Safe Intelligent Systems

Final Project – Paper Reading

Daniel Elmaleh

"Reasoning about Safety of Learning-Enabled Components in Autonomous Cyber-physical Systems"

Introduction

I am presenting a paper reading and analysis about safety verification in cyber physical systems with a focus on the paper "Reasoning about Safety of Learning-Enabled Components in Autonomous Cyber-physical Systems", which mainly presents an approach to those safety verifications based on simulations for generating barrier certificate functions to treat CPS that contain NN based controllers. A linear programming solver is used to find the generate function, then a level set of the function is selected to be the barrier certificate for the system (No unsafe states are reachable from an initial state). The barrier certificate properties are then verified using an SMT solver. It is then demonstrated on a simple autonomous car following a path example. I will then discuss other approaches to the topic as well as give my personal view.

Controller design and synthesis is one of the most fundamental problems in control theory. In recent years, especially with the boom of deep learning, there has been considerable research activities in the use of neural networks (NNs) for control of nonlinear systems. NNs feature the versatile representational ability of nonlinear maps and fast computation, making them an ideal candidate for sophisticated control tasks. Typical examples include self-driving cars, drones, and smart cities. It is noteworthy that many of these applications are safety-critical systems, where safety refers to, in a basic form, that the system cannot reach a dangerous or unwanted state. For control systems in a multitude of Cyber-Physical-System domains, designing safe controllers which can guarantee safety behaviors of the controlled systems is of paramount importance.

Summary of the reading paper

Goal

The key idea is to automatically learn safety invariants for a closed loop model (output is fed back with the inputs of the system). Those safety invariants are the barrier certificates.

Using simulations, the candidate barrier functions are generated. Then there is a verification of the overall safety of the system for checking the validity of those barrier certificate using a nonlinear SMT solver – dReal. To demonstrate this idea, they gave a simple autonomous vehicle example of path following case. The vehicle uses an NN controller trained by reinforced learning. And they prove that for a certain kinematic model the car never leaves a safe region around the path.

In this paper they focused on one of two main kinds of NN controllers, based on stateless feedforward NN whose behavior is defined by instantaneous mappings between inputs and outputs. The other kind would be RNN (recurrent neural networks) which employ feedback control and are stateful and whose behavior is defined by dynamical equations and are functions of inputs and internal states.

The model considered is described by :

$$\begin{aligned} \dot{\mathbf{x}} &= f_p(\mathbf{x}, \mathbf{u}), \\ \mathbf{y} &= g(\mathbf{x}), \end{aligned}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the state, $\mathbf{u} \in \mathbb{R}^m$ is the input to the plant, f_p is a locally Lipschitz-continuous vector field, and $g : \mathbb{R}^n \to \mathbb{R}^q$ defines the plant outputs.

And the NN controller is given by:

$$\mathbf{u} = h(\mathbf{y}),$$

where $h : \mathbb{R}^q \to \mathbb{R}^m$ is a function that maps plant outputs to plant inputs.

It is assumed that h performs all of the processing required to implement the NN, including applying the weights and activation functions that define the NN, as well as performing any necessary preprocessing of the inputs.

The composition of the equations gives:

$$\dot{\mathbf{x}} = f_p(\mathbf{x}, h(g(\mathbf{x}))),$$

Safety verification with strict barrier certificates

We assume that we are given an autonomous dynamical system described by (4), a set of possible initial states X_0, and a set of unsafe states U. Then, we define the barrier certificate as follows.

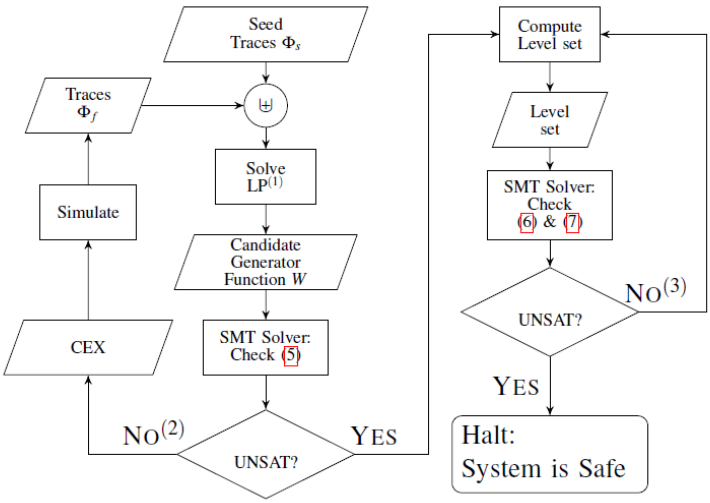
The strict barrier certificate is a differentiable function B from the set of states of the dynamical system to the set of reals.

$$\begin{aligned} &(1) \quad \forall \mathbf{x} \in X_0 : \quad B(\mathbf{x}) \leq 0, \\ &(2) \quad \forall \mathbf{x} \in U : \quad B(\mathbf{x}) > 0, \\ &(3) \quad \forall \mathbf{x} : \quad (B(\mathbf{x}) = 0) \implies (\nabla B)^T \cdot f(\mathbf{x}) < 0. \end{aligned}$$

the existence of a suitable barrier certificate demonstrates that along any system trajectory with the initial state in X_0, a state in U is not reachable (in finite or infinite time). Thus, a barrier certificate provides a powerful unbounded-time safety certificate of the system.

Solution

The solution consists in defining a barrier certificate as a level set of a generator function W(x), when B(x)=W(x)-l. The generator function is a positive function that decreases along the system trajectories. The method starts by performing a collection of simulations to generate a set of linear constraints that specify the positivity of the candidate generator function. Condition (3) is then checked using an SMT solver, which in turn produces a counterexample in case of unsatisfaction or in confirmation of the candidate. Finally, the generator function is used to find the right value of l that separates the initial condition set from the unsafe set, and thus acts as the barrier certificate for the system. This process is described in more detail in the flow chart figure below :



First, they create a collection of linear constraints using results from the simulations ϕs. A linear program is solved to obtain a solution which corresponds to the candidate generator function W(x). Next an SMT solver is used to check :

$$\exists \mathbf{x} \in \mathscr{D} : \ (\mathbf{x} \notin X_0) \wedge ((\nabla W)^T \cdot f(\mathbf{x})) \geq -\gamma.$$

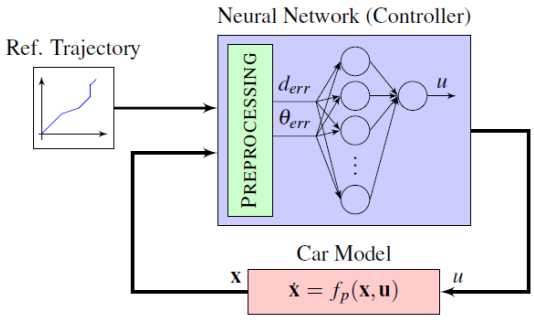
This iterative process is repeated until UNSAT is returned, and in this case the computation of the level set size l such that B(x) contains the initial conditions X0 does not intersect with the unsafe set U. once the level set is selected, a pair of SMT solvers are used to check the negation of :

$$\exists \mathbf{x} \in X_0, \mathbf{x} \notin L,$$
$$\exists \mathbf{x} \in L, \mathbf{x} \in U,$$

Which will return UNSAT if the desired property holds. In case SAT is returned, this iterative process continues until obtaining the right l. At the end we get a barrier certificate B(x)=W(x)-l that guarantees the safety of the system – no unsafe states are reachable.

Case study

They presented a simple case study to demonstrate the method to proof the safety of an autonomous CPS with a NN controller described in the figure below :



Vehicle model

It's a 2-D model defined by its position (X_v,Y_v) and its orientation (θ_v) :

$$\dot{x}_v = V \sin\theta_v$$
$$\dot{y}_v = V \cos\theta_v$$
$$\dot{\theta}_v = u$$

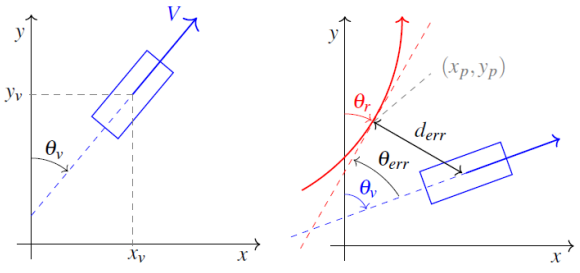
U is the turn rate control, or steering. V is supposed to be constant.

Path following and error dynamics

As seen on the figure below d_{err} and θ_{err} define the distance and the angle error with respect to the desired path. With a few trigonometric identities and a bit of algebra, considering the NN controller as a function h which maps inputs (d_{err}, θ_{err}) to output u (the input of the plant), the closed loop dynamics of the system can be defined by :

$$\mathbf{x} = [d_{err}\ \theta_{err}]^T$$
$$\dot{\mathbf{x}} = f_p(\mathbf{x}, \mathbf{u})$$
$$= \begin{bmatrix} -V \sin(\theta_r - \theta_{err}) \cos(\theta_r) + V \cos(\theta_r - \theta_{err}) \sin(\theta_r) \\ -u \end{bmatrix}$$
$$\mathbf{y} = g(\mathbf{x}) = [d_{err}\ \theta_{err}]^T$$
$$u = h(\mathbf{y}).$$

Where x is the system stat vector.



(a) Dubins car model. (b) Path following errors.

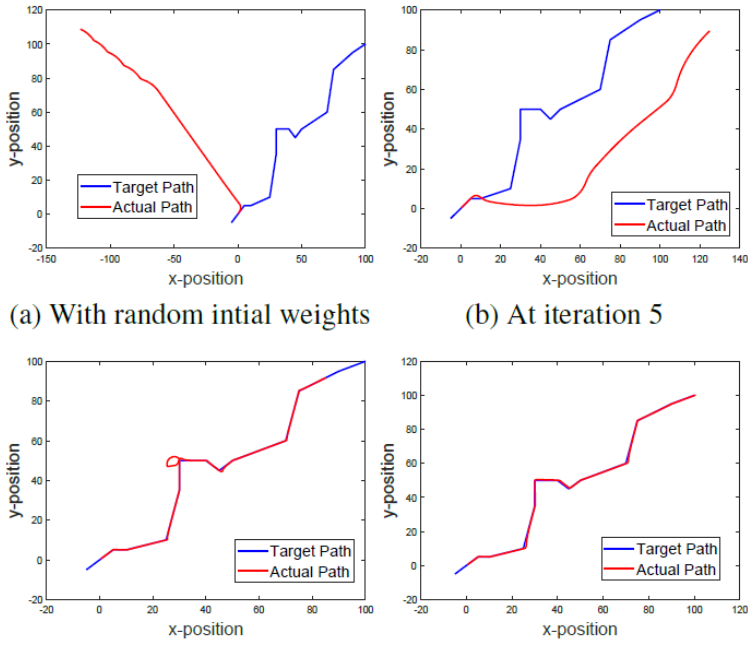
Learning the controller

They first select the structure of the NN to be a feedforward NN with one hidden layer and Nh neurons, with 2 inputs and 1 output. They use a MATLAB reinforced learning NN algorithm CMA-ES, which is implemented with the cost function :

$$J = \sum_{k\in\{0,\dots,N\}} \left(100 d_{err_k}^2 + 10^5 \theta_{err_k}^2 + 100 u_k^2\right) + 10^3 |(x_{end}, y_{end}) - (x_{vN}, y_{vN})|^2.$$

Where N represents the number of time steps in the simulation.

In the figure below we can see a few of their results in different iterations during the learning process :



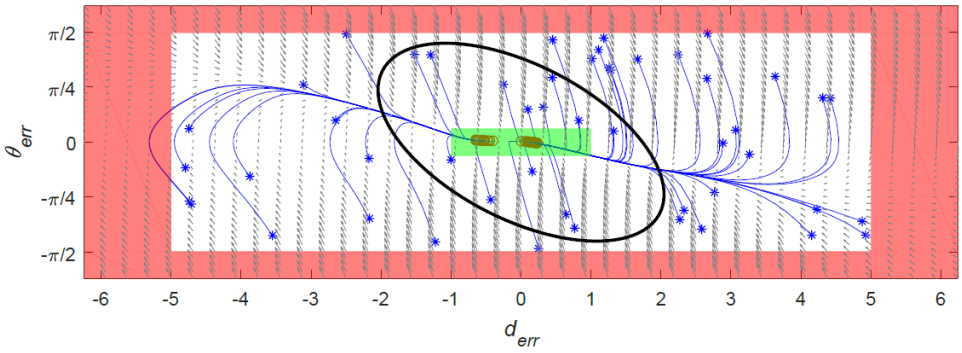
(a) With random intial weights (b) At iteration 5

Verification of the results

They applied their approach to different versions of NN controllers with different numbers of neurons in the hidden layer. It was demonstrated that the method scales well with the size of the NN. It was assumed to be a straight line as path for the controller. For each verification, X_0 is given by the rectangular area defined by the diagonal corners and, and U is the complement (outside) of the rectangle. The table below shows their results:

Number of Neurons	Computing Generator				Time Spent in Other Steps (s.)	Total Time (s.)
	Avg. Num. Iterations	Time Spent (s.)				
		LP	Query	Total		
10	3.0	1.1	4.25	43	12	55
20	1.8	1.1	2.6	21	11	32
40	1.7	1.2	5.3	26	14	40
50	1.5	1.6	4.8	35	14	49
70	2.8	1.8	15.6	106	16	122
80	1.2	1.7	4.3	28	15	43
90	1.0	2.0	4.7	27	16	43
100	1.7	1.1	4.1	21	14	35
300	1.7	1.8	379.8	698	48	746
500	1.3	1.9	379.4	536	107	643
700	1.0	2.0	19.1	41	35	76
1000	1.0	2.0	50.4	74	79	153

The figure below illustrates the results of verification for one of the cases captured in the Table. The lateral axis of the figure represents the position error, and the vertical axis represents the angle error (θ_{err}). The initial condition set X_0 is shown in green, and the unsafe set U is shown in red. The simulation trajectories PHIs are shown in blue. initial conditions for each trajectory are marked with an * and end points are marked with a yellow circle. The ellipsoid between the X_0 and U sets is a level set of a generator function found using this approach. The barrier properties are all determined to be UNSAT by the dReal SMT solver. Hence, the ellipsoid is a barrier certificate for the system, which means that the system is safe.



Conclusion

In the paper, a technique to reason about safety of for a closed loop control system using a learning-enabled controller. Particularly focusing on feedforward artificial neural network-based controllers. The key idea is to reduce the safety verification problem to the identification of a barrier certificate candidate, using simulations. Then to perform verifications on the synthesized barrier certificate. The final verification is done by using a nonlinear SMT solver, which lets the treatment of arbitrary nonlinear activated NN. This method was demonstrated on a simple closed-loop model of a car following a straight-line path.

Conclusion

Over simplicity of the example. Not clear enough. we have not yet applied our technique to real-world ACPS designs and has thus not encountered the concomitant scalability challenges.

Future work will focus on improving the scalability of our technique and investigating stateful controllers based on recurrent neural networks. We will also investigate algorithms to simultaneously train the neural network while satisfying safety guarantees.

I was able to learn and understand better how model verifications are done with more of a mathematical perspective and could better understand NN and their use in different applications, as controllers, as verification tools and so on.

Relation to course

This topic and especially this paper relate very well with the content of the course focusing on verifying safety properties for hybrid systems and verification of neural networks. Reading this paper gave me a better understanding of the lecture's material and vice versa. The importance of generalization and automation of the verifications of systems, especially of the safety properties and reachability of states, is very important due to the increasing complexity of systems and since a lot of modern systems use machine learning tools to operate which makes a big part of the system invisible to the designer. To achieve all this, we can state that mathematical rigurosity is key for these subjects because of the abstraction needed to get good results in a modern complex world based on artificial intelligence.

State of art exploration

In other scientific papers we can find a variety of other methods, such as using a NN to approximate the barrier certificate function itself, instead of a linear program solver like in the article treated in my reading. In addition, those approaches consist of synthesizing NN controllers for nonlinear systems with control against safety properties with verification in-the-loop, where the controller and its certificates are trained simultaneously. Those methods were shown effective in article [1].

In another paper [2] about safety verifications in CPS with reinforcement learning controllers a new forward reachability analysis is proposed. The approach lies on two efficient, exact, and over-approximate reachability for neural network control systems using star sets, which is an efficient representation of polyhedral. Using those algorithms, the safe initial conditions are determined by incrementally searching for a critical initial condition where the safety of the system cannot be established. This approach produces tight over-approximation error, and it is computationally efficient, which allows the application to practical CPS with learning enable components (LECs). In this paper it was shown that this method is computationally cheaper and less conservative than the existing polyhedron approach, and applicable to real-world applications.

In this paper from 2021 [3] a work providing a survey of state of art safety validation techniques for CPS with a focus on applied algorithms and their modifications for the safety validation problem. In this article were presented algorithms in the domains of optimization, path planning, reinforced learning and

importance sampling. Problem decomposition techniques are presented to help scale algorithms to large state spaces, which are common for CPS.

Three safety validation tasks for a system with safety properties were considered. First, falsification aims to find an example disturbance in the environment that causes the system to violate the property. This formulation is useful for discovering previously unknown failure modes and finding regions where the system can operate safely. The second safety validation task is to find the most-likely failure according to a probabilistic model of the disturbances. The model can be created through expert knowledge or data to reflect the probabilities in the real environment. The third safety validation task is to estimate the probability that a failure will occur. Failure probability estimation is important for acceptance and certification. Global optimization, path planning, and reinforcement learning algorithms have been used to find falsifying examples, while importance sampling methods have been used to estimate the probability of failure even when it is close to zero.

References

[1] “Learning Safe Neural Network Controllers with Barrier Certificates” –

https://arxiv.org/pdf/2009.09826.pdf

[2] ”Safety Verification of Cyber-Physical Systems with Reinforcement Learning Control” - https://dl.acm.org/doi/pdf/10.1145/3358230

[3] “A Survey of Algorithms for Black-Box Safety Validation of Cyber-Physical Systems” - https://www.jair.org/index.php/jair/article/view/12716/26727