View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by CiteSeerX

Estimating the Probability of a Timely Traffic-Hazard Warning via Simulation

Nils Müllner, Martin Fränzle, Sibylle Fröschle
Carl von Ossietzky Universität Oldenburg
and OFFIS Institute for Computer Science
nils.muellner | fraenzle | froeschle @informatik.uni-oldenburg.de

Keywords: traffic simulation, hazard and analysis, timely warning, safety
ACM Classification Keywords: C.2.1, C.2.4, C.3, C.4, G.1.6, I.6

Abstract

Traffic flow simulation is exploited for estimating the probability that a message — a hazard warning in this case — is correctly transmitted to an approaching car in time, that is, before overstepping a safety threshold. The results derived by simulation provide valuable insights in the functional relation between the numerous authoritative parameters and the reliability of timely message reception.

INTRODUCTION

The automotive domain provides new innovations like no other, be it autonomous driving or braking systems that can cope with sustained aquaplaning. This paper focuses on the benefits and perils of decentralized vehicle-to-vehicle communication for hazard warning. The timely delivery of such crucial information is a safety goal. In this context, the timely reception of the warning marks the time point when the safety goal is reached. Before that, the system is unsafe.

The goal is to identify the relevant basic input and output parameters for such a setting. The radio coverage of the cars between the ego car — the car under observation for which the probability for a timely warning is measured — and the hazard, the velocities of cars and the distribution of cars between ego car and hazard are obvious input parameters. For the output parameters on the other hand we are not only interested in the probability that the warning is successfully delivered, but also in the standard deviation. For instance consider a certain fixed scenario (i.e. distance to hazard, probability distribution over positions of other cars, probability distribution over velocities and radio coverage of other cars). It is not just interesting how well a warning is propagated, but also how much the results fluctuate, how *reliable* they are.

Simulation allows to generate results in order to reason about reasonable safety margins. We discuss three basic questions: 1) How many trials are required to acquire resilient results? 2) How is the standard deviation of *stable* results influenced by the number of cars deployed between ego car and hazard? And 3) how well do results increase/decrease when the radio coverage of all cars is increased/decreased?

ANSS 2015 April 13-16, 2015, Alexandria, VA
Copyright © 2015 Society for Modeling & Simulation International (SCS)

The formal setting underlying the simulation is explained in Section *Case Study* and selected related work is discussed in Section *Related Work*. The simulation framework conducting the scenario and simplifications as well as limitations are introduced in Section *Simulation Framework*, pointing out all the relevant scenario parameters to tune the simulation and its performance. The results of three distinguished scenarios targeting the questions above are discussed in Section *Results* and interpreted in Section *Interpretation*. Directions for future research conclude our work in Section *Conclusion*.

CASE STUDY

Consider a (green) car approaching a (red) hazard as shown in Figure 1. The car, called *ego-car*, is located at point s_0 with a hazard being 100 units in front of it on a straight route. Fog continuously prevents the ego-car from visually assessing the threat posed by the hazard. The ego-car approaches the hazard with a velocity of $v_{ego} = 1$ unit per time-step. For sake of generality the simulation framework avoids units here. It applies discrete time steps of length 1. The ego-car reaches the hazard after 100 time steps in case it does not receive a warning and thus initiates breaking.

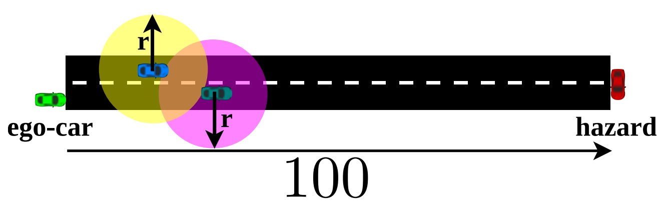

Figure 1. Scenario setting

The ego-car is equipped with an antenna and has a communication radius[1] of $r = 5$ length units. We consider one standard antenna with the same radius for each car. This radius determines the minimal amount of cars that is required to establish a momentary communication path from the hazard towards the ego-car to provide a warning. The common format for warnings is a *decentralized environmental notification message* as discussed in the next section. Such messages are continually sent. Since we employ a discrete time

[1]The figure is not true to scale.

model, it fits to consider such messages being sent at intervals of one time step. For simplicity we further consider cars to only have memory for processing current data. They cannot store information and provide it later. An instantaneous bridge relay between hazard and ego-car is hence required to communicate a warning.

The minimal number of cars required to establish such a momentous transitive link in this setting is computed with min_cars$= 100/5 = 20$ for the provided parameters. With a considered length of 2 length units per car and two lanes, the upper boundary — which is the maximal number of cars — is set to max_cars$= 100$.

Instead of drawing the probabilities to select the distribution of non-ego-cars once and letting them succeed with a constant velocity like the ego-car, the non-ego-cars have a randomly drawn velocity with a normal distribution over interval [v_min, v_max]. The velocity is selected randomly for each non-ego-car at each time step individually.

Approximate Traffic model This paragraph briefly reasons about the benefits of employing an unrealistic traffic model. In the featured approach the focus is on determining the relevant parameters. Employing normal distributions for each influence (i.e. initial positioning of non-ego cars, velocity of non-ego cars) and independence of all contributing factors (cars do not influence each other) ensures normal distribution of the results. Although this might not be realistic as cars can for instance exhibit *stutter* behavior and backlog of breaking maneuvers is not accounted for, normal distribution of results ensures lets us confirm that the simulation works as expected to this end. Section *Future Work* picks up on this strain and proposes directions to enhance the presented work towards realistic traffic simulation.

One might argue that it is probable with the dynamic velocity of non-ego cars, that even one single car that continuously drives slower than the ego car can communicate a timely warning when it discovers the hazard. Yet, as the results will show, that probability sufficiently small to be negligible. It is reasonable to set the lower boundary to a realistic number of 20 cars here.

Each number of cars within interval [min_cars,max_cars] $= [20, 100]$ is simulated numerous times to determine *how well* the warning is propagated *in average* for each specific possible number of non-ego cars deployed between the ego car and the hazard. The warning is delivered correctly to the ego-car, thus satisfying the safety predicate, if the ego-car has not reached the hazard yet and the maximal distance between each two following cars between hazard and ego-car does not exceed $r = 5m$. Safety is ultimately violated for the ego-car iff it reaches the safety threshold in front of the hazard without a warning.

In terms of fault tolerance terminology the predicate focuses on *instantaneous window reliability* [12], which is the probability that an unsafe region (undelivered hazard warning) is left within a restricted time window, or else the system will fail once and forever. The attribute *instantaneous* derives from the definition of instantaneous availability, which is the probability of a system to be *safe* after finitely many time steps [15] based on a distinct initial configuration. The following section concentrates related work and Section *Simulation Framework* then goes into detail, refining three distinguished scenarios that can be analyzed with this case study simulation.

RELATED WORK

Common temporary road hazards are street segments that are slippery due to ice or aquaplaning or oil spill, crosswinds, limited sight due to dust storms or fog, fords (flooded segments) possibly caused by clogged storm drains, missing manhole covers, collisions, moving oversized loads like road trains, pot- and sinkholes possibly caused by washout, roadkill and rockfall. This paper focuses on *stationary* hazards to limit the number of free parameters for now. Such hazard scenarios are for instance discussed for the automotive domain in the ETSI standard 102638 [7] which categorizes various of such settings.

While Lamport provides a general definition for safety [10], the automotive industry utilizes an adapted version, commonly referred to as *functional safety*, as standardized by the IEC61508 [14] from 1998 and its successor, the ISO26262 [8] from 2011. In our case, safety simply holds when the ego car receives a timely hazard warning, thus specifying safety simply and precise for our distinct scenario.

The standard distinguish between *Cooperative Awareness Messages* (CAMs) and *Decentralized Environmental Notification Messages* (DENMs). According to this classification our scenarios focus on DENMs relaying.

The selected fault tolerance measure quantifying safety in this context is *instantaneous window reliability* (IWR) described in [12]. While the optimal solution providing precise results would rely on model checking as for instance introduced by Baier and Katoen [1], the scenario featured here contains too many free parameters (let alone traffic evolution. Hence, simulation, as for instance introduced by Jain [9], provides the necessary means to learn about interaction of input parameters and how they influence goal functionalities. Learning about the parameters and their interaction is required to contemplate about *optimal adjustment* and *tuning* of the system, either focusing on decreasing the cost or increasing the functionality (i.e. safety in this case), as for instance discussed by Deb [6].

There are numerous excellent publications on diverse adjacent topics indicating its importance. This paragraph briefly presents five selected of those publications. Focusing on network related parameters, Biswas et al. [3] investigate the importance of packet error rate and latency. The setting presented in the present paper can be easily adapted to account for environmental influence with regard to probabilistic disturbances. Yet, to maintain the focus on the three main questions proposed in the next section, environmental probabilistic parameters are not regarded at this point. Similarly, Briesemeister et al. [4] look into the physical background of the radio communication to determine the necessary percentage of vehicles to be equipped with radio for a successful warning system, yet without accounting for traffic dynamics. Chen and Cai [5] on the other hand compare different vehicle grouping strategies — referred to as architectures — and their efficiency for providing a timely hazard warning. The traffic simulation featured in our paper is also not realistic in that regard— as discussed in Section *Future Work* — as deliberate car clustering is not considered. Resta et al. [13] contribute a theoretical analysis backed up by results from simulation. Since both are fed by the same parameters the results mostly coincide nicely. Their main scope is the trade-off between *safety level* and *emergency message resource wastage*. In other words, reliability of one-hop communication can be increased by redundancy (i.e. multiple repetition) at the cost of bandwidth for non-safety related communication and the proposed approach aids in finding a good trade-off. Although missing that the numerical values in the analysis stem from real experiments, the paper shows that an analytic approach for a limited number of parameters is very well possible. Similarly, Xu et al. [16] inspect layer-2 protocols to determine the impact of parameters like communication (broadcast, local, geo-significant) and mobile network (ad-hoc, highly mobile, large number of contending nodes). Their work provides a valuable background for modeling the physical-technical layer to be integrated in our proposed approach at a later stage.

For future work, the Sumo simulation framework [2] seems promising for covering the required background for realistic traffic flow simulation in replacement of the simplified traffic simulation featured here. For reasons discussed above our current work exploits some approximations and Sumo would only slow down the simulation.

SIMULATION FRAMEWORK

There are three major questions the simulation shall answer:

- How does the simulation scale?
- How does the velocity interval of non-ego cars influence results?
- How does the radio radius influence results?

Before discussing these questions in detail in Sections *Simulation Scaling*, *Interval Impact* and *Radio Radius*, the remainder of this introduction is dedicated to technical details of the simulation. As discussed in Section *Case Study*, the simulation deploys between 20 to 100 cars between the ego car and the hazard that is 100 length units away from it. Each specific number of cars is referred to as *scenario*. Each scenario is simulated multiple times. Each simulation run a referred to as *trial*. Hence, there are 81 scenarios that are simulated for a certain amount of trials each. The simulation is coded in MatLab and provided[2] online. The simulation schematics are depicted as UML diagram in Figure2. All relevant parameters are set in the file header.

There are two main for-loops: counting the number of cars deployed between ego-car and hazard, and the other counting the number of trials per number of cars embedded. In case the current trial is not complete (neither was the hazard reached nor the warning successfully transmitted to the ego-car), the traffic field progresses one time step. Otherwise, the result is recorded and a new trial begins. This continues until all trials for all numbers of non-ego cars are completed.

The main file is `probtrafjamd.m` which first initializes the scenario and then traverses through each trial. In the main file, the parameter `num_trials` determines the number of trials for each scenario. Reasonable values are within $[10, 100000]$. Further important parameters are

- position s_0, determining the distance between the ego-car and the hazard,
- car length `len_car`, limiting the number of cars fitting at most in the space[3] between ego car and hazard,
- radio radius r, setting the lower boundary of cars that are at least necessary to establish a radio bridge between the hazard and ego-car, and
- velocity interval [`v_min`, `v_max`] from which the velocity of each non-ego-car at each time step is selected.

The minimal number of cars computes with `min_cars =` `floor(`s_0/r`);` to 20. The upper boundary is determined by the car length, which resolves — with an average considered car length of 2 length units divided by number of lanes — to at most 100 cars. The parameter `v_ego` is the velocity of the ego-car and the parameters `v_min` and `v_max` provide the velocity spectrum for all other cars.

The initialization generates an initial driver field between the ego-car and the hazard, one for each trial. As long as the ego-car has not passed the hazard and no radio bridge was successfully established yet, the velocity is randomly selected

[2] `http://www.mue-tech.com/software/AnSS15.zip`

[3] To be precise, the number of lanes is also required. To minimize the number of parameters we implicitly consider two lanes by default.

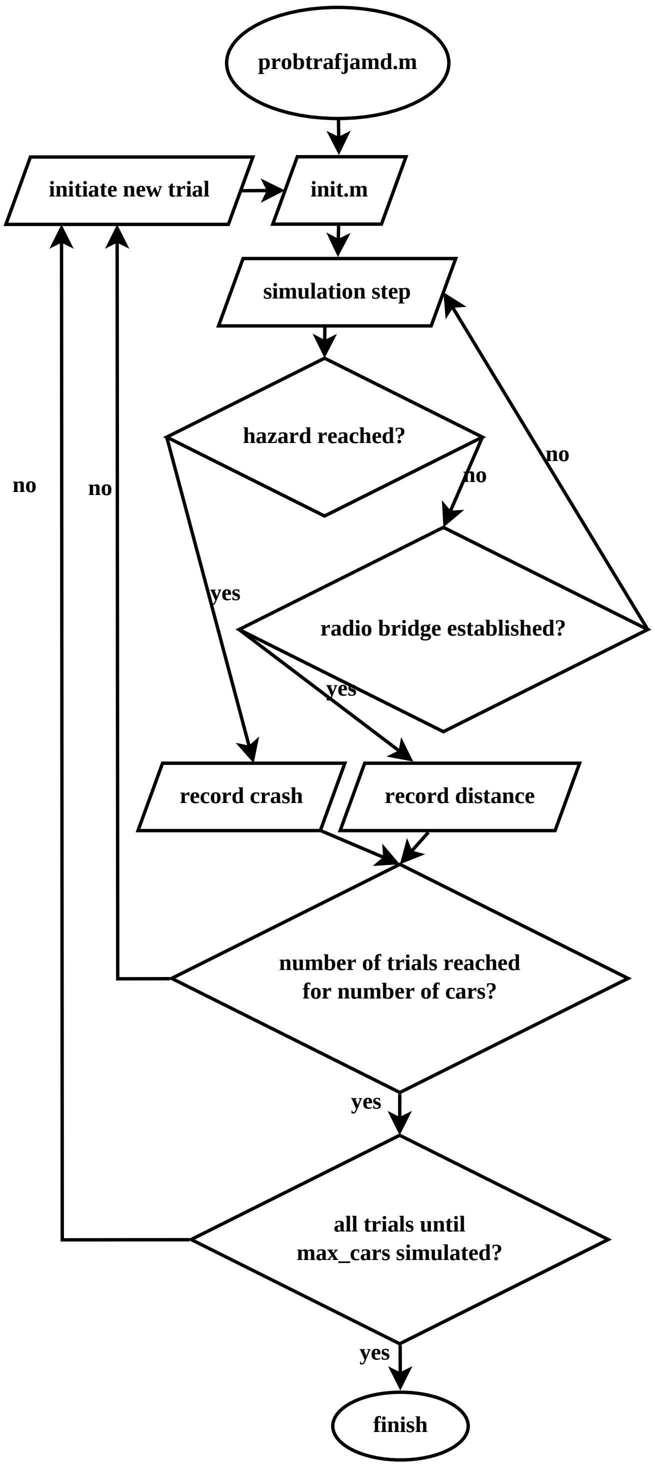

Figure 2. UML Diagram of the Simulator

for each car from the interval `[v_min,v_max]`. Then. the driver field is updated and the ego-car proceeds with `v_ego`. Once the ego-car reaches the hazard, the safety goal on that trial is ultimately failed. Otherwise, in case of a radio bridge being successfully established, the position of the ego-car is recorded. Since the required breaking distance depends on external factors like weather the simulation blindly measures the distance to the hazard upon the first successful notification. If that distance suffices can later be filtered if desired.

The framework is designed for performance. Conducting 100000 trials for each of $[20, 100]$ numbers of cars can be recorded with 8GB of RAM. Furthermore, the code is designed for easy adaptation, for instance to determine, how various parameters influence the results.

Some simplifications The first minor simplification is that the vertical differential between cars from different lanes is neglected. This means that two subsequent cars on different lanes are treated as if they were on the same lane regarding their radio radius. The second simplification is discussed in Sections *Case Study* and *Future Work* and targets the absence of interaction between cars.

Simulation Scaling

When simulating to determine the impact of distinct parameters on the result to derive a functional dependency among them, it is important to determine how many trials are required. The first two goals for answering this question are the *probability of successful warning* (regardless of the distance of the car towards the hazard) and the *mean distance to the hazard* for cases of successful warning. Since the probability for a successful warning is expected to correlate with the number of cars between ego car and hazard, we expect that more trials are required for lower number of cars between ego car and hazard.

The setting is first fixed in all parameters and the number of trials is subsequently set to 10, 1000 and 100000.

Interval Impact

In the first scenario, the non-ego cars' speed is drawn from a fixed velocity interval $[0.5, 1.5]$ with a uniform probability distribution. We are interested how the standard deviation for the probability of successful warning correlates when this interval is set to $[0.7, 1.3]$ and $[0.9, 1.1]$.

One question delayed in the previous setting was the standard deviation: How much do the stopping distances fluctuate around the average value for each scenario? Does the width of the velocity interval really matter, and if so, how much?

Radio Radius

Obviously increasing the radio radius will improve the results. With a radius of 100 length units one car in front of the ego car could suffice to send a timely warning. But how does increasing (or even decreasing) the radius scale with the results? Tuning the radio radius as second parameter provides an answer and allows to develop a cost function.

RESULTS

The simulation is conducted as discussed in the previous section.

First Setting: Simulation Scaling

Simulating 20 to 100 cars with 100000 trials each with the provided source code takes about 9.64 hours on an Intel® Core™ i5-3317U CPU at 1.7 GHz and 8GB DDR3 SODIMM in MatLab. Reducing the number of trials scales linearly. Executing 10 trials takes about four seconds while executing 1000 trials takes about seven minutes. We compare sequences of 10, 1000 and 100000 trials to determine the number of trials that is necessary to produce *smooth* resilient plots.

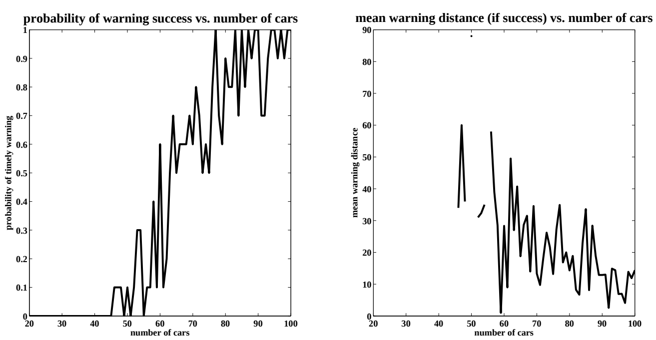

Figure 3. Ten trials per scenario

As shown in Figure 3, ten trials are insufficient. The left hand figure shows the overall probability that the ego car receives a warning before hitting the hazard, while the right hand side shows the average distance to the hazard at the time the warning is first received. Next, the number of trials is increased by a factor of 100.

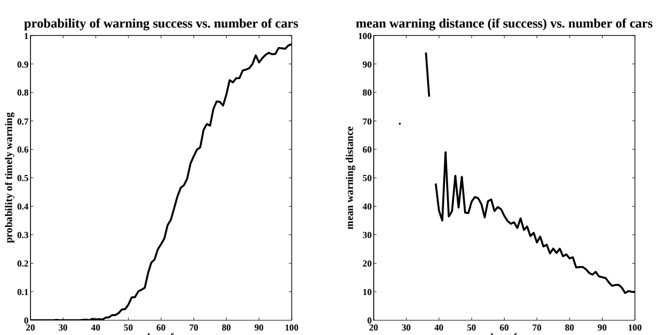

Figure 4. $1,000$ trials per scenario

Conducting 1000 trials per scenario increases the quality of the plots as expected as shown in Figure 4. Yet, the plot on the left hand side is still not strictly monotonic.

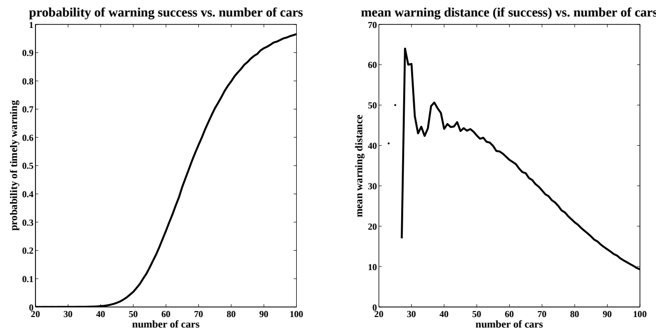

Figure 5. 100000 trials per scenario

Increasing the number of trials again by a factor of 100 helps in that regard. The left hand plot in Figure 5 shows how the probability for successful timely delivery of the hazard warning increases with the number of cars deployed initially between hazard and ego-car now with strict monotonicity. While the probability for up to about 40 cars is zero or close to zero, the probability monotonically increases with being cars added.

The right hand side of Figure 5 shows the mean distance between the ego-car and the hazard when the ego-car receives the warning. While fluctuations until up to about 60 cars between the ego-car and the hazard are obvious, the graph becomes smooth for larger numbers of cars between 60 to 100. This *increased smoothing* is caused by the th fact that only *successful trials* are accounted for to compute the standard deviation. MatLab here uses the `nanstd()` function instead of `std()`. While the plots on the right hand side in all Figures 3 to 5 suffer from this effect, it is more sever for lower numbers of trials. The results are interpreted in Section *Interpretation*.

Second Setting: Tweaking the Velocity

The velocity of the non-ego-cars is randomly selected per time step from an interval limiting the minimal and maximal velocity. The probability is uniformly distributed within each interval. In the standard scenario previously discussed in Section *Simulation Scaling* the interval was set to $[0.5, 1.5]$ with v_{ego} set to the interval's center 1. This second setting conducts the experiment for intervals $[0.7, 1.3]$ and $[0.9, 1.1]$, too. Since the non-ego cars' velocity is drawn from intervals with equally distributed probability around the same center, it is expected that the graph should be similar for all intervals. Since 100000 showed to be a reasonable number of trials in the previous setting, the same amount is selected again for this scenario.

Figure 6 shows that the swaying stabilizes in all three

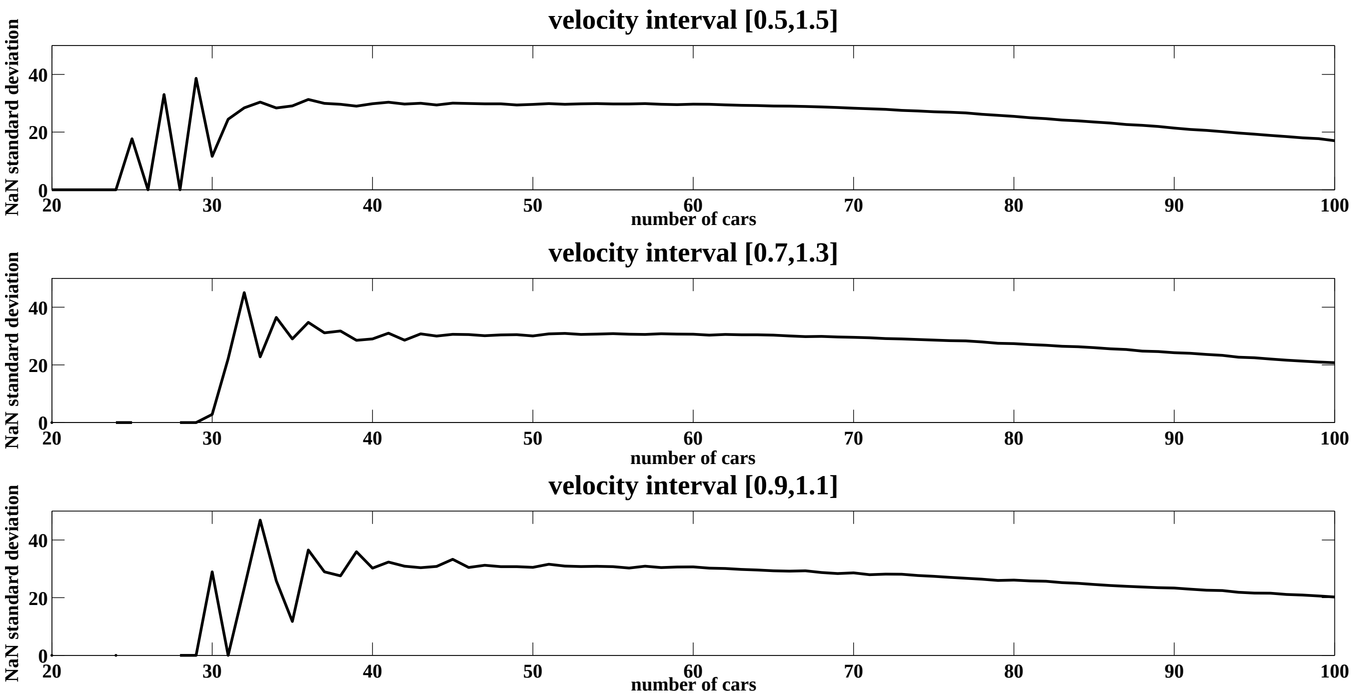

Figure 6. Standard deviation of warning distance for different velocity intervals

graphs when the number of cars is increased. It stabilizes faster for broader intervals and continues until about 35 for interval $[0.5, 1.5]$, to about 40 for interval $[0.7, 1.3]$ and until about 45 for interval $[0.9, 1.1]$.

Third Setting: Tweaking the Radius

The radio radius correlates with the probability for a successful timely warning: The higher the radius the higher the probability. How this parameter influences the result in relation to the amount of cars deployed between the ego car and the hazard is shown in Figure 7. The radii tested are 3, 4, 5, 6 and 7. Notably, the time required for the simulation also depends on the parameter. While the standard scenario took about 9.64 hours, a radius of 3 requires about 13.4 hours and a radius of 4 takes about 13 hours. This increase in simulation time is caused by the longer trials. Since the warning probability is lower, trials are simulated longer. The adverse effect happens with greater values for the radius.

INTERPRETATION

The three scenarios produced in the previous section target different spots: The first scenario from Section *Simulation Scaling* has the sole purpose of discussing the working method of the simulator and to point out that the quality of results relies on the number of trials. The second scenario from Section *Second Setting* shows how the standard deviation of the mean warning distance sways in relation to the number of cars. The third scenario tested different values for the radius to determine their influence in Section *Third Setting* . This section now interprets the results.

First Setting: Simulation Scaling

The first setting presented in Section *Simulation Scaling* shows simply that the number of (successful) trials increases the quality of the results. While the left hand graphs in Figures 3, 4 and 5 show how the overall probability for successful warning is refined is refined via additional trials, the right hand side graphs point out that fewer non-ego cars means less successful trials. Thus, computing the standard deviation does — contrary to probability of successful warning — not *only* rely on the number of trials, but also on the number of cars. One opportune filter might be a minimum bounding box that checks the maximal difference in the past n results and simply continues generating trials as long as the results exceed the minimum bounding box as for instance presented in [11].

The graphical representation provided only serves to demonstrate how further trials contribute to smooth the plots. A numerical analysis is skipped here as it would only reflect the unsteadiness of one distinct simulation set. It would however not contribute to the argument that there is not only a least required number of trials, but rather a least number of *successful* trials to achieve smooth plots (i.e. *good* results) for low numbers of cars.

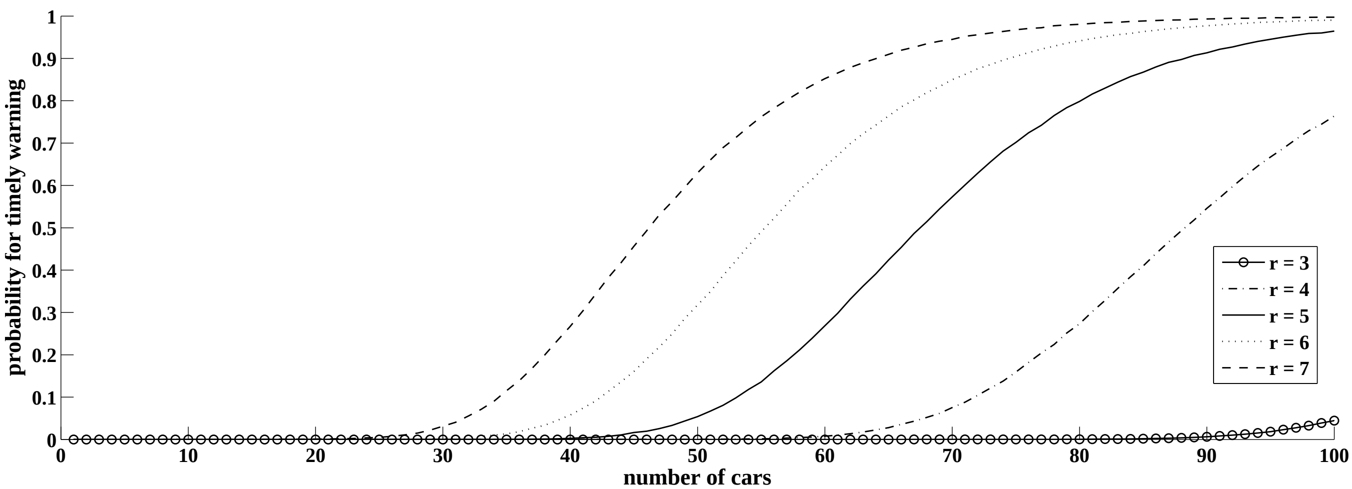

Figure 7. Determining the influence of the radius on the result

Second Setting: Tweaking the Velocity

The second scenario introduced in Section *Second Setting* showed how the standard deviation of results relies on the number of non-ego cars deployed. By tweaking the velocity interval length, we are interested in how much the single trials sway around the mean values. Since the number of *successful trials* depends on the *number of cars* as determined in the previous setting, and the latter type of graphs from the previous section showed how stability of results (mean warning distance) correlates with the number of cars, we assume a similar effect for narrowing the velocity interval.

While narrower margins in the velocity dynamics make the outcome of a single trial more reliant on the initial configuration, the sheer number of trials can compensate. This means, the narrower the velocity margin, the *coarser* the graph; the higher the number of trials, the smoother the graph. While all velocity ranges should generate similar graphs, their *smoothness* is expected to be better for larger intervals.

Third Setting: Tweaking the Radius

Finally, the third scenario presented in Section *Third Setting* provides some interesting results. The graph's shape is similar to the Fermi distribution function with some dampening for higher number of cars. Another interesting point is that linearly increasing the radius seems to clinch the graph logarithmically to the left. With such knowledge it is fairly easy to validate desired safety goals, for instance to compute the minimal required radius for a certain number of cars to achieve a desired probability for a timely warning. On the contrary, it is possible to derive the probability safety holds with provided parameters for radius and cars.

With the characteristic form of the graph it seems also a near goal to specify a function to replace the simulation. This can be achieved via curve fitting as discussed in the following section.

CONCLUSION

This section concludes our work and provides an outlook on promising directions for future research.

Summary

This papers introduced a slim simulation framework coded in MatLab for a specific scenario. The scenario features an ego-car with zero sight approaching a hazard. Vehicle-to-vehicle communication can be exploited by non-ego cars between ego-car and hazard to relay a hazard warning. The simulator gives indication about the probability that a warning is received before the ego-car hits the hazard.

The paper featured three scenarios for the simulation: simulation scaling, impact of velocity interval of non-ego cars, and impact of radio radius. These provided insights into determining relevant parameters, into determining their functional interaction, and their impact on goal functions.

Future Work

We propose five opportunities for future work in this field. The first one is simple and discussed in Section *Simulation Scaling*. For some goal functions a fixed number of trials is not advisable (e.g. for scenarios shown in right hand side Figures 3 to 5). An evaluation during run-time like the proposed minimum bounding box seems like a good solution. Further scenario adaptations that might reveal interesting results — and further parameters to tune — are dynamic hazards and dynamic ego-car velocity, as discussed in Section *Related Work*.

Two major alterations to the proposed simulation are i) realistic traffic simulation and ii) radio relays with memorizing cars. The first one can be achieved by interfacing the simulator with existing software like Sumo. The latter would require some adaptations within the provided MatLab code.

Despite future work on the simulator, one major goal is to determine functional dependencies among input parameters to determine the function computing the result directly. This will be achieved with curve fitting methods, if possible already accounting for the extensions proposed above. Finally, the dynamic within the models (e.g. realistic traffic) will cause some backlog and thus abolish uniform distributions for goal functionalities like safety. Then, *sweet spots* for possible attackers will add a security goal. Attackers might for instance place radio frequency jammers prevent relay of safety information like hazard warnings. By regarding all relevant input parameters one can find the Achilles heels to relay safety warnings and only then be able to discuss counter measures.

REFERENCES

1. Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking (Representation and Mind Series)*. The MIT Press, 2008.

2. Michael Behrisch, Laura Bieker, Jakob Erdmann, and Daniel Krajzewicz. SUMO - Simulation of Urban MObility: An Overview. In *SIMUL 2011, The Third International Conference on Advances in System Simulation*, pages 63–68, Barcelona, Spain, October 2011.

3. Subir Biswas., Raymond Tatchikou, and Francois Dion. Vehicle-to-vehicle wireless communication protocols for enhancing highway traffic safety. *Communications Magazine, IEEE*, 44(1):74–82, Jan 2006.

4. Linda Briesemeister, Lorenz Schäfers, and Günter Hommel. Disseminating messages among highly mobile hosts based on inter-vehicle communication. In *Intelligent Vehicles Symposium, 2000. IV 2000. Proceedings of the IEEE*, pages 522–527, 2000.

5. Wai Chen and Shengwei Cai. Ad hoc peer-to-peer network architecture for vehicle safety communications. *Communications Magazine, IEEE*, 43(4):100–107, April 2005.

6. Kalyanmoy Deb. *Multi-objective Optimization Using Evolutionary Algorithms*. Wiley-Interscience series in systems and optimization. John Wiley & Sons, 2001.

7. ETSI. Intelligent Transport Systems (ITS), Vehicular Communications, Basic Set of Applications, Definitions. Technical Report 102 638, European Telecommunications Standards Institute, 650 Route des Lucioles, F-06921 Sophia Antipolis Cedex - FRANCE, jun 2009.

8. ISO. Iso 26262-1:2011: Road vehicles – functional safety, 2011.

9. Raj Jain. *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. John Wiley & Sons, 1991.

10. Leslie Lamport. A New Solution of Dijkstra's Concurrent Programming Problem. *Commun. ACM*, 17(8):453 – 455, August 1974.

11. Nils H. Müllner. Simulation of Self-Stabilizing Distributed Algorithms to Determine Fault Tolerance Measures. Diplomarbeit, Department für Informatik, Fakultät II - Informatik, Wirtschafts- und Rechtswissenschaften, Carl von Ossietzky Universität Oldenburg, Oldenburg (Oldb), Germany, April 2007.

12. Nils H. Müllner. *Unmasking fault tolerance: Quantifying deterministic recovery dynamics in probabilistic environments*. BIS-Verlag der Carl von Ossietzky Universität Oldenburg, Ammerländer Heerstr. 114-118, 26111 Oldenburg, February 2014. The books is the revised version of a PhD thesis with the same title.

13. Giovanni Resta, Paolo Santi, and Janos Simon. Analysis of Multi-hop Emergency Message Propagation in Vehicular Ad Hoc Networks. In *Proceedings of the 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '07, pages 140–149, New York, NY, USA, 2007. ACM.

14. IEC SC65A. Functional safety of electrical/electronic/programmable electronic safety-related systems, 1998.

15. Kishor S. Trivedi. *Probability and Statistics with Reliability, Queuing, and Computer Science Applications*. John Wiley & Sons, second edition, 2002.

16. Qing Xu, Tony Mak, Jeff Ko, and Raja Sengupta. Vehicle-to-vehicle Safety Messaging in DSRC. In *Proceedings of the 1st ACM International Workshop on Vehicular Ad Hoc Networks*, VANET '04, pages 19–28, New York, NY, USA, 2004. ACM.

Acknowledgments

This work was partly supported by the German Research Council under grant SFB/TR 14 AVACS and by the funding initiative *Niedersächsisches Vorab* of the Volkswagen Foundation and the Ministry of Science and Culture of Lower Saxony (as part of the Interdisciplinary Research Center on Critical Systems Engineering for Socio-Technical Systems).