Proceedings of the
2nd Workshop on Artificial Intelligence and Formal Verification, Logics, Automata and Synthesis (OVERLAY),
September 25, 2020

Adversarial Learning of Robust and Safe Controllers for Cyber-Physical Systems

Luca Bortolussi[1,2], Francesca Cairoli[1], Ginevra Carbone[1], and Francesco Franchina[1]

[1]Department of Mathematics and Geoscience, University of Trieste, Italy
[2]Modelling and Simulation Group, Saarland University, Germany

Abstract

We introduce a novel learning-based approach to synthesize safe and robust controllers for autonomous Cyber-Physical Systems and, at the same time, to generate challenging tests. This procedure combines formal methods for model verification with Generative Adversarial Networks. The method learns two Neural Networks: the first one aims at generating troubling scenarios for the controller, while the second one aims at enforcing the safety constraints. We test the proposed method on a variety of case studies.

1 Introduction

Controlling Cyber-Physical Systems (CPS) is a well-established problem in classic control theory [6]. State of the art solutions apply to all those models in which a complete knowledge of the system is available, i.e., scenarios in which the environment is supposed to follow deterministic rules. For such models a high level of predictability, along with good robustness, is achieved. However, as soon as these unpredictable scenarios come into play, traditional controllers are challenged and could fail. Ongoing research is trying to guarantee more flexibility and resilience in this context by using Deep Learning [9] and, in particular, Reinforcement Learning for robust control [1]. State of the art solutions perform reasonably well, but they still present evident limits in case of unexpected situations. The so called *open world scenarios* are difficult to model and to control, due to the significant amount of stochastic variables that are needed in their modelling and to the variety of uncertain scenarios that they present. Therefore, while trying to ensure safety and robustness, we need to be cautious about not trading them with model effectiveness.

In this work we investigate autonomous learning of safe and robust controllers in open world scenarios. Our approach consists in training two neural networks, inspired by Generative Adversarial Networks (GAN) [4], that have opposite goals: the *attacker* network tries to generate troubling scenarios for the *defender*, which in turn tries to learn how to face them without violating some safety constraints. The outcome of this training procedure is twofold: on one side we get a robust controller, whereas on the other we get a generator of adverse tests.

The learned controller is a black-box device that should be able to deal with adverse or unobserved scenarios, but that does not provide worst-case guarantees. In this regard one could additionally rely on a shield-based approach, as proposed in [2].

2 Problem Statement

Safety of a system is guaranteed by the satisfaction of a set of requirements. Checking the satisfiability of properties in hybrid systems, where both discrete and continuous components are involved, is often too computationally complex or undecidable [10]; this especially holds true in the presence of stochastic components. A popular approach to mathematically express safety requirements is by means of *Signal Temporal Logic* (STL) [8]. Temporal logic is a logical formalism used in the context of *formal verification* to formalize the behaviour in time of systems. It extends propositional logic with a set of modal operators capturing the temporal properties of events [5]. STL, in particular, deals with properties of continuous-time signals [8] (i.e. multivariate time series), featuring time-bounded since and until modal operators. In our application, we rely on STL *quantitative semantics*, which returns a real valued measure of satisfiability capturing how much the input signal can be shifted without changing the truth value. Such measure is often referred to as *robustness* and is exploited in this work as the objective function of an optimization problem.

We model the interaction of an agent with an adversarial environment as a *zero-sum game*, similarly to the strategy behind GANs [8]. The concept of zero-sum game is borrowed from game theory and denotes those situations in which one player's gain is equivalent to another's loss. In such situations, the best strategy for each player is to minimize its loss, while assuming that the opponent is playing at its best. This concept is known in literature as *minmax strategy.* In practice, we use GAN architectural and theoretical design to reach two main objectives: a controller, that safely acts under adverse conditions, and an attacker, which gains insights about troubling scenarios for the opponent.

Agent-Environment Model. Due to coexistence of continuous and discrete components, CPSs are typically represented as hybrid models: the continuous part is represented by differential equations that describe the behaviour of the plant; the discrete part, instead, identifies the possible states of the controller. We decompose our model in two interacting parts: the *agent a* and the *environment e*. Both of them are able to observe at least part of the whole state space $\mathcal{S}$, i.e. they are aware of some observable states $\mathcal{O} \subset \mathcal{S}$. By distinguishing between the observable states of the agent $\mathcal{O}_a \subseteq \mathcal{O}$ and of the environment $\mathcal{O}_b \subseteq \mathcal{O}$, we are able to force uneven levels of knowledge between them.

Let $\mathcal{U}_a$ and $\mathcal{U}_e$ be the spaces of all possible actions for the two components. We discretize the evolution of the system as a discrete-time system with step Δt, which evolves according to a function $\psi : \mathcal{S} \times \mathcal{U}_a \times \mathcal{U}_e \times \mathbb{R} \longrightarrow \mathcal{S}$. By taking control actions at fixed time intervals of length Δt, we obtain a discrete evolution of the form $s_{i+1} = s_i + \psi(s_i, u_a^i, u_e^i, t_i)$, where $t_i := t_0 + i \cdot \Delta t$, $u^i := u(t_i)$ and $s_i := s(t_i)$. Therefore, we are able to simulate the entire evolution of the system over a time horizon H via ψ and to obtain a complete trajectory $\xi = s_0 \dots s_{H-1}$ in the state space.

Optimization strategy. The proposed framework builds on GAN architectural design, in which two NNs compete in a minmax game to reach opposite goals. One network, denoted by A, represents the *attacker*, while the other, denoted by D, represents the *defender*. The aim of the former is to generate environment configurations in which the defender is not able to act safely, whereas, the latter tries to keep the CPS as safe as possible. In practice, the defender D can be interpreted as a controller for the agent. The safety requirement is expressed as a Signal Temporal Logic formula Φ over a finite time horizon H. We are leveraging the notion of robustness in quantitative semantics to measure the satisfiability of the STL property and to determine how safe the system is in a given configuration. We denote robustness as a function $R_\Phi : \mathcal{S}^H \to \mathbb{R}$, measuring the maximum shift that can be applied to a given trajectory $\xi = s_0 \dots s_{H-1}$ without violating the requirements of Φ. It is straightforward to use this measure as the objective function in the minmax game. When the system is in a state s_0, the evolution of ξ is obtained by evaluating ψ at time steps $t_i = t_0, \dots, t_{H-1}$ over two sequences of actions $\mathbf{u}_a = (u_a^0, \dots, u_a^{H-1})$ and $\mathbf{u}_e = (u_e^0, \dots, u_e^{H-1})$. We introduce two *policy functions*, Π_A for the attacker and Π_D for the defender, with the aim of reducing the output dimension. The two policies $\Pi_A : \Theta_A \times \mathbb{R} \to \mathcal{U}_e$ and $\Pi_D : \Theta_D \times \mathbb{R} \to \mathcal{U}_a$ are represented by a finite set of basis functions of time, with coefficients given by the output of the networks; in this work they are polynomial functions. For example, Π_A can encode the output θ_A of network A as a polynomial function of degree d_A

$$u_e(t) = \Pi_A(\theta_A, t) := \sum_{j=1}^{d_A} \theta_{Aj} t^j$$

and the resulting $u_e(t)$ is evaluated at each time step, from t_0 to t_{H-1}, to produce the desired sequence of actions $\mathbf{u}_e$. The same reasoning holds for Π_D and $\mathbf{u}_a$. These policies have the benefit of producing a smoothing of the chosen actions, that prevents incoherent behaviours at subsequent instants.

Let $\mathbf{w}_A$ be the weights of the attacker's network A and $\mathbf{w}_D$ the weights of the defender's network D. The formalism introduced by the two policies transfers the problem of finding the best sequences of actions, $\mathbf{u}_a$ and $\mathbf{u}_e$, to that of finding the best networks' parameters, $\mathbf{w}_D$ and $\mathbf{w}_A$. The minmax game can now be expressed in terms of the loss function $\mathscr{L}(\mathbf{w}_A, \mathbf{w}_D) = -R_\Phi(s_0, \mathbf{w}_D, \mathbf{w}_A)$ as

$$\min_{\mathbf{w}_D} \max_{\mathbf{w}_A} \mathscr{L}(\mathbf{w}_A, \mathbf{w}_D).$$

In this setting, the defender aims at generating safe actions by tuning its weights in favour of a loss minimization (i.e. robustness maximization). The attacker, instead, aims at generating troubling scenarios for the opponent by maximizing the loss (i.e., minimizing the robustness).

3 Experimental Results

Car platooning. A *platoon* [3] is a group of vehicles travelling together very closely and safely. This problem is usually faced with techniques that coordinate the actions of the entire pool of vehicles as a single entity [7]. This approach, though, requires specific hardware and a distributed system of coordination that might be difficult to realise in complex scenarios. Our method, instead, builds a robust controller for individual decision-making, hence it fits into the autonomous driving field. In this setting, we assume that all vehicles are equipped with an hardware component called *LIDAR scanner*, which is able to measure the distance between two cars by using a laser beam.

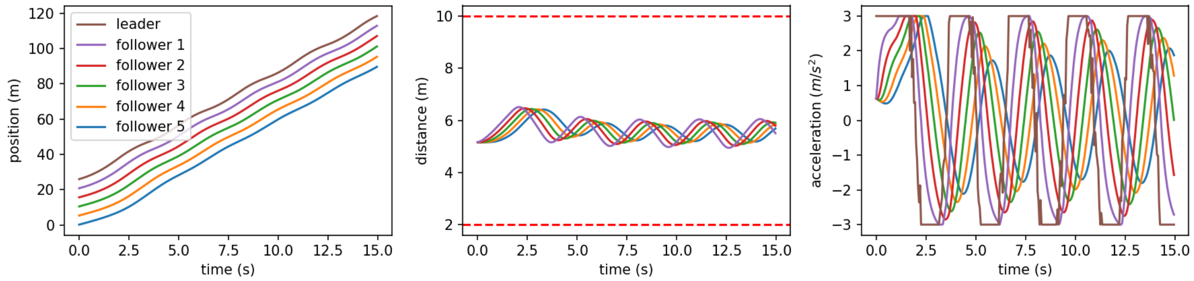

Figure 1: The leader acts according to the attacker's policy, while followers are controlled by copies of the same defender's network. Initial configuration: distance $d_0 = 6.33\ m$ and velocity $v_0 = 2.87\ m$ for all cars.

Training and Testing. Platooning involves n cars that can only move forward along a straight line. We first consider the simple case of two cars, one *leader* l and one *follower* f, whose internal states are position x, velocity v and acceleration a. The follower f acts as the agent of this system, while the leader l is considered to be part of the environment, representing for instance a cyber-attack scenario. They have the same observable states $\mathbf{o}_a = \mathbf{o}_e = (v_l, v_f, d)$, given by their velocities and by their relative distance d. The policy functions Π_A and Π_D output the accelerations $\mathbf{u}_a = (a_f)$ and $\mathbf{u}_e = (a_l)$, which are used to update the internal states of both cars. We describe the dynamic of a car with mass m and velocity v as $m\frac{\mathrm{d}v}{\mathrm{d}t} = ma_{in} - \nu mg$, where a_{in} is the input acceleration provided by one of the two policies, ν is the friction coefficient and g is the gravity constant. We impose the STL requirement $\Phi = \mathsf{globally}(d \leq d_{\max} \land d \geq d_{\min})$ on the distance $d = x_l - x_f$ between the two vehicles, where $d_{\min}$ and $d_{\max}$ are the minimum and maximum distances allowed. Note that the $\mathsf{globally}$ operator forces the STL condition to hold for the whole trajectory of the car.

Results. Car platooning problem trivially extends to the case of n cars, where the first one is the leader and each of the other cars simply follows the one in front. Our simulations start from an initial configuration of equispaced vehicles, thus the first couple of subsequent cars acts as described in the two-cars model, while the other followers are controlled by copies of the same defender's network.

This model has been tested in four different adverse configurations. The leader in Figure 1 acts according to the attacker's policy, with sudden accelerations and brakes, and all followers are able to manage the unpredictable behaviour of the attacker by maintaining their relative distances within the safety range. We ran $10k$ simulations of different trajectories for each possible scenario. At each time step we computed the total percentage of safe trajectories and 100% of them achieved positive robustness.

4 Conclusions

Classical control theory fails in giving adequate safety guarantees in many complex real world scenarios. New reinforcement learning techniques aim at modelling the behaviour of complex systems and learning optimal controllers from the observed data. Therefore, they are particularly suitable for stochastic optimal control problems where the transition dynamics and the reward functions are unknown. We proposed a new learning technique, whose architecture is inspired by Generative Adversarial Networks, and tested its full potential against the vehicle platooning problem. Our approach has been able to enforce safety of the model, while also gaining insights about adverse configurations of the environment. As future work, we plan to test more scenarios and investigate the scalability of this approach. We also plan to extend this control synthesis strategy to stochastic hybrid systems.[1]

References

[1] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath. A brief survey of deep reinforcement learning. *arXiv preprint arXiv:1708.05866*, 2017.

[2] G. Avni, R. Bloem, K. Chatterjee, T. A. Henzinger, B. Könighofer, and S. Pranger. Run-time optimization for learned controllers through quantitative games. In *International Conference on Computer Aided Verification*, pages 630–649. Springer, 2019.

[3] L. Banjanovic-Mehmedovic, I. Butigan, F. Mehmedovic, and M. Kantardzic. Hybrid automaton based vehicle platoon modelling and cooperation behaviour profile prediction. *Tehnicki vjesnik - Technical Gazette*, 25(3), Jun 2018.

[4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[5] V. Goranko and A. Rumberg. Temporal logic. In E. N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, spring 2020 edition, 2020.

[6] S. Howes, I. Mohler, and N. Bolf. Multivariable identification and pid/apc optimization for real plant application. In *ACHEMA–World Forum and Leading Show for the Process Industries*, 2018.

[7] D. Jia, K. Lu, J. Wang, X. Zhang, and X. Shen. A survey on platoon-based vehicular cyber-physical systems. *IEEE Communications Surveys & Tutorials*, 18(1):263–284, 2016.

[8] O. Maler and D. Nickovic. Monitoring temporal properties of continuous signals. In Y. Lakhnech and S. Yovine, editors, *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems, Joint International Conferences on Formal Modelling and Analysis of Timed Systems, FORMATS 2004 and Formal Techniques in Real-Time and Fault-Tolerant Systems, FTRTFT 2004, Grenoble, France, September 22-24, 2004, Proceedings*, volume 3253 of *Lecture Notes in Computer Science*, pages 152–166. Springer, 2004.

[9] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[10] X. Zheng and C. Julien. Verification and validation in cyber physical systems: Research challenges and a way forward. In *2015 IEEE/ACM 1st International Workshop on Software Engineering for Smart Cyber-Physical Systems*, page 15–18. IEEE, May 2015.

[1] This work has been partially supported by the PRIN project “SEDUCE” n. 2017TWRCNB.