Evaluating Two Semantics for Falsi?cation using an Autonomous Driving Example

Citation for the original published paper (version of record):
Ramezani, Z., Smallbone, N., Fabian, M. et al (2019)
Evaluating Two Semantics for Falsi?cation using an Autonomous Driving Example
2019 IEEE 17th International Conference on Industrial Informatics (INDIN): 386-391
http://dx.doi.org/10.1109/INDIN41052.2019.8972229

Evaluating Two Semantics for Falsification using an Autonomous Driving Example

Zahra Ramezani[†], Nicholas Smallbone[‡], Martin Fabian[†], Knut Åkesson[†]
[†]*Department of Electrical Engineering*
[‡]*Department of Computer Science and Engineering*
Chalmers University of Technology
Gothenburg, Sweden
{rzahra, nicsma, fabian, knut}@chalmers.se

***Abstract*—We consider the falsification of temporal logic properties as a method to test complex systems, such as autonomous systems. Since these systems are often safety-critical, it is important to assess whether they fulfill given specifications or not. An adaptive cruise controller for an autonomous car is considered where the closed-loop model has unknown parameters and an important problem is to find parameter combinations for which given specification are broken. We assume that the closed-loop system can be simulated with the known given parameters, no other information is available to the testing framework. The specification, such as, the ability to avoid collisions, is expressed using Signal Temporal Logic (STL). In general, systems consist of a large number of parameters, and it is not possible or feasible to explicitly enumerate all combinations of the parameters. Thus, an optimization-based approach is used to guide the search for parameters that might falsify the specification. However, a key challenge is how to select the objective function such that the falsification of the specification, if it can be falsified, can be falsified using as few simulations as possible. For falsification using optimization it is required to have a measure representing the distance to the falsification of the specification. The way the measure is defined results in different objective functions used during optimization. Different measures have been proposed in the literature and in this paper the properties of the *Max* Semantics (*MAX*) and the Mean Alternative Robustness Value (*MARV*) semantics are discussed. After evaluating these two semantics on an adaptive cruise control example, we discuss their strengths and weaknesses to better understand the properties of the two semantics.**

***Index Terms*—Testing, Falsification, Max Semantics, Mean Alternative Robustness Value, Autonomous Driving.**

I. Introduction

For autonomous systems in general and autonomous vehicles in particular, it is critical to use rigorous testing methods so that such vehicles will be significantly safer than they are with humans in the loop. Autonomous systems consist of perception, sensor-fusion, decision and control modules implemented in software that interact with the physical sensors and actuators of the system. As remarked in [1], the biggest challenge for autonomous vehicles is in creating an end-to-end design and deployment process that integrates the safety concerns. Formal verification and correct by construction techniques should certainly be used for those sub-systems where they can be applied. However, it is known [2] that for hybrid systems, i.e., systems consisting of both digital and analog components, the problem of deciding if a state is reachable or not, is undecidable in general. Thus, for any autonomous systems of reasonable complexity, testing will be an important part of the design process.

Model-based design is often used for the design of autonomous systems, and building high fidelity models of both software and hardware is a part of the design process. Since testing on physical hardware is both time-consuming and limited by the available physical hardware, it is an advantage to do as much testing as possible using only the models. One approach [3] is to use formal specifications of properties that the closed-loop system should satisfy combined with the use of simulation of the models to evaluate whether the desired properties are fulfilled or not. This can be combined with *falsification* techniques that search for counterexamples to given specifications of the closed-loop system.

Metric Interval Temporal Logic (MITL) [4] and Signal Temporal Logic (STL) [5] can be used to describe real-time properties of systems. Several quantitative semantics for these logics have been proposed to not only allow reasoning about the correctness of a signal with respect to a model but also give a real value that indicates how far a signal is from satisfying or violating a specification. During falsification, these values are used by an optimizer to find new input signals with a higher likelihood of violating the specification. The quantitative semantics chosen will influence the efficiency of the falsification procedure, and the efficiency of different quantitative semantics is problem-dependent.

To facilitate different quantitative semantics the concept of *Valued Booleans* (VBools) was introduced in [6]. Multiple quantitative semantics can be expressed using VBool, for example the *MAX* semantics, that is a widely used semantics for optimization based falsification. Also [7] introduced and investigated several alternative robustness measures, of which the Mean Alternative Robustness Value (*MARV*) will be considered here.

In this paper, we evaluate the feasibility of automated falsification techniques for one sub-system, an adaptive cruise controller, of an autonomous vehicle. The purpose of this paper is not to evaluate how automated falsification techniques can be used for fully autonomous vehicles but to improve our understanding of how different semantics can be used to facilitate the falsification process. Given formal specification

978-1-7281-2927-3/19/$31.00 ©2019 IEEE

and simulation traces, the semantic will result in a scalar that not only expresses if the specification is fulfilled or not but also to what extent. This additional information is used by an optimizer to adjust the input signals and parameters to the next simulation with the intent of finding a new set of input signals and parameters that are more likely to falsify the specification.

We aim with this paper to show how a rigorous method based on optimization can be used in the design process of autonomous vehicles and to give the reader an insight into how the quantitative semantics works for this particular problem.

The rest of the paper is organized as follows. Section II introduces the falsification of temporal properties. Section III introduces the adaptive cruise controller example. Section IV evaluates the performance of the optimization algorithm when using *MAX* and *MARV*. Finally, Section V summarizes the contributions.

II. Falsification

Falsification of temporal logical properties is based on an optimization procedure where the objective function is determined by the definition of a robustness semantics for the temporal logic formalism. Breach [8] and S-TaLiRo [9] are two tools implemented on top of Matlab/Simulink that can do falsification assuming that the closed-loop system can be simulated. In this work, Breach is used for the simulation and hence STL [10] is used to model the specifications, but the discussion in this paper can be applied to MITL used in S-TaLiRo as well.

A. Signal Temporal Logic

The syntax of STL is defined as follows [11]:

$$\varphi ::= \mu \,|\, \neg\mu \,|\, \varphi \wedge \psi \,|\, \varphi \vee \psi \,|\, \Box_{[a,b]}\psi \,|\, \Diamond_{[a,b]}\psi \,|\, \varphi\,\mathcal{U}_{[a,b]}\psi,$$

where the predicate μ is $\mu \equiv \mu(x) > 0$; φ and ψ are STL formulas; $\Box_{[a,b]}$ denotes the *globally* operator between a and b; $\Diamond_{[a,b]}$ denotes the *finally* operator between a and b; and $\mathcal{U}_{[a,b]}$ denotes the *until* operator between a and b. The semantics of STL are defined by considering the discrete signal x at time instant k [11]:

$$\begin{aligned}
(x,k) &\models \mu && \Leftrightarrow \quad \mu(x[k])\\
(x,k) &\models \neg\mu && \Leftrightarrow \quad \neg((x,k) \models \mu)\\
(x,k) &\models \varphi \wedge \psi && \Leftrightarrow \quad (x,k) \models \varphi \wedge (x,k) \models \psi\\
(x,k) &\models \varphi \vee \psi && \Leftrightarrow \quad (x,k) \models \varphi \vee (x,k) \models \psi\\
(x,k) &\models \Box_{[a,b]}\varphi && \Leftrightarrow \quad \forall k' \in [k+a, k+b], (x,k') \models \varphi\\
(x,k) &\models \Diamond_{[a,b]}\varphi && \Leftrightarrow \quad \exists k' \in [k+a, k+b], (x,k') \models \varphi\\
(x,k) &\models \varphi\; \mathcal{U}_{[a,b]}\psi && \Leftrightarrow \quad \exists k' \in [k+a, k+b] \quad (x,k') \models \psi\\
& && \qquad \wedge \forall k'' \in [k,k'], (x,k'') \models \varphi
\end{aligned}$$

Instead of just checking the Boolean satisfaction of an STL formula, the notion of a robust semantics is defined to measure how far away a specification is from being satisfied. In the next part, these robust semantics will be introduced.

B. Valued Booleans and MAX semantics

A VBool is a combination of a Boolean value together with a *robustness* value, a non-negative real number that indicates how true or false the VBool is. In [6], VBools are used to define two semantics aimed at measuring the robustness of STL formulas in a testing setting; *MAX* evaluated in this paper, and the *additive* semantics.

In this work, the robustness value will be used as a measure of how convincingly a test passed, or how severely it failed, respectively. The comparison operator $\leq_{\mathrm{v}}$ corresponds to $\leq$ and takes the difference between its arguments as its robustness. This is because, in order for the value of $x \leq y$ to change, one of the arguments has to change by at least $|x-y|$.

$$\leq_{\mathrm{v}} : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{V}$$

$$x \leq_{\mathrm{v}} y = \begin{cases} (\top, y-x) & \text{if } x \leq y\\ (\bot, x-y) & \text{otherwise,} \end{cases}$$

where $\top$ and $\bot$ denote true and false, respectively.

MAX is defined by the *MAX-and*, *MAX-or*, *MAX-always*, and *MAX-eventually* operators.

The *MAX-and* operator $\wedge_{MAX}$ is defined as:

$$\begin{aligned}
(\top,x) \wedge_{MAX} (\top,y) &= (\top, min(x,y)) \qquad (1)\\
(\top,x) \wedge_{MAX} (\bot,y) &= (\bot, y)\\
(\bot,x) \wedge_{MAX} (\top,y) &= (\bot, x)\\
(\bot,x) \wedge_{MAX} (\bot,y) &= (\bot, max(x,y)).
\end{aligned}$$

The *MAX-always* operator is defined over [a, b] as:

$$\Box_{MAX,[\mathrm{a,b}]}\,\varphi = \bigwedge_{k=a}^{b}{}_{MAX}\;\varphi[k], \qquad (2)$$

where φ is a finite sequence of VBools defined for all the discrete time instants in the interval [a, b].

Other operators, like *MAX-or*, *MAX-eventually* and *MAX-until* can be expressed in terms of the above operators. These operators are not used in this paper. For implementation of *MAX* in Breach, a VBool is represented as a single real value, where a negative value represents that the VBool is $\bot$, and a positive value represents that it is $\top$. In both cases, the magnitude of the real value is the robustness value.

C. Mean Alternative Robustness Value (MARV) semantics

In this paper we restrict the comparison to timed always operators since this is a common temporal operator. For *MARV* [7] the discrete-time always operator is described by the following formula.

$$\Box_{MARV,[a,b]}\varphi = \begin{cases} \frac{1}{b-a}\sum_{i=0}^{M-1}\rho(\varphi,t_i)\,(t_{i+1}-t_i) & \rho \geq 0\\ \rho(\varphi,t_i) & \rho < 0 \end{cases} \qquad (3)$$

where $\rho(\varphi,t_i)$ is a real-valued function that gives the objective value at each time instant, t_i, and M is the number of sampling times over the interval [a, b].

For positive robustness values, which represent the case when $\Box_{MARV,[a,b]}\varphi$ is satisifed *MARV* calculates the mean over the interval.

III. Use-Case

This study of the two semantics uses a simple example[1] involving two vehicles, an autonomous vehicle, called the *ego* vehicle, and a *lead* vehicle. The two vehicles travel on a road in the same direction (Fig. 1).

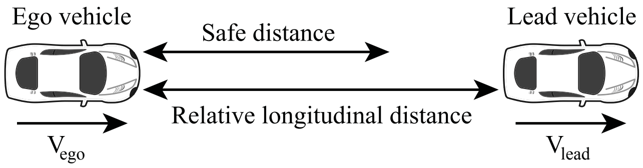

Fig. 1: Definition of the distances between the ego vehicle and the lead vehicle.

The ego vehicle is equipped with adaptive cruise control (ACC). It has a sensor that measures the relative distance to, and the relative velocity of, the lead car. The ACC system operates in two modes: *speed* mode and *safety* mode. In the speed mode, the ego vehicle travels at a driver-set speed, and in the safety mode, the ego car maintains a safe distance from the lead car. The ACC system decides which mode to use based on real-time radar measurements; either it needs to keep a safe distance, or if the relative distance is safe, it can increase its speed to the driver-set speed.

A. Specification of Safe Longitudinal Distance

As mentioned earlier, safety guarantees are important in autonomous driving. However, the safe distance between two vehicles depends on many parameters, including road friction, braking force, and response time. These parameters are largely unknown: while some of these parameters might be estimated by the ego vehicles, less is known about, for example the maximal braking force of the lead vehicle. Various studies, [12], [13], have addressed the issue of the minimum safe longitudinal distance between two vehicles. In our experimental setup the controller in the ACC system is designed to keep the safe distance d_{safe} between the two vehicles by calculating the set-point distance using the following formula:

$$d_{safe} = d_{default} + t_{gap}\, v_{ego}, \tag{4}$$

where $d_{default}$ is the standstill default spacing, in this case set to 10 m, and t_{gap} denotes the time gap between the vehicles which is set to 1.4 s. The distance d_{safe} is used as a reference value for the ACC controller, but it cannot be used as a specification because when the ego car is in the safe mode, any deceleration by the lead car will make the longitudinal distance less than d_{safe}.

Note, the formula above is used to generate the set-point distance between the two vehicles, but it is not suitable as a specification because having a shorter distance between the vehicles does not necessarily imply that a collision will occur. It is certainly possible to formulate a specification that models that the two vehicles do not collide by specifying that the distance between the two vehicles should be positive. However, falsification may be easier if we strengthen the specification by exploiting physical insight. In our example, we will calculate the *minimal* distance d_{min} between two vehicles using the approach in [13] based on physical properties, such that if the vehicles are never closer than d_{min} to each other, collisions are guaranteed to be avoidable when the lead car brakes:

$$d_{min} = \Bigg[v_{ego}\, t_r + \frac{1}{2} a_{max,acc}\, t_r^2 + \frac{(v_{ego} + a_{max,acc}\, t_r)^2}{2\, a_{min,brake}} - \frac{v_{lead}^2}{2\, a_{max,brake}} \Bigg]_+ , \tag{5}$$

where $[x]_+$ means the maximum of x and 0, and where the parameters are described in Table I.

The specification used for falsification now expresses that at all times the relative distance between the cars must be greater than the safe distance d_{min}. With T as the simulation time, this is formulated as:

$$\Box_{[0,T]}\big(relative\ longitudinal\ distance > d_{min}\big). \tag{6}$$

The falsification process is significantly easier when we specify that the relative distance between the vehicles must be at least d_{min}, rather than just positive. This strengthening of the specification is justified by, as soon as, the relative distance between the vehicles is less than d_{min}, immediate braking by the lead vehicle might result in a collision. Thus the specification amounts to assuming worst-case behavior from the lead vehicle.

IV. Evaluation Results

To evaluate the performance of the *MAX* and *MARV* semantics, falsification of the AD example is studied. The vehicle parameters used during the simulations of the closed-loop behavior are presented in Table I.

TABLE I: Parameters used in the example.

Parameters	Notations and Values
Velocity of lead car (m/s)	v_{lead}
Velocity of ego car (m/s)	v_{ego}
The driver-set velocity (m/s)	v_{set} = 30
Max acceleration of ego car (m/s^2)	$a_{max,acc}$ = 3
Min acceleration of ego car to full stop (m/s^2)	$a_{min,brake}$ = -2.5
Max acceleration of lead car to full stop (m/s^2)	$a_{max,brake}$ = -3
Response time (sec)	$t_r = 0.1$

The simulation takes the acceleration of the lead vehicle a_{lead} as input and simulates the behavior of the closed-loop system. Before a simulation of the closed-loop system starts, the values of input parameters are selected by the falsification algorithm; in this example, the parameters are a_{lead0} and a_{lead1}. The simulation time is $T = 30$ s and the simulation starts with a_{lead0} chosen in the range $[0, 3]$ and a_{lead1} in the range $[-3, 0]$. For both the *MAX* and *MARV* semantics the

[1] A demo example from the Matlab®/Simulink® toolbox.

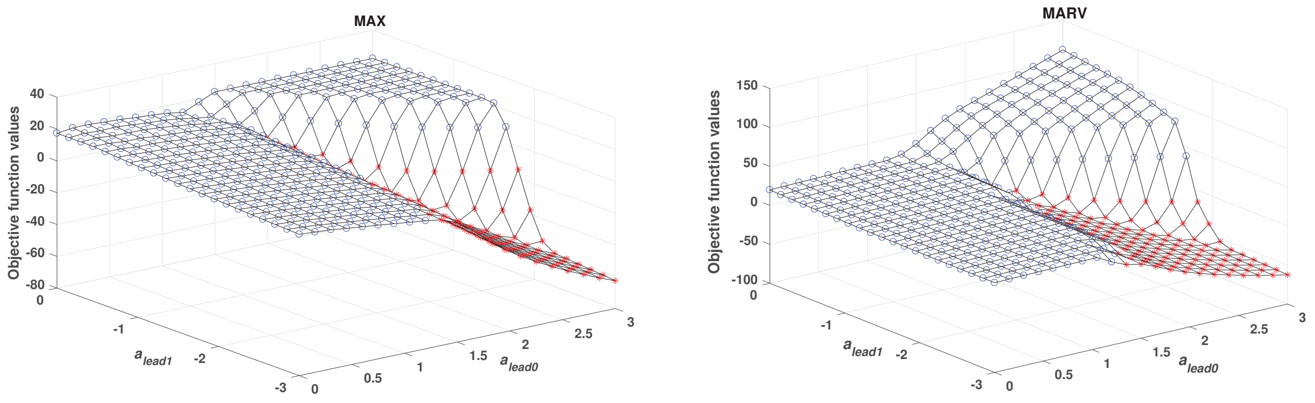

Fig. 2: The objective values for combinations of the accelerations (a_{lead0}, a_{lead1}) for the two semantics *MAX* and *MARV*. Positive values (○) mean that the specification is satisfied, while negative values ($*$) mean that it is falsified. Note that the signs of the values should be the same for both semantics while the absolute values might be different.

specification is given by formula (6). To illustrate the similarities and differences between the *MAX* and *MARV* semantics the objective value is calculated for different combinations of the parameters a_{lead0} and a_{lead1}. In this case a_{lead0} and a_{lead1} are divided into 20 equidistant points resulting in 400 parameter combinations, each requiring its own simulation.

The objective function values calculated for each of the 400 simulations for both *MAX* and *MARV* semantics are shown in Fig. 2. The main purpose of calculating an objective value is to guide the falsification process in the right direction by choosing the next set of parameters to be simulated such that the likelihood of falsifying the specification is increased. Note that in this work we do not assume that gradients can be derived analytically; instead, they have to be estimated by evaluating multiple parameter combinations. Ideally, a semantic should be such that when a parameter change brings the system closer to falsifying a specification, then the objective value of the specification should decrease.

By comparing the left and right graphs in Fig. 2 we notice that they have the same sign for every parameter combination. This is expected, since the sign indicates if the specification is fulfilled, which does not depend on which semantics is used. However, we observe that in the upper right corner (where close to $a_{lead0} = 3$ and $a_{lead1} = 0$) the two semantics result in different estimates of the gradients. In this region, for *MAX* the objective function values are the same for each point (the upper triangular side of the left graph). In order to illustrate why, Fig. 3 presents for each choice of a_{lead0} and a_{lead1} the first time at which the objective value reaches its minimum when using *MAX*. We observe that for parameters close to $a_{lead0} = 3$ and $a_{lead1} = 0$, the simulation time at which the minimal objective value is reached is time 0. The reason for this can be seen in Fig. 4, which shows the relative and safe distance, and the velocities of the lead and ego vehicles, when $a_{lead0} = 3$ and $a_{lead1} = 0$. According to this figure, when the lead car continuously accelerates, the ego car increases its speed, too. But the ego car has a driver-set velocity limit (30 m/s) so that the relative distance between the vehicles always increases and the *minimum* objective function value occurs at the beginning of the simulation where the relative and safe distances are the closest. Thus, the *MAX* semantics consider the different simulations to be equally good/bad for parameters that are close to $a_{lead0} = 3$ and $a_{lead1} = 0$, resulting in no information that can be used by the optimization algorithm. On the other hand, as can been seen in the right graph of Fig. 2, these points have different values under *MARV*.

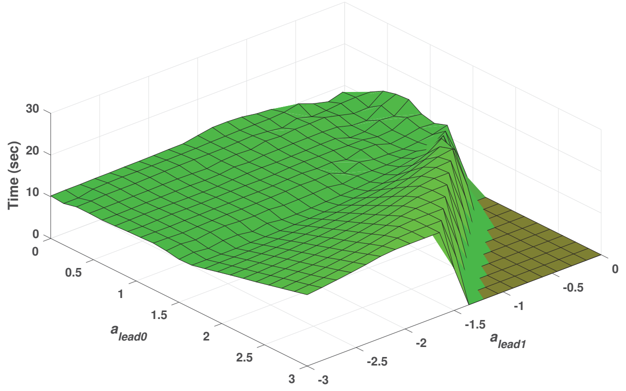

Fig. 3: The simulation time when the minimum objective function value is reached for the first time for *MAX*. Note that a_{lead1} is here to the right.

In Fig. 2, where acceleration of the lead vehicle a_{lead0} is in the range $[0, 1.2]$, and a_{lead1} in the range $[-3, 0]$, for all parameter combinations in these ranges, the ego vehicle behaves safely and keeps the safe distance from the lead vehicle. In order to show the behaviour of both vehicles in these ranges, the relative and safe distances and the velocities of the ego car v_{ego} and the lead car v_{lead} are shown for $a_{lead0} = 0$ and $a_{lead1} = -3$ in Fig. 5. As can be seen, the relative distance is greater than d_{min} for the whole of the simulation, it means when the lead car brakes the ego car starts braking too, and it can stop within safe distance from the lead car.

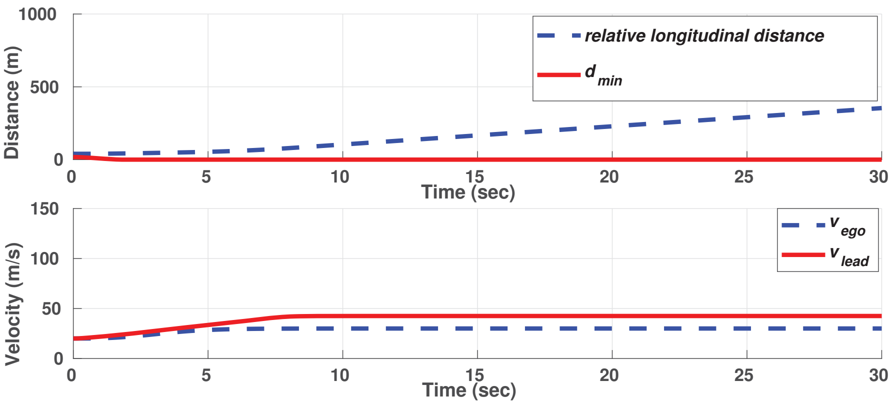

Fig. 4: The relative and safe distance, and the vehicle velocities for $a_{lead0} = 3$ and $a_{lead1} = 0$. The minimum objective function value occurs at time zero, where the relative and safe distances are the closest.

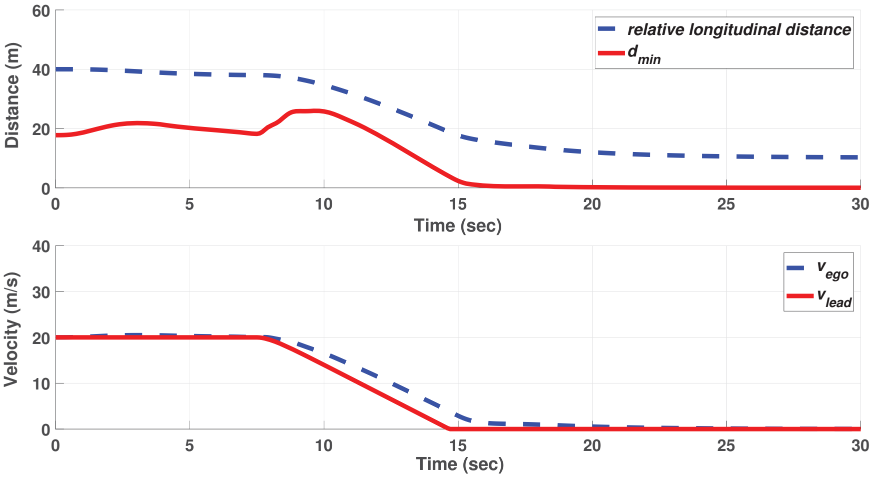

Fig. 5: The relative and safe distance, and the vehicle velocities for $a_{lead0} = 0$ and $a_{lead1} = -3$.

In Fig. 2, the safety formula (6) is violated where the objective values are negative. The relative and safe distances, and the velocities of both cars for the point $a_{lead0} = 3$ and $a_{lead1} = -3$, are shown in Fig. 6. As can be seen, when the lead car accelerates, the ego car increases its velocity to reach the driver-set velocity. Then, when the lead car brakes, because their distance is larger than d_{safe} (4), the controller is in speed mode and only after a delay does the ego car switch to safety mode and adjust its speed to maintain a safe distance from the lead car. As a result, not only does the relative distance between the vehicles become less than d_{min}, but the cars even crash. Note that while the cars crash at around 22 s in this scenario, in Fig. 6 the ego car does not stop until around 26 s. This is due to the simple model used in the example that does not model the actual collision.

By comparing the objective values from the two semantics *MAX* and *MARV*, we observe that since the *MAX-always* only considers the minimal value of the objective function it is possible to end up with objective values that do not differ between different simulations. Thus, the optimizer has no information in which direction to further explore the search. In this example we observed that in this case it might be beneficial to consider *MARV* since this semantic will result in higher objective values when the vehicles move further away from each other, thus providing information to the optimization algorithm that might guide the optimizer in the right direction to falsify the specification.

In this work, we have only considered the objective functions used by the optimization algorithm but not the optimization algorithms themselves. However, gradient-free optimization algorithms like Nelder-Mead or simulated annealing are typically used in the falsification process, and it is clear that the performance of these algorithms depends on having objective values that are not constant and that direct the search in a direction where the objective values decrease, increasing the chances of falsifying the specification.

V. Conclusion

In this paper, we showed how the efficiency of falsification might be affected by the semantics used to evaluate the specification. An adaptive cruise controller is used as an example and

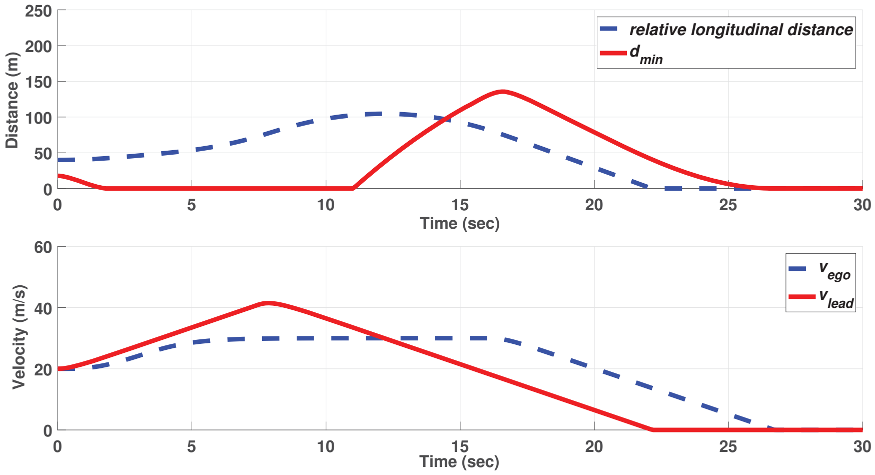

Fig. 6: The relative and safe distance, and the vehicle velocities for $a_{lead0} = 3$ and $a_{lead1} = -3$. At around 14 s the relative distance between the vehicles becomes less than d_{min}, and a collision occurs at around 22 s.

the objective is to test if the certain parameter combinations result in the possibility for two vehicles to collide. Using the example, we showed a situation where the *MAX* semantics will result in the same objective value for closely related parameter values, while *MARV* results in a non-constant objective value, which might guide the optimization algorithm in a direction that increases the chances of falsifying the specification. The objective of this paper was not to compare the efficiency of *MAX* and *MARV*, but rather to illustrate with an example of the importance of choosing a suitable semantics for the problem at hand. From our experience with industrial-scale systems we have observed that the simulation time is the most limiting factor, not the evaluation of the simulation results using different semantics. Thus, a future strategy might be to evaluate the simulation using multiple objective functions and use a high-level algorithm that during the optimization will take into account multiple objective values computed using several different semantics.

ACKNOWLEDGEMENTS

This work was supported by the Swedish Research Council (VR) project SyTeC VR 2016-06204 and from the Swedish Governmental Agency for Innovation Systems (VINNOVA) under project TESTRON 2015-04893.

REFERENCES

[1] P. Koopman and M. Wagner, “Autonomous vehicle safety: An interdisciplinary challenge,” *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 1, pp. 90–96, Spring 2017.

[2] T. A. Henzinger, P. W. Kopke, A. Puri, and P. Varaiya, “What’s decidable about hybrid automata?” *Journal of Computer and System Sciences*, vol. 57, no. 1, pp. 94 – 124, 1998. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0022000098915811

[3] E. Bartocci, J. Deshmukh, A. Donzé, G. Fainekos, O. Maler, D. Nickovic, and S. Sankaranarayanan, “Specification-based monitoring of cyber-physical systems: A survey on theory, tools and applications,” in *Lectures on Runtime Verification*, ser. Lecture Notes in Computer Science, Feb 2018, vol. 10457, pp. 135–175.

[4] R. Koymans, “Specifying real-time properties with metric temporal logic,” *Real-Time Systems*, vol. 2, no. 4, pp. 255–299, Nov 1990. [Online]. Available: https://doi.org/10.1007/BF01995674.

[5] O. Maler and D. Nickovic, “Monitoring temporal properties of continuous signals,” in *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems*, Y. Lakhnech and S. Yovine, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 152–166.

[6] K. Claessen, N. Smallbone, J. Eddeland, Z. Ramezani, and K. Åkesson, “Using valued booleans to find simpler counterexamples in random testing of cyber-physical systems,” *IFAC-PapersOnLine*, vol. 51, no. 7, pp. 408 – 415, 2018, 14th IFAC Workshop on Discrete Event Systems WODES 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2405896318306633

[7] J. Eddeland, S. Miremadi, M. Fabian, and K. Åkesson, “Objective functions for falsification of signal temporal logic properties in cyber-physical systems,” in *2017 13th IEEE Conference on Automation Science and Engineering (CASE)*, Aug 2017, pp. 1326–1331.

[8] A. Donzé, “Breach, a toolbox for verification and parameter synthesis of hybrid systems,” in *Computer Aided Verification*, T. Touili, B. Cook, and P. Jackson, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 167–170.

[9] Y. Annpureddy, C. Liu, G. Fainekos, and S. Sankaranarayanan, “S-taliro: A tool for temporal logic falsification for hybrid systems,” in *Tools and Algorithms for the Construction and Analysis of Systems*, P. A. Abdulla and K. R. M. Leino, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 254–257.

[10] A. Donzé and O. Maler, “Robust satisfaction of temporal logic over real-valued signals,” in *Formal Modeling and Analysis of Timed Systems*, K. Chatterjee and T. A. Henzinger, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 92–106.

[11] V. Raman, A. Donz, M. Maasoumy, R. M. Murray, A. Sangiovanni-Vincentelli, and S. A. Seshia, “Model predictive control with signal temporal logic specifications,” in *53rd IEEE Conference on Decision and Control*, Dec 2014, pp. 81–87.

[12] G. Feng, W. Wang, J. Feng, H. Tan, and F. Li, “Modelling and simulation for safe following distance based on vehicle braking process,” in *2010 IEEE 7th International Conference on E-Business Engineering*, Nov 2010, pp. 385–388.

[13] S. Shalev-Shwartz, S. Shammah, and A. Shashua, “On a formal model of safe and scalable self-driving cars,” *CoRR*, vol. abs/1708.06374, 2017. [Online]. Available: https://arxiv.org/pdf/1708.06374.pdf.