 2021. https://standards.ieee.org/project/7009.html.

[18] International Organization for Standardization (ISO). ISO 10218 – Robots and Robotic Devices – Safety Requirements for Industrial Robots, 2011. https://www.iso.org/standard/51330.html.

[19] P. N. Johnson-Laird and P. C. Wason, editors. *Thinking — Readings in Cognitive Science.* Cambridge University Press, 1983.

[20] V. Koeman, L. A. Dennis, M. Webster, M. Fisher, and K. Hindriks. The"Why did you do that?" Button: Answering Why-questions for end users of Robotic Systems. In *Proc. of the 7th Int. Workshop on Engineering Multi-Agent Systems (EMAS)*, 2019.

[21] N. Lincoln, S. M. Veres, L. A. Dennis, M. Fisher, and A. Lisitsa. An Agent Based Framework for Adaptive Control and Decision Making of Autonomous Vehicles. In *Proc. IFAC Workshop on Adaptation and Learning in Control and Signal Processing (ALCOSP)*, 2010.

[22] M. Luckcuck, M. Farrell, L. A. Dennis, C. Dixon, and M. Fisher. Formal Specification and Verification of Autonomous Robotic Systems: A survey. *ACM Computing Surveys*, 52(5):100:1–100:41, 2019. doi: 10.1145/3342355.

[23] C. Maple, M. Bradbury, H. Yuan, M. Farrell, C. Dixon, M. Fisher, and U. I. Atmaca. Security-Minded Verification of Space Systems. In *Proc. IEEE Aerospace Conference*, pages 1–13, 2020. doi: 10.1109/AERO47225.2020.9172563.

[24] J. A. Marvel. Performance Metrics of Speed and Separation Monitoring in Shared Workspaces. *IEEE Trans. Automation Science and Engineering*, 10(2):405–414, 2013. doi: 10.1109/TASE.2013.2237904.

[25] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng. ROS: An Open-source Robot Operating System. In *Proc. ICRA Workshop on Open Source Software*, 2009.

[26] M. Webster, N. Cameron, M. Fisher, and M. Jump. Generating Certification Evidence for Autonomous Unmanned Aircraft Using Model Checking and Simulation. *Journal of Aerospace Information Systems*, 11(5):258–279, 2014. doi: 10.2514/1.I010096

[27] M. Webster, D. Western, D. Araiza-Illan, C. Dixon, K. Eder, M. Fisher, and A. G. Pipe. A Corroborative Approach to Verification and Validation of Human–Robot Teams. *Int. J. Robotics Research*, 39(1), 2020. doi: 10.1177/0278364919883338.

[28] X. Zhang, M. Leucker, and W. Dong. Runtime Verification with Predictive Semantics. In *Proc. 4th Int. Symp. NASA Formal Methods*, volume 7226 of *LNCS*, pages 418–432. Springer, 2012. doi: 10.1007/978-3-642-28891-3_37.

[29] X. Zhao, M. Osborne, J. Lantair, V. Robu, D. Flynn, X. Huang, M. Fisher, F. Papacchini, and A. Ferrando. Towards Integrating Formal Verification of Autonomous Robots with Battery Prognostics and Health Management. In *Proc. 17th Int. Conf. Software Engineering and Formal Methods (SEFM)*, volume 11724 of *LNCS*, pages 105–124. Springer, 2019. doi: 10.1007/978-3-030-30446-1_6.