Identification of Safety Regions in Vehicle Platooning via Machine Learning

Alessandro Fermi, Maurizio Mongelli, Marco Muselli
Institute of Electronics, Computer and Telecommunication Engineering
National Research Council of Italy
Genova, Italy
Email: name.surname@ieiit.cnr.it

Enrico Ferrari
Impara S.r.l.
Genova, Italy
Email: e.ferrari@impara-ai.com

***Abstract*—The paper introduces the use of machine learning with rule generation to validate collision avoidance in vehicle platooning. Cooperative Adaptive Cruise Control is under test over a range of system parameters including speed and distance of the vehicles as well as packet error rate of the communication channel. Safety regions are evidenced on test data with statistical error very close to zero.**

I. Introduction

Safety in *cyber-physical systems* (CPS) means preventing all the conditions that may lead to dangerous trajectories of the system. The approach of machine learning consists in employing a sample of data to extract a model that, starting from a set of system parameters, returns a prediction about its safe or unsafe evolution.

This approach can lead to a great benefit as finding closed-form formulas of control performance over communication channels in CPS constitutes a formidable problem (see, e.g., [1], [2]). *Vehicle platooning* (*VP*) is taken as a reference here as being representative of one of the most challenging CPS of the automotive sector [3]. The main goal in VP is finding the best trade-off between performance (i.e., maximizing speed and minimizing vehicles reciprocal distance) and safety (i.e., avoiding collision) [4]. The aim of the work is to show how intelligible analytics facilitate sensitivity analysis of safety conditions.

II. System under test

A. Safety metric

The following scenario is considered. Given the platoon at a steady state of speed and reciprocal distance of the vehicles, a braking is applied by the leader of the platoon [5], [6]. The behavior of the dynamical system is then investigated to predict collision between adjacent vehicles. A collision is registered when the reciprocal distance reaches a lower bound (e.g., 2 m).

B. Dynamics

The *Cooperative Adaptive Cruise Control* (CACC) is considered, having the following time dynamics [6], [7]. Let $\chi_i = x_{i-1} - l_{i-1} + g_i^{des}$ be the desired i-th vehicle position (x_{i-1} and l_{i-1} being the position and length of the preceding vehicle, respectively and g_i^{des} the intervehicle desired gap), the desired acceleration is tuned according to:

$$\begin{aligned}\ddot{\chi}_i = (1 - C_1)\ddot{x}_{i-1} + C_1\ddot{x}_0 - \\ (2\xi - C_1(\xi + \sqrt{\xi^2 - 1}))\omega_n\dot{\epsilon}_i - \\ (\xi + \sqrt{\xi^2 - 1})\omega_n C_1(v_i - v_0) - \omega_n^2\end{aligned} \tag{1}$$

C_1, ω_n and ξ being gain, damping ratio and bandwidth of the controller, v_i the speed of vehicle i and $\epsilon_i = d_i + l_{i-1} + g_i^{des}$. Stability is guaranteed under $\xi \geq 1$ and $C_1 < 1$ [7]; [8] (section IV.E) outlines how CACC may become unstable under 60% of packet losses.

C. Information vector

The behavior of the dynamical system is synthesized by the following vector of features:

$$\boldsymbol{I} = [N, \boldsymbol{\iota}(0), \boldsymbol{m}, F_0, \boldsymbol{q}, \boldsymbol{p}] \tag{2}$$

$N + 1$ being the number of vehicles in the platoon [1], $\boldsymbol{\iota} = [\boldsymbol{d}, \boldsymbol{v}, \boldsymbol{a}]$ the vectors of reciprocal distance, speed and acceleration of the vehicles, respectively[2], $\boldsymbol{m}$ the vectors of the masses of the vehicles, F_0 the braking force applied by the leader, $\boldsymbol{q}$ the vector of quality measures of the communication medium, *packet error rate* (PER) is considered in the paper, $\boldsymbol{p}$ the vector of tuning parameters of the control scheme.

D. Performance prediction

The problem consists of predicting whether the dynamics may achieve undesired or unsafe behavior of the system in the near future after braking. The Information vector $\boldsymbol{I}(\cdot)$ drives the forecast. Let $f(\boldsymbol{I}(\cdot), \cdot)$ be a generic prediction function that, after proper training at design time through simulations or historical data collected on the real system, may be interrogated on line in order to drive predictions. This means $\boldsymbol{I}(\cdot)$ is updated at run time to trigger decisions while the system is working on line [3].

The work has been done in the framework of the project SafeCOP, call H2020-ECSEL-2015-RIA-two-stage, ref. 692529-2.

[1] Subscript $i = 0$ defines the index of the leader.

[2] $\boldsymbol{\iota}(0)$ denotes that the quantities are sampled at time t=0, after which a braking force is applied by the leader [5]. Simulations are set in order to manage possible transient periods and achieve a steady state of $\boldsymbol{\iota}$ before applying the braking.

III. Problem

The prediction function $f(\boldsymbol{I}(\cdot),\cdot)$ is investigated by setting a supervised learning problem. Let $\omega \in \{0,1\}$ denote the under and over threshold events related to the phenomenon of interest, e.g. a collision ($\omega = 1$) or a safe evolution ($\omega = 0$) in the platoon after leader braking. Let $\aleph = \{(\boldsymbol{I}^{\kappa}, \omega^{\kappa}), \kappa = 1, ..., K\}$ be a dataset corresponding to the collection of events representing the platoon evolution (ω) under different system settings ($\boldsymbol{I}(\cdot)$). The collection of points in $\aleph$ are derived by iteration over the available system under test (platooning simulator, in this case). The classification problem consists of finding the best boundary function $f(\boldsymbol{I}(\cdot),\cdot)$ separating the $\boldsymbol{I}^{\kappa}$ points in $\aleph$, according to the two classes defined by ω.

IV. Methodology

The derivation of $f(\boldsymbol{I}(\cdot),\cdot)$ is made by conventional Decision Tree (DT) and Logic Learning Machine (LLM) [9][3]. They are both based on a set of intelligible rules of the type **if** (*premise*) **then** (*consequence*), where (*premise*) is a logical product (AND, $\wedge$) of conditions and (*consequence*) provides a class assignment for the output. In the present study, the class $\omega = 1$ corresponds to the occurrence of a collision whereas $\omega = 0$ is associated with a safe evolution.

A. Safety regions

The final goal of the validation process is to identify the largest subset of parameters with no false negatives (i.e., prediction of no collision, but collision in reality). To do this, three methods are studied here.

1) First, data visualization of the 2 (or 3) most important features after ranking may be applied to manually inspect the most relevant regions for safety (see the appendix of [10] for details on ranking through analysis of rules).

2) Secondly, the LLM is used; differently from DT, it identifies rules by clustering data in a Boolean space. Collision avoindance points are selected with a safe margin because LLM is trained with zero error, namely, each rule derived by LLM for a given class does not misclassify any point of the other class. The rules for the safe class with the largest covering are then joined together under logical OR ($\vee$) to build the predictor. This approach builds a strict boundary around safe points that could be even too conservative, especially if few rules with highest covering are selected, but with the necessary guarantee to limit false negatives.

3) A more refined procedure is elaborated as well. Since trained classifiers with zero error have the risk to polarize the model on training data, thus precluding the capability for generalization to new (test) datasets, we still train the classifiers while keeping a non-zero margin for error (e.g., 5%), but we progressively extract unsafe points from the original dataset until only safe points are obtained. The mechanism is shown in Fig. 1. The principle of κ-fold cross-validation has

Data set $\aleph$ divided into κ portions $\aleph_\kappa$.
$\aleph_{1\kappa}$ includes only unsafe points from $\aleph_\kappa$

repeat κ times
  repeat
    1. Train($\aleph_\kappa + \aleph_{1\kappa}$)
    2. $R_{0\kappa}$ subset of 'safe' rules
    3. Manual inspection $R_{0\kappa} \to R'_{0\kappa}$
    4. Apply($R'_{0\kappa}$, $\aleph_\kappa$)
  until $\aleph_\kappa$ contains safe points only

Choose most stringent conditions from $R'_{0\kappa}$.

Fig. 1: Procedure for safety rule extraction.

been taken as a reference[4]. At the beginning, $\aleph_{1\kappa}$ is already included in $\aleph_\kappa$ at step 1. However, the presence of duplicated data is ignored by the training algorithm. Step 2 presents rules in the $R_{0\kappa}$ subset, ordered by covering. The subsequent step selects the first rules having a difference on covering between 5% and compares them as follows. If they express identical features, the most stringent thresholds are selected for the feature (e.g., if two rules state $v(0) < 30 \wedge d(0) > 10$ and $v(0) < 29 \wedge d(0) > 15$, respectively, the output becomes $v(0) < 29 \wedge d(0) > 15$).

V. Performance evaluation

The Plexe simulator [6], [8] is used to register samples of the model presented in section II-B[5], under the following ranges:

$N \in [3,8]$, $F_0 \in [-8,-1] \cdot 10^3$ N[6], $PER \in [0.2, 0.5]$, $d(0) \in [4,9]$ m, $v(0) \in [10,90]$ Km/h.

Vehicles have equal masses of 1500 Kg. Communication channel is IEEE 802.11p. In order to stress working conditions, only the information related to the precedent vehicle is delivered to each vehicle; this is obtained by setting the control gain $C_1 = 0$ [7]. Potential situations of collision are investigated as outlined in section II-A. The complex profile of the two classes is evidenced in Fig. 2, in which samples are presented with respect to the first two features after feature ranking. The dataset contains $3 \cdot 10^3$ points. Other $12 \cdot 10^3$ points for test have been derived from further simulations. *False positive* (FPR) and *false negative* (FNR) *rates* are used for validation of the reliability of prediction. The three methods defined in section IV-A are then applied to infer safety regions.

A. Rules for safety regions

By inspecting Fig. 2, it is intuitive to identify the following safety region (first method of section IV-A):

[3]The analysis was performed through the Rulex software suite, developed and distributed by Rulex Inc (http://www.rulex.ai/).

[4]It consists in subdividing the total dataset into κ parts of equal number and, at each step, the κ-th part of the dataset comes to be the validation dataset, while the remaining part constitutes the training dataset

[5]Eqs. (7-12) of [11] are implemented in the simulator as stated in [6].

[6]From now on, notation ($\cdot 10^3$) is omitted when referring to thresholds applied to F_0.

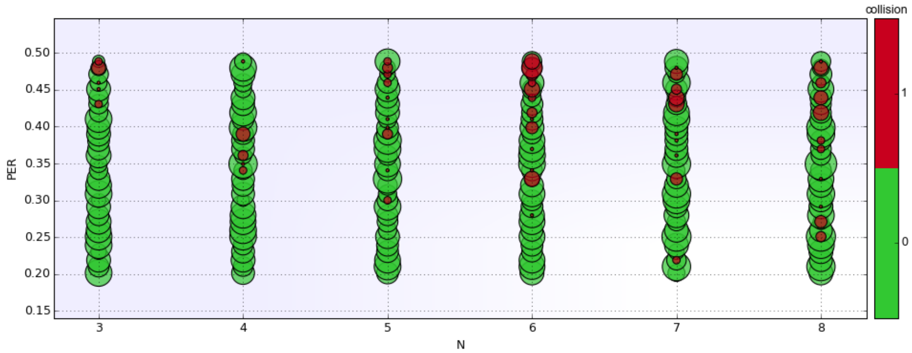

Fig. 2: Dataset with respect to scatter plot of N-PER.

$(N \leq 6) \wedge (PER < 0.26)$ or, by more accurate inspection:

manual calibration:
if $((N = 6) \wedge (PER < 0.253))$ **then safe**;
if $((N = 5) \wedge (PER < 0.258))$ **then safe**;
if $((N = 4) \wedge (PER < 0.325))$ **then safe**;
if $((N = 3) \wedge (PER < 0.42))$ **then safe**;

LLM and DT are then tuned according to the second and third method outlined in section IV-A:

LLM:
if $((PER \leq 0.325) \wedge (N \leq 7) \wedge (F_0 \geq -8) \wedge (d(0) \geq 4.2385))$ $(C = 30\%, E = 0\%)$
$\vee$ (**if** $(F_0 \geq -8) \wedge (d(0) \geq 4.69) \wedge (v(0) \leq 37))$) $(C = 27\%, E = 0\%)$
$\vee$ (**if** $(F_0 \geq -7) \wedge (PER \leq 0.445) \wedge (v(0) \leq 41))$) $(C = 26\%, E = 0\%)$
$\vee$ (**if** $(F_0 \geq -8) \wedge (PER \leq 0.405) \wedge (d(0) \geq 5.5055) \wedge (v(0) \leq 53))$) $(C = 26\%, E = 0\%)$
$\vee$ (**if** $(v(0) \leq 28))$) $(C = 25\%, E = 0\%)$
then safe

DT:
if $(v(0) \leq 28)$ **then safe** $C = 59\%, E = 0\%$;

Two iterations of the inner steps of the rule extraction procedure (Fig. 1) are needed to derive the DT rule; at the first iteration, the rule found after the first training ($v(0) \leq 42$) still leaves 1% of unsafe points. The most stringent condition on $v(0)$ is chosen after $\kappa = 5$ cross validation. On the other hand, no further iterations have been applied to LLM as it identifies only safe points at the subsequent iteration.

B. Size of safety regions

Fig. 3 presents how many points are safe in different extractions of 10^3 subsets with different sizes from 8% to 50% of the total points available $(12 \cdot 10^3)$; $11 \cdot 10^3$ trials in total. LLM is considered with respect to the 5 rules in logical OR above ($\vee$) and to the first one with highest covering. The LLM follows the boundary in Fig. 2 better than the other methods, thus extracting a higher number of safe points. FNR is always 0% for DT and very close to 0% for the other techniques.

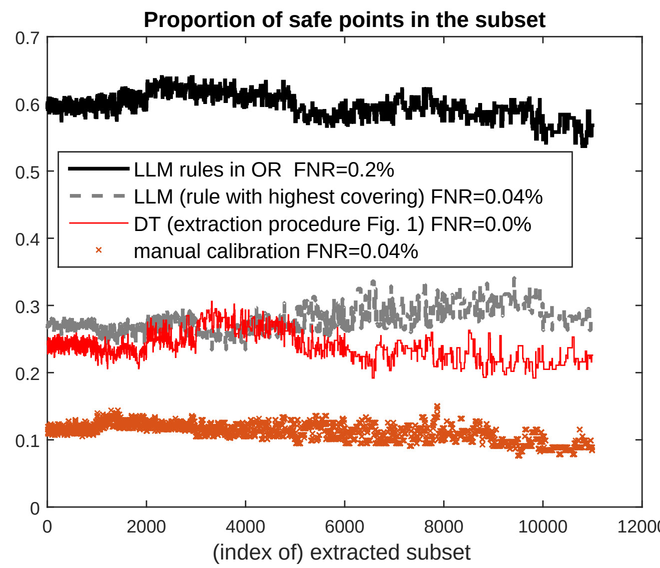

Fig. 3: Size of safety regions and FNR.

The averages of FNR over the trials are: 0.0%, 0.04%, 0.2% and 0.04% for DT, LLM single rule, LLM 5 rules and manual calibration, respectively. 99 percentiles are: 0.0%, 0.2%, 0.7% and 0.2%. Averages of FPR are: 7%, 7%, 3.4% and 8%.

C. Discussion

To summarize, the analysis gives a clear indication about the quantity of safe points in the chosen ranges of the parameters: up to 60% if 0.2% of FNR may be accepted, less than 30% if one wants to identify 0% of FNR. The corresponding DT rule for the latter is simple, $v(0) \leq 28$ Km/h, but impractical as it leads to platoons working at low speed. LLM 5 rules includes the DT rule (last rule of LLM), but also identifies safe platoons with higher speeds. For example, the first LLM rule selects safe points independently to $v(0) \in [10, 90]$ Km/h and the following 3 rules accept $v(0) > 28$ Km/h. It is finally worth noting that the gap between the safety regions under 0.2% and 0% of FNR is considerable and denotes the difficulty of the problem.

VI. Conclusion and Future Work

The paper shows how machine learning with rule generation helps derive sensitivity of safety conditions in vehicle platooning. Future work includes comparison with black-box approaches, control parameters setting [15], inclusion of vehicle actuators impact [8] and security issues [19].

References

[1] L. Xiao, M. Johansson, H. Hindi, S. Boyd, and A. Goldsmith, "Joint optimization of communication rates and linear systems," *IEEE Transactions on Automatic Control*, vol. 48, no. 1, pp. 148–153, Jan 2003.

[2] N. Xiao, L. Xie, and L. Qiu, "Feedback stabilization of discrete-time networked systems over fading channels," *IEEE Transactions on Automatic Control*, vol. 57, no. 9, pp. 2176–2189, Sept 2012.

[3] P. Pop, D. Scholle, I. Sljivo, H. Hansson, G. Widforss, and M. Rosqvist, "Safe cooperating cyber-physical systems using wireless communication," *Microprocessors and Microsystems*, vol. 53, pp. 42–50, 2017.

[4] D. Jia, K. Lu, J. Wang, X. Zhang, and X. Shen, "A survey on platoon-based vehicular cyber-physical systems," *IEEE Communications Surveys Tutorials*, vol. 18, no. 1, pp. 263–284, Firstquarter 2016.

[5] L. Xu, L. Y. Wang, G. Yin, and H. Zhang, "Communication information structures and contents for enhanced safety of highway vehicle platoons," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 9, pp. 4206–4220, Nov 2014.

[6] S. Santini, A. Salvi, A. S. Valente, A. Pescape, M. Segata, and R. L. Cigno, "A consensus-based approach for platooning with inter-vehicular communications," in *2015 IEEE Conference on Computer Communications (INFOCOM)*, April 2015, pp. 1158–1166.

[7] P. Fernandes and U. Nunes, "Platooning with ivc-enabled autonomous vehicles: Strategies to mitigate communication delays, improve safety and traffic flow," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 1, pp. 91–106, March 2012.

[8] S. Santini, A. Salvi, A. S. Valente, A. Pescape, M. Segata, and R. L. Cigno, "A consensus-based approach for platooning with intervehicular communications and its validation in realistic scenarios," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 3, pp. 1985–1999, March 2017.

[9] M. Muselli and E. Ferrari, "Coupling logical analysis of data and shadow clustering for partially defined positive boolean function reconstruction," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 1, pp. 37–50, Jan 2011.

[10] D. Cangelosi, M. Muselli, S. Parodi, F. Blengio, P. Becherini, R. Versteeg, M. Conte, and L. Varesio, "Use of attribute driven incremental discretization and logic learning machine to build a prognostic classifier for neuroblastoma patients," *BMC Bioinformatics*, vol. 15, no. 5, p. S4, May 2014. [Online]. Available: https://doi.org/10.1186/1471-2105-15-S5-S4

[11] M. Segata, S. Joerer, B. Bloessl, C. Sommer, F. Dressler, and R. L. Cigno, "Plexe: A platooning extension for veins," in *2014 IEEE Vehicular Networking Conference (VNC)*, Dec 2014, pp. 53–60.

[12] S. Nair, J. L. de la Vara, M. Sabetzadeh, and L. Briand, "An extended systematic literature review on provision of evidence for safety certification," *Information and Software Technology*, vol. 56, no. 7, pp. 689 – 717, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0950584914000603

[13] J. Cámara, R. de Lemos, N. Laranjeiro, R. Ventura, and M. Vieira, "Testing the robustness of controllers for self-adaptive systems," *Journal of the Brazilian Computer Society*, vol. 20, no. 1, p. 1, Jan 2014. [Online]. Available: https://doi.org/10.1186/1678-4804-20-1

[14] A. Aerts, M. Reniers, and M. Mousavi, "Chapter 19 - model-based testing of cyber-physical systems," in *Cyber-Physical Systems*, ser. Intelligent Data-Centric Systems, H. Song, D. B. Rawat, S. Jeschke, and C. Brecher, Eds. Boston: Academic Press, 2017, pp. 287 – 304. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128038017000195

[15] S. Oncu, N. van de Wouw, and H. Nijmeijer, "Cooperative adaptive cruise control: Tradeoffs between control and network specifications," in *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Oct 2011, pp. 2051–2056.

[16] M. Segata and R. L. Cigno, "Automatic emergency braking: Realistic analysis of car dynamics and network performance," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 9, pp. 4150–4161, Nov 2013.

[17] J. Ge and G. Orosz, "Dynamics of connected vehicle systems with delayed acceleration feedback," *Transportation Research Part C: Emerging Technologies*, vol. 46, pp. 46–64, 2014, cited By 90. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84901592910&doi=10.1016%2fj.trc.2014.04.014&partnerID=40&md5=42caadf7c8ca2afa2c3861221daafe3e

[18] K. Meinke, "Learning-based testing of cyber-physical systems-of-systems: A platooning study," in *Computer Performance Engineering*, P. Reinecke and A. Di Marco, Eds. Cham: Springer International Publishing, 2017, pp. 135–151.

[19] S. Marchesani, L. Pomante, M. Pugliese, and F. Santucci, *Definition and Development of a Topology-Based Cryptographic Scheme for Wireless Sensor Networks*. Cham: Springer International Publishing, 2013, pp. 47–64. [Online]. Available: https://doi.org/10.1007/978-3-319-04166-7_4