Behavior Prediction of Cyber-Physical Systems for Dynamic Risk Assessment

Marta Grobelna

Fraunhofer IKS, Munich, Germany
marta.grobelna@iks.fraunhofer.de

Abstract. Cyber-Physical Systems, such as autonomous vehicles, have the potential for providing more safety by restricting the impact of potentially unreliable human operators. However, ensuring that the system, i.e. the CPS under consideration, will behave safely under any conditions is not straightforward. The complexity of the environment and the system itself, causes uncertainties that need to be considered by the safety measures. The challenge for an autonomous system is to find the optimal trade-off between safety and utility without human intervention. Consequently, such systems has to be self-adaptive and predictive in order to forecast hazardous situations and react to them before the happen. This paper sketches how reachability analysis in combination with game theory can be used to predict risk of hazardous situations.

Keywords: dynamic risk assessment, game theory, reachability analysis, self-adaptation

1 Introduction

According to the US National Highway Traffic Safety Administration, driver's inattention, distractions, and inadequate surveillance are the main reasons for human caused accidents on the roads [13]. This indicates that autonomous Cyber-Physical Systems (CPS)s, such as autonomous vehicles (AV)s, have the potential for providing more safety by restricting the impact of potentially unreliable human operators. Guaranteeing safety of such systems under any conditions is challenging as a full functional specification of the system and its environment is infeasible. First, the high complexity of the environment in which the system operates makes it impossible to consider all factors that influence its behavior. Second, it might be unknown how the factors affect the behavior of the system. For instance, the behavior of human traffic participants represent such factor as their intentions are unknown.

On account of these shortcomings, CPSs have to be *self-adaptive*. This means that such systems have to be able to detect hazards and, if necessary, calculate adequate adaptation steps to counter the hazards on time without human intervention. The self-adaptation process consists of four steps: monitor, analyze, plan, and execute. The system monitors the environment and analyzes the data

to extract information needed to understand the current situation. Subsequently, if necessary, adaptation steps, such as trajectory adaptations, are planned and then executed. Since the actions of the system have an impact on the environment, and changes in the environment have an impact on the behavior of the CPS, there is a feedback loop from the execution step to the monitoring step [1].

In order to plan the adaptation steps on time, and so prevent or recover from an undesired situation, the situation has to be *predicted* [2]. This requires the estimation of future states of the system, i.e. of the CPS which safe behavior needs to be guaranteed, and other agents that are part of the system's environment, e.g. other traffic participants. For this purpose, the system has to use models that reflect the behavior of the agents at a higher level of abstraction, making the required calculations feasible during the run time. After the estimation of the future states, the risk of a hazardous situation can be quantitatively assessed. Here, risk is defined according to ISO 26262 as a 'combination of the probability of occurrence of harm and the severity of that harm' [7]. In the context of AVs, an accident is an example of a hazardous situation.

Unfortunately, the non-determinism of the environment, imprecise measurements and models cause uncertainties that might have an impact on the needed estimates and so on the decision making of the system. In order to consider the uncertainties, they need to quantified, while an optimal trade-off between safety and utility has to be taken into account. On the one hand, an over-cautious treatment of the uncertainties can cause a significant decrease of system's utility and so possibly causing threats to safety. On the other hand, the system must not be too optimistic since this might cause violations of safety requirements and hazards such as accidents [14].

In context of AVs, an important source of information will be the Vehicle-to-X (V2X) communication which will enable wireless exchange of data and information with vehicles, infrastructures, and pedestrians. Even though communication is more robust against environmental circumstances such as weather conditions, there are still issues that need to be considered while using data and information received using V2X communication.

One of the main shortcomings of communication networks is the fact that the system receiving data has to trust that the information is correct. In particular, even in the era of 6G, uncertainty provided by the sensors of the sending system will remain. Further, malfunctioning or deliberate sharing of malicious data, i.e. in case of a cyber-attack, cannot be fully excluded. Further, it has to be considered that there will be a phase where cars that are not capable to communicate via V2X will be present on the roads. Also, intentions of human drivers, pedestrians, and cyclists either cannot be exchanged via V2X. Therefore, the risk assessment function has to be predictive and dynamic, meaning that it has to consider future states of the environment and adapt to the current level of uncertainty that can vary during run time. This paper presents a sketch of an approach for dynamic risk assessment that considers the need for a trade-off between safety and utility of the system while taking into account known and potentially unknown uncertainties.

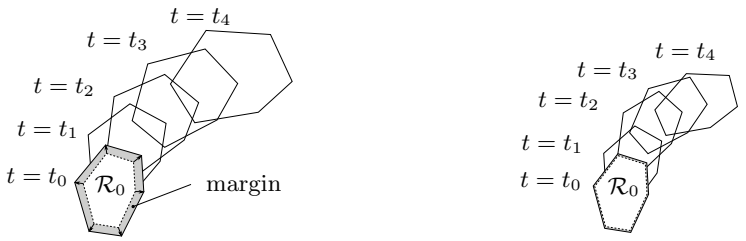
2 Behavior Prediction under Uncertainty

Self-adaptation requires prediction of potentially hazardous events which in turn requires estimation of future state of the system and the agents in its environment. This section presents how future states can be calculated under consideration of uncertainties and interactions among agents.

Three types of uncertainties are defined [4]: aleatory, epistemic and ontological. The aleatory uncertainty concerns the randomness of a process, which is considered to be irreducible. Epistemic uncertainty concerns the discrepancy between the true behavior of a system and its model. Its impact can be reduced when more information about the system is known. The ontological uncertainty is caused by a complete ignorance of a relevant factor in the model. This work focuses on aleatory and epistemic uncertainties.

2.1 Calculating Reachable States

Systems that exhibit continuous behavior can be described in terms of differential equations. In case the system can switch between different modes of dynamics, the system is called hybrid and can be represented by a hybrid automaton where each mode of dynamics is associated with a separate location [6]. Given the model, future states of the system can be calculated using reachability analysis (RA) which is a well-known formal method to iteratively calculate reachable sets of states within a finite time horizon given an initial state. In classical model checking, RA is usually used to estimate if the system fulfills some safety properties. For this purpose, in each iteration step the algorithm estimates if the currently reachable states intersect with the set of states that do not fulfill the desired safety properties [6, 12].



(a) Margin to compensate uncertainty.

(b) Margin for precise information.

Fig. 1: Reachability analysis output for an initial state $\mathcal{R}_0$ and the time horizon $t = [t_0, t_4]$.

Here, the RA is used to calculate the future states of the involved agents. Fig. 1 illustrates an output of a RA. The initial state, denoted by $\mathcal{R}_0$, is represented by a polyhedron and in order to account for aleatory uncertainties, such as perceptual uncertainties, it is over-approximated, meaning that in each dimension of the state, e.g. position or velocity, a margin is added. The successor states are

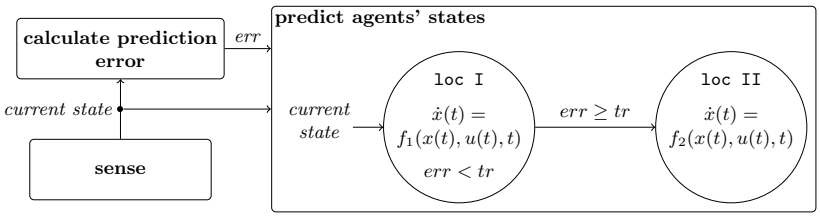


Fig. 2: Model update based on the level of trust.

then calculated by applying the dynamics to the polyhedron that is considered in current iteration step. To consider uncertainty propagation and epistemic uncertainties, in each iteration step an over-approximation is conducted. The resulting new polyhedron is then used as initial state for the next iteration of the algorithm. The system has to calculate the reachable sets of states for each agent within its environment. Based on the estimated future occupancy of the agents, it can plan its adaptation steps if necessary.

The challenge is to over-approximate the states such that on the one hand, all relevant states are considered but on the other hand, the over-approximations are not over-conservative, i.e. the added margin is not too large. In particular, this is important for finding the optimal trade-off between safety and utility of the system. The greater the over-approximations, the more cautious are the predictions, leaving the system less degrees of freedom for adaptations.

One possible way to overcome this challenge is to link the over-approximation magnitude with the prediction error, i.e. the discrepancy between the predicted and the true behavior of the particular agent. Fig. 2 illustrates the idea for a single agent. By sensing, the system gains information about the current state of the other agent. Given the predicted states from the previous cycle, it can estimate the prediction error. To calculate the future states of the system, a hybrid automaton with at least two locations is needed. For instance, one location might reflect defensive behavior and the second aggressive behavior. Further, for each location a different over-approximation margin can be defined. In Fig. 2 the automaton has two locations – `loc I` and `loc II` – each having different dynamics $\dot{f}_1(x, u, t)$ and $\dot{f}_2(x, u, t)$ where x is state, u is control and t is time . As long as the invariant $err < tr$ in `loc I` is satisfied, i.e. when the prediction error denoted by err in Fig. 2 is lower than a certain threshold tr, the system considers `loc I` and is not allowed to enter `loc II` due to the transition guard $err \geq tr$ that prohibits the switch. The output will then look like in Fig. 1b. Otherwise, `loc II` where more coarse over-approximations are conducted so the output might look like in Fig. 1a. Hence, with more precise calculations the system will have more degrees of freedom for planning new trajectories.

2.2 Modeling Agent Interactions

The proposed model update process enables the system to adapt its model to the observed behavior of the corresponding agent. However, using RA all possible

trajectories of the agents are calculated, including those that might not be desired by a certain agent. Further, a trajectory chosen by an agent might depend on the trajectory chosen by the system, and vice versa. In order to consider these dependencies among the agents, dynamic game theory can be applied. Dynamic games occur when a number of agents interact with each other over time while each has its own objective function [3, 8].

In order to apply dynamic game theory for the estimation of future trajectories of a number of agents, several challenges need to be overcome. First, the formulated game need to deal with uncertainties regarding the objective functions of the agents. In particular, each agent might have different objectives and different preferences over multiple objectives. Both, the objectives and the priorities, might be unknown. Further, constraints such as traffic rules have to be respected, however, temporal contempt should be considered since situations might occur where agents will violate them in order to achieve a higher priority objective such as collision avoidance. Consequently, temporal relaxation of constraints should be integrated.

Another problem might be the uncertainty representation in context of reachability analysis. Recall, that the output of RA is a set of polyhedra which would be the input for the algorithm that uses the game theoretic approach.

Finally, the problem of dimensionality needs to be addressed. In order to calculate optimal strategies for multiple agents, a system of (partial) differential equations needs to be solved. Unfortunately, numerical methods suffer from the curse of dimensionality, meaning that the calculation time increases with the dimensionality of the system and so the number of considered agents. In recent years, solving high-dimensional differential equations using machine learning has received more attention. There is a number of encouraging approaches [5, 9, 10]. In [15] the author illustrated the effectiveness of so called Physics-Informed Neural Networks [11], however the evaluated examples were theoretical and not as complex as AVs. Thus, further research in this area is needed.

2.3 Illustrative Example

In order to illustrate the proposed approach potential and the challenges that lie ahead, a simplistic example is presented. Consider a merging scenario with two vehicles $\mathcal{V}_1$, the ego vehicle, driving on an acceleration lane and vehicle $\mathcal{V}_2$ driving on the adjacent lane which $\mathcal{V}_1$ wants to enter. A state of a vehicle $\mathcal{V}_i$, denoted by x_i where $i = \{1, 2\}$, is defined as $x_i(t) = (s_i(t), v_i(t), a_i(t))$, where t is the time, $s_i(t)$ is the position, $v_i(t)$ is the velocity. Acceleration $a_i(t)$ is a control value, and is the only parameter in the dynamics of both vehicles. In this example the initial states of the both vehicles are given by

$$x_1(0) = \left(50\,\mathrm{m},\, 27.8\,\mathrm{m/s},\, 0\,\mathrm{m/s^2}\right) \text{ and } x_2(0) = \left(25\,\mathrm{m},\, 34.7\,\mathrm{m/s},\, 0\,\mathrm{m/s^2}\right), \quad (1)$$

The ego vehicle is allowed to initiate the lane change if and only if it can maintain a safe distance to $\mathcal{V}_2$. Further, it is assumed that the vehicles cannot communicate with each other. Hence, $\mathcal{V}_1$ does not know if $\mathcal{V}_2$ will let it merge in front of it.

Table 1: Defensive dynamics for $\mathcal{V}_2$.

$\mathcal{V}_1$ \ $\mathcal{V}_2$	$a_2 = -2$	$a_2 = 0$	$a_2 = 2$
$a_1 = -2$	(-120.3, -394.3)	(-70.9, -430.3)	(-58.64, -471.5)
$a_1 = 0$	(-103.9, -354.3)	(-116.3, -390.3)	(-66.9, -431.5)
$a_1 = 2$	(-61.0, -314.3)	(-99.9, -350.3)	(-112.3, -391.5)

Table 2: Aggressive dynamics for $\mathcal{V}_2$.

$\mathcal{V}_1$ \ $\mathcal{V}_2$	$a_2 = 0$	$a_2 = 4$
$a_1 = -2$	(-70.9, -263.6)	(-52.0,6.6)
$a_1 = 0$	(-116.3, -223.6)	(-54.6, -295.6)
$a_1 = 2$	(-99.9, -183.6)	(-62.9, -255.6)

The model used by $\mathcal{V}_1$ to predict the behavior of $\mathcal{V}_2$ consists of two locations, where in each the following well-known equations of motion are contained

$$s_i(t) = s_{i,0} + v_{i,0} \cdot t + \frac{1}{2} \cdot a_i(t) \cdot t^2,\ v_i(t) = v_{i,0} + a_i(t) \cdot t, \tag{2}$$

where $s_{i,0}$ is the initial position and $v_{i,0}$ the initial velocity. The first location reflects defensive dynamics where $\mathcal{V}_2$ respects the road speed limit of 36.1 m/s and its control value a_2 is restricted by the interval $[-2, 2]\,\mathrm{m/s^2}$. The second location reflects aggressive dynamics where $\mathcal{V}_2$ does not respect the speed limit and a_2 is within the interval $[0, 4]\,\mathrm{m/s^2}$.

It is assumed that $\mathcal{V}_1$ wants to plan its trajectory for the next 2 s. Further, for sake of simplicity, it is assumed that it has perfect information and so calculation of future states reduces to evaluation of (2). The objective function of $\mathcal{V}_\in$ is given by

$$\max J_2(x_1, x_2, a_1, a_2, t) = -\underbrace{(v_{\max} - v_2(t))}_{\text{maximize velocity}} - 10 \cdot \underbrace{(d_{\text{safe}} - (s_1(t) - s_2(t)))}_{\text{maximize distance}}. \tag{3}$$

The first term expresses that $\mathcal{V}_2$ wants to maximize its velocity and the second term expresses that it wants to maximize the distance to $\mathcal{V}_1$ is case it follows $\mathcal{V}_2$. The second term is scaled by factor 10 to model priority of safe distance over optimal velocity. In defensive mode d_{safe} for $\mathcal{V}_2$ is 50 m and in aggressive mode 33.3 m.

The objective of $\mathcal{V}_1$ is to minimize the time until merging so its first objective is to minimize the function obtained by solving $(s_1(t) - s_2(t))^2 = d_{\text{safe}}^2$ for t. Further, it also wants to maximize its velocity and maintain safe distance which is always 50 m.

Now, a strategic game can be formulated where $\mathcal{V}_1$ is the row player and $\mathcal{V}_2$ is the column player. The objective functions were formulated such that both need to be maximized. The goal is to calculate Nash equilibria where neither player can improve its utility by changing its strategy.

Tab. 1 contains the payoff matrix obtained for the game where defensive behavior of $\mathcal{V}_2$ was assumed. This game has only one equilibrium that is marked blue in Tab. 1. As expected, $\mathcal{V}_2$ will let $\mathcal{V}_1$ merge in front of it, since the best output for both vehicles is achieved when $\mathcal{V}_1$ accelerates while $\mathcal{V}_2$ decelerates. However, the output of the game is different when $\mathcal{V}_1$ has the information that $\mathcal{V}_2$'s initial position is at 46.84 m. In this case the game has four equilibria indicating that state uncertainty has a significant impact on the output of the game and sophisticated solution to this problem needs to be found.

In case aggressive behavior of $\mathcal{V}_2$ is assumed, the payoffs contained in Tab. 2 are obtained. This game has again a single Nash equilibria, which shows that $\mathcal{V}_2$ will accelerate and not allow $\mathcal{V}_1$ to merge in front of it.

While being simplistic and only considering two vehicles with perfect information and limited choices, this illustrative example shows the potential for analysis and optimization of the proposed approach. Regarding the challenges that lie ahead, defining scalable models (several vehicles/choices), improving upon imperfect information by enabling communication to make better decisions, and modeling uncertainties (e.g., with Bayesian game models), among others, will be tackled.

3 Risk Assessment

The evaluation of the models will enable timely detection of hazardous situations. In particular, a collision is detected as soon as the set of reachable states of the system intersects with a reachable set of states of any other agent surrounding it. Due to the fact that during the computation of the reachable sets of states over-approximations are made, the criterion of intersection might be too hard. Instead, the risk of that intersection should be calculated. This is a relaxation of the intersection criterion which will avoid over-cautious behavior of the system and so improve its utility. Consequently, in each iteration step of the RA, instead of a simple intersection check the risk of that intersection is calculated.

Assume n agents ($n \in \mathbb{N}$ and $n < \infty$) in the system's environment that have to be considered. Denote the sets of states that are reachable by the n agents within a time horizon t_h by $\mathcal{R}_{\text{all}}$. Since RA is iterative, the time is discretized using a time step size of t_s. Hence, for each agent the system calculated $m := t_h/t_s$ reachable sets of states. The set that an agent $a \in \{1, \ldots, n\}$ will reach at time $t_k \in \{t_1,\ t_2, t_3, \ldots, t_m\}$ is denoted by $\mathcal{R}_a^{t_k}$. Then the overall risk R of an intersection is given by

$$R(\mathcal{R}_{\text{all}}) := \sum_{t=1}^{m} f_{\text{s}} \left(\bigcap_{a=1}^{n} \mathcal{R}_a^{t_k} \right) \cdot f_{\text{p}} \left(\bigcap_{a=1}^{n} \mathcal{R}_a^{t_k} \right), \tag{4}$$

where $f_{\text{s}}(\cdot)$ is a function that estimates the severity of an intersection, $f_{\text{p}}(\cdot)$ is a function that estimates the probability of that intersection. Since for the evaluation of (4), the system will always consider the current information about the agents, and the behavior model for each agent can be updated, the system estimates the risk in a dynamical way under consideration of dependencies among agents due to the game theoretic approach integrated in the RA approach.

Based on the calculated risk of each trajectory that the system might choose to proceed with, it can decide which one is the safest. A high level or risk means that the safety of the system is low, while a low level of risk means a high level of safety. Hence, risk is inversely proportional to safety. Note that the same holds for the utility of a system which can also be describe by a function. The higher the risk of a hazard, the less utility can be achieved.

4 Conclusion

This paper proposes an approach that enables CPSs to be self-adaptive and account for uncertainties while finding suitable trade-offs between safety and utility of the system. In order to plan optimal adaptation steps on time, the system has to predict its own and the environments' states. This requires models able to approximate the behavior of other agents while being computationally affordable. Besides aleatory uncertainties, epistemic uncertainties as well as agent interactions have to be considered. Combining RA with game theory will allow conducting precise predictions of hazardous situations and so enable the CPSs to react to such situations on time and potentially in a more sophisticated way.

References

1. Arcaini, P., Riccobene, E., Scandurra, P.: Modeling and analyzing MAPE-K feedback loops for self-adaptation. In: 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (2015)
2. Calinescu, R., Ghezzi, C., Kwiatkowska, M., Mirandola, R.: Self-adaptive software needs quantitative verification at runtime. Communications of the ACM (2012)
3. Engwerda, J., Reddy, P.: A positioning of cooperative differential games. 5th Conference on Performance Evaluation Methodologies and Tools (2011)
4. Gansch, R., Adee, A.: System theoretic view on uncertainties. In: Design, Automation Test in Europe Conference Exhibition (2020)
5. Han, J., Jentzen, A., Ee, W.: Solving high-dimensional partial differential equations using deep learning. Proceedings of the National Academy of Sciences (2017)
6. Henzinger, T.: The theory of hybrid automata. In: Proceedings 11th Annual IEEE Symposium on Logic in Computer Science (1996)
7. International Organization for Standardization: Road vehicles — functional safety. Standard ISO 26262-1:2018, ISO (2019)
8. Jafary, B., Rabiei, E., Diaconeasa, M., Masoomi, H., Fiondella, L., Mosleh, A.: A survey on autonomous vehicles interactions with human and other vehicles. In: 14th Conference on Probabilistic Safety Assessment and Management (2018)
9. Long, Z., Lu, Y., Ma, X., Dong, B.: PDE-net: Learning PDEs from data. arXiv:1710.09668 (2018)
10. Nakamura-Zimmerer, T., Gong, Q., Kang, W.: Adaptive deep learning for high-dimensional hamilton-jacobi-bellman equations. arXiv:1907.05317 (2020)
11. Raissi, M., Perdikaris, P., Karniadakis, G.E.: Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational Physics (2019)
12. Riedmaier, S., Danquah, B., Schick, B., Diermeyer, F.: Unified framework and survey for model verification, validation and uncertainty quantification. Archives of Computational Methods in Engineering (2020)
13. Singh, S.: Critical reasons for crashes investigated in the national motor vehicle crash causation survey. Traffic Safety Facts - Crash Stats (2015)
14. Trapp, M., Schneider, D., Weiss, G.: Towards safety-awareness and dynamic safety management. In: 2018 14th European Dependable Computing Conference (2018)
15. Winkler, S.N.: A framework including artificial neural networks in modelling hybrid dynamical systems. Ph.D. thesis, TU Wien (2020)