Trustworthy Autonomous Systems

Andrew Sogokon, Neeraj Suri (PI) EP/V026763/1

School of Computing and Communications, Lancaster University

Engineering and Physical Sciences Research Council

Introduction

Many modern examples of autonomous systems (such as autonomous cars and aerial vehicles) are examples of cyber-physical systems with multiple interacting components.

Increasingly, components employing machine learning (ML) are being deployed. For example, neural networks are currently being used in the design of both controllers and *perception modules*.

These developments present serious challenges because *trustworthy* autonomy demands strong guarantees about the behaviour of the overall autonomous system, which are difficult to obtain when neural networks are involved.

Properties of autonomous systems

Some common requirements for autonomous systems feature properties that fall under the definition of *safety* (e.g that vehicles will not collide with obstacles) and *liveness* (e.g. that a vehicle must attain a certain state), or a combination thereof (e.g. that a vehicle must progress towards its goal while avoiding all obstacles: so-called *reach-avoid* properties).

Safety and liveness requirements can be specified precisely and formally using a *temporal logic*.

It is a practical challenge to *verify* properties of autonomous systems composed of many components when the specification for some of the components is not known precisely.

Security properties

Security properties are often stated in terms of safety and liveness properties.

More recent work by Clarkson and Schneider introduced so-called *hyperproperties*, along with new logics which can capture a wider class of security properties (e.g. secure information flow) than is possible using standard safety and liveness properties.

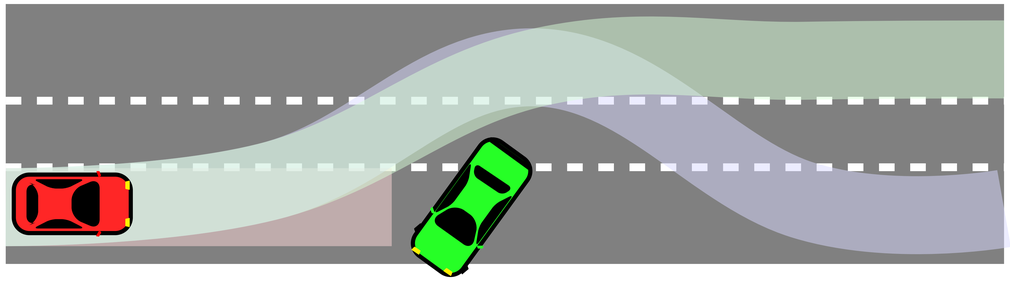

Figure: Desirable property of autonomous vehicles: collision freedom.

Formal verification

Formal verification provides a mathematically rigorous way of proving properties about autonomous systems and gives the strongest possible guarantees about their behaviour. However, in practice, applying formal verification in this domain is fraught with many difficulties.

One of the challenges is that performing verification as part of a system's design relies on making assumptions about its operating environment (which in practice is often uncertain).

Runtime monitoring

As an alternative to performing verification offline during the design, one approach is to monitor the behaviour of the system as it is running and determine if it is about to violate any of its requirements *on the fly* (and switching into a different mode in order to avoid this). The so-called Simplex architecture in principle enables safe use of unverified controllers. In order to make the Simplex approach trustworthy, one requires strong (formal) guarantees about the behaviour of the decision module which governs the switching.

Runtime monitoring and verification (the Simplex approach)

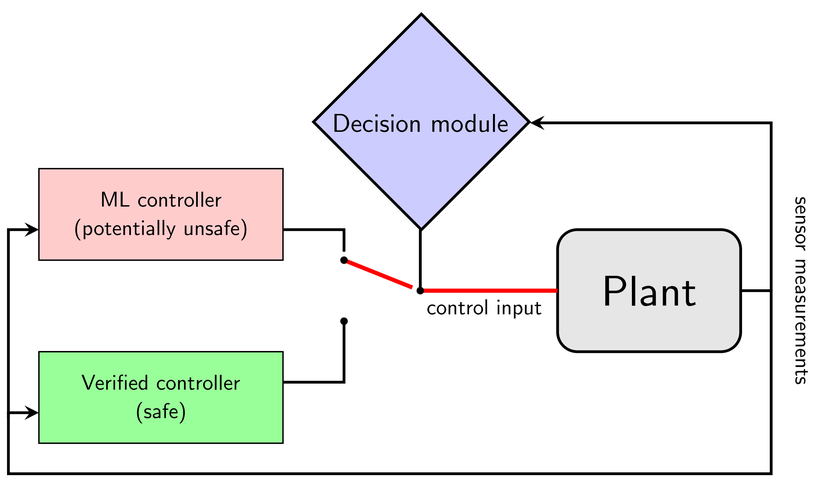

Figure: Simplex Architecture.

Stability

Stability of motion (which is an example of a *hyperproperty*) is often a critically important requirement in cyber-physical systems. If the system becomes unstable, e.g. due to failure or malfunction of some of its components (which may happen as a result of an adversarial attack), the behaviour of the overall system can become dangerous.

A formal (sufficient) criterion for establishing stability properties in control involves the use of so-called *Lyapunov functions*.

The classic sufficient criterion for stability was generalised by R. Bellman in 1962 in what he termed *vector Lyapunov functions*. The biggest practical bottleneck lies in finding the Lyapunov function (whether vector or scalar) for a given system.

Recent methods employing neural networks in order to search for Lyapunov functions could in principle be applied to search for Bellman's extended class of vector functions.

References

[1] Danbing Seto and et al. Krogh.
The Simplex architecture for safe online control system upgrades.
In *ACC'98*, volume 6, pages 3504–3508, 1998.

[2] Richard Bellman.
Vector Lyapunov functions.
Journal of the SIAM, Series A: Control, 1(1):32–34, 1962.

[3] Michael R Clarkson and Fred B Schneider.
Hyperproperties.
J. Comp. Sec., 18(6):1157–1210, 2010.

Lancaster University

Acknowledgements

This work is supported, in part, by the Engineering and Physical Sciences Research Council (grant number: **EP/V026763/1**).